II. Regression using Least Squares

The fundamental problem in supervised machine learning is to fit a function to a series of point evaluations. We are given data

$$(\boldsymbol{t}_1, y_1), \ldots, (\boldsymbol{t}_M, y_m), \quad \text{with } \boldsymbol{t}_m \in \mathbb{R}^D \text{ and } y_m \in \mathbb{R},$$

and want to find a function $f : \mathbb{R}^D \to \mathbb{R}$ such that

$$f(\boldsymbol{t}_m) \approx y_m, \quad m = 1, \ldots, M. \tag{1}$$

When the y_m (and the range of f) are continuous-valued, the process of fitting such an f is called **regression**. This is a terrible name whose English meaning is only tangentially related to this problem. But the terminology has become ingrained through 130+ years of use, so we are stuck with it.

We need two key ingredients to putting the task in (1) on firm mathematical ground, a model function class and a loss function.

Function class. For the problem (1) to make sense, our search needs to be limited to a class of functions $\mathcal{F}$. For example, we might restrict f to be polynomial, or twice differentiable, etc. Choosing this $\mathcal{F}$ is a modeling problem, as it basically encodes what we believe to be true (or reasonably close to true) about the function we are trying to discover. It also has to be something we can compute with. Examples below will take $\mathcal{F}$ to be an appropriately chosen Hilbert space, or a subspace of a Hilbert space; these will both lead to algorithms that lean heavily on linear algebra.

In some cases, this $\mathcal{F}$ is chosen implicitly. This is the case in modern multi-layer neural networks, where the structure for computing the function is defined, but it is hard to describe exactly the class of functions they can represent.

With the choice of $\mathcal{F}$ fixed, the problem (1) now becomes

$$\text{find } f \in \mathcal{F} \text{ such that } f(\boldsymbol{t}_m) \approx y_m, \quad m = 1, \ldots, M. \tag{2}$$

Loss function. This penalizes the deviation of the $f(\boldsymbol{t}_m)$ from the y_m. Given a loss function $\ell(\cdot,\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ that quantifies the penalty for the deviation at a single sample location, we can assign a (positive) numeric score to the performance of every candidate f by summing over all the samples. This allows us to write (2) more precisely as an optimization problem:

$$\underset{f \in \mathcal{F}}{\text{minimize}} \sum_{m=1}^{M} \ell(f(\boldsymbol{t}_m), y_m) \tag{3}$$

There are again many loss functions you might consider, and depending on the context, some might be more natural than others. We will focus almost all of our efforts on the **squared loss** $\ell(u,v) = |u-v|^2$. Then (3) becomes

$$\underset{f \in \mathcal{F}}{\text{minimize}} \sum_{m=1}^{M} |y_m - f(\boldsymbol{t}_m)|^2. \tag{4}$$

We will see below that this choice, coupled with a subspace or Hilbert space model for $\mathcal{F}$, allows us to completely solve the regression problem using linear algebra.

Linear regression

Perhaps the most classical choice for the function class $\mathcal{F}$ is that it contains all linear functions on $\mathbb{R}^D$. A function $f : \mathbb{R}^D \to \mathbb{R}$ is **linear** if

$$f(\alpha \boldsymbol{t}_1 + \beta \boldsymbol{t}_2) = \alpha f(\boldsymbol{t}_1) + \beta f(\boldsymbol{t}_2), \quad \text{for all } \alpha, \beta \in \mathbb{R},\ \boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{R}^D.$$

It is a fact[1] that every linear functional on $\mathbb{R}^D$ is uniquely represented by a vector $\boldsymbol{c}_f \in \mathbb{R}^D$, where

$$f(\boldsymbol{t}) = \langle \boldsymbol{t}, \boldsymbol{w}_f \rangle = \boldsymbol{t}^{\mathrm{T}} \boldsymbol{w}_f.$$

So given the $\{(\boldsymbol{t}_m, y_m)\}$, we want to find $\boldsymbol{w}$ such that

$$y_m \approx \boldsymbol{t}_m^{\mathrm{T}} \boldsymbol{w},$$

and using the squared-loss to measure the mismatch, we have the optimization problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^D}{\text{minimize}} \; \sum_{m=1}^{M} \left| y_m - \boldsymbol{t}_m^{\mathrm{T}} \boldsymbol{w} \right|^2.$$

If we stacking up the $\boldsymbol{t}_m$ as rows in a $M \times D$ matrix $\boldsymbol{A}$, the sum in the optimization program above becomes $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_2^2$, where

$$\boldsymbol{A} = \begin{bmatrix} - & \boldsymbol{t}_1^{\mathrm{T}} & - \\ - & \boldsymbol{t}_2^{\mathrm{T}} & - \\ & \vdots & \\ - & \boldsymbol{t}_M^{\mathrm{T}} & - \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}.$$

We can then find the best linear function on $\mathbb{R}^N$ by solving the finite-dimensional least-squares problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^D}{\text{minimize}} \, \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_2^2.$$

Given a solution $\hat{\boldsymbol{w}}$ to the above, we have the best linear function $\hat{f}$, where

$$\hat{f}(\boldsymbol{t}) = \boldsymbol{t}^{\mathrm{T}} \hat{\boldsymbol{w}} = \hat{w}_1 t_1 + \hat{w}_2 t_2 + \cdots + \hat{w}_D t_D.$$

[1] The name by which this fact goes is the *Riesz Representation Theorem*, which we will encounter a little later in the course.

It is often the case that we want to also want to add a constant offset to the above, and find an **affine** function of the form

$$f(\boldsymbol{t}) = w_0 + w_1 t_1 + \cdots + w_D t_D.$$

This is done simply by adding a column of all ones to $\boldsymbol{A}$:

$$\boldsymbol{A}' = \begin{bmatrix} 1 & - & \boldsymbol{t}_1^{\mathrm{T}} & - \\ 1 & - & \boldsymbol{t}_2^{\mathrm{T}} & - \\ \vdots & & \vdots & \\ 1 & - & \boldsymbol{t}_M^{\mathrm{T}} & - \end{bmatrix},$$

and then solving

$$\underset{\boldsymbol{w}' \in \mathbb{R}^{D+1}}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{A}'\boldsymbol{w}'\|_2^2. \tag{5}$$

Nonlinear regression using a basis

It is easy to generalize what is in the previous section to fit nonlinear functions to data. What is interesting is that we still end up with a very similar linear least-squares problem.

We do this by letting $\mathcal{F}$ be a finite-dimensional subspace (of a Hilbert space) spanned by a set of specified basis functions. Given a set of building blocks $\boldsymbol{\psi}_n : \mathbb{R}^D \to \mathbb{R}$ (i.e. basis functions), the model implicit here is that our target function $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}$ is (at least approximately) in the subspace spanned by $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N$, that is, there exists $\{x_n\}$ such that

$$f(\boldsymbol{t}) = \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}).$$

Fitting a function in this space $\mathcal{F}$ is the same as fitting a $\boldsymbol{x} \in \mathbb{R}^N$ such that

$$y_1 \approx \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}_1), \quad y_2 \approx \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}_2), \quad \ldots \quad y_M \approx \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}_M).$$

With the least-squares loss, we want to solve

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}} \quad \sum_{m=1}^{M} \left| y_m - \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}_m) \right|^2 = \sum_{m=1}^{M} \left| y_m - \boldsymbol{\Psi}(\boldsymbol{t}_m)^{\mathrm{T}} \boldsymbol{x} \right|^2, \quad (6)$$

where $\boldsymbol{\Psi}(\cdot) : \mathbb{R}^D \to \mathbb{R}^N$ is

$$\boldsymbol{\Psi}(\boldsymbol{t}) = \begin{bmatrix} \psi_1(\boldsymbol{t}) \\ \psi_2(\boldsymbol{t}) \\ \vdots \\ \psi_N(\boldsymbol{t}) \end{bmatrix}.$$

Constructing the $M \times N$ matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \begin{bmatrix} - & \boldsymbol{\Psi}(\boldsymbol{t}_1)^{\mathrm{T}} & - \\ - & \boldsymbol{\Psi}(\boldsymbol{t}_2)^{\mathrm{T}} & - \\ & \vdots & \\ - & \boldsymbol{\Psi}(\boldsymbol{t}_M)^{\mathrm{T}} & - \end{bmatrix} = \begin{bmatrix} \psi_1(\boldsymbol{t}_1) & \psi_2(\boldsymbol{t}_1) & \cdots & \psi_N(\boldsymbol{t}_1) \\ \psi_1(\boldsymbol{t}_2) & \psi_2(\boldsymbol{t}_2) & \cdots & \psi_N(\boldsymbol{t}_2) \\ \vdots & & \ddots & \\ \psi_1(\boldsymbol{t}_M) & \psi_2(\boldsymbol{t}_M) & \cdots & \psi_N(\boldsymbol{t}_M) \end{bmatrix}, \quad (7)$$

the optimization program (4) becomes

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2. \quad (8)$$

From a solution $\hat{\boldsymbol{x}}$ to the program above, we can synthesize the solution to (6) as

$$\hat{f}(\boldsymbol{t}) = \sum_{n=1}^{N} \hat{x}_n \, \psi_n(\boldsymbol{t}).$$

Even though we are ultimately recovering a nonlinear function of a continuous variable, introducing a basis allows us to put the nonlinear regression problem into the exact same computational framework as linear regression.

Example

Play around with the code in `regression_examples.m`. Here, we take M (noisy) samples of an underlying function, and then can perform regression using N basis functions .... code for polynomials, "bumps", and Fourier (trigonometric polynomial) are all included.

Notice that increasing N makes the class richer, and corresponds to adding columns in $\boldsymbol{A}$. This increases our ability to match the samples, but also increases the risk of "over fitting" the model.

The least-squares problem

We have seen that in both the linear regression case (5) and the nonlinear regression case (5), the problem reduces to same form. We will look at problems like this multiple times in this course, and it is worth starting to study this kind of problem from the perspective of linear algebra.

We start with the following fundamental result:

Any solution $\hat{\boldsymbol{x}}$ to

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \tag{9}$$

must obey the **normal equations**

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}. \tag{10}$$

It is a fact[2] that $g(\boldsymbol{x}) = \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2$ is a convex, differentiable function on all of $\mathbb{R}^N$. Thus a necessary condition for $\hat{\boldsymbol{x}}$ to be minimizer of $g(\boldsymbol{x})$ is that the gradient (with respect to $\boldsymbol{x}$) is equal to zero at $\hat{\boldsymbol{x}}$, $\nabla g(\hat{\boldsymbol{x}}) = \boldsymbol{0}$. We have

$$\begin{aligned}\nabla\left(\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2\right) &= \nabla\left(\|\boldsymbol{y}\|_2^2 - 2\,\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \|\boldsymbol{A}\boldsymbol{x}\|_2^2\right)\\ &= -2\,\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x},\end{aligned}$$

which is $\boldsymbol{0}$ when (10) holds.

There are several natural questions at this point. First, is there any guarantee that (10) even has a solution? Are there conditions under

[2] Prove this at home!

which the solution is guaranteed to be unique or non-unique? And if the solution is non-unique, what should we do?

We will answer these questions using the following two facts from linear algebra. For any $M \times N$ matrix $\boldsymbol{A}$,

- $\text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Null}(\boldsymbol{A})$, and
- $\text{Col}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Row}(\boldsymbol{A})$.

Proof of these facts can be found in the Technical Details section.

It is now easy to argue that:

1. The system (10) always has a solution, no matter what $\boldsymbol{A}$ and $\boldsymbol{y}$ are. This follows immediately from the fact that

   $$\boldsymbol{A}^{\text{T}}\boldsymbol{y} \in \text{Row}(\boldsymbol{A}) = \text{Col}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}).$$

   This means that there is always at least one minimizer of (9).

2. If $\text{rank}(\boldsymbol{A}) = N$, then (10) has the unique solution, and so (9) has a unique minimizer. This unique minimizer is simply

   $$\hat{\boldsymbol{x}} = (\boldsymbol{A}^{\text{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\text{T}}\boldsymbol{y}.$$

3. If $\text{rank}(\boldsymbol{A}) < N$, then there are an infinite number of solutions to (9). In this case, $\boldsymbol{A}$ has a non-trivial null space, so if $\hat{\boldsymbol{x}}$ is a solution to (9), so is $\hat{\boldsymbol{x}} + \boldsymbol{v}$ for all $\boldsymbol{v} \in \text{Null}(\boldsymbol{A})$, as

   $$\|\boldsymbol{y} - \boldsymbol{A}(\hat{\boldsymbol{x}} + \boldsymbol{v})\|_2^2 = \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{v}\|_2^2 = \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2^2,$$

   since $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{0}$.

4. If $\text{rank}(\boldsymbol{A}) = M$, then there exists at least one $\hat{\boldsymbol{x}}$ such that $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2^2 = 0$, that is $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$. If in addition $M < N$, then there will be an infinity of such solutions.

Minimum ℓ_2 solutions

When there are an infinite number of solutions, we need a way to pick one of them. For this, modeling will again come into play ... we need some kind of value system other than the objective in (9) to judge which of the solutions is best. There are many, many ways this can be done; we will look two variations of one idea below.

One principle is to choose the solution that is the smallest (i.e. closest to the origin). In many applications this is justified using some kind of *minimum energy* principle — the sum of squares being a proxy for the amount of resources it takes to implement a solution. It can also be thought of as a kind of Occam's razor, in that the solution should not be any bigger than it needs to be.

The minimum energy least-squares solution is now the solution to the optimization program

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}. \tag{11}$$

It happens that there is now always a unique solution to this program, we will see why this is true when we revisit this problem in a later section of the course (and look at it through the lens of the singular value decomposition. For now, we will look at the case where $\boldsymbol{A}$ has full row rank, $\text{rank}(\boldsymbol{A}) = M$. In this case, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}$ if and only if $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, so (11) simplifies to

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{12}$$

The first thing to realize (and this holds in the general case (11) as well), is that the solution to the above will be in $\text{Row}(\boldsymbol{A})$. We

know that the row and null spaces are orthogonal complements of one another, and so every $\boldsymbol{x} \in \mathbb{R}^N$ can be written as

$$\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2, \quad \text{where} \;\; \boldsymbol{x}_1 \in \text{Row}(\boldsymbol{A}), \;\; \boldsymbol{x}_2 \in \text{Null}(\boldsymbol{A}),$$

and of course $\boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_2 = 0$. We can recast the optimization in (12) as a search over $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

$$\underset{\substack{\boldsymbol{x}_1 \in \text{Row}(\boldsymbol{A}) \\ \boldsymbol{x}_2 \in \text{Null}(\boldsymbol{A})}}{\text{minimize}} \;\; \|\boldsymbol{x}_1 + \boldsymbol{x}_2\|_2^2 \quad \text{subject to} \quad \boldsymbol{A}(\boldsymbol{x}_1 + \boldsymbol{x}_2) = \boldsymbol{y}.$$

Since $\boldsymbol{A}\boldsymbol{x}_2 = \boldsymbol{0}$ and $\|\boldsymbol{x}_1 + \boldsymbol{x}_2\|_2^2 = \|\boldsymbol{x}_1\|_2^2 + \|\boldsymbol{x}_2\|_2^2$ (Pythagorous), the above simplifies to

$$\underset{\substack{\boldsymbol{x}_1 \in \text{Row}(\boldsymbol{A}) \\ \boldsymbol{x}_2 \in \text{Null}(\boldsymbol{A})}}{\text{minimize}} \;\; \|\boldsymbol{x}_1\|_2^2 + \|\boldsymbol{x}_2\|_2^2 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{y}.$$

Since now $\boldsymbol{x}_2$ does not appear in the constraints, we see that the minimizer will have $\boldsymbol{x}_2 = \boldsymbol{0}$, i.e. the solution lies entirely in $\text{Row}(\boldsymbol{A})$.

When $\text{rank}(\boldsymbol{A}) = M$, we can use the fact that $\hat{\boldsymbol{x}} \in \text{Row}(\boldsymbol{A})$ to derive a closed-form solution. We know that there exists a $\hat{\boldsymbol{v}}$ such that

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}} \hat{\boldsymbol{v}}, \quad \text{and} \quad \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \hat{\boldsymbol{v}} = \boldsymbol{y}.$$

Since $\text{rank}(\boldsymbol{A}) = M$, we know that the $M \times M$ matrix $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ is invertible, and so there is exactly one $\hat{\boldsymbol{v}}$ that obeys the second condition above, namely $\hat{\boldsymbol{v}} = (\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{y}$. Thus the solution to (12) is

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{y}.$$

We will derive the solution to this problem for $\text{rank}(\boldsymbol{A}) < M$ in a few weeks when we talk about the singular value decomposition.

Regularization

The problem with solving (11) when $\boldsymbol{A}$ has a non-trivial null space is not just that there are an infinity of solutions, it is also that the space of solutions is unbounded — you can add a vector from the null space of arbitrary size and not change the functional. Even when $\boldsymbol{A}$ technically has full column rank, if it is poorly conditioned, meaning that

$$\|(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}(\boldsymbol{x}-\boldsymbol{z})\|_2 \gg \|\boldsymbol{x}-\boldsymbol{z}\|_2, \quad \text{for some } \boldsymbol{x},\boldsymbol{z} \in \mathbb{R}^N,$$

then a small change in $\boldsymbol{y}$ can amount to a massive change in $\hat{\boldsymbol{x}}$.

We will look much more carefully at describing how well-conditioned the least-squares problem is when we talk about the SVD. For now, know that there is a way to temper this behavior by penalizing the size of the solution. In place of (11), we solve

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \delta\|\boldsymbol{x}\|_2^2, \tag{13}$$

for some $\delta \geq 0$. It is similar in spirit to the minimum norm problem in (12) in that we are favoring solutions close to the origin, but it can be applied for any $\boldsymbol{A}$, no matter the rank. It also gives us a little more flexibility in the model, as we can treat δ as a "knob" that sets the trade-off between how closely we want $\boldsymbol{A}\boldsymbol{x}$ to match $\boldsymbol{y}$, and how large $\boldsymbol{x}$ can be. Note that when $\text{rank}(\boldsymbol{A}) = M$, the solution to (13) goes to the solution to (12) as $\delta \to 0$.

The analog of the normal equations for (13) is that a solution $\hat{\boldsymbol{x}}$ must obey

$$(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta\mathbf{I})\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}. \tag{14}$$

However, the matrix $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta\mathbf{I}$ is always invertible for $\delta > 0$ no matter what $\boldsymbol{A}$ is (we will see proof of this later in the course). Thus

the unique solution to (13) is

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta\mathbf{I})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}.$$

It also turns out that we can write this as

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} + \delta\mathbf{I})^{-1}\boldsymbol{y}.$$

You can verify this by plugging the expression above into the left hand side of (14).

From a high level perspective, solving(13) in place of (9) recognizes that making $\boldsymbol{A}\boldsymbol{x}$ as close to $\boldsymbol{y}$ as possible shouldn't be the only thing we are interested in. Models (which ultimately is where $\boldsymbol{A}$ comes from) are rarely perfect, and so pulling out all the stops to push $\boldsymbol{A}\boldsymbol{x}$ just a little closer to $\boldsymbol{y}$ can be self-defeating.

In the context of regression, using (13) in place of (11) is often referred to as **ridge regression**.

Ridge regression using a basis

Let's return to our fundamental problem of fitting a function to data (regression). Above, we saw that once a basis was introduced, we could set this up as a least-squares problem (equations (7) and (8) above), and then we saw one way to make this least-squares problem better posed using regularization.

Another variation is to impose the regularization in the function space. Suppose that we are looking for an $\boldsymbol{f}$ in a Hilbert space $\mathcal{S}$ that match observed data $\{(\boldsymbol{t}_m, y_m)\}$. The associated regularized

least-squares problem is

$$\underset{f\in\mathcal{S}}{\text{minimize}} \sum_{m=1}^{M} |y_m - f(\boldsymbol{t}_m)|^2 + \delta \|\boldsymbol{f}\|_S^2. \tag{15}$$

Note that we are using the Hilbert space norm to penalize the size of the function. If we again introduce a basis, modeling the target function as being in the span of $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N$, then we can rewrite this as

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \delta \left\| \sum_{n=1}^{N} x_n \boldsymbol{\psi}_n \right\|_S^2,$$

where $\boldsymbol{A}$ is constructed as in (7). Expanding the term on the right above gives us

$$\begin{aligned} \left\| \sum_{n=1}^{N} x_n \boldsymbol{\psi}_n \right\|_S^2 &= \left\langle \sum_{n=1}^{N} x_n \boldsymbol{\psi}_n, \sum_{k=1}^{N} x_k \boldsymbol{\psi}_k \right\rangle_S \\ &= \sum_{n=1}^{N} \sum_{k=1}^{N} x_n x_k \langle \boldsymbol{\psi}_n, \boldsymbol{\psi}_k \rangle_S \\ &= \boldsymbol{x}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{x}, \end{aligned}$$

where $\boldsymbol{G}$ is the Gram matrix for the basis, $G_{n,k} = \langle \boldsymbol{\psi}_n, \boldsymbol{\psi}_k \rangle_S$. Thus the optimization (15) in Hilbert space becomes the finite-dimensional problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \delta \boldsymbol{x}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{x}, \tag{16}$$

which has closed form solution[3]

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta \boldsymbol{G})^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{y}.$$

[3] Actually, there is no guarantee here that $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \delta\boldsymbol{G}$ is invertible for all $\delta > 0$, but we will assume that it is.

We then synthesize the solution as before,

$$\hat{f}(\boldsymbol{t}) = \sum_{n=1}^{N} x_n \psi_n(\boldsymbol{t}).$$

Of course, when the $\{\boldsymbol{\psi}_n\}$ are orthonormal, then $\boldsymbol{G} = \mathbf{I}$, and (16) is equivalent to the standard regularized least-square problem (13). When the basis is not orthonormal, then solving (16) and (13) will in general produce different weights (and hence different synthesized functions). But in practice, if the basis is reasonable (meaning that $\boldsymbol{G}$ is well-conditioned), then the difference will be minor.

Technical Details: Two easy linear algebra facts

Proposition. Let $\boldsymbol{A}$ be an $M \times N$ matrix. Then $\text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Null}(\boldsymbol{A})$.

Proof. It is clear that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{A}^{\text{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, and so $\text{Null}(\boldsymbol{A}) \subset \text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A})$. For inclusion the other way, realize that if $\boldsymbol{A}^{\text{T}}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, then $\boldsymbol{x}^{\text{T}}\boldsymbol{A}^{\text{T}}\boldsymbol{A}\boldsymbol{x} = 0$ which means $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, since $\boldsymbol{w}^{\text{T}}\boldsymbol{w} = 0$ if and only if $\boldsymbol{w} = \boldsymbol{0}$, and so $\text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) \subset \text{Null}(\boldsymbol{A})$.

Proposition. Let $\boldsymbol{A}$ be an $M \times N$ matrix. Then $\text{Col}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Row}(\boldsymbol{A})$.

Proof. It is clear that if two subspaces of $\mathbb{R}^N$ have the same orthogonal complement, then they are equal to one another. The proposition follows from the facts that

1. $\text{Null}(\boldsymbol{A})$ is the orthogonal complement of $\text{Row}(\boldsymbol{A})$,
2. $\text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A})$ is the orthogonal complement of $\text{Row}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Col}(\boldsymbol{A}^{\text{T}}\boldsymbol{A})$ (where this equality comes from the fact that $\boldsymbol{A}^{\text{T}}\boldsymbol{A}$ is symmetric), and
3. $\text{Null}(\boldsymbol{A}^{\text{T}}\boldsymbol{A}) = \text{Null}(\boldsymbol{A})$.