CPSC 540 Notes on Methods for Least Squares

Mark Schmidt

Fall 2014

In class we showed that the least squares estimator,

$$\arg\min_w \frac{1}{2}\|Ax - b\|^2,$$

is the solution to the linear system

$$(X^T X)w = X^T Y,$$

where $X \in \mathbb{R}^{N\times d}$, $Y = \mathbb{R}^{N\times 1}$, $w \in \mathbb{R}^{d\times 1}$. A solution to this problem always exists, but it may not be unique (but if multiple solutions exist, threy will all minimize the least squares objective function).

We'll also show that the ridge regression estimator,

$$\arg\min_w \frac{1}{2}\|Ax - b\|^2 + \frac{\lambda}{2}\|w\|^2,$$

is the solution to

$$(X^T X + \lambda I)w = X^T Y,$$

where λ is a scalar. In case, there is a unique value of w that satisfies the equation. Note that $X^T X$ is positive semi-definite while $(X^T X + \lambda I)$ is positive definite (although $X^T X$ will also be positive definite if the columns of X are independent).

The first way to solve these problems is by treating them as linear systems,

$$Ax = b,$$

and using methods to solve generic linear systems. In this case there are a few options available:

1. **Inverse**: If you compute A^{-1}, then you can simply set $x = A^{-1}b$. However, $X^T X$ may not have an inverse (if it is not positive definite), and this is both slower and less numerically stable than the methods below.
2. **Pseudo-Inverse**: The singular value decomposition of a matrix is a way to re-write a matrix in the form $A = UDV^T$, where U and V are orthogonal and D is diagonal. A special case for symmetric matrices is the spectral decomposition $A = VDV^T$ (here V is orthogonal contains the eigenvalues) while D is diagonal containing the eigenvalues (which are non-negative for positive semi-definite matrices like $X^T X$). If A is invertible, then its inverse is given by $VD^{-1}U^T$ (because U and V are orthogonal, and note that the inverse of D is a diagonal matrix with the reciprocals of the diagonal of D). This does not work when A is not invertible because we will have zeros on the diagonal of D. The pseudo-inverse is $VD^{\dagger}U^T$, where the diagonal matrix $D^{\dagger}$ is D^{-1} but with 0 in places where we would divide by zero. A solution to the linear system is given by $x = VD^{\dagger}U^T b$. This is more numerically stable than the inverse method, and works even when the inverse doesn't exist. In Matlab, the *pinv* function will give you the pseudo-inverse.

3. **Least-norm solution**: The pseudo-inverse method gives the least-norm solution (i.e., among all possible solutions it returns the one minimizing $\|w\|$). A faster way to compute this is with the QR-factorization, where you write $A = QR$ where Q is orthogonal and R is upper-triangular (so you can get x by solving the upper triangular system $Rx = Q^T b$ by back-substitution). This is the method used by Matlab when A is not square, and will be used by Matlab if you type $w = X\backslash y$, but note that it assumes that we have $Xw = y$ for some w (so in fact, the QR-factorization is less useful for machine learning).

4. **Gaussian elimination**: This decomposes the matrix as $A = LU$, where L is lower triangular and U is upper triangular. You can then use back-substituion to get the answer (this is equivalent to the Gauss-Jordan elimination you will see in an introductory linear algebra class). This is used by Matlab for square matrices where the Cholesky factorization fails. This is faster than QR, but may fail in the same cases that QR and the explicit inverse fail.

5. **Cholesky factorization**: For positive definite-matrices, this decomposes the matrix as $A = LL^T$. By using symmetry (and the fact that you do not need to exchange rows to perform Gaussian elimination in this case), this is faster than Gaussian eliminiation.

Conclusion: if you are doing ridge regression or know that $X^T X$ is positive-definite, use Cholesky. Otherwise, use the pseudo-inverse.

All of the above methods cost $O(Nd^2)$ to form $X^T X$ and $O(d^3)$ to run the solver. An alternative to direct solvers are iterative methods, which generate a sequence of iterations whose limit is the solution. The most common class of iterative methods only do multiplications with X and X^T, so their iteration cost is the much smaller $O(Nd)$. This includes methods like gradient descent and conjugate gradient (for positive definite matrices). Even faster methods only work with individual columns of X (coordinate descent) or individual rows of X (stochastic gradient), so their iteration costs are the even-lower $O(N)$ and $O(d)$.