**2019-10-18**

# 1 Least squares and minimal norm problems

The least squares problem with Tikhonov regularization is

$$\text{minimize } \frac{1}{2}\|Ax - b\|_2^2 + \frac{\eta^2}{2}\|x\|^2.$$

The Tikhonov regularized problem is useful for understanding the connection between least squares solutions to overdetermined problems and minimal norm solutions to underdetermined problem. For $\eta > 0$, the system admits a unique solution independent of whether $m \geq n$ or $m < n$, and independent of whether or not $A$ has maximal rank. The limit when $\eta \to 0$ is also well-defined: it is the smallest norm $x$ that minimizes the residual $\|Ax - b\|$.

We usually write the Tikhonov-regularized solution as

$$x_\eta = (A^T A + \eta^2 I)^{-1} A^T b.$$

However, we can get some interesting insights by writing the regularized normal equations

$$\begin{bmatrix} -I & A \\ A^T & \eta^2 I \end{bmatrix} \begin{bmatrix} r_\eta \\ x_\eta \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

The first equation in this system defines the residual ($r = Ax - b$), while the second gives the regularized version of the normal equation ($A^T r + \eta^2 x = 0$). Eliminating the $r$ variable gives us the regularized normal equation in the form we have seen before; but we can also eliminate $x$ to yield

$$(-I - \eta^{-2} A A^T) r_\eta = b.$$

Scaling variables, we have

$$(AA^T + \eta^2 I) r_\eta = -\eta^2 b,$$

and by substituting into the equation $A^T r + \eta^2 x = 0$, we have an alternate expression for the solution to the regularized problem:

$$x_\eta = A^T (AA^T + \eta^2 I)^{-1} b.$$

Thus, playing around with the regularized normal equations gives us two different expressions for $x_\eta$:

$$\begin{aligned} x_\eta &= (A^T A + \eta^2 I)^{-1} b A^T \\ &= A^T (A A^T + \eta^2 I)^{-1} b \end{aligned}$$

In the full-rank overdetermined case ($m > n$), the former expression gives us the usual least-squares solutions $(A^T A)^{-1} A^T b$; in the full-rank underdetermined case ($m < n$), the latter expression gives us the usual minimum-norm solution $A^T (A A^T)^{-1} b$.

For the majority of this lecture, we will focus on the minimum-norm solution to overdetermined problems and its role in *kernel methods*. However, the connection between the regularized form of the minimum-norm solution in the overdetermined case and the regularized form of the least squares problem in the underdetermined case will be relevant to a discussion at the end of the lecture on (one) fast method for kernel-based fitting.

# 2 Feature maps and the kernel trick

In our first lecture on least squares, we described one of the standard uses of least squares: fitting a linear model to data. That is, given (possibly noisy) observations $y_i = f(x_i)$ for $x \in \mathbb{R}^N$, we fit $f(x) \approx s(x) = x^T \beta$ by minimizing the squared error:

$$\min_\beta \|X\beta - y\|^2,$$

where $X$ is a matrix whose $i$th row is the vector of coordinates for the $i$th data point. Even in simple applications of least squares, however, a purely linear model may not be adequate for modeling $f$; we might at least want to consider affine or polynomial functions in the coordinates, if not something more common. A simple way to get more complex models is to introduce a *feature map* that takes our original points in $\mathbb{R}^n$ and maps them into a higher-dimensional space where we will fit our linear models

$$s(x) = \phi(x)^T \beta, \quad \phi : \mathbb{R}^n \to \mathbb{R}^N.$$

The *features* $\phi_1, \ldots, \phi_N$ are chosen in advance; the regression coefficients $\beta$ are fit according to the data. When $m > N$, we would fit the regression coefficients by minimizing the residual norm $\|\Phi\beta - y\|$, where $[\Phi]_{ij} = \phi_j(x_i)$.

But often we are interested in the case when $N \gg m$, in which case we seek a minimal norm solution to the overdetermined problem, i.e.

$$\beta = \Phi^T(\Phi\Phi^T)^{-1}y.$$

Substituting this into our formula for $s$, we have

$$s(x) = \phi(x)^T\Phi^T(\Phi\Phi^T)^{-1}y.$$

Now, define the *kernel function* $k(x, x') = \phi(x)^T\phi(x')$; then we can rewrite $s(x)$ in terms of the kernel function as

$$s(x) = k_{xX}(K_{XX})^{-1}y$$

where $X = (x_1, x_2, \ldots, x_m)$ is the list of sample coordinates, and the subscript $X$ means "form a matrix or vector where $x_1, \ldots, x_m$ are inserted into this argument in turn," i.e.

$$k_{xX} = \begin{bmatrix} k(x,x_1) & k(x,x_2) & \ldots & k(x,x_m) \end{bmatrix},$$
$$K_{XX} = \begin{bmatrix} k(x_1,x_1) & \ldots & k(x_1,x_m) \\ \vdots & \ddots & \vdots \\ k(x_m,x_2) & \ldots & k(x_m,x_m) \end{bmatrix}.$$

Having expressed our interpolant purely in terms of the kernel function, we can now dispense with the feature map and the corresponding $\beta$ coefficient: only the kernel matters. For common kernels used in approximation theory and statistics (such as the Matérn family or the squared exponential kernels), we usually don't bother to write down an associated feature map.

# 3 Placing parens and alternate interpretations

The expression

$$s(x) = \phi(x)^T\Phi^T(\Phi\Phi^T)^{-1}y$$

involves a product of several terms that we can group in different ways:

$$\begin{aligned} s(x) &= \phi(x)^T\beta, & \beta &= \Phi^\dagger y \\ s(x) &= k_{xX}c, & c &= K_{XX}^{-1}y \\ s(x) &= d(x)^T y, & d(x) &= K_{XX}^{-1}K_{Xx} = (\Phi^T)^\dagger\phi(x) \end{aligned}$$

We have already discussed the meaning of the first of these groupings, with $\beta$ as a minimal-norm solution to an overdetermined linear system relating features to observations. We now comment on the other two.

The expression

$$s(x) = k_{xX}c = \sum_{i=1}^{m} k(x, x_i)c_i$$

involves basis functions $x \mapsto k(x, x_i)$ depending on the location of the data sites $x_1, \ldots, x_m$. Many common kernels depend only on the distance between the two arguments; for example, the squared exponential kernel is

$$k^{\text{SE}}(x, x') = \psi(\|x - x'\|; \sigma), \quad \psi(r) = \exp(-r^2/2\sigma^2).$$

In this case, we would have

$$s(x) = \sum_{i=1}^{m} \psi(\|x - x_i\|)c_i,$$

i.e. $s(x)$ is a linear combination of *translates* of the function $\psi$. The coefficients $c_i$ are simply chosen to satisfy the interpolation conditions.

The expression

$$s(x) = d(x)^T y = \sum_{i=1}^{m} d_i(x)y_i$$

is an expansion of $s$ in terms of the *Lagrange functions* $d_i$, which satisfy $d_i(x_j) = \delta_{ij}$. Another way of thinking about $d(x)$ involves the least squares formulation:

$$\text{minimize } \|\Phi^T d(x) - \phi(x)\|^2.$$

Why is this a sensible thing to do? The least squares formulation is attempting to solve the approximation problem

$$\phi_i(x) \approx \sum_{j=1}^{m} \phi_i(x_j)d_j(x)$$

in a least squares sense; that is, for a collection of representative functions (the features), we are trying to predict the value at $x$ as a linear combination of values at the sample points $x_1, \ldots, x_m$. Once we have that combination, together with the function values $f(x_1), \ldots, f(x_m)$ (the $y$ vector), we use the same linear combination to predict the value of $f(x)$.

# 4 From kernels back to least squares

While there are several interpretations for the kernel system, in practice we usually compute

$$K_{XX}c = y$$

and then predict using $s(x) = k_{xX}c$. In general, this costs $O(m^3)$ time for the initial fit and $O(m)$ time to evaluate the interpolant. However, we can sometimes use the structure of the kernel to more quickly compute the coefficients or predict at new points. Standard approaches typically exploit either *smoothness* of the kernel or *low-dimensional* structure of the distribution of points in the original space. We will briefly discuss the former, using the connection between minimal norm problems and least squares that we discussed earlier in the lecture.

For very smooth kernel functions with long length scales relative to the spacing between points, the kernel matrix $K_{XX}$ — though positive definite — will be very ill-conditioned. In this case, we often work with a regularized version of the fitting problem:

$$(K_{XX} + \eta^2 I)c = y$$

Often far fewer than $m$ eigenvalues of $K_{XX}$ that are much greater than $\eta^2$, and so we can effectively approximate the system by

$$(AA^T + \eta^2 I)\hat{c} = y.$$

Here, we can think of the rows of $A$ as being "reduced" feature vectors. From our earlier discussion, we recognize that $\hat{c}$ is the scaled residual for a regularized least squares problem with $A$; that is, if we solve

$$\text{minimize } \frac{1}{2}\|Au - b\|^2 + \frac{\eta^2}{2}\|u\|^2$$

then

$$\hat{c} = \eta^{-2}(b - Au) = -\eta^{-2}r.$$

Moreover, suppose we know how to compute the reduced feature vector at an evaluation point $x$, i.e. we can find $a_x$ such that

$$a_x^T A^T = k_{xX}.$$

Then using the regularized normal equation $A^T r + \eta^2 u = 0$, we have

$$k_{xX}\hat{c} = a_x^T A^T(-\eta^{-2} r) = a_x^T u.$$

That is, solving the regularized kernel problem is (up to error associated with a low-rank approximation) equivalent to solving a regularized least squares problem, and we get the same predictions whether we compute the least squares predictor $a_x^T u$ or the kernel-based predictor $k_{xX}\hat{c}$.