

2023-03-01

Recall from last time that when there are many solutions that give an almost minimal residual $Ax - b$, we generally *regularize* the least squares problem by adding some assumption about which possible solutions are more likely or somehow more preferable. 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*.

1 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. $r_{11} \geq r_{22} \geq \dots$). 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.

2 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

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

where

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

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

3 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.

4 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.

5 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.