

# Machine Learning 2019

袁欣

2019 年 3 月 10 日

## 1 线性模型

### 1.1 西瓜数据

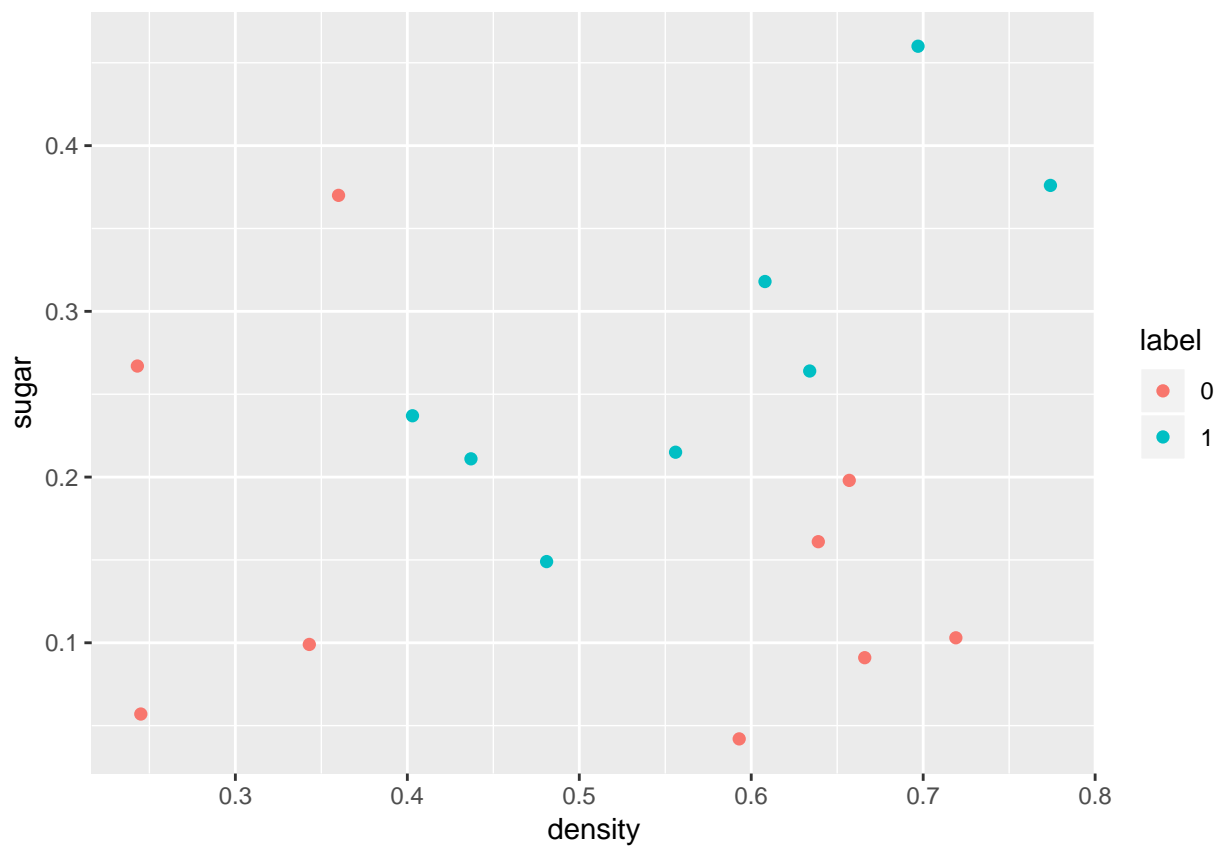
从 csv 文件中读取西瓜数据，并进行展示。

- 数据展示：

idx	density	sugar	label
1	0.697	0.460	1
2	0.774	0.376	1
3	0.634	0.264	1
4	0.608	0.318	1
5	0.556	0.215	1
6	0.403	0.237	1

```
##      density      sugar      label
##  Min.   :0.2430  Min.   :0.0420  0:9
## 1st Qu.:0.4030  1st Qu.:0.1030  1:8
## Median :0.5930  Median :0.2110
## Mean   :0.5326  Mean   :0.2128
## 3rd Qu.:0.6570  3rd Qu.:0.2670
## Max.   :0.7740  Max.   :0.4600
```

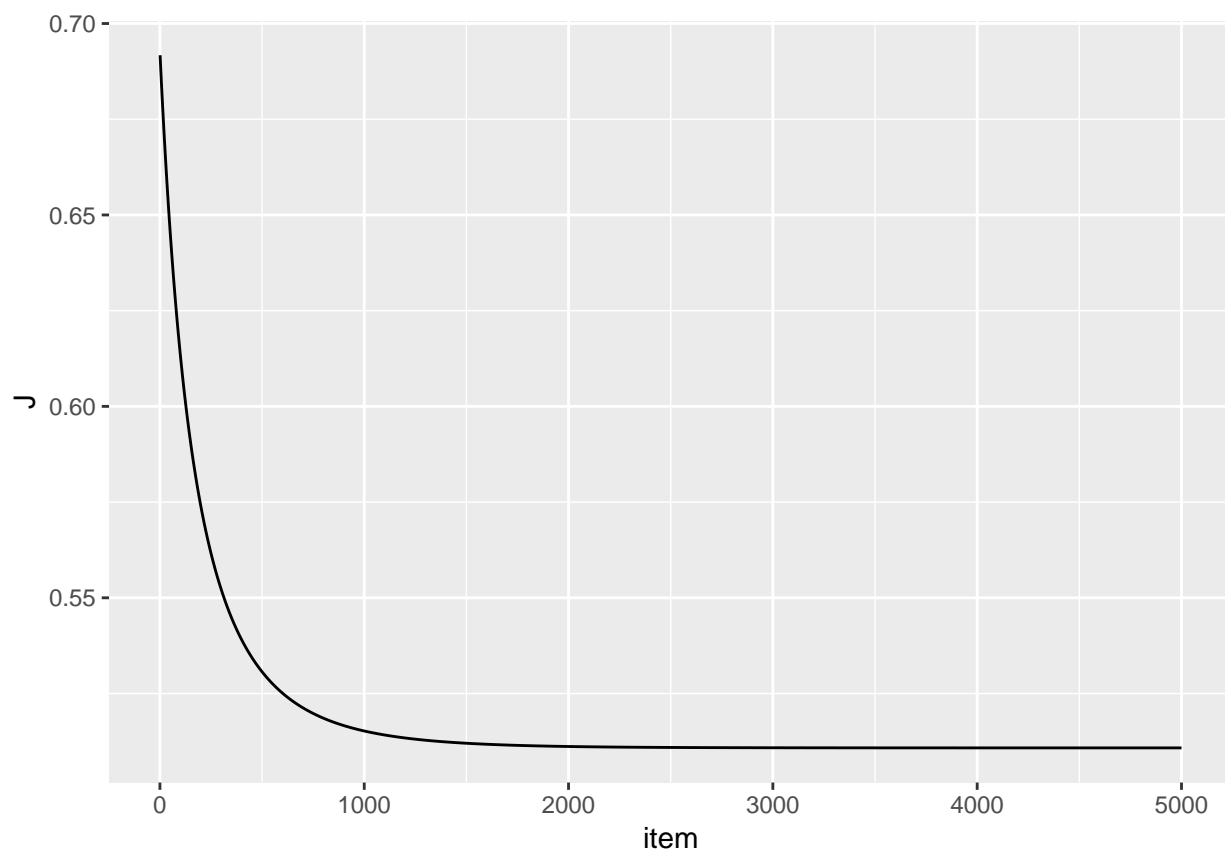
- 画图：



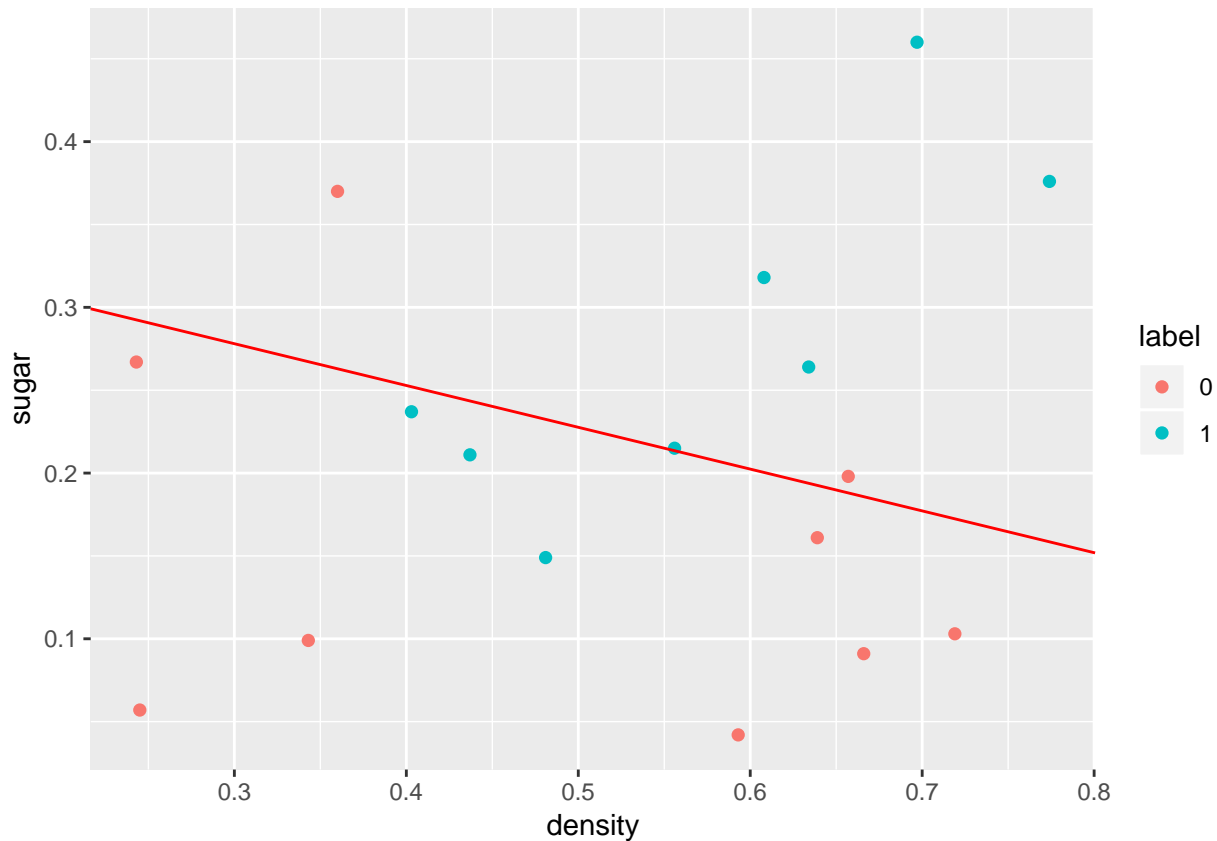
## 1.2 逻辑回归

- 编程实现逻辑回归的梯度下降算法
- 回归参数与代价函数曲线如下：

```
## [1] -4.419850  3.151482 12.495210
```



- 决策边界如下：



### 1.2.1 mapFeature

从上图可以看到，逻辑回归对西瓜集的分类是较差的。我们也可以直观的看到西瓜集是线性不可分的！所以这里引入了高阶特征，构建非线性边界去划分西瓜集。

构建方法为选择最高次数，将两变量映射到高阶上形成新特征。例如构建最高幂次为 6 的特征，此时会产生新特征如： $x_1^6$ 、 $x_2^6$ 、 $x_1^5x_2$ 、.....、 $x_1x_2$ 、 $x_2$ 、 $x_1$  共 28 个特征。

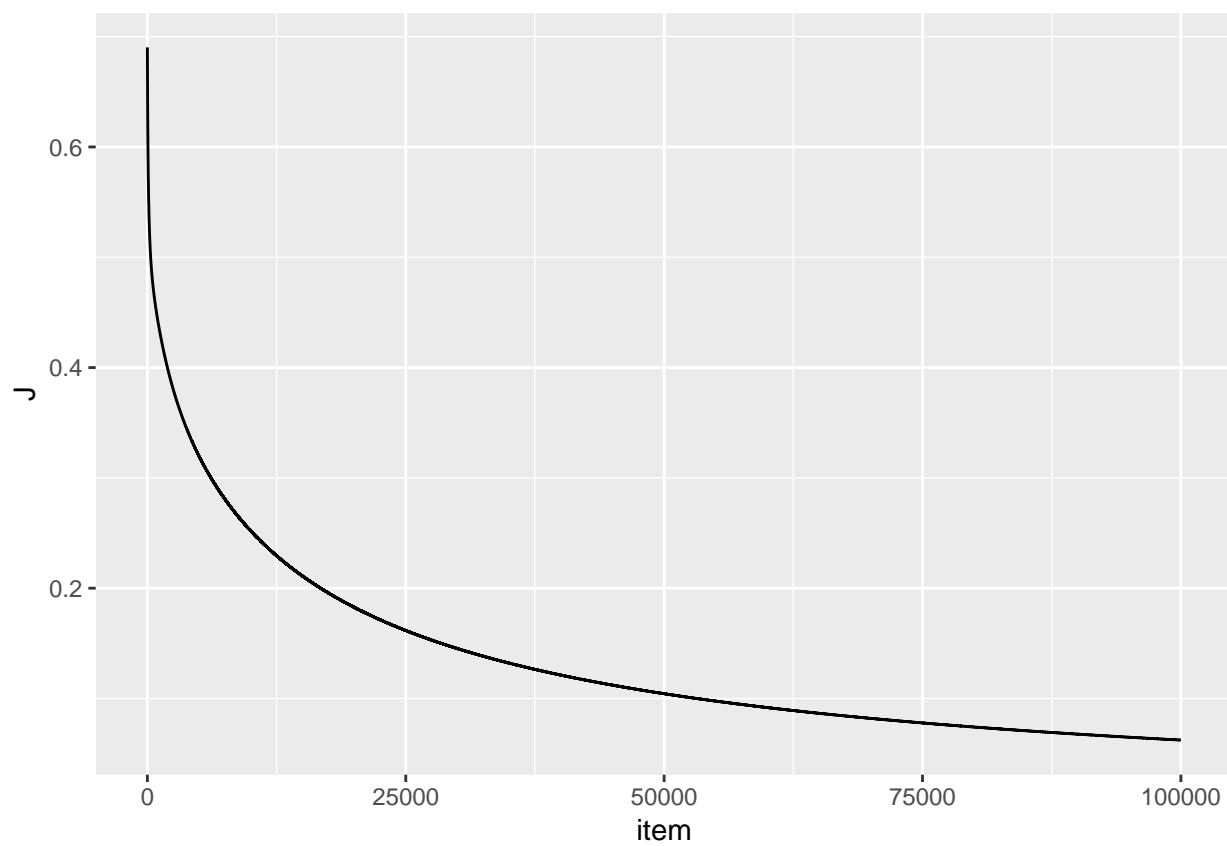
- 构建函数

### 1.2.2 正则化

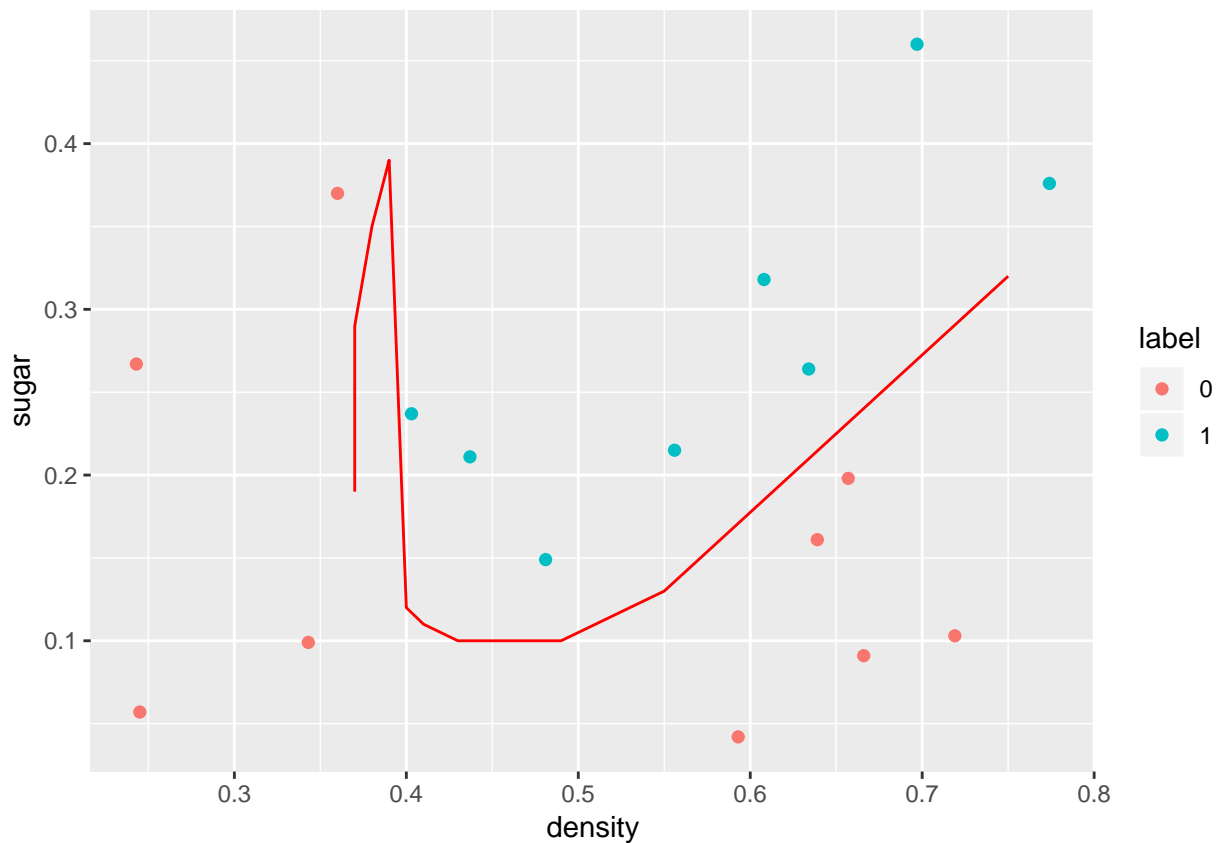
- 编程实现正则化逻辑回归的梯度下降算法
- 回归参数、预测精度、代价函数曲线如下：

```
## [1] -18.2576688 55.5141514 -10.7170173 -12.4644799 66.1030944
## [6] -49.1665058 -48.6936829 60.3009426 14.7737220 -32.2204811
## [11] -50.7437367 44.6194226 26.2238727 1.2997743 -15.4884559
## [16] -37.5326066 32.8471126 24.1358494 9.3827915 -0.7082506
## [21] -6.5816698 -21.6262370 25.2981189 19.6096944 9.8597332
## [26] 3.2328093 -0.5628648 -2.6294491
```

```
## [1] "prediction accuracy = 100 %"
```



- 非线性决策边界如下：



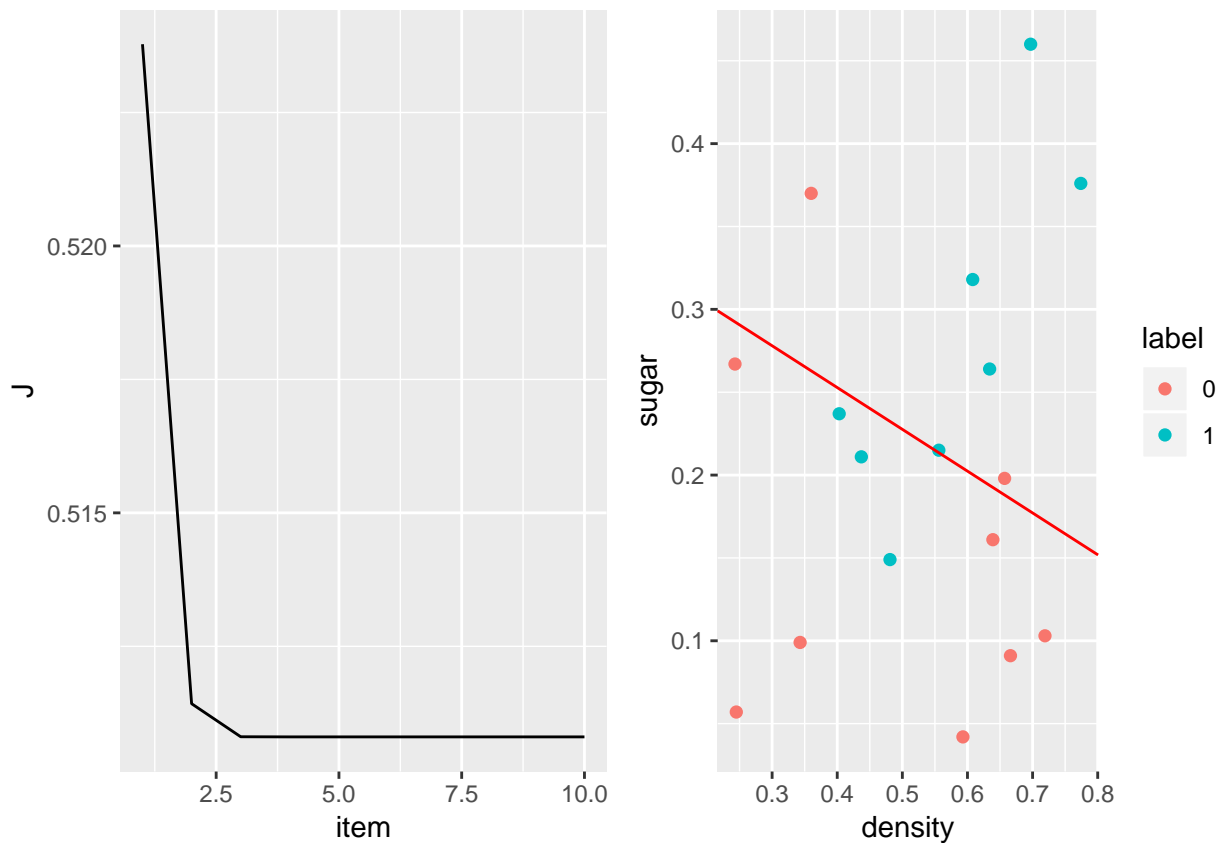
### 1.2.3 多分类问题

可以利用 One-vs-all 算法, 创建伪训练集, 例如: 预测天气 (Sunny、Cloudy、Rain、Snow), 可以学习四个逻辑回归, 判断哪个概率最高, 则属于哪一类。

### 1.2.4 利用牛顿法求解

- 回归参数、代价函数曲线、决策边界如下:

```
## [1] -4.428865  3.158330 12.521196
```

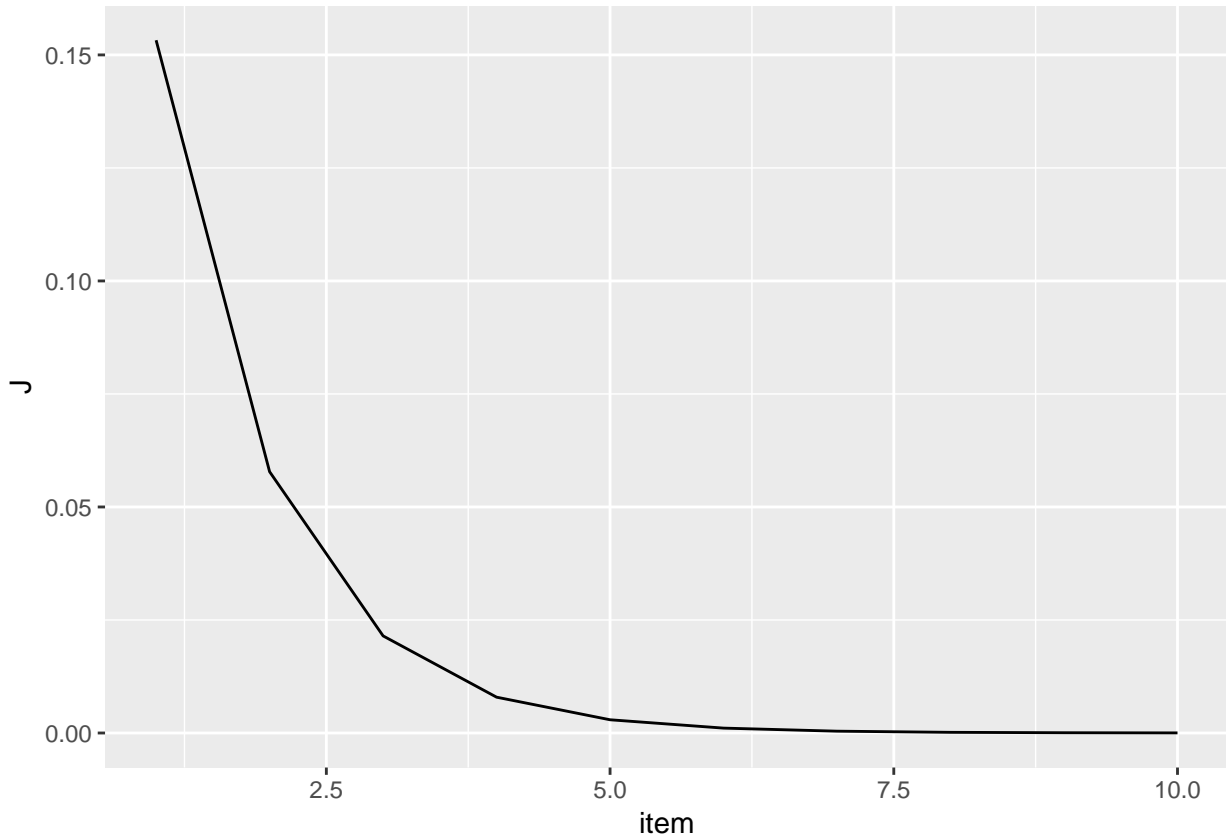


### 1.2.5 牛顿法正则化

- 回归参数、代价函数曲线如下：

```
## [1] 260.58899 -2305.59070 131.21556 5662.94237 -738.76919
## [6] -205.44986 -2183.57580 2171.56434 -331.49342 268.51504
## [11] -5259.69998 -94.64807 1782.42515 -318.80433 -889.53148
## [16] -933.53588 -2313.77204 1750.42685 388.24622 -1015.93544
## [21] -1193.47150 6438.19215 -2860.52426 1047.96320 531.73182
## [26] -599.01104 -1036.47230 -927.34067

## [1] "prediction accuracy = 100 %"
```



### 1.2.6 小结

1. 对比可以发现牛顿法比梯度下降法收敛速度快的多！
2. 在最小化代价函数的过程中还有很多更高级的方法，如 BFGS（共轭梯度）法、L-BFGS 等，它们的优点是选择参数  $\alpha$ 、收敛速度更快，但是它们也更复杂。
3. 在非线性边界画图中利用的是等值线绘图，也就是将图形分成一个个小的密度点，计算每个密度点的概率值。密度点概率值为 0.5 的等值线即为边界线。但是在实现过程中 `geom_isobands()` 并不能很好实现这个过程。Matlab 可以利用函数 `contour()` 实现，切记在利用这个函数之前将  $X$  转置。
4. 在 HessianMatrix 矩阵的求逆过程中并没有利用 `solve()` 函数，而是利用了 MASS 包里的 `ginv` 函数，当矩阵不可逆时，这个函数求得矩阵伪逆。类似于 Matlab 中 `inv` 与 `pinv` 的关系。

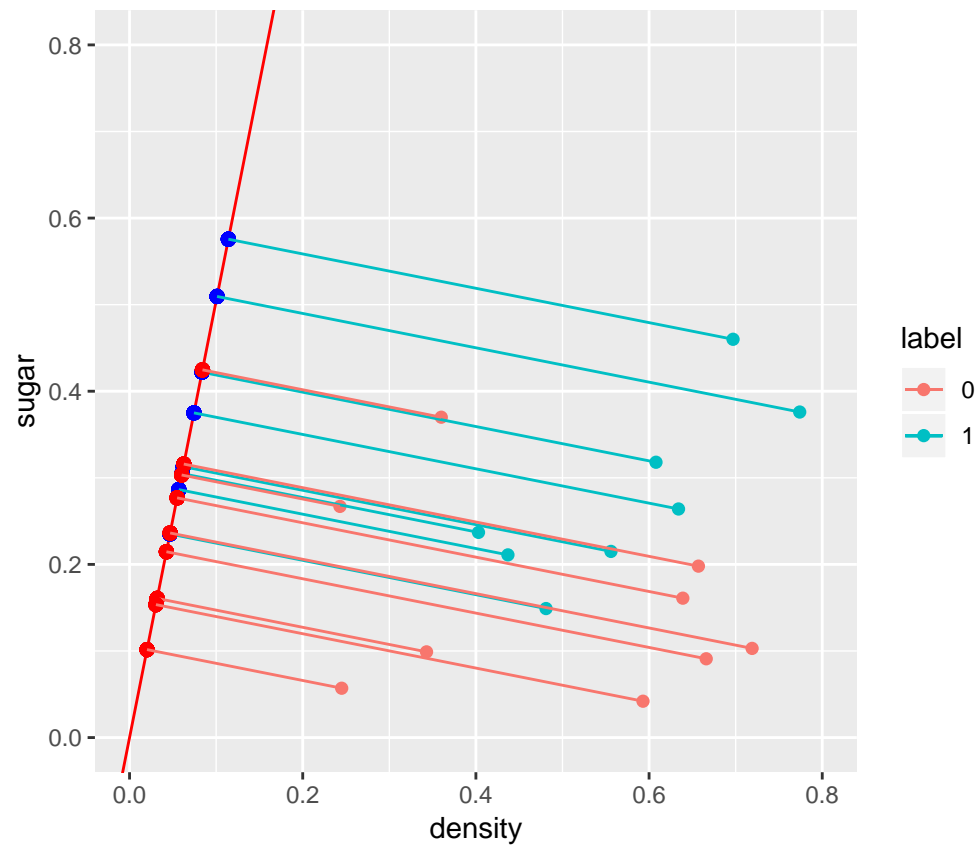
## 1.3 线性判别分析

线性判别分析（Linear Discriminant Analysis, 简称 LDA）是一种经典的线性学习方法。LDA 的思想非常朴素：给定训练集，设法将样本投射到一条直线上，使得同类样本的投影点尽可能接近、异类样本投影点尽可能远离；在对新样本进行分类时，将其投影到同样的直线上，再根据投影点的位置来确定新样本的类型。

```
##           [,1]
## [1,] -0.1465098
```



## [2,] -0.7387156



## 1.3.1 决策边界

