Naïve Bayes Classifier

黄 欢

（计算机科学与技术系　计 33　2013011331）

1 Design

根据 Naïve Bayes 原理：

1.9. Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes' theorem states the following relationship:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots x_n \mid y)}{P(x_1, \dots, x_n)}$$

Using the naive independence assumption that

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

设计 adult.py 的过程如下：

1. 读入训练集，随机选取一定比例用于训练（issue 1）。
2. 训练数据：记录不同分类的数目，以及不同分类下各属性的数目；其中，对于数字属性，根据需要进行分级(issue 3)。
3. 读入测试集，并计算 $\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$ ，需要考虑 Zero-probabilities（issue 2）。
4. 对结果进行评估，计算当前训练数据比例(train_percent)、分段数量(bucket_num)、训练数据数量（train_num）下的准确率（accuracy）、召回率（p_recall）、F 值（f_measure）。

2 Results

参数:

train_percent #当前训练数据比例

is_bucket #是否对数字属性进行分段

bucket_num #分段数量

train_num #训练数据数量

中间参数：

Ak #正确分到 k 类的数量

Bk #错误分到 k 类的数量

Ck #属于该类而未被分到该类的数量

结果：

accuracy #准确率

p_precission #精确率

p_recall #召回率

f_measure #F 值

time #时间

根据如下公式：

$$Accuracy = \frac{\text{number of correctly classified records}}{\text{number of test records}}$$

$$precision: P = \frac{\sum_k \left(\frac{a_k}{a_k + b_k}\right)}{K}$$

$$recall: R = \frac{\sum_k \left(\frac{a_k}{a_k + c_k}\right)}{K}$$

$$f-score: F = \frac{2PR}{P+R}$$

可以得到：

```
E:\【learn】\3\32-2016春季学期\机器学习\Experiment\Experiment 1 Naive Bayes Clas
sifier\黄欢_2013011331\src>python adult.py
when train_percent = 1.0
bucket_num = 15
train_num = 32561
accuracy =  0.827160493827
p_precission = 0.764425733433
p_recall = 0.810255544118
f_measure = 0.786673719197
time = 0.967000007629

E:\【learn】\3\32-2016春季学期\机器学习\Experiment\Experiment 1 Naive Bayes Clas
sifier\黄欢_2013011331\src>_
```

Accuracy = 82.72%

3 Analysis

3.1 Issue 1: The impact of the size of training set

When is_bucket = 1, bucket_num = 15,

训练比例	0.05		0.5		1.0	
次数	数量	准确率	数量	准确率	数量	准确率
1	1609	82.46%	16240	82.49%	32561	82.72%
2	1582	82.70%	16214	82.67%		
3	1675	82.38%	16214	82.77%		
4	1642	82.91%	16336	82.66%		
5	1593	82.78%	16322	82.75%		
最大值	82.91%		82.77%		82.72%	
最小值	82.38%		82.49%		82.72%	
平均值	82.646%		82.668%		82.720%	

依据表格中数据，准确率在大体上随着训练集的增大而增大，但相差不多。

3.2 Issue 2: Zero-probabilities

假设在 training set 中，没有 x_i = k, y = c 的记录，则

$$\hat{P}(y=c|x_1,\dots,x_i=k,\dots,x_n)=0$$

解决方案为 Smoothing: $\hat{P}(x_i=k|y=c)=\frac{\#\{y=c,x_i=k\}+\alpha}{\#\{y=c\}+M\alpha}$，其中 M is the number of unique class label，且 alpha=1e-7.

3.3 Issue 3: Continuous and missing attributes

对于连续的属性值，可以根据需要进行分级，设置 is_bucket（是否分级）和合理的 bucket_num （分级数量）可以提高准确率。

缺失属性（'?'）看做独立的类别。

When train_percent = 1.0, train_num = 32561,

是否分级	是	是	是	是	否
分级数量	3	6	10	15	-
准确率	81.94%	81.87%	82.11%	82.72%	79.82%

依据表格中的数据，不进行分级的准确率最低，在一定范围内，准确率随着分级数量的增加而增加；可以尝试不断调整分级数量，从而得到较高的准确率。