Modern Information Retrieval

现代信息检索

第13讲：决策树与面向文档的机器学习

文本分类

- 上一讲介绍了文本分类的基础算法
  - 朴素贝叶斯
    - 简单，计算代价低，高偏差，线性
  - KNN
    - 简单，测试阶段计算代价高，高方差，非线性
  - 向量空间分类：Rocchio
    - 简单的线性判别式分类器; 也许过于简单
- 本讲内容
  - 决策树
  - 一些经验性的评价与比较
  - 决策树集成（ensembles）
    - 以及基于树方法（GBRT）的排序
  - 文本分类中的一些问题

文本分类评价：经典Reuters-21578 数据集

- 最常用的数据集
- 21578 个文档
- 9603篇 训练，3299篇 测试文章 (ModApte/Lewis split)
- 118 个类别
  - 一篇文章可以隶属于多个类别
  - 学习118个类别的二元分类
- 平均文档长度：200个词条
- 平均隶属类别
  - 每篇文档平均隶属于1.24个类别
- 118个类别中，仅有大约10个较大

Common categories (#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369,119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

Reuters-21578 中的文档：一个例子

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

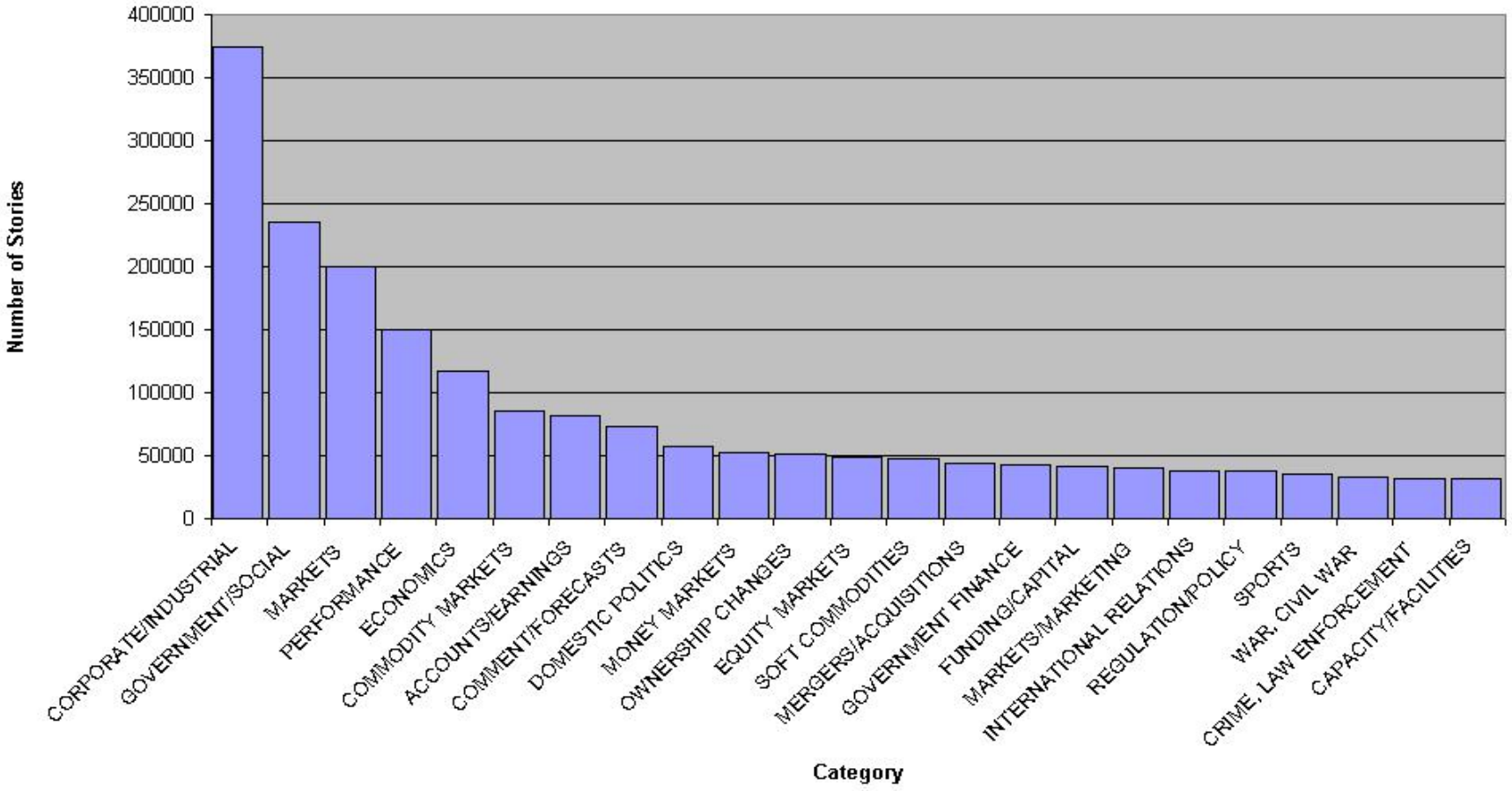
Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

&#3;</BODY></TEXT></REUTERS>

更新一些的 Reuters 数据: RCV1: 810,000 个文档

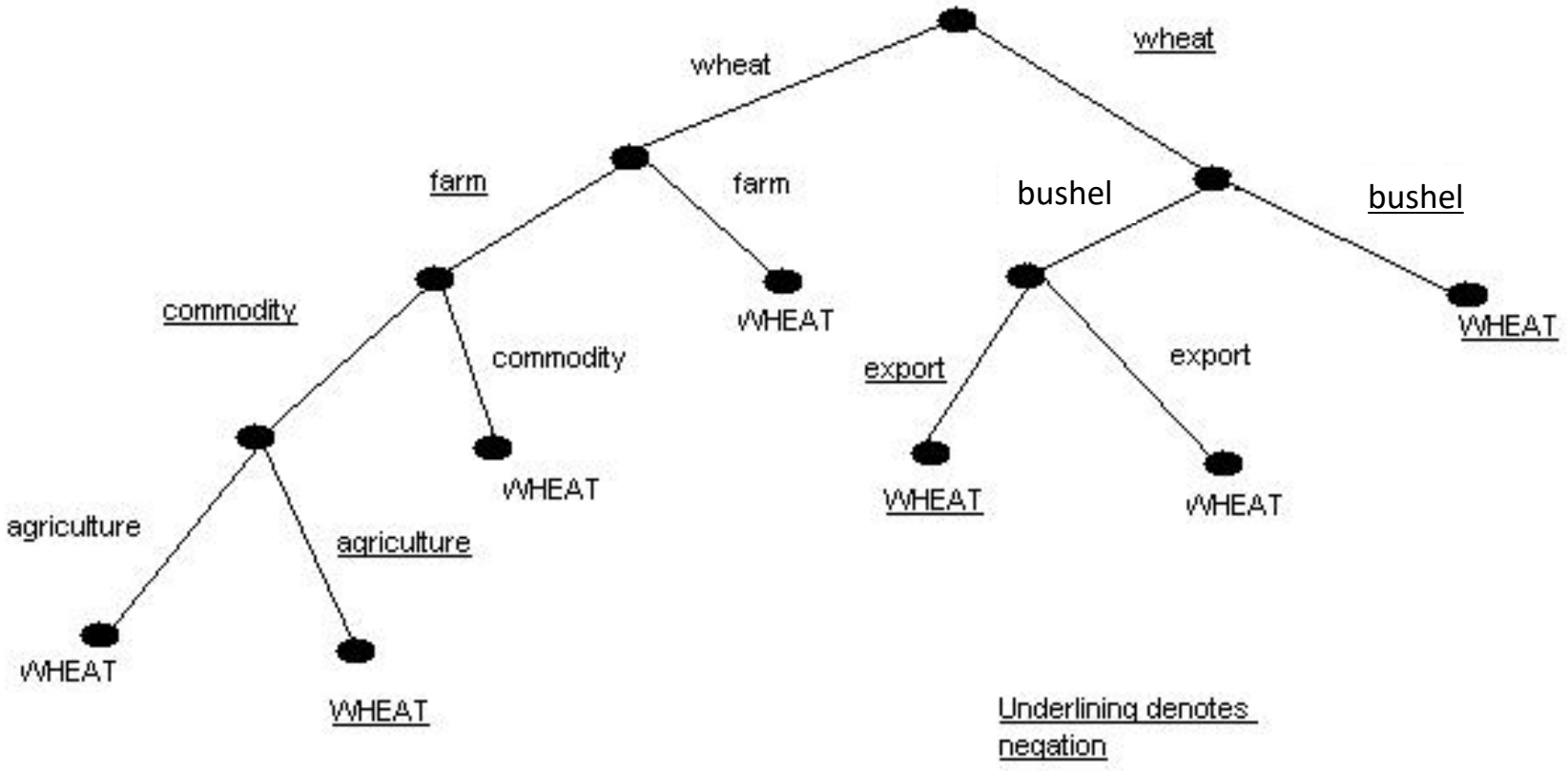
- Reuters RCV1最常见的主题



面向文本分类的决策树

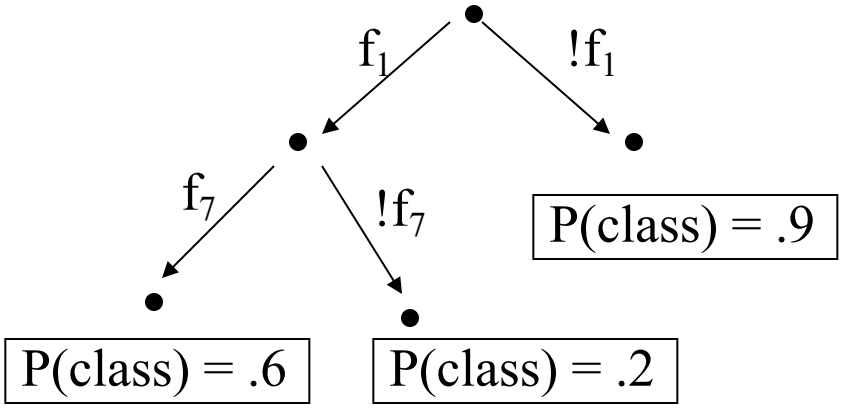
- 树结构：内部节点由词项作为标记
- 分支标记：词项权重的“测试” (test)，或仅仅是 出现/不出现
- 叶节点标记：类别
- 分类器
  - 分类器通过“测试”后的降序树对文档进行分类
  - 然后将叶节点的标签分配给文档
- 大多数决策树都是二叉树（永远不会是一个不利因素/never disadvantageous；可能需要额外的内部节点）

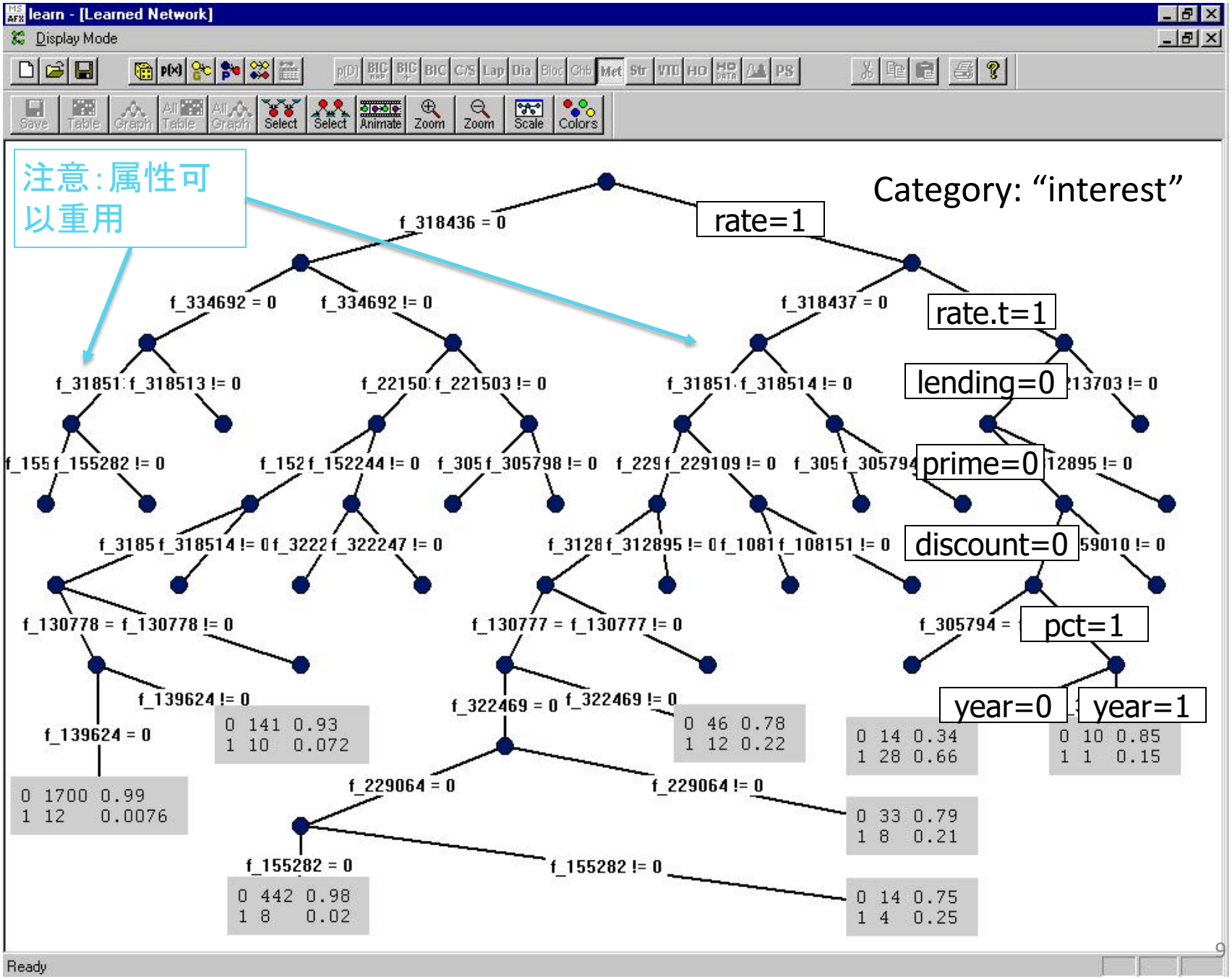
决策树：例子


wheat
wheat
farm
farm
bushel
bushel
commodity
commodity
export
export
WHEAT
WHEAT
WHEAT
WHEAT
WHEAT
agriculture
agriculture
WHEAT
WHEAT
Underlining denotes negation

决策树的学习

- 学习一个序列的特征测试，典型的做法是由上到下的贪心搜索
  - 每一步选择具有最高信息收益的未使用特征
- 叶节点标记：yes/no 类别标记，或 连续值（如图中例子）



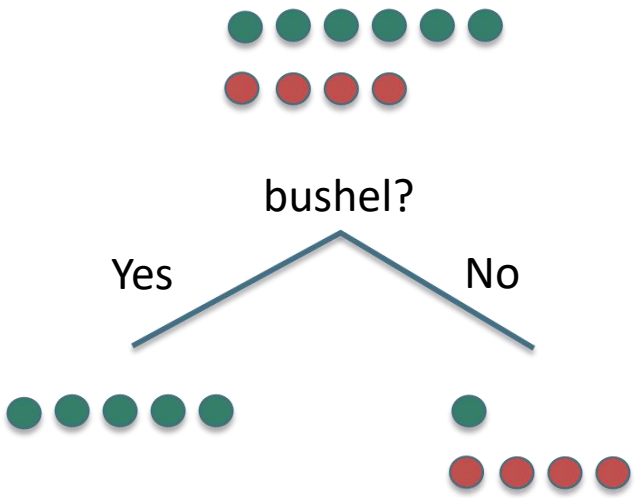

learn - [Learned Network]
Display Mode
Save
Table
Graph
All Table
All Graph
Select
Select
Animate
Zoom
Zoom
Scale
Colors
注意:属性可以重用
Category: "interest"
rate=1
f_318436 = 0
f_334692 = 0
f_334692 != 0
f_318437 = 0
rate.t=1
lending=0
prime=0
discount=0
pct=1
year=0
year=1
f_31851 f_318513 != 0
f_22150 f_221503 != 0
f_31851 f_318514 != 0
f_155 f_155282 != 0
f_152 f_152244 != 0
f_305 f_305798 != 0
f_229 f_229109 != 0
f_305 f_305794
f_3185 f_318514 != 0
f_3222 f_322247 != 0
f_3128 f_312895 != 0
f_1081 f_108151 != 0
f_130778 = f_130778 != 0
f_130777 = f_130777 != 0
f_305794 =
f_139624 != 0
f_139624 = 0
f_322469 = 0
f_322469 != 0
f_229064 = 0
f_229064 != 0
f_155282 = 0
f_155282 != 0
0 141 0.93
1 10 0.072
0 1700 0.99
1 12 0.0076
0 46 0.78
1 12 0.22
0 14 0.34
1 28 0.66
0 10 0.85
1 1 0.15
0 33 0.79
1 8 0.21
0 442 0.98
1 8 0.02
0 14 0.75
1 4 0.25
Ready

决策树的学习

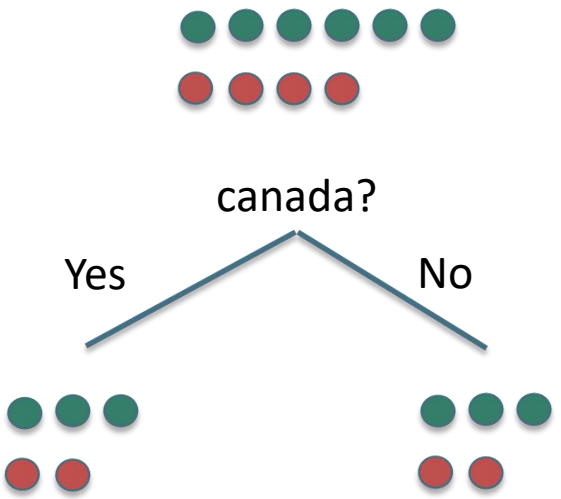
- 如果有k个特征，决策树的节点数量上限是2^k。
  - 一般来说2^k个节点数过大，会带来计算开支等方面的问题
- 我们希望能找到一个“有效率的(efficient)”的树
  - 小，但是分类效果好
- 我们可以通过**在每个节点上递归选择最佳拆分特征，以贪心的方式创建树**

属性/attributes 选择

- 基本思想：（理想情况下）一个好的特征能够把所有样本划分成“全部正样本”和“全部负样本”两个子集
  - 这是二元分类情况。相同的思想和方法也适用于n元情况，但是会偏重于更有价值的特征



“好”特征



“坏”特征: 正确率依然是 60%

利用信息论

- **信息熵（Entropy）**：考虑每个节点的类分解（class breakdown）。设：
  - p_i 为类别 i 中样本占所有样本的比例
  - p_i^f 为类别 i 中包含特征 f 的样本的比例
  - $p_i^{\neg f}$ 为类别 i 中不包含特征 f 的样本的比例

- 最后，设 pf 和 $p_i^{\neg f}$ 分别为 包含 和 不包含 特征f 的节点（在所有节点中的）比例

信息增益（Information Gain）

- 使用 f 分类之前，信息熵：

$$E = -\sum_{i=1}^{m} p_i \log p_i$$

- 使用f分类后，信息熵：

$$E_f = -p^f \sum_{i=1}^{m} p_i^f \log p_i^f - p^{\neg f} \sum_{i=1}^{m} p_i^{\neg f} \log p_i^{\neg f}$$

- 信息增益 = E - E_f (information= - entropy)

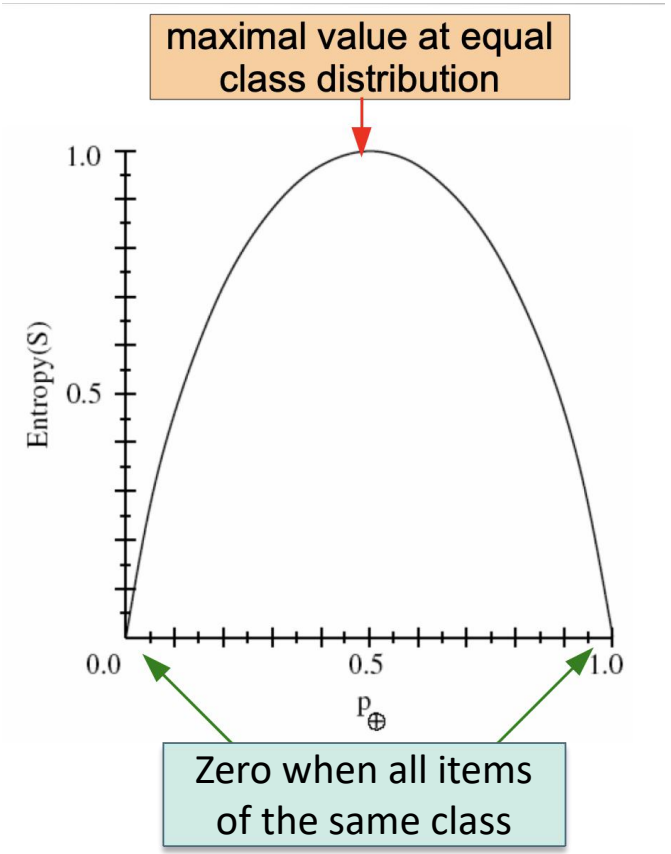
使用信息论

信息熵：类分布的不确定性的量

二类分类情况：

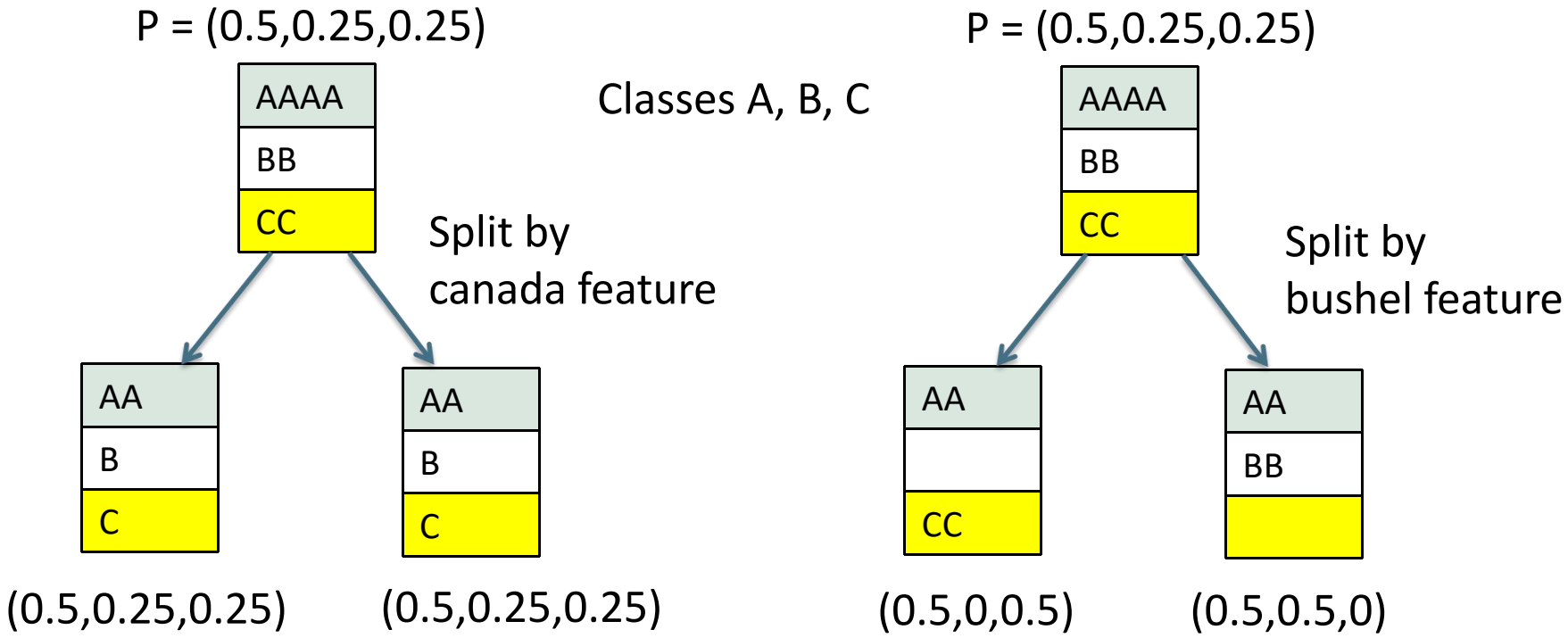
上：当两类均匀分布时（比如抛一枚均匀的硬币），熵最大 = 1

下：当所有样本属于同一个类，熵最小 = 0



一个例子

$$E = -\sum_{i=1}^{m} p_i \log p_i$$



$E = -\sum p_i \log p_i =$

0.5*1+0.25*2+0.25*2 = 1.5 bits

After:

E_f = (0.5+0.5)*1.5= 1.5 bits

No gain!

Before: E = 1.5 bits

After:

E_f = (0.5+0.5)*1 bits = 1 bits

Gain = $E - E_f$ = 0.5 bits

选择最佳特征

对每个节点，我们选择使**信息增益最大**的特征*f*

往往：树越深，节点的分类会越来越**“纯净”（“pure”）**
即一个节点的大多数样本属于同一个类别

如果一个节点具有类c的所有样本，将其设为叶节点并输出“ c”。 否则，继续树的构建（除非“纯度”已达到阈值）

如果叶节点的分类仍然具有混合分布（即并非所有样本都属于同一个类），在该节点输出**最常见类（ the most popular class ）**

数值特征（例如tf-idf）

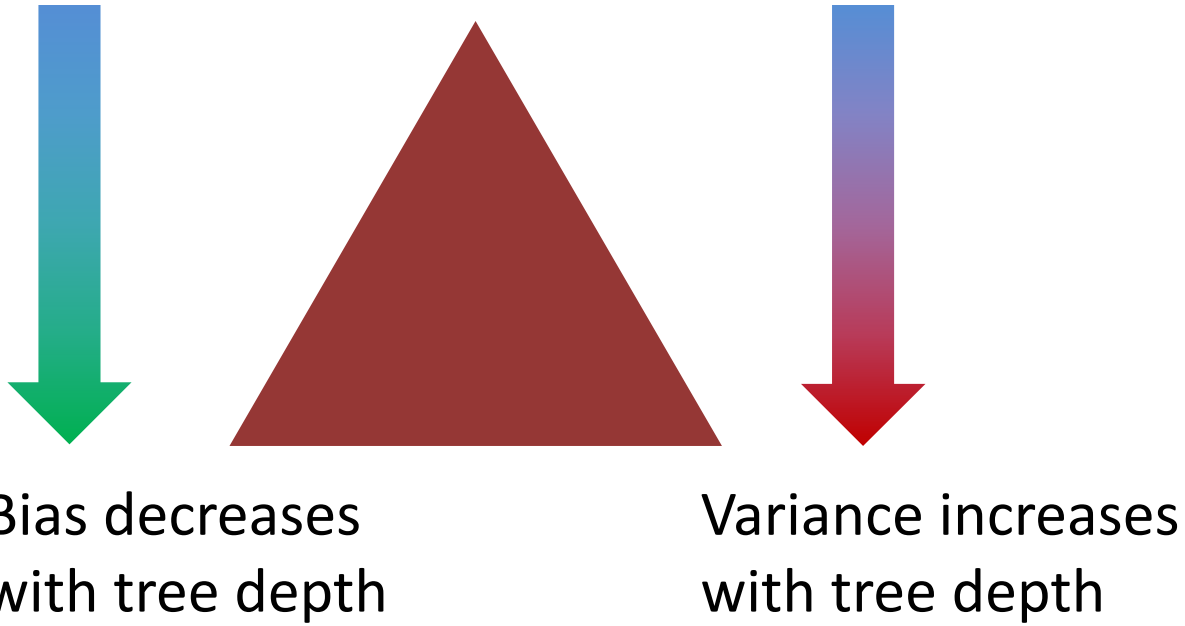
- 通常使用二元的切分（f < t），t怎样确定?
- 穷尽式（搜索）：评估观察值之间的每个分割点的**信息增益**。
  - 慢；通过优化计数方法可以稍微提高效率
- 分箱（Discretize into bins）
  - 将所有的数值切分到*k*个箱中
  - （连续的数值）特征被离散化
  - 分箱操作可以基于整个语料集的统计
    - 例如，可以对特征值使用k均值聚类

（树的构建）什么时候停止?

- 当一个节点的所有样本都属于同一个类别
- 当树的深度*d*达到一个固定阈值
- 如果没有属性，则可以在拆分中区分具有统计意义的类（例如，使用卡方检验或Fisher精确检验）
- **最常用/最佳方法：使用单独的验证数据**
  - 构建一个较大的树（可以给树的深度设定阈值）
  - 自下而上的修剪未能（显著）改善验证数据分类性能的节点

决策树模型

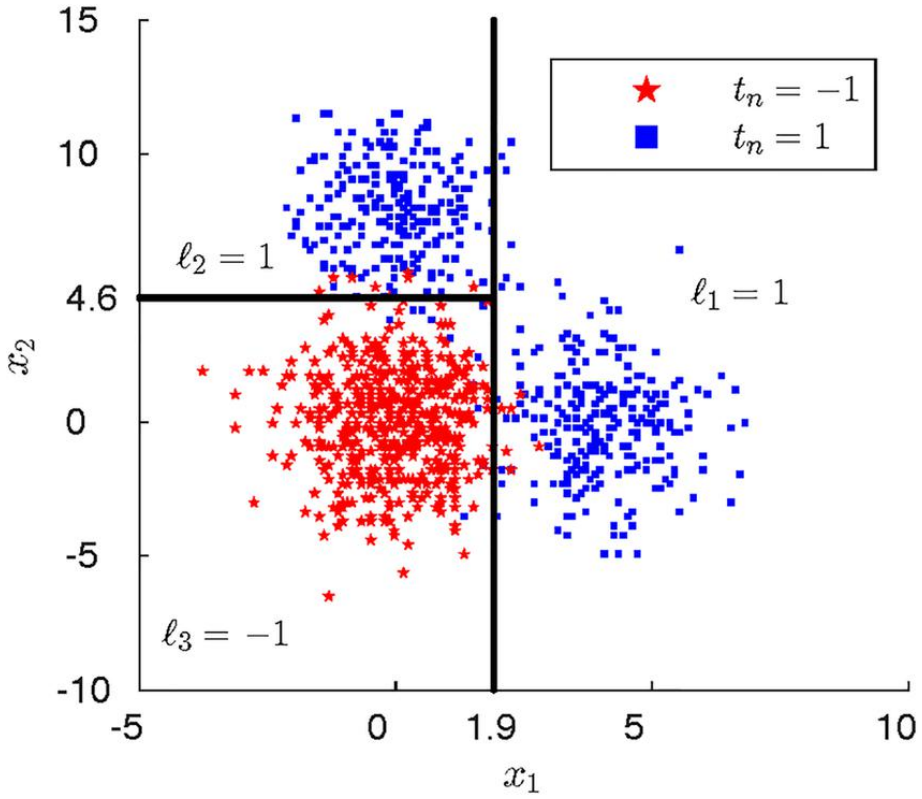
- 随着树深的增加，偏差会减少，方差通常会增加。 为什么？（提示：方差(variance)=记忆量(capacity)）



- 某些在树的深层拟合的因素(stuff)是训练数据中一些随机事实和关联(random facts and association in the training data)

基于决策树的矩形区域分类

（矩形）区域与特征轴对齐
模型无法学习任意线性决策边界
总体而言，（学习的）结果会是一个非线性分类器



面向文本的决策树学习

- 大多数人的直觉是，文本中有很多单词，提供了很多弱证据特征 (weak-evidence features)
  - 因此，使用少量特征测试(feature tests)可能不利于文本分类
  - 但实际上，该方法有时可能效果很好-例如在Reuters数据集上。主题(topics)可以包含标记词 (即重要指示性特征)。
- 决策树可解释性强-比Naive Bayes这样的方法容易理解得多
- 实际上，可以从决策树中提取规则

文本分类：每类的评价方法

- 召回率(Recall) - 类别*i*中的正确分类的文档比例 $\frac{c_{ii}}{\sum_j c_{ij}}$
- 正确率(Precision)：被分配到类别i的样本中正确(即实际属于i)的比例 $\frac{c_{ii}}{\sum_j c_{ji}}$
- 准确率(Accuracy)：（1 - error rate），正确分类的文档比例 $\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$

微平均与宏平均

- 如果我们有一个以上的类别，如何将多个类的性能指标合并为一个数值？
- 宏平均：计算每个类别的性能指标，然后取平均值
- 微平均：收集所有类别的决策（分类）结果，计算列联表，评价

微平均与宏平均的例子

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- 宏平均 precision: (0.5 + 0.9)/2 = 0.7
- 微平均 precision: 100/120 = .83

- 微平均指标数值主要由常见类的指标数值决定

Dumais et al. 1998: Reuters – Break-even F1 (平衡点F1)

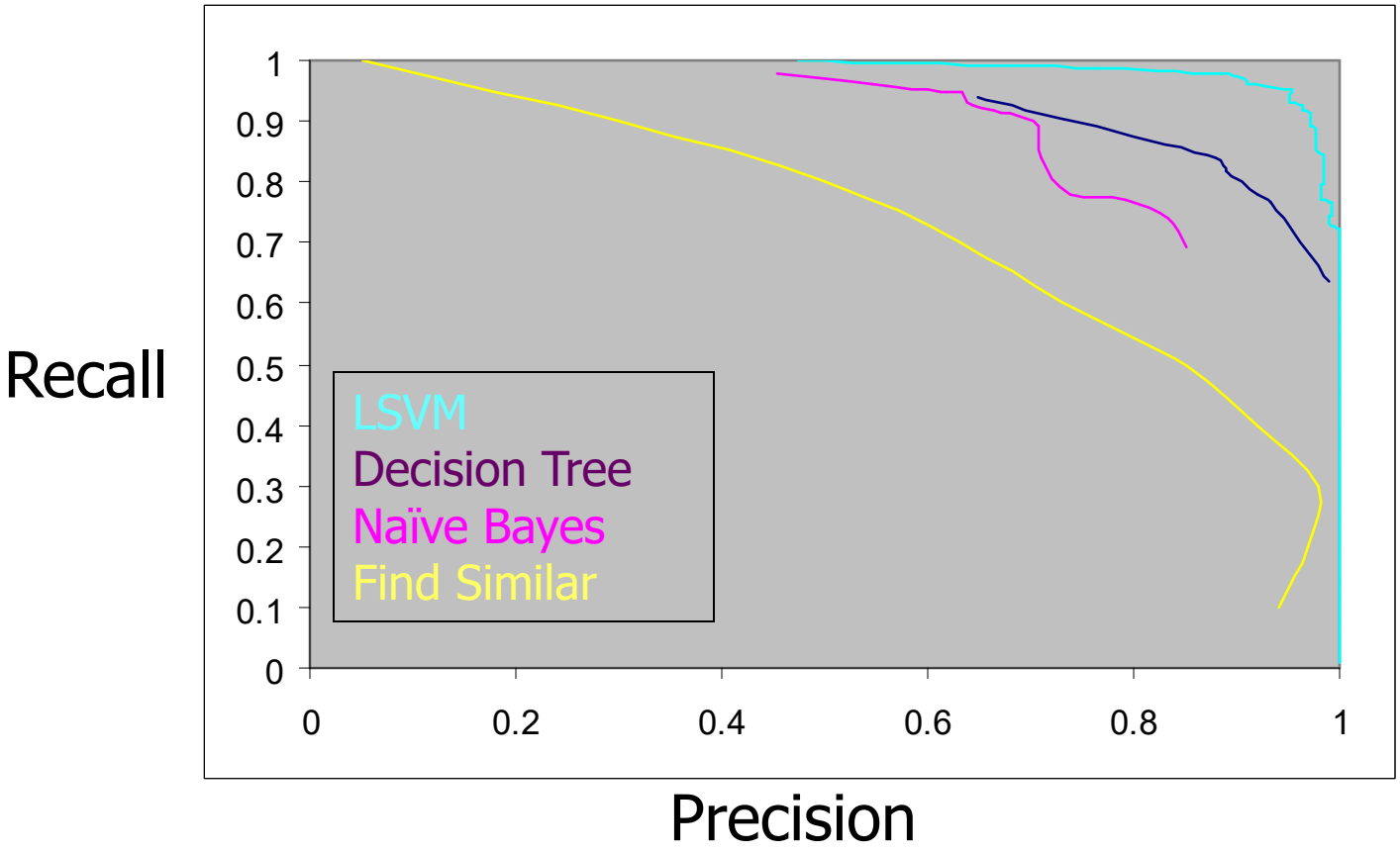
	Findsim	NBayes	BayesNets	Trees	LinearSVM
earn	92.9%	95.9%	95.8%	97.8%	98.2%
acq	64.7%	87.8%	88.3%	89.7%	92.8%
money-fx	46.7%	56.6%	58.8%	66.2%	74.0%
grain	67.5%	78.8%	81.4%	85.0%	92.4%
crude	70.1%	79.5%	79.6%	85.0%	88.3%
trade	65.1%	63.9%	69.0%	72.5%	73.5%
interest	63.4%	64.9%	71.3%	67.1%	76.3%
ship	49.2%	85.4%	84.4%	74.2%	78.0%
wheat	68.9%	69.7%	82.7%	92.5%	89.7%
corn	48.2%	65.3%	76.4%	91.8%	91.1%
Micro					
Avg Top 10	64.6%	81.5%	85.0%	88.4%	91.4%
Avg All Cat	61.7%	75.2%	80.0%	na	86.4%

Recall: % labeled in category among those stories that are really in category

Precision: % really in category among those stories labeled in category

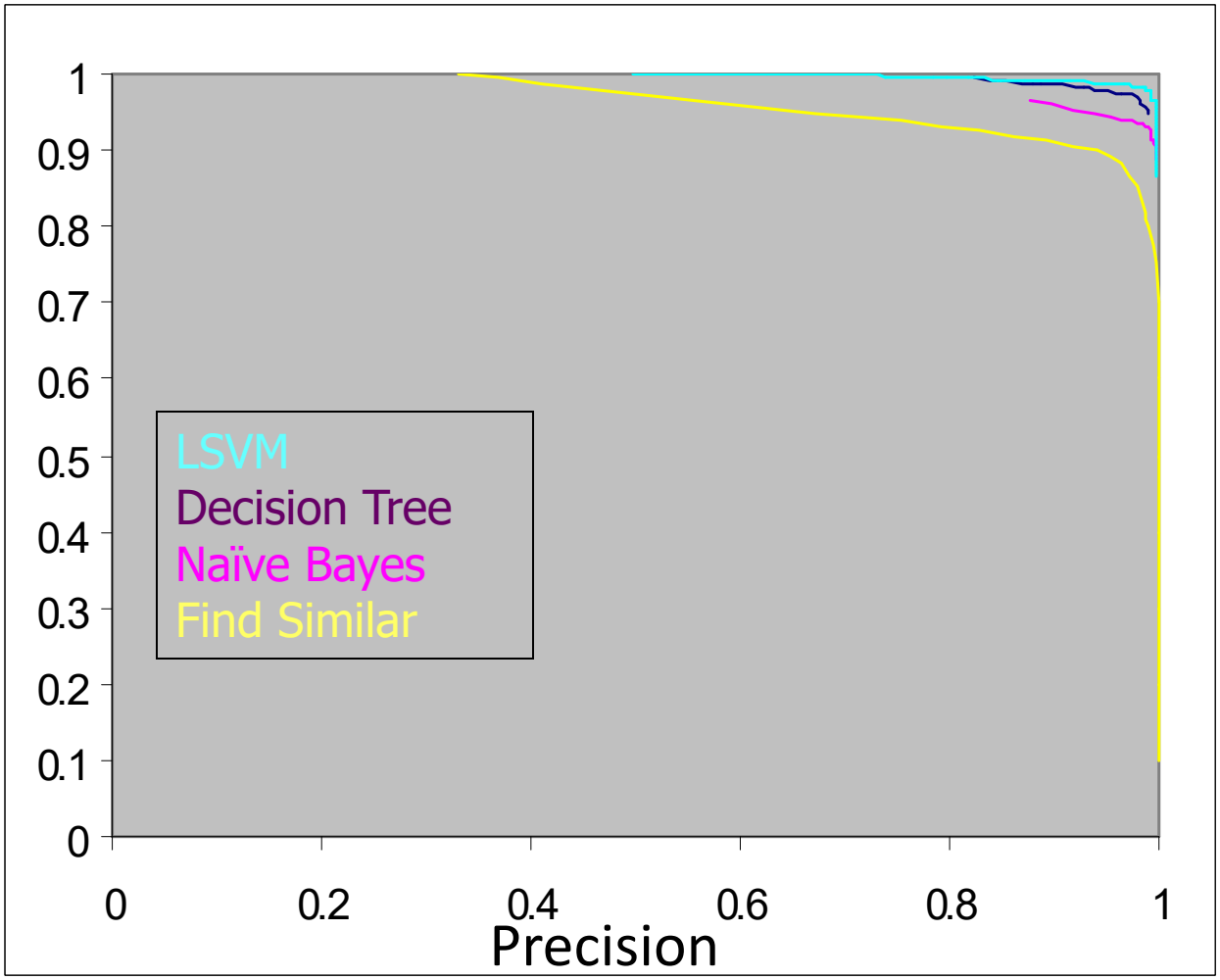
Break Even: 当 recall 等于 precision

Reuters ROC 曲线

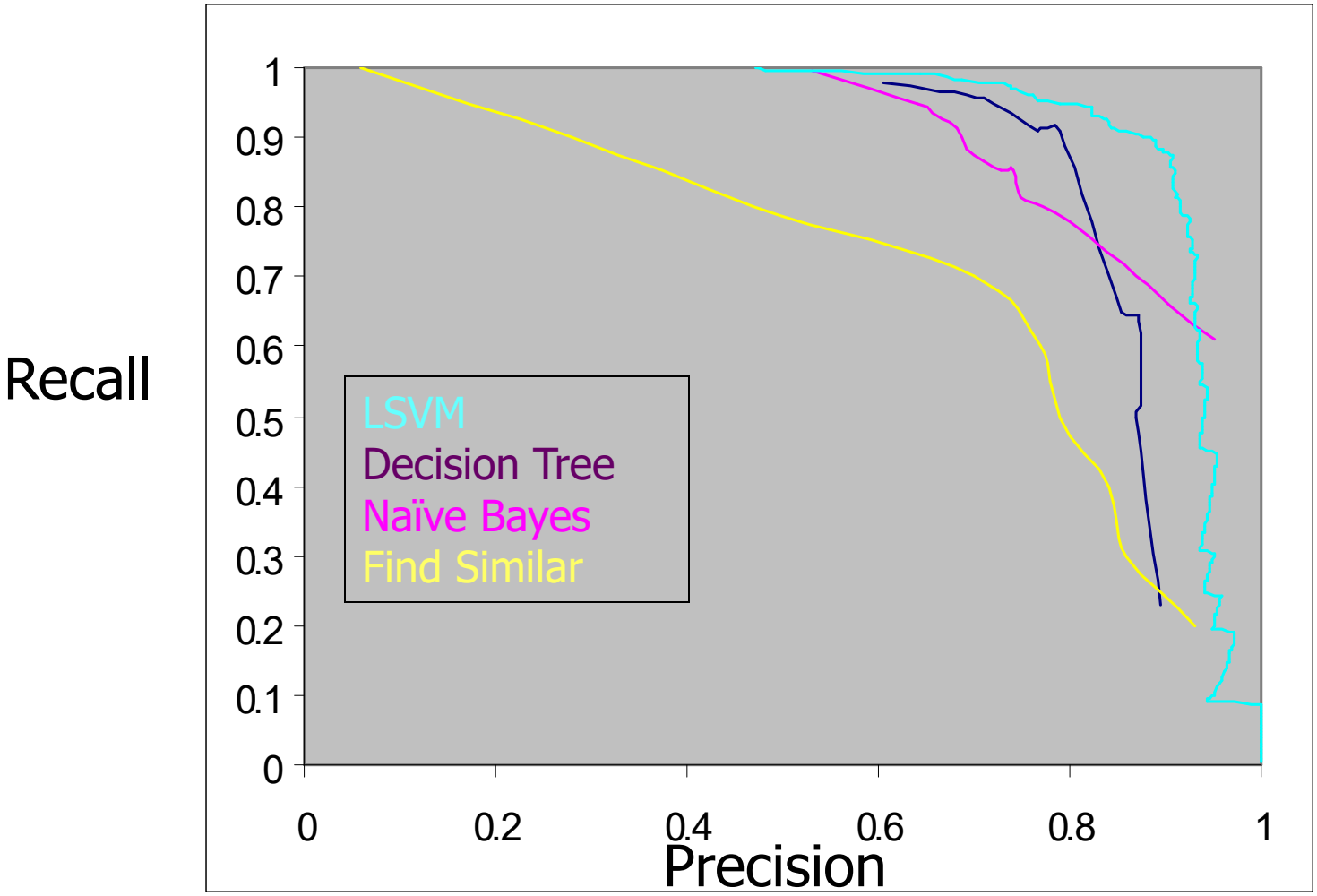


Roc 曲线：X轴-正确率，Y轴-召回率。线下面积（AUC：Area Under the Curve）越大，性能越好

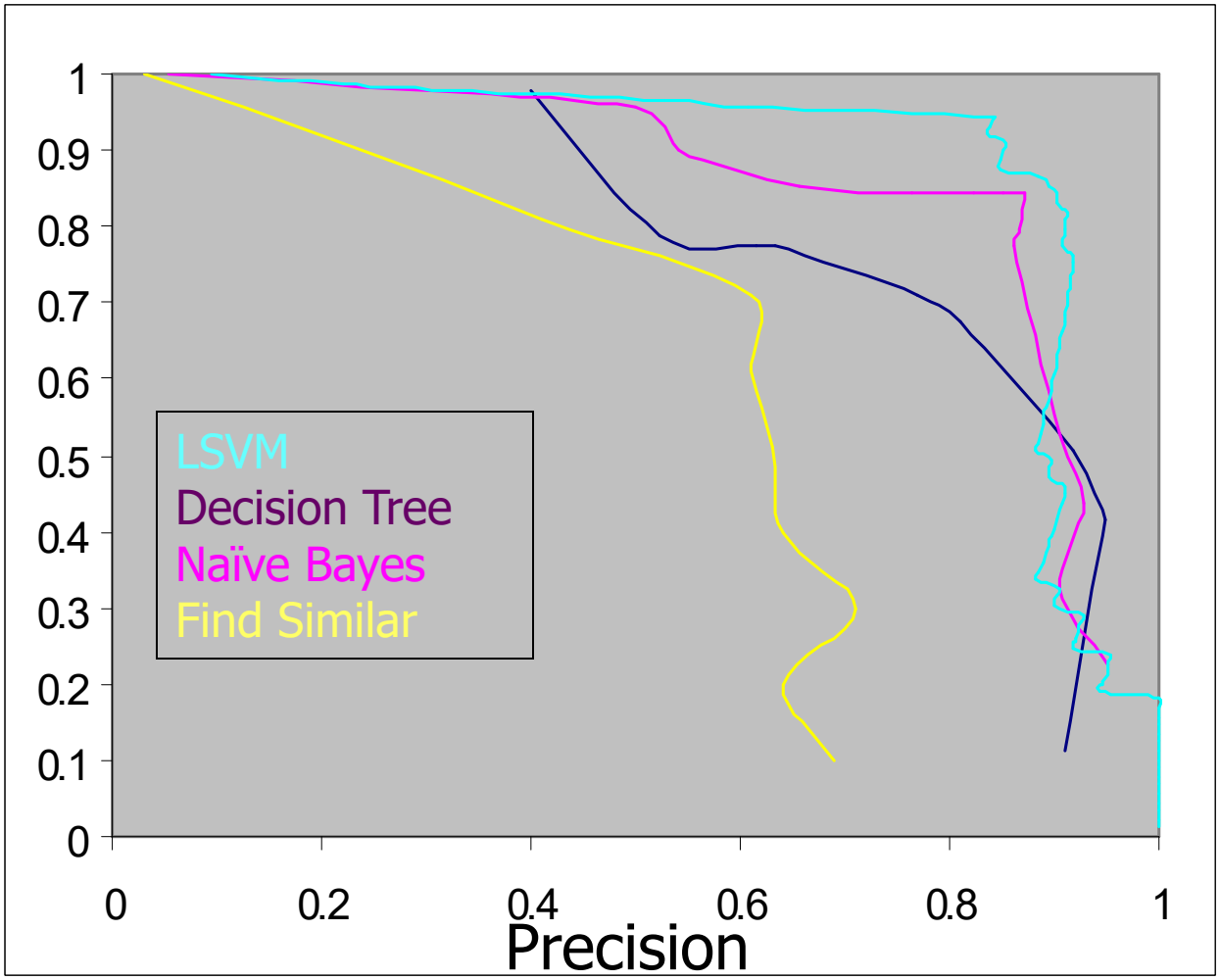
类别Earn的ROC曲线



类别Crude（原油）的ROC曲线



类别Ship的ROC曲线



(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1

判别式 (discriminative) 分类方法: Logistic Regression (逻辑回归) 与 Support vector machines (支持向量机)

- 给定词项为条件，直接预测类别
- 二元逻辑回归：

$$\log \frac{P(C \mid d)}{P(\overline{C} \mid d)} = \alpha + \sum_{w \in d} \beta_w \times w$$

- 通过调节参数β_w来优化条件似然
  - 如果是SVM，则调节间隔（margin）
- 基于统计学的分类决策方法

逻辑回归（LogR）/SVM 的性能

- LogR的早期结果令人失望，因为人们不了解*正则化*（平滑）LogR以应对**稀疏文本特征**的方法
- 如果能够以正确的方式进行正则化, LogR 的分类效果明显超过 朴素贝叶斯
- 在1997 - 2005年，SVMs 被视为最好的通用文本分类方法
- LogR 看上去和 SVMs 一样好 (Tong & Oles 2001)
- 但是现在面临其它方法的挑战:
  - 神经网络方法 (提升词项相似度估计的模型)
  - Ensemble (集成/组合) 方法，例如 **随机森林, boosting**

Ensemble 方法

类似 **众包机器学习方法**:

- 基于多个简单的*弱*学习器
- 将弱学习器的输出结果组合起来构建一个更好的学习器

Ensemble的不同类型:

- **Bagging:** 数据多次采样，在不同采样上并行训练学习器，然后通过投票（离散输出）或平均（连续输出）进行组合
- **Stacking:** 训练过程分为不同阶段，第一阶段模型（可以是多个模型）的输出作为特征输入第二阶段学习器（例如逻辑回归）
- **Boosting:** 在较早学习器的过滤/加权输出上训练后续学习器，以便修复较早学习器出错的内容

随机森林 （Random Forests）

从原始数据集重复采样（bootstrap采样），在采样数据上构建K个树，p=特征数量

- 获得K个大小为N的bootstrap采样，其中N是原始数据集大小
- 通过在每个节点的p个特征中随机选择m个，并选择最佳特征来扩展每个决策树。
  - m 的典型取值：sqrt(p)
- **预测(Runtime)**：汇总树的预测（最受欢迎的投票）以产生最终分类

随机森林

原则：我们希望在不同的学习器(learner)之间进行投票，因此我们不希望这些模型过于相似。这两个标准确保了各个树的多样性：

- 数据 bagging：获得K个大小为N的bootstrap采样：
  - 每个树都在不同的数据上训练
- 特征 bagging: 通过在每个节点上从p个特征中随机选择m个构成特征集合，然后选择最佳特征进行拆分（split，即产生分支节点），来构建决策树。
  - 通常，不同树中的对应节点无法使用同一特征进行拆分（从而保证各个树的多样性）

随机森林

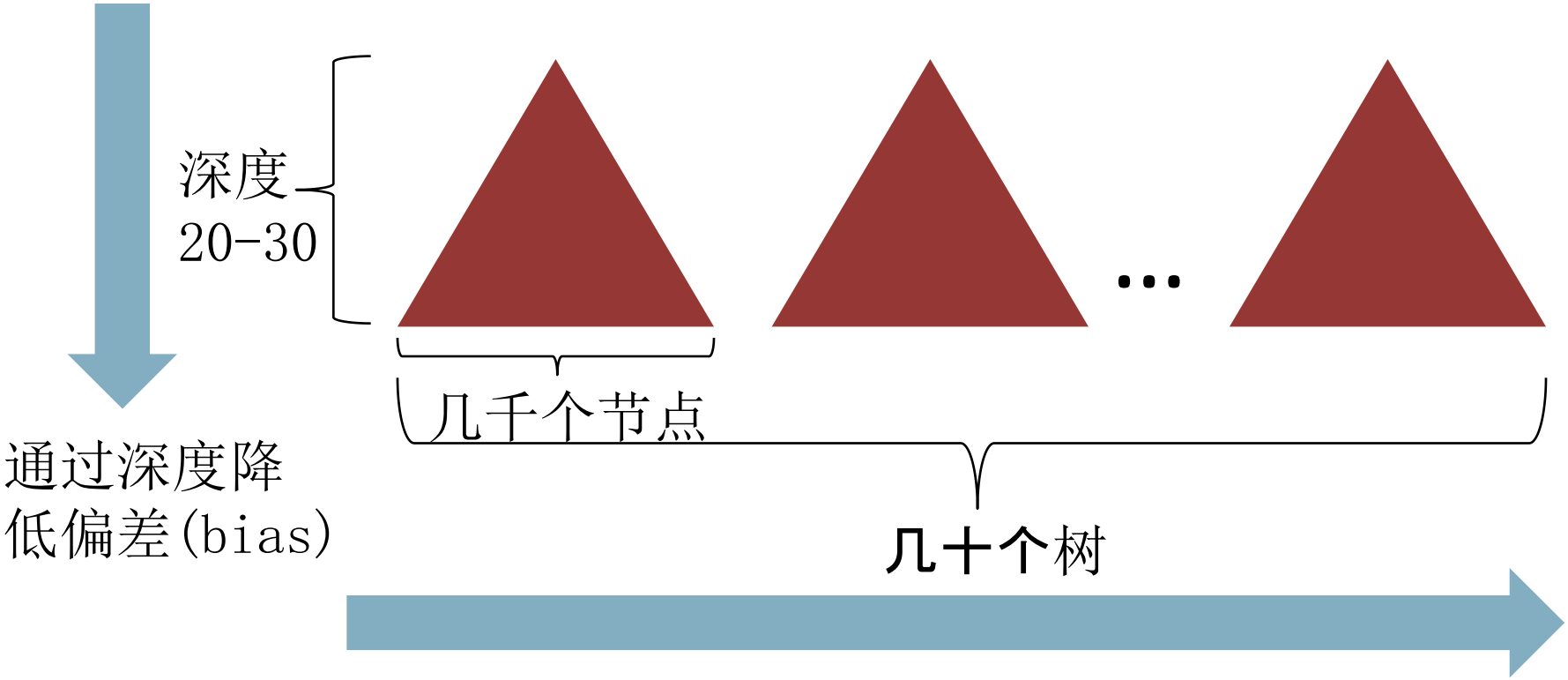
- **在实践中非常流行，**有段时间是密集数据（dense data）上最流行的分类器（<=几千个特征）
- **容易实现**（训练多个树）.
- **容易并行化**（但并不意味着效率高）。适合用于 MapReduce.
- **现在和一些新方法相比准确度并不高** - Gradient-boosted trees（特征少）与 深度神经网络(视觉，语音，语言，…)通常更好
- **需要多次遍历数据** - 至少是树的最大深度（虽然远小于 boosted trees ）
- **容易过拟合** - 需要权衡准确度（accuracy）与拟合度（fit）

Boosted Decision Trees (BDT, 增强决策树)

- 一个比随机森林（RF）提出更晚的算法
- 与独立训练树的RF不同，BDT的树通过增强(boosting)的方式依次(sequetially)训练树：
  - 每个树都在加权数据上训练，通过加权强调之前（训练）的树错误标记的样本
- 这两个方法都能够通过训练产生高质量模型
- 但是BDT通常都更适用于中等数量特征的数据集

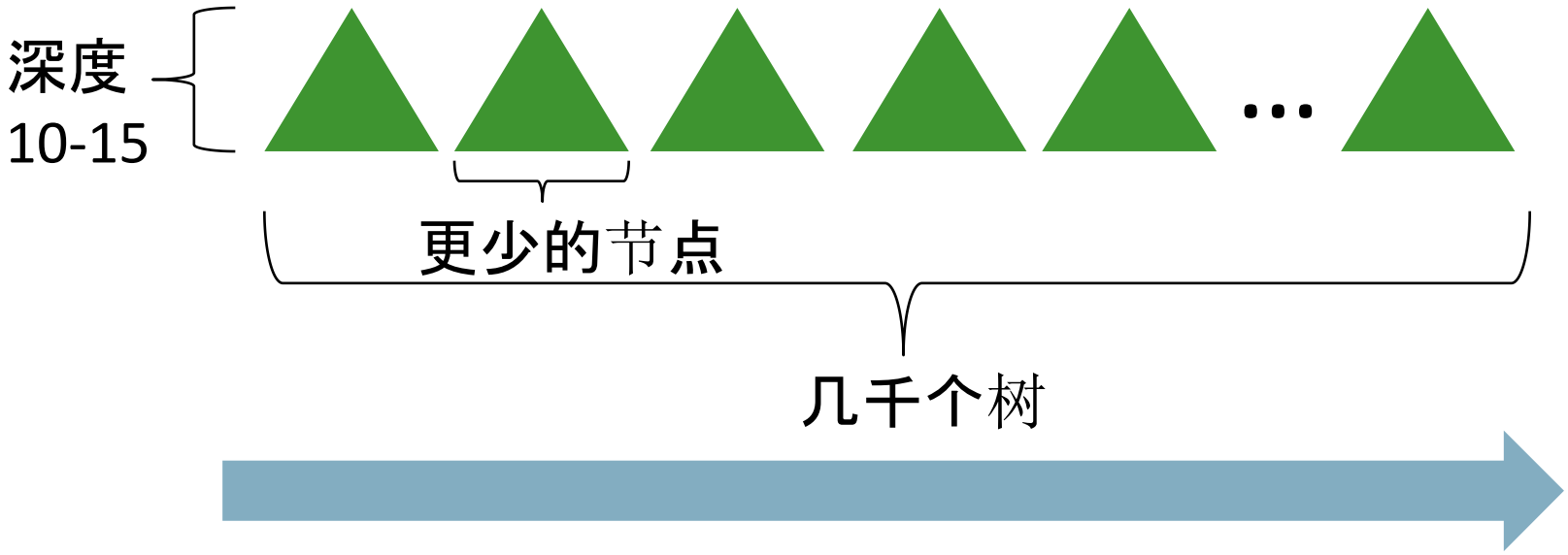
随机森林 vs 增强树

- 从“几何”角度看，两者有很大不同:
- 随机森林使用10多个既深(deep)又大(large) 的树:



随机森林 vs 增强树

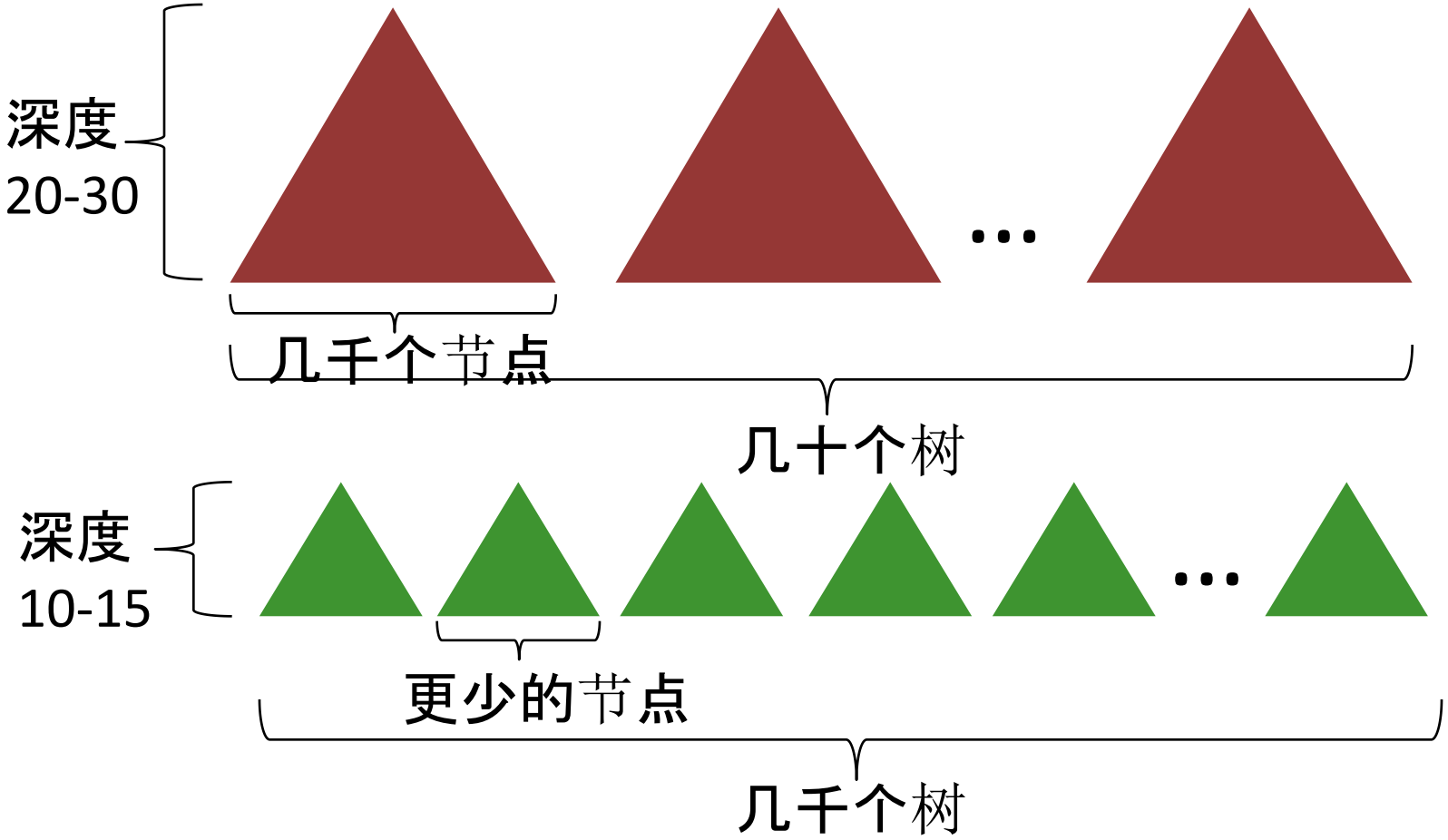
- 从“几何”角度看，两者有很大不同:
- 增强决策树使用几千个浅(shallow)且小(small)的树



通过boosting降低偏差(Bias) - 方差(Variance)已经较低

随机森林（RF） vs 增强树（BDT）

- RF的训练可以很方便的并行，因此很快
- 同样因为并行，预测也很快



真实世界（Real World）

P. Jackson and I. Moulinier. 2002. *Natural Language Processing for Online Applications*

- “根据内容进行自动分类文档的商业价值是毫无疑问的。对于企业内部网、政府部门和Internet发布者来说，这种功能有无数的潜在应用。”

- 理解数据是成功分类的关键之一，但这是大多数分类工具供应商都非常薄弱的一个领域。 市场上许多“一刀切”的工具尚未在多种内容类型上进行测试。

真实世界（Real World）

- 现实应用需要考虑的问题

- 有多少训练数据可用?
  - None / 完全没有
  - Very little / 很少
  - Quite a lot / 较多
  - A huge amount and its growing / 大量数据且与日俱增

人工编写规则

- 如果没有训练数据，是否有足够的人工？
- 永远不要忘记人工编写（分类）规则！(Never forget the hand-written rules solution!)
  - If (wheat or grain) and not (whole or bread) then Categorize as grain
- 在现实中，规则会很庞大
  - 可以通过 *tf* 或 *tf.idf* 权重反映特征的重要性
- 经过精心设计的人工规则可以达到很高的性能：
  - Construe: 94% recall, 84% precision over 675 categories (Hayes and Weinstein IAAI 1990)
- 人工工作量巨大
  - 大约每个类需要两天时间，加上维护时间

少量数据情况下的分类

- 如果只是进行有监督分类，则应该保持较高的偏差（Bias，即使用记忆量小的算法）
  - “理论性的结果”表明朴素贝叶斯在这种情况下表现良好 (Ng and Jordan 2002 NIPS)
- 一个有趣的理论上的答案是探索半监督训练方法：
  - Bootstrapping，未标记文档上的EM算法，…
- 实际一些的答案是尽快获取更多带标签的数据
  - 如何才能让人们愿意为你标记数据？

拥有一定量数据

- 分类应用的较理想情况
- 可以使用各种分类器
- logistic regression/SVMs/random forests等

如果有海量数据

- 理论上可以进行精确分类…
- 但这也意味着一些代价大的方法如 SVMs（训练时间）或 kNN（测试时间）会不太实用

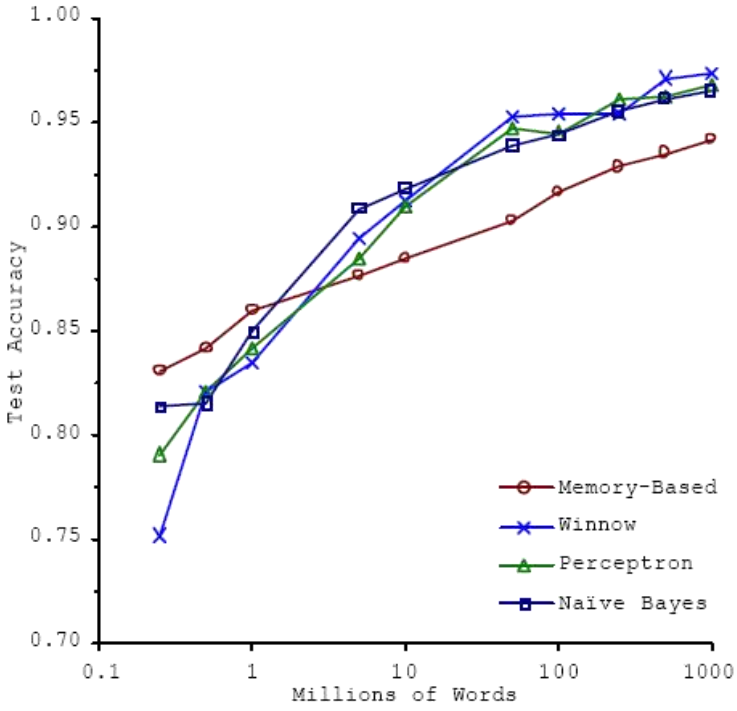
- 朴素贝叶斯相对而言效率较高
  - 或其他具有线性训练/测试复杂性的方法，例如（**正则化/regularized）逻辑回归**（尽管训练成本高得多）

准确率作为数据量的一个函数

- 有了足够的数据，分类器的选择可能不会有太大关系，最佳选择尚不清楚

数据：Brill和Banko关于上下文相关的拼写更正

- 但是必须不断加倍数据以提高性能这一事实可能会有些麻烦



有多少个类？

- 少量（类之间可以很好的分隔开）？
  - 易于处理
- 成千上万的在向量空间中接近的样本？
  - 设想：Yahoo！分类目录，国科图分类，法律应用
  - 大量相似样本的分类会很困难
    - 分类器的组合是一种有用的技术
      - Voting, bagging, or boosting 多个分类器
    - 学界已有大量层次化（Hierarchical）分类的方法
      - 能多有效尚不清楚，但多少有用（Tie-Yan Liu et al. 2005）
      - **但是肯定有助于可扩展性(scalability)**，即便不能提升准确率
    - 也许需要一种综合人工与自动分类的混合方法

学习曲线 （Learning Curves）

- 在现实应用中，标记数据往往难以获取
- 希望了解算法性能随训练样本数量变化的情况
- *学习曲线*中，Y轴是在测试数据上的分类准确率，X轴是训练样本数量

- 可以从交叉验证中得出多次测试的平均学习曲线

一些提高性能的技巧

- 利用领域相关的有用特征
  - 例如，作者的副标题或邮件标题
- 将一些属于同一类的特征统一处理
  - 例如，章节编号，化学公式
- 特征设计和非线性加权在现实系统的性能中非常重要
  - 实际系统中最易于提高性能的途径 (Easiest way to improve practical systems)

提权(Upweighting)

- 将文档中不同区域的文本区别对待，分别计算权重
- 将如果词项在某些文档区域(zone)中(例如摘要)出现，会赋予额外的权重
  - 对标题词提权有助于提升分类性能 (Cohen & Singer 1996)
    - 标题词权重x2
    - 与 BM25F 类似
  - 提升每个段落第一个句子的权重(Murata, 1999)
  - 提升包含标题词的句子权重 (Ko *et al*, 2002)

关于文档域的两个技术

1. 对于不同的区域，如标题，有一套完全独立的特征/参数
2. 在域之间使用相同的功能（汇总(pooling)/绑定(tying)其参数），但会通过提权体现不同域的贡献

- 通常第二种方法更成功：就稀疏数据而言，它不花任何代价，但可以大大提高性能

文本分类中的文本摘要（Text Summarization）技术

- 文本摘要：从文本中提取关键片段的过程，通常是通过反映位置和内容的句子特征来实现的
- 这类技术大部分可以用来计算文本分类中词项的权重
  - See: Kolcz, Prabakarmurthi, and Kalita, CIKM 2001: Summarization as feature selection for text categorization
  - 根据标题分类
  - 仅根据第一句分类
  - 根据包含关键字最多的句子分类
  - 根据第一句和最后一句分类
  - 等等。。。

词干还原/小写化/等预处理是否有帮助?

- 没有确定的答案，通常需要在具体数据上通过经验性的评价来确定
- 但是需要注意的是，例如词干还原这样的技术在文本分类和IR任务中的角色不同：
  - 在IR中，通常需要归并单词的不同形式，例如 *oxygenate* 和 *oxygenation*, 这是因为包含其中任何一个单词的文档都和查询关键字 *oxygenation* *相关*
  - 对于文本分类，如果有充足的训练数据，词干还原没什么好处。仅仅在数据稀疏的情况下，词干还原可以起到补偿作用（注：归并词项减少特征，减少参数数量，降低模型复杂度。数据稀疏在文本分类应用中通常是一个严重的问题）。一般来说，*激进的词干还原容易导致分类性能下降*

衡量分类效果的指标

- 分类准确率/Accuracy of classification
  - 学术界的主要评价指标
- 统计分类器的训练速度(training speed of statistical classifier)
  - 有些很快 - 朴素贝叶斯，Rocchio，KNN...
  - 有些代价昂贵 - SVM, 深度神经网络...
- 分类速度 (文档/小时)
  - 大部分算法速度接近
  - 除了kNN
- 人工构建的分类器：训练集创建的开销
  - 每个主题(topic)数据标记所花费的人工时间

衡量分类效果的指标

- 除准确率外，在现实应用中还需要考虑经济因素：
  - 面对一堆文本数据，怎样找到给定主题的文章？
    - 不分类
      - 开销大，难以计算
    - 人工分类
      - 开销可预估，取决于语料规模
    - 自动分类
      - 存在一定的错误率
    - 自动分类与人工检验的结合
      - 人工检验 不确定的/难以分类的/新的 样本
  - 通常最后一种方法准确率高，开销较小，被广泛应用
    - With more theory and Turkers: Werling, Chaganty, Liang, and Manning (2015). On-the-Job Learning with Bayesian Decision Theory. http://arxiv.org/abs/1506.03140

常见问题：概念漂移(Concept Drift)

- 类别随时间变换
- 例如：“POTUS (美国总统)”
  - 1998：clinton 是一个好特征
  - 2018：clinton 不是一个好特征
- 衡量文本分类系统好坏的一个指标是避免概念漂移的能力
  - 简单的模型，例如朴素贝叶斯，能够较好的避免概念漂移
- 特征选择：可能会导致概念漂移的问题

小结

- 决策树
  - 简单的非线性判别式分类器
  - 易于解释
  - 中等的（Moderately effective）文本分类效果
- 逻辑回归与支持向量机（SVM）
  - 线性判别式分类器
  - 接近最先进的(state of the art)分类器(也许人工神经网络除外)
- 分类器集成（ensembles）
  - 随机森林（bagging）
  - Boosting
- 不同方法的比较评价
- 真实应用：利用领域专有的结构（exploit domain specific structure）

相关资源

- S. T. Dumais. 1998. Using SVMs for text categorization, IEEE Intelligent Systems, 13(4)
- Yiming Yang, Xin Liu. 1999. A re-examination of text categorization methods. 22nd Annual International SIGIR
- Tong Zhang, Frank J. Oles. 2001. Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval* 4(1): 5-31
- Trevor Hastie, Robert Tibshirani and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, New York.
- T. Joachims, *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.
- Fan Li, Yiming Yang. 2003. A Loss Function Analysis for Classification Methods in Text Categorization. ICML 2003: 472-479.
- Tie-Yan Liu, Yiming Yang, Hao Wan, et al. 2005. Support Vector Machines Classification with Very Large Scale Taxonomy, SIGKDD Explorations, 7(1): 36-43.
- ‘Classic’ Reuters-21578 data set: http://www.daviddlewis.com/resources/testcollections/reuters21578/