Applications: Approximation

Theo Diamandis

January 19, 2023

In this lecture, we will look at a variety of approximation problems, which come up in almost every field under different names, including 'reconstruction' in signal processing, 'regression' or 'estimation' in statistics and machine learning, 'design' in several engineering fields, and so on. This lecture follows [BV04, Ch. 6].

1 Approximation

In class, we've already seen a number of problems of the form

$$\text{minimize} \quad \|Ax - b\|, \tag{1}$$

where the problem data are $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ with $m \geq n$. We will call

$$r = Ax - b$$

the *residual*. The solution $x^\star$ to (1) has several interpretations:

- **geometry:** $Ax^\star$ is the point in $\mathcal{R}(A)$ closest to b, as measured by $\|\cdot\|$.
- **estimation:** $x^\star$ is the maximum likelihood estimate of x under a linear measurement model $y = Ax + v$ where v is the measurement noise. Note that the choice of norm implies a prior on the distribution of v (or a prior on v would imply the 'correct' norm to use).
- **design:** $Ax^\star$ is a design, where x are the design variables and b is the target design.

We've already seen examples of this problem for the ℓ_1, ℓ_2, and ℓ_∞ norms. In this lecture, we'll examine the properties each of these norms (and other, more general *penalty functions*), induce in the residual.

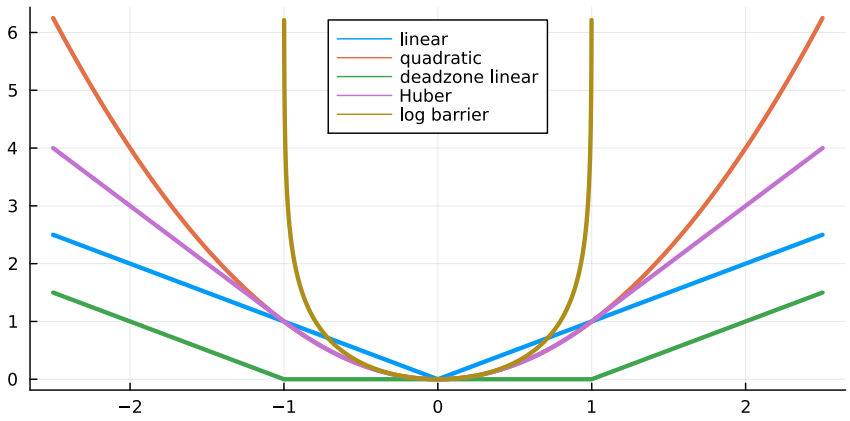

Figure 1: Penalty functions.

Penalty function approximation. More generally, we can consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \phi(r_m) \\ \text{subject to} & r = Ax - b, \end{array} \tag{2}$$

where $\phi : \mathbf{R} \to \mathbf{R}$ is a convex penalty function. Many norms fit into this framework. For example, if we take $\phi(u) = u^2$, then we recover an equivalent problem to ℓ_2 norm minimization. We can even approximate the ℓ_∞ norm by taking $\phi(u) = e^u$ (recall the softmax function). Other penalty functions include deadzone-linear with width a,

$$\phi(u) = \max\{0, |u| - a\},$$

the Huber penalty function with parameter a,

$$\phi(u) = \begin{cases} u^2 & \text{if } |u| \leq a, \\ a(2|u| - a) & \text{otherwise,} \end{cases}$$

and the log-barrier function with limit a,

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

Think about what each of these penalties promotes in the residuals r. Figure 1 plots each of these penalty functions, and figure 2 shows the histogram of residuals for a randomly generated problem. Of course, we can consider asymmetric penalty functions as well.

Robust norm approximation. We consider the norm approximation problem where data matrix A is not known exactly. Instead, we know that

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

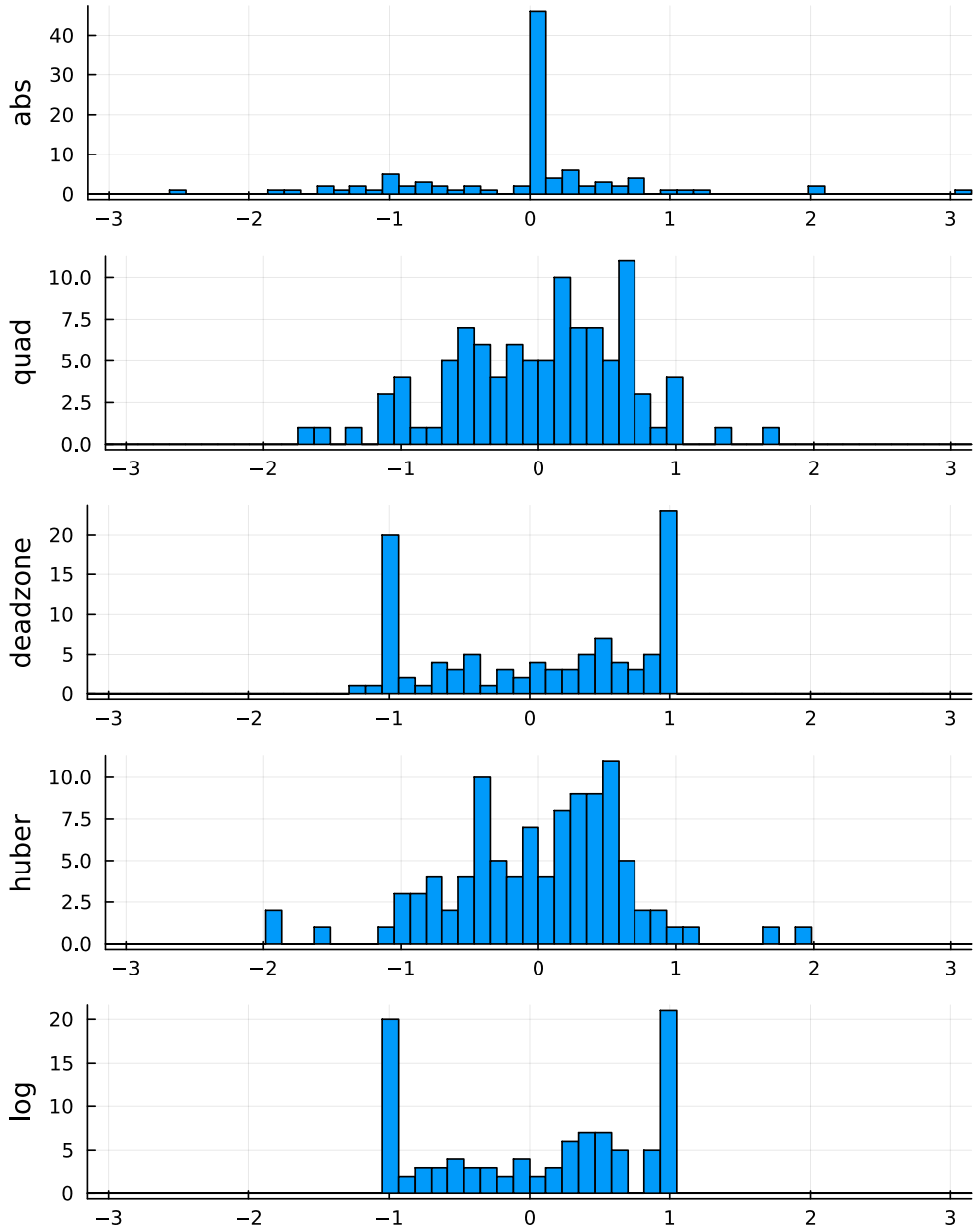

Figure 2: Residuals for the approximation problem under a number of penalty functions.

which we assume is nonempty and bounded. The *robust approximation problem* is to minimize the worst case error over the uncertainty set $\mathcal{A}$:

$$\text{minimize} \quad \sup_{A \in \mathcal{A}} \|Ax - b\|_2.$$

While this problem is always convex, its tractability depends on the norm used and the description of the uncertainty set $\mathcal{A}$. If this set is finite, the problem is easy to solve. Here, we consider the case where $\mathcal{A}$ is a norm ball

$$\mathcal{A} = \{\bar{A} + U \mid \|U\| \leq a\},$$

where we take $\|\cdot\|$ to be the spectral norm (*i.e.,* the maximum signal value). The worst case error is attained for $U = auv^T$ where

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

The resulting worst-case error is

$$\|\bar{A}x - b\|_2 + a\|x\|_2.$$

Thus, solving this robust approximation problem is an SOCP. (Note that it is not a regularized least squares problem since we do not square both terms.)

Quantile regression. Consider the *tilted* ℓ_1 penalty with parameter $\tau \in (0, 1)$,

$$\phi(u) = \tau(u)_+ + (1 - \tau)(u)_- = (1/2)|u| + (\tau - 1/2)u.$$

The *quantile regression* problem chooses x to minimize the sum of these penalties. Consider what happens if we choose $\tau = 0.5$. Then we recover the ℓ_1 regression problem, which assigns an equal penalty to over and underestimating the target b with our predictor Ax. Furthermore, we'd expect around half of the residuals (on the 'training data') $r = Ax - b \in \mathbf{R}^m$ to be negative and half to be positive. However, if we set $\tau = 0.9$, there is a $9\times$ greater penalty for over-estimating b than underestimating b. Roughly speaking, we expect that

$$\tau|\{i \mid r_i > 0\}| = (1 - \tau)|\{i \mid r_i < 0\}|.$$

The τ-quantile of the optimal residuals is zero. Solving this problem with a number of τ's gives us a set of solutions $\{x^\tau\}_{\tau \in T}$ which provide predictors for different quantiles of the data. This can be useful if we want not only a point estimate but also upper and lower bounds. **[TD: Maybe add concrete example here or homework problem.]**

1.1 Least norm approximation.

Sometimes, we have a data matrix $A \in \mathbf{R}^{m\times n}$ such that $m \leq n$. In this case, there may be many solutions to $Ax = b$. We will instead solve the least norm problem

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

which aims to find the smallest x (wrt the chosen norm) that is a solution to $Ax = b$. Again, the solution $x^\star$ has several interpretations:

- **geometry:** $x^\star$ is the point in the set $\{x \mid Ax = b\}$ with minimum distance to 0.
- **estimation:** $x^\star$ is the smallest estimate consistent with the (perfect) measurements $b = Ax$.
- **design:** $x^\star$ is the most 'efficient' design that satisfies the requirements.

Recall the basis pursuit problem, where we seek to find the sparsest vector x consistent with the measurements $Ax = b$. The ℓ_1 norm is used as a convex approximation to the cardinality.

Example: Colorization. *(From Convex Optimization additional exercises.)* A $m \times n$ color image is represented as three matrices of intensities R, G, $B \in \mathbf{R}^{m\times n}$, with entries in $[0, 1]$, representing the red, green, and blue pixel intensities, respectively. A color image is converted to a monochrome image, represented as one matrix $M \in \mathbf{R}^{m\times n}$, using

$$M = 0.299R + 0.587G + 0.114B,$$

where the weights come from the 'perceived brightness' of each color. In *colorization*, we are given the monochrome version of an image M and the color values at a handful of pixels. Our goal is to guess the colors at the rest of the pixels. Since this problem is underdetermined, we will do so by solving a least norm problem, where we minimize the *total variation* of (R, G, B), which is an approximation of the spatial gradient, defined as

$$\mathbf{tv}(R, G, B) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} \left\| \begin{bmatrix} R_{ij} - R_{i,j+1} \\ R_{ij} - R_{i+1,j} \\ G_{ij} - G_{i,j+1} \\ G_{ij} - G_{i+1,j} \\ B_{ij} - B_{i,j+1} \\ B_{ij} - B_{i+1,j} \end{bmatrix} \right\|_2 .$$

Thus, we can colorize an image by solving the optimization problem

$$\begin{array}{ll} \text{minimize} & \mathbf{tv}(R, G, B) \\ \text{subject to} & M = 0.299R + 0.587G + 0.114B \\ & R_i = R_i^{\text{known}}, \qquad i \in I_{\text{known}} \\ & G_i = G_i^{\text{known}}, \qquad i \in I_{\text{known}} \\ & B_i = B_i^{\text{known}}, \qquad i \in I_{\text{known}}. \end{array}$$

Figure 3: Original image (left), monochrome image with a few randomly colored pixels (center) and its colorized verison (right).

An example using this technique is shown in Figure 3.

2 Regularized Approximation

Often, we want to tradeoff between minimizing $\|Ax-b\|$ and minimizing $\|x\|$. This *regularized approximation* problem can be phrased as

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

where the two norms may be different. This problem has several interpretations

- **estimation:** We have a noisy linear measurement model $y = Ax + v$ and prior knowledge that $\|x\|$ is small.
- **design:** The linear model $y = Ax$ is only valid for small x (*e.g.,* the first-order approximation of a nonlinear function), or a small x is cheaper to build.
- **robust approximation:** A good approximation $Ax \approx b$ is less sensitive to errors in A than a good approximation with large x.

Recall from last lecture that these problems are solved by scalarization, *i.e.,* we solve the problem

$$\text{minimize} \quad \|Ax-b\| + \lambda\|x\|$$

for varying values of $\lambda > 0$. Note that in regression problems, we often do not include the offset variable in the regularization.

Example: signal reconstruction. Consider a signal $x \in \mathbf{R}^T$ which we'd like to recover given a noise-corrupted version of the signal x_{cor}. Here, we are considering one dimensional signals (*e.g.,* audio signals). This reconstruction can be phrased as the optimization problem

$$\text{minimize} \quad \|x - x_{\text{cor}}\| + \lambda\phi(x),$$

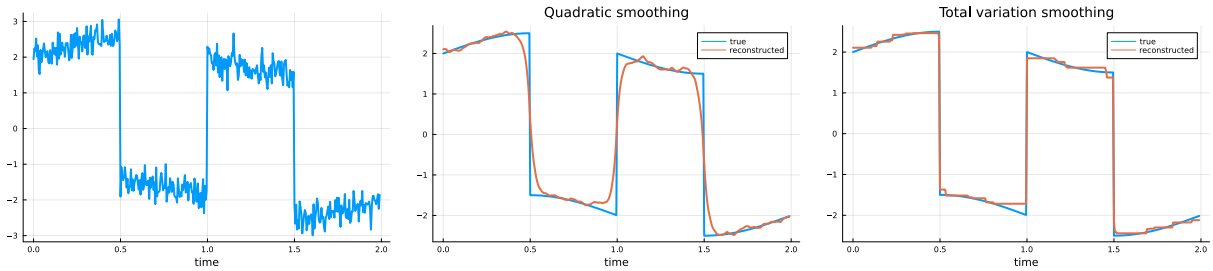

Figure 4: Original signal (left), quadratic smoothing reconstruction (center), and total variation reconstruction (right). The total variation penalty better preserves 'jumps' in the original signal.

where we add a penalty to induce desired properties of the reconstruction (usually corresponding to priors we have about the signal). For example, it is common to assume the signal does not vary too rapidly compared to its sampling rate; *i.e.,* we expect $x_i \approx x_{i+1}$. In this case, we can use the quadratic smoothing penalty

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

where $D \in \mathbf{R}^{n-1\times n}$ is a first difference matrix. Sometimes, however, the original signal varies rapidly as well (*e.g.,* systems sending bits where a 1 is one value and a 0 is another). Quadratic smoothing would dampen rapid variations in the reconstruction. A better penalty in this case is the total variation penalty

$$\phi_{\text{tv}}(x) = \sum_{i=1}^{n-1}|x_{i_1} - x_i| = \|Dx\|_1.$$

Reconstructions using each of these penalties is given in Figure 4, where we use the ℓ_2 penalty on $\|x - x_{\text{cor}}\|$. Of course, we could use a combination of the two penalties, which likely would yield better results in practice with some tuning of the parameters.

3 Example: Minimax polynomial fitting

Consider some function $y(t)$ for $\alpha \leq t \leq \beta$. We sample this function at points $t_1, \ldots, t_k$, which have corresponding functions values $y_1, \ldots, y_k$. Our goal is to fit a rational function $p(t)/q(t)$ to given data, while constraining the denominator $q(t)$ to be positive on the interval $[\alpha, \beta]$.[1] We parameterize p and q by vectors $a \in \mathbf{R}^{m+1}$ and $b \in \mathbf{R}^n$:

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

[1]Polynomial approximations are often used in practice. For example, check out the `exp` implementation in Julia at `https://github.com/JuliaLang/julia/blob/17cfb8e65ead377bf1b4598d8a9869144142c84e/base/special/exp.jl#L188`. which (with a few extra tricks) only needs a degree three polynomial to get floating point accuracy.

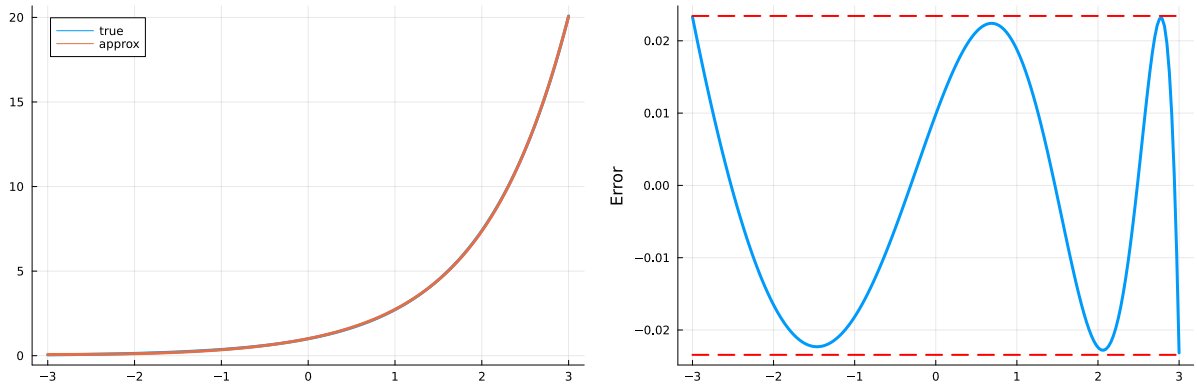

Figure 5: The function compared to its polynomial approximation (left) and the error between these two curves (right). The dashed lines indicate $\pm u$.

We want to find a and b that provide the best minimax rational fit to the data, *i.e.,* that solve

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

This problem is not convex; however, it is quasiconvex. Note that

$$\max_{i=1,\ldots,k} \left| \frac{p(t_i)}{q(t_i)} - y_i \right| \leq t \iff |p(t_i) - q(t_i)y_i| \leq tq(t_i) \qquad \text{for } i = 1, \ldots, k,$$

where we use the fact that $q > 0$. This allows us to solve this problem via a sequence of the convex feasibility problem

$$\begin{array}{ll} \text{find} & a, b \\ \text{s.t.} & |p(t_i) - q(t_i)y_i| \leq tq(t_i) \qquad i = 1, \ldots, k, \end{array}$$

which is parameterized by t.

Now, we consider a specific instance of approximating the exponential function on the interval $[-3, 3]$. The data is

$$t_i = -3 + 6(i-1)/(k-1), \qquad y = e^{t_i}, \qquad i = 1, \ldots, k,$$

where $k = 201$. We consider a function $f(t)$ of the form

$$f(t) = \frac{a_0 + a_1 t + a_2 t^2}{1 + b_1 t + b_2 t^2},$$

where we require that $1 + b_1 t_i + b_2 t_i^2 > 0$ for all $i = 1, \ldots, k$. Solving this problem using bisection method to a tolerance of 0.001 yields the approximation in Figure 3. The final lower and upper bounds on the optimal value are $(l, u) = (0.0224609375, 0.0234375)$.

References

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.