机器学习方法：回归

崔羽飞

01 前言

2017 年 3 月 15 日 AlphaGo 打败韩国棋手李世石 ,其背后谷歌公司的人工智能技术让机器学习这个领域重新变得火热起来。

2017 年 6 月，我开始接触机器学习这个领域。第一个学习的算法，当然是 KNN。不过这篇文章的主角不是 KNN，而是回归。回归分为线性回归和非线性回归。一般而言，说到回归，往往都想到是线性回归和逻辑回归。逻辑回归与线性回归本质上是一样的。下面主要根据自己学习进行的总结，欢迎大家交流。

02 什么是回归？

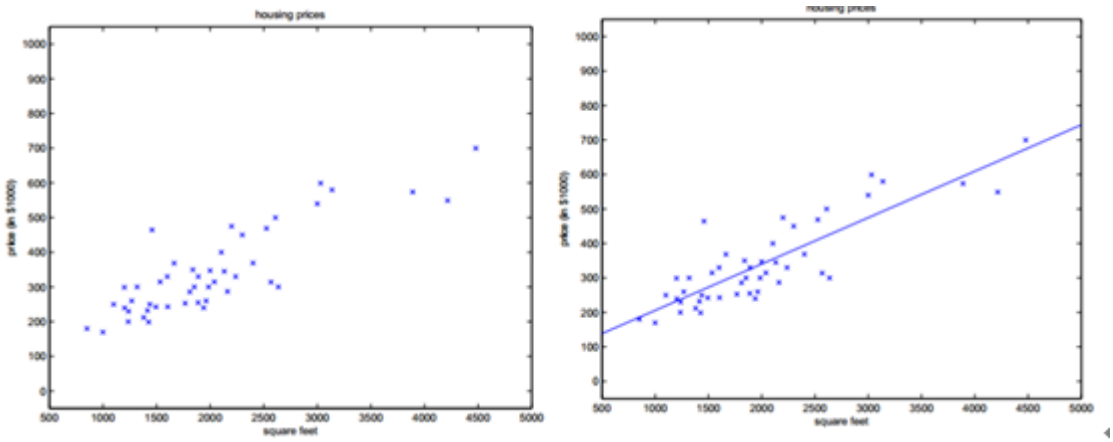
高尔顿曾经做过一个种豆子的实验。通过大量的数据统计发现，个子小的豆子往往倾向于产生比其更大的子代 ,而个子大的豆子则倾向于产生比其小的子代。高尔顿认为新个体在向豆子的平均尺寸“回归”。也就是说事物总是朝着某种“平均”发展，即回归到事物的本来面目。

在数学上 ,回归指研究一组随机变量(Y1 ,Y2 ,… ,Yi)和另一组(X1 ,X2 ,…，Xk)变量之间关系的统计分析方法，又称多重回归分析。通常 Y1，Y2，…，Yi 是因变量，X1、X2，…，Xk 是自变量。

I 输入变量为一个的时候

假如，目前知道某地区的房屋面积与价格的数据，我们总能通过一个方程来近似的表示房屋面积和价格的关系。如下

$$y=ax+b$$

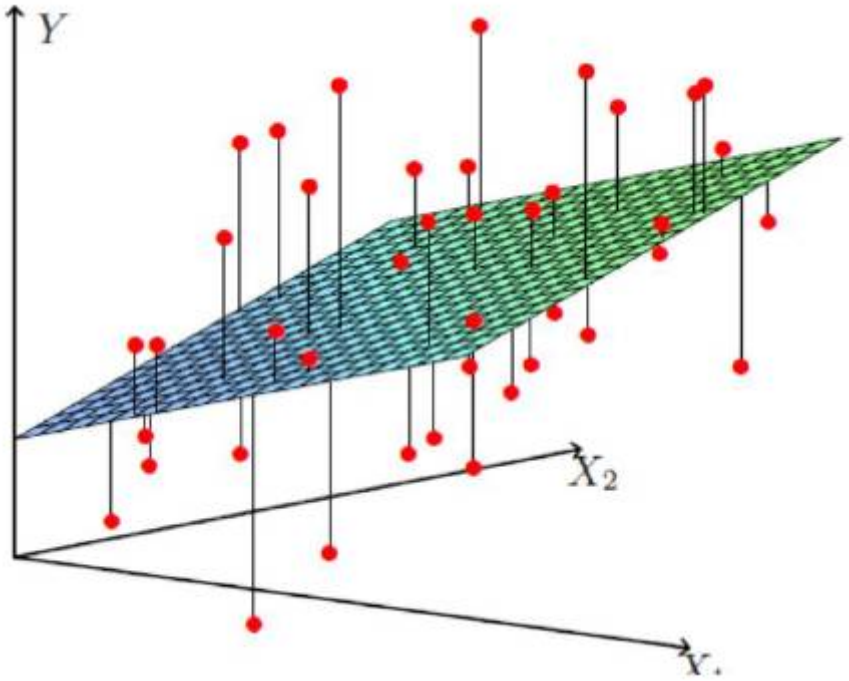


其中，a 和 b 就是我们要求的回归系数。

II 输入变量为多个的时候

假如 ,目前除了知道房屋的面积和价格外 ,我们还知道房屋具有的房间数目。此时我们也能通过一个方程来近似的表示房屋面积、房屋的房间数目和房屋价格之间的关系。如下

$$h_w(x) = \theta_0 + \theta_1 X_1 + \theta_2 X_2$$



$$h_\theta(x) = \sum_{i=0}^{n} \theta_i X_i = \theta^T X$$

其中，θ^T为回归系数。

03 回归的应用

逻辑回归在鸢尾花的例子

利用逻辑回归预测鸢尾花的类型。数据集共有 150 行数据，每一行是一个样本。每个样本包含 5 个字段，分别为花萼长度，花萼宽度，花瓣长度和花瓣宽度。数据集格式如下所示：

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
```

1.首先引入 python 中需要用到的包

```
import csv
import numpy as np
from sklearn.cross_validation import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.preprocessing.data import StandardScaler
from sklearn.pipeline import Pipeline
import pandas as pd
from sklearn import preprocessing
```

2.读入鸢尾花数据并将目标预测字段替换为 0,1,2

```
path = '../data/regression/iris.data'
path = '../data/regression/iris.data'
df = pd.read_csv(path, delimiter=',')
x = df.values[:,:4]
y = df.values[:, -1]

#通过sklearn来处理label问题
le = preprocessing.LabelEncoder()
le.fit(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'])
print le.classes_
y = le.transform(y)
```

预测的鸢尾花类比变为

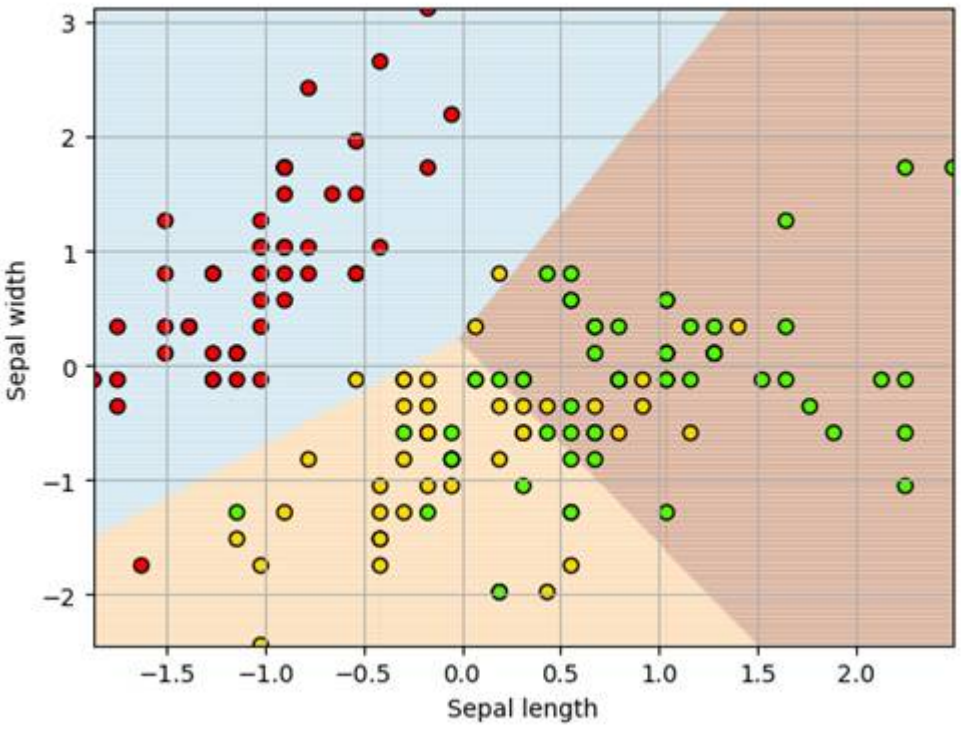
3.创建逻辑回顾对象，并显示分类结果

```
#创建逻辑回归对象
logreg = LogisticRegression()
dt_clf = logreg.fit(x, y.ravel())

# 画图
N, M = 500, 500
x1_min, x1_max = x[:, 0].min(), x[:, 0].max()
x2_min, x2_max = x[:, 1].min(), x[:, 1].max()
t1 = np.linspace(x1_min, x1_max, N)
t2 = np.linspace(x2_min, x2_max, M)
# 生成网格采样点
x1, x2 = np.meshgrid(t1, t2)
x_test = np.stack((x1.flat, x2.flat), axis=1)

y_hat = logreg.predict(x_test)
y_hat = y_hat.reshape(x1.shape)
plt.pcolormesh(x1, x2, y_hat, cmap=plt.cm.Paired, alpha=0.1)
plt.scatter(x[:, 0], x[:, 1], c=y, edgecolors='k', cmap=plt.cm.prism)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(x1_min, x1_max)
plt.ylim(x2_min, x2_max)
plt.grid()
plt.show()
```

分类结果的示意图如下：



上图为花萼长度（sepal length）和花萼宽度（sepal width）分布统计图。

4.显示预测结果

```
# 训练集上的预测结果
y_hat = dt_clf.predict(x)
y = y.reshape(-1)
#print y_hat.shape
#print y.shape
result = (y_hat == y)
#print y_hat
#print y
#print result
c = np.count_nonzero(result)
print "预测正确的个数：%d" %c
print '正确率：%.2f%%' % (100 * float(c) / float(len(result)))
```

结果如下：

```
[ 2.]
[ 2.]
[ 2.]]
预测正确的个数：119
正确率：79.33%
```

预测争取的个数为 119 个，正确率为 79.33%。

04 小结

回归是机器学习中的基础算法，但是它很重要。好多算法都是基于这个进行推导的。这里我只是对回归的基本含义做了简要的概括。下次我会对于线性回归和逻辑回归的具体推到过程进行详细讲解。

本文转自容数据服务集结号。

作者简介：

崔羽飞，毕业于北京邮电大学信息与通信工程专业，中国联通研究院工程师，主要研究方向数据加工、模型开发。