朴素贝叶斯(Naive Bayes)

作者：刘伟杰 日期：2015-11-27
参考：
[1]《统计学习方法》 李航 2012年3月第一版
[2]《机器学习实战》 Peter Harrington

1. 理论

1. 概述：

   贝叶斯算法中认为某个类在特征空间中出现某种特征的组合的概率为p。如果新输入一个实例，计算各个类出现这个新样本的特征的概率，选取概率最大的一个类作为新样本的分类（当然也可以根据贝叶斯公式给出各个分类的概率，即可能性）。p通过统计训练集中的样本来获得，其中会假设各个特征之间相互独立（这也是被称作naive的原因）。

2. 关键原理：

   当分类Y=ck时，出现特征组合X = x的概率为:

   $$P(X=x|Y=ck)=\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=ck)$$

   上公等号左边的P可以通过统计训练数据集得出，这里可以连乘是因为假设各个特效相互独立。

   由贝叶斯公式，就可以求出新样本X=x时，属于分类Y=c的概率:

   $$P(Y=ck|X=x)=\frac{P\left(X=x|Y=ck\right)P\left(Y=ck\right)}{\sum_{k}P\left(X=x|Y=ck\right)P\left(Y=ck\right)}$$

   P(Y=ck|X=x)可以想象成一个特征空间里的概率分布函数。

   朴素贝叶斯本质为损失函数选为0-1损伤函数，风险函数即为损失函数的期望。训练的目标是使得风险函数最小。

2. python实现

1. 我的实现：

https://github.com/autoliuweijie/MachineLearning/tree/master/bayes (https://github.com/autoliuweijie/MachineLearning/tree/master/bayes)

2. scikit-learen:

   示例：

```
#Import Library
from sklearn.naive_bayes import GaussianNB
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(p
model = GaussianNB()
# Train the model using the training sets and check score
model.fit(X, y)
#Predict Output
predicted= model.predict(x_test)
```

3. 扩展

1. 作回归：

   如果标签为序数型的数据，可以用求标签的期望值来做回归。

2. 参数估计：

   最大似然估计，可以用来估计一个系统的参数。最大似然估计的基本原理：假设参数为x，用x表示已发送事件的概率，求使得这个概率最大的x。

3. 贝叶斯网络：

   如果假设特征之间并不是相互独立的，模型就变成了贝叶斯网络，可以参考Bishop C. Pattern Recognition and Machine Learning, Springer,2006