自动分类

构建分类器的方法

1） Classifier：Rocchio method

每一类确定一个中心点（代表元），计算带分类的文档与各类代表元间的距离，并作为判定是否属于该类的判据。

构造方法：给定一个类，훈련데이터셋에서 이 类에 속하는 문서가 상응하는 벡터량의 모든 分量을 양수로 표시하고, 이 类에 속하지 않은 문서의 상응하는 벡터량의 모든 分量을 음수로 표현한다. 그리고 모든 向量을 더한 和向量이 바로 类의 **原型向量**이다.

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|},$$

이 두 向量의 相似度를 两个向量夹角的余弦이라고 한다.

2） KNN classifier（最近邻算法）

基于统计的模式识别方法。

核心思想：한편에 문서를 시스템이 훈련데이터중에 k개의 가장 가까운 이웃을 찾고, k개의 이웃의 类别를 문서의 候选类别로 사용한다. 이 문서와 k개의 이웃간의 相似度는 按类别分别求和하고, 가장 먼저 얻은 截尾阈值을 빼고 나면 이 문서의 类别测度가 된다.

可以把邻居文档和测试文档的相似度作为邻居文档所在类的类权重。如果k个邻居中的部分文档属于同一个类，则该分类中的每个邻居的类权重之和作为该类别和测试文档的相似度。通过对候选类评分的排序，然后给出一个阈值，就可以判断测试文档的类别。

训练时间复杂度为0， 本身不要使用训练集训练。

데이터에 문서 몇 개 있는 만큼 복잡도 가 정해짐, n개 문서면 复杂度为O(n)。

分类结果评估的主要指标： 准确率，召回率，宏平均，微平均，F测度值

特征选择（시험에 나왔음）：是一个**维数归约**的过程，即**删除不相关的特征**（索引语，关键词）从而减少特征空间的维数，通常使用特征子集的选择方法

主要方法：文档频率，信息增益，互信息，开方检验

1) 文档频率（DF）：在训练集中包含某个特征项t的文档数。DF가 적은 特征项은 분류결과에 대한 영향이 적은 편. 반대로 噪音数据일 수도 있다. 그렇기 때문에 먼저 DF가 큰 特征项부터 선택을하고, 적은 것은 삭제한다. 즉 特征项을 DF值대로 배열한다. 이렇게하면 维数를 크게 압축할 수 있는 것이다.

特征抽取：通过属性间的关系，如组合不同的属性得到新的属性，这样就改变了原来的特征空间。主要方法：Latent Semantic Indexing（LSI），Latent Dirichlet Allocation(LDA)，Singular Value Decomposition(SVD)

여기까지는 文档을 자동분류하는 알고리즘과 방법들이었고, 이제는 문서가 아닌 데이를 대상으로 하는 자동분류 알고리즘이다.... 이름하여 决策树.....

决策树（시험에 나왔음）

从给定的训练数据中学得一个（树结构）的模型用来对新示例进行分类。

包含一个根节点，若干个内部节点和若干个叶节点。 叶节点对应于决策结果，其他每个节点则对应于一个属性测试，每个节点包含的样本集合根据属性测试的结果被划分到子结点中；个节点包含样本全集。从根节点到每个叶节点的路径对应了一个判定测试序列。

决策树学习的目的是为了产生一颗泛化能力强的决策树

信息熵：用于量化一个随机变量的不确定性程度(不纯度)。

$$H(p1,p2,\ldots pn)= -\sum_{i=1}^{n} p_i \log p_i \quad \text{或记为, } H(P),\ H(X)$$

熵：对二元变量，X去p，1-p

联合熵：描述二元随机变量的不确定性程度

$$H(X, Y)=-\sum\sum p(a_i, b_j)\log p(a_i, b_j)$$

条件熵：已知变量X的条件下，Y的不确定性程度 H(Y丨X)

相对熵（相对信息，KL散度）：用于衡量两个分布之间的距离。

互信息：假设有两个不同的变量的概率分布p(ai)，p(bj)。互信息是在获得一个变量的知识时，对另一个变量的不确定性减少的量。

最优属性划分: **信息增益** (이 파트 중요함)

选择具有最高信息增益的属性

그럼 信息增益가 무엇이냐: 属性A的值而导致的上的期望压缩

- 设 p_i为D中属于类C_i的概率，用$|C_{i,D}|/|D|$作为估计值
- 样本集D本身的期望信息（信息熵），即：不考虑任何属性:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- 使用属性A，将D划分为v个子集后的信息熵

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- 信息增益：知道属性A的值而导致的熵的期望压缩。

$$Gain(A) = Info(D) - Info_A(D)$$

信息增益 계산 실전:

属性选择：信息增益

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

信息管理系 北京大学

ID3이라는 알고리즘이 **信息增益를 度量标准으로** 사용하고 있음(decision tree만들 때 사용하는 알고리즘) → 选择具有最大信息增益的属性。分裂之。

대략적인 步骤:

1) 특징집합과 데이터집합을 초기화 한다
2) 정보 엔트로피와 모든 특징의 조건엔트로피를 계산한다, 그 중에서 정보 이득이 가장 큰 특징을 골라 현재 决策节点으로 만든다
3) 특징집합과 데이터집합을 업데이트 한다.(이전에 사용했던 특징을 삭제하기, 그리고 다시 특징값에 따라서 데이터 나누기)
4) 2,3번을 반복한다, 특징이 하나만 남으면 分支叶节点으로 만든다

이렇게 정보 이득이 가장 큰 속성을 골라서 나누는 것은, 선택사항이 많은 속성에게 유리하다. 이렇게 하면 정보 이득이 가장 크기 때문이다. 하지만 아직 모르는 데이터에 대해서는 믿을 만한 예측을 내놓기 힘들다.

개선 방법:

限制测试条件输出，只能是二元划分； 修改评估划分的标准，把属性测试条件的输出数量考虑进去。

그리고 나온 개념 **增益率** (시험에 나왔음)

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \quad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

보통 取值越多的属性， 该值也越大。

C4.5算法选择了增益率作为主要的度量指标。바로 이걸 쓰는게 아니라, 使用了一个启发式先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的。

왜냐하면, 但增益率准则**对取值数目较少的属性有所偏好(信息增益랑 반대되는 성질)**。

基尼指数：数据集数据集不纯度를 나타내는 지표

$$\begin{aligned}\mathrm{Gini}(D) &= \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} \\ &= 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 \ .\end{aligned}$$

属性A的基尼指数：

$$\mathrm{Gini\_index}(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Gini}(D^v) \ .$$

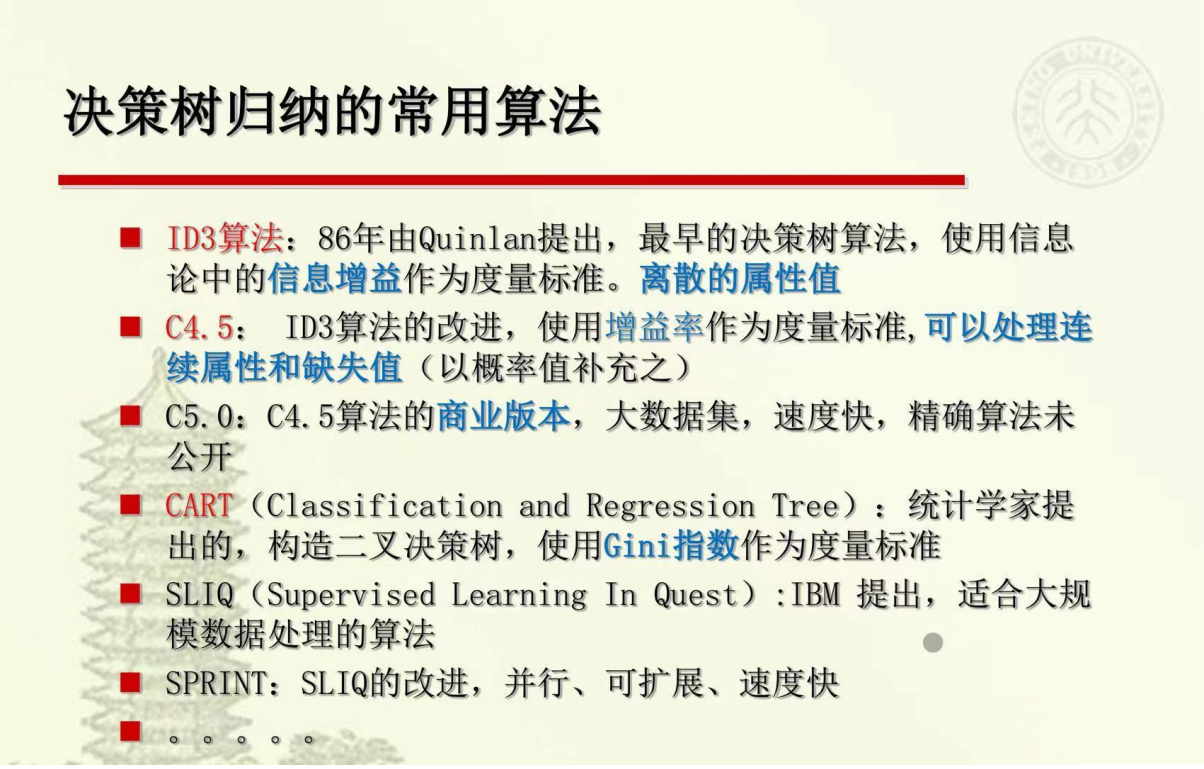
CART算法（分类与回归树）：이거는 decision tree알고리즘인데, 보통 머신러닝의 분류나 회귀에 광범위하게 사용됨.

主要特点：二叉树结构，每个节点有两个子节点； 内部节点根据特征进行划分，叶子节点对应的是最终的预测值

工作原理：

-分类任务：지니지수로 데이터 불순도를 측정한다음에, 지니 지수를 가장 줄일 수 있는 특징으로 나눈다, 조건에 만족할때 까지（如：最大深度）

-回归任务：MSE（方差）로 分裂质量을 측정한다음, 오차를 최소화 할 수 있는 特征과 阈值로 나눈다.



ID3算法 – 信息增益

C4.5 – 增益率

CART – Gini指数

이렇게 속성이나 특징을 골라서 划分하는 것은 decision tree 성능에 도움이 되지만, 泛化性能에는 제약이 있다.

그렇기에 **剪枝을** 통해서 **泛化能力에** 도움을 주고자 한다!

최대한 정확하게 분류하려고 样本을 분류하려고 보면, 分支이 너무 많게 될 수 있다(**过拟合**), 그렇기에 주동적으로 分支을 빼내면서 过拟合하는 위험을 줄일 수 있다

预剪枝：提前终止某些分支的生长。不仅降低过拟合的风险，还显著较少了决策树的训练时间开销和测试时间开销，但**欠拟合风险增加**

后剪枝：生成一颗完全树，再“回头”剪枝。降低过拟合的风险，但是训练时间开销增加（测试时间은 여전히 내려감），欠拟合风险基本不变

泛化能力：后>预

连续值나올땐? → 连续属性离散化

-주로 二分法사용해서 실현

缺失值나올땐(시험에 나옴) →样本赋权，权重划分

그냥 缺失值없는 데이터만 사용하는 거는 낭비다.

결측값이 있는 데이터를 사용할때 두가지 문제점이 발생하는데,

-如何选择属性?

-样本在该属性上有缺失值，如何进行划分?

样本赋权，权重划分이게 무슨뜻일까, 데이터 셋안에 모든 데이터들의 权重을 1로 주고, 한 속성중(결측값 제외하고)에서 信息熵 을 구한다, 그리고 그 안의 取值안의 信息熵을 구하고 나서, 처음에 선택한 속성의 信息增益를 구한다. 그렇게 모든 속성의 信息增益를 구하고, 가장 높은 속성을 구해서 그 속성의 取值을 分支에 넣는다. 그리고 그 속성에 결측값 제외하고 n개가 있다고 하면, 取值이 몇 개 있는지에 따라서 n빵해서 权重을 준다(많을수록 많이 준다)

多变量决策树：每个非叶节点不仅考虑一个属性（如：斜决策树），不是寻找最优划分属性，而是建立一个线性分类器。