

模式识别

最近邻 &
模式识别系统框架及其各模块的简介

吴建鑫
南京大学计算机系，2018

目标

- ✓ 理解并能熟练运用最近邻方法进行分类
- ✓ 了解最近邻方法的限制、缺陷以及可能的解决办法
- ✓ 理解并掌握模式识别系统各模块的作用、基本概念和解决方案的分类
- ✓ 提高目标
 - 进一步能将最近邻方法应用到实际研究问题中去（研究生、部分本科生）
 - DHS第7—10章、PRML第8—12章（偏Bayesian角度）将不详细讲授，感兴趣的同学可以自学

最近邻规则

Nearest neighbor rule

问题设置problem setup

✓ 分类问题classification

- 训练集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$
- 训练样本(sample): $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$
- 样本的标记(label): $y_i \in \mathcal{Y} = \{1, 2, \dots, C\}$
 - 样本一共被分为 C 个类别(category)
 - 例如, 在我们的例子里, $C = 2$, $y_i = 1$ (男) 或者 $y_i = 2$ (女)

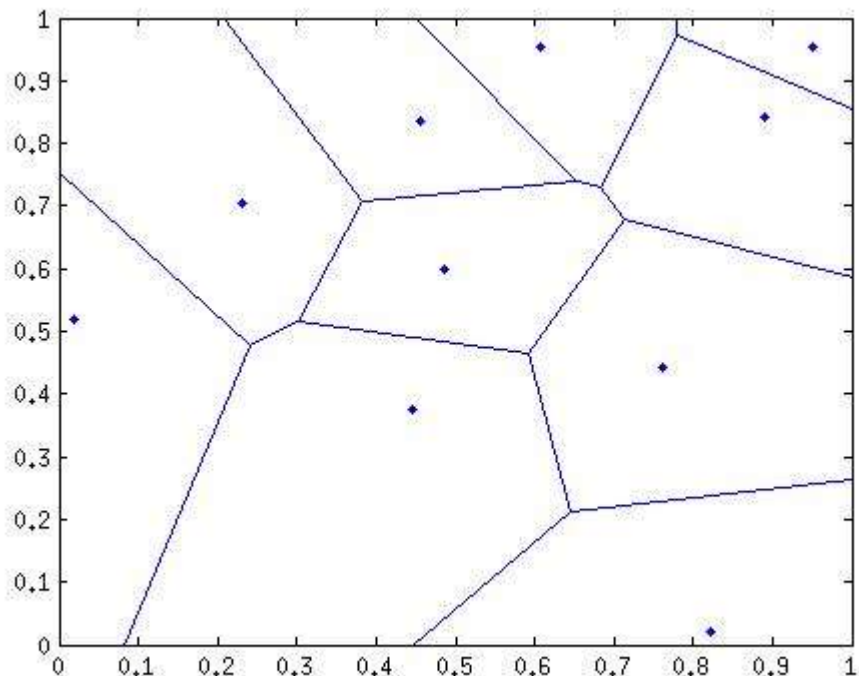
✓ 存在一个距离(distance)函数: $d(\mathbf{x}, \mathbf{y}) \in \mathbb{R}$

- 能够度量 \mathbf{x} 和 \mathbf{y} 之间的距离, 或者不相似程度(level of dissimilarity)

最近邻规则和Voronoi图

给定一个测试样例 \mathbf{x}

1. 发现其最近邻 $i^* = \underset{i}{\operatorname{argmin}} d(\mathbf{x}, \mathbf{x}_i)$
2. 输出对 \mathbf{x} 的预测: y_{i^*}



Voronoi图
(Voronoi
Diagram)

最近邻可能出现的问题

- ✓ 如果出现平局 (tie) ?
 - $d(\mathbf{x}, \mathbf{x}_i) = d(\mathbf{x}, \mathbf{x}_j)$
 - $y_i = y_j?$ $y_i \neq y_j?$
- ✓ 如果出现离群点 (outlier) ?
 - K-近邻 (kNN, k-nearest neighbor) 规则
 - 可能遇到的问题?
- ✓ 能做的多好?
 - 当训练样本趋于无穷时 ($n \rightarrow \infty$), 最近邻的错误率最多是最佳错误率的两倍
 - 有限样本 (finite sample) 时的结论尚不清楚

计算、存储代价(cost)

- ✓ 假设 $d(\mathbf{x}, \mathbf{y})$ 是欧式距离 (Euclidean distance, ℓ_2 distance)
 - 其复杂度 (complexity) 是 $O(d)$
 - NN的复杂度 $O(nd)$
 - DHS 152页的复杂度是错的
 - K-NN的复杂度**同样**是 $O(nd)$
 - 或者是 $O(nd) + O(n) + O(k)$, 但通常 k 较小, 可以忽略
 - 从 n 个数 (距离) 中选择 k 个最小的, 复杂度是?
- ✓ 考虑一下, 如果是ILSVRC, 需要**多长时间, 多大的存储空间? 这是NN的主要问题**
 - $n = 1,200,000$
 - $d = 262,144$

降低NN的计算、存储代价

- ✓ 近似最近邻 (approximate nearest neighbor, ANN)
 - 不要求一定是距离最短的 k 个
 - 如第 k 个NN，其距离是 d_k ，则ANN要求其选取的所有 k 个样例的距离 \hat{d} 满足 $\hat{d} \leq (1 + \epsilon)d_k$ 即可
 - 可以将kNN搜索 (search、查找) 速度提高几个数量级
- ✓ 二值哈希 (binary hashing)
 - hash函数 f_i : 将 \mathbb{R}^d 分为两部分，分别用 $f_i = 0, 1$ 表示
 - 设计 m 个hash函数 f_1, \dots, f_m ，每个 \mathbf{x} 表示为 m 个bit
 - $m \ll d$ ，计算和存储大幅简化，需要设计好的hash

系统各模块(混合)简介

Introducing various components in a mixed order

细化(refined)的框架

✓ 机器学习 $f: \mathcal{X} \mapsto \mathcal{Y}$

1. 与领域无关的特征变换和特征抽取

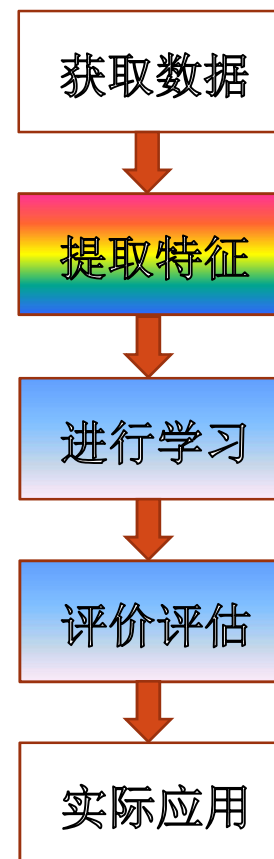
■ Normalization, PCA, FLD, ...

2. 针对不同数据特点的不同学习算法

■ SVM, Decision Tree, imbalanced learning, HMM, DTW, graphical model, deep learning, pLSA, ...

3. 机器学习方法常见分类、策略

✓ 针对不同问题的评价准则 (evaluation criterion)



评价准则—泛化和测试误差

✓ 暂时只考虑分类问题的评价

✓ 假设 $(\mathbf{x}, y) \sim p(\mathbf{x}, y)$

- 泛化误差 generalization error: $E_{(\mathbf{x}, y) \sim p(\mathbf{x}, y)}[f(\mathbf{x}) \neq y]$

- 通常无法实际计算

- 根本假设: 训练集 D_{train} 和测试集 D_{test} 都是服从真实数据分布 $p(\mathbf{x})$ 的, 或者, 他们的样例是从 $p(\mathbf{x})$ 中取样 (sample) 的

- 测试误差 (testing error)

$$err = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(f(\mathbf{x}_i) \neq y_i), \quad \mathbf{x}_i \in D_{test}$$

- 精确度 (accuracy): $acc = 1 - err$

一种常见的学习框架

✓ 代价最小化 cost minimization

- 错误是最常考虑的代价，所以现在我们可以说学习的目标是在训练集上获得最小的代价

✓ $\min_f \frac{1}{n} \sum_{i=1}^n \mathbb{I}(f(\mathbf{x}_i) \neq y_i), \mathbf{x}_i \in D_{train}$

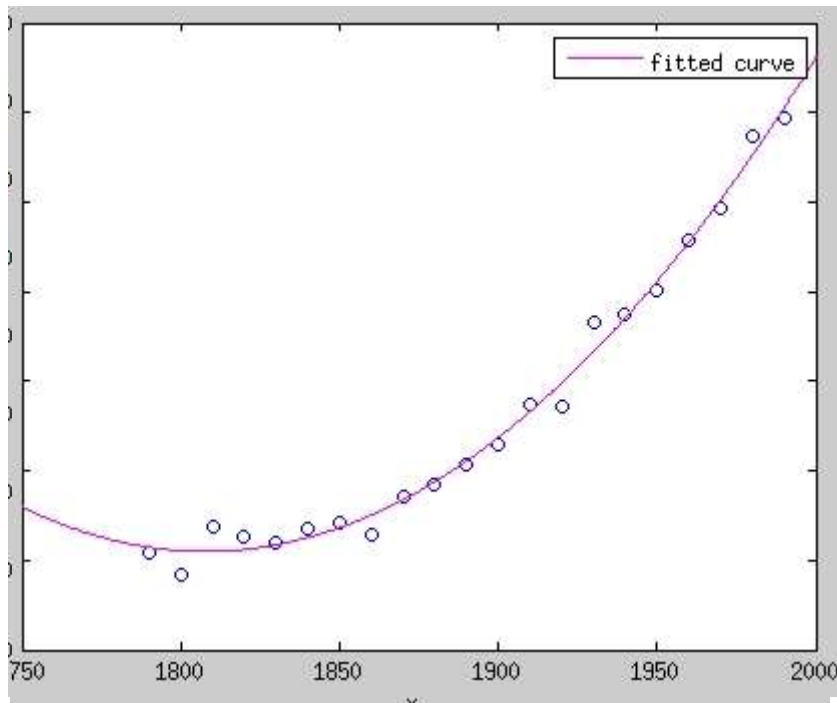
- 难以优化 - 怎样求解？
- 一种方法是：把不连续的指示函数(indicator function)换成性质相似，但好优化的函数
- 如, $(f(\mathbf{x}_i) - y_i)^2$

✓ 学习这种思路：形式化、简化、优化

- Formalization, simplification, optimization

过拟合和欠拟合

✓ Overfitting & underfitting, 以回归 (regression) 为例



- 以七阶多项式拟合 (直线)
- 学习模型的复杂性 远大于数据的复杂性
- 称为过拟合 underfitting

正则化regularization

- ✓ 通常难以精确估计学习模型、数据的复杂性
 - 往往选用较复杂的学习模型
 - 训练集误差通常小于测试集误差(需要两者不相交)
- ✓ 那么如何降低overfitting的可能性呢？
 - 正则化regularization, 会在SVM部分看到例子
- ✓ 进一步阅读：
 - 正则化如何能降低模型的复杂性？PRML以及ESL(The Elements of Statistical Learning)

如果没有测试集

- ✓ 例如，总的数据量比较小（如医学图像）
 - 如何评估？
- ✓ 交叉验证cross validation
 1. 将训练集分为大小大致相等的 N 部分
 2. for $i = 1:N$
 1. 取第 i 部分的数据为测试集
 2. 取所有其余（一共 $N - 1$ 个部分）的数据为训练集
 3. 学习模型并评估/测试得到错误率为 err_i
 3. 交叉验证得到的错误率为 $\frac{1}{N} \sum_{i=1}^N err_i$
 - 称为 N 倍交叉验证N-fold CV （常用 $N=5$ or 10 ）
- ✓ 可能需要进行多次试验（后面会讲）

数据、代价的不平衡性imbalance

✓ 例如，两类问题中，一类数据远比另一类数据多

- 如，体检中阴性和阳性
- 男女比例
- 或在一类犯错的代价远高于另一类

- 不平衡学习 (imbalance learning)
- 代价敏感学习 (cost-sensitive learning)

• 进一步阅读：周志华教授主页和论文

<http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/publication.htm>

评价不平衡时的准则(1)

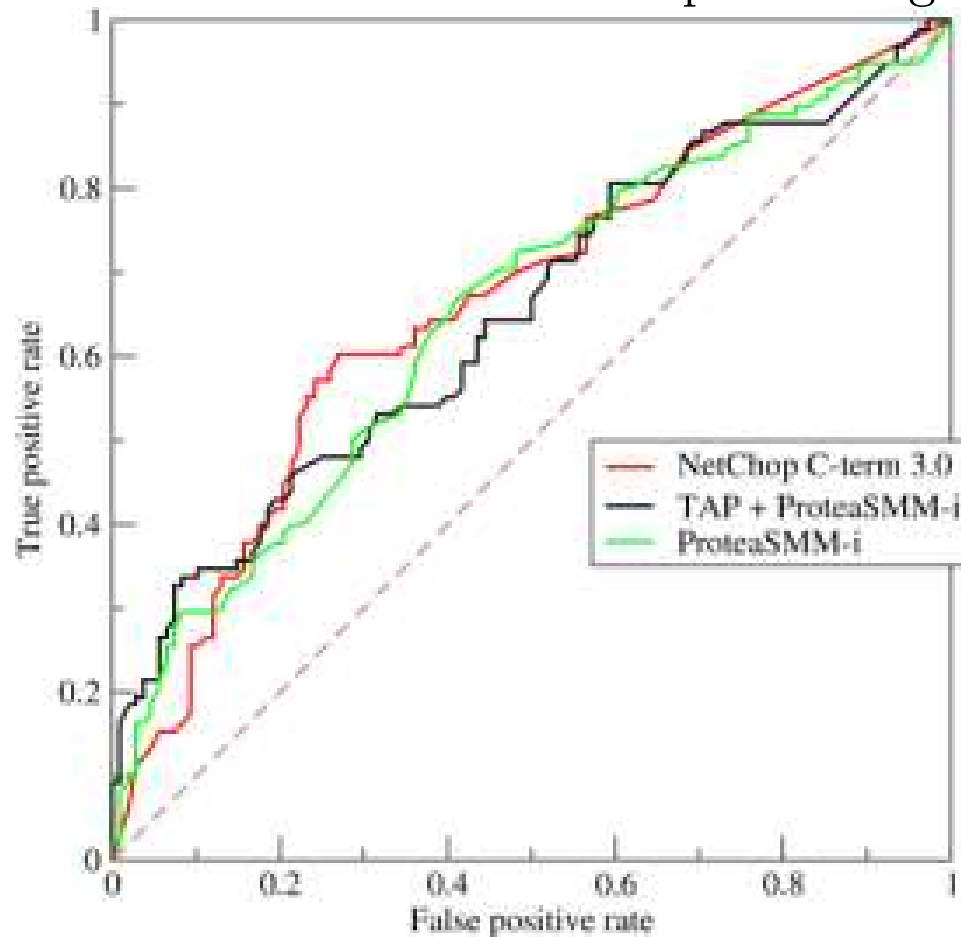
	预测为positive	预测为negative
真实值为positive	True positive (真阳性)	False negative (伪阴性)
真实值为negative	False positive (伪阳性)	True negative (真阴性)

- ✓ TP、TN、FP、FN: 标记四种情况的样例数目
- ✓ TOTAL: 总数 $TP+TN+FP+FN$
 - 正样本数目: $P = TP+FN$, 负样本数目: $N = FP+TN$
- ✓ False positive rate: $FPR = FP / N$
- ✓ False negative rate: $FNR = FN / P$
- ✓ True positive rate: $TPR = TP / P$
- ✓ Accuracy: $ACC = (TP+TN) / TOTAL$

评价不平衡时的准则(2)

✓ AUC-ROC (Area Under the ROC Curve)

• ROC - Receiver operating characteristic



- Y轴: TPR
- X轴: FPR
- 其值为面积
- 为什么?
- 对角线是?
- 非减

<http://upload.wikimedia.org/wikipedia/commons/6/6b/Roccurves.png>

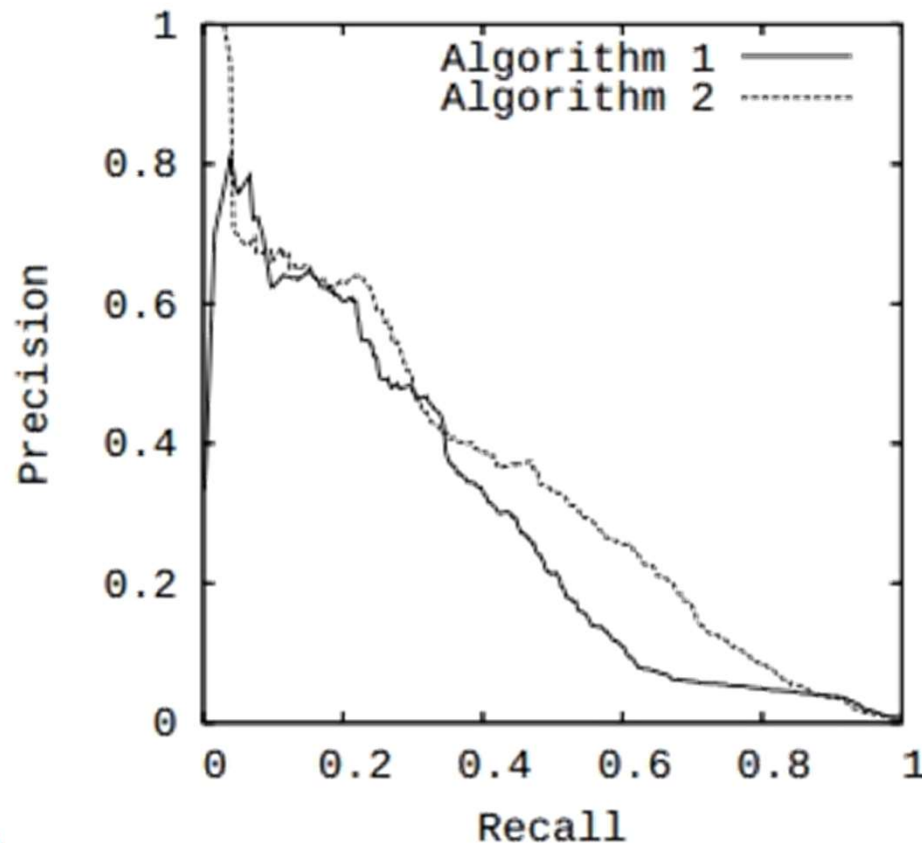
评价不平衡时的准则(3)

- ✓ Precision (查准率) : $PRE = TP / (TP + FP)$
- ✓ Recall (查全率) : $REC = TP / P$ (和TPR一样)
- ✓ F1 score: Precision和Recall的调和平均 (harmonic mean)
 - 调和平均: $\left(\frac{x^{-1}+y^{-1}}{2}\right)^{-1} = \frac{2xy}{x+y}$
 - $F1 = 2TP / (2TP+FP+FN)$
 - 推导一下
 - 为什么?



评价不平衡时的准则(4)

✓ AUC-PR (Area Under the Precision-Recall Curve)



- Y轴: Precision
- X轴: Recall
- 其值为面积
- 为什么?
- 单调吗?

进一步阅读: [The Relationship Between Precision-Recall and ROC Curves](#), 左边的图来自该论文

代价矩阵

- ✓ 目前常见的为 $\begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
 - λ_{ij} : 当真实值为 i 、模型预测为 j 时付的代价
 - 0-1代价: 即分类正确代价为0, 分类错误代价为1
 - 但是, 根据实际情况, 可以给 λ_{ij} 设置任何值
 - 代价不平衡学习
- ✓ 对代价的计算: $E_{(x,y)}[\lambda_{y,f(x)}]$
 - 当使用0-1代价时, 和错误率一致

真实值Groundtruth

- ✓ 大多数时候，是手工标注的(manual annotation)
 - 或者人也不知道确切的答案
 - 有时候疲劳或其他因素会导致标注的错误
 - 很耗时、昂贵
- ✓ 真实值的形式
 - 分类：一个离散的类别
 - 回归regression：一个连续的值
 - 结构structured output：例如，输出一个句子的分词结果“一个/句子/的/分词/结果”

能100%准确吗: Bayes框架的回答(1)

- ✓ $f: \mathcal{X} \mapsto \mathcal{Y}$, 一个数据对 (data pair): (\mathbf{x}, y)
 - 假设注重于分类: $\mathcal{Y} = \{1, 2, \dots, m\}$
 - 先验概率prior probability: $p(y = i)$
 - 在没有看到任何数据时, 怎么分类?
 - 后验概率posterior probability: $p(y = i | \mathbf{x})$
 - 看到数据 \mathbf{x} 后, 得到更多的信息, 可以对分类有更好的估计
 - 类条件概率class conditional probability: $p(\mathbf{x} | y = i)$
 - 数据总的分布 $p(\mathbf{x})$ 和每个类别内部的分布 $p(\mathbf{x} | y = i)$ 不一样

- ✓ 贝叶斯定理Bayes' theorem

$$p(y = i | \mathbf{x}) = \frac{p(\mathbf{x} | y = i)p(y = i)}{p(\mathbf{x})} = \frac{\text{条件} \times \text{先验}}{\text{数据}}$$

能100%准确吗: Bayes框架的回答(2)

✓ 贝叶斯决策规则Bayes decision rule:

- 选择代价最小的类别输出

$$\operatorname{argmin}_y E_{(x,y)} [\lambda_{y,f(x)}]$$

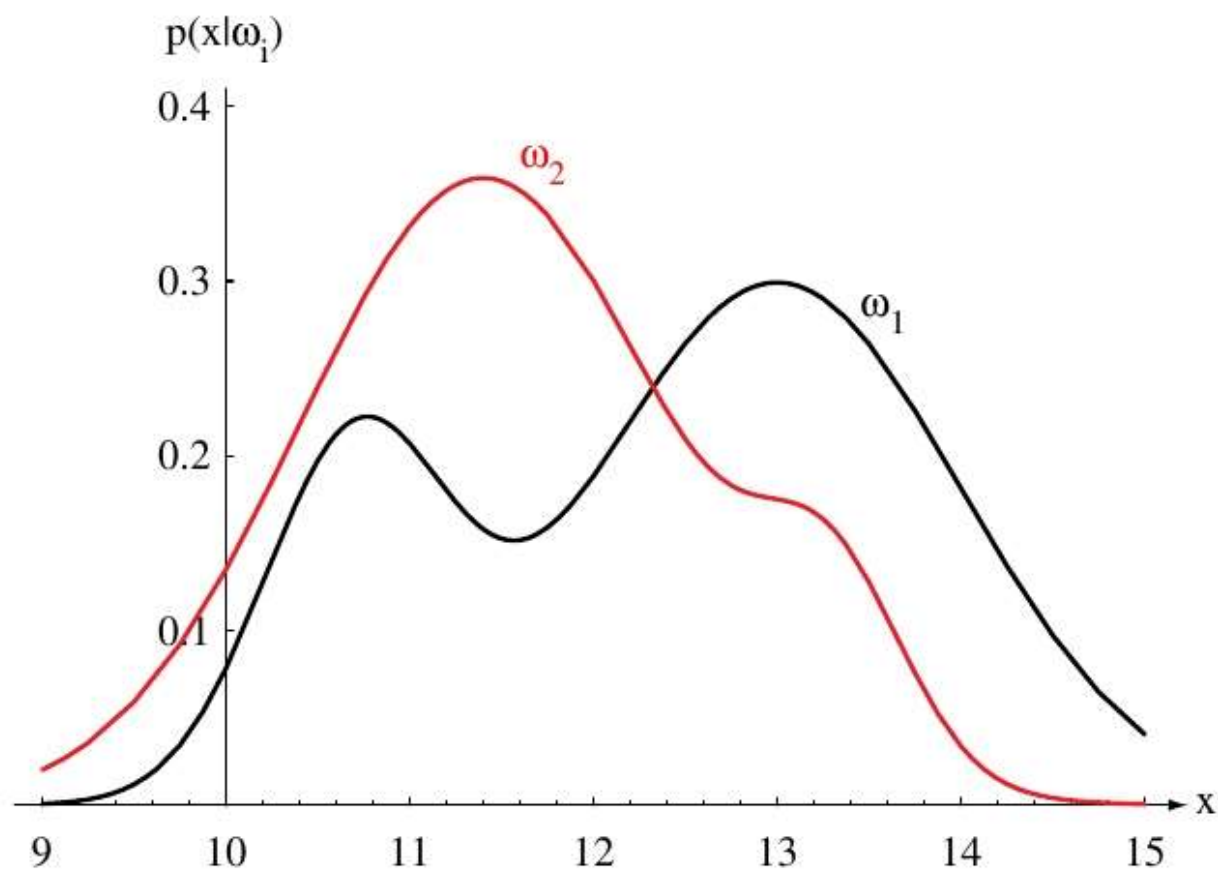
- 贝叶斯风险Bayes risk: 使用贝叶斯决策规则的风险
- 其是理论上我们能得到的最好的结果, 记为 R^*

✓ 在使用0-1风险时, 风险和错误率等价

- 所以, R^* 是我们理论上能得到的最小误差
- $1 - R^*$ 是理论上最高的准确率!

✓ 自学: DHS2.1 DHS2.2 (包括似然比规则 likelihood ratio rule)

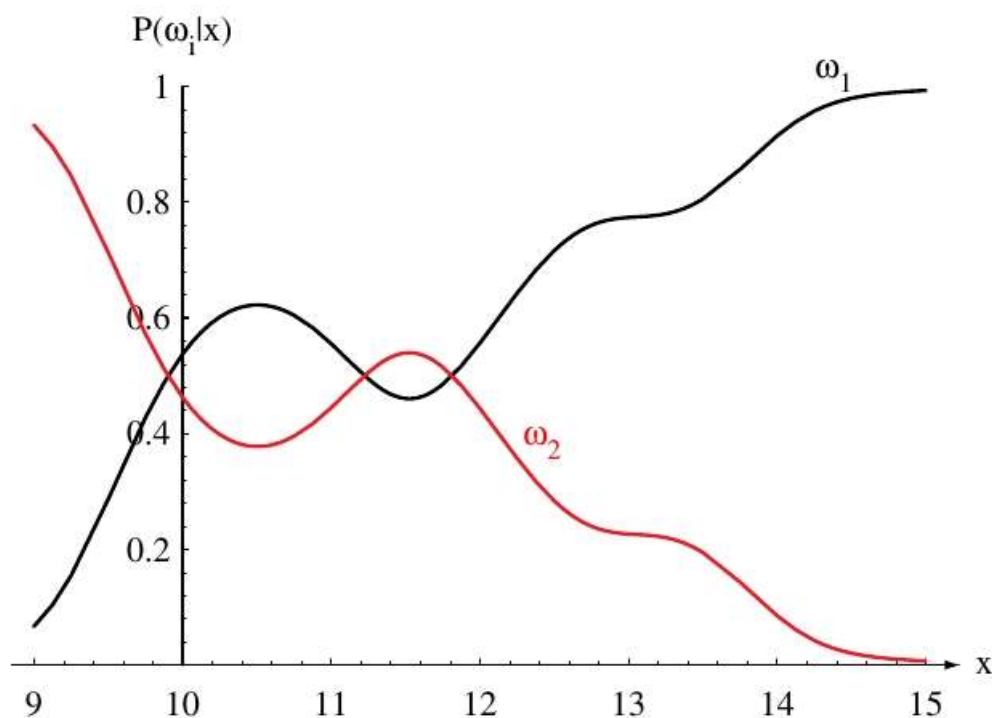
类条件概率示意图



该在哪里分开？错误（或风险）是多少？
图片来自教程DHS

贝叶斯决策规则

- ✓ 在0-1风险时，选择后验概率最大的那个类别
 $\operatorname{argmax}_i p(y = i | \mathbf{x})$



其中第一类prior为2/3
第二类为1/3

图片来自教程DHS

错误从哪里来—以回归为例？

✓ 真实（但未知）的函数 $F(\mathbf{x})$

- 用由其产生的数据集 D 来学习，即 $y = F(\mathbf{x})$ 没有误差
- 回归的代价函数是欧几里得距离

$$\checkmark E_D \left[(f(\mathbf{x}; D) - F(\mathbf{x}))^2 \right] = (E_D[f(\mathbf{x}; D)] - F(\mathbf{x}))^2 + E_D[(f(\mathbf{x}; D) - E_D[f(\mathbf{x}; D)])^2]$$

- \mathbf{x} 和 $F(\mathbf{x})$ 是定值(constant)，只有 D 出现时才取期望
- 简写为 $E[(f - F)^2] = (F - Ef)^2 + E[(f - Ef)^2]$
- DHS 376页的处理（或翻译）有问题

偏置-方差分解

- ✓ Bias-variance decomposition
 - $E[(f - F)^2] = (F - Ef)^2 + E[(f - Ef)^2]$
 - $E[F - Ef]$ -- 偏置bias
 - 当训练集取样有差异时，其值不变
 - $E[(f - Ef)^2] = Var_D(f(\mathbf{x}; D))$ 方差
 - 当训练集取样有差异时，会带来预测的差异（误差不同）
- ✓ 误差=偏置²+方差
- ✓ 当考虑到 $y = F(\mathbf{x})$ 有误差是（白噪声）
 - 误差=偏置²+方差+噪声
 - 估计误差时，如没有测试集，需多次平均
- ✓ 进一步阅读：分类时候的分解(DHS9.3.2)

对分解的解读

- ✓ 偏置与数据无关，是由模型（的复杂度）决定的
 - 例如，线性分类器（1阶多项式）的偏置大
 - 但是，7阶多项式的复杂度高，偏置小
- ✓ 但是，方差 $Var_D(f(\mathbf{x}; D))$ 和抽样得到的训练集以及模型两者都有关系
 - 例如，高阶多项式的方差大
- ✓ 怎么减少误差？
 - 对于噪音，机器学习没有办法——高质量的数据获取！
 - 减少偏置和方差
 - 如集成方法(ensemble methods)

进一步的阅读

- ✓ NN & kNN: DHS 4.5 & 4.6, 可以等到课程讲完 Bayesian 相关的内容之后
- ✓ FLANN: <http://www.cs.ubc.ca/research/flann/>
 - ANN 软件和相关论文、文档
- ✓ Hash:
http://en.wikipedia.org/wiki/Locality-sensitive_hashing 及其页面中的资源
- ✓ 其他见各页面的进一步阅读资源