Abstract

本期我们学习线性分类-软输出-概率生成模型的另一种算法：朴素贝叶斯假设。

Idea

上一期我们学习的高斯判别分析是对数据集总体做出了高斯分布的假设，同时引入伯努利分布作为标签的先验，从而利用最大后验估计求得假设的参数。

而本期我们学习的朴素贝叶斯假设则是对数据的属性之间的关系做出了假设：条件独立性假设。

Algorithm

一般情况下，我们要得到 $p(x|y)$ 这个概率，由于 x 有 p 个维度，因此需要对这 p 个随机变量组成的联合分布进行采样，但我们知道：对于如此高维度的空间，需要采集极其庞大数量的样本才能获得较为准确的概率近似。

在一般的有向概率图模型中，通常对各个属性维度之间的条件独立关系做出了不同的假设，其中最为简单的假设就是在朴素贝叶斯模型中描述的条件独立性假设：

$$p(x|y)=\prod_{i=1}^{p}p(x_i|y)p(y) \tag{4}$$

用数学语言来描述：

$$x_i \perp x_j|y, \forall i \neq j \tag{5}$$

利用贝叶斯定理，对于单次观测：

$$p(y|x)=\frac{p(x|y)p(y)}{p(x)}=\frac{\prod_{i=1}^{p}p(x_i|y)p(y)}{p(x)} \tag{6}$$

与高斯判别分析类似，下面对数据的分布做出一些假设：

- x_i为离散变量:
  - 一般设x_i服从类别分布(Categorical)：$p(x_i=i|y)=\theta_i, \sum_{i=1}^{p}\theta_i=1$
- x_i为连续变量：
  - 一般设 x_i 服从高斯分布：$p(x_i|y)=N(\mu_i,\Sigma_i)$
- 二分类：
  - $y\sim Bernoulli(\phi):p(y)=\phi^y(1-\phi)^{(1-y)}$
- 多分类：
  - $y\sim Categorical\ Dist \quad p(y_i)=\theta_i \quad \sum_{i=1}^{k}\theta_i=1$

对于这些参数的估计，一般可以直接通过对数据集的采样来估计。参数估计好后，预测时代入贝叶斯定理求出后验概率。

Implement

```
import numpy as np
import os
```

```
os.chdir("../")
from models.linear_models import NaiveBayesClassifier

num_test = 100
x = np.linspace(0, 10, 1000)
k1, k2 = 0.1, 0.3
b1, b2 = 1, 2
x_train = x[:-num_test]
x_test = x[-num_test:]
v_1 = x_train * k1 + b1
v_2 = x_train * k2 + b2
train_data = np.r_[np.c_[x_train, v_1], np.c_[x_train, v_2]]
train_label = np.r_[np.ones_like(x_train), np.zeros_like(x_train)]

model = NaiveBayesClassifier()
model.fit(train_data, train_label)
print(model.get_params())

v_test = x_test * k2 + b2
data_test = np.c_[x_test, v_test]
print("accuary:", model.predict(data_test, 0))
```

```
([6.763511927008758, 0.0676351192700876], [4.499499499499499, 1.44994994994995],
[6.763511927008758, 0.6087160734307883], [4.499499499499499, 3.34984984984985])
accuary: 1.0
```