信息组织与检索

第14讲：基于向量空间的分类器

主讲人：张蓉

华东师范大学数据科学与工程学院

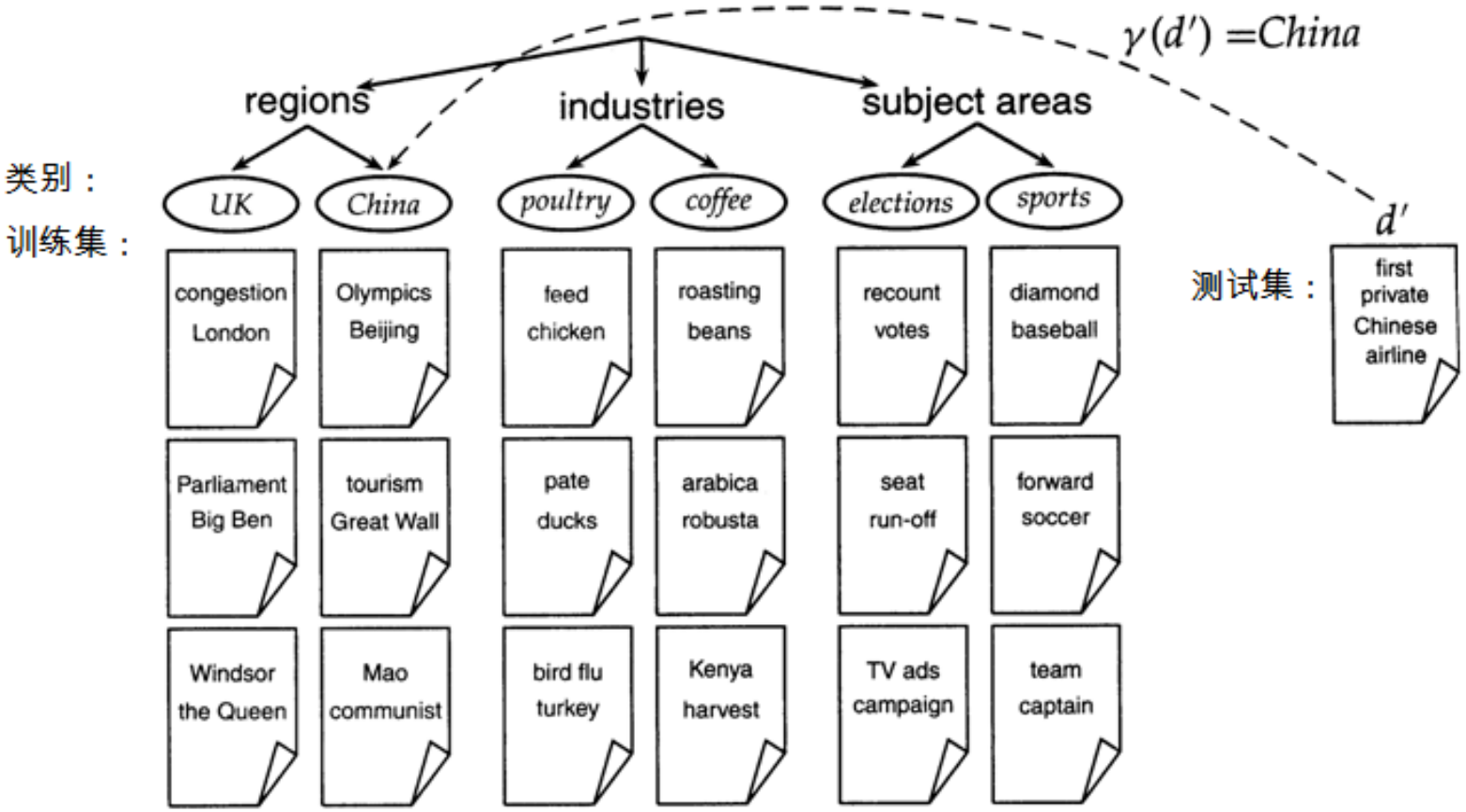
提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 线性分类器
7. 多类情况

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 线性分类器
7. 多类情况

文本(主题)分类


$\gamma(d') = China$
regions
industries
subject areas
类别：
UK
China
poultry
coffee
elections
sports
训练集：
congestion London
Olympics Beijing
feed chicken
roasting beans
recount votes
diamond baseball
Parliament Big Ben
tourism Great Wall
pate ducks
arabica robusta
seat run-off
forward soccer
Windsor the Queen
Mao communist
bird flu turkey
Kenya harvest
TV ads campaign
team captain
测试集：
d'
first private Chinese airline

本讲内容

- 特征选择：如何从原始特征空间中选出一部分子集？
- 向量空间分类：将文档表示成空间中的向量，然后进行分类
- Rocchio分类器：将Rocchio相关反馈思想应用于文本分类领域
- *K*近邻分类器
- 线性分类器
- 多类问题

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 线性分类器
7. 多类情况

特征选择

- 文本分类中，通常要将文本表示在一个高维空间下，每一维对应一个词项
- 本讲义中，我们不特意区分不同的概念：每个坐标轴 = 维 = 词语 = 词项 = 特征
- 许多维上对应是罕见词
- 罕见词可能会误导分类器
- 这些会误导分类器的罕见词被称为噪音特征（noise feature）
- 去掉这些噪音特征会同时提高文本分类的效率和效果
- 上述过程称为特征选择（feature selection）

噪音特征的例子

- 比如我们将对文本是否属于China类进行判断
- 假定某个罕见词项，比如 ARACHNOCENTRIC，没有任何关于 China 类的信息
- . . . 但是在训练集中，ARACHNOCENTRIC的所有出现正好都在 China这个类别中
- 这种情况下，我们就可能训练得到一个分类器，它认为ARACHNOCENTRIC标志着类别 China的出现
- 这种从训练集中的偶然现象学习得到的一般化结果称为过学习(overfitting)
- 特征选择能减少过学习可能，并提高分类器的精度

不同的特征选择方法

- 特征选择方法主要基于其所使用特征效用指标来定义。
- 特征效用指标：
  - 频率法 – 选择高频词项
  - 互信息(Mutual information) – 选择具有最高互信息的那些词项
    - 这里的互信息也叫做信息增益（information gain）
  - 卡方检验(Chi-square)

互信息(Mutual information)

- 特征效用 $A(t, c)$ 采用词项 t 和类别 c 的期望互信息 *(Expected Mutual Information)* 来计算
- MI给出的是词项所包含的有关类别的信息及类别包含的有关词项的信息量
- 比如，如果词项的出现与否与类别独立(不同类别中包含和不包含词项的文档比例完全一样)
- 定义：

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)}$$

(期望)互信息的另一种定义

- 信息增益(Information Gain, *IG*)：该term为整个分类所能提供的信息量(不考虑任何特征的熵和考虑该特征后的熵的差值)

$$S(p_1, p_2, \cdots, p_n) = -K\sum_{i=1}^{n} p_i \log_2 p_i$$

$$\mathrm{IG(t)} = \mathrm{Entropy}(S) - \mathrm{Expected\ Entropy}(S_t) = -\sum_{i=1}^{M} P(c_i)\log P(c_i)$$

$$-[P(t)(-\sum_{i=1}^{M} P(c_i \mid t)\log P(c_i \mid t)) + P(\bar{t})(-\sum_{i=1}^{M} P(c_i \mid \bar{t})\log P(c_i \mid \bar{t}))]$$

如何计算互信息MI

- 基于MLE估计，实际使用的计算公式为：

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

- N10: 包含 $t(et=1)$但是不属于c($ec=0$)的文档数目；
- N11:包含 $t(et=1)$同时属于c($ec=1$)的文档数目；
- N01: 不包含 $t(et=0)$但是属于c($ec=1$)的文档数目；
- N00:不包含 $t(et=0)$也不属于c($ec=0$)的文档数目；
- N = N00 + N01 + N10 + N11

$$I(U;C)=\sum_{e_t\in\{1,0\}}\sum_{e_c\in\{1,0\}}P(U=e_t,C=e_c)\log_2\frac{P(U=e_t,C=e_c)}{P(U=e_t)P(C=e_c)}$$

Reuters 语料中*poultry*/EXPORT的MI计算

	$e_c = e_{\text{poultry}} = 1$	$e_c = e_{\text{poultry}} = 0$
$e_t = e_{\text{export}} = 1$	$N_{11} = 49$	$N_{10} = 27\,652$
$e_t = e_{\text{export}} = 0$	$N_{01} = 141$	$N_{00} = 774\,106$

$$
\begin{aligned}
I(U;C) = {} & \frac{49}{801\,948}\log_2\frac{801\,948\times 49}{(49+27\,652)(49+141)} \\
& + \frac{141}{801\,948}\log_2\frac{801\,948\times 141}{(141+774\,106)(49+141)} \\
& + \frac{27\,652}{801\,948}\log_2\frac{801\,948\times 27\,652}{(49+27\,652)(27\,652+774\,106)} \\
& + \frac{774\,106}{801\,948}\log_2\frac{801\,948\times 774\,106}{(141+774\,106)(27\,652+774\,106)} \\
\approx {} & 0.000\,110\,5
\end{aligned}
$$

MI 特征选择的结果

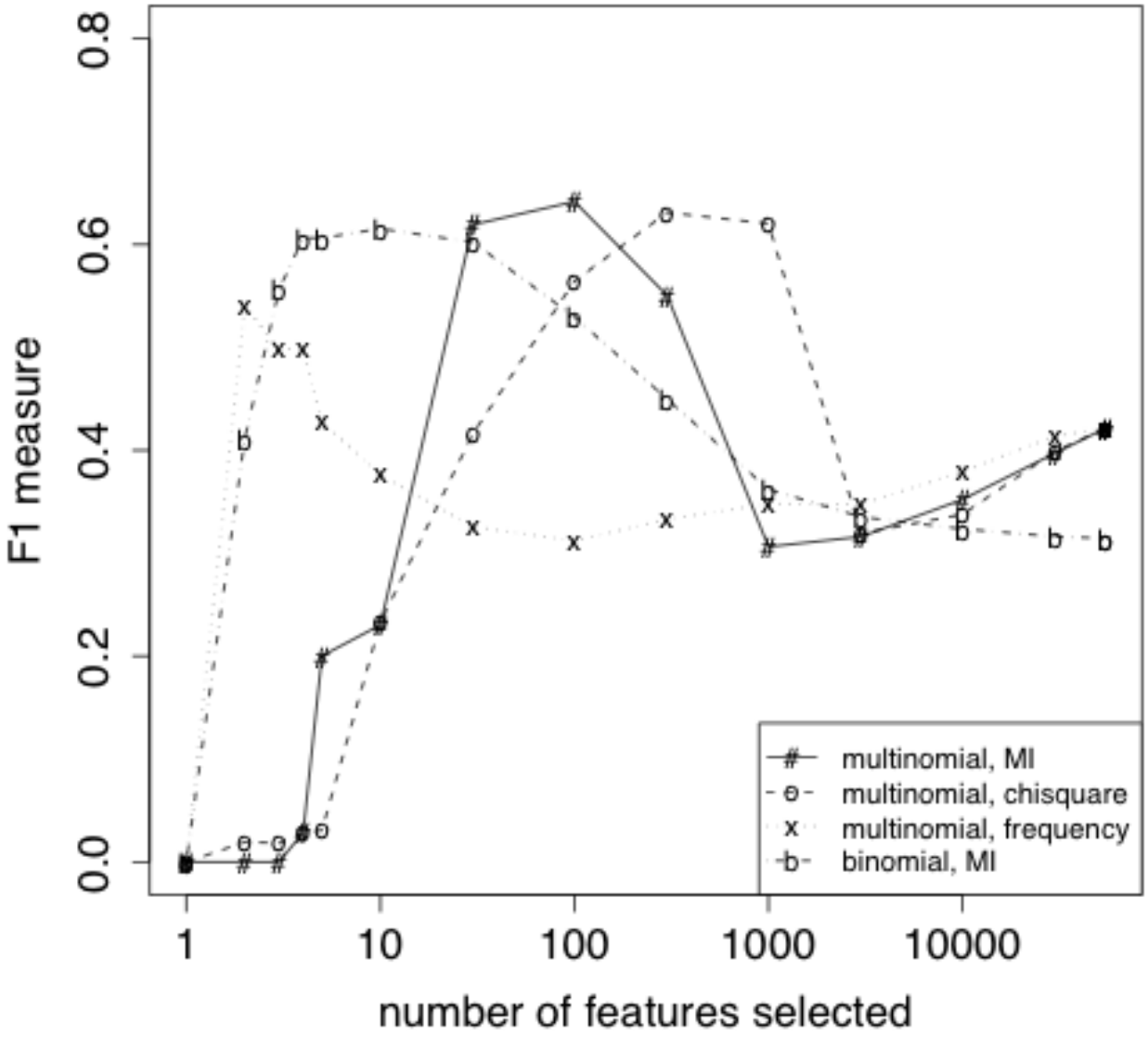
Class: *coffee*

term	MI
COFFEE	0.0111
BAGS	0.0042
GROWERS	0.0025
KG	0.0019
COLOMBIA	0.0018
BRAZIL	0.0016
EXPORT	0.0014
EXPORTERS	0.0013
EXPORTS	0.0013
CROP	0.0012

Class: *sports*

term	MI
SOCCER	0.0681
CUP	0.0515
MATCH	0.0441
MATCHES	0.0408
PLAYED	0.0388
LEAGUE	0.0386
BEAT	0.0301
GAME	0.0299
GAMES	0.0284
TEAM	0.0264

朴素贝叶斯: 特征选择的效果



(multinomial = 多项式朴素贝叶斯)
binomial= 贝努利朴素贝叶斯)

朴素贝叶斯中的特征选择

- 一般来说，为了获得较好的结果，朴素贝叶斯有必要进行特征选择
- 对于一些其他文本分类器方法来说，特征选择也是获得好结果的必要手段

其它特征选择方法

- 基于*DF*的选择方法 (DF Thresholding)
  - Term的*DF*小于某个阈值去掉(太少，没有代表性)

其它特征选择方法(续)

- χ^2 统计量(念xi, chi，卡方法)：度量两者(term和类别)独立性的缺乏程度， χ^2 越大，独立性越小，相关性越大(*N=A+B+C+D*)

$$\chi^2(t,c)=\frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}$$

	c	~c
t	A	B
~t	C	D

$$\chi^2_{AVG}(t)=\sum_{i=1}^{m}P(c_i)\chi^2(t,c_i)$$

$$\chi^2_{MAX}(t)=\max_{i=1}^{m}\{\chi^2(t,c_i)\}$$

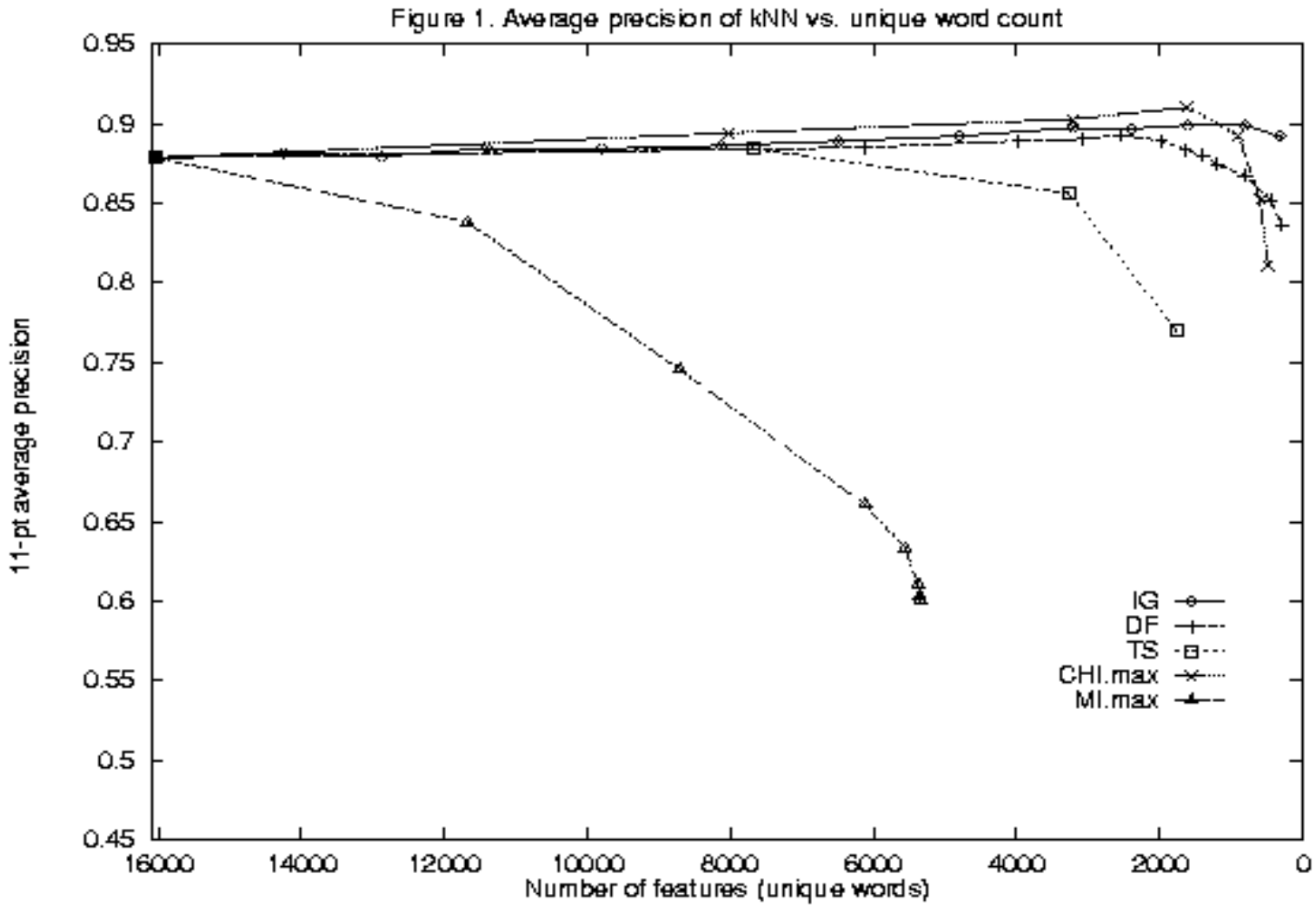
- (点)互信息(Pointwise Mutual Information， *PMI*)：*MI* 越大*t*和*c*共现程度越大

$$I(t,c)=\log\frac{P(t\wedge c)}{P(t)P(c)}=\log\frac{P(t|c)}{P(t)}=\log\frac{A\times N}{(A+C)(A+B)}$$

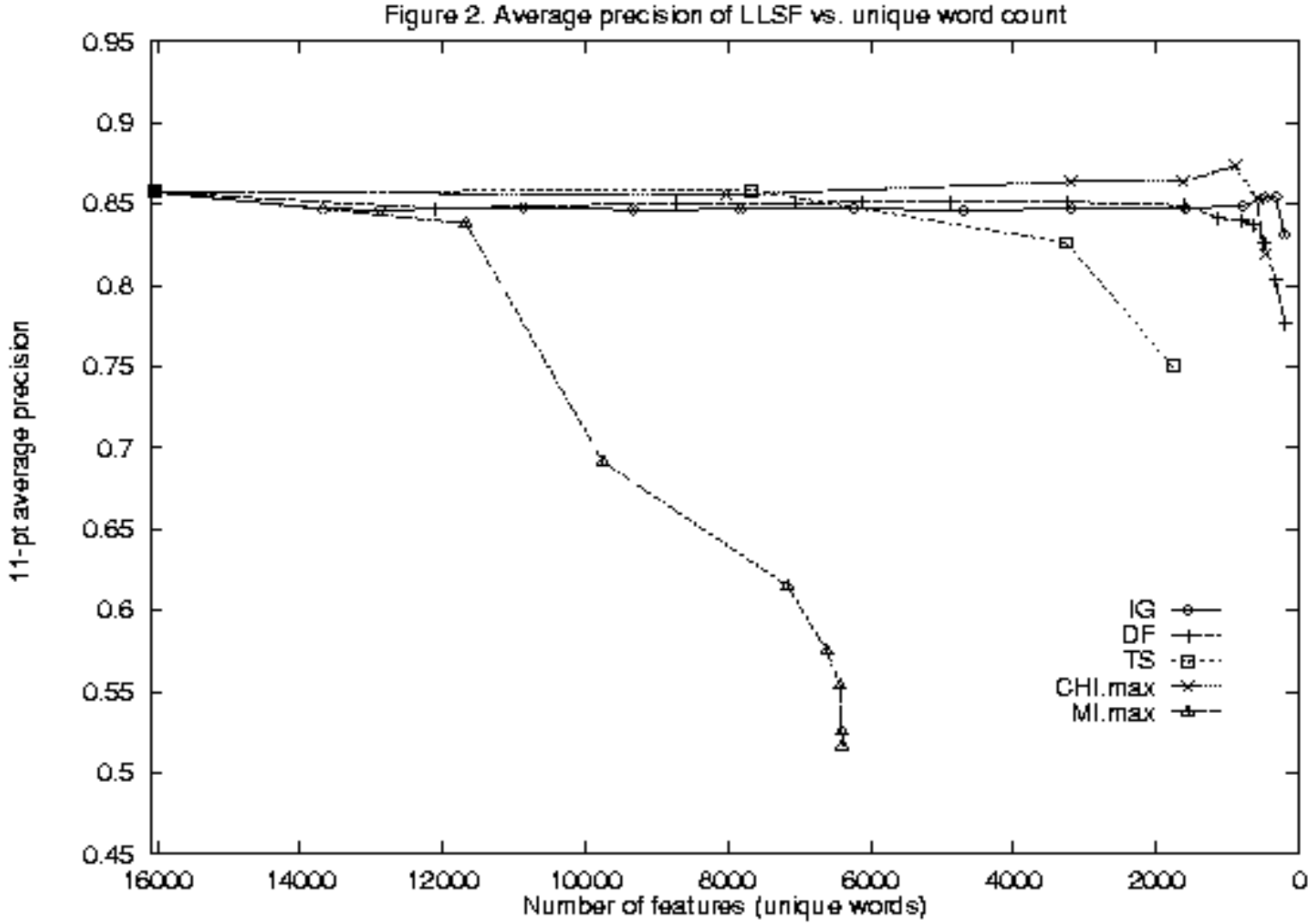
$$I_{AVG}(t)=\sum_{i=1}^{m}P(c_i)I(t,c_i)$$

$$I_{MAX}(t)=\max_{i=1}^{m}P(c_i)I(t,c_i)$$

特征选择方法的性能比较(1)



特征选择方法的性能比较(2)



特征选择方法的性能比较(3)

Yang Yi-ming（2008） 的实验结论

Method	DF	IG	CHI	MI	TS
favoring common terms	Y	Y	Y	N	Y/N
using categories	N	Y	Y	Y	N
using term absence	N	Y	Y	N	N
performance in kNN/LLSF	excellent	excellent	excellent	poor	ok

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 线性分类器
7. 多类情况

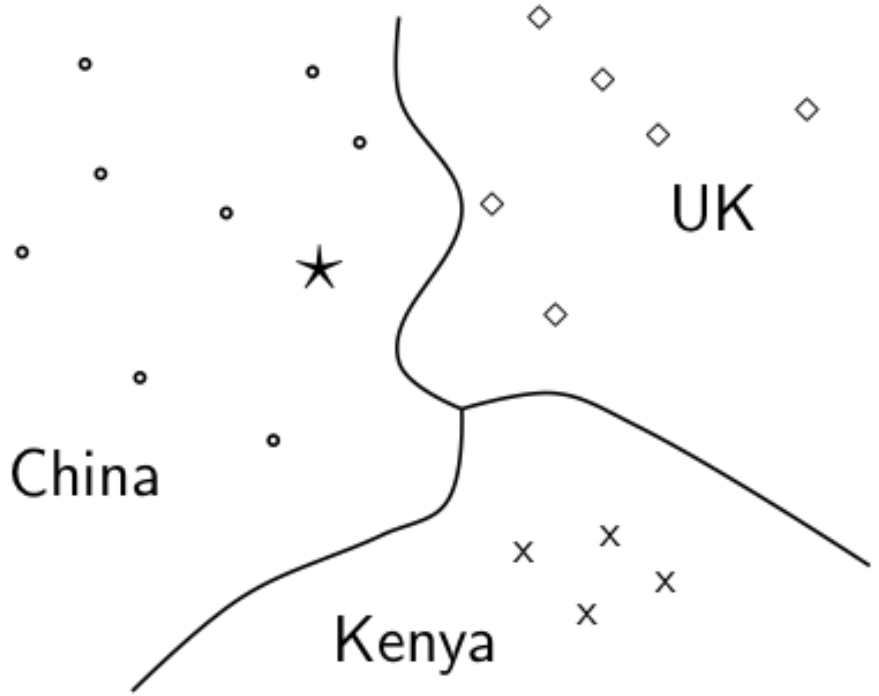
向量空间表示回顾

- 每篇文档都表示一个向量，每一维对应一个词项
- 词项就是坐标轴
- 通常都高维: 100,000多维
- 通常要将向量归一化到单位长度
- 如何在该空间下进行分类?

向量空间分类

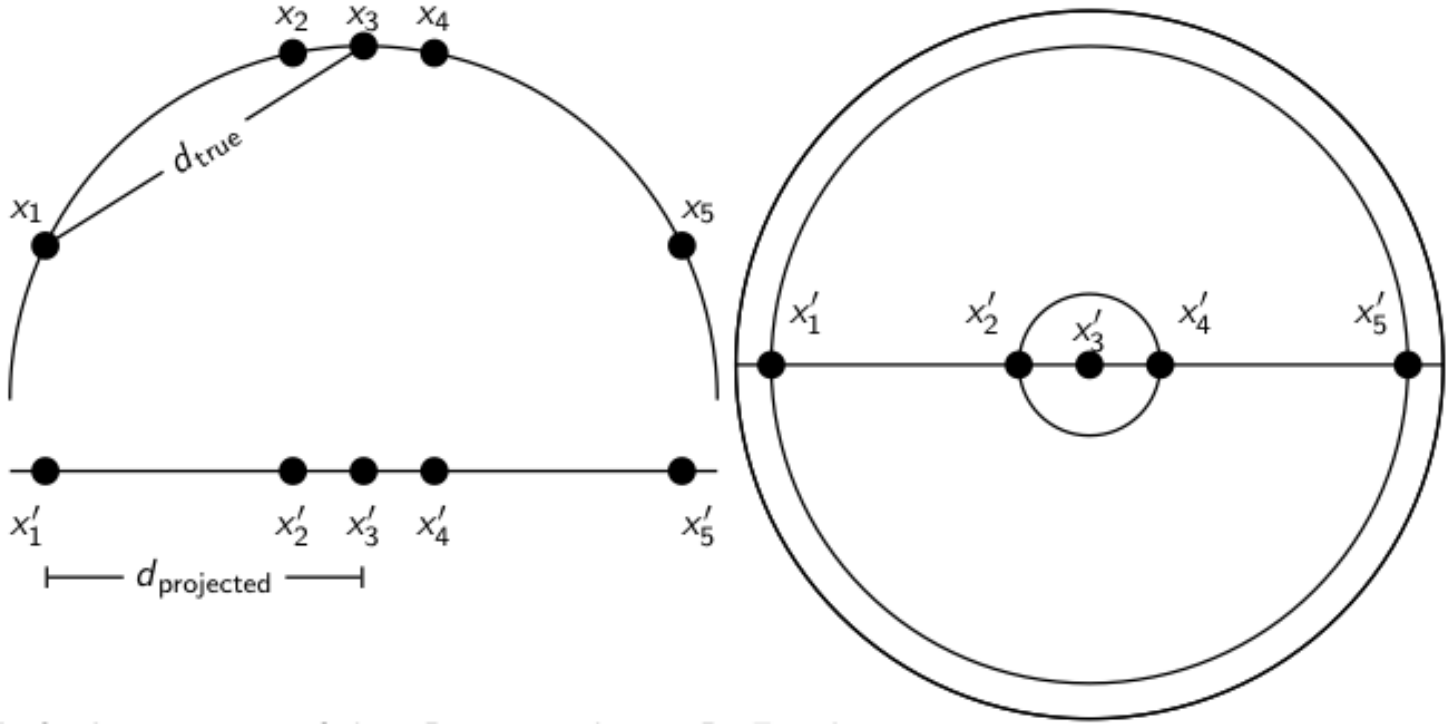
- 同前面一样，训练集包含一系列文档，每篇都标记着它的类别
- 在向量空间分类中，该集合对应着空间中一系列标记的点或向量
- 假设 1: 同一类中的文档会构成一片连续区域（contiguous region）
- 假设2: 来自不同类别的文档没有交集
- 接下来我们定义直线、平面、超平面来将上述不同区域分开

向量空间中的类别



- 文档*到底是属于UK、China还是Kenya类？首先找到上述类别之间的分类面，然后确定文档所属类别，很显然按照图中分类面，文档应该属于China类
- 如何找到分类面并将文档判定给正确类别是本讲的重点。

题外话: 2D/3D 图形可能会起误导作用



左图：从二维空间的半圆映射到一维直线上。点x_1、x_2、x_3、x_4、x_5的X轴坐标分别是−0.9、−0.2、0、0.2和0.9，距离$|x_2x_3| \approx 0.201$，和$|x_2'x_3'|=0.2$只有0.5%的差异，但是当对较大的区域进行投影的话，比如$|x_1x_3| / |x_1'x_3'| = d_{\text{true}}/d_{\text{projected}} \approx 1.06/0.9 \approx 1.18$却会产生较大的差异（18%）。

右图：相应的从三维的半球面到二维平面上的投影

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 线性分类器
7. 多类情况

相关反馈(Relevance feedback)

- 在相关反馈中，用户将文档标记为相关/不相关
- 相关/不相关可以看成两类
- 对每篇文档，用户觉得它到底属于哪个类别
- IR 系统使用用户的类别判定结果来构建一个能反映信息需求的更好的查询
- . . . 并返回更好的文档
- 相关反馈可以看成文本分类的一种形式。

利用Rocchio方法进行向量空间分类

- 相关反馈和文本分类的主要区别在于:
- 在文本分类中，训练集作为输入的一部分事先给定
- 在相关反馈中，训练集在交互中创建

Rocchio分类: 基本思想

- 计算每个类的中心向量
  - 中心向量是所有文档向量的算术平均
- 将每篇测试文档分到离它最近的那个中心向量

中心向量的定义

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

其中 Dc 是所有属于类别 c 的文档， $\vec{v}(d)$是文档d的向量空间表示

Rocchio算法

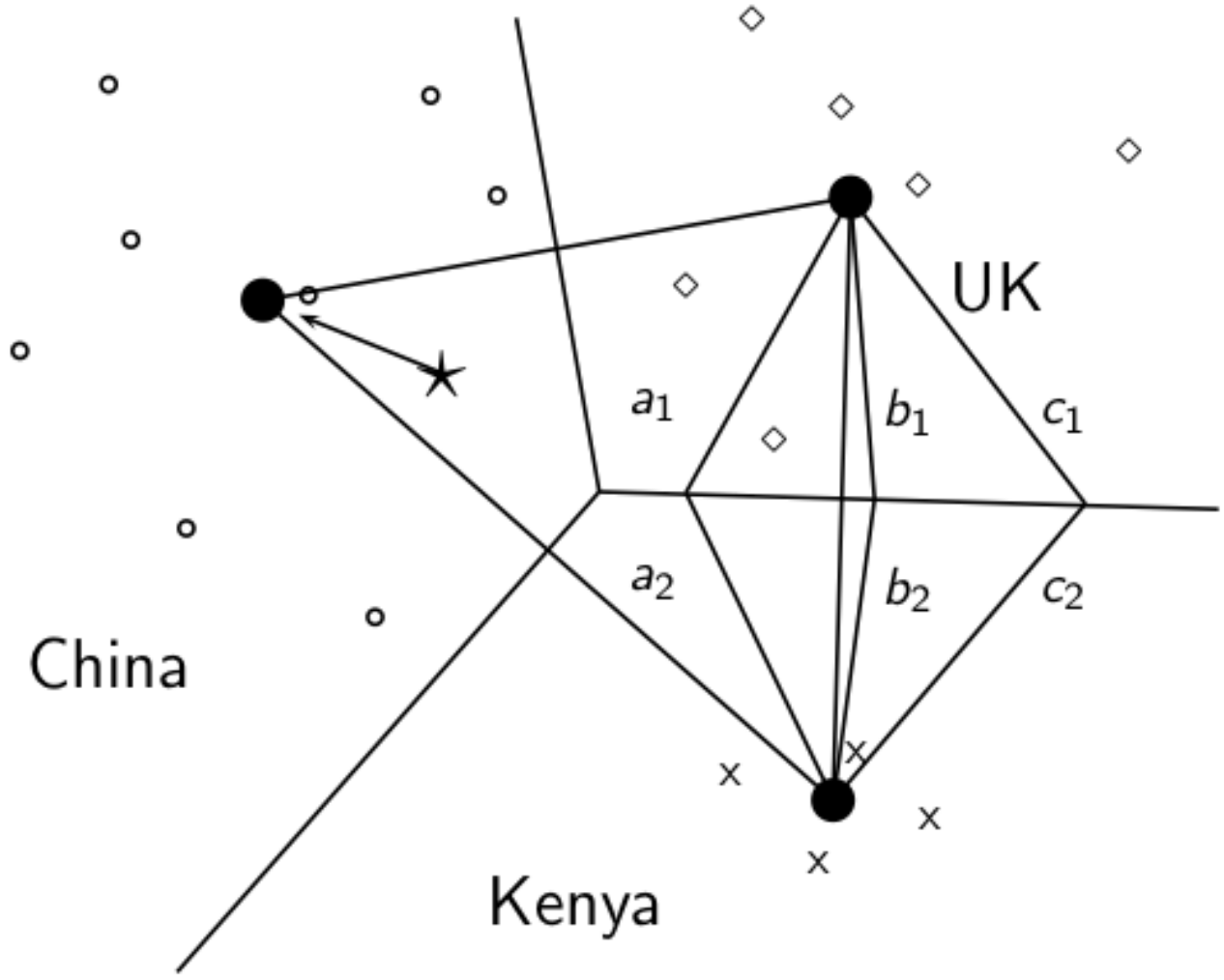
TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)

1 **for each** $c_j \in \mathbb{C}$
2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3 $\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)

1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

Rocchio算法示意



Rocchio性质

- Rocchio简单地将每个类别表示成其中心向量
  - 中心向量可以看成类别的原型(prototype)
- 分类基于文档向量到原型的相似度或聚类来进行
- 并不保证分类结果与训练集一致，即得到分类器后，不能保证训练集中的文档能否正确分类

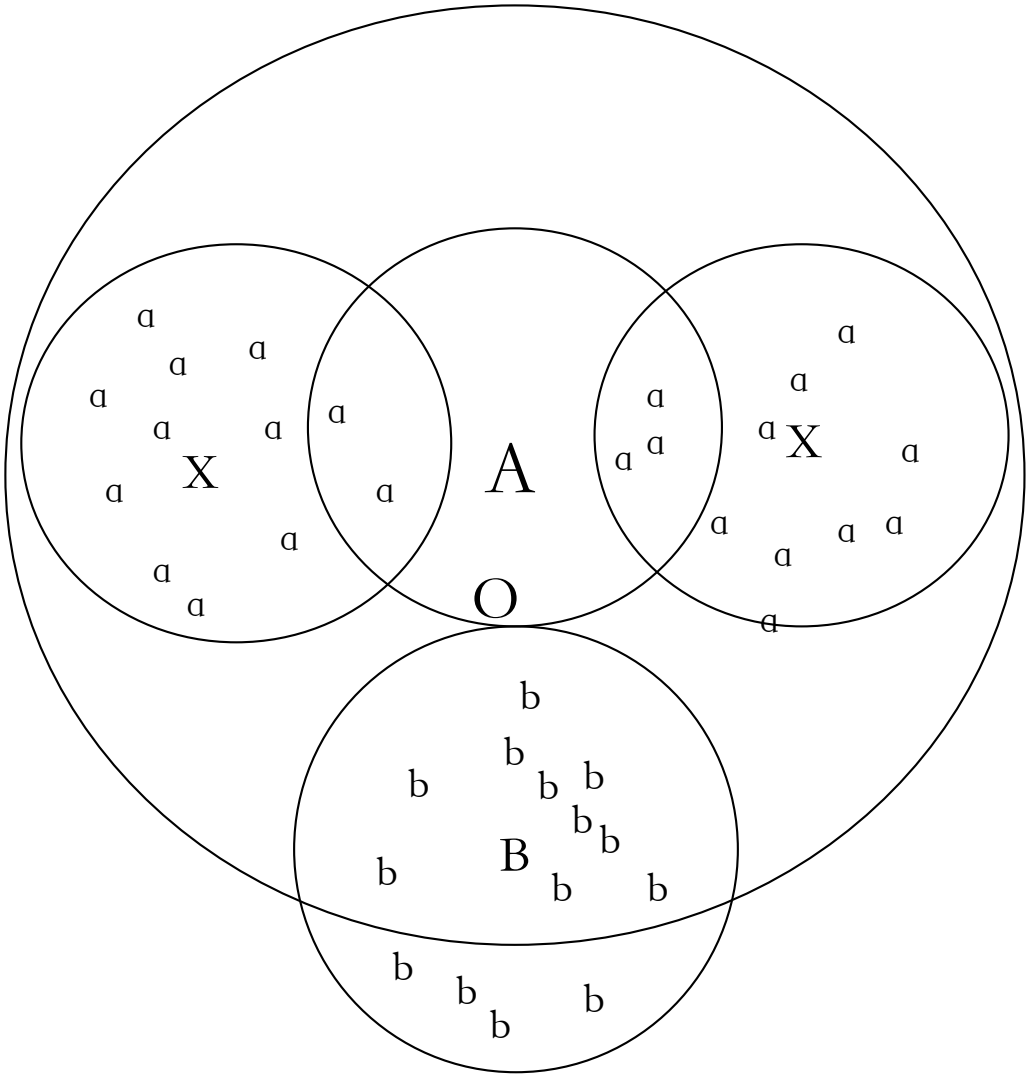
Rocchio算法的时间复杂度

mode	time complexity
training	$\Theta(\lvert\mathbb{D}\rvert L_{\text{ave}} + \lvert\mathbb{C}\rvert\lvert V\rvert) \approx \Theta(\lvert\mathbb{D}\rvert L_{\text{ave}})$
testing	$\Theta(L_{\text{a}} + \lvert\mathbb{C}\rvert M_{\text{a}}) \approx \Theta(\lvert\mathbb{C}\rvert M_{\text{a}})$

Rocchio vs. 朴素贝叶斯

- 很多情况下，Rocchio的效果不如朴素贝叶斯
- 一个原因是，Rocchio算法不能正确处理非凸、多模式类别问题

Rocchio不能正确处理非凸、多模式类别问题



课堂练习: 对于左图的A/B分类问题，为什么Rocchio方法难以有效处理?

- A 是所有a的中心向量，B是所有b的中心向量
- 点o 离A更近
- 但是o更适合于b类
- A 是一个有两个原型多模式类别
- 但是，在Rocchio算法中，每个类别只有一个原型

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 多类情况

kNN分类器

- kNN 是另外一种基于向量空间的分类方法
- 该方法非常简单，也容易实现
- 在大多数情况下，kNN的效果比朴素贝叶斯和Rocchio要好
- 如果你急切需要一种精度很高分类器并很快投入运行 . .
- . . . 如果你不是特别关注效率 . . .
- . . . 那么就使用kNN

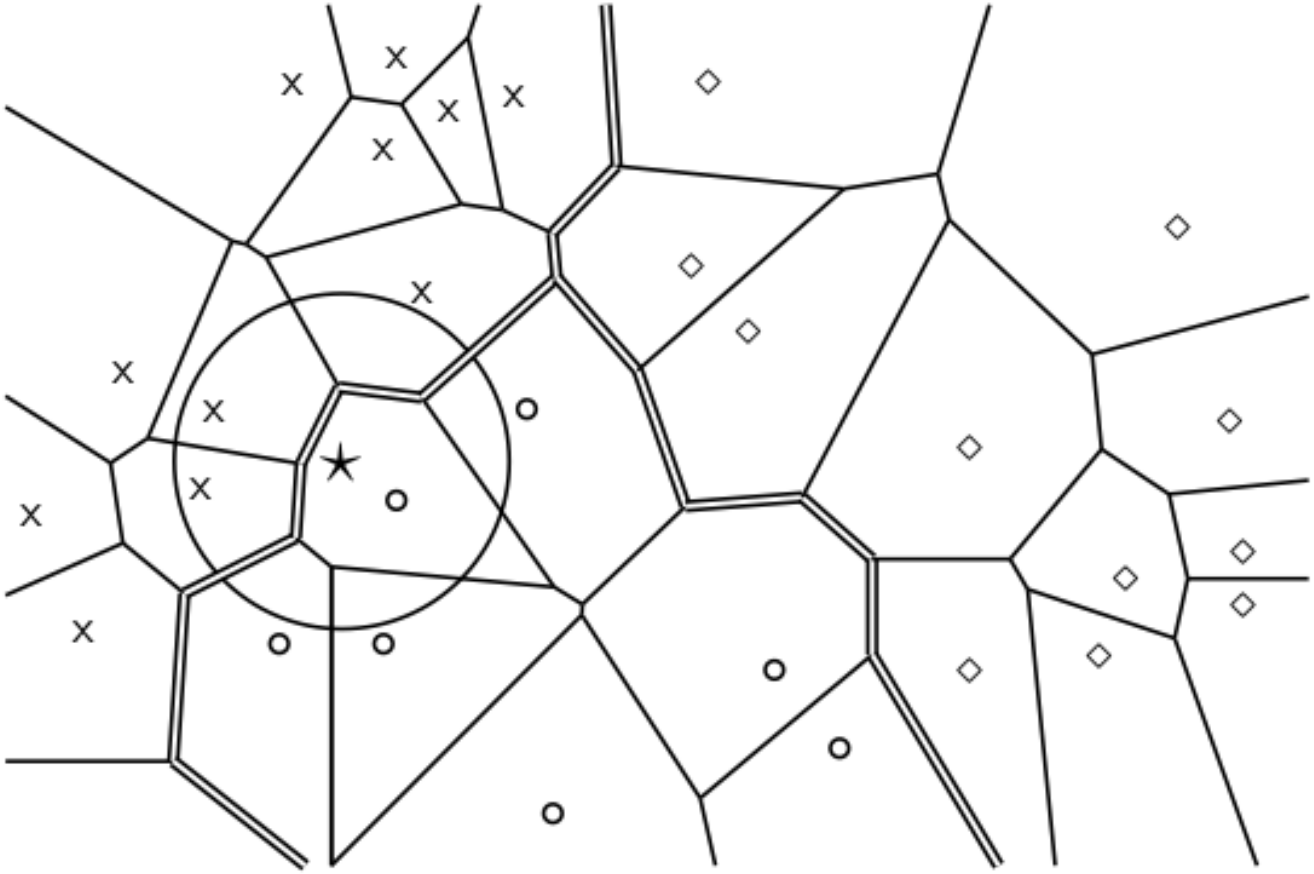
kNN分类

- kNN = *k* nearest neighbors，*k*近邻
- $k = 1$ 情况下的kNN (最近邻): 将每篇测试文档分给训练集中离它最近的那篇文档所属的类别。
- 1NN 不很鲁棒——一篇文档可能会分错类或者这篇文档本身就很反常
- $k > 1$ 情况下的kNN: 将每篇测试文档分到训练集中离它最近的k篇文档所属类别中最多的那个类别
- kNN的基本原理: 邻近性假设
  - 我们期望一篇测试文档*d*与训练集中*d*周围邻域文档的类别标签一样。

概率型kNN

- kNN的概率型版本: $P(c|d)$ = d的最近的k个邻居中属于c类的比例
- 概率型kNN: 将d分到具有最高概率$P(c|d)$的类别c中

概率kNN



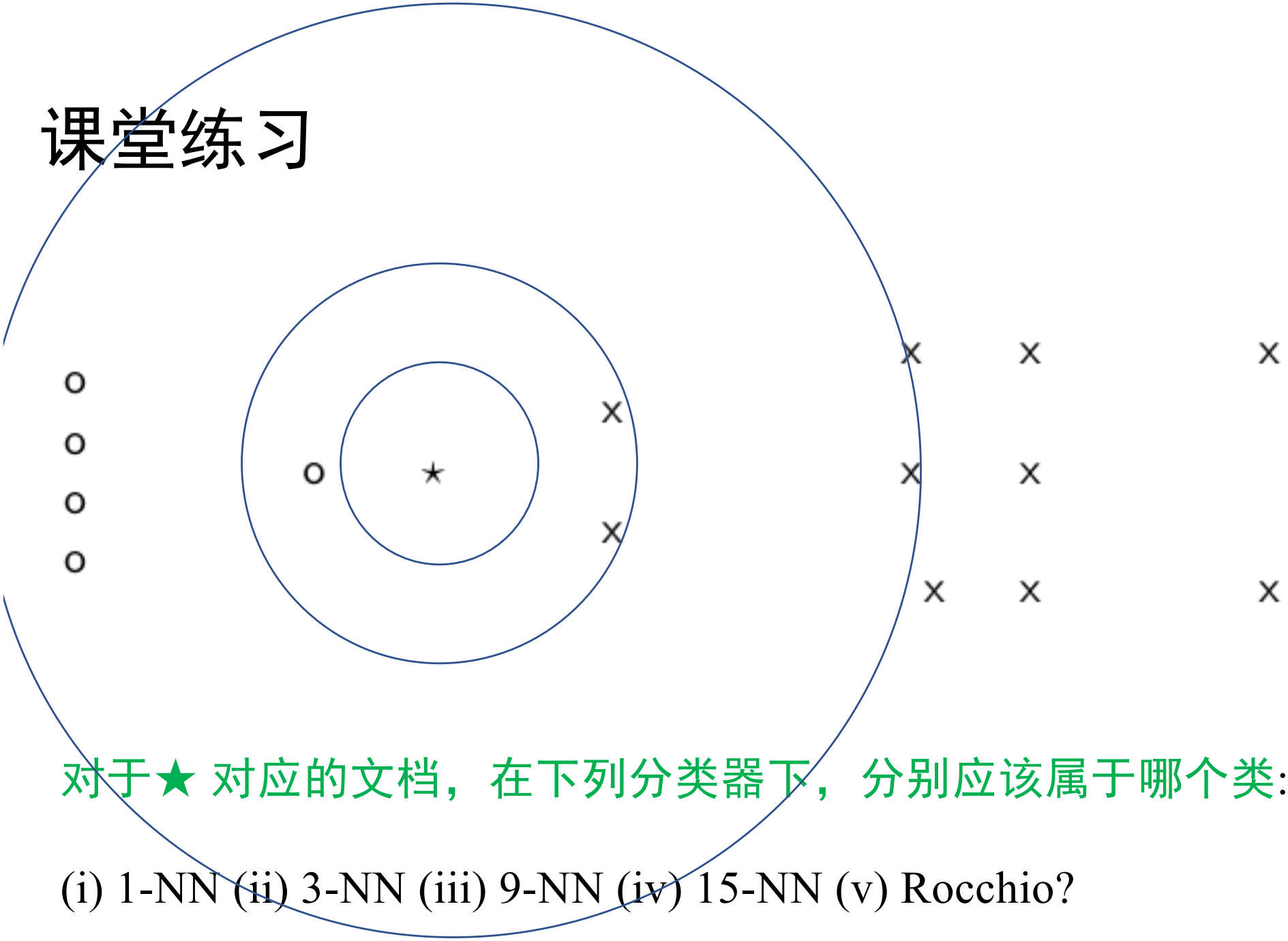
对于★ 对应的文档，在1NN和 3NN下，分别应该属于哪个类？

kNN 算法

TRAIN-KNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' \leftarrow$ PREPROCESS($\mathbb{D}$)
2 $k \leftarrow$ SELECT-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathbb{D}', k$

APPLY-KNN($\mathbb{D}', k, d$)
1 $S_k \leftarrow$ COMPUTENEARESTNEIGHBORS($\mathbb{D}', k, d$)
2 **for each** $c_j \in \mathbb{C}(\mathbb{D}')$
3 **do** $p_j \leftarrow |S_k \cap c_j|/k$
4 **return** $\arg\max_j p_j$

课堂练习



对于★ 对应的文档，在下列分类器下，分别应该属于哪个类:

(i) 1-NN (ii) 3-NN (iii) 9-NN (iv) 15-NN (v) Rocchio?

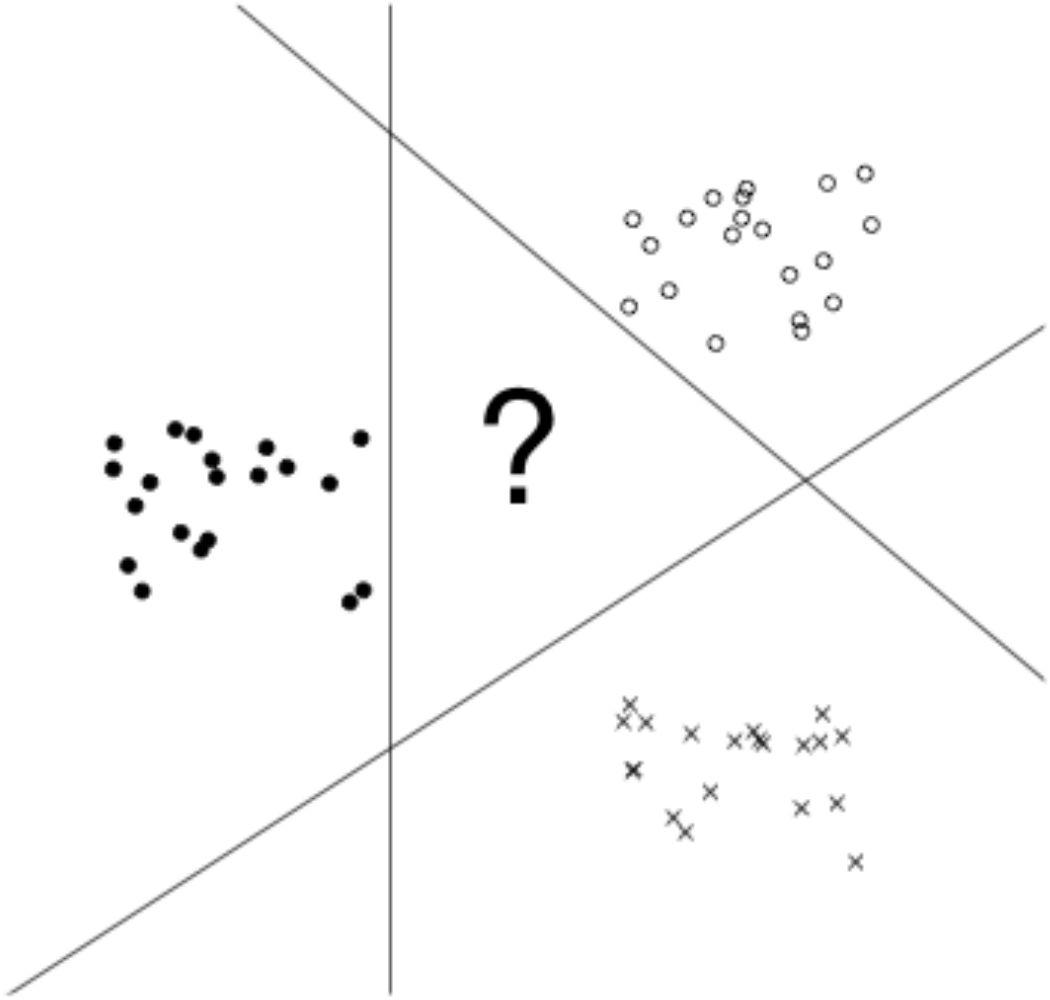
kNN: 讨论

- 不需要训练过程
  - 但是，文档的线性预处理过程和朴素贝叶斯的训练开销相当
  - 对于训练集来说我们一般都要进行预处理，因此现实当中kNN的训练时间是线性的。
- 当训练集非常大的时候，kNN分类的精度很高
- 如果训练集很小，kNN可能效果很差。

提纲

1. 上一讲回顾
2. 特征选择
3. 基于向量空间的分类方法
4. Rocchio
5. kNN
6. 多类情况

多类(>2)的情况下的超平面组合



单标签问题（One-of problem）

- 单标签分类问题，也称single label problem
  - 类别之间互斥
  - 每篇文档属于且仅属于某一个类
  - 例子：文档的语言类型 (假定：任何一篇文档都只包含一种语言)

基于线性分类器的单标签分类

- 对于单标签分类问题（比如A、B、C三类），可以将多个二类线性分类器(A vsBC、B vs. AC、C vs. AB)进行如下组合：
  - 对于输入文档，独立运行每个分类器
  - 将分类器排序 (比如按照每个分类器输出的在A、B、C上的得分)
  - 选择具有最高得分的类别

多标签问题(Any-of problem)

- 多标签分类问题，也称multilabel classification
  - 一篇文档可以属于0、1或更多个类
  - 针对某个类的决策并不影响其他类别上的决策
  - 一种“独立”类型，但是不是统计意义上的“独立”
  - 例子：主题分类
  - 通常：地区、主题领域、工业等类别之上的决策是互相独立的

基于线性分类器的多标签分类

- 对于多标签分类问题（比如A、B、C三类），可以将多个二类线性分类器(A vsBC、B vs. AC、C vs. AB)进行如下组合：
  - 对测试文档独立地运行每个分类器
  - 按照每个分类器自己的输出结果进行独立决策

本讲小结

- 特征选择：MI
- 向量空间分类：将文档表示成空间中的向量，然后进行分类
- Rocchio分类器：将Rocchio相关反馈思想应用于文本分类领域
- *K*近邻分类器
- 多类问题