信息检索导论--中文文本分类

授课人：李永可

邮箱：lyk@xjau.edu.cn

目 录

分类器设计

- 文本分类的方法大部分来自于模式分类，基本上可以分为三大类：
  - 一种是基于统计的方法，如Naïve Bayes，KNN、类中心向量、回归模型、支持向量机、最大熵模型等方法
  - 另一种是基于连接的方法，即人工神经网络
  - 还有一种是基于规则的方法，如决策树、关联规则等，这些方法的主要区别在于规则获取方法

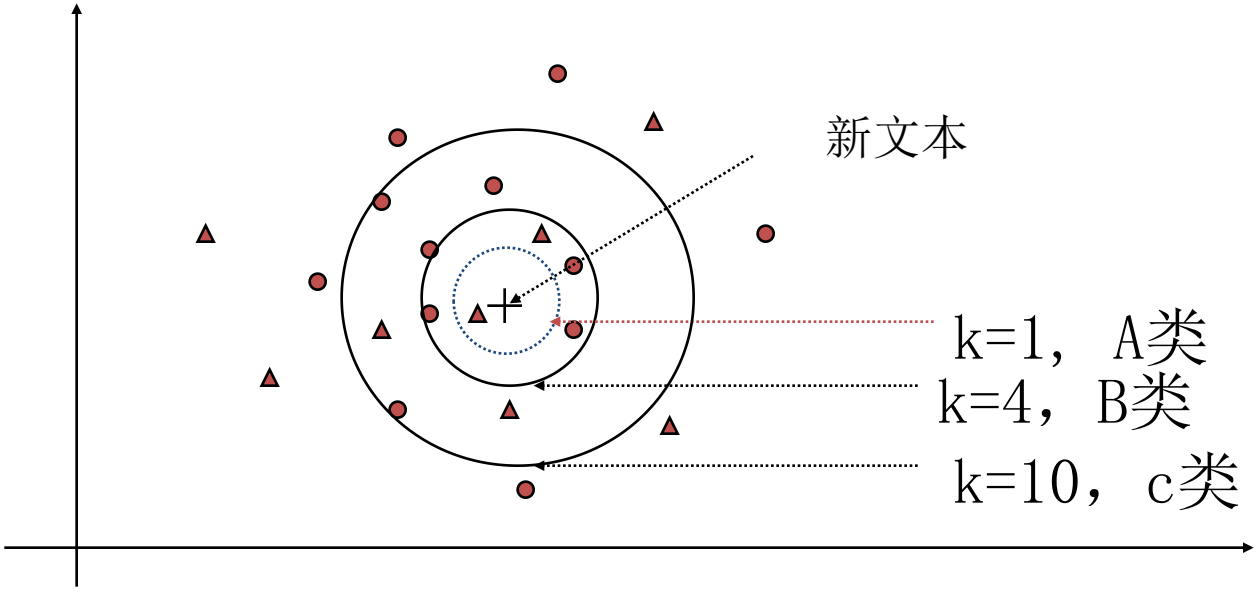
分类器设计

- K近邻算法-KNN
- 决策树算法-Decision Tree
- 神经网络算法- Neural Networks
- 支持向量机算法-SVM
- 朴素贝叶斯算法- Naïve Bayes

分类器设计

K近邻算法-KNN

- 基本思想是：
  - 在给定新文本后，考虑在训练文本集中与该新文本距离最近(最相似)的K篇文本
  - 根据这K篇文本所属的类别判定新文本所属的类别



分类器设计

K近邻算法-KNN

- 具体的算法步骤 ：
  - 根据特征项集合重新描述训练文本向量
  - 在新文本到达后，根据特征词，确定新文本的向量表示
  - 在训练文本集中选出与新文本最相似的K个文本，计算公式为：

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{M} w_{ik} \times w_{jk}}{\sqrt{(\sum_{k=1}^{M} w_{ik}^{2})(\sum_{k=1}^{M} w_{jk}^{2})}}$$

其中，K值的确定目前没有很好的方法，一般先定一个初始值，然后根据试验测试的结果调整K值，一般初始值定在几百到几千之间

分类器设计

K近邻算法-KNN

- 在新文本的k个邻居中，依次计算每类的权重，计算公式如下：

$$p(\vec{x},c_j)=\sum_{\vec{d}_i\in KNN} sim(\vec{x},\vec{d}_i)y(\vec{d}_i,c_j)$$

其中，$\vec{x}$ 为新文本的特征向量，$sim(\vec{x},\vec{d}_i)$ 为相似度计算公式，与上一步骤的计算公式相同，而 $y(\vec{d}_i,c_j)$ 为类别属性函数，即如果 $\vec{d}_i$ 属于类 c_j，那么函数值为1，否则为0；

- 比较每类的权重，将文本分到权重最大的那个类别中

分类器设计

决策树算法–Decision Tree

- 决策树方法的起源是概念学习系统CLS，然后发展到ID3方法而为高潮，最后又演化为能处理连续属性的C4.5。有名的决策树方法还有CART和Assistant

分类器设计

决策树的表示法

- 决策树通过把实例从根节点排列到某个叶子节点来分类实例，叶子节点即为实例所属的分类。
- 树上的每一个节点说明了对实例的某个属性的测试，并且该节点的每一个后继分支对应于该属性的一个可能值

分类器设计

ID3决策树算法简介

基本思路是不断选取产生信息增益最大的属性来划 分样例集和，构造决策树。信息增益定义为结点与其子结点的信息熵之差。

$$Entropy\ (S) = -\sum_{i=1}^{n} P_i \log P_i$$

Pi为子集合中不同性(而二元分类即正样例和负样例)的样例的比例。

分类器设计

ID3决策树算法简介

这样信息收益可以定义为样本按照某属性划分时造成熵减少的期望，可以区分训练样本中正负样本的能力，其计算公式是

$$Gain(S,A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- V(A)是属性A的值域
- S是样本集合
- S_v 是S中在属性A上值等于v的样本集合

分类器设计

ID3算法实例

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

分类器设计

计算信息增益

$$Values(Wind) = Weak, Strong$$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$

$$Gain(S, Wind) = Entroy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy$$

$$= Entropy(S) - (8/14)Entropy(S_{Weak}) - (6/14)Entropy(S_{Strong})$$

$$= 0.949 - (8/14)0.811 - (6/14)1.00$$

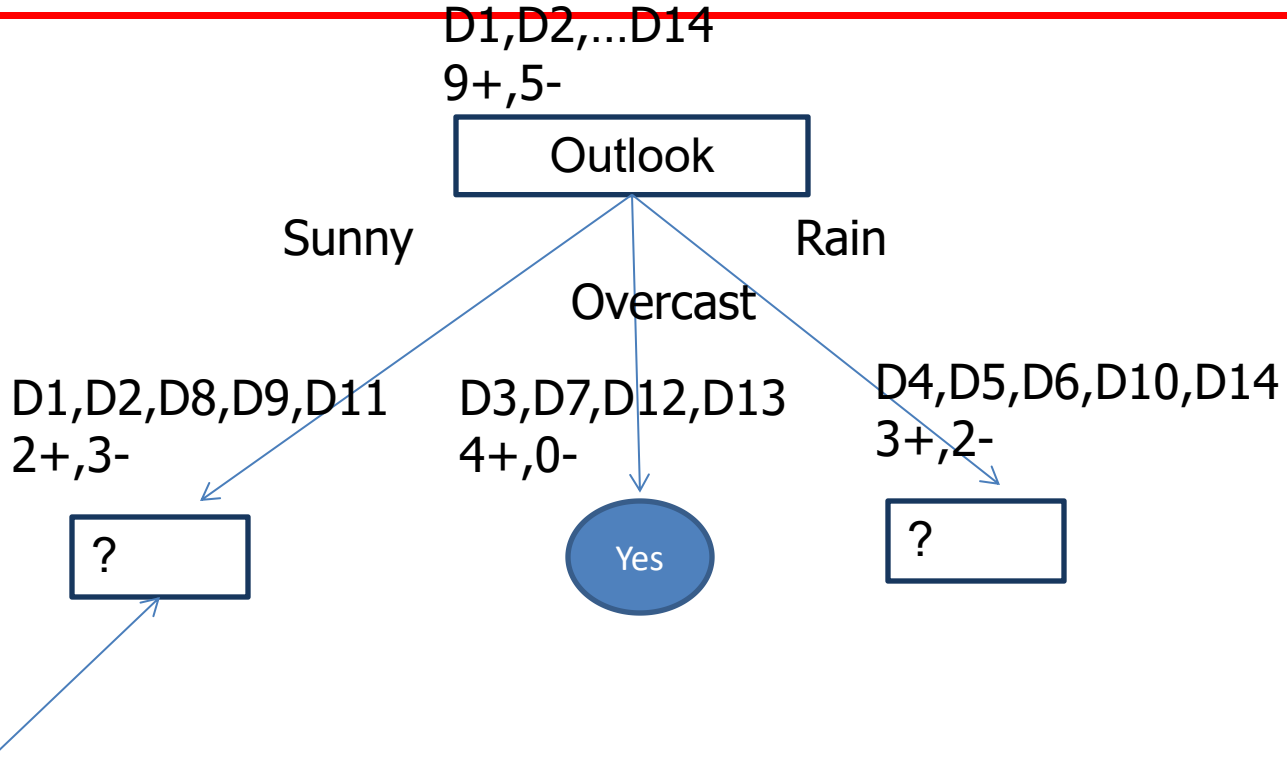
$$= 0.048$$

分类器设计

不同属性的信息增益

- 计算各属性的熵值
  - Gain(S,Outlook)=0.246
  - Gain(S,Humidity)=0.151
  - Gain(S,Wind)=0.048
  - Gain(S,Temperature)=0.029
- 可以看到，Outlook得信息增益最大

分类器设计

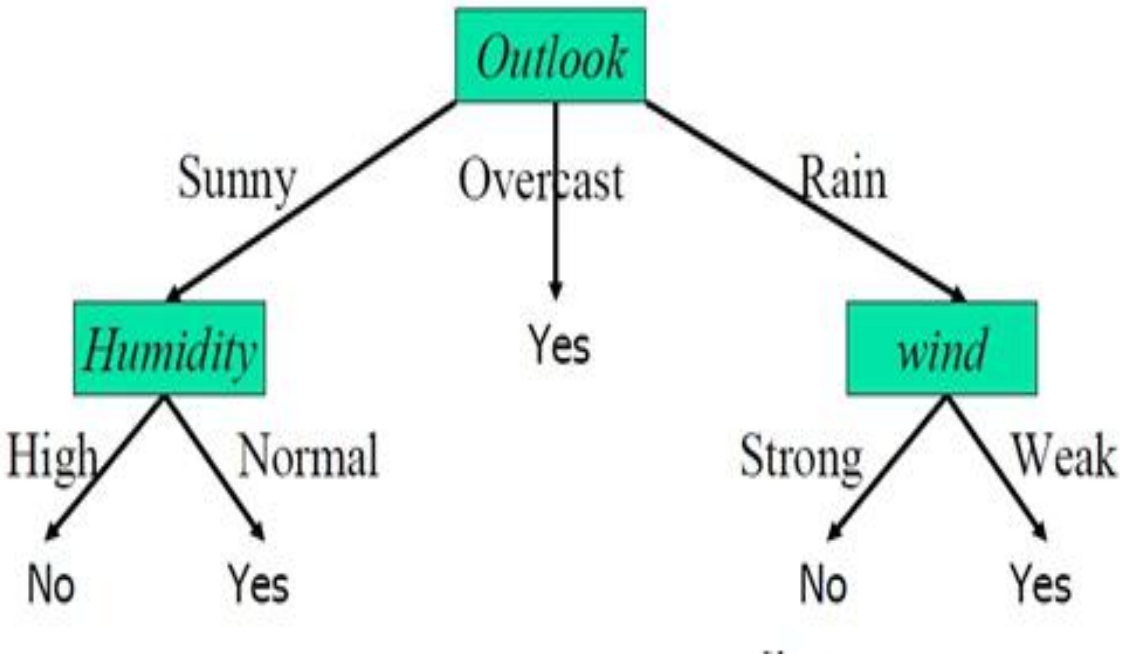


哪一个属性在这里被测试？

S_{sunny}={D1,D2,D8,D9,D11}
Gain（S_{sunny},Humidity）=0.970-(3/5)0.0-(2/5)0.0=0.970
Gain(S_{sunny}, Temperature)=0.970-(2/5)0.0-(2/5)1.0-(1/5)0.0=0.570
Gain(S_{sunny}, Wind)=0.970-(2/5)1.0-(3/5)0.918=0.019

分类器设计

最终得到的决策树



有了决策树后，就可以根据气候条件做预测了
例如如果气候数据是{Sunny,Cool,Normal,Strong} ，根据决策树到左侧的yes叶节点，可以判定属于P。

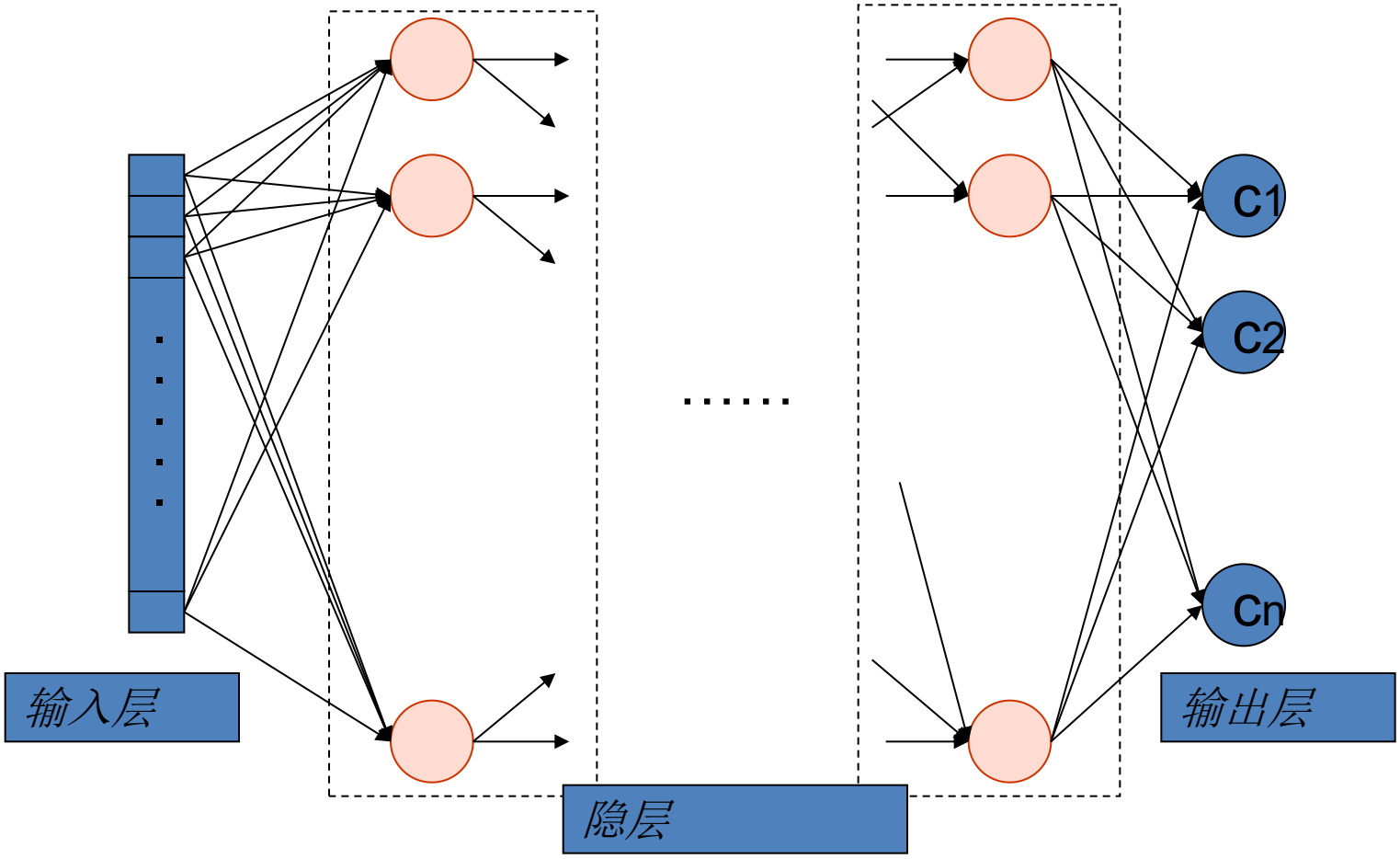
分类器设计

神经网络算法- Neural Networks

- 基本思想：
  - 神经网络是模仿人脑神经网络的结构和某些工作机制而建立的一种计算模型
  - 常用的神经计算模型有多层感知机、反传网络、自适应映射网络等
  - 神经网络通常由输入层、输出层和若干个隐层组成
    - 输入层的神经元个数等于样本的特征数
    - 输出层就是分类判决层，它的神经元个数等于样本类数

分类器设计

BP网络



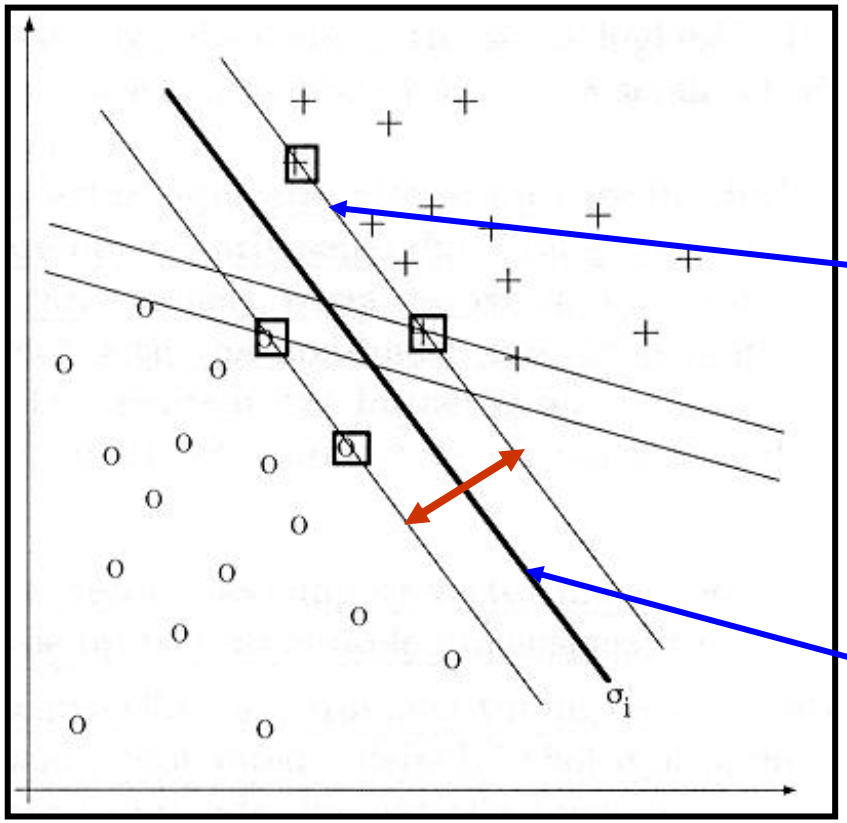
分类器设计

支持向量机算法-SVM

- 主要思想是：
  - 针对两类分类问题，在高维空间中寻找一个超平面作为两类的分割，以保证最小的分类错误率
  - 它通过非线性变换，将输入向量映射到一个高维空间H
  - 在H中构造最优分类超平面，从而达到最好的泛化能力

分类器设计

支持向量机算法-SVM



分类器设计

朴素贝叶斯算法– Naïve Bayes

- 基本思想：
  - 计算文本属于类别的概率。
  - 文本属于类别的概率等于文本中的每个词属于类别的概率的综合表达式。

分类器设计

朴素贝叶斯算法– Naïve Bayes

- 设各个类别的集合为 $\{c_1, c_2, \ldots c_n\}$
- 设d为实例的描述
- 确定d的类别

$$P(c_i \mid \mathrm{d}) = \frac{P(c_i)P(\mathrm{d} \mid c_i)}{P(\mathrm{d})}$$

- P(D) 可以根据下式确定

$$\sum_{i=1}^{n} P(c_i \mid \mathrm{d}) = \sum_{i=1}^{n} \frac{P(c_i)P(\mathrm{d} \mid c_i)}{P(\mathrm{d})} = 1$$

$$P(\mathrm{d}) = \sum_{i=1}^{n} P(c_i)P(\mathrm{d} \mid c_i)$$

分类器设计

朴素贝叶斯算法– Naïve Bayes

- 如果假定样例的特征是独立的，可以写为：

$$P(\mathrm{d} \mid c_i) = P(W_1 \wedge W_2 \wedge \cdots \wedge W_m \mid c_i) = \prod_{j=1}^{m} P(W_j \mid c_i)$$

- 因此，只需要知道每个特征和类别的$P(w_j \mid c_i)$

- 如果只计算单个特征的分布，大大地减少了计算量

分类器设计

朴素贝叶斯算法– Naïve Bayes

设V为文档集合D所有词词表

对每个类别 $c_i \in C$

D_i 是文档D中类别C_i的文档集合

$P(c_i) = |D_i| / |D|$

设 n_i 为D_i中词的总数

对每个词 $w_j \in V$

令 n_{ij} 为D_i中w_{ij}的数量

$$P(w_i | c_i) = (n_{ij} + 1) / (n_i + |V|)$$

分类器设计

朴素贝叶斯算法– Naïve Bayes

- 给定测试文档 X
- 设 n 为X中词的个数
- 返回的类别:

$$\underset{c_i \in C}{\operatorname{argmax}} P(c_i) \prod_{i=1}^{n} P(w_i \mid c_i)$$

- w_i是X中第i个位置的词

目 录

分类器评价

两类分类评价

- 二值分类列联表 Contingency Table

	真正属于该类的文档数	真正不属于该类的文档数
判断为属于该类的文档数	a	b
判断为不属于该类的文档数	c	d

分类器评价

两类分类评价

- 查全率 （Recall，简记为r）

$$r = a / (a + c)$$

- 查准率 (Precision，简记为p)

$$p = a / (a + b)$$

分类器评价

两类分类评价

宏观平均是先对每一个类统计r，p值，然后对所有的类求p的平均值，即

$$\bar{r} = \left(\sum_{\infty C} r_c\right) \Big/ |C| \qquad \bar{p} = \left(\sum_{\infty C} P_c\right) \Big/ |C|$$

微观平均是先建立一个全局列联表，然后根据这个全局列联表进行计算，即

$$\bar{r} = \frac{\sum_{\infty C} a}{\sum_{\infty C} a + \sum_{\infty C} c} \qquad \bar{p} = \frac{\sum_{\infty C} a}{\sum_{\infty C} a + \sum_{\infty C} b}$$

分类器评价

两类分类评价

- 平衡点(Break-Even Point)

  对于分类系统来说，r和p值是互相影响的，一种做法是选取r和p相等时的值来表征系统性能，这个值叫做平衡点(Break-Even Point，简称BEP)值

- F值(F-measure)

  另一种常用的将查全率和查准率结合起来的性能评价方法，其计算公式为

$$F_\beta = \frac{\left(\beta^2 + 1\right) * p * r}{\beta^2 * p + r}$$

分类器评价

多类分类评价

- P = 找到的该文档所属的正确类别数目 / 判断为该文档所属类的类别数目
- R = 找到的该文档所属的正确类别数目 / 该文档所属的所有类别数目
- 整个分类器的评估应该是对所有测试文档的这两个指标的统计平均
- 通常使用的统计平均为11点插值平均查准率 (Interpolated 11-point Average Precision)

目 录

有意义串对分类的改进

三点原因

因为有意义串可以发现跟类别相关的新用语、人名等特征词汇。

与单个词相比较，有意义串包含更多信息量，能够更准确地表达语义概念，减少歧义产生的情况

有意义串除了能作为特征改进分类效果以外，其属性值也可以用来标示特征的权重

有意义串对分类的改进

分类可分为训练阶段和分类阶段

在训练阶段，一组已被标记好的文档集合被用来训练分类器；在分类阶段，分类系统用已经训练好的分类器将给定的文档标记为一个预先设定好的类别。

首先利用有意义串提取模块，从训练文档集合的所有文件中提取有意义串并生成有意义串集合**MSSet**.接着将**MSSet**和词典**D**中的词条输入到特征提取模块中，利用特征提取方法从**MSSet**和**D**中提取有效的特征向量 $\overline{T} = \{t_1, t_2, \ldots, t_m\}$

利用有意义串的属性值对特征向量中的每一个特征赋权值并将每个文档表示成**m**维的向量 $\overline{d}_j = \{w_{j,1}, w_{j,2}, \ldots, w_{j,m}\}$

最后用生成的文档向量集合来训练分类器。生成的分类器即可对新的文档进行分类

有意义串对分类的改进

有意义串属性对特征权值的改进是核心

字符串S的邻接类别*AV*值：S左边邻接的词的类别数目和右边邻接的词的类别数目的最小值，能够反映一个字符串的语用丰富程度，语用环境越多样，说明该字符串的重要程度越高。

词长度：字符串包含词典词的数目。词长度越大的字符串语义越明确，具有更高的分类能力，因此要加大该字符串的权重

$$SIG（S）=AV（S）*（\log Len(S)+1）$$

假设S=W1W2W3…Wn，Wi为词典词，AV（S）表示S的邻接类别值，Len(S)表示S的词长度。SIG（S）表示一个字符串的重要性字符串S可以选择只用SIG（S）作为特征权重，也可以用SIG（S）与传统的TF—IDF想结合来表示特征的权重。

有意义串对分类的改进

电脑语料加入有意义串的分类准确率

	Dict	Dict+MS
Naïve Bayes	68.9433	71.8315(+0.042)
LibSVM	73.6323	74.21(+0.008)
C4.5	63.8464	64.3221(+0.007)

科技语料加入有意义串的分类准确率

	Dict	Dict+MS
Naïve Bayes	79.2916	79.7033(+0.005)
LibSVM	73.4332	75.6131(+0.030)
C4.5	72.2071	71.7984(-0.007)

有意义串对分类的改进

语料规模对改进效果的影响（Naive Bayes）

	DICT	DICT+MS
500	78.85	79（+0.02）
1000	80.675	81.275（+0.007）
1500	80.9667	81.5667(+0.007)
2000	80.85	81.35(+0.006)

有意义串对分类的改进

有意义串属性对特征权重的改进

	TF*IDF	SIG	SIG*TF*IDF
电脑语料	70.2345	71.4917（+0.018）	70.6082
科技语料	80.6731	83.9423（+0.041）	79.1346