法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

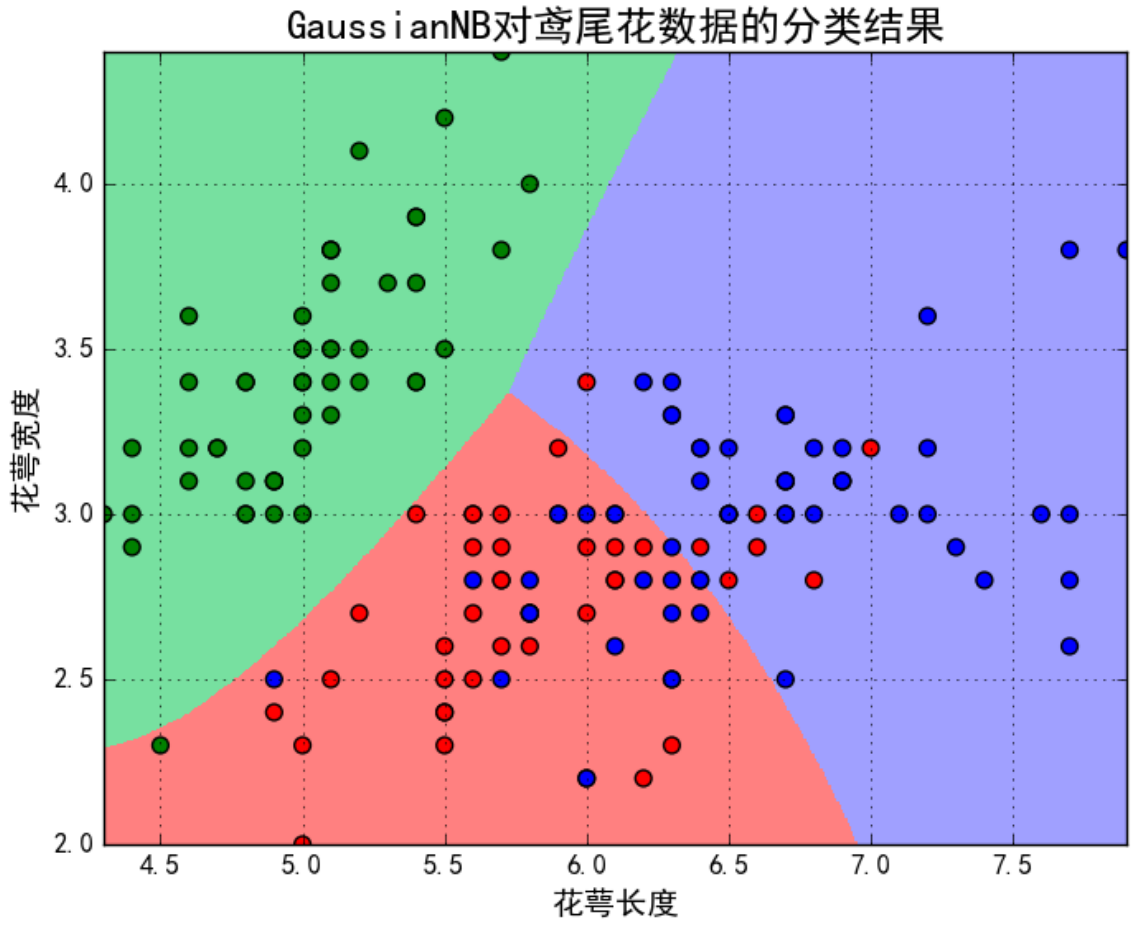
朴素贝叶斯实践

邹博

主要内容

- 朴素贝叶斯的推导和应用
- 文本数据的处理流程
- 使用TF-IDF得到文本特征

GaussianNB


GaussianNB对鸢尾花数据的分类结果
花萼宽度
花萼长度
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

GaussianNB / MultinomialNB

```
np.random.seed(0)
M = 20
N = 5
x = np.random.randint(2, size=(M, N))     # [low, high
x = np.array(list(set([tuple(t) for t in x])))
M = len(x)
y = np.arange(M)
print '样本个数：%d，特征数目：%d' % x.shape
print '样本：\n', x
mnb = MultinomialNB(alpha=1)    # 动手：换成GaussianNB(
mnb.fit(x, y)
y_hat = mnb.predict(x)
print '预测类别：', y_hat
print '准确率：%.2f%%' % (100*np.mean(y_hat == y))
print '系统得分：', mnb.score(x, y)
```

20.1.Iris_GaussianNB | 20.2.MultinomialNB_intro | 20.3.text_classification

```
[0 0 0 0 1]
[1 0 0 1 0]
[1 1 1 1 1]
[0 1 1 1 1]
[1 1 0 0 0]]
预测类别： [ 0  1  0  3  4  5  6  7  8  9 10 11 12 13  2 15 16]
准确率：88.24%
系统得分： 0.882352941176
2 :    [0 0 0 0 0] 被认为与 [1 1 0 1 0] 一个类别
14 :    [1 1 1 1 1] 被认为与 [0 0 0 0 0] 一个类别
```

朴素贝叶斯的假设

- 一个特征出现的概率，与其他特征(条件)独立(特征独立性)
  - 其实是：对于给定分类的条件下，特征独立
- 每个特征同等重要(特征均衡性)

朴素贝叶斯的推导

- 朴素贝叶斯(Naive Bayes，NB)是基于“特征之间是独立的”这一朴素假设，应用贝叶斯定理的监督学习算法。
- 对于给定的特征向量 $x_1, x_2, \cdots, x_n$
- 类别y的概率可以根据贝叶斯公式得到：

$$P(y \mid x_1, x_2, \cdots, x_n) = \frac{P(y)P(x_1, x_2, \cdots, x_n \mid y)}{P(x_1, x_2, \cdots, x_n)}$$

朴素贝叶斯的推导

- 使用朴素的独立性假设：

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

- 类别y的概率可简化为：

$$P(y \mid x_1, x_2, \cdots, x_n) = \frac{P(y)P(x_1, x_2, \cdots, x_n \mid y)}{P(x_1, x_2, \cdots, x_n)} = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, x_2, \cdots, x_n)}$$

- 在给定样本的前提下，$P(x_1, x_2, \cdots, x_n)$ 是常数：

$$P(y \mid x_1, x_2, \cdots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

- 从而：

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

高斯朴素贝叶斯Gaussian Naive Bayes

- 根据样本使用MAP(Maximum A Posteriori)估计$P(y)$，建立合理的模型估计$P(x_i \mid y)$，从而得到样本的类别。

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

- 假设特征服从高斯分布，即：

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi}\sigma_y}\exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

  - 参数使用MLE估计即可。

多项分布朴素贝叶斯Multinomial Naive Bayes

- 假设特征服从多项分布，从而，对于每个类别y，参数为$\theta_y = (\theta_{y1}, \theta_{y2}, \cdots, \theta_{yn})$，其中n为特征的数目，$P(x_i \mid y)$的概率为$\theta_{yi}$。
- 参数θ_y使用MLE估计的结果为：$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha \cdot n}$，$\alpha \geq 0$
- 假定训练集为T，有：$\begin{cases} N_{yi} = \sum_{x \in T} x_i \\ N_y = \sum_{i=1}^{|T|} N_{yi} \end{cases}$
- 其中，
  - $\alpha = 1$称为Laplace平滑，
  - $\alpha < 1$称为Lidstone平滑。

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

以文本分类为例

- 样本：1000封邮件，每个邮件被标记为垃圾邮件或者非垃圾邮件
- 分类目标：给定第1001封邮件，确定它是垃圾邮件还是非垃圾邮件
- 方法：朴素贝叶斯

分析

- 类别C：垃圾邮件c_1，非垃圾邮件c_2
- 词汇表，两种建立方法：
  - 使用现成的单词词典；
  - 将所有邮件中出现的单词都统计出来，得到词典。
  - 记单词数目为N
- 将每个邮件m映射成维度为N的向量**x**
  - 若单词w_i在邮件m中出现过，则$x_i=1$，否则，$x_i=0$。即邮件的向量化：$m \rightarrow (x_1,x_2.....x_N)$
- 贝叶斯公式：$P(c|\mathbf{x})=P(\mathbf{x}|c)*P(c) / P(\mathbf{x})$
  - $P(c_1|\mathbf{x})=P(\mathbf{x}|c_1)*P(c_1) / P(\mathbf{x})$
  - $P(c_2|\mathbf{x})=P(\mathbf{x}|c_2)*P(c_2) / P(\mathbf{x})$
    - 注意这里**x**是向量

分解

- $P(c|\mathbf{x})=P(\mathbf{x}|c)*P(c) / P(\mathbf{x})$
- $P(\mathbf{x}|c)=P(x_1,x_2\ldots x_N|c)=P(x_1|c)*P(x_2|c)\ldots P(x_N|c)$
  - 特征条件独立假设
- $P(\mathbf{x})=P(x_1,x_2\ldots x_N)=P(x_1)*P(x_2)\ldots P(x_N)$
  - 特征独立假设
- 带入公式： $P(c|\mathbf{x})=P(\mathbf{x}|c)*P(c) / P(\mathbf{x})$

- 等式右侧各项的含义：
  - $P(x_i|c_j)$：在c_j(此题目，cj要么为垃圾邮件1，要么为非垃圾邮件2)的前提下，第i个单词x_i出现的概率
  - $P(x_i)$：在所有样本中，单词x_i出现的概率
  - $P(c_j)$：在所有样本中，邮件类别c_j出现的概率

拉普拉斯平滑

- $p(x_1|c_1)$是指的:在垃圾邮件c_1这个类别中，单词x1出现的概率。
  - x_1是待考察的邮件中的某个单词
- 定义符号
  - n_1：在所有垃圾邮件中单词x_1出现的次数。如果x_1没有出现过，则$n_1=0$。
  - n：属于c_1类的所有文档的出现过的单词总数目。
- 得到公式：$$p(x_1|c_1)=\frac{n_1}{n}$$
- 拉普拉斯平滑：$$p(x_1|c_1)=\frac{n_1+1}{n+N}$$
  - 其中，N是所有单词的数目。修正分母是为了保证概率和为1
- 同理，以同样的平滑方案处理$p(x_1)$

对朴素贝叶斯的思考

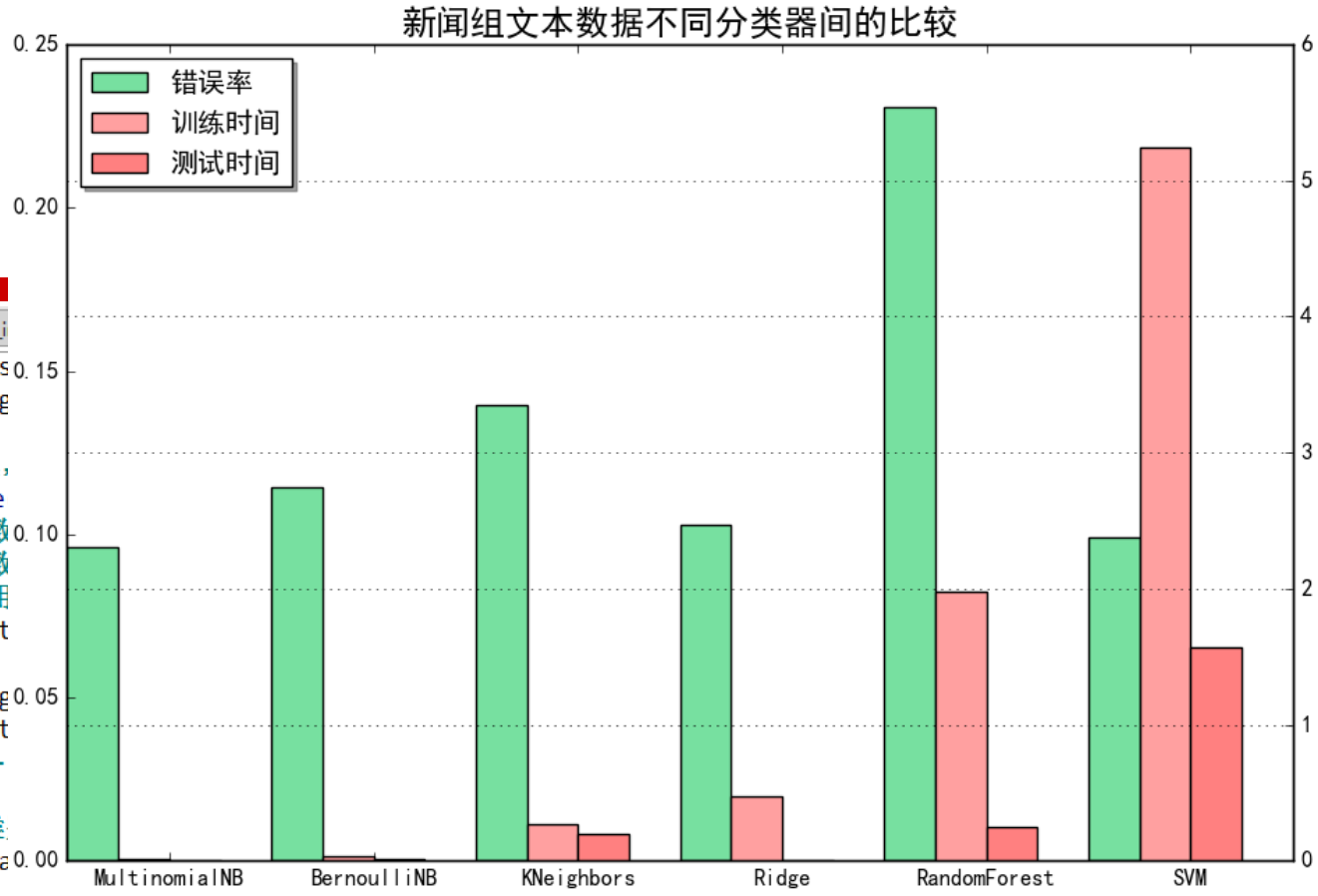
- 拉普拉斯平滑能够避免0/0带来的算法异常
- 要比较的是P(c1|x)和P(c2|x) 的相对大小，而根据公式P(c|x) =P(x|c)*P(c) / P(x)，二者的分母都是除以P(x)，实践时可以不计算该系数。
- 编程的限制：小数乘积下溢出怎么办？
- 问题：一个词在样本中出现多次，和一个词在样本中出现一次，形成的词向量相同
  - 由0/1向量改成频数向量或TF-IDF向量
- 如何判断两个文档的距离
  - 夹角余弦
- 如何给定合适的超参数 $\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha \cdot n}, \quad \alpha \geq 0$
  - 交叉验证

文本分类实验

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

- 实验数据：新闻组中的20个类别，原始文本数目约两万个，根据新闻组中文本的时间前后，划分成训练集(60%)和测试集(40%)。
  - 该数据最初应该是Ken Lang搜集整理。
- 数据获取：
  - 可使用sklearn.datasets.fetch_20newsgroups获取原始文本
  - 或者使用sklearn.datasets.fetch_20newsgroups_vectorized返回文本向量
- 该原始数据可以在该网页完整下载：
  - http://qwone.com/~jason/20Newsgroups/
    - 该课程的配套数据中已经包含该原始数据。

实验结果



20.1.Iris_GaussianNB.py × 20.2.MultinomialNB_i

```
    data_train = fetch_20news
    data_test = fetch_20newsg
    t_end = time()
    print u'下载/加载数据完成,
    print u'数据类型：', type
    print u'训练集包含的文本数
    print u'测试集包含的文本数
    print u'训练集和测试集使用
    categories = data_train.t
    pprint(categories)
    y_train = data_train.targ
    y_test = data_test.target
    print u' -- 前10个文本 --
    for i in np.arange(10):
        print u'文本%d(属于类
        print data_train.data
        print '\n\n'
vectorizer = TfidfVectorizer(input='content', stop_words='english'
x_train = vectorizer.fit_transform(data_train.data)  # x_train是稀
x_test = vectorizer.transform(data_test.data)
print u'训练集样本个数：%d，特征个数：%d' % x_train.shape
print u'停止词:\n',
pprint(vectorizer.get_stop_words())
feature_names = np.asarray(vectorizer.get_feature_names())

print u'\n\n====================\n分类器的比较：\n'
clfs = (MultinomialNB(),                    # 0.87(0.017), 0.002, 90.3
        BernoulliNB(),                      # 1.592(0.032), 0.010, 88.
        KNeighborsClassifier(),             # 19.737(0.282), 0.208, 86
        RidgeClassifier(),                  # 25.6(0.512), 0.003, 89.7
        RandomForestClassifier(n_estimators=200),   # 59.319(1.977
```

我们在这里

- http://wenda.ChinaHadoop.cn
  - 视频/课程/社区
- 微博
  - @ChinaHadoop
  - @邹博_机器学习
- 微信公众号
  - 小象
  - 大数据分析挖掘

感谢大家！

恳请大家批评指正！