Machine Learning Course - CS-433

Least Squares

Oct 4, 2016

minor changes by Martin Jaggi 2016

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

In rare cases, one can compute the optimum of the cost function analytically. Linear regression using MSE is one such case. Here the solution can be obtained explicitly, from using a linear equation system, which is called the normal equations. This method is one of the most popular methods for data fitting, and called least squares.

To derive the normal equations, we use the optimality conditions for convex functions. See the previous lecture notes on optimization.

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

This is a system of D equations.

Exercise: Derive the normal equation for a 1-parameter linear model, for $\mathcal{L} = MSE$.

Normal Equations

Recall the expression of the gradient for multiple linear regression, under mean squared error:

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

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

$$\mathbf{X}^\top \mathbf{e} = \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) \stackrel{!}{=} \mathbf{0}$$

implying that the error is orthogonal to all rows of $\mathbf{X}^\top$ (= columns of $\mathbf{X}$).

Geometric Interpretation

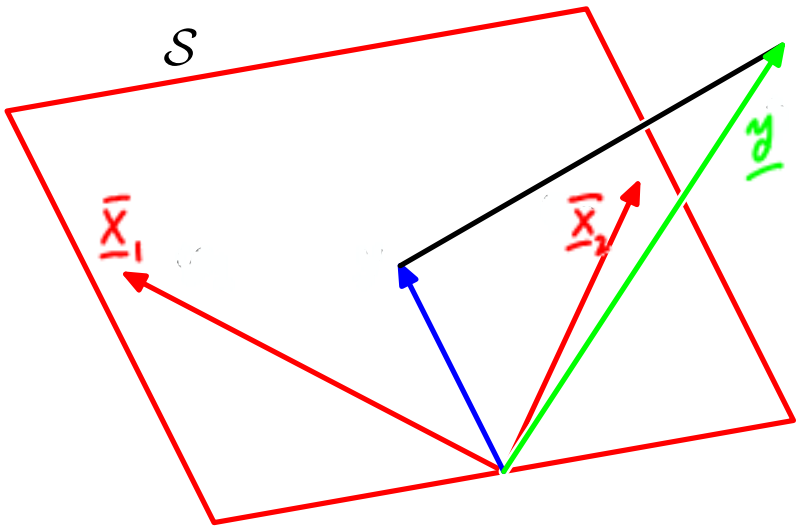
Denote the d'th column of $\mathbf{X}$ by $\mathbf{x}_d$.

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

The normal equations tell us to choose a vector in the span of $\mathbf{X}$, such that the error vector $\mathbf{e}$ will be orthogonal to the span.

The following figure illustrates this:

(taken from Bishop's book)



Least Squares

When $\mathbf{X}^\top\mathbf{X}$ is invertible, we have 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

The Gram matrix $\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{D\times D}$ is invertible iff $\mathbf{X}$ has full column rank, or in other words $rank(\mathbf{X}) = D$.

Proof: Fundamental theorem of linear algebra.

Rank Deficiency and Ill-Conditioning

Unfortunately, in practice, $\mathbf{X}$ is often rank deficient! (meaning $rank(\mathbf{X}) < D$)

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

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:

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

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

ToDo

1. Revise linear algebra to understand why $\mathbf{X}$ needs to have full rank. Read the Wikipedia page on the rank of a matrix.
2. Understand the robust implementation of the linear system solver, and play with it during the lab. Read Kevin Murphy's Section 7.5.2 for details.
3. 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.
4. Work out the computational complexity of least-squares, when using an existing linear systems solver (use the Wikipedia page on computational complexity).