Statistical Machine Learning
Part 3

Horacio Gómez-Acevedo
Department of Biomedical Informatics

February 9, 2021

(Multi)Linear Regression

It is very insightful to check the ordinary linear regression as a linear algebra problem. If we have training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, let's write X as a $n \times 2$ matrix

$$X = \begin{pmatrix} \mathbb{1} & \mathbb{X} \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

The response variable vector Y, the parameters θ_0, θ_1 and the vector of random errors are expressed as

$$\mathbb{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

(Multi)Linear Regression (cont)

Then we can write the population regression formula in a matrix form

$$\mathbb{Y} = X \cdot \theta + \varepsilon = \theta_0 \mathbb{1} + \theta_1 \mathbb{X} + \varepsilon \tag{1}$$

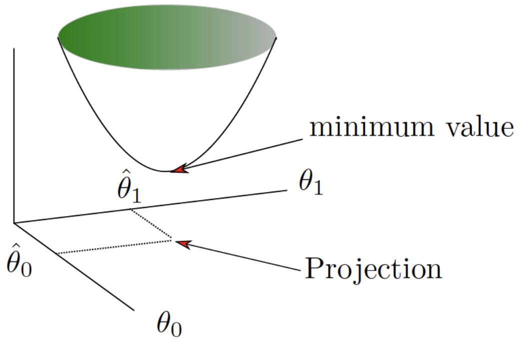
The method of least squares reduces to find $(\hat{\theta}_0, \hat{\theta}_1)$ that minimizes the function

$$(\theta_0, \theta_1) \mapsto \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

Least Squares Function

As we can see in the following graph, the projection of the minimum value will be the estimates of $\hat{\theta}_0$ and $\hat{\theta}_1$.

$$f(\theta_0, \theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta_0 - \theta_1 x_i)^2$$



Multidimensional Interpretation

If we plot the vectors $\mathbb{1}$ and $\mathbb{X}$, the hyperplane (generated) by those vectors, contains a vector $\mathbb{V}$ which is closest to the vector $\mathbb{Y}$. It can be proven that the vector $\mathbb{V}$ can be written as

$$\mathbb{V} = X(X^tX)^{-1}X^t\mathbb{Y}$$

On the other hand

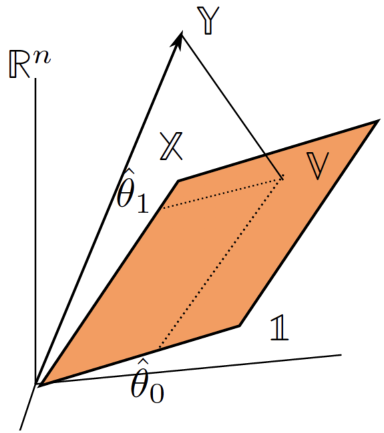
$$\mathbb{V} = X \cdot \begin{pmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \end{pmatrix}$$

Thus

$$\begin{pmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \end{pmatrix} = (X^tX)^{-1}X^t\mathbb{Y} \tag{2}$$

This last equation is called the **normal form**

Multidimensional Interpretation



Computational Complexity

The solution of the linear regression problem using the normal equation (2), requires significant amount of computation resources as n tend to increase.
Let $f, g \colon \mathbb{N} \to \mathbb{R}^+$ two non-negative functions. We say that $f(n) = O(g(n))$ (big O) if there are two non-negative integers n_0 and c such that for every $n \geq n_0$

$$f(n) \leq c \cdot g(n)$$

Example. Let $f(n) = 5n^3 + 3n^2 + 1000n + 10000$. In this case $f(n) = O(n^3)$ since when n is very large, $f(n)$ behaves like the polynomial $5n^3$.
The complexity of matrix inversion is between $O(n^{2.4})$ and $O(n^3)$, depending on the algorithm.

Gradient Descent Revisited

In a previous lecture, we have defined the general gradient descent rational as reaching the (hopefully global) minimum iteratively using a formula like

$$\theta^{k+1} = \theta^k - \eta(k)\nabla J(\theta^k)$$

In the context of linear regression, $J(\mathbf{x}) = MSE(\mathbf{x})$. Since the MSE is a convex function, a global minimum can be reached! Thus after a certain number of steps m we say that θ^m is (close enough) to the linear regression problem

$$X \cdot \theta^m \approx \mathbb{Y}$$

Note. In Géron's book the notation is just a bit different (by using the transpose), but the concepts are totally equivalent.

$$MSE(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2$$

Stochastic Gradient Descent

For the **cost function** MSE, the gradient is calculated as

$$\nabla_\theta MSE(\theta) = \frac{2}{m} X^t (X \cdot \theta) - \mathbb{Y}$$

Thus, at each step of the gradient descent, we need to recalculate the gradient using the whole dataset. This operation is expensive as matrix multiplication continue to pile up. One solution is to use the calculations of the gradient taking instances in the training set at every step and computing gradients based only on that instance.

Regularized Linear Models (Shrinkage)

Traditionally these methods involves using least squares to fit a linear model that contains a subset of the predictors (ergo the name shrinkage). These techniques *constrains* or *regularizes* the coefficient estimates towards zero.

$$RSS = \sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2$$

In **ridge regression**, the coefficients are estimated by minimizing a slightly different quantity. The ridge regression coefficient estimates $\hat{\theta}^R$ that minimize

$$\sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \theta_j^2,$$

where $\lambda \geq 0$ is a tuning parameter (hyperparameter) to be determined separately.

Ridge Regression (cont)

The shrinkage penalty is applies to $\theta_1, \ldots, \theta_p$ but not to intercept θ_0. Since this is the measure of the mean value of the response when $x_{i1} = x_{i2} = \cdots = x_{ip} = 0$.

One of the main reasons to use ridge regression is in the bias-variance trade-off. As λ increases, the flexibility of the ridge regression fit decreases leading to decreased variance but increased bias.

One disadvantage of ridge regression is that it will include all p predictors in the final model. The penalty term $\lambda \sum_{i=1}^{p} \theta_j$ will shrink all the coefficients towards zero, but it will not set any of them exactly to zero.

Lasso

The **Lasso** alternative calculates the coeffcients $\hat{\theta}^L_\lambda$ by minimizing a slightly different formula

$$\sum_{i=1}^{n}\left(y_i - \theta_0 - \sum_{j=1}^{p}\theta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\theta_j|,$$

The advantage of lasso is that it forces some of the coefficien estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. Thus, we say that lasso yields *sparse* models.

Optimization perspective

Ridge regression can be seen as the minimization of the following problem

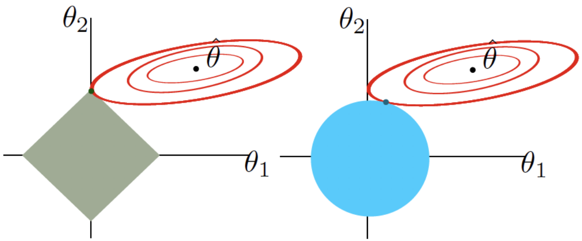
$$\min_{\theta}\left\{\sum_{i=1}^{n}\left(y_i-\theta_0-\sum_{j=1}^{p}\theta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}\theta_j^2 \leq s$$

whereas the lasso becomes

$$\min_{\theta}\left\{\sum_{i=1}^{n}\left(y_i-\theta_0-\sum_{j=1}^{p}\theta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}|\theta_j| \leq s$$

Optimization picture

The ellipses that are centered around $\hat{\theta}$ represent regions of constant RSS. The intersections between the lasso and ridge regression coefficient estimates with the ellipses. Notice that for the lasso the point of contact has $\theta_1 = 0$, whereas for the ridge regression the value of θ_1 is "small" but does not vanish.



Problem with Lasso

For Lasso Regression the cost function is defined as

$$J(\theta) = MSE(\theta) + \alpha \sum_{i=1}^{n} |\theta_i|$$

However, the absolute function is not differentiable at zero! So, we use the right or left hand derivatives. More precisely,

$$g(\theta, J) = \nabla_\theta MSE(\theta) + \lambda \begin{pmatrix} \operatorname{sign}(\theta_1) \\ \vdots \\ \operatorname{sign}(\theta_n) \end{pmatrix}$$

where

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Elastic Network

This model is a (convex) combination between Ridge Regression and Lasso Regression. For $r \in [0,1]$ and $s \geq 0$ the optimization goal is

$$\min_{\theta} \left\{ \sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 \right\}$$

$$\text{subject to } (1-r) \sum_{j=1}^{p} |\theta_j| + r \sum_{j=1}^{p} \theta_j^2 \leq s$$

References

Materials and some of the pictures are from (1),(2), and (3).

1. Gareth James et al. *An Introduction to Statistical Learning with applications in R*. Springer (2015)
2. Richard O. Duda et al. *Pattern Classification* John Wiley (2001).
3. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn & TensorFlow* O'Relly (2017)
4. Wiebe R. Pestman *Mathematical Statistics* de Gruyter (1998)

I have used some of the graphs by hacking TiKz code from StakExchange, Inkscape for more aesthetic plots and other old tricks of $\TeX$