Linear Regression
线性回归和正则项

Jingbo Xia

Huazhong Agricultural University

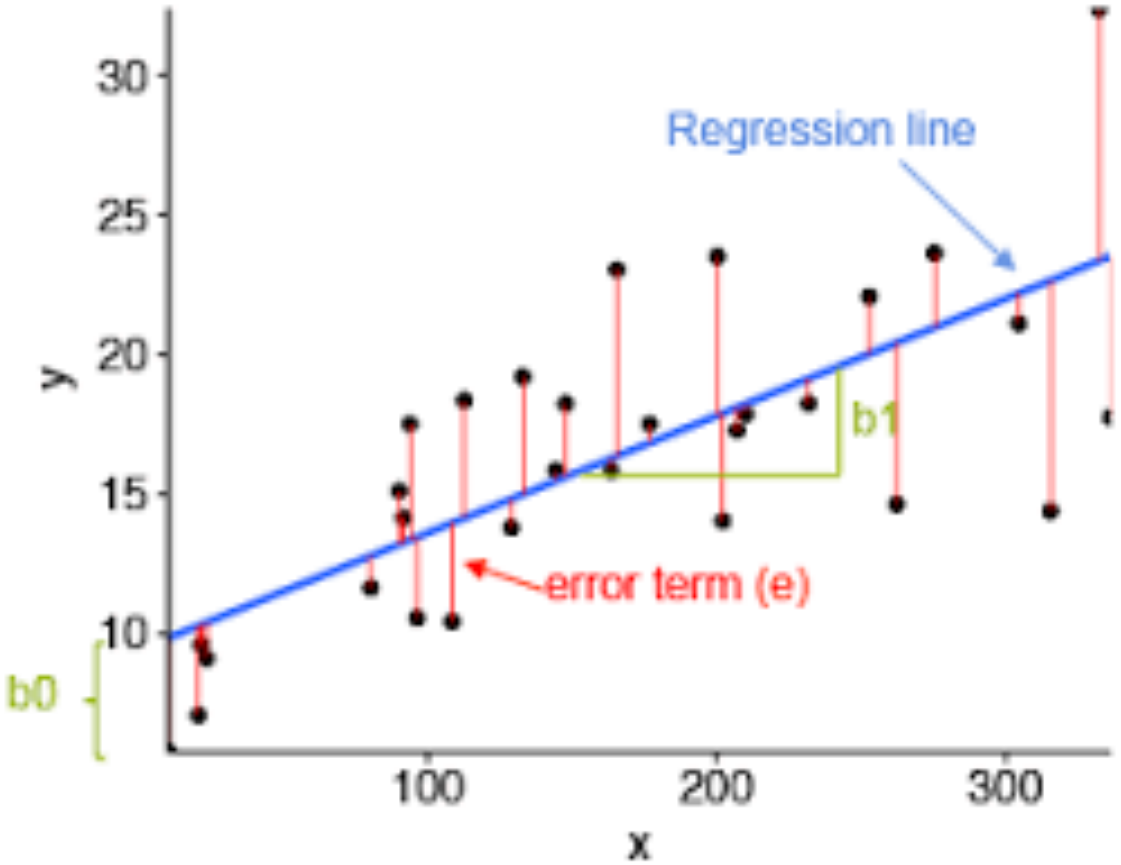
xiajingbo.math@gmail.com

2023-11-14

Table of contents I

Linear Regression

线性回归是一个经典的回归学习算法，有利于理解 Loss 函数和正则项的作用。我们从几张漫画开始。



Outline

Linear Regression with Least Square I

最小二乘线性回归

Notations:
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 linear regression model is to compute a regression function $f(x, w)$ such that approximate $\tilde{y}_i = f(x_i, w)$ to y_i :

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

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

[a] 该公式实际是一个简洁表示，完整表述应为 $f(x_i, w, b) = \sum_{j=1}^{p} w_j x_{ij} + b = w^T x_i + b$.
我们重写公式为 $f(x_i, w, b) := \hat{f}(\hat{x}_i, \hat{w})$，此处我们用 $\hat{x}_i = (x_i^T, 1)^T = (x_{i1}, \cdots, x_{ip}, 1)^T$，和 $\hat{w} = (w^T, b) = (w_1, \cdots, w_p, b)^T$ 作为辅助记号。
不是一般性，我们在后面使用简洁的公式表示。

Linear Regression with Least Square II

最小二乘线性回归

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

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

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. [a]

[a] 此处请留意向量范数 l_2被用以表示 Loss 函数。当然，这仅仅只是一个记号上的使用而已。

Linear Regression with Least Square III

最小二乘线性回归

Please notice that there is an explicit solution to this problem if $X^T X$ is an invertible matrix[1], say:

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

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

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

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

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

Let the gradient equal to zero, we have

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

关于这个结果的几个注解：[2] [3] [4] [5]

[2] 注：首先，这是一个显示解，求解过程中适度使用了一些矩阵微分的技巧。
[3] 其次，这个模型及其求解所对应的“机器学习”思想在哪里？
[4] 但是，这个解并不完美。为什么？
[5] 一个细节：针对 $\mathcal{J}(w)$的梯度下降是如何进行的，公式是？

Linear Regression with Least Square I

附：Gradient Descent

Gradient descent (aka., steepest descent) is for minimizing multidimensional smooth convex objective functions of the form $\mathcal{J} : \mathbb{R}^n \to \mathbb{R}$.

定理 (Gradient Descent)

1: ***Input:*** *Initial point* w_0*, gradient norm tolerance* ε
2: *Set t=0*
3: ***while*** $||\nabla \mathcal{J}(w_t)|| \geq \varepsilon$ ***do***
4: $w_{t+1} = w_t - \eta_t \nabla_t J(w_t)$
5: $t = t + 1$
6: ***end while***
7: ***Return:*** w_t

Here,

$$w_{t+1} = w_t - \eta_t \nabla_t J(w_t) \tag{8}$$

is the core iteration.

Outline

Consider a variant of **linear regression model**, **ridge regression model**, which takes a $L2$ regularizer:

$$\mathcal{J}_{Ridge}(w) = \frac{1}{n}||y - Xw||_2^2 + \lambda||w||_2^2. \text{ —Ridge regression.} \tag{9}$$

a *b*

---

[a]凸规划的结论告诉我们，增加了正则项后，以上 Loss 函数的最优化问题等同于

$$\min_w \frac{1}{n}||y - Xw||_2^2, \qquad s.t., ||w||_2 < C. \tag{10}$$

此处 C与 Lagrange 常数 λ有关.

[b]增加的正则项使得解搜索需要在 $||w||_2 < C$取值范围的约束下进行. 这减小了可行解的搜索空间。

几个问题：

增添了正则项后，对比最小二乘下的线性回归 w估计结论，对这个结论的影响在哪里？

对 $\mathcal{J}(w)$的梯度下降的训练影响在哪里？
这是可以通过列式求解获知的。请尝试。

接上述讨论，仅保证 $\mathcal{J}(w)$的凸性并不意味着寻优的高效性。

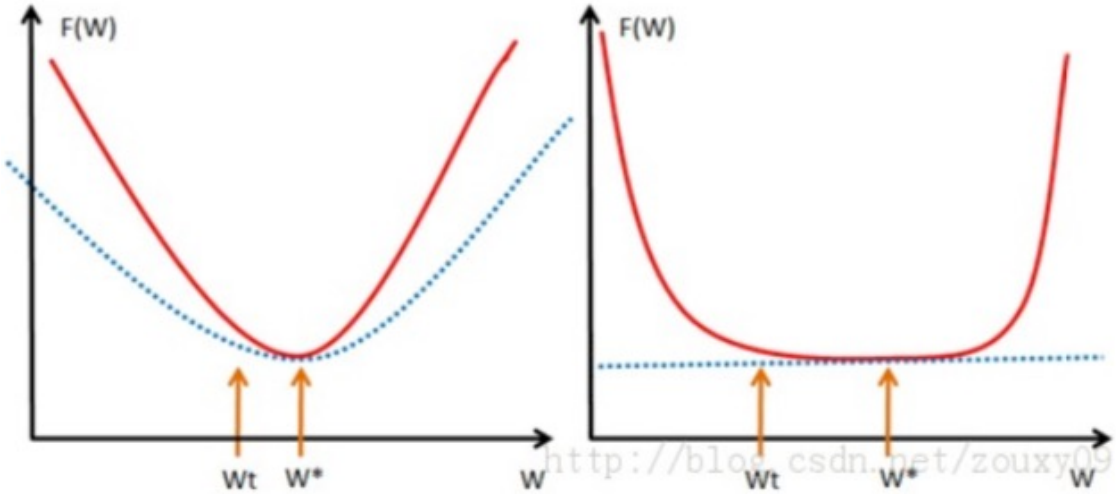


图 1: Convexity of loss function is not always amusing. (Take an example when w is a 2-dimensional vector)

正则项的加入，减小了寻优空间。——这是成功策略的一部分，但不是全部。

Linear Reg. and Regularization— Ridge and LASSO II

向量范数的挑选，L2 还是 L1？

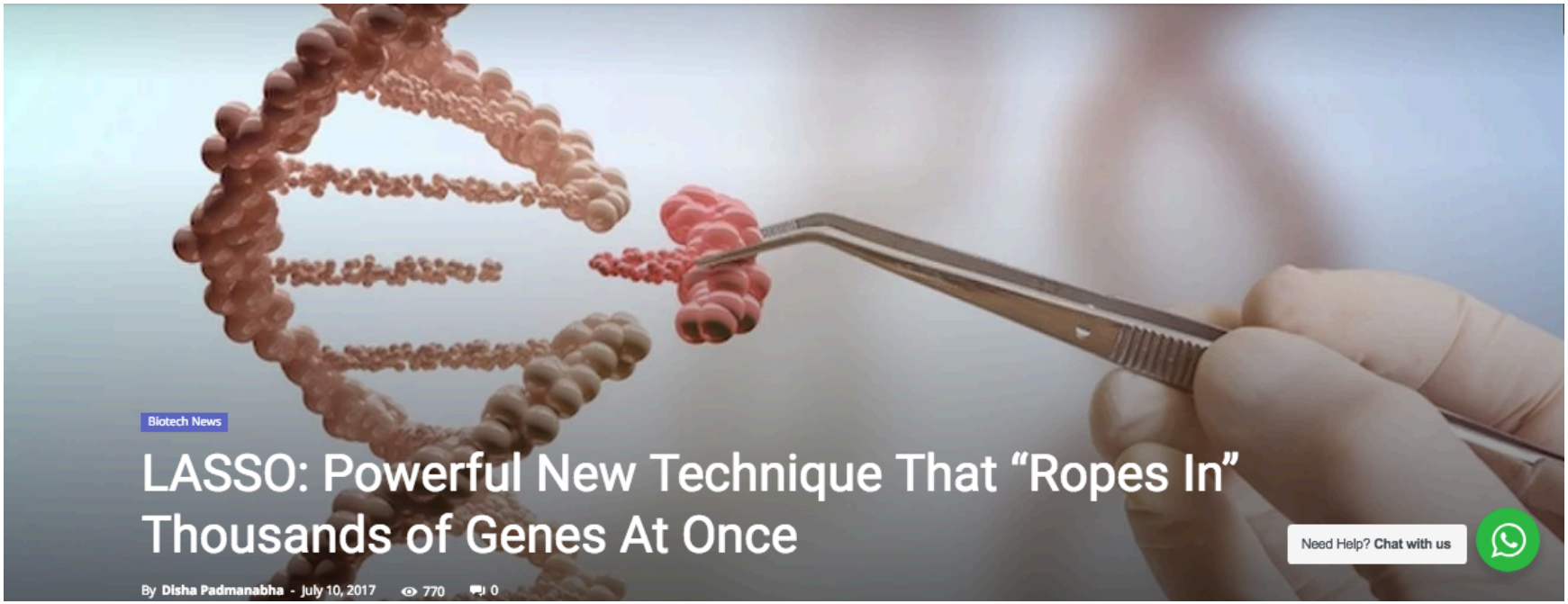
Question

在很多问题求解的情况下，我们希望线性回归能获得一个包含很多 0 分量的 w，以达到 “Sparsity”。为什么会这样？

怎么理解一个具有 Sparsity 特性的 w？例如在 Figure 1中？

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" [a].



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

利用 L_1范数，我们获得 **Lasso Regression** 的 Loss 函数。

$$\mathcal{J}_{Lasso}(w) = \frac{1}{n}||y - Xw||_2^2 + \lambda||w||_1^2. \text{ —Lasso regression.} \tag{11}$$

a b c

---

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

$$\min_w \frac{1}{n}||y - Xw||_2^2, \qquad s.t., ||w||_1 < C. \tag{12}$$

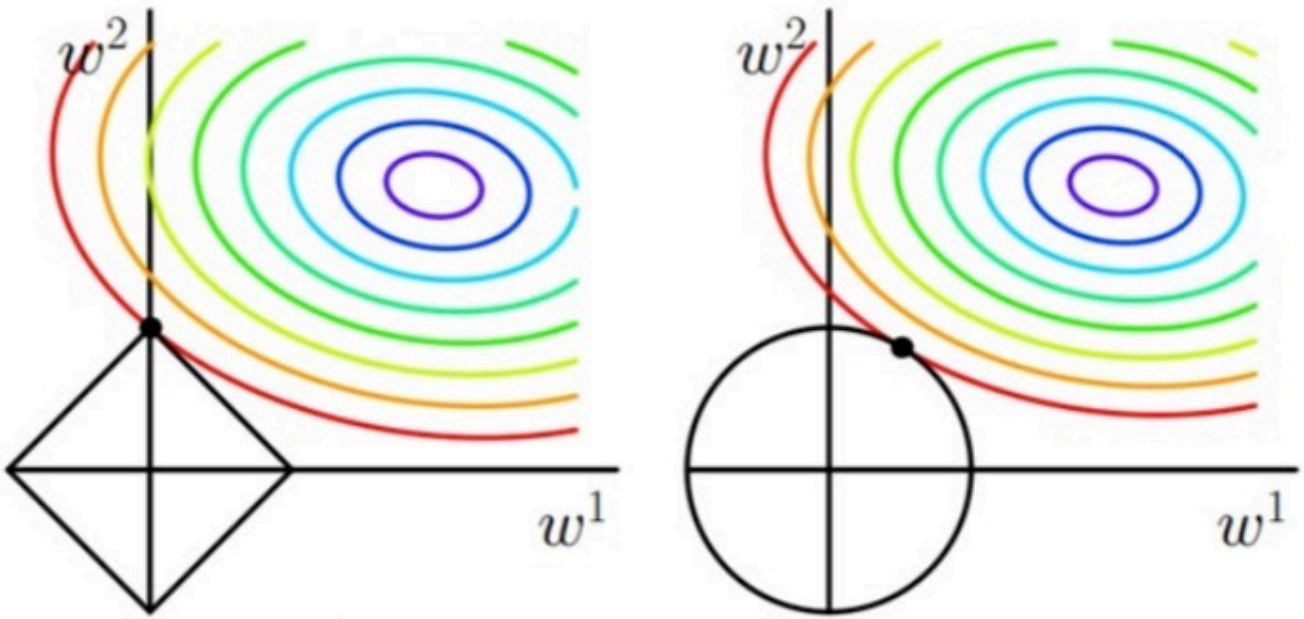
Here, C is a constant, related to λ.

[b]请对照 Ridge 回归，LASSO 的唯一区别在于正则项中的范数选择。不同的范数选择，为什么会带来 w的 Sparsity 呢？

[c]实际上，$L0$范数会带来更加直接的特征约减。选用 $L-1$的原因是为了计算的可能性。

Linear Reg. and Regularization— Ridge and LASSO V

向量范数的挑选，L2 还是 L1？



(a) ℓ_1-ball meets quadratic function. ℓ_1-ball has corners. It's very likely that the meet-point is at one of the corners.

(b) ℓ_2-ball meets quadratic function. ℓ_2-ball has no corner. It is very unlikely that the meet-point is on any of axes."

图 2: When $L2$-norm ball or $L1$-norm ball meet contour map of a quadratic function

$L1$ norm ball wins more chance to meet the contour map in the affine, leading to more zero items in w. Say, sparsity.

Outline

LASSO Regression 和 Python 代码示例

示例代码来自知乎[10].

例 (Command lines)

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

The raw data are:

例 (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
...
```

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

LASSO Regression 和 Python 代码示例

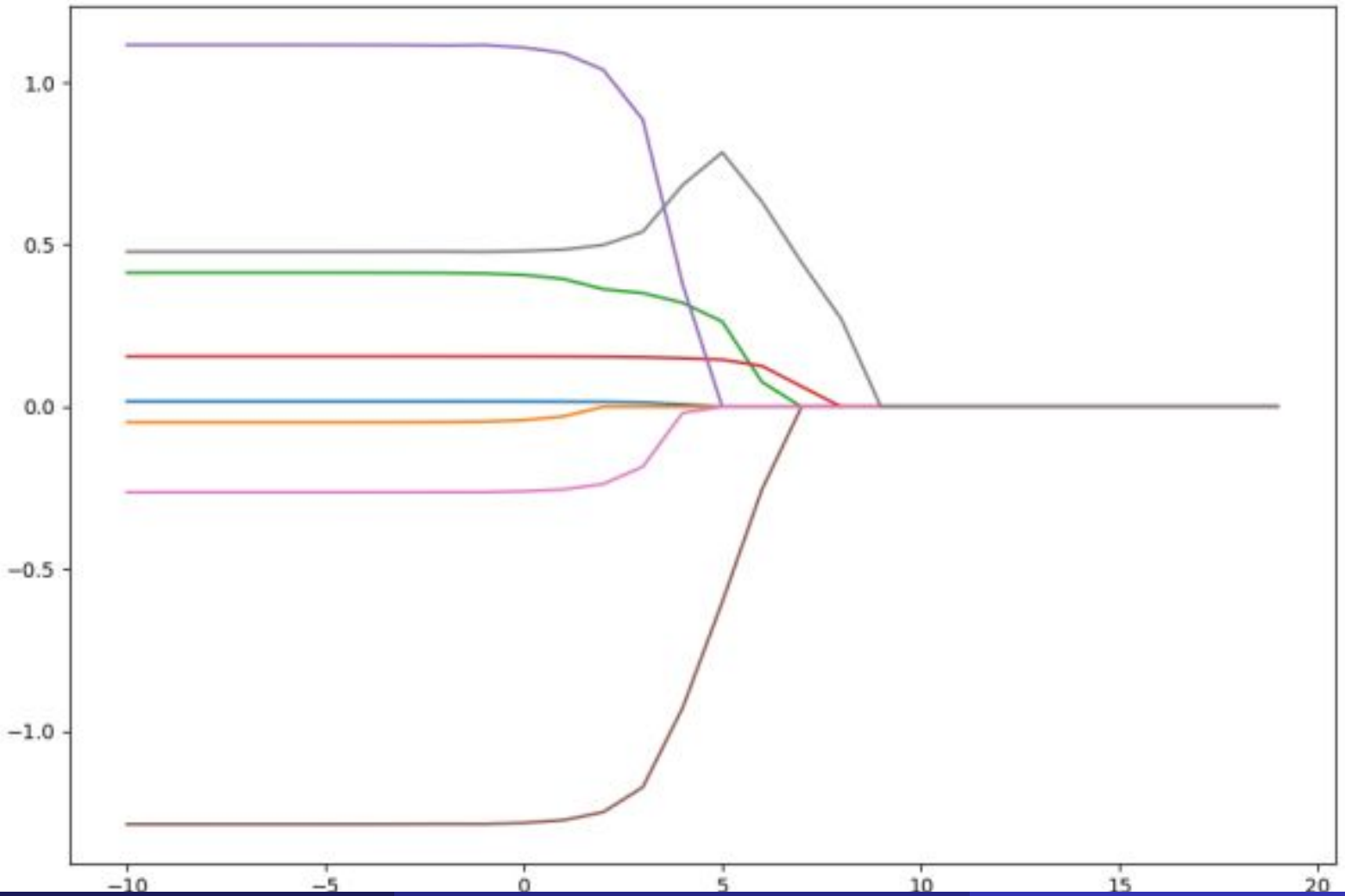
w approximate 0, while λ increases.

例 (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.]]
```

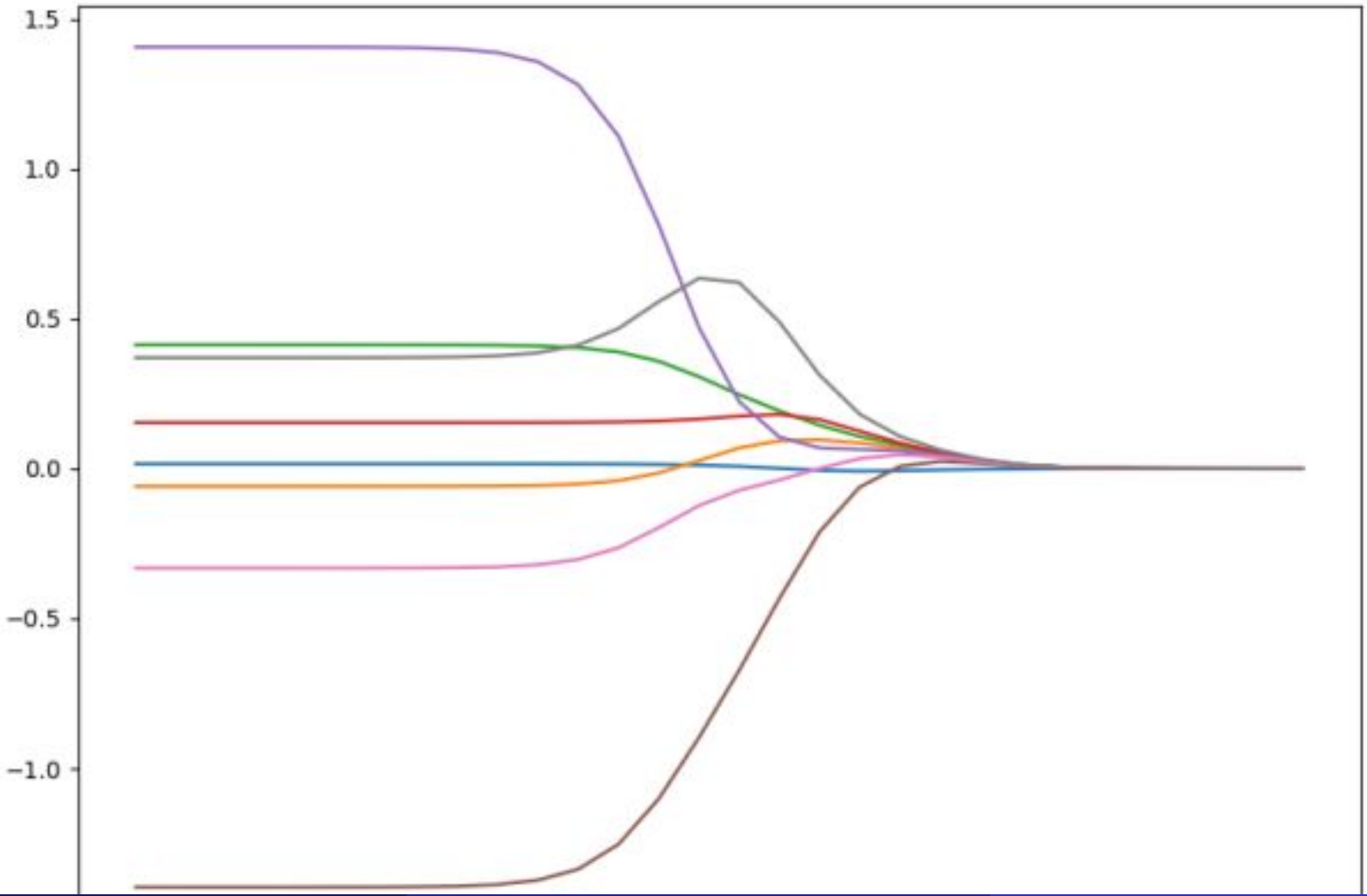
LASSO Regression 和 Python 代码示例

LASSO regression: w_i approximates zero when λ increases.



LASSO Regression 和 Python 代码示例

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 (8)) for updating w for Ridge regression (See formula (9))?

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

Outline

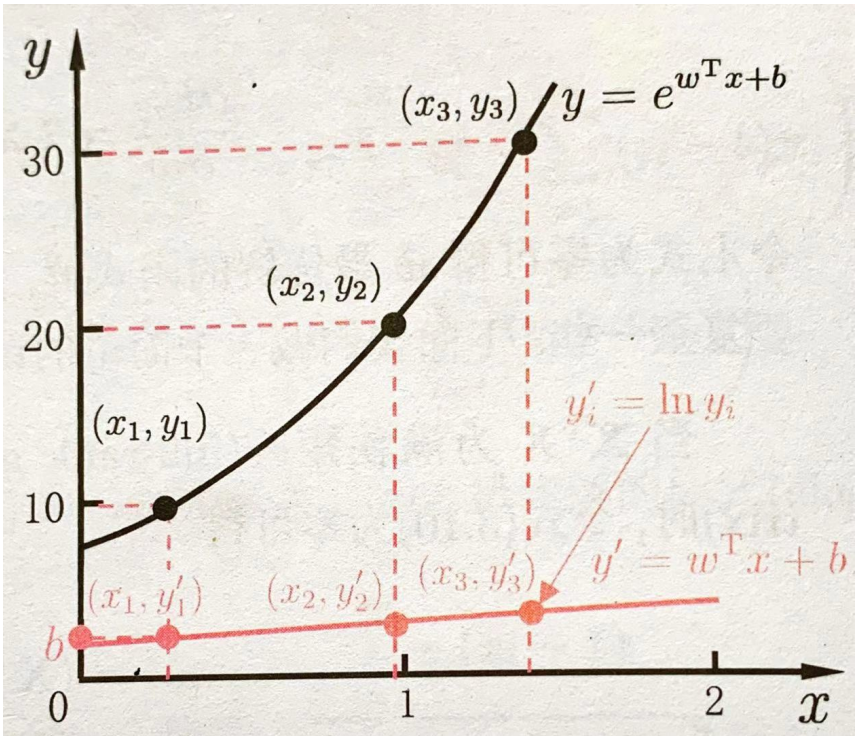
一般情形下的线性模型，从回归到分类

Log-linear regression

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,

$$\ln y = w^T x + b \tag{13}$$

is called "log-linear regression".



一般情形下的线性模型，从回归到分类

Generalized linear model

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

$$y = g^{-1}(w^T x + b) \tag{14}$$

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

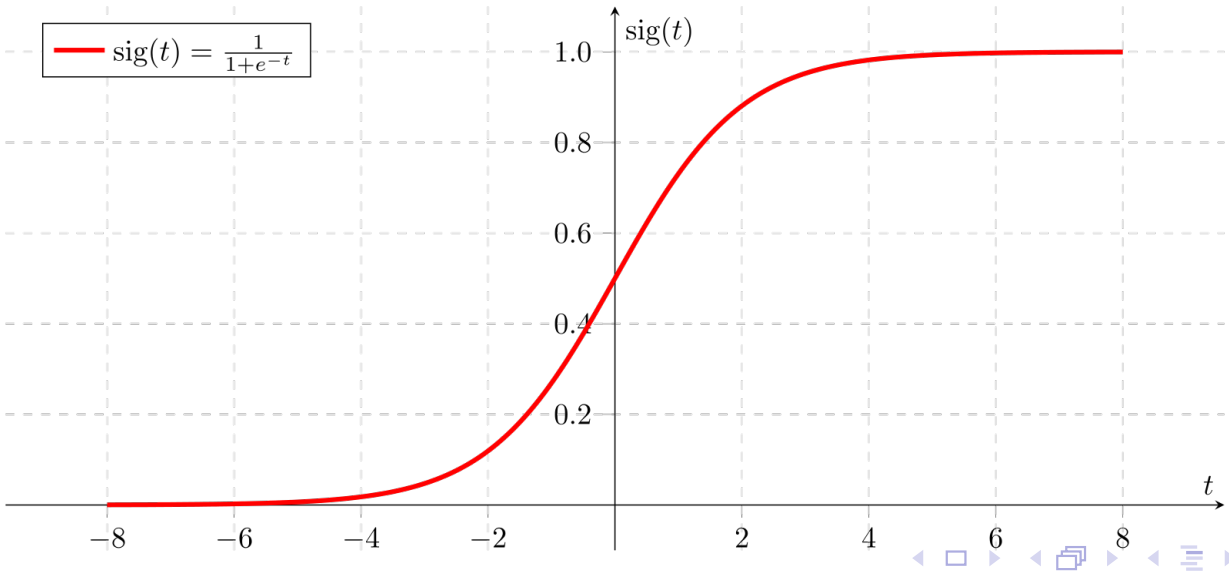
一般情形下的线性模型，从回归到分类

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":

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0. \end{cases} \tag{15}$$

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



一般情形下的线性模型，从回归到分类

更多的讨论，引向 Logistic 回归。
让我们转到下一章。

Acknowledgement

谢谢！