Notes for 2016-02-26

A cautionary tale

You have been dropped on a desert island with a laptop with a magic battery of infinite life, a MATLAB license, and a complete lack of knowledge of soem basic geometry. In particular, while you know about least squares fitting, you have forgotten how to compute the perimeter of a square. You vaguely feel that it ought to be related to the perimeter or side length, though, so you set up the following model:

$$\text{perimeter} = \alpha \cdot \text{side length} + \beta \cdot \textit{diagonal}.$$

After measuring several squares, you set up a least squares system $Ax = b$; with your real eyes, you know that this must look like

$$A = \begin{bmatrix} s & \sqrt{2}s \end{bmatrix}, \quad b = 4s$$

where s is a vector of side lengths. The normal equations are therefore

$$A^T A = \|s\|^2 \begin{bmatrix} 1 & \sqrt{2} \\ \sqrt{2} & 2 \end{bmatrix}, \quad A^T b = \|s\|^2 \begin{bmatrix} 4 \\ 4\sqrt{2} \end{bmatrix}.$$

This system does have a solution; the problem is that it has far more than one. The equations are singular, but consistent. We have no data that would lead us to prefer to write $p = 4s$ or $p = 2\sqrt{2}d$ or something in between. The fitting problem is *ill-posed*.

We deliberately started with an extreme case, but some ill-posedness is common in least squares problems. As a more natural example, suppose that we measure the height, waist girth, chest girth, and weight of a large number of people, and try to use these factors to predict some other factor such as proclivity to heart disease. Naive linear regression – or any other naively applied statistical estimation technique – is likely to run into trouble, as the height, weight, and girth measurements are highly correlated. It is not that we cannot fit a good linear model; rather, we have too many models that are each almost as good as the others at fitting the data! We need a way to choose between these models, and this is the point of *regularization*.

Bias-variance tradeoffs in the matrix setting

Least squares is often used to fit a model to be used for prediction in the future. In learning theory, there is a notion of *bias-variance* decomposition of the prediction error: the prediction error consists of a bias term due to using a space of models that does not actually fit the data, and a term that is related to variance in the model as a function of measurement noise on the input. These are concepts that we can connect concretely to the type of sensitivity analysis we have seen before, a task we turn to now.

Suppose $A \in \mathbb{R}^{M \times n}$ is a matrix of factors that we wish to use in predicting the entries of $b \in \mathbb{R}^M$ via the linear model

$$Ax \approx b.$$

We partition A and b into the first m rows (where we have observations) and the remaining $M - m$ rows (where we wish to use the model for prediction):

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_e \end{bmatrix}$$

If we could access all of b, we would compute x by the least square problem

$$Ax = b + r, \quad r \perp \mathcal{R}(A).$$

In practice, we are given only A_1 and $b_1 + e$ where e is a vector of random errors, and we fit the model coefficients $\hat{x}$ by solving

$$\text{minimize } \|A_1 \hat{x} - (b_1 + e)\|^2.$$

Our question, then: what is the least squared error in using $\hat{x}$ for prediction, and how does it compare to the best error possible? That is, what is the relation between $\|A\hat{x} - b\|^2$ and $\|r\|^2$?

Note that

$$A\hat{x} - b = A(\hat{x} - x) + r$$

and by the Pythagorean theorem and orthogonality of the residual,

$$\|A\hat{x} - b\|^2 = \|A(\hat{x} - x)\|^2 + \|r\|^2.$$

The term $\|\hat{r}\|^2$ is the (squared) bias term, the part of the error that is due to lack of power in our model. The term $\|A(\hat{x} - x)\|^2$ is the variance term, and is associated with sensitivity of the fitting process. If we dig further into this, we can see that

$$x = A_1^\dagger (b_1 + r_1) \qquad \hat{x} = A_1^\dagger (b_1 + e),$$

and so

$$\|A(\hat{x} - x)\|^2 = \|AA_1^\dagger (e - r_1)\|^2$$

Taking norm bounds, we find

$$\|A(\hat{x} - x)\| \leq \|A\| \|A_1^\dagger\| (\|e\| + \|r_1)\|),$$

and putting everything together,

$$\|A\hat{x} - b\| \leq (1 + \|A\| \|A_1^\dagger\|) \|r\| + \|A\| \|A_1^\dagger\| \|e\|.$$

If there were no measurement error e, we would have a *quasi-optimality* bound saying that the squared error in prediction via $\hat{x}$ is within a factor of $1 + \|A\| \|A_1^\dagger\|$ of the best squared error available for any similar model. If we scale the factor matrix A so that $\|A\|$ is moderate in size, everything boils down to $\|A_1^\dagger\|$.

When $\|A_1^\dagger\|$ is large, the problem of fitting to training data is ill-posed, and the accuracy can be compromised. What can we do? As we discussed in the last section, the problem with ill-posed problems is that they admit many solutions of very similar quality. In order to distinguish between these possible solutions to find a model with good predictive power, we consider *regularization*: that is, we assume that the coefficient vector x is not too large in norm, or that it is sparse. Different statistical assumptions give rise to different regularization strategies; for the current discussion, we shall focus on the computational properties of a few of the more common regularization strategies without going into the details of the statistical assumptions. In particular, we consider four strategies in turn

1. *Factor selection* via *pivoted QR.*

2. *Tikhonov regularization* and its solution.

3. *Truncated SVD regularization.*

4. ℓ^1 *regularization* or the *lasso.*

Factor selection and pivoted QR

In ill-conditioned problems, the columns of A are nearly linearly dependent; we can effectively predict some columns as linear combinations of other columns. The goal of the column pivoted QR algorithm is to find a set of columns that are "as linearly independent as possible." This is not such a simple task, and so we settle for a greedy strategy: at each step, we select the column that is least well predicted (in the sense of residual norm) by columns already selected. This leads to the *pivoted QR factorization*

$$A\Pi = QR$$

where Π is a permutation and the diagonal entries of R appear in descending order (i.e. $r_{11} \geq r_{22} \geq \ldots$). To decide on how many factors to keep in the factorization, we either automatically take the first k or we dynamically choose to take k factors where r_{kk} is greater than some tolerance and $r_{k+1,k+1}$ is not.

The pivoted QR approach has a few advantages. It yields *parsimonious* models that predict from a subset of the columns of A – that is, we need to measure fewer than n factors to produce an entry of b in a new column. It can also be computed relatively cheaply, even for large matrices that may be sparse.

Tikhonov

A second approach is to say that we want a model in which the coefficients are not too large. To accomplish this, we add a penalty term to the usual least squares problem:

$$\text{minimize } \|Ax - b\|^2 + \lambda^2 \|x\|^2.$$

Equivalently, we can write

$$\text{minimize } \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2,$$

which leads to the regularized version of the normal equations

$$(A^T A + \lambda^2 I)x = A^T b.$$

In some cases, we may want to regularize with a more general norm $\|x\|_M^2 = x^T M x$ where M is symmetric and positive definite, which leads to the regularized equations

$$(A^T A + \lambda^2 M)x = A^T b.$$

If we know of no particular problem structure in advance, the standard choice of $M = I$ is a good default.

It is useful to compare the usual least squares solution to the regularized solution via the SVD. If $A = U\Sigma V^T$ is the economy SVD, then

$$\begin{aligned} x_{LS} &= V\Sigma^{-1}U^T b \\ x_{Tik} &= Vf(\Sigma)^{-1}U^T b \end{aligned}$$

where

$$f(\sigma) = \frac{1}{\sqrt{\sigma^{-1} + \lambda^2}}.$$

This *filter* of the inverse singular values affects the larger singular values only slightly, but damps the effect of very small singular values.

Truncated SVD

The Tikhonov filter reduces the effect of small singular values on the solution, but it does not eliminate that effect. By contrast, the *truncated SVD* approach uses the filter

$$f(z) = \begin{cases} z, & z > \sigma_{\min} \\ \infty, & \text{otherwise.} \end{cases}$$

In other words, in the truncated SVD approach, we use

$$x = V_k \Sigma_k^{-1} U_k^T b$$

where U_k and V_k represent the leading k columns of U and V, respectively, while Σ_k is the diagonal matrix consisting of the k largest singular values.

ℓ^1 and the lasso

An alternative to Tikhonov regularization (based on a Euclidean norm of the coefficient vector) is an ℓ^1 regularized problem

$$\text{minimize } \|Ax - b\|^2 + \lambda\|x\|_1.$$

This is sometimes known as the "lasso" approach. The ℓ^1 regularized problem has the property that the solutions tend to become sparse as λ becomes larger. That is, the ℓ^1 regularization effectively imposes a factor selection process like that we saw in the pivoted QR approach. Unlike the pivoted QR approach, however, the ℓ^1 regularized solution cannot be computed by one of the standard factorizations of numerical linear algebra. Instead, one treats it as a more general *convex optimization* problem. We will discuss some approaches to the solution of such problems later in the semester.

Tradeoffs and tactics

All four of the regularization approaches we have described are used in practice, and each has something to recommend it. The pivoted QR approach is relatively inexpensive, and it results in a model that depends on only a few factors. If taking the measurements to compute a prediction costs money — or even costs storage or bandwidth for the factor data! — such a model may be to our advantage. The Tikhonov approach is likewise inexpensive, and has a nice Bayesian interpretation (though we didn't talk about it). The truncated SVD approach involves the best approximation rank k approximation to the original factor matrix, and can be interpreted as finding the k best factors that are linear combinations of the original measurements. The ℓ_1 approach again produces models with sparse coefficients; but unlike QR with column pivoting, the ℓ_1 regularized solutions incorporate information about the vector b along with the matrix A.

So which regularization approach should one use? In terms of prediction quality, all can provide a reasonable deterrent against ill-posedness and overfitting due to highly correlated factors. Also, all of the methods described have a parameter (the number of retained factors, or a penalty parameter λ) that governs the tradeoff between how well-conditioned the fitting problem will be and the increase in bias that naturally comes from looking at a smaller class of models. Choosing this tradeoff intelligently may be rather more important than the specific choice of regularization strategy. A detailed discussion of how to make this tradeoff is beyond the scope of the class; but we will see some of the computational tricks involved in implementing specific strategies for choosing regularization parameters before we are done.