Least Squares

Mohammad Emtiyaz Khan
EPFL

Sep 24, 2015

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

In rare cases, we can compute the minimum of the cost function analytically. Linear regression using MSE is one such case. The solution is obtained using normal equations. This is called least squares.

To derive the equation, we use the optimality conditions. See the lecture notes for Gradient Descent.

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}} = 0$$

Using this, derive the normal equation for 1-parameter model.

Normal equations

Recall the expression of the gradient for multiple linear regression:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = -\frac{1}{N}\widetilde{\mathbf{X}}^T \mathbf{e} = -\frac{1}{N}\widetilde{\mathbf{X}}^T(\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta})$$

Set it to zero to get the normal equations for linear regression.

$$\widetilde{\mathbf{X}}^T \mathbf{e} = \widetilde{\mathbf{X}}^T(\mathbf{y} - \widetilde{\mathbf{X}}\boldsymbol{\beta}) = 0$$

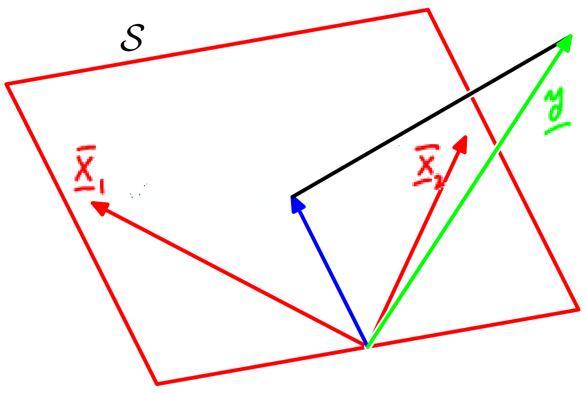
implying that the error is orthogonal to rows of $\widetilde{\mathbf{X}}^T$ and columns of $\widetilde{\mathbf{X}}$.

Geometric Interpretation

Denote the d'th column of $\widetilde{\mathbf{X}}$ by $\bar{\mathbf{x}}_d$.

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

The normal equations suggest to choose a vector in the span of $\widetilde{\mathbf{X}}$. The following figure illustrates this (taken from Bishop's book).



Least-squares

When $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ is invertible, we have a closed-form expression for the minimum.

$$\boldsymbol{\beta}^* = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$$

We can predict values for a new $\mathbf{x}_*$.

$$\hat{y}_* = \widetilde{\mathbf{x}}_*^T\boldsymbol{\beta}^* = \widetilde{\mathbf{x}}_*^T(\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{y}$$

Invertibility and uniqueness

The Gram matrix $\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$ is invertible iff $\widetilde{\mathbf{X}}$ has full column rank.

Proof: Assume $N > D$. The fundamental theorem of linear algebra states that the dimensionality of null space is zero for full column rank. This implies that the Gram matrix is positive definite, which implies invertibility.

Rank deficiency and ill-conditioning

Unfortunately, $\widetilde{\mathbf{X}}$ could often be rank deficient in practice, e.g. when $D > N$, or when the columns $\bar{\mathbf{x}}_d$ are (nearly) collinear. In the later case, the matrix is ill-conditioned, leading to numerical issues.

Summary of linear regression

We have studied three methods:

1. Grid search
2. (Stochastic) gradient descent
3. Least squares

Additional Notes

Closed-form solution for MAE

Can you derive close-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 implement matrix inversion, but using QR decomposition is one of the most robust ways. Matlab's backslash operator implements this (and much more) in just one line.

```
beta = inv(X'*X) * (X'*y)
beta = pinv(X'*X) * (X'*y)
beta = (X'*X) \ (X'*y)
```

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

To do

1. Revise linear algebra to understand why $\widetilde{\mathbf{X}}$ needs to have full rank. Read the Wikipedia page on rank of a matrix.
2. For details on the geometrical interpretation, see Bishop 3.1.2. However, better to read this after the lecture on "basis-function expansion". Also, note that notation in the book is different. This might make the reading difficult.
3. Understand matrix inversion robust implementation and play with it during the lab. Read Kevin Murphy's section 7.5.2 for details.

4. Understand ill-conditioning. Reading about the "condition number" in Wikipedia will help. Also, understanding SVD is essential. Here is another link provided by Dana Kianfar (EPFL) `http://www.cs.uleth.ca/~holzmann/notes/illconditioned.pdf`.

5. Work out the computational complexity of least-squares (use the Wikipedia page on computational complexity).