Nearest neighbor rule

-PROMISE-

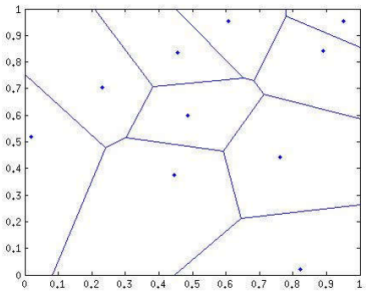
Tuesday 5th April, 2022

1 最近邻规则

1.1 最近邻规则介绍

• 设置训练集为 $D = (x_1, y_1), (x_2, y_2), ...(x_n, y_n)$，其中 $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$,$y_i \in \mathcal{Y} = 1, 2, ..., C$

• 设置距离函数 $d(x, y) \in \mathbb{R}$，比较 x,y 之间的距离，找到距离最近的 x_i, 即 $i* = argmin_i d(x, x_i)$, 输出对 x 的分类预测：y_{i*}



Voronoi图 (Voronoi Diagram)

• 考虑可能出现的情况
平局：$d(x, x_i) = d(x, x_j)$:
出现离群点：KNN（k 近邻规则）

1.2 最近邻性质分析

• 最近邻的正确率：当训练样本趋于无穷的时候，最近邻的错误了最多是最佳错误率的两倍（有限样本的情况尚不清楚）

• 计算、存储代价：假设 d(x,y) 是欧式距离

复杂度是 O(d)

NN 的复杂度是 O(nd)

K-NN 复杂度是 O(nd)，或者是 O(nd)+O(n)+O(k)，但是通常 k 较小，可以忽略

如果是 ILSVRC，需要多长时间，多大存储空间?

1.3 降低 NN 的计算、存储代价

• 近似最近邻（ANN）不要求一定时距离最短的 k 个，如第 k 个 NN，其距离是 d_k，则 ANN 要求选取所有的 k 个样例，距离满足 $d \leq (1+\epsilon)d_k$ 即可，可以将 KNN 速度提高几个量级

• 二值哈希：将 $\mathbb{R}^d$ 分成两个部分，分别用 $f_i = 0, 1$ 表示，设计 m 个这样的 hash 函数，每个 x 用 m 个 bit 表示，$m \ll d$，计算和E大幅简化，需要设计好 hash

2 系统各模块混合简介

2.1 机器学习框架

2.2 评价标准

• 暂时只考虑分类问题的评价：

假设 (x,y) p(x,y)

泛化误差：$E_{(x,y)\ p(x,y)}[f(x) \neq y]$(通常无法实际计算)

假设训练集和测试集都是服从真实数据分布 p(x) 的，或者，他们的样例是从 p(x) 中取样的，测试误差就是 $err = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(f(x_i \neq y_i)), (x_i \in D_{test})$

精确度 acc=1-err

2.3 一种常见的学习框架

• 将代价最小化，在学习的例子上，可以理解成错误最小化，也就是 $\min_f \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(f(x_i \neq y_i)), (x_i \in D_{train})$

2.4 过拟合和欠拟合

• 学习模型的复杂性小于数据的复杂性，为欠拟合；反之为过拟合 (可以通过正则化降低过拟合的可能性，会再 SVM 看到例子)

2.5 交叉验证

没有测试集的时候，可以将训练集分成大小相等的 N 部分，用其中一部分作为测试集，其他（N-1）部分作为训练集，取 N 次错误率的平均值为交叉验证得到的错误率

2.6 数据、代价的不平衡性

例如两类问题中一类数据远比另一类多，不平衡学习，利用代价敏感学习解决

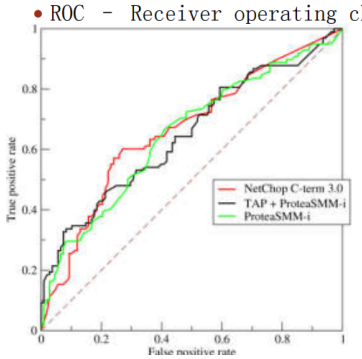
2.6.1 评价不平衡准则

可以理解为分开考虑

	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

- ✓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

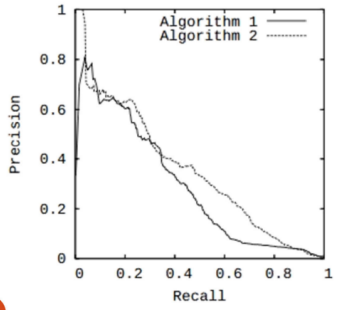
18

查准率 Precision：$PRE = \frac{TP}{TP+FP}$

查全率 Recall: $REC = \frac{TP}{P}$

F1 score:Precision 和 Recall 的调和平均：$(\frac{x^{-1}+y^{-1}}{2})^{-1} = \frac{2xy}{x+y}$,$F1 = \frac{2TP}{2TP+FP+FN}$

- ✓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

2.7 代价矩阵

$$\begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

λ_{ij} 表示真实值 i，模型预测为 j 时的代价

代价计算：$E_{(x,y)}[\lambda_{y,f(x)}]$

2.8 真实值

大部分时候是人工标注的

2.9 贝叶斯框架分析准确率

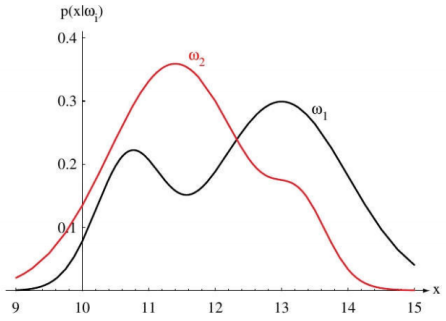
- 对于分类：

  先验概率：p(y=i)

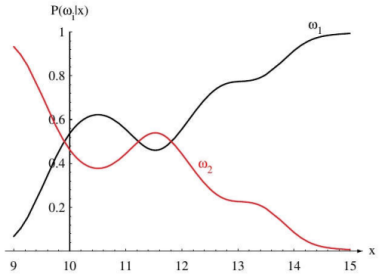
  后验概率：p(y=j|x)(贝叶斯定理)

  类条件概率：p(x|y=j)
- 贝叶斯决策规则：选择代价最小的类别输出 $(\underset{y}{argmin}\, E_{(x,y)}[\lambda_{y,f(x)}])$

类条件概率示意图



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



其中第一类prior为2/3
第二类为1/3

图片来自教程DHS

2.10 错误来源

没看懂：

错误从哪里来—以回归为例？

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

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

- $\boldsymbol{x}$和$F(\boldsymbol{x})$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
- $E[(f-F)^2] = (F-Ef)^2 + E[(f-Ef)^2]$
- $F - Ef$ -- 偏置bias
  - 当训练集取样有差异时，其值不变
- $E[(f-Ef)^2] = Var_D(f(\boldsymbol{x};D))$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$y = F(\boldsymbol{x})$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)