

信息检索与数据挖掘

第10章 文本分类

part1: 文本分类及朴素贝叶斯方法

part2: 基于向量空间的文本分类

part3: 支持向量机及机器学习方法

回顾：什么是文本分类

- **Taxonomies and Classification**
- 文本分类中，给定文档 $d \in X$ 和一个固定的类别集合 $C = \{c_1, c_2, \dots, c_J\}$ ，其中 X 表示文档空间（**document space**），类别（**class**）也通常称为类（**category**）或类标签（**label**）。
- 分类方法
 - 手工方法 \rightarrow 规则方法 \rightarrow 基于学习的文本分类
- 文本分类中的类别、训练集及测试集
- 无监督/有监督的学习
- **IR** 中的文本分类应用

回顾：Naive Bayes text classification

- 在文本分类中，我们的目标是找出文档最可能属于的类别。对于NB 分类来说，最可能的类是具有MAP估计值的结果 c_{map} ：

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg \max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)$$

- 如何估计参数 $\hat{P}(c)$ 及 $\hat{P}(t_k | c)$ ？

$$\hat{P}(c) = \frac{N_c}{N} \quad \hat{P}(t | c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

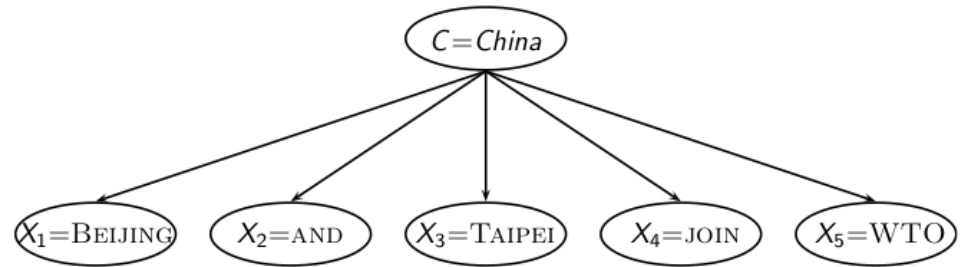
- 零概率问题→平滑

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

回顾：朴素贝叶斯分类器的生成模型

- 文本分类的步骤

- 训练
- 测试

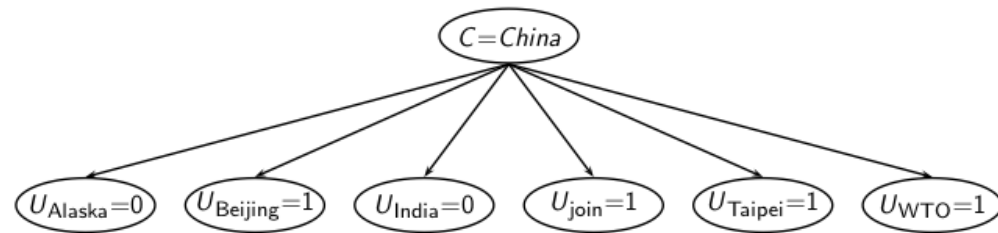


- 建立 NB 分类器有两种不同的方法

- Multinomial NB model
- Bernoulli model

- **Naive Bayes algorithm**

- $\hat{P}(t | c)$ 的估计策略不同
- 未出现词项在分类中的使用不同



Naive Bayes algorithm

$\hat{P}(t/c)$ 的估计策略不同

未出现词项在分类中的使用不同

TRAINMULTINOMIALNB(C, D)	TRAINBERNOULLINB(C, D)
<pre> 1 V ← EXTRACTVOCABULARY(D) 2 N ← COUNTDOCUMENTS(D) 3 for each c ∈ C 4 do N_c ← COUNTWORDS(D, c) 5 prior[c] ← N_c / N 6 text_c ← COUNTTERMS(D, c) 7 for each t ∈ V 8 do T_{ct} ← COUNT(D, c, t) 9 for each t ∈ V 10 do condprob[t][c] ← (T_{ct} + 1) / (N_c + 2) 11 return V, prior, condprob </pre>	<pre> 1 V ← EXTRACTVOCABULARY(D) 2 for each c ∈ C 3 do class ← CLASSCONTAININGTERM(D, c, t) 4 for each t ∈ V 5 do score[c] += log condprob[t][c] 6 return arg max_{c ∈ C} score[c] </pre>
$\arg \max_{c \in C} P(c) \prod_{1 \leq k \leq n_d} P(X_k = t_k c)$	$\arg \max_{c \in C} P(c) \prod_{1 \leq i \leq M} P(U_i = e_i c)$
学习方法不同，得到的分类函数 γ 不同	
<pre> 1 W ← EXTRACTVOCABULARY(D) 2 for each c ∈ C 3 do score[c] ← log prior[c] 4 for each t ∈ V 5 do score[c] += log condprob[t][c] 6 return arg max_{c ∈ C} score[c] </pre>	<pre> 6 then score[c] += log condprob[t][c] 7 else score[c] += log(1 - condprob[t][c]) 8 return arg max_{c ∈ C} score[c] </pre>

multinomial model

Bernoulli model

回顾：朴素贝叶斯分类器的性质

- 多项式模型 $P(d|c) = P(\langle t_1, \dots, t_k, \dots, t_{n_d} \rangle | c)$
- 贝努利模型 $P(d|c) = P(\langle e_1, \dots, e_i, \dots, e_M \rangle | c)$
- 朴素贝叶斯的条件独立性假设

$$\text{Multinomial } P(d|c) = P(\langle t_1, \dots, t_{n_d} \rangle | c) = \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$$

$$\text{Bernoulli } P(d|c) = P(\langle e_1, \dots, e_M \rangle | c) = \prod_{1 \leq i \leq M} P(U_i = e_i | c).$$

- 朴素贝叶斯的位置独立性假设

$$P(X_{k_1} = t | c) = P(X_{k_2} = t | c).$$

- 准确估计概率 \Rightarrow 精确预测， 反之并不成立！

回顾4-1：文本分类的评价

- 文本分类的目标
 - 使得测试数据上的分类错误率最小
- 常用的指标
 - 正确率、召回率、F1值、分类精确率等
- 多个分类器的文档集
 - 当对具有多个分类器的文档集进行处理时，往往需要计算出一个融合了每个分类器指标的综合指标
- 宏平均和微平均
 - 微平均计算中大类起支配作用
 - 度量小类上的效果，往往需要计算宏平均指标

回顾4-2: 文本分类的评价

“第6章 检索的评价” 中的评价方法

- 无序检索结果的评价 ← 基于集合的评价方法

- 查准率/正确率 Precision
- 查全率/召回率 Recall
- F值是查准率和查全率的加权调和平均数

单个查询

- 有序检索结果的评价

- 查准率-查全率曲线
- 固定检索等级的查准率 Precision@k
- 平均正确率 (Average Precision, AP)
- 平均查准率均值 Mean Average Precision (MAP)
- GMAP (Geometric MAP)
- NDCG

多个查询

回顾4-3：文本分类的评价 检索评价的MAP准则

教材 8.4小节, p109

- **平均查准率均值 Mean Average Precision (MAP)**

- 在每个相关文档位置上查准率的平均值，被称为平均查准率 (AP)
- 对所有查询求平均，就得到平均查准率均值 (MAP)

$$\text{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} \text{Precision}(R_{jk})$$

- **参数说明**

MAP是宏平均还是微平均？

- Q 为信息需求, $q_j \in Q$ 所对应的所有相关文档集合为 $\{d_1, d_2, \dots, d_{m_j}\}$, R_{jk} 是查询 q_j 的返回结果、该结果中包含 $\{d_1, d_2, \dots, d_k\}$ 而不含有 d_{k+1} 及以后的相关文档

MAP:某查询集合对应的多条正确率-召回率曲线下面积的平均值

回顾4-4：文本分类的评价 多个查询的评价指标

- 多个查询的评价指标，一般就是对单个查询的评价进行求平均。平均的求法一般有两种：
 - 宏平均（Macro Average）：对每个查询求出某个指标，然后对这些指标进行算术平均
 - 微平均（Micro Average）：将所有查询视为一个查询，将各种情况的文档总数求和，然后进行指标的计算

查询q1、q2的标准答案数分别为100个和50个，某系统对q1检索出80个结果，其中正确数目为40，系统对q2检索出30个结果，其中正确数目为24，则：

$$P1=40/80=0.5, R1=40/100=0.4$$

$$P2=24/30=0.8, R2=24/50=0.48$$

$$\text{MacroP}=(P1+P2)/2=0.65$$

$$\text{MacroR}=(R1+R2)/2=0.44$$

$$\text{MicroP}=(40+24)/(80+30)=0.58$$

$$\text{MicroR}=(40+24)/(100+50)=0.43$$

文本分类评价时候的宏平均
和微平均指标计算与多个查
询的评价一致

回顾3-1：通过特征选择提高分类器效率

互信息 $A(t, c) = I(U_t; C_c)$
 χ^2 统计量 $A(t, c) = \chi^2 t, c)$
 词项频率 $A(t, c) = N(t, c)$

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

$$F_{\beta=1} = \frac{2PR}{P+R}$$

Precision = $P(\text{relevant}|\text{retrieved})$

Recall = $P(\text{retrieved}|\text{relevant})$

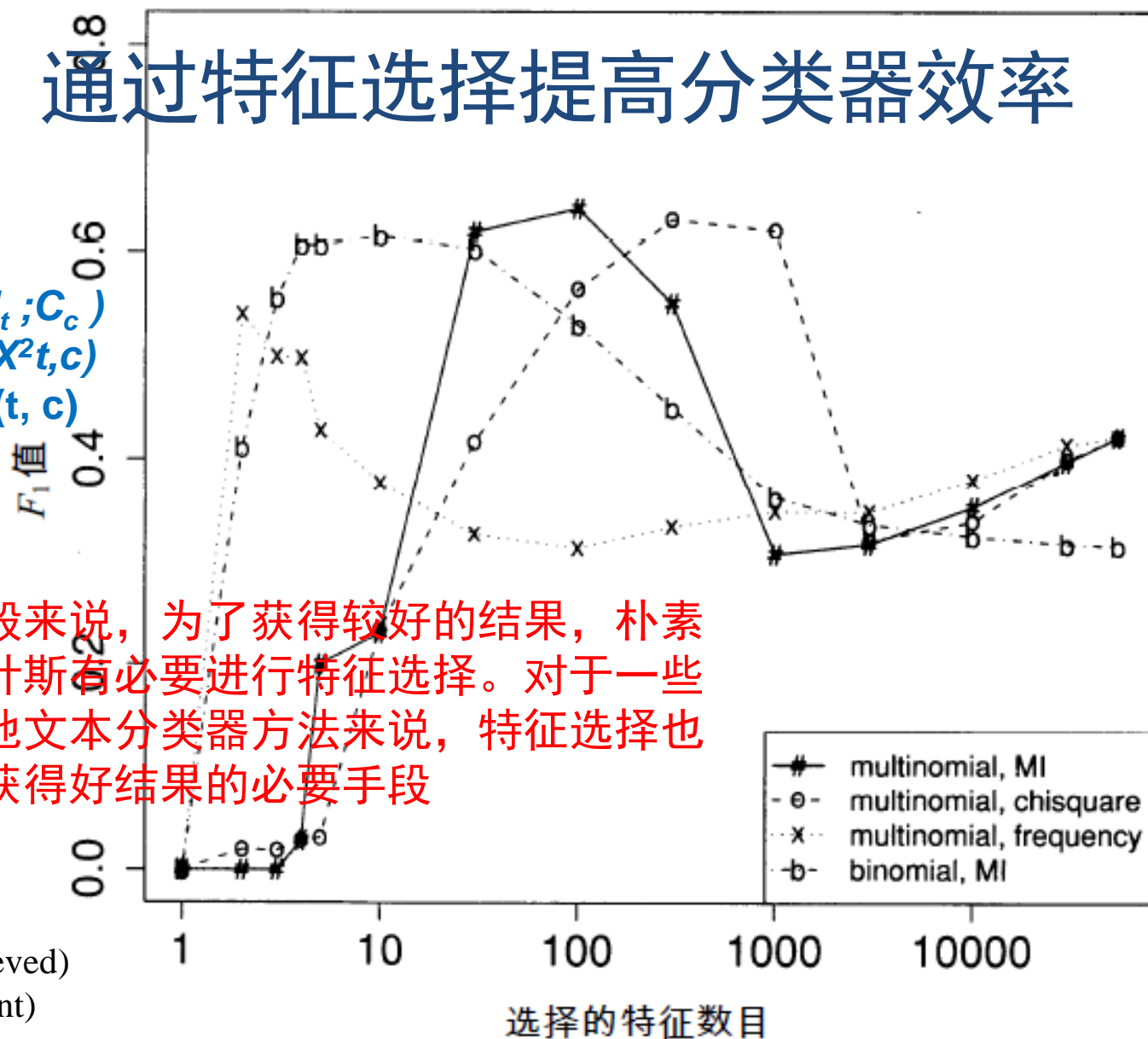
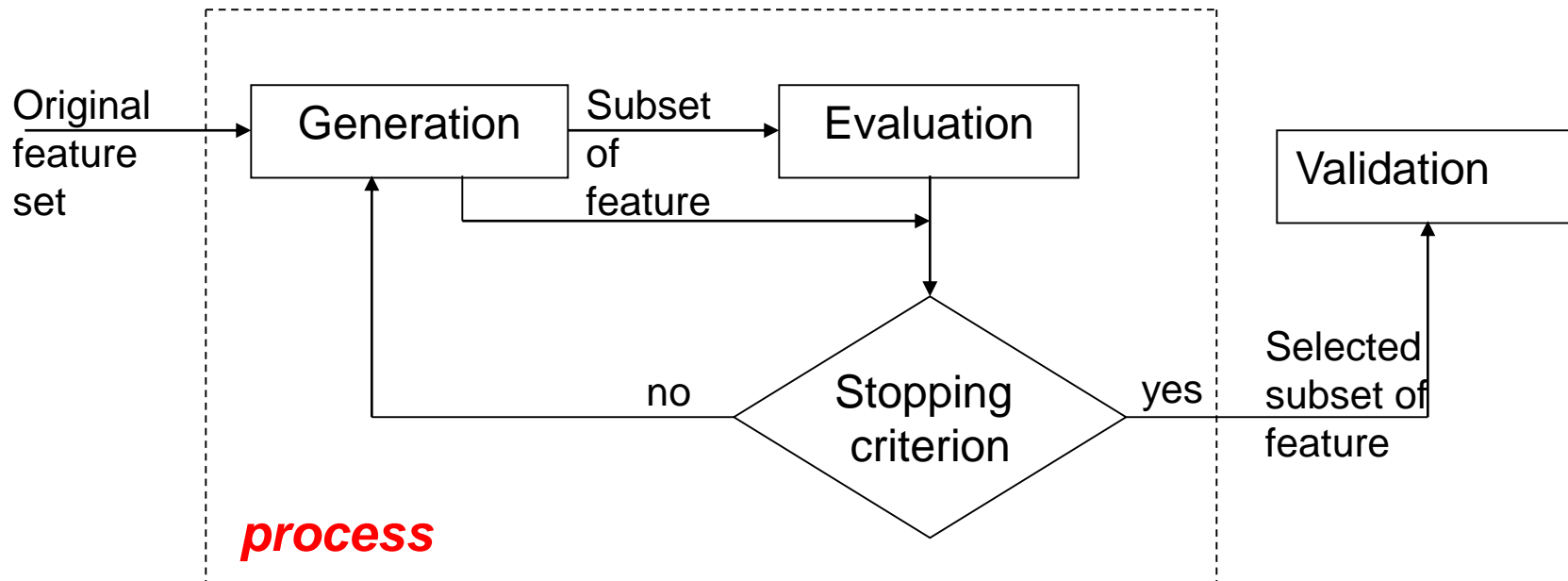


图 13-8 不同特征数目下多项式模型和贝努利模型分类效果

回顾3-2：通过特征选择提高分类器效率

特征选择的主要步骤



Generation = select feature subset candidate.

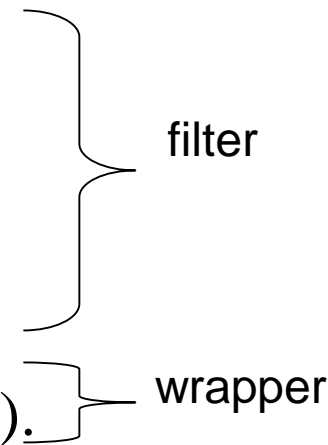
Evaluation = compute relevancy value of the subset.

Stopping criterion = determine whether subset is relevant.

Validation = verify subset validity.

回顾3-3：通过特征选择提高分类器效率

Evaluation

- **determine the relevancy of the generated feature subset candidate towards the classification task.**
 - **5 main type of evaluation functions.**
 - (8.1) distance (euclidean distance measure).
 - (8.2) information (entropy, information gain, etc.)
 - (8.3) dependency (correlation coefficient).
 - (8.4) **consistency** (min-features bias).
 - (8.5) **classifier error rate** (the classifier themselves).
- 

Filter: 特征选择算法独立于学习算法

Wrapper: 特征选择算法依赖于学习算法

讨论3-1: 文本分类的形式化定义

- 训练集 (**training set**)
- \mathbb{D} 是 $\langle d, c \rangle$ 的集合, $\langle d, c \rangle \in \mathbb{X} \times \mathbb{C}$
 $\langle d, c \rangle = \langle \text{Beijing joins the World Trade Organization}, \text{China} \rangle$
- 学习方法 (**learning method**) $\Gamma(\mathbb{D}) = \gamma$
- 分类器 (**classification function**) $\gamma : \mathbb{X} \rightarrow \mathbb{C}$
- 测试集 (**test set**) 中某文档 $d \{ \textit{first private Chinese airline} \}$
 $\gamma(d) = \text{China}$

讨论3-2: 基于概率的文本分类

$$\begin{array}{l} \arg \max_{c \in \mathbb{C}} P(c|d) \\ \quad \downarrow \\ \arg \max_{c \in \mathbb{C}} \frac{P(d|c)P(c)}{P(d)} \\ \quad \downarrow \\ \arg \max_{c \in \mathbb{C}} P(d|c)P(c) \\ \quad \downarrow \\ \arg \max_{c \in \mathbb{C}} P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \end{array}$$

$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$

$P(d)$ 对于任何 c 的取值相同忽略

独立性假设
 $P(d/c) = P(\langle t_1, \dots, t_k, \dots, t_{n_d} \rangle / c)$

训练:根据训练集学习(估计)出 $P(c)$ 和 $P(t_k/c)$

分类:根据测试文档中的词条 $\langle t_1, \dots, t_k, \dots, t_{n_d} \rangle$ 计算 $P(c/d)$

讨论3-3-1：分类器（分类函数）

- 学习方法不同，得到的分类函数 γ 不同
- 若学习方法固定，训练集 \mathcal{D} 不同， γ 是否相同？

$$\gamma(d) = \arg \max_{c \in \mathbb{C}} P(c) \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$$

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} \hat{P}(c | d) = \arg \max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k | c)$$

$$\hat{P}(c) = \frac{N_c}{N} \quad \hat{P}(t | c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- 训练集 \mathcal{D} 改变则 $\hat{P}(c)$ 和 $\hat{P}(t_k/c)$ 改变 $\rightarrow \gamma$ 改变

对于某固定学习方法，训练集改变使分类函数变化，不同的分类函数产生的决策结果如果基本一致，我们说该学习方法的方差不大，如果不同分类函数的决策结果差异性很大，我们说该学习方法的方差大

讨论3-3-2：分类器（分类函数）的误差

- 实际情况是 $P(c/d)$
- 学习后分类器输出 $\hat{P}(c/d)$
- $\hat{P}(c/d)$ 与 $P(c/d)$ 之间的差异就是误差

14.6章节, p216

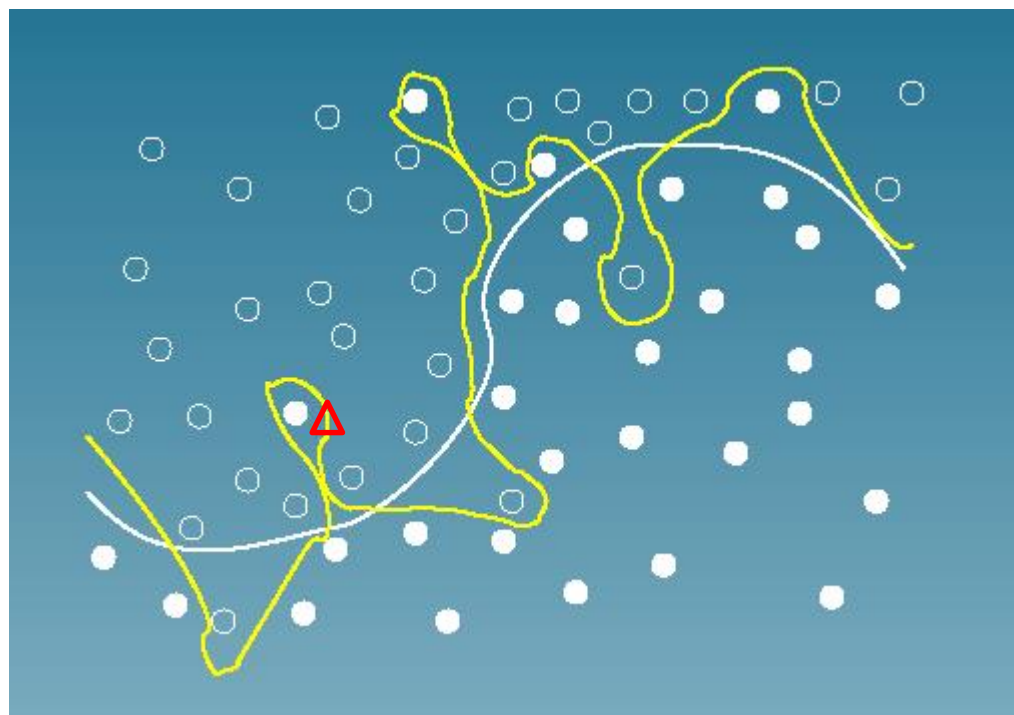
- 误差常用MSE衡量, MSE: $E_d[\hat{P}(c/d) - P(c/d)]^2$
- $E[x - \alpha]^2 = Ex^2 - 2Ex\alpha + \alpha^2 = [Ex - \alpha]^2 + E[x - Ex]^2$
- 令 $x = \hat{P}(c/d)$ 记为 \hat{P} , $\alpha = P(c/d)$ 记为 P
- $E[\hat{P} - P]^2 = [E\hat{P} - P]^2 + E[\hat{P} - E\hat{P}]^2$
- 偏差 *bias*、方差 *variance*

讨论3-3-3：偏差—方差折中准则

- **学习误差 = 偏差 + 方差**。通常情况下，这两个部分不会同时最小。当我们比较两个学习方法 Γ_1 和 Γ_2 时，大部分情况下最后的结果都是，其中一个方法**偏差高方差低**而另一个方法**偏差低方差高**。因此，从两个学习方法中选择一个时，我们不是简单地选择能够在不同训练集上产生好的分类器的学习方法（方差小），也不是选择那些能学出复杂决策边界的学习方法（偏差小）。实际的做法是，根据应用的需要，选择不同的权重对偏差和方差进行加权求和。这种折衷称为**偏差-方差折衷准则**（bias-variance tradeoff）。

讨论3-3-4: “偏差—方差” 示例

白色分类边界：偏差大
（一直存在错分）；但方差小
（不怎么受零星出现在某一类别中的另一类别文档的影响）



黄色分类边界：偏差小，但是方差大（大部分情况下正确，但如果有点出现在三角形所示位置，容易出现错分。故总体判决表现为时好时坏）

课程内容

- 第1章 绪论
- 第2章 布尔检索及倒排索引
- 第3章 词项词典和倒排记录表
- 第4章 索引构建和索引压缩
- 第5章 向量模型及检索系统
- 第6章 检索的评价
- 第7章 相关反馈和查询扩展
- 第8章 概率模型
- 第9章 基于语言建模的检索模型
- **第10章 文本分类**
 - 文本分类及朴素贝叶斯方法
 - 基于向量空间的文本分类
 - 支持向量机及机器学习方法
- 第11章 文本聚类
- 第12章 Web搜索
- 第13章 多媒体信息检索
- 第14章 其他应用简介

Information Retrieval(IR): 从大规模非结构化数据（通常是文本）的集合（通常保存在计算机上）中找出满足用户信息需求的资料（通常是文档）的过程

数据挖掘（Data Mining）从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程

本讲内容：基于向量空间的文本分类

• 第10章 文本分类

- 文本分类及朴素贝叶斯方法
- 基于向量空间的文本分类
 - Rocchio方法
 - kNN (k 近邻) 方法
 - 线性分类器
- 支持向量机及机器学习方法

多项式模型: $\langle t_1, \dots, t_{nd} \rangle$ 是在 d 中出现的词项序列

贝努利模型: $\langle e_1, \dots, e_M \rangle$ 是一个 M 维的布尔向量

向量空间模型: 每个词项对应一个维度 (分量)

基于向量空间模型的文本分类的思路

• 向量空间模型

长度归一化的欧式距离计算与余弦相似度计算结果是一致的

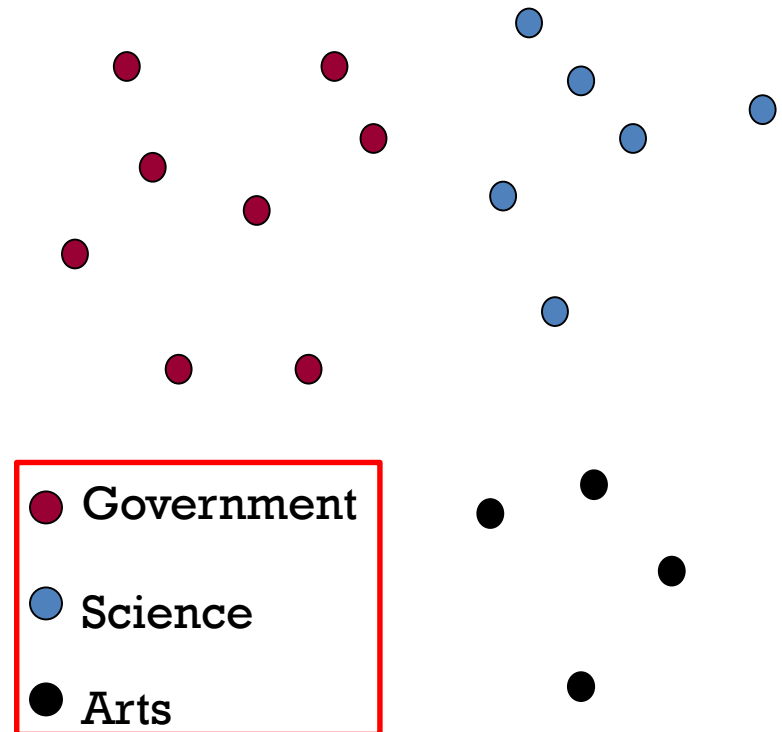
- 词项-文档矩阵：二值→计数→权重矩阵（**tf-idf值**）
- 相关性=向量距离：欧氏距离→夹角→余弦相似度

利用向量空间模型进行文本分类的思路主要基于邻近假设

（contiguity hypothesis）：

- ①**同一类的文档会构成一个邻近区域**， ②而**不同类的邻近区域之间是互不重叠的**。

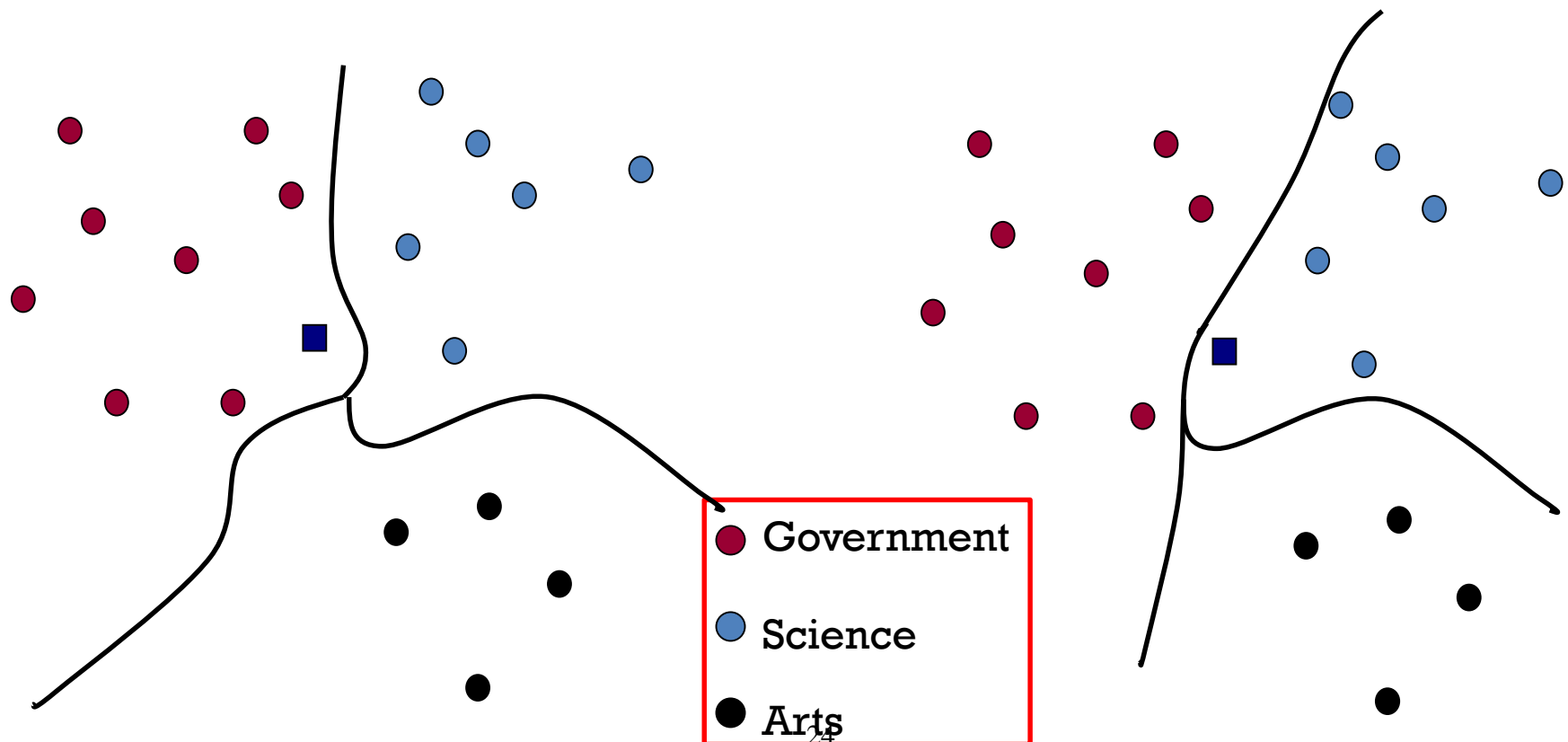
核心问题是如何找到分类面决策边界（decision boundary）



Test Document = Government?

Test Document = Science?

- 给定训练集可能存在多种分类面方案
- 选定的分类面方案有可能将测试文档归入错误的类中

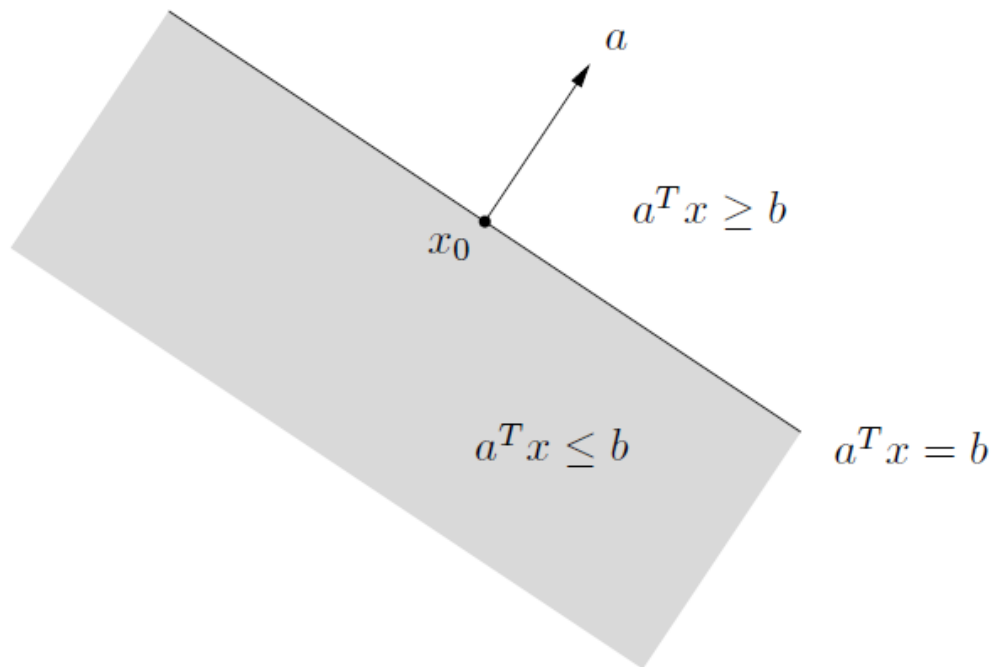


例：可用超平面来分割多维空间

- A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} = b\}$, where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} \neq 0$, and $b \in \mathbb{R}$.
- Geometrically, the hyperplane $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} = b\}$ can be interpreted as the set of points with a constant inner product to a given vector \mathbf{a} , or as a hyperplane with *normal vector* \mathbf{a} ; the constant $b \in \mathbb{R}$ determines the *offset* of the hyperplane from the origin.
- A hyperplane divides \mathbb{R}^n into **two halfspaces**. A (closed) halfspace is a set of the form $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} \leq b\}$,

例：可用超平面来分割多维空间

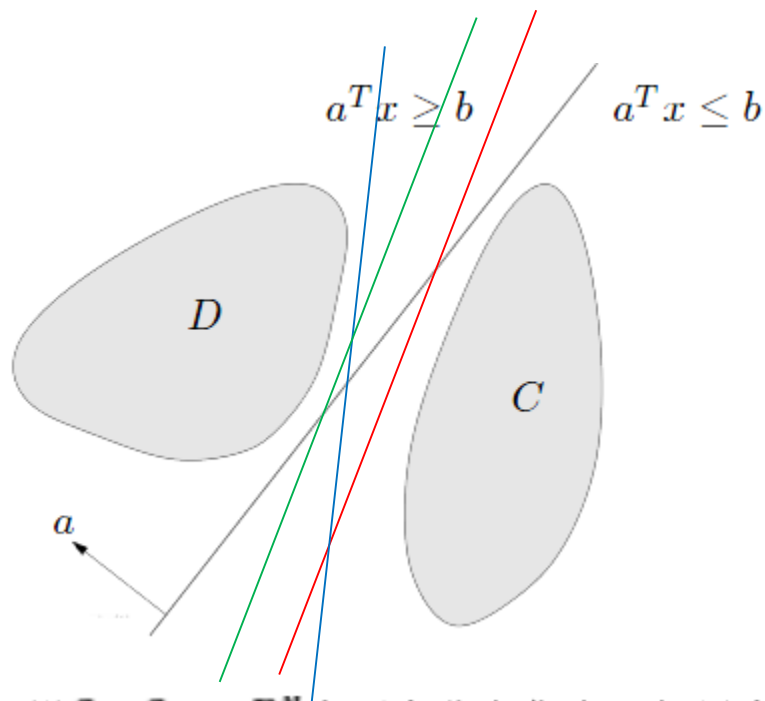
- A hyperplane divides \mathbb{R}^n into **two halfspaces**. A (closed) halfspace is a set of the form $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} \leq b\}$,



超平面只能将
空间分成两类

$n=2 \rightarrow$ 直线、 $n=3 \rightarrow$ 平面、 $n>3 \rightarrow$ 超平面

例：可用超平面来分割多维空间



凸集分离定理（超平面分离定理）是应用凸集到最优化理论中的重要结果，这个结果在最优化理论中有重要的位置。所谓两个凸集分离，直观地看是指两个凸集合没有交叉和重合的部分，因此可以用一张超平面将两者隔在两边。

存在多个这样的超平面

设 $S_1, S_2 \subseteq R^n$ 为两个非空集合，如果存在非零向量 $p \in R^n$ 及 $\alpha \in R$ 使得

$$S_1 \subseteq H^- = \{x \in R^n | p^T x \leq \alpha\}$$

$$S_2 \subseteq H^+ = \{x \in R^n | p^T x \geq \alpha\}$$

则称超平面 $H = \{x \in R^n | p^T x = \alpha\}$ 分离了集合 S_1 与 S_2 。

小结：基于向量空间的分类

- 邻近假设 (**contiguity hypothesis**)

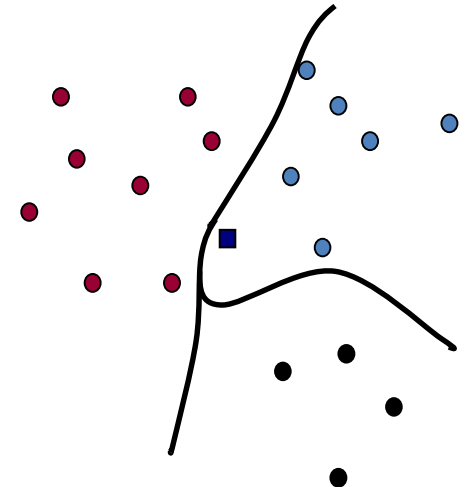
- ①同一类的文档会构成一个邻近区域，
- ②而不同类的邻近区域之间是互不重叠的

文档集是否会映射成邻近区域取决于在文档表示中的很多选项，例如权重计算方法、停用词表等。

- 核心问题是如何找到分类面

- 决策边界 (**decision boundary**)

- *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} = b\}$
- $n=2 \rightarrow$ 直线、 $n=3 \rightarrow$ 平面、 $n>3 \rightarrow$ 超平面



本讲内容：基于向量空间的文本分类

- 第10章 文本分类

- 文本分类及朴素贝叶斯方法
- 基于向量空间的文本分类
 - Rocchio方法
 - kNN (k 近邻) 方法
 - 线性分类器
- 支持向量机及机器学习方法

回顾：相关反馈(Relevance feedback)

Rocchio算法

教材9.1.1章节，p122

- 我们将文档看作高维空间中的点，质心是一堆点的质量的中心

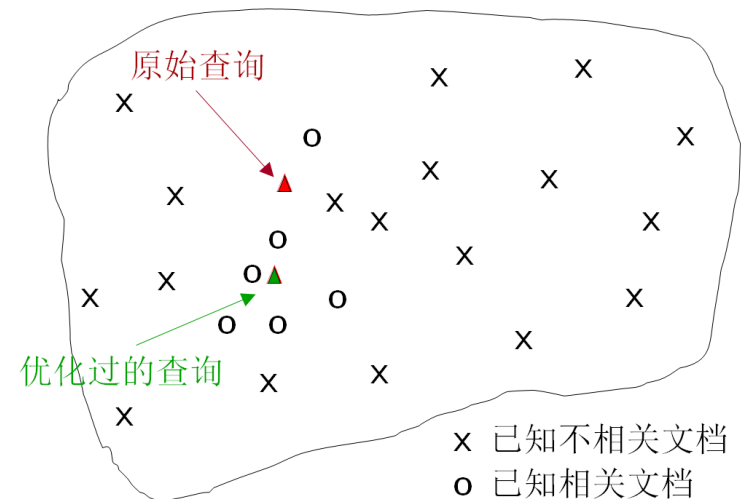
$$\vec{\mu}(C) = \frac{1}{|C|} \sum_{d \in C} \vec{d}$$

- Rocchio** 算法试图寻找一个查询 $\rightarrow q_{opt}$ ，使得：

$$\vec{q}_{opt} = \arg \max_{\vec{q}} [\cos(\vec{q}, \vec{\mu}(C_r)) - \cos(\vec{q}, \vec{\mu}(C_{nr}))]$$

- Rocchio 1971 算法 (SMART)**

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$



Rocchio方法进行向量空间分类的思路

- 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（Rocchio classification）方法**
 - 利用质心（centroid）来定义分类边界。一个类别 c 的质心可以通过类中文档向量的平均向量或者质心向量来计算，即
$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$
 - 其中， D_c 是文档集 D 中属于类别 c 的文档子集： $D_c = \{d: \langle d, c \rangle \in D\}$ 。这里将归一化的文档向量记为 $\vec{v}(d)$

Rocchio算法

- (1)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
- (2)将每篇测试文档分到离它最近的那个中心向量

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      $\vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$ 
4  return  $\{\vec{\mu}_1, \dots, \vec{\mu}_J\}$ 

```

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

```

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

```

mode	time complexity
training	$\Theta(\mathbb{D} L_{\text{ave}} + \mathbb{C} V) \approx \Theta(\mathbb{D} L_{\text{ave}})$
testing	$\Theta(L_a + \mathbb{C} M_a) \approx \Theta(\mathbb{C} M_a)$

Rocchio算法的时间复杂度

与NB方法在训练上具有相同的时间复杂度

Rocchio分类示例

表13-1 用于参数估计的数据

	文档ID	文档中的词	属于c=China类?
训练集	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
测试集	5	Chinese Chinese Chinese Tokyo Japan	?

$$\{x \mid a^T x = b\}$$

$$a \approx (0 \ -0.71 \ -0.71 \ 1/3 \ 1/3 \ 1/3)^T$$

$$b = -1/3$$

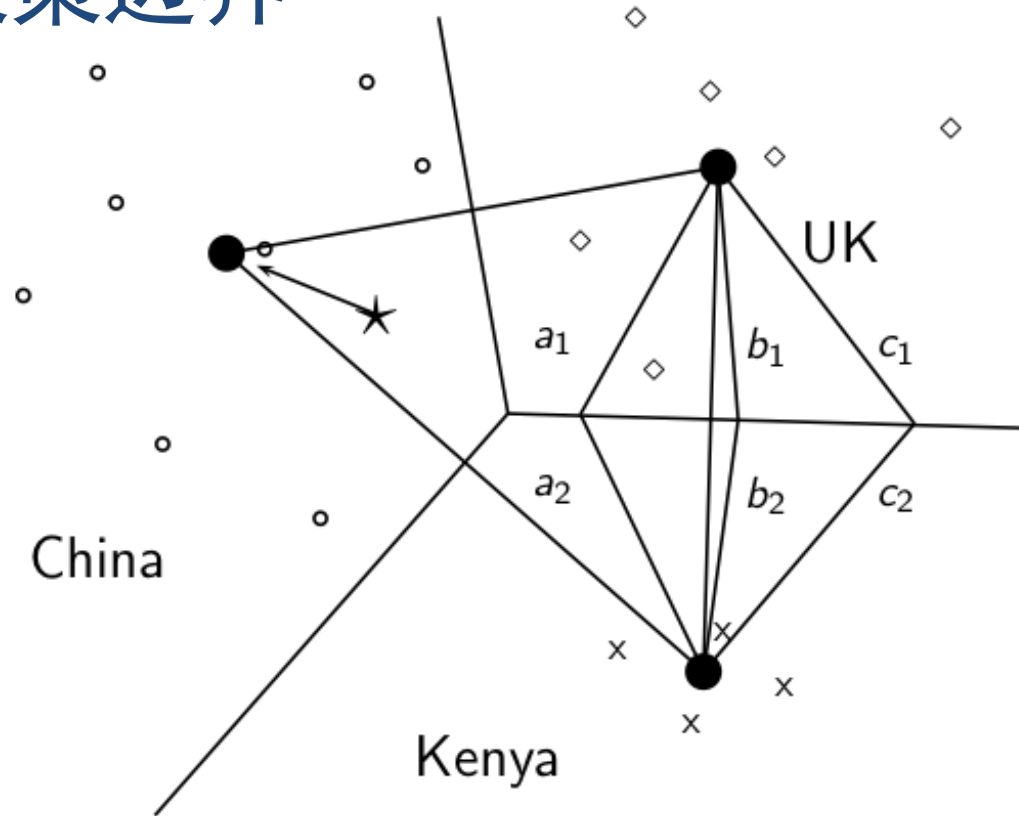
表14-1 表13-1中数据对应的文档向量及类别质心向量

向量	词项权重					
	Chinese	Japan	Tokyo	Macao	Beijing	Shanghai
\vec{d}_1	0	0	0	0	1.0	0
\vec{d}_2	0	0	0	0	0	1.0
\vec{d}_3	0	0	0	1.0	0	0
\vec{d}_4	0	0.71	0.71	0	0	0
\vec{d}_5	0	0.71	0.71	0	0	0
$\vec{\mu}_c$	0	0	0	0.33	0.33	0.33
$\vec{\mu}_{\bar{c}}$	0	0.71	0.71	0	0	0

$$|\vec{\mu}(c) - \vec{d}_5| \approx 1.15$$

$$|\vec{\mu}(\bar{c}) - \vec{d}_5| = 0.0$$

Rocchio算法中的决策边界



Rocchio 分类方法利用质心（centroid）来定义分类边界。两类的边界由那些到两个类质心等距的点集组成（超平面）。如图有 $|a_1| = |a_2|$ 、 $|b_1| = |b_2|$ 和 $|c_1| = |c_2|$ 。二维平面上的一条直线在M维空间中可以推广成一个超平面。

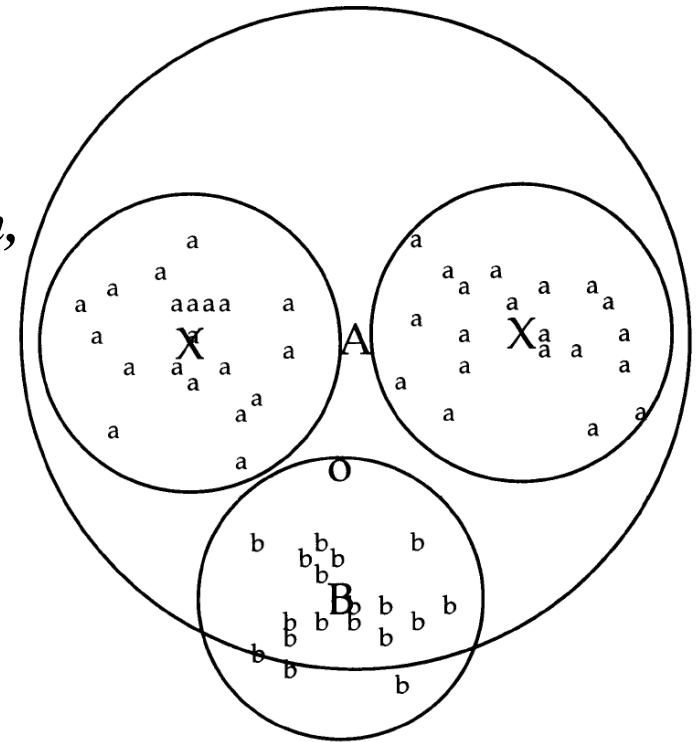
Rocchio分类方法的缺陷

A (Euclidean) ball (or just ball) in R^n has the form

$$B(x_c, r) = \{x \mid \|x - x_c\|_2 \leq r\}$$

where $r > 0$, and $\|\cdot\|_2$ denotes the Euclidean norm, i.e., $\|u\|_2 = (u^T u)^{1/2}$. The vector x_c is the center of the ball and the scalar r is its radius; $B(x_c, r)$ consists of all points within a distance r of the center x_c .

为了遵循邻近性的要求，Rocchio
分类中的每个类别一定要近似球形，
并且它们之间具有相似的球半径。



多模态类别“a”由两个不同簇（分别是以X为中心的
两个小圆）组成。由于“O”更接近“a”的中心A，
因此，Rocchio分类会将其错分到“a”类

小结： Rocchio分类方法

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

- 算法步骤

- (1)计算每个类的中心向量
- (2)将每篇测试文档分到离它最近的那个中心向量

- 特性

- Rocchio 分类方法类的边界由那些到两个类质心等距的点集组成（超平面）。
- Rocchio 分类中的每个类别一定要近似球形，并且它们之间具有相似的球半径。当某类的内部文档并不近似分布在半径相近的球体之内时，其分类精度并不高。
- Rocchio算法的时间复杂度与NB方法在训练上具有相同的时间复杂度

本讲内容：基于向量空间的文本分类

- 第10章 文本分类

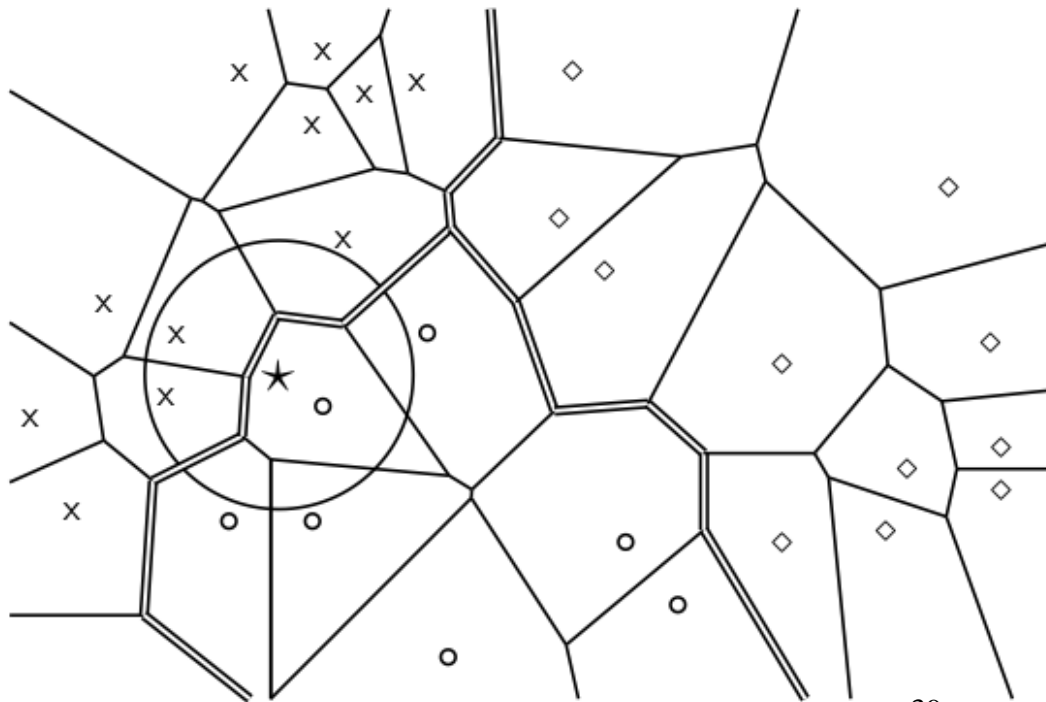
- 文本分类及朴素贝叶斯方法
- 基于向量空间的文本分类
 - Rocchio方法
 - **kNN (k 近邻) 方法**
 - 线性分类器
- 支持向量机及机器学习方法

kNN (k 近邻) 方法

- **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 的基本依据**
 - 根据邻近假设, 一篇测试文档d 将和其邻域中的训练文档应该具有相同的类别。

1NN分类器

- 1NN分类器的判别边界是**Voronoi剖分**（Voronoi tessellation）形成的多个线段的连接。Voronoi剖分会将整个平面分成 $|D|$ 个凸多边形，**每个多边形都仅包含其对应的文档**，而每个凸多边形是在二维空间中通过直线围成的**凸**区域。

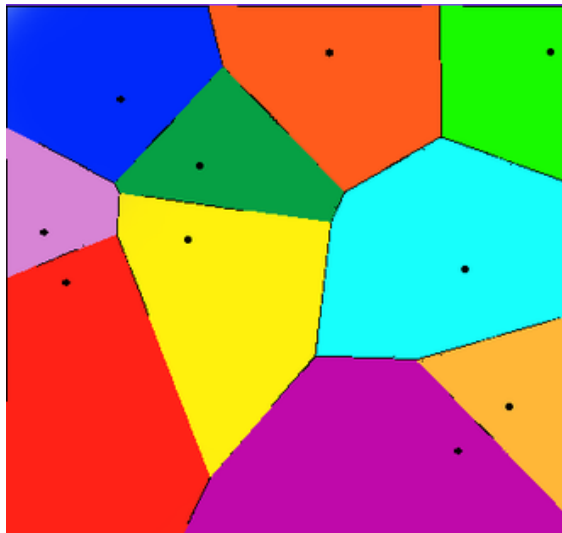


1NN 分类中的Voronoi 剖分及分类边界(双线表示)。3 个类别分别采用x、圆圈和菱形表示

Voronoi图

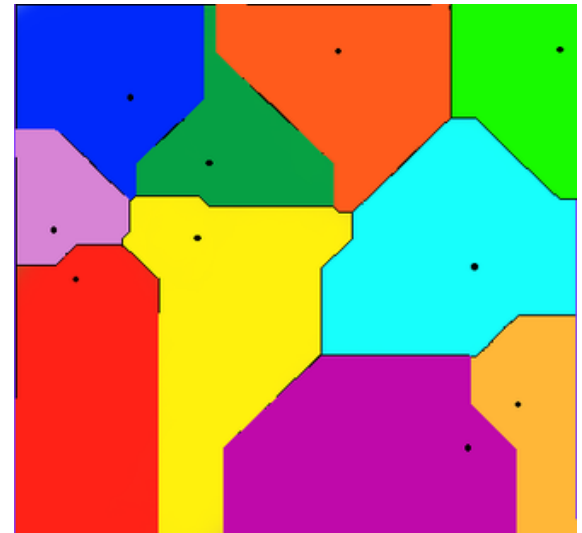
(俄国数学家M.G.Voronoi于1908年发现)

- 对平面 n 个离散点而言，**V图**把平面分为几个区，每一个区包括一个点，该点所在的区是到该点距离最近点的集合。
- 设 P 是一离散点集合 $P_1, P_2, \dots, P_n \in P$ ，定义 P_i 的Voronoi区域 **$V(P_i)$** 为所有到 P_i 距离最小点的集合
- $V(P_i) = \{P \mid d(P, P_i) \leq d(P, P_j), j \neq i, j = 1, 2, \dots, n\}$



← 10 shops in a flat city
and their Voronoi cells
(**euclidean distance**)

The same 10 shops, now
under the **Manhattan**
distance. →

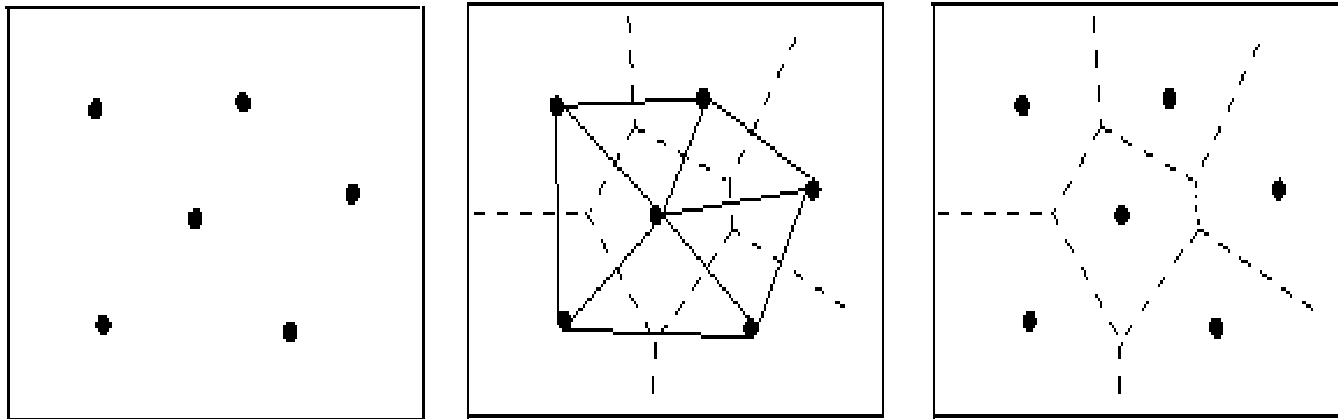


$$d((a_1, a_2), (b_1, b_2)) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

$$d((a_1, a_2), (b_1, b_2)) = |a_1 - b_1| + |a_2 - b_2|$$

V图生成方法

- 生成V图的方法很多，如矢量方法（对偶生成法、增添法、部件合成法）、栅格方法（数学形态学距离变换法、地图代数距离变换法）等。
- **对偶生成法**：生成V图时先生成其对偶元**Delaunay三角网**，再做三角网每一三角形三条边的**中垂线**，形成以每一三角形顶点为生成元的多边形网。



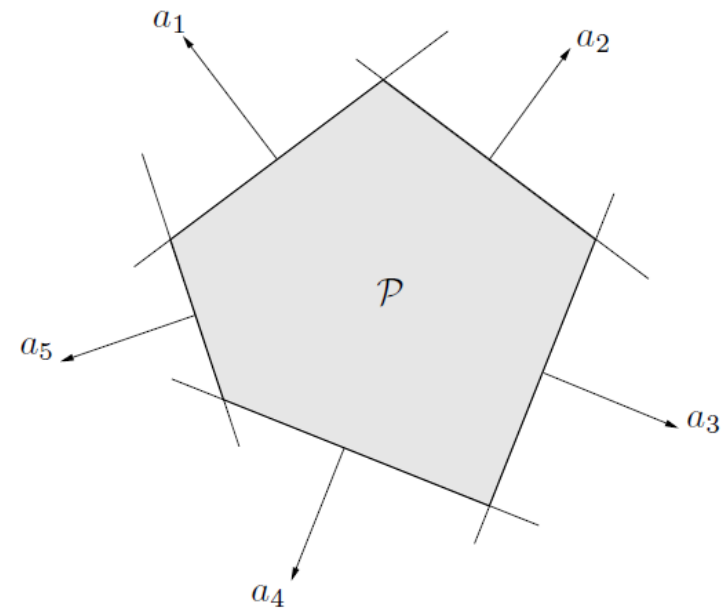
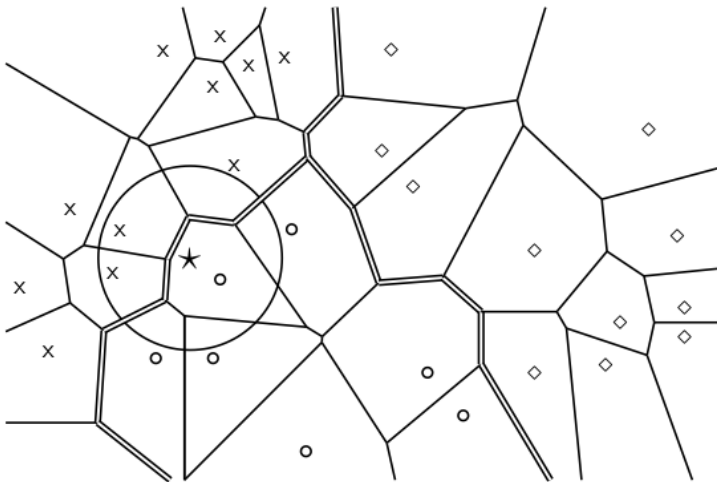
Voronoi剖分 二维→多维

A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{a}^T \mathbf{x} = \mathbf{b}\}$

A polyhedron is defined as the solution set of a finite number of linear equalities and inequalities:

$$\mathcal{P} = \{x \mid a_j^T x \leq b_j, j = 1, \dots, m, c_j^T x = d_j, j = 1, \dots, p\}$$

A polyhedron is thus the intersection of a finite number of halfspaces and hyperplanes.



多边形扩展到高一维空间就是多面体

Voronoi sets and polyhedral decomposition

2.9 Voronoi sets and polyhedral decomposition. Let $x_0, \dots, x_K \in \mathbf{R}^n$. Consider the set of points that are closer (in Euclidean norm) to x_0 than the other x_i , i.e.,

$$V = \{x \in \mathbf{R}^n \mid \|x - x_0\|_2 \leq \|x - x_i\|_2, i = 1, \dots, K\}.$$

V is called the *Voronoi region* around x_0 with respect to x_1, \dots, x_K .

- (a) Show that V is a polyhedron. Express V in the form $V = \{x \mid Ax \preceq b\}$.
- (b) Conversely, given a polyhedron P with nonempty interior, show how to find x_0, \dots, x_K so that the polyhedron is the Voronoi region of x_0 with respect to x_1, \dots, x_K .
- (c) We can also consider the sets

$$V_k = \{x \in \mathbf{R}^n \mid \|x - x_k\|_2 \leq \|x - x_i\|_2, i \neq k\}.$$

The set V_k consists of points in \mathbf{R}^n for which the closest point in the set $\{x_0, \dots, x_K\}$ is x_k .

The sets V_0, \dots, V_K give a polyhedral decomposition of \mathbf{R}^n . More precisely, the sets V_k are polyhedra, $\bigcup_{k=0}^K V_k = \mathbf{R}^n$, and $\text{int } V_i \cap \text{int } V_j = \emptyset$ for $i \neq j$, i.e., V_i and V_j intersect at most along a boundary.

Suppose that P_1, \dots, P_m are polyhedra such that $\bigcup_{i=1}^m P_i = \mathbf{R}^n$, and $\text{int } P_i \cap \text{int } P_j = \emptyset$ for $i \neq j$. Can this polyhedral decomposition of \mathbf{R}^n be described as the Voronoi regions generated by an appropriate set of points?

Source: 《Convex Optimization》, Stephen Boyd

kNN思路的改进

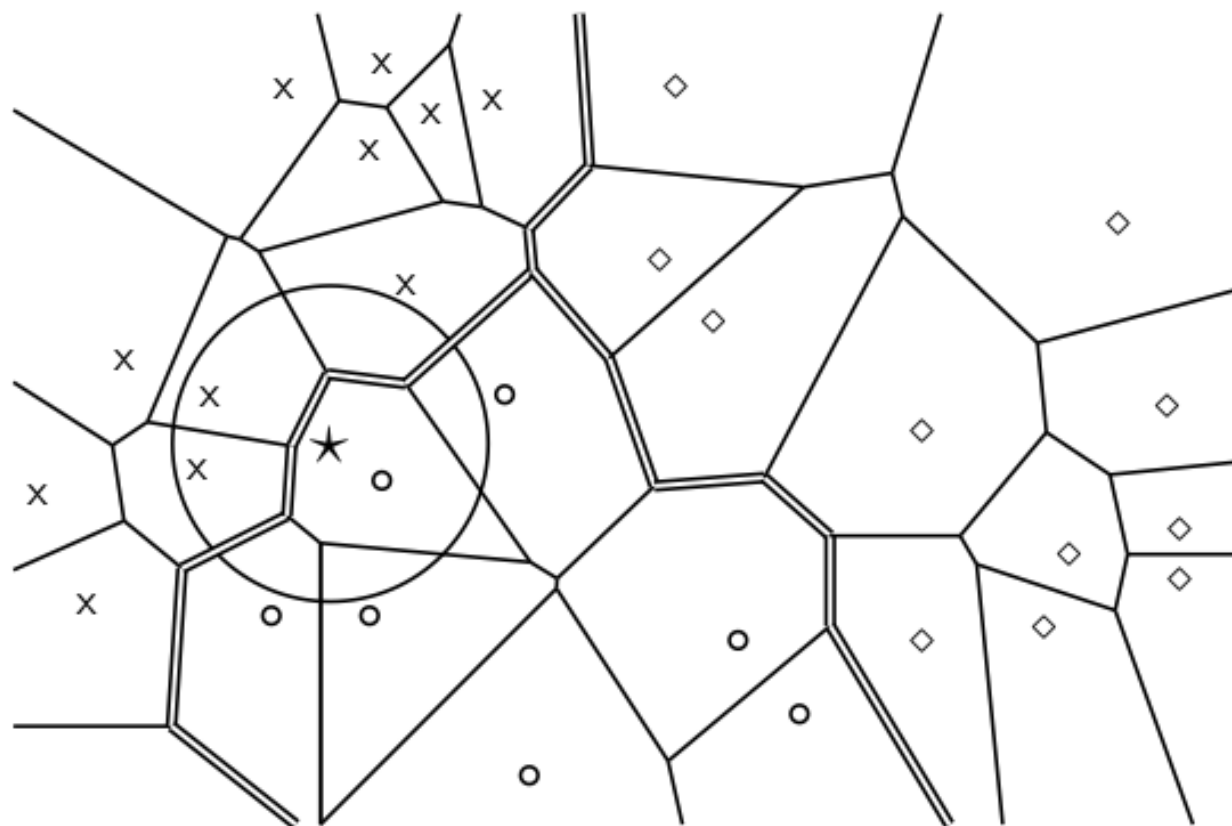
- **【改进1】kNN的概率型版本**：将属于类别c 的概率估计为k 个近邻中属于类别c 的文档比例
 - $P(c|d)$ = d的最近的k个邻居中属于c类的比例
 - 将d分到具有最高概率 $P(c|d)$ 的类别c中
- **【改进2】**也可以将k 个近邻基于其余弦相似度进行加权。这种情况下，文档d 属于某个类别c的得分计算如下

$$\text{score}(c, d) = \sum_{d' \in S_k} I_c(d') \cos(\vec{v}(d'), \vec{v}(d))$$

- 其中， S_k 表示的是文档d ‘的k 个近邻文档组成的集合，如果d ‘属于类别c 则 $I_c(d ‘)=1$ ，否则 $I_c(d ‘)=0$ 。最后将得分最高的类别赋予文档d ‘。

kNN示例1

- 对于★ 对应的文档，在1NN和 3NN下，分别应该属于哪个类？



kNN 算法

- 对于 $k \in \mathbb{N}$ 的一般kNN分类来说，考虑k个最近邻的区域的方法同前面一样。这里会再次得到一个凸多边形，整个空间也会划分为多个凸多边形，每个凸多边形中的k个近邻组成的集合是不变的

TRAIN-KNN(\mathbb{C}, \mathbb{D})

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

APPLY-KNN($\mathbb{C}, \mathbb{D}', k, d$)

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

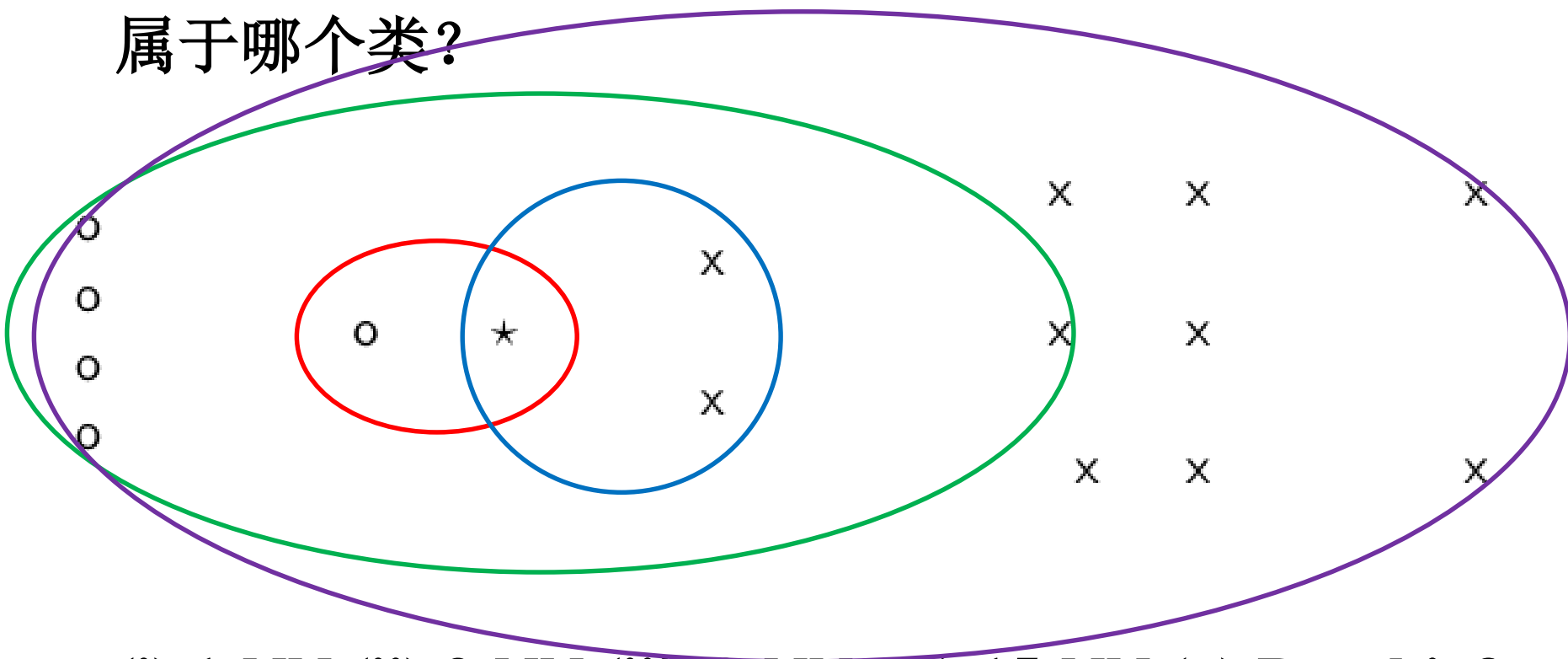
kNN 算法的流程：

kNN 的训练（包括预处理）和分类过程。

S_k 表示的是文档d的k个近邻文档组成的集合
 p_j 是概率 $P(c_j|S_k) = P(c_j|d)$ 的估计值，
 c_j 表示的是类别 c_j 中的所有文档

kNN示例2

- 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的时间复杂度

kNN with preprocessing of training set

training $\Theta(|\mathcal{D}|L_{ave})$

testing $\Theta(L_a + |\mathcal{D}|M_{ave}M_a) = \Theta(|\mathcal{D}|M_{ave}M_a)$

kNN without preprocessing of training set

training $\Theta(1)$

testing $\Theta(L_a + |\mathcal{D}|L_{ave}M_a) = \Theta(|\mathcal{D}|L_{ave}M_a)$

kNN 分类器的训练和测试时间复杂度， M_{ave} 是文档集中每篇文档的平均词汇量大小（即平均词项个数）， L_{ave} 是文档的平均长度， L_a 和 M_a 分别是测试文档中词条及词条类（即不同词项）的数目

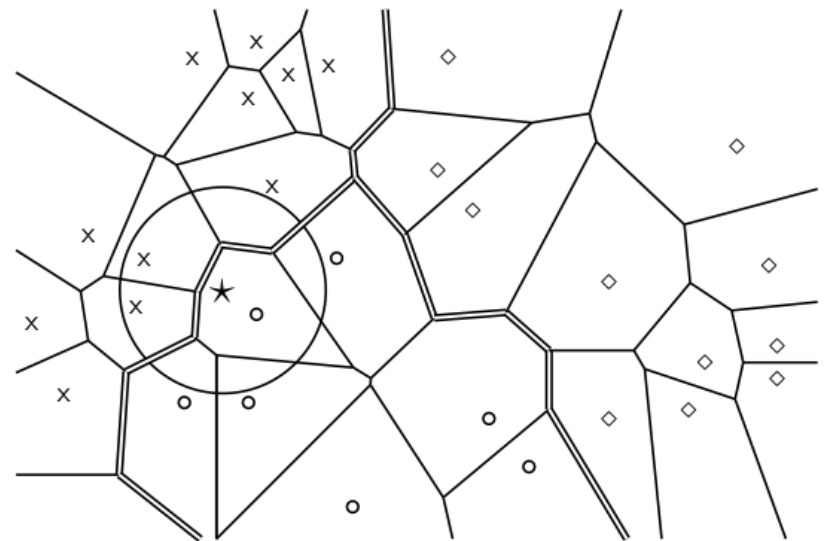
•不需要训练过程

- 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。

小结：kNN（k 近邻）方法

- 思路：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 的基本依据：根据邻近假设，一篇测试文档d将和其邻域中的训练文档应该具有相同的类别。

- 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。



本讲内容：基于向量空间的文本分类

• 第10章 文本分类

- 文本分类及朴素贝叶斯方法
- 基于向量空间的文本分类
 - Rocchio方法
 - kNN (k 近邻) 方法
 - 线性分类器
 - 常见的线性/非线性分类器
 - 非线性的分类问题
 - 多类别的分类问题
- 支持向量机及机器学习方法

线性分类器

- 定义:

- 线性分类器计算特征值的一个线性加权和 $\sum_i w_i x_i$
- 决策规则: $\sum_i w_i x_i > \theta?$ 其中, θ 是一个参数

- 考虑二元分类器

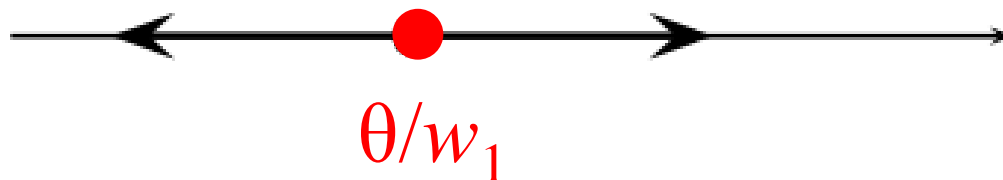
- 从几何上说, 二元分类器相当于二维平面上的一条直线、三维空间中的一个平面或者更高维下的超平面, 称为分类面

- 分类面

- 基于训练集来寻找该分类面
- 寻找分类面的方法: 感知机(Perceptron)、Rocchio, Naïve Bayes – 我们将解释为什么后两种方法也是二元分类器

一维下的线性分类器

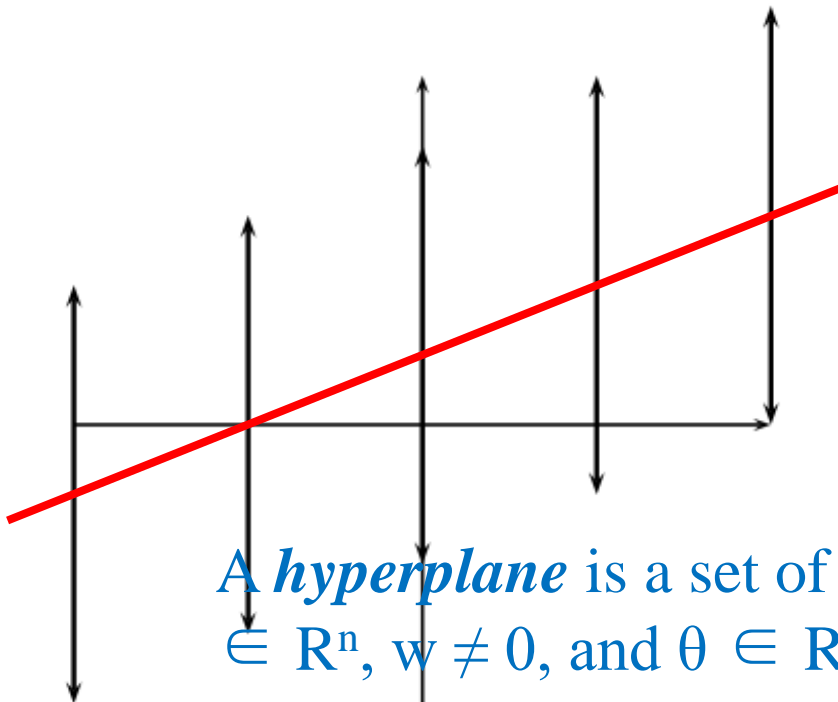
- 一维下的分类器是方程 $w_1 d_1 = \theta$ 对应的点
- 点的位置是 θ/w_1
- 那些满足 $w_1 d_1 \geq \theta$ 的点 d_1 属于类别 c
- 而那些 $w_1 d_1 < \theta$ 的点 d_1 属于类别 \bar{c} .



A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{w}^T \mathbf{x} = \theta\}$, where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w} \neq \mathbf{0}$, and $\theta \in \mathbb{R}$.

二维平面下的线性分类器

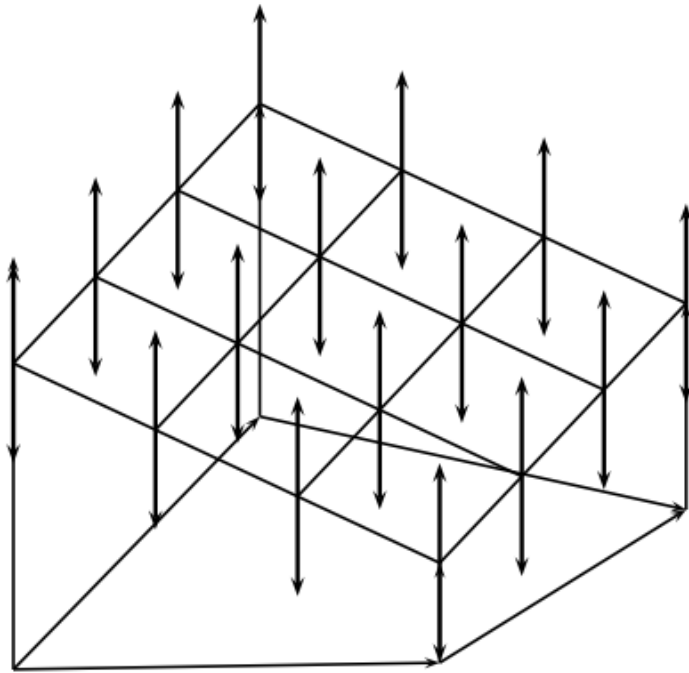
- 二维下的分类器是方程 $w_1d_1 + w_2d_2 = \theta$ 对应的直线
- 那些满足 $w_1d_1 + w_2d_2 \geq \theta$ 的点 (d_1, d_2) 属于类别 c
- 那些满足 $w_1d_1 + w_2d_2 < \theta$ 的点 (d_1, d_2) 属于类别 \bar{c} .



A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{w}^T \mathbf{x} = \theta\}$, where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w} \neq 0$, and $\theta \in \mathbb{R}$.

三维空间下的线性分类器

- 三维空间下分类器是方程 $w_1d_1 + w_2d_2 + w_3d_3 = \theta$ 对应的平面
- 那些满足 $w_1d_1 + w_2d_2 + w_3d_3 \geq \theta$ 的点 $(d_1 d_2 d_3)$ 属于类别 c
- 那些满足 $w_1d_1 + w_2d_2 + w_3d_3 < \theta$ 的点 $(d_1 d_2 d_3)$ 属于类别 \bar{c} .



A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{w}^T \mathbf{x} = \theta\}$, where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w} \neq 0$, and $\theta \in \mathbb{R}$.

Two-class Rocchio as a linear classifier

- **Line or hyperplane defined by:**

$$\sum_{i=1}^M w_i d_i = \theta$$

- **For Rocchio, set:**

$$\vec{w} = \vec{\mu}(c_1) - \vec{\mu}(c_2)$$

$$\theta = 0.5 \times (|\vec{\mu}(c_1)|^2 - |\vec{\mu}(c_2)|^2)$$

A *hyperplane* is a set of the form $\{\mathbf{x} \mid \mathbf{w}^T \mathbf{x} = \theta\}$, where $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w} \neq 0$, and $\theta \in \mathbb{R}$.

Naive Bayes is a linear classifier

- **Two-class Naive Bayes. We compute:**

$$\log \frac{P(C | d)}{P(\bar{C} | d)} = \log \frac{P(C)}{P(\bar{C})} + \sum_{w \in d} \log \frac{P(w | C)}{P(w | \bar{C})}$$

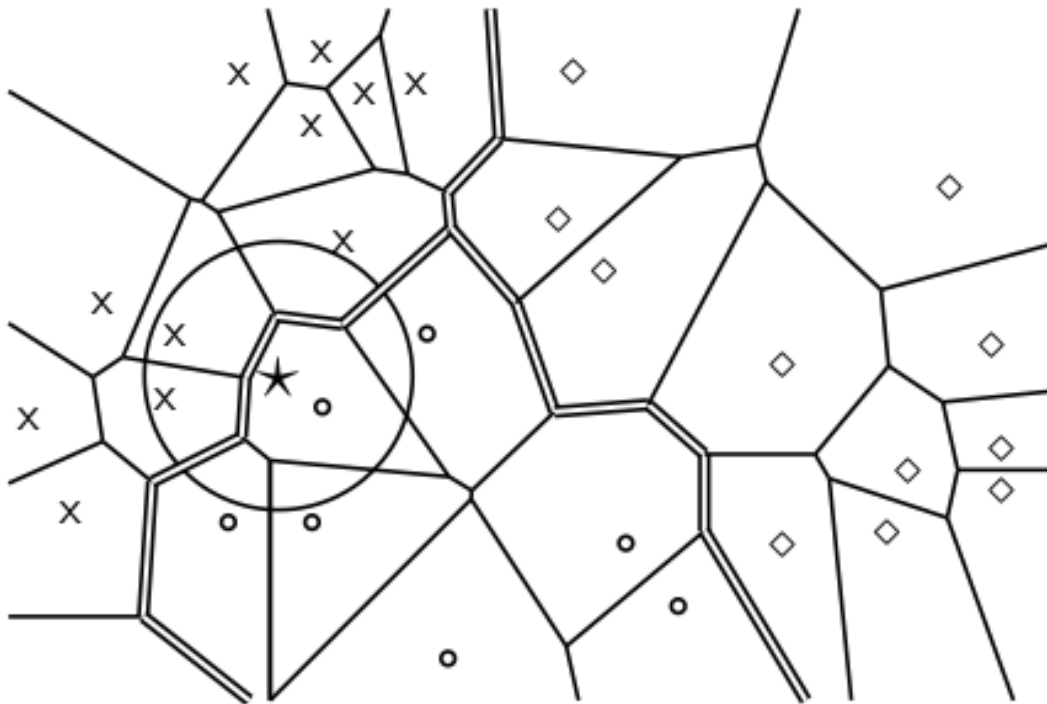
- **Decide class C if the odds is greater than 1, i.e., if the log odds is greater than 0.**
- **So decision boundary is hyperplane:**

$$\alpha + \sum_{w \in V} \beta_w \times n_w = 0 \quad \text{where } \alpha = \log \frac{P(C)}{P(\bar{C})};$$

$$\beta_w = \log \frac{P(w | C)}{P(w | \bar{C})}; \quad n_w = \# \text{ of occurrences of } w \text{ in } d$$

kNN不是线性分类器

- **kNN**分类决策取决于**k**个邻居类中的多数类
- 类别之间的分类面是分段线性的
- 但是一般来说，很难表示成如下的 线性分类器



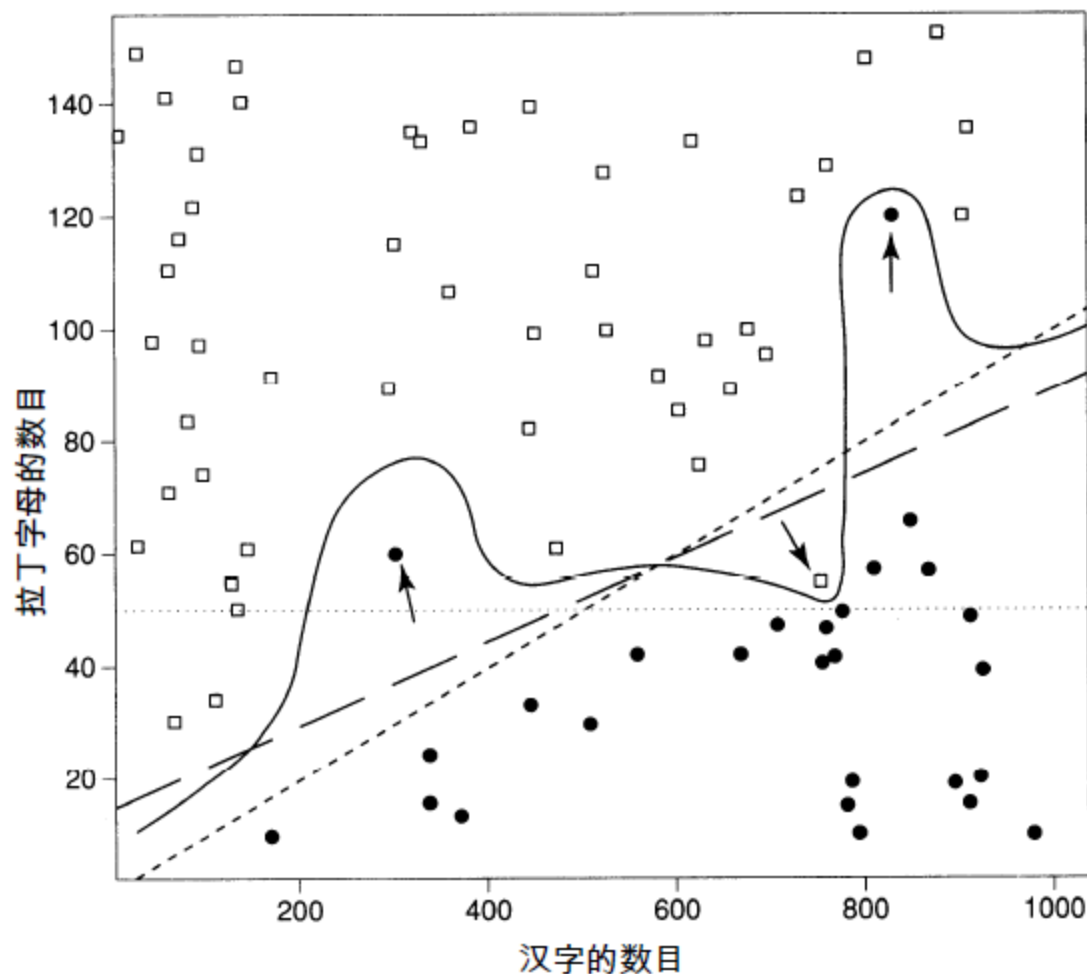
$$\sum_{i=1}^M w_i d_i = \theta.$$

线性分类器: 讨论

- 很多常用的文本分类器都是线性分类器：朴素贝叶斯、Rocchio、logistic回归、线性SVM等等
- 不同的方法在测试文档分类性能时存在巨大差异（分类面的选择不同）
- 能否通过更强大的非线性分类器来获得更好的分类性能？一般情况下不能，给定数量的训练集可能足以估计一个线性分类面，但是不足以估计一个更复杂的非线性分类面

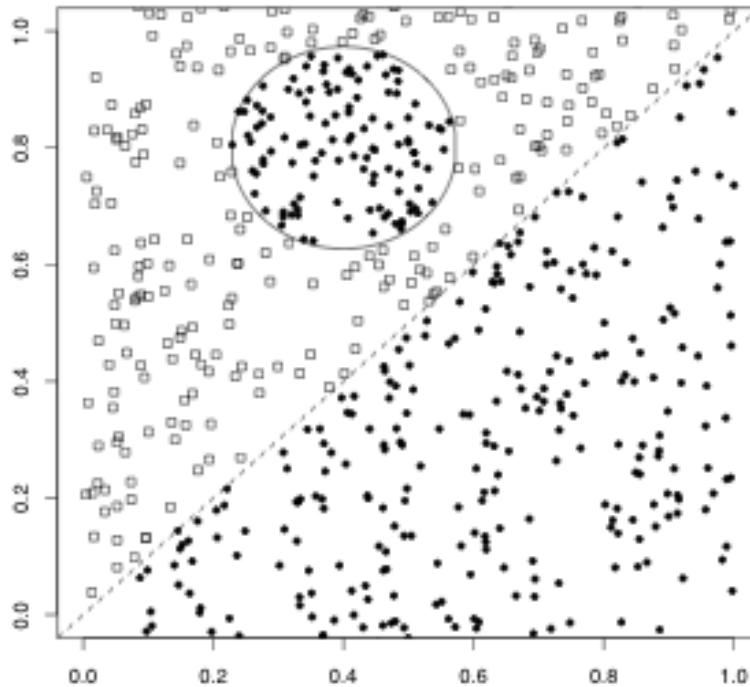
线性分类器训练困难的原因之一：噪音文档

一个带噪音的线性问题。
在这个假想的 Web 网页
分类下，仅包含中文的网
页用实心圆表示，而中英
文混合网页用小方块表示。
除了3 篇噪音文档（用箭
头标记）外，这两个类可
以被一个线性类别边界
（用短破折号虚线表示）
分开



非线性的分类问题

Linear / nonlinear classifiers



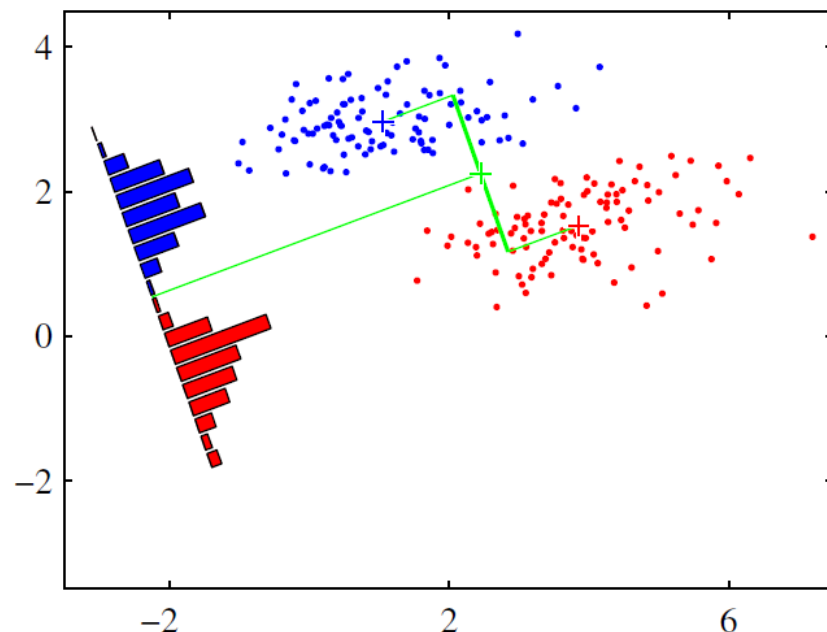
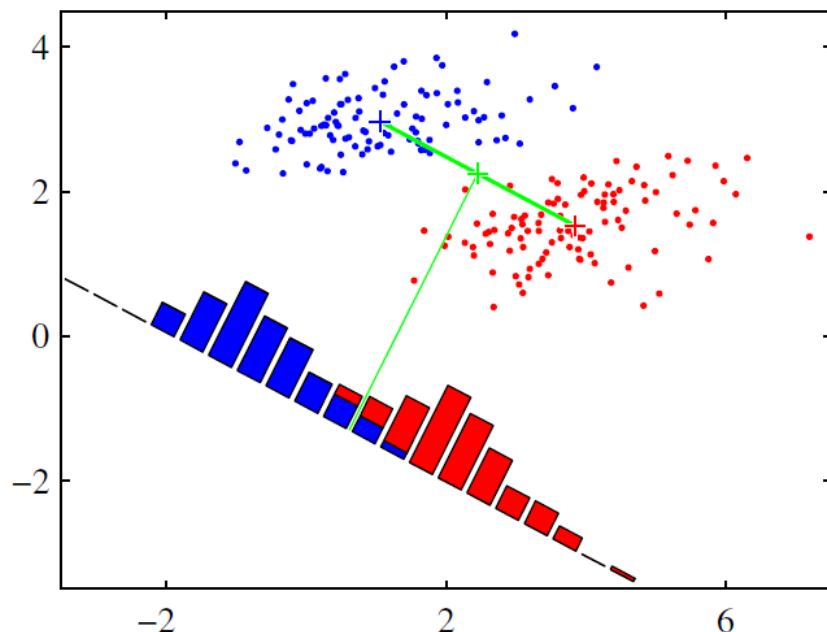
- 诸如Rocchio的线性分类器在处理上述问题时效果很差
- 在训练集规模充分时，kNN 可以获得好的效果

高维非线性分类→一维线性分类

Fisher's linear discriminant

Christopher M. Bishop

《Pattern Recognition and Machine Learning》 Chapter4



Fisher判别的基本思路就是投影。对P维空间中的某点 $\mathbf{x}=(x_1, x_2, \dots, x_p)$ 寻找一个能使它降为一维数值的线性函数 $y(\mathbf{x}) = \sum C_j x_j$ 。用 $y(\mathbf{x})$ 把P维空间中的样本都变换为一维数据，再根据其间的亲疏程度把未知归属的样本点判定其归属。 $y(\mathbf{x})$ 应该能够在把P维空间中的所有点转化为一维数值之后，既能最大限度地缩小同类中各个样本点之间的差异，又能最大限度地扩大不同类别中各个样本点之间的差异，这样才可能获得较高的判别效率。

Fisher's linear discriminant

最佳投影方向的求解

S_B is the between-class covariance matrix
 S_W is the total within-class covariance matrix

样本在d维X空间

(1) 各类样本均值向量 m_i

$$m_i = \frac{1}{n_i} \sum_{x_k \in X_i} x_k, \quad i=1,2$$

(2) 样本类内离散度矩阵 S_i 与总类内离散度矩阵 S_w

$$S_i = \sum_{X \in \mathfrak{X}_i} (X - m_i)(X - m_i)^T, \quad i=1,2$$

$$S_w = S_1 + S_2$$

(3) 样本类间离散度矩阵 S_b

$$S_b = (m_1 - m_2)(m_1 - m_2)^T$$

样本在一维Y空间

(1) 各类样本均值 \tilde{m}_i

$$\tilde{m}_i = \frac{1}{N_i} \sum_{y \in Y_i} y, \quad i=1,2$$

(2) 样本类内离散度 \tilde{S}_i^2 和总类内离散度 \tilde{S}_w

$$\tilde{S}_i = \sum_{y \in Y_i} (y - \tilde{m}_i)^2, \quad i=1,2$$

$$\tilde{S}_w = \tilde{S}_1 + \tilde{S}_2$$

$$\tilde{S}_b = (\tilde{m}_1 - \tilde{m}_2)^2$$

Fisher准则函数

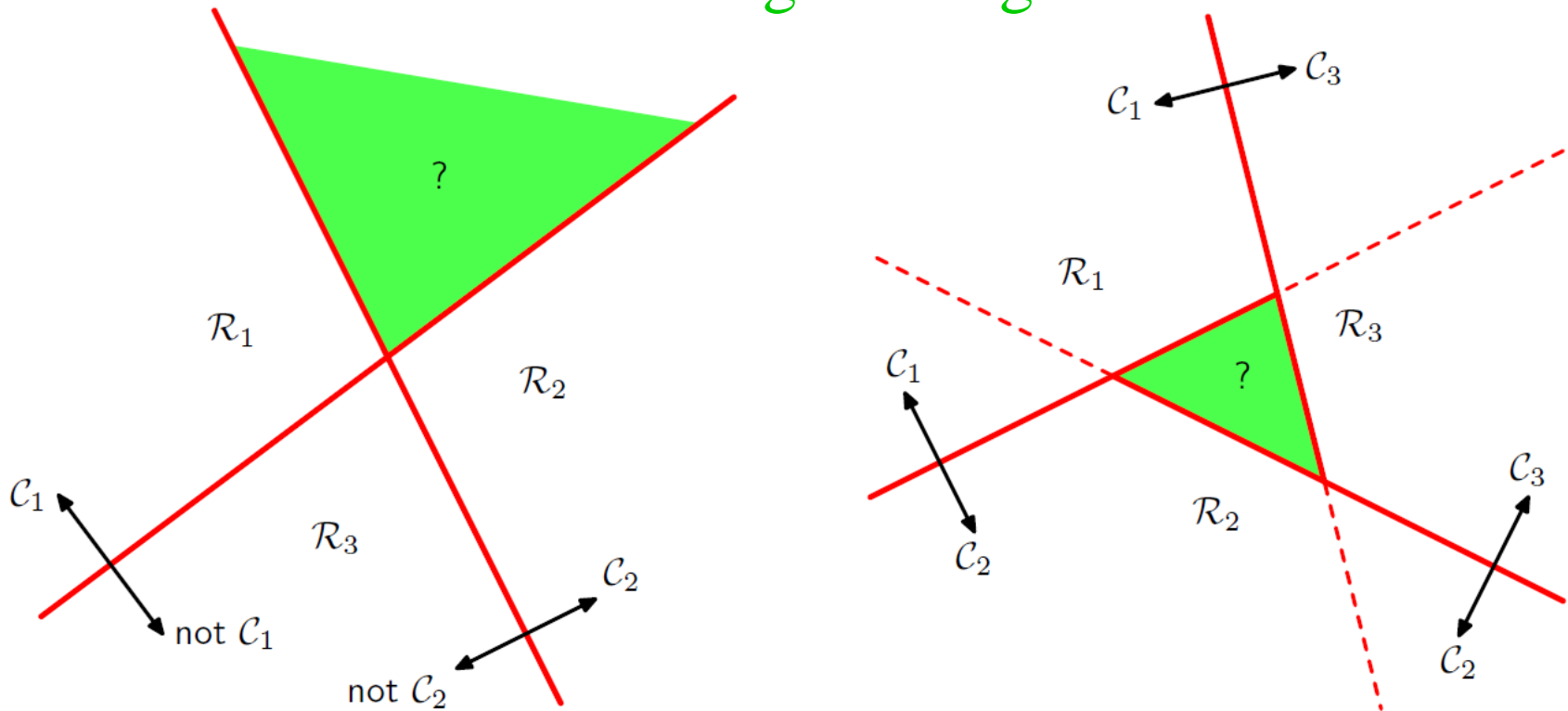
$$J_F(w) = \frac{\tilde{S}_b}{\tilde{S}_1 + \tilde{S}_2} = \frac{w^T S_b w}{w^T S_w w}$$

Fisher最佳投影方向求解

$$w^* = \underset{w}{\operatorname{argmax}} J_F(w)$$

多类问题, $K > 2$ classes

ambiguous regions

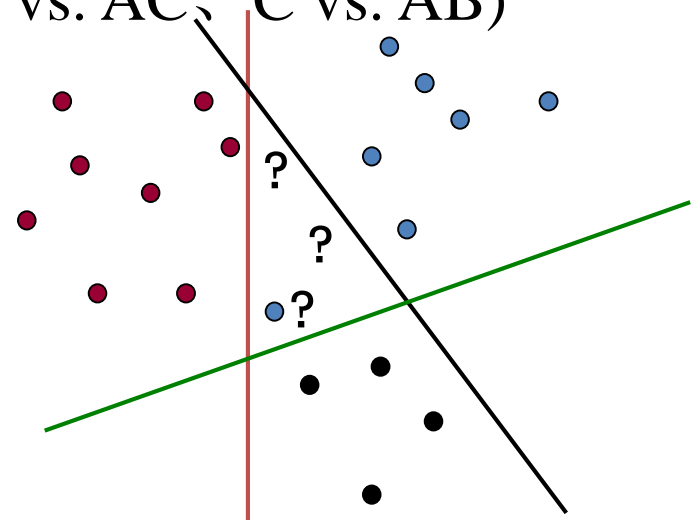


左图 one-versus-the-rest classifier: $K-1$ classifiers

右图 one-versus-one classifier: $K(K-1)/2$ classifiers

多标签分类问题

- 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
- 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
 - 对于多标签分类问题（比如A、B、C三类），可以组合为多个二类线性分类器(A vs BC、B vs. AC、C vs. AB)



多类情形分类器的评估

混淆矩阵（confusion matrix）

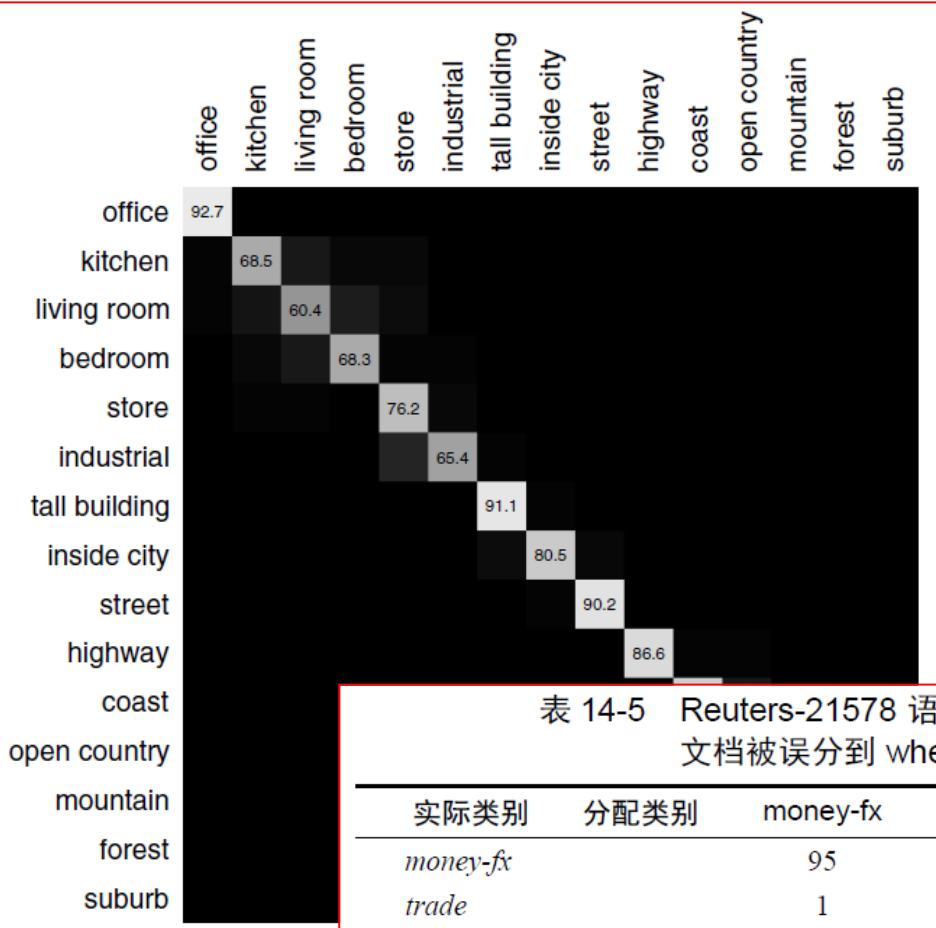


表 14-5 Reuters-21578 语料上的一个混淆矩阵。比如，14 篇属于 grain 类的文档被误分到 wheat 类中。选自 Picca 等人（2006）

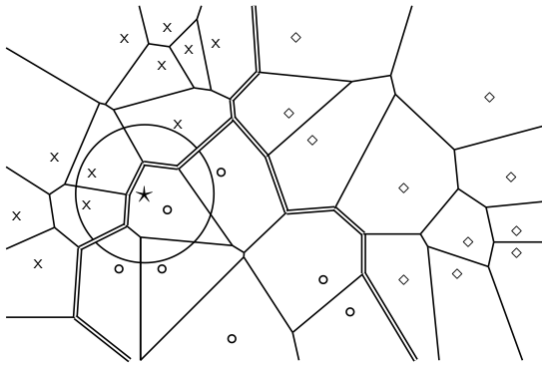
实际类别	分配类别	money-fx	trade	interest	wheat	corn	grain
<i>money-fx</i>		95	0	10	0	0	0
<i>trade</i>		1	1	90	0	1	0
<i>interest</i>		13	0	0	0	0	0
<i>wheat</i>		0	0	1	34	3	7
<i>corn</i>		1	0	2	13	26	5
<i>grain</i>		0	0	2	14	5	10

小结：线性分类器

- 线性分类器：超平面 $\sum_i w_i x_i > \theta$?
- **Two-class Rocchio as a linear classifier**
- **Naive Bayes is a linear classifier**
- **kNN不是线性分类器**
- **Linear / nonlinear classifiers**
 - Noise documents
 - Fisher's linear discriminant
- **single label problem \rightarrow multilabel classification**

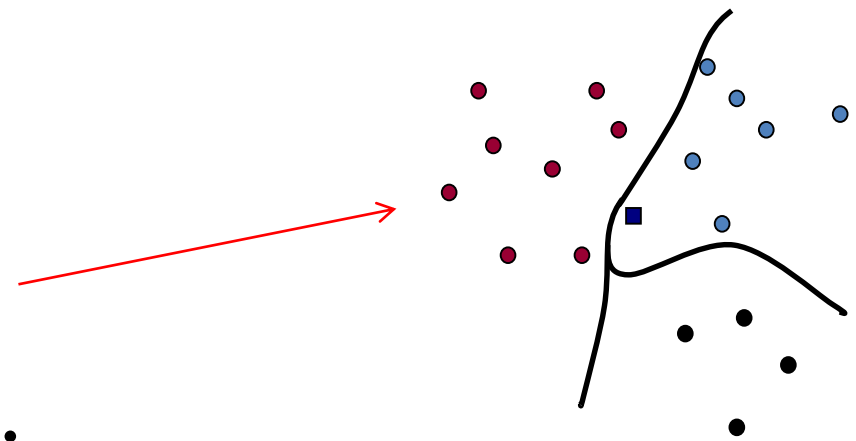
本讲要点回顾

- 基于向量空间的分类



Rocchio

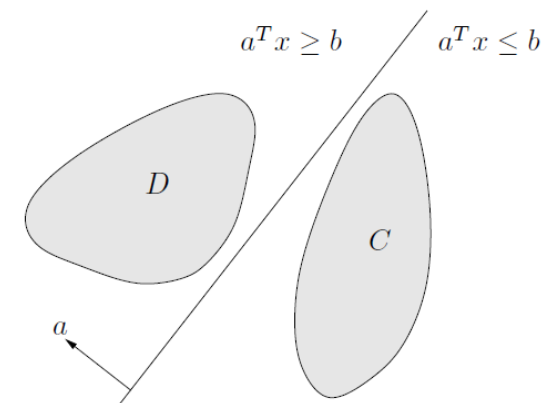
kNN



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

- 线性分类器: **hyperplane**

- Rocchio、Naive Bayes



谢谢大家!