8.2 REGULARIZED LEAST SQUARES

Before this section begins, here is an advance look at what it will bring. You will see the new "two-square problem" right away. It connects to the old problems, but it has its own place and its own applications:

Ordinary least squares Minimize $\|Au - b\|^2$ by solving $A^{\mathrm{T}}A\widehat{u} = A^{\mathrm{T}}b$

Weighted least squares Minimize $(b - Au)^{\mathrm{T}}C(b - Au)$ by $A^{\mathrm{T}}CA\widehat{u} = A^{\mathrm{T}}Cb$

New problem Two squares Minimize $\|Au - b\|^2 + \alpha\|Bu - d\|^2$ by solving $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \boldsymbol{\alpha}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})\widehat{u} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} + \boldsymbol{\alpha}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{d}$. (1)

This equation (1) is not truly new. It is a special case of weighted least squares, if you adjust the notation to fit A and B into one problem. Then C is $[\,I \;\; 0 \;\; ; \;\; 0 \;\; \alpha I\,]$:

Combined matrix $\begin{bmatrix} A \\ B \end{bmatrix}$

$$\begin{bmatrix} A^{\mathrm{T}} & B^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \widehat{u} = \begin{bmatrix} A^{\mathrm{T}} & B^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} b \\ d \end{bmatrix}. \qquad (2)$$

This is equation (1). The solution $\widehat{u}$ depends on the weight α, which appears in that block matrix C. Choosing the parameter α wisely is often the hardest part.

Here are two important applications that lead to this sum of two squares:

Regularized least squares The original problem $A^{\mathrm{T}}A\widehat{u} = A^{\mathrm{T}}b$ can be very "ill-posed." This is typical of **inverse problems**, when we are trying to determine a cause from the effect it produces. The usual solution with $\alpha = 0$ is unreliable when A is highly ill-conditioned. For $A^{\mathrm{T}}A$, the ratio of largest to smallest eigenvalue might be 10^6 or 10^{10} or worse. Extreme examples have $m < n$ and singular $A^{\mathrm{T}}A$.

Adding $\alpha B^{\mathrm{T}}B$ regularizes the matrix $A^{\mathrm{T}}A$. It is like smoothing—we try to reduce the noise but save the signal. The weight α allows us to look for the right balance.

Constrained least squares To achieve $Bu = d$, increase the weight α. In the limit as $\alpha \to \infty$, we expect $\|B\widehat{u}_\alpha - d\|^2 \to 0$. The limiting $\widehat{u}_\infty$ solves a key problem:

Equality constraint ***Minimize*** $\|\boldsymbol{Au} - \boldsymbol{b}\|^2$ ***subject to*** $\boldsymbol{Bu} = \boldsymbol{d}$. (3)

Inverse problems have a tremendous range of applications. In most cases the words "least squares" never appear! To impose constraints we use large α. We will apply three leading methods to the simple constraint $Bu = u_1 - u_2 = 8$.

First we mention a key regularizing example (small α). Then come constraints.

I think this is truly the fundamental ill-posed problem of applied mathematics:

$$\textit{Estimate the velocity } \frac{dx}{dt} \textit{ from position } x \text{ (not exact) } \textit{at times } t_1, t_2, \ldots$$

Sometimes the problem comes in exactly that form. A GPS receiver gives positions $x(t)$ with great accuracy. It also estimates the velocity dx/dt, but how? The first idea is a finite difference like $x(t_2) - x(t_1)$ divided by $t_2 - t_1$. For high accuracy you need t_2 very near t_1. But when you divide by $t_2 - t_1$, any small position errors (*noise in the data*) are greatly amplified.

This is typical of ill-posed problems. **Small input errors, large output errors.** I will write the same problem as an *integral equation of the first kind*:

$$\textbf{Integral equation for } v \qquad \int_0^t v(s)\,ds = \int_0^t \frac{dx}{ds}\,ds = x(t) - x(0)\,. \tag{4}$$

The function $x(t)$ is given, the function $v(t)$ is the unknown. Many scientific problems look like this, often including a known kernel function $K(t,s)$ inside the integral. The equation is "Volterra" when the endpoints include the variable t, and "Fredholm" when they don't. Second kind equations add an extra term $cv(t)$, much easier.

Derivative estimation goes quickly into high dimensions. Many genes (some important, others not) may act to produce an expression $x(g_1, g_2, \ldots, g_N)$. The sizes of the derivatives $\partial x/\partial g_i$ tell which genes are important. It is an enormous problem to estimate all those derivatives from a limited number of sample values (measurements of x, often very noisy). Usually we discretize and then regularize by a small α. We return to these ill-posed problems after studying the other extreme, when α is large.

Large Penalty

We will minimize $u_1^2 + u_2^2$ with $u_1 - u_2 = 8$. This equality constraint $Bu = d$ fits Problem (3). B has n columns but only p rows (and rank p).

$$\textbf{Key example} \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & -1 \end{bmatrix} \qquad d = \begin{bmatrix} 8 \end{bmatrix}. \tag{5}$$

You could solve that problem without a Ph.D. Just substitute $u_2 = u_1 - 8$ into u_2^2. Minimizing $u_1^2 + (u_1 - 8)^2$ gives $u_1 = 4$. This approach is "the nullspace method" and we will extend it to other problems A, b, B, d. First come two other methods:

1. Large penalty	Minimize $u_1^2 + u_2^2 + \alpha(u_1 - u_2 - 8)^2$ and let $\alpha \to \infty$
2. Lagrange multiplier	Find a saddle point of $L = \frac{1}{2}(u_1^2 + u_2^2) + w(u_1 - u_2 - 8)$
3. Nullspace method	Solve $Bu = d$ and look for the shortest solution.

We start with the large penalty method, which is equation (1). Its big advantage is that we don't need a new computer code, beyond weighted least squares. This practical advantage should not be underestimated, and the key example with $u_1 = u_2 = 4$ will show that *the error in u decreases like* $1/\alpha$.

$$A^{\mathrm{T}}A = I \quad B^{\mathrm{T}}B = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1+\alpha & -\alpha \\ -\alpha & 1+\alpha \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 8\alpha \\ -8\alpha \end{bmatrix} = \alpha\, B^{\mathrm{T}} d. \quad (6)$$

Adding the equations gives $u_1 + u_2 = 0$. Then the first equation is $(1+2\alpha)u_1 = 8\alpha$:

$$u_1 = \frac{8\alpha}{1+2\alpha} = \frac{4}{1+(1/2\alpha)} = 4 - \frac{4}{2\alpha} + \cdots \text{ approaches the correct } \quad u_1 = 4. \quad (7)$$

The error is of order $1/\alpha$. So we need large α for good accuracy in u_1 and u_2. In this situation we are intentionally making the problem ill-conditioned. The matrix in (6) has eigenvalues 1 and $1 + 2\alpha$. Roundoff error could be serious at $\alpha = 10^{10}$.

Let me describe without proof the limit $\widehat{u}_\infty$ of the penalty method as $\alpha \to \infty$:

$$\widehat{\boldsymbol{u}}_\infty \ \textbf{ minimizes } \ \|\boldsymbol{Au} - \boldsymbol{b}\|^2 \ \textbf{ among all minimizers of } \ \|\boldsymbol{Bu} - \boldsymbol{d}\|^2.$$

Large α concentrates first on $\|Bu - d\|^2$. There will be many minimizers when $B^{\mathrm{T}}B$ is singular. Then the limiting $\widehat{u}_\infty$ is the one among them that minimizes the other term $\|Au - b\|^2$. We only require that $\left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$ has full column rank n, so the matrix $A^{\mathrm{T}}A + \alpha\, B^{\mathrm{T}}B$ is invertible.

Here is an interesting point. Suppose I divide equation (1) by α. Then as $\alpha \to \infty$, the equation becomes $B^{\mathrm{T}}B\,\widehat{u}_\infty = B^{\mathrm{T}}d$. All traces of A and b have disappeared from the limiting equation! But the penalty method is smarter than this, when $B^{\mathrm{T}}B$ is singular. Even as A and b fade out, minimizing with the $\|Au - b\|^2$ term included decides which limit $\widehat{u}_\infty$ the large penalty method will approach.

Lagrange Multipliers

The usual way to deal with a constraint $Bu = d$ is by a Lagrange multiplier. Elsewhere in this book, the constraint is $A^{\mathrm{T}}w = f$ and the multiplier is u. Now the constraint applies to u, so the multiplier will be called w. If we have p constraints $Bu = d$, we need p multipliers $w = (w_1, \ldots, w_p)$. The constraints go into L, multiplied by the w's:

$$\textbf{\textit{Lagrangian}} \ \boldsymbol{L(u,w)} = \frac{1}{2}\,\|\boldsymbol{Au} - \boldsymbol{b}\|^2 + \boldsymbol{w}^{\mathrm{T}}(\boldsymbol{Bu} - \boldsymbol{d}). \quad \textbf{\textit{Set}} \ \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{u}} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{w}} = \boldsymbol{0}.$$

The derivatives of L are zero at the saddle point u, w:

New saddle matrix S^* $$\begin{bmatrix} A^{\mathrm{T}}A & B^{\mathrm{T}} \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} A^{\mathrm{T}}b \\ d \end{bmatrix} \quad \begin{matrix} (n \text{ rows}) \\ (p \text{ rows}) \end{matrix} \quad (8)$$

Notice the differences from the saddle-point matrix S in Section 8.1. The new upper left block $A^{\mathrm{T}}A$ might be only positive *semidefinite* (possibly singular). The letters are all different, as expected. S^* will not be invertible unless the p rows of B are independent. Furthermore $\left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$ must have full column rank n to make $A^{\mathrm{T}}A + B^{\mathrm{T}}B$ invertible—this matrix appears when B^{T} times row 2 is added to row 1.

Our example can be solved in this Lagrange form, without any α:

$$\begin{array}{ll} \boldsymbol{A} = \boldsymbol{I} & \boldsymbol{b} = \mathbf{0} \\ \boldsymbol{B} = \begin{bmatrix} 1 & -1 \end{bmatrix} & \boldsymbol{d} = \mathbf{8} \end{array} \qquad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ -4 \\ -4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix}. \tag{9}$$

The optimal u_1, u_2 is $4, -4$ as earlier. The multiplier is $w = -4$.

The multiplier w always measures the sensitivity of the output P_{min} to the input d. P_{min} is the minimum value of $(u_1^2 + u_2^2)/2$. When you solve the problem for any d, you find $u_1 = d/2$ and $u_2 = w = -d/2$. Then $-w$ is the derivative of P:

$$\textbf{Sensitivity} \qquad P_{\mathsf{min}} = \frac{1}{2}(u_1^2 + u_2^2) = \frac{d^2}{4} \quad \text{has derivative} \quad \frac{d}{2} = \frac{8}{2} = -w. \tag{10}$$

So Lagrange gives something extra for solving a larger system.

Nullspace Method

The third approach to constrained minimization begins by solving $Bu = d$ directly. For $u_1 - u_2 = 8$, we did that at the start of the section. The result $u_2 = 8 - u_1$ was substituted into $u_1^2 + u_2^2$, which we minimized to get $u_1 = 4$.

When the matrix B is p by n, I could propose the same plan: Solve $Bu = d$ for p of the variables in terms of the other $n - p$. Substitute for those p variables in $\|Au - b\|^2$ and minimize. But this is not really a safe way.

The reason it's not safe is that a p by p block of B might be nearly singular. Then those p variables are the wrong ones to solve for. We would have to exchange columns and test condition numbers to find a good p by p submatrix. Much better to orthogonalize the p rows of B once and for all.

The plan of the nullspace method is simple: ***Solve $\boldsymbol{Bu = d}$ for $\boldsymbol{u = u_n + u_r}$.*** The *nullspace* vectors u_n solve $Bu_n = 0$. If the $n - p$ columns of Q_n are a basis for the nullspace, then every u_n is a combination $Q_n z$. One vector u_r in the row space solves $Bu_r = d$. Substitute $\boldsymbol{u = Q_n z + u_r}$ into $\|Au - b\|^2$ and find the minimum:

Nullspace method $\quad$ Minimize $\|A(u_n + u_r) - b\|^2 = \|AQ_n z - (b - Au_r)\|^2$

The vector z has only $n-p$ unknowns. Where Lagrange multipliers made the problem larger, this nullspace method makes it smaller. There are no constraints on z and we solve $n - p$ normal equations for the best $\widehat{z}$ in $AQ_n z = b - Au_r$:

$$\textbf{Reduced normal equations} \qquad Q_n^{\mathrm{T}} A^{\mathrm{T}} A Q_n \widehat{z} = Q_n^{\mathrm{T}} A^{\mathrm{T}} (b - Au_r). \tag{11}$$

Then $u = u_r + Q_n \widehat{z}$ minimizes $\|Au - b\|^2$ in the original problem subject to $Bu = d$.

We will solve the example $b_1 - b_2 = 8$ this way. First we keep A, b, B, and d, to construct a MATLAB code for the whole method. It might seem rather strange that only now, near the end of the book, we finally solve $Bu = d$! Linear equations are the centerpiece of this subject, and basic courses use elimination. The "reduced row echelon form" rref(B) gives an answer like $u_2 = u_1 - 8$ in textbooks. But *orthogonalization using* qr(B') gives a better answer in practice.

The usual Gram-Schmidt process converts the p columns of B^{T} into p orthonormal columns. The matrix is being factored into $B^{\mathrm{T}} = QR = (\boldsymbol{n}$ by $\boldsymbol{p})(\boldsymbol{p}$ by $\boldsymbol{p})$:

Gram-Schmidt $\quad QR = (p \text{ orthonormal columns})(\text{square triangular } R)\,. \qquad (12)$

MATLAB's qr command does more. It adds $n - p$ new orthonormal columns into Q, multiplying $n - p$ new zero rows in R. This is the $(\boldsymbol{n}$ by $\boldsymbol{n})(\boldsymbol{n}$ by $\boldsymbol{p})$ "unreduced" form. The letter r will stand for *reduced* and also for *row space*; the p columns of Q_r are a basis for the row space of B. The letter n indicates *new* and also *nullspace.*

Matlab: qr(B') is unreduced

$$B^{\mathrm{T}} = \begin{bmatrix} Q_r & Q_n \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \begin{matrix} p & \text{rows} \\ n-p & \text{rows} \end{matrix} \tag{13}$$

The $n - p$ orthonormal columns of Q_n solve $Bu = 0$ to give the nullspace:

Nullspace of B

$$BQ_n = \begin{bmatrix} R^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} Q_r^{\mathrm{T}} \\ Q_n^{\mathrm{T}} \end{bmatrix} Q_n = \begin{bmatrix} R^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ I \end{bmatrix} = 0\,. \tag{14}$$

The p columns of Q_r are orthogonal to each other ($Q_r^{\mathrm{T}} Q_r = I_p$), and orthogonal to the columns of Q_n. Our particular solution u_r comes from the row space of B:

Particular solution

$$\boxed{u_r = Q_r (R^{-1})^{\mathrm{T}} d} \quad \text{and} \quad \boxed{Bu_r = (Q_r R)^{\mathrm{T}} Q_r (R^{-1})^{\mathrm{T}} d = d\,.} \tag{15}$$

This is the particular solution given by the pseudoinverse, $u_r = B^+ d = \text{pinv}(B) * d$. It is orthogonal to all u_n. Householder's qr algorithm (better than Gram-Schmidt) has produced a square orthogonal matrix $\begin{bmatrix} Q_r & Q_n \end{bmatrix}$. Those two parts Q_r and Q_n lead to very stable forms of u_r and u_n. For an incidence matrix, Q_n will find loops.

We collect the 5 steps of the nullspace method into a MATLAB code:

1 $[Q, R] = \text{qr}(B')$; $\quad$ % square Q, triangular R has $n - p$ zero rows
2 $Qr = Q(1:p,:)$; $\;Qn = Q(p{+}1:n,:)$; $\;E = A * Qn$; $\quad$ % split Q into $[Qr \;\; Qn]$
3 $y = R(1:p, 1:p)' \backslash d$; $\;ur = Qr * y$; $\quad$ % particular solution ur to $Bu = d$
4 $z = (E' * E) \backslash (E' * (b - A * ur))$; $\quad$ % best un in the nullspace is $Qn * z$
5 $uopt = ur + Qn * z$; $\quad$ % $uopt$ minimizes $\|Au - b\|^2$ with $Bu = d$

Example $(u_1 - u_2 = 8)$ $\quad B^{\mathrm{T}} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ factors into $QR = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$.

The particular solution from $(1, -1)$ in Q_r is $u_r = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} \end{bmatrix}^{-1} \begin{bmatrix} 8 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \end{bmatrix}$.

The nullspace of $B = \begin{bmatrix} 1 & -1 \end{bmatrix}$ contains all multiples $u_n = Q_n z = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} z$.

In this example the squared distance happens to be a minimum at the particular u_r. **We don't want any of u_n, and the minimizing u has $z = 0$.** This case is very important and we focus on it now. It leads to the *pseudoinverse*.

Notation In most of this book, the constraint has been $A^{\mathrm{T}}w = f$. When B is A^{T}, the first line of the code will take qr(A). We are moving from the *large α problem* with $Bu \approx d$ to the *small α problem* with $Au \approx b$.

The Pseudoinverse

Suppose A is an m by n matrix, and the vector b has m components. The equation $Au = b$ may be solvable or not. The idea of least squares is to find the best solution $\widehat{u}$ from the normal equations $A^{\mathrm{T}}A\widehat{u} = A^{\mathrm{T}}b$. But this only produces $\widehat{u}$ when $A^{\mathrm{T}}A$ is invertible. **The idea of the pseudoinverse is to find the best solution u^+, even when the columns of A are dependent and $A^{\mathrm{T}}A$ is singular:**

Two properties $u^+ = A^+b$ is the **shortest vector** that solves $A^{\mathrm{T}}Au^+ = A^{\mathrm{T}}b$.

The other solutions, which are longer than u^+, have components in the nullspace of A. We will show that u^+ *is the particular solution with no nullspace component.*

There is an n by m matrix A^+ that produces u^+ linearly from b by $\boldsymbol{u^+ = A^+b}$. This matrix A^+ is the ***pseudoinverse*** of A. In case A is square and invertible, $u = A^{-1}b$ is the best solution and A^+ is the same as A^{-1}. When a rectangular A has independent columns, $\widehat{u} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$ is the only solution and then A^+ is $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$. In case A has *dependent columns* and therefore a nonzero nullspace, those inverses break down. Then the best (shortest) $u^+ = A^+b$ is something new.

You can see u^+ and A^+ in Figure 8.6, which shows how A^+ "inverts" A, from column space back to row space. The Four Fundamental Subspaces are drawn as rectangles. (In reality they are points or lines or planes.) From left to right, A takes all vectors $u = u_{\mathsf{row}} + u_{\mathsf{null}}$ to the column space. Since u_{row} is orthogonal to u_{null}, that nullspace part increases the length of u! The best solution is $u^+ = u_{\mathsf{row}}$.

This vector won't solve $Au^+ = b$ when that is impossible. It does solve $Au^+ = p$, the projection of b onto the column space. So the error $\|e\| = \|b - p\| = \|b - Au^+\|$ is a minimum. Altogether, u^+ is in the row space (*to be shortest*) and $Au^+ = p$ (*to be closest to b*). Then u^+ minimizes e and solves $A^{\mathrm{T}}Au^+ = A^{\mathrm{T}}b$.

How is u^+ computed? The direct way is by the Singular Value Decomposition:

SVD $$A = U\Sigma V^{\mathrm{T}} = \begin{bmatrix} U_{\mathsf{col}} & U_{\mathsf{null}} \end{bmatrix} \begin{bmatrix} \Sigma_{\mathsf{pos}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{\mathsf{row}} & V_{\mathsf{null}} \end{bmatrix}^{\mathrm{T}}. \tag{16}$$

The square matrices U and V have orthonormal columns: $U^{\mathrm{T}}U = I$ and $V^{\mathrm{T}}V = I$. The first r columns U_{col} and V_{row} are bases for the column space and row space of A.