Machine Learning Course - CS-433

Least Squares

Oct 3, 2017

minor changes by Martin Jaggi 2016

small changes by Rüdiger Urbanke 2017

Last updated on: September 25, 2017

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

In rare cases, one can compute the optimum of the cost function analytically. Linear regression using a mean-squared error cost function is one such case. Here the solution can be obtained explicitly, by solving a linear system of equations. These equations are sometimes called the normal equations. This method is one of the most popular methods for data fitting. It is called least squares.
To derive the normal equations, we use the optimality conditions for convex functions (see the previous lecture notes on optimization). I.e., at the optimium parameter, call it $\mathbf{w}^\star$, it must be true that the gradient of the cost function is 0. In other words, we must have that

$$\nabla \mathcal{L}(\mathbf{w}^\star) = \mathbf{0}.$$

This is a system of D equations.

Normal Equations

Recall that the cost function for linear regression with a mean-square error criterion is given by

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 = \frac{1}{2N} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}),$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix}.$$

If we take the gradient of this expression with respect to the weight vector $\mathbf{w}$ we get

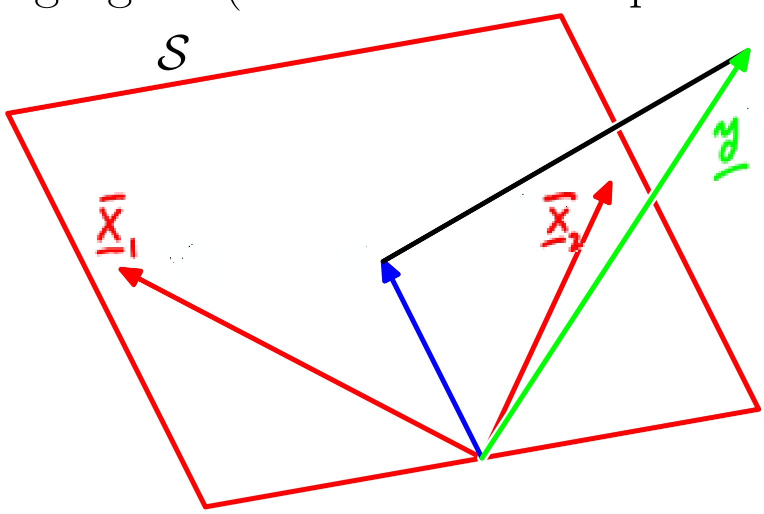
$$\nabla \mathcal{L}(\mathbf{w}) = -\frac{1}{N}\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}).$$

If we set this expression to 0 we get the normal equations for linear regression,

$$\mathbf{X}^\top \underbrace{(\mathbf{y} - \mathbf{X}\mathbf{w})}_{\text{error}} = \mathbf{0}. \tag{1}$$

Geometric Interpretation

Let $\mathcal{S}$ denote the space spanned by the columns of $\mathbf{X}$. Note that $\mathbf{x} = \mathbf{X}\mathbf{w}$ is an element of $\mathcal{S}$. I.e., by choosing $\mathbf{w}$ we choose $\mathbf{x} \in \mathcal{S}$. What element of $\mathcal{S}$ shall we take? The normal equations tell us that the optimum choice for $\mathbf{x}$, call it $\mathbf{x}^\star$, is that element so that $\mathbf{y} - \mathbf{x}^\star$ is orthogonal to $\mathcal{S}$. In other words, we we should pick $\mathbf{x}^\star$ to be equal to the projection of $\mathbf{y}$ onto $\mathcal{S}$.
The following figure (taken from Bishop's book) illustrates this point:



Rewriting the normal equations (1) by expanding the terms and we get

$$\mathbf{X}^\top \mathbf{X} \mathbf{w}^\star = \mathbf{X}^\top \mathbf{y}. \tag{2}$$

Least Squares

The matrix $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{D \times D}$ is called the Gram matrix. If it is invertible, we can multiply (2) by the inverse of the Gram matrix from the left to get a closed-form expression for the minimum.

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

We can use this model to predict a new value for an unseen datapoint (test point) $\mathbf{x}_m$:

$$\hat{y}_m := \mathbf{x}_m^\top \mathbf{w}^\star = \mathbf{x}_m^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

Invertibility and Uniqueness

Note that the Gram matrix $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{D \times D}$ is invertible if and only if $\mathbf{X}$ has full column rank, or in other words $rank(\mathbf{X}) = D$.

Proof: To see this assume first that $rank(\mathbf{X}) < D$. Then there exists a non-zero vector $\mathbf{u}$ so that $\mathbf{X}\mathbf{u} = 0$. It follows that $\mathbf{X}^\top \mathbf{X} \mathbf{u} = 0$, and so $rank(\mathbf{X}^\top \mathbf{X}) < D$. Therfore, $\mathbf{X}^\top \mathbf{X}$ is not invertible.

Conversely, assume that $\mathbf{X}^\top \mathbf{X}$ is not invertible. Hence, there exists a non-zero vector $\mathbf{v}$ so that $\mathbf{X}^\top \mathbf{X} \mathbf{v} = 0$. It follows that

$$0 = \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} = (\mathbf{X}\mathbf{v})^\top (\mathbf{X}\mathbf{v}) = \|\mathbf{X}\mathbf{v}\|^2.$$

This implies that $\mathbf{X}\mathbf{v} = 0$, i.e., $rank(\mathbf{X}) < D$.

Rank Deficiency and Ill-Conditioning

Unfortunately, in practice, $\mathbf{X}$ is often rank deficient.

- If $D > N$, we always have $rank(\mathbf{X}) < D$ (since row rank = col. rank)
- If $D \leq N$, but some of the columns $\mathbf{x}_{:d}$ are (nearly) collinear, then the matrix is ill-conditioned, leading to numerical issues when solving the linear system.

So what do we do when we either have a truly rank deficient matrix $\mathbf{X}$ or $\mathbf{X}$ is badly conditioned? We use the singular-value decomposition (SVD). We will learn more about the SVD in later lectures. Just for completeness let us write down the solution here. Let

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

be the SVD of $\mathbf{X}$. Here, $\mathbf{U}$ is an $N \times N$ unitary matrix. The matrix $\mathbf{S}$ is of dimension $N \times D$. It has zero entries except along the diagonal where the entries are non-negative and ordered from largest to smallest. Note further that the number of non-zero entries is equal to the rank of $\mathbf{X}$. Finally, $\mathbf{V}$ is a $D \times D$ unitary matrix. The equation we have to solve is then of the form

$$\mathbf{V}\mathbf{S}^\top\mathbf{S}\mathbf{V}^\top\mathbf{w}^\star = \mathbf{V}\mathbf{S}^\top\mathbf{U}^\top\mathbf{y}.$$

Multiplying by the left with $\mathbf{V}^\top$ we get

$$\mathbf{S}^\top\mathbf{S}\mathbf{V}^\top\mathbf{w}^\star = \mathbf{S}^\top\mathbf{U}^\top\mathbf{y}.$$

Note that this equation has in general a whole space of solutions. To find one particular one, define the so-called *pseudo-inverse* $\tilde{\mathbf{S}}$ which is a $D \times N$ matrix. On the diagonal of $\tilde{\mathbf{S}}$ take the non-zero diagonal entries of $\mathbf{S}$ and invert them. All other entries are zero. It is called a *pseudo-inverse* since $\tilde{\mathbf{S}}\mathbf{S}$ is a $D \times D$ matrix with zero entries except along the diagonal where the first $rank(\mathbf{X})$ entries are 1 and the rest are 0. And in a similar manner, $\mathbf{S}\tilde{\mathbf{S}}$ is a $N \times N$ matrix with zero entries except along the diagonal where the first $rank(\mathbf{X})$ entries are 1 and the rest are 0.
Multiply our equation from the left by $\mathbf{V}\tilde{\mathbf{S}}\tilde{\mathbf{S}}^\top$. We then have the solution

$$\mathbf{w}^\star = \mathbf{V}\tilde{\mathbf{S}}\mathbf{U}^\top\mathbf{y}.$$

Note that $\mathbf{V}\tilde{\mathbf{S}}\mathbf{U}^\top$ is known as the *pseudo-inverse* of $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ for the same reason that $\tilde{\mathbf{S}}$ is the pseudo-inverse of $\mathbf{S}$.

Summary of Linear Regression

We have studied three types of methods:

1. Grid Search

2. Iterative Optimization Algorithms
   (Stochastic) Gradient Descent

3. Least squares
   closed-form solution, for linear MSE

Additional Notes

Closed-form solution for MAE

Can you derive closed-form solution for 1-parameter model when using MAE cost function?
See this short article: http://www.johnmyleswhite.com/notebook/2013/03/22/modes-medians-and-means-an-unifying-perspective/.

Implementation

There are many ways to solve a linear system, but using the QR decomposition is one of the most robust ways. Matlab's backslash operator and also NumPy's linalg package implement this in just one line:

```
    w = np.linalg.solve(X, y)
```

For a robust implementation, see Sec. 7.5.2 of Kevin Murphy's book.