Linear Regression

Jingbo Xia

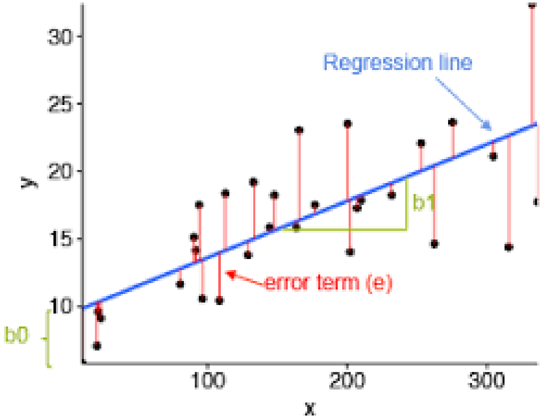
Huazhong Agricultural University

xiajingbo.math@gmail.com

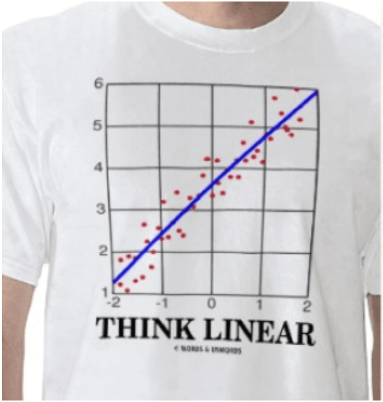
April 3, 2019

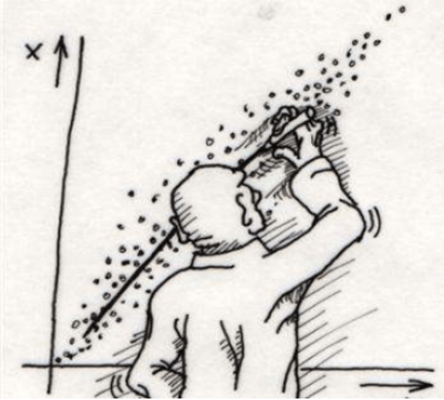
Table of contents I

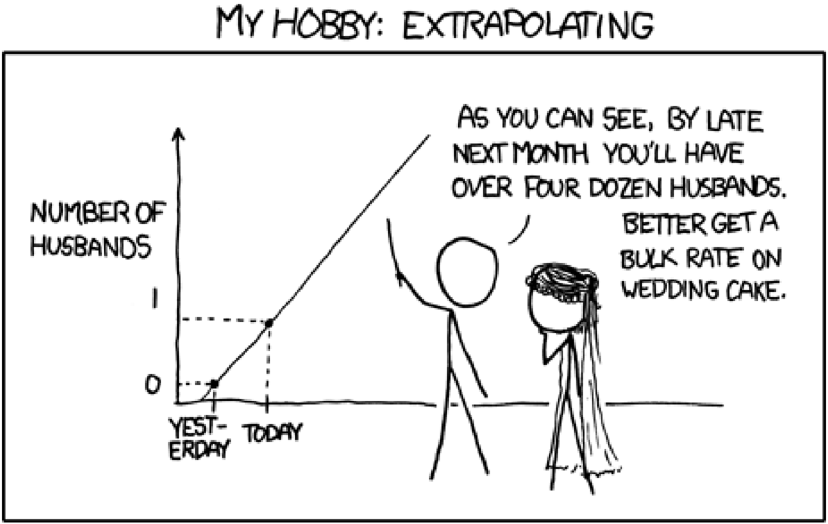
In order to understand loss function and regularization, let's get started from the classical linear regression problem.



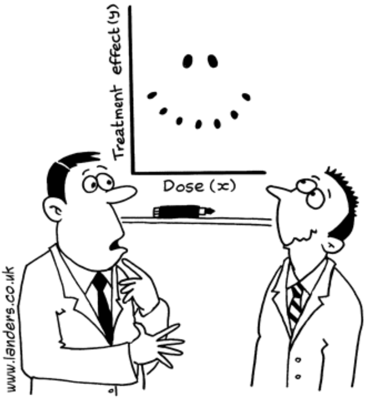
Linear Regression






MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS.
BETTER GET A BULK RATE ON WEDDING CAKE.

Linear Regression



"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Outline

Linear Regression with Least Square

In order to understand loss function and regularization, let's get started from a linear regression problem. For example, we have n p-demensional sample data x_i, and their regression value is $y_i \in \mathbb{R}$, here, $x_i = (x_{i1}, \cdots, x_{ip})^T$, $i = 1, 2, \cdots, n$. The form of linear regression model is to find a regression function $f(x, w)$ such that approximate $\tilde{y}_i = f(x_i, w)$ to y_i :

$$y_i \longleftarrow \tilde{y}_i = f(x_i, w) = \sum_{j=1}^{p} w_j x_{ij} = w^T x_i{}^1, \tag{1}$$

where $w = (w_1, \cdots, w_p)^T$.

[1] Actually, this is a short form of the regression function $f(x_i, w, b) = \sum_{j=1}^{p} w_j x_{ij} + b = w^T x_i + b$. The complete form of the linear function can also be rewritten in a short form, $f(x_i, w, b) := \hat{f}(\hat{x_i}, \hat{w})$, if we denote $\hat{x_i} = (x_i^T, 1)^T = (x_{i1}, \cdots, x_{ip}, 1)^T$, and $\hat{w} = (w^T, b) = (w_1, \cdots, w_p, b)^T$.
So, we use the short form for brevity.

Linear Regression with Least Square

The square loss learning will suffices to minimize the following loss function

$$\mathcal{J}(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 := \frac{1}{n}||y - Xw||_2^2 = \frac{1}{n}(y - Xw)^T(y - Xw), \tag{2}$$

here $||\cdot||_2$ is a $l2$ norm, while $X = (x_1, \cdots, x_n)^T \in \mathbb{R}^{n\times p}$ is the matrix for sample data, and $y = (y_1, \cdots, y_n)^T$ is the regression value vector.

Please notice that there is an explicit solution of this problem, if X^TX is a invertable matrix[2], say:

$$\hat{w} = (X^TX)^{-1}X^Ty. \tag{3}$$

However, if $p > n$, rank of X^TX is not full, and make the above one unsolvable.
In another word, there are infinite solutions for the problem.
[Assignment] Prove formula (3).

[2]Gradient analysis would solve this problem directly, and I'd like to make it an assignment.

Linear Regression with Least Square

[Answer sheet]:
Compute the gradient of $\mathcal{J}(w)$, we have

$$
\begin{aligned}
0 &= \frac{\partial \mathcal{J}(w)}{\partial w} = \frac{\partial(\frac{1}{n}(y-Xw)^T(y-Xw))}{\partial w} = \frac{1}{n}\frac{\partial((y^T-w^TX^T)(y-Xw))}{\partial w} \\
&= \frac{1}{n}\frac{\partial(y^Ty-y^TXw-w^TX^Ty+w^TX^TXw)}{\partial w} \\
&= \frac{1}{n}(0-X^Ty-X^Ty+2X^TXw) \\
&= \frac{2}{n}(-X^Ty+X^TXw)
\end{aligned} \tag{4}
$$

Let the gradient equal to zero, we have

$$
\hat{w} = (X^TX)^{-1}X^Ty. \tag{5}
$$

Outline

Let's convert the **linear regression model** into a so-called **ridge regression model**, by adding a $l2$ regularizer:

$$\mathcal{J}_{Ridge}(w) = \frac{1}{n}||y - Xw||_2^2 + \lambda||w||_2^2. \text{—Ridge regression.} \tag{6}$$

The addition of this regularizer made the solution of the model yield to the solution with smaller $||w||_2$. Equivalently, from a view of convex optimization, the minimization of the above loss function suffices to:

$$\min_w \frac{1}{n}||y - Xw||_2^2, \qquad s.t., ||w||_2 < C. \tag{7}$$

Here, C is a constant, related to λ.
Therefore, we restrict the norm of w, and shrink the searching space of the solution.

Do you remember the convex definition of a loss function? The convexity of a $\mathcal{J}(w)$ can not ensure the solution being found quickly, or within your patience, or within limited searching time.

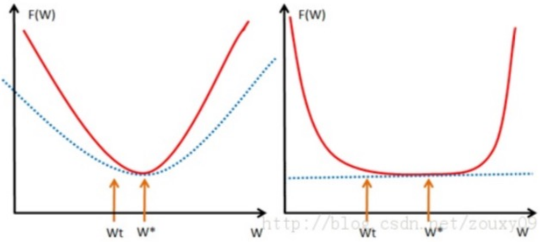


Figure 1: Convexity of loss function is not alway amusing. (Take an example when w is a 2-dimensional vector)

To add the regularization, we make the loss function more strong convex, and so as to accelerate the convergence.

Question

In the meantime, in many cases we want to find solution with many zeros in w, which is called sparsity. But why?

My second question here: If a w met the sparsity requirement, where is this point w in figure 1.

Why do we like to have sparsity in w? Here is an example:

Associate genotypes to a given phenotype. Ref "LASSO: Powerful New Technique That 'Ropes In' Thousands of Genes At Once" [3].

[3] https://www.biotecnika.org/2017/07/lasso-powerful-new-technique-that-ropes-in-thousands-of-genes-at-once/

Comparison of various lp norm hints that $l0$ norm is the best straightforward one for achieving sparsity. Unfortunately, it is not convex. Actually, we used l_1 norm to replace l_0 for the purpose of sparsity, say, feature reduction. It is called **Lasso regression.**

$$\mathcal{J}_{Lasso}(w) = \frac{1}{n}||y - Xw||_2^2 + \lambda||w||_1^2. \text{ —Lasso regression.} \tag{8}$$

Equivalently, from a view of convex optimization, the minimization of the above loss function suffices to:

$$\min_w \frac{1}{n}||y - Xw||_2^2, \qquad s.t., ||w||_1 < C. \tag{9}$$

Here, C is a constant, related to λ.

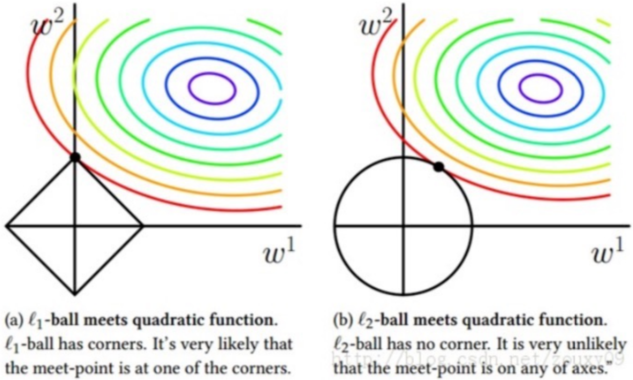


Figure 2: When $l2$ or norm ball or $l1$ norm ball meet contour map of a quadratic function

It shows that $l1$ norm ball has much more chance to meet the contour map in the affine. This suffices to a zero value of w. For arbitrary case of dimension n for w, this means sparsity.

Outline

LASSO regression, and a python codes example

A nice example comes from Zhihu blog[4].

Example (Command lines)

```
>git clone https://github.com/PytLab/MLBox.git
```

The raw data are:

Example (Raw data for regression)

```
1     0.455  0.365  0.095  0.514   0.2245  0.101   0.15   15
1     0.35   0.265  0.09   0.2255  0.0995  0.0485  0.07   7
-1    0.53   0.42   0.135  0.677   0.2565  0.1415  0.21   9
1     0.44   0.365  0.125  0.516   0.2155  0.114   0.155  10
0     0.33   0.255  0.08   0.205   0.0895  0.0395  0.055  7
0     0.425  0.3    0.095  0.3515  0.141   0.0775  0.12   8
-1    0.53   0.415  0.15   0.7775  0.237   0.1415  0.33   20
...
```

[4] https://zhuanlan.zhihu.com/p/30535220]

LASSO regression, and a python codes example

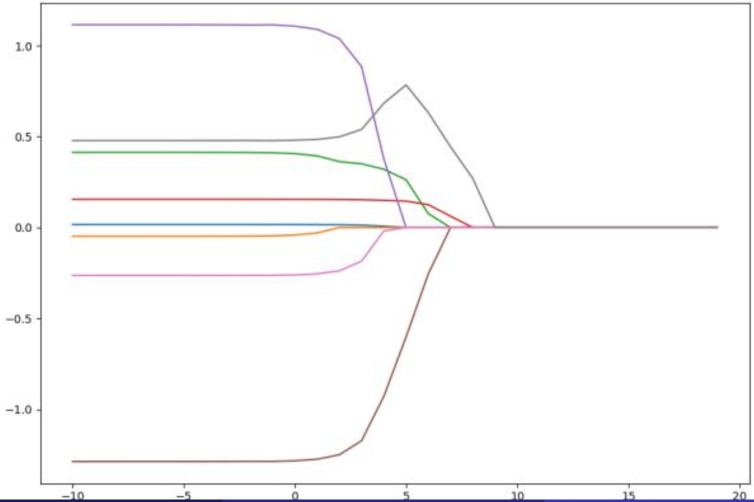
w approximate 0, while λ increases.

Example (Command lines)

```
lambda = e^(0), w = [[ 0.0164  -0.0412  0.4066   0.1553  1.1076 -1.2
lambda = e^(1), w = [[ 0.0161 -0.0295  0.3941  0.1550   1.0905 -1.27
lambda = e^(2), w = [[ 0.0153   0.      0.3626  0.1542  1.0391 -1.249
lambda = e^(3), w = [[ 0.01325  0.      0.3505  0.1528  0.8850 -1.172
lambda = e^(4), w = [[ 0.0076  0.           0.3209  0.1497  0.3782 -0
lambda = e^(5), w = [[ 0.   0.      0.2627  0.1453  0.          -0.601
lambda = e^(6), w = [[ 0.    0.      0.0766  0.1260  0.          -0.25
lambda = e^(7), w = [[ 0.     0.     0.  0.0628  0.  0. 0.   0.4449]]
lambda = e^(8), w = [[ 0.   0.      0.   0.     0.  0. 0.   0.2707]]
lambda = e^(9), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
lambda = e^(10), w = [[ 0.  0.  0.  0.  0.  0.  0. 0.]]
lambda = e^(11), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
lambda = e^(12), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
lambda = e^(13), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
lambda = e^(14), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
lambda = e^(15), w = [[ 0.  0.  0.  0.  0.  0.  0.  0.]]
```

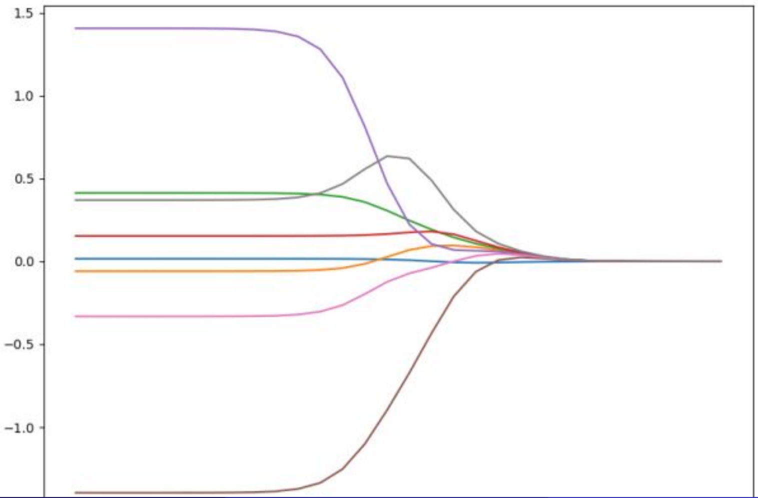
LASSO regression, and a python codes example

LASSO regression: w_i approximates zero when λ increases.



LASSO regression, and a python codes example

Ridge regression: w_i doesn't approximates zero very quickly when λ increases.



Outline

Wrap-up!

You might ask:

Hey! Why should us collect that many definitions? See! Norms, regularization, matrix calculus... I kind of remember you once mentioned regression. What a mixture!

:(

Suggestion...

Shall we calculate the gradient descent rule(See formula (**??**)) for updating w for Ridge regression (See formula (6))?

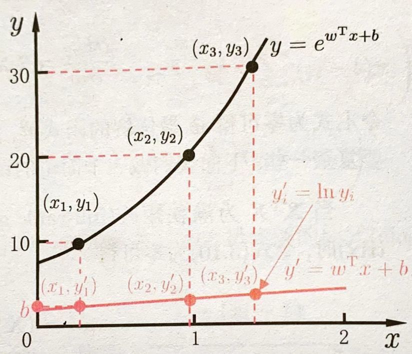
This is how the ridge regression work. Won't be hard.

Outline

We've already known how to make regression by using a linear regression model. Sometimes, we would like to generalize the linear regression model and make it approximate a series of observations with non-linear values. For example,

$$\ln y = w^T x + b \tag{10}$$

is called "log-linear regression".



Generally, if we consider a monotonic differentiable function $g(\cdot)$,

$$y = g^{-1}(w^T x + b) \tag{11}$$

is called "generalized linear model". The funtion, $g(\cdot)$ is called "link function".

Generalized linear model, from regression to classification

Unit-step function for classification

Consider a two-class classification, and the label is $y \in \{0, 1\}$, and the only attempt needed is to convert a real number $z = w^T + b$ to a binary value y. The ideal choice is "unit-step function":

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0. \end{cases} \tag{12}$$

However unit-step function is not continuous. So, "sigmoid" function replaces it. That's Logistic regression model for binary classification!