2022-09-29

1 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*.

2 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. However, pivoted QR is not the only approach! A related approach due to Golub, Klema, and Stewart computes $A = U\Sigma V^T$ and chooses a subset of the factors based on pivoted QR of V^T. More generally, approaches such as the lasso yield an automatic factor selection.

3 Tikhonov regularization (ridge regression)

Another 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 want to incorporate prior information that pushes x toward some initial guess x_0, we may pose the least squares problem in terms of $z = x - x_0$ and use some form of Tikhonov regularization. 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

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

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.

4 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.

5 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.

6 Regularization via iteration

We have briefly talked about one iterative method already (iterative refinement), and will talk about other iterative methods later in the semester. Some of these iterations have a regularizing effect when they are truncated early. In fact, there is an argument that slowly convergent methods may be beneficial in some cases!

As an example, consider the *Landweber iteration*, which is gradient descent applied to linear least squares problems:

$$x^{k+1} = x^k - \alpha_k A^T (Ax^k - b).$$

If we start from the initial guess $x^0 = 0$ and let the step size be a fixed $\alpha_k = \alpha$, each subsequent step is a partial sum of a Neumann series

$$\begin{aligned}
x^{k+1} &= \sum_{j=0}^{k} (I - \alpha A^T A)^j \alpha A^T b \\
&= \left(I - (I - \alpha A^T A)^{k+1}\right) (\alpha A^T A)^{-1} \alpha A^T b \\
&= \left(I - (I - \alpha A^T A)^{k+1}\right) A^\dagger b.
\end{aligned}$$

Alternately, we can write the iterates in terms of the singular value decomposition with a filter for regularization:

$$x^{k+1} = V \tilde{\Sigma}^{-1} U^T b, \quad \tilde{\sigma}_j^{-1} = (1 - (1 - \alpha \sigma_j^2)^{k+1}) \sigma_j^{-1}.$$

Usually, the Landweber iteration is stopped when k is large enough so that the filter is nearly the identity for large singular values, but is small enough to suppress the influence of small singular values.

The Landweber iteration is not alone in having a regularizing effect, but it is easier to analyze than some of the more sophisticated Krylov-based methods that we will describe later in the semester.

7 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.

8 Choice of regularization

All of the regularization methods we have discussed share a common trait: they define a parametric family of models. With more regularization, we restrict the range of models we can easily generate (adding bias), but we also reduce the sensitivity of the fit (reducing variance). The choice of the regularization parameter is a key aspect of these methods, and we now briefly discuss three different ways of systematically making that choice. In all cases, we rely on the assumption that the sample observations we use for the fit are representative of the population of observations where we might want to predict.

8.1 Morozov's discrepancy principle

Suppose that we want to fit $Ax \approx \hat{b}$ by regularized least squares, and the (noisy) observation vector $\hat{b}$ is known to be within some error bound $\|e\|$ of the true values b. The discrepancy principle says that we should choose the regularization parameter so the residual norm is approximately $\|e\|$. That is, we seek the most stable fitting problem we can get subject to the constraint that the residual error for the regularized solution (with the noisy vector $\hat{b}$) is not much bigger than we would get from unknown true solution.

One of the most obvious drawbacks of the discrepancy principle is that it requires that we have an estimate for the norm of the error in the data. Sadly, such estimates are not always available.

8.2 The L-curve

A second approach to the regularization parameter is the *L-curve*. If we draw a parametric curve of the residual error versus solution norm on a log-log plot, with $\log \|r_\lambda\|$ on the x axis and $\log \|x_\lambda\|$ on the y axis, we often see an "L" shape. In the top of the vertical bar (small λ), we find that increasing regularization decreases the solution norm significantly without significantly increasing the residual error. Along the end of the horizontal part, increasing regularization increases the residual error, but does not significantly help with the solution norm. We want the corner of the curve, where the regularization is chosen to minimize the norm of the solution subject to the constraint that the residual is close to the smallest possible residual (which we would have without regularization).

Computing the inflection point on the L-curve is a neat calculus exercise which we will not attempt here.

8.3 Cross-validation

The idea with cross-validation is to choose the parameter by fitting the model on a subset of the data and testing on the remaining data. We may do this with multiple partitions into data used for fitting versus data reserved for checking predictions. We often choose regularization parameters to give the smallest error on the predictions in a cross-validation study.

One variant of cross-validation involves minimizing the *leave-one-out cross-validation* (LOOCV) statistic:

$$\text{LOOCV} = \frac{1}{m} \sum_{i=1}^{m} \left[Ax^{(-i)} - b\right]_i^2,$$

where $x^{(-i)}$ denotes the model coefficients fit using all but the ith data point.

To compute the LOOCV statistic in the most obvious way, we would delete each row a_i^T of A in turn, fit the model coefficients $x^{(-i)}$, and then evaluate $r^{(-i)} = b_i - a_i^T x^{(-i)}$. This involves m least squares problems, for a total cost of $O(m^2n^2)$ (as opposed to the usual $O(mn^2)$ cost for an ordinary least squares problem). Let us find a better way! For the sake of concreteness, we will focus on the Tikhonov-regularized version of the problem

The key is to write the equations for $x^{(-i)}$ as a small change to the equations for $(A^TA + \lambda^2 I)x^* = A^Tb$:

$$(A^TA + \lambda^2 I - a_i a_i^T)x^{(-i)} = A^Tb - a_i b_i.$$

This subtracts the influence of row i from both sides of the normal equations. By introducing the auxiliary variable $\gamma = -a_i^T x^{(-i)}$, we have

$$\begin{bmatrix} A^TA + \lambda^2 I & a_i \\ a_i^T & 1 \end{bmatrix} \begin{bmatrix} x^{(-i)} \\ \gamma \end{bmatrix} = \begin{bmatrix} A^Tb - a_i b_i \\ 0 \end{bmatrix}.$$

Eliminating $x^{(-i)}$ gives

$$(1 - \ell_i^2)\gamma = \ell_i^2 b_i - a_i^T x^*$$

where $\ell_i^2 = a_i^T(A^TA + \lambda^2 I)^{-1} a_i$ is called the *leverage score* for row i. Now, observe that if $r = b - Ax^*$ is the residual for the full problem, then

$$(1 - \ell_i^2)r^{(-i)} = (1 - \ell_i^2)(b_i + \gamma) = (1 - \ell_i^2)b_i + \ell_i^2 b_i - a_i^T x_* = r_i,$$

or, equivalently

$$r^{(-i)} = \frac{r_i}{1 - \ell_i^2}.$$

We finish the job by observing that ℓ_i^2 is the ith diagonal element of the orthogonal projector $\Pi = A(A^T A + \lambda I)A^{-1}$, which we can also write in terms of the economy QR decomposition

$$\begin{bmatrix} A \\ \lambda I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$$

as $\Pi = Q_1 Q_1^T$. Hence, ℓ_i^2 is the squared row sum of Q_1 in the QR factorization.