readme

Table of Contents

1 第1章 绪论

1.1 引言

- 机器学习/machine learning
  - 主要研究特定的"学习算法", 即在计算机上从数据产生模型的算法.
- 学习算法/learning algorithm
- 模型/model
  - 泛指: 从数据中学得的结果.
- 模式
  - 局部性结果(例如, 一条规则).

1.2 基本术语

- 数据集/data set
  - 示例/样本的集合(?并非要求样例)
- 示例/instance/样本/sample
  - 一个事件或对象的描述
- 属性/attribute/特征/feature
  - 反映事件或对象在某个方面的表现或性质的事项
- 属性值/attribute value/特征值
  - 属性上的取值
- 属性空间/attribute space/样本空间/sample space/输入空间
  - 属性张成的空间
- 特征向量/feature vector
  - 一个示例, 也称为一个特征向量
- 维数/dimensionality
- 学习/learning/训练/training
  - 从数据中学得模型的过程
- 训练数据/training data
  - 训练过程使用的数据
- 训练样本/training sample/训练示例/training instance
  - 训练过程中使用的样本
- 训练集/training set
  - 训练数据组成的集合
- 假设/hypothesis
  - 对应了关于数据的某种潜在规律的学得模型
- 真相/真实/ground$_{\text{truth}}$

    - 数据的潜在规律本身
- 学习过程
    - 在所有假设组成的空间进行搜索的过程,目的是为了找出或逼近真相
- 学习器/learner
    - 模型
- 标记/label
    - 示例的结果信息
- 样例/example
    - 拥有了标记信息的样例
- 标记空间/label space/输出空间
    - 所有标记的集合
- 分类/classification
    - 欲预测的值是离散值
- 回归/regression
    - 欲预测的值是连续值
- 二分类/binary classification
    - 只涉及两个类别的分类任务
- 正类/positive class
    - 二分类中的其中一类别
- 反类/negative class
    - 二分类中除正类之外的另一类别
- 多分类/multi-class classification
    - 多类别识别
- 测试/testing
    - 学得模型后使用其进行预测的过程(?)
- 测试样本/testing sample
    - 被预测的样本
- 聚类/clustering
    - 训练集中只有样本, 没有标记, 将样本分成若干组(?)
- 簇/cluster
    - 聚类中的每一组
- 监督学习/supervised learning
    - 训练数据拥有标记
- 无监督学习/unsupervised learning
    - 训练数据没有标记
- 泛化能力/generalization
    - 学得模型适用于新样本的能力
- 分布/distribution
- 独立同分布/independent and identically distributed, i.i.d
    - 每个样本都是独立地从这个分布上采样获得, 称样本独立同分布

1.3 假设空间

- 归纳/induction
  - 特殊到一般的"泛化"过程, 即从具体的事实归结出一般性规律
- 演绎/deduction
  - 从一般到特殊的"特化/specialization"过程, 即从基础原理推演出具体情况
- 归纳学习/inductive learning
  - 归纳的过程, "从样例中学习"是归纳学习
  - 狭义: 学得概念; 广义: 学得"黑箱"模型
- 概念/concept
- 概念学习/概念形成
  - 从训练数据中学得概念
- 版本空间/version space
  - 同训练集"匹配/fit"的假设空间

1.4 归纳偏好

- 归纳偏好/inductive bias/偏好
  - 机器学习算法在学习过程中对某种类型假设的偏好
- 奥卡姆剃刀/Occam's razor
  - 一种常用的, 自然科学研究中最基本的原则(假设偏好), "若有多个假设与观察一致, 则选最简单的那个"
- 没有免费的午餐/No Free Lunch Theorem/NFL
  - 在所有"问题"出现的机会相同, 或所有问题同等重要的 **前提** 下, 所有学习算法的期望性能都跟随机胡猜差不多.
  - 现实问题通常不满足NFL的前提, 但NFL的寓意是 脱离具体问题, 空泛的谈论"什么学习算法更好"毫无意义, 因为若考虑所有潜在的问题(所有样本出现概览一样?), 则所有学习算法一样好.

1.5 发展历程

- 人工智能/artificial intelligence
- 推理期
  - 二十世纪五十年代到七十年代初, 人工智能处于的研究阶段, 那时人们以为只要能赋予机器逻辑推理能力, 机器就能具有智能
- 知识期
  - 二十世纪七十年代中期开始, 人工智能处于的研究阶段, 人们认为要使机器拥有智能, 就必须设法使机器拥有知识
- 学习期
  - 图灵1950年提到机器学习的可能, 逐步发展, 到二十世纪八十年代成为独立学科领域, 各类技术百花齐放
- 机器学习
- 机械学习
  - 机器学习的一种划分, 但实际机器并未学习, 仅将信息存储与需要时原封不动地取出使用
- 示例学习/类比学习/从指令中学习/通过观察和发现学习
- 归纳学习/从样例中学习(*)

    - 从训练样例中归纳出学习结果
- 符号主义学习
    - 二十世纪八十年代, 从样例中学习的一大主流. 逻辑和知识的结合, 代表技术, 决策树和基于逻辑的学习
- 连接主义学习
    - 二十世纪九十年代中期之前, 从样例中学习的另一主流, 基于神经网络的连接主义学习. 当时面临调参难题
    - 二十世纪初卷土重来, 以深度学习之名, 此时大数据时代, 有数据, 有计算能力
- 统计学习
    - 二十世纪九十年代中期, 成为从样例中学习的主流, 研究以统计学习理论支撑的技术, 代表技术, 支持向量机, 核方法

1.6 应用现状

- 众包/crowdsouring

1.7 阅读材料

- WEKA
    - 著名的免费机器学习算法学习程序库
- 多释原则/principle of multiple explanations
    - 主张保留和经验观察一致的所有假设, 与集成学习方面的研究很吻合
- 国际机器学习会议/ICML
- 国际神经信息处理系统会议/NIPS
- 国际学习理论会议/COLT
- 欧洲机器学习会议/ECML
- 亚洲机器学习会议/ACML
- Journal of Machine Learning Research
- Machine Learning
- IJCAI
- AAAI
- Arificial Intelligence
- Journal of Artificial Intelligence Research
- KDD
- ICDM
- ACM Transaction on Knowledge Discovery from Data
- Data Mining and Knowledge Discovery
- CVPR
- IEEE Ransactions on Pattern Analysis and Machine Intelligence
- Neural Computation
- IEEE Transactions on Neural Networks and Learning Systems
- Annals of Statistics

2 第2章 模型评估与选择

2.1 经验误差与过拟合

- 错误率/error rate

    - m个样本中有a个样本分类错误, 则错误率为a/m
- 精度/accuracy
    - 等于1-错误率
- 误差/error
    - 学习器的实际预测输出与样本的真实输出之间的差异
- 训练误差/training error/经验误差/empirical error
    - 学习器在训练集上的误差
- 泛化误差/generalization error
    - 学习器在新样本上的误差
- 欠拟合/underfitting
    - 学习算法学习能力低下, 样本特性没有学到.
    - 通常表现: 训练误差大
- 过拟合/overfitting(*)
    - 学习算法能力过于强大, 把训练样本中包含的不太一般的特性都学到了
    - 通常表现: 训练误差小, 泛化误差大
- NP难
    - P问题: 有多项式时间算法, 算起来快
    - NP问题: 算起来不确定快不快, 但我们可以快速检验这个问题的解
    - NP-complete问题/NPC问题: 属于NP问题, 且术语NP-hard问题
    - NP-hard问题/NP难问题: 比NP问题都要难的问题
- 模型选择/model selection

2.2 评估方法

- 测试集/testing set
- 测试误差/testing error
    - 通常将测试误差作为泛化误差的近似
- 留出法/hold-out
    - 直接将数据集分为两个互斥的数据集, 一个为训练集, 一个为测试集
- 采样/sampling
    - 可将数据集切分的过程以采样角度看待
- 分层采样/stratified sampling
    - 数据集切分过程中保留类别比例的采样方式
- 交叉验证/cross validation
    - 将数据集以保持数据分布一致性的方式(通常为分层采样)分成k份, k-1份训练, 1份测试, 可进行k次训练测试, 最终测试结果为k次测试的均值.
- k折交叉验证/k-fold cross validation
    - 同上
- 留一法/Leave-One-Out/LOO
    - k折交叉验证特例, k=m, m为数据集大小
    - 通常认为LOO的评估结果比较准确, 因为训练集大小接近m
    - 但数据集大时, 计算开销难以接受
- 自助法/bootstrapping

    - 以自助采样/bootstrap sampling为基础. 给定包含m个样本的数据集D, 对它采样得到同样为m个样本的数据集D'.采样方式为每次随机从D中采一个样本放到D',并放回到D
    - D中一部分样本在D'中出现多次, 一部分不出现. 样本在m次采样始终不被采到的概率为$p=(1-m)^m$, 对m->无穷求极限, p~0.368
    - 数据集较小, 难以划分训练测试集时, 很有用. 但训练集改变了初始数据集分布,会引入估计偏差
- 包外估计/out-of-bag estimate
    - 自助法用D'进行训练, D-D'用于测试, 测试结果称为包外估计
- 调参与最终模型
- 参数/parameter
    - 包含2种, 算法参数和模型参数
    - 通常算法参数称为 超参数
    - 模型参数就称为参数
- 超参数/hyperparameter
- 调参/parameter tuning
    - 调参指的是调节算法参数/超参数
    - 类似算法选择, 但因为参数选择范围大, 导致候选参数多, 调参工作量大
- 粗调
    - 通常为调参的第一阶段
    - 因调参工作量大, 初步将候选参数取值范围定的比较粗
- 精调
    - 通常为调参的第二阶段
    - 基于粗调得到候选参数, 在参数附近进行小范围搜索优化参数
- 验证集/validation set
    - 用于模型选择和调参
    - 测试集是模型实际使用时遇到的数据

2.3 性能度量

- 性能度量/performance measure
    - 衡量模型泛化能力的评价标准
- 均方误差/mean squared error
    - 回归问题常用性能度量
- 错误率
    - 分类问题常用性能度量
- 精度
    - 分能问题常用性能度量
- 查准率/precision/准确率
    - 二分类, 召回数据的精度
- 查全率/recall/召回率
    - 二分类, 召回数据对应召回数据的占比
- 真正例/true positive
    - 预测正确, 预测为正例

- 假正例/false positive
  - 预测错误, 预测为正例
- 真反例/true negative
  - 预测正确, 预测为反例
- 假反例/false negative
  - 预测错误, 预测为反例
- 混淆矩阵/confusion matrix
  - 表示分类结果的矩阵
- 单一评价指标/单一性能度量
- P-R曲线
  - 变化概率阈值, 得到的纵坐标为P查准率, 横坐标为R查全率的曲线
  - 如果两个模型, 第一个模型P-R曲线完全包住了第二个, 可断定第一个模型性能更优
- 平衡点/Break-Even Point/BEP
  - 是查全率等于查准率的取值
  - 如果两个模型, 第一个模型BEP大于第二个, 可认定第一个性能更优
- F1
  - F1 = (2*P*R)/(P+R)
- 调和平均/harmonic mean
  - 1/F1 = 1/2*(1/P+1/F)
  - 更重视更小值
- 加权F1
  - F_{beta} = (1+$beta^2$)*P*R/(($beta^2$*P)+R)
  - beta>0, 度量查全率对查准率的相对重要性. beta>1, 查全率有更大影响; beta<1, 查准率有更大影响
- 加权调和平均
  - 1/F_{beta} = 1/(1+$beta^2$)*(1/P + $beta^2$/R)
- 算数平均
  - (P+R)/2
- 几何平均
  - sqrt(PxR)
- 多分类中n个二分类混淆矩阵的考察
  - n个二分类混淆矩阵的考察, 分别有(P1,R1),…,(Pn,Rn)
- 宏查准率/macro-P
  - macro-P = 1/n(对i求和(Pi))
- 宏查全率/macro-R
  - macro-R = 1/n(对i求和(Ri))
- 宏F1/macro-F1
  - macro-F1 = 2*macro-P1*macro-R1/(macro-P1+macro-R1)
- 微查准率/micro-P
  - micro-P = mean(TP)/(mean(TP)+mean(FP))
- 微查全率/micro-R
  - micro-R = mean(TP)/(mean(TP)+mean(FN))
- 微F1/micro-F1
  - micro-F1 = 2*micro-P*micro-R/(micro-P+micro-R)
- 阈值/threshold/截断点/cut point

    - 学习器测试样本时输出一个实值/概率预测值, 将这个预测值同threshold/cut point比较, 若大于阈值则为正, 否则为反类
- ROC/受实验者工作特性(曲线)/Receiver Operating Characteristic
    - 变化阈值, 得到的纵坐标为真正例率TPR, 横坐标为假正例率FPR的曲线
- 真正例率/True Positive Rate/TPR
    - TPR = TP/(TP+FN)
    - 判对的正例的占正例总量比例
- 假正例率/False Positive Rate/FPR
    - FPR = FP/(TN+FP)
    - 判错的整理占反例总量比例
- AUC/Area Under ROC Curve
    - ROC的面积
    - 考虑的是样本预测的排序质量
    - AUC = 1 - l_{rank}
- 排序损失
    - l_{rank} = 1/(m_+*m_-)求和$x_{+属于D+}$&求和$x_{-属于D-}$-(punish(f(x_+)<f(x_-)) + 1/2*punish(f(x_+)=f(x_-)))
    - m_{+}是正例个数, m_{-}是反例个数, D_{+}是正例集合, D_{-}是反例集合
- 非均等代价/unequal cost
    - 为不同类型错误所造成的不同损失, 可为错误赋予"非均等代价"
- 代价矩阵/cost matrix
    - 样本分类的代价/数据集分类的代价, $cost_{ij}$: 将第i类样本预测为j的代价
- 代价敏感/cost-sensitive错误率
    - 加入代价权重
- 代价曲线/cost curve
    - 横坐标是[0,1]的正例概率代价, P(+)cost = $p*cost_{10}/(p*cost_{10} + (1-p)*cost_{01})$,p是样例为正例的概率
        - 个人不理解p的含义, 觉得更像是提供一个自变量, 对正例代价和负例代价做分配
    - 纵坐标是取值为[0,1]的归一化代价, $cost_{norm} = (FNR*p*cost_{10}+FPR*(1-p)*cost_{01})/(p*cost_{10}+(1-p)*cost_{01})$, FNR为假反例率, FPR为假正例率.
        - 个人理解为, 对正例代价和负例代价做分配时, 代价的归一化
    - 限定条件即为固定FPR和FNR时, y和x是线性关系, y=FNR*x+FPR*(1-x), 图中表示为线段. 因为x值域为[0,1], 求线段下的面积即为y求均值, 面积为期望总体代价, 等于(FPR+FNR)/2
        - 个人理解: 面积为期望归一化总体代价
    - 代价曲线为不同条件下所有线段的下限, 即P(+)cost下所有条件下的最小$cost_{norm}$.
    - 所有条件下的期望总体代价为代价曲线下的面积.
        - 个人理解为, 最小归一化总体代价的期望
- 规范化/normalization
    - 将不同变化范围的值映射到相同的固定范围, 常见[0,1], 称为"归一化"

2.4 比较检验

- 统计假设检验/hypothesis test

    - 对学习器性能比较提供了重要依据. 基于假设检验结果我们可推断出, 若在测试集上观察到学习器A比B号, 则A的泛化性能是否在统计意义上由于B, 以及这个结论把握有多大.
- 假设检验中的"假设"
    - 对学习器泛化错误率分布的某种判断或猜想
- 二项分布/binomial
    - 在n次独立重复的伯努利试验中，设每次试验中事件A发生的概率为p。用X表示n重伯努利试验中事件A发生的次数，则X的可能取值为0，1，…，n,且对每一个k（0≤k≤n）,事件{X=k}即为“n次试验中事件A恰好发生k次”，随机变量X的离散概率分布即为二项分布（Binomial Distribution）
- 二项检验/binomial test
    - 假设epson小于等于$epson_{0成立}$，若$epson^{小于等于"epson"}$的概率不小于1-alpha ，则接受假设，即若P($epson^{小于等于"epson"}$|epson小于等于$epson_0$)成立，则认为假设猜对了！
    - 如果要使泛化错误率epson小于等于$epson_{0这个假设的置信度大于1}$-alpha
    - 使k>$epson_0$*m的概率小于alpha时, 最大的epson="epson"(临界值)
    - 如果测试错误率$epson^{小于"epson"}$, 可得在alpha的显著度下, 假设epson小于等于$epson_{0不能被拒绝}$·
    - 二项检测同标准假设检验不同的地方是求k>$epson_0$*m+1时使用的概率分布是二项分布, 即泛化错误率为epson, 测试集有m个样本, 上错误样本数k满足二项分布
- 置信度/confidence
- 显著度
- t检验/t-test
    - 多次测试得到{epson^}, mean({epson^}), variance({epson^}), 满足t分布
    - 假设mean=$epson_{0和显著度alpha}$,(应该是说epson=$epson_0$), 最大错误率为临界值(双边). 如果$tao_{t在临界值范围内}$, 接受假设.
- t分布
    - k个测试错误率可看做泛化错误率$epson_{0的独立采样}$, tao_t = sqrt(k)(mean-$epson_0$)/variance, 服从自由度k-1的t分布
- 自由度为k-1的t分布
- 双边假设/two-tailed
    - (负无穷, $t_{-alpha}/2$]和[$t_{alpha}/2$, 正无穷]
- 交叉验证t检验
    - 对A,B两个学习器的性能没有显著差别做假设检验. 因为使用了k次测试, 即使用t检验, 检验泛化均值为0.
    - 因为使用n轮m折交叉验证, 测试集独立性有影响, 则做了特殊处理.
- 成对t检验/paired t-test
    - 成对指学习器A,B测试集相同的测试结果成对处理
- McNemar检验
    - 利用学习器分类结果的差别, 验证两者性能是否相同, 即应e1=e01, |e01-e10|服从正态分布
    - |e01-e10|小于临界值则接受
- 列联表/contingency table
    - 学习器A,B之间的性能关系, 列A行B, 内容为正确,错误. A,B正确e00, A,B错误e00,A对B错e10, A错B对e01.

- Friedman检验
- 后续检验/post-hoc test
- Nemenyi后续检验

2.5 偏差与方差

- 偏差/bias
  - 期望输出和真实标记的差别, 学习算法本身的拟合能力
  - $bias^2 = (E_D(f(x;D))-y)^2$
- 方差/variance
  - 使用样本数相同的不同训练集产生的方差, 数据扰动造成的影响
  - $var = E_{D[f}(x;D)- E_D(f(x;D))]$, $E_{D对训练集D求期望}$, f(x;D)为在训练集D上训练的模型在x上的输出
- 噪声
  - 当前任务上任何学习算法所能达到的期望泛化误差的下界, 体现学习任务本身的难度
  - $noise^2 = E_D((yD-y)^2)$, yD为数据集标记
- 偏差-方差分解/bias-variance decomposition
  - E(f;D) 算法期望泛化误差 $= E_D((f(x;D)-yD)^2) = bias^2 + var + noise^2$,假设噪声期望为零 $E_D(y_D-y)=0$
  - 泛化误差为偏差,方差和噪声之和, 体现出由学习算法的能力, 数据的充分性以及学习难度之和
- 偏差-方差窘境/bias-variance dilemma
  - 学习器拟合不充分时, 训练数据的扰动不足以使学习器产生显著变化. 偏差主导泛化误差.
  - 学习器拟合充分时, 学习到了训练数据集本身的特性, 被训练集的扰动影响, 此时方差主导泛化误差.

3 第3章 线性模型

3.1 基本形式

- 线性模型/linear model
  - $f(x) = w^T*x+b$, 其中x为示例, w, b为参数

3.2 线性回归

- 序/order
  - 顺序, 如大小, 前后等.
  - 有序离散属性可连续化
- 欧式距离/Euclidean distance/欧几里得距离
  - $dist = ||X_1-X_{2||}^2$
- 均方误差/mse/平方误差/square loss
  - 很好的几何意义, 对应欧式距离
- 最小二乘法/least square method

    - 基于均方误差最小化来进行模型求解的方法
- 凸函数
    - 任意两点x1,x2满足f((x1+x2)/2)<=(f(x1)+f(x2))/2
    - 二阶导数在区间上非负
- 线性回归模型的最小二乘参数估计
    - $f(x) = x^T*(X^T*X)^{-1}*X^T*y$
    - 因为在现实情况中参数数量多于样本数,X^T*X不是满秩矩阵, 会引入正则项
- 闭式解/closed-form/解析解/analytical solution
    - 就是一些严格的公式,给出任意的自变量就可以求出其因变量,也就是问题的解
- 多元线性回归/multivariate linear regression
- 满秩矩阵/full-rank matrix
    - A为n阶方阵, r(A)为n(狭义)
    - 秩: 用初等行变换将矩阵A化为阶梯形矩阵, 则矩阵中非零行的个数就定义为这个矩阵的秩, 记为r(A)
- 正定矩阵/positive definite matrix
    - 设M是n阶方阵，如果对任何非零向量z，都有$z^T*M*z> 0$，就称M为正定矩阵
    - X^T*X为正定矩阵
    - 等价命题
        - M的特征值均为正
- 正则化项/regularization
    - 损失函数中对参数的约束项, 过拟合的处理方式
- 对数线性回归/log-linear regression
    - 输出标记在指数尺度上变化, 将输出标记的对数作为线性模型逼近的目标
- 广义线性模型/generalized linear model
    - 利用单调可回函数g(.)对输出标记做非线性映射, 然后利用线性模型拟合映射.
- 联系函数/link function
    - g(.)

3.3 对数几率回归

- 单位阶跃函数/unit-step function
    - y=0,z<0; y=0.5, z=0; y=1, z>0
- 替代函数/surrogate function
    - 一定程度上近似单位阶跃函数, 且单调可微
- 对数几率函数/logistic function
    - y = 1/(1+exp(-z))
- 几率/odds
    - y/(1-y), 视y为x为正例的可能性,1-y为x为反例的可能性, 两者的比值.
- 对数几率/log odds/logit
    - In(y/(1-y))
- 对数几率回归/logistic regression/logit regression/对率回归/逻辑回归
    - 函数: $In(y/1-y) = w^T*x + b$
    - 目的: 用线性回归模型的预测结果去逼近真实标记的对数几率

    - 求解: 利用最大似然法,得到损失函数, loss = 求和($-y_i*beta^T*x_i + In(1+exp(beta^T*x_i))$, loss为凸函数
- 极大似然法/maximum likelihood method
    - 假设每次实验独立, 使得所有实验的后验概率最大
    - 步骤: 1)写似然函数,2)似然函数求对数整理,3)求倒数,4)解似然方程
- 梯度下降法/gradient descent
    - $beta^j+1 = beta^j$ - alpha * 损失函数对beta求偏导
    - 又名最速下降法, 梯度是下降方向
- 牛顿法/Newton method
    - $beta^j+1 = beta^j$ - 一阶偏导/二阶偏导
    - 求极大极小值,基于偏导等于零,利用一阶泰勒展开式.
        - 求解f'(x+delta(x))=0时的x+delta(x).
        - 一阶泰勒展开式: f'(x+delta(x)) ~= f'(x) + f''(x)*delta(x)
        - delta(x) ~= - f'(x)/f''(x)
        - x+delta(x) ~= x - f'(x)/f''(x)
        - 初始x, 可迭代逐步得到真实的x+delta(x), 收敛条件|delta(x)| < sigma
- 泰勒展开式
    - $f(x+delta(x)) ~= f(x) + f'(x)*delta(x) + f''(x)*delta^2(x) + \dots$

3.4 线性判别分析

- 线性判别分析/Linear Discriminant Analysis/LDA/Fisher判别分析
    - 模型: 将样例投影到一条直线上
        - 直线是一维空间
    - 训练: 使得同类样例投影点尽可能接近, 异类样例的投影点尽可能远离
        - $X_{i示例集合}$, $u_{i均值向量}$, $sigma_{i协方差矩阵}$, i是类型
        - 同类样例投影点的协方差尽可能小, 即$w^T*sigma_0*w + w^T*sigma_1*w$
        - 类中心之间的距离尽可能大, 即$||w^T*u_0 - w^T*u1||_2^2$
        - 最大化: $J = ||w^T*u_0-w^T*u_{1||2}^2 / (w^T*sigma_0*w+w^T*sigma_1*w)$
            - $J = (w^T*S_b*w)/(w^T*S_w*w)$, $S_{b类间散度矩阵}$, $S_{w类内散度矩阵}$
            - $min(-w^T*S_b*w)$, s.t. $w^T*S_w*w = 1$
            - 利用拉格朗日乘子法, 求其中一个解, $w=S_w^{-1}*(u_0-u_1)$
                - 其中S_w^{-1}可通过奇异值分解求得, $S_w^{-1}= V*sigma^{-1}*U^T$
            - 从贝叶斯决策理论的角度, 当两类数据同先验, 满足高斯分布且协方差相等时,LDA可达到最优分类
    - 预测: 将新样本投影到直线上, 根据投影点位置判断类别
- 均值向量/u
    - 针对特征/变量
- 协方差矩阵/sigma
    - $x_{i随机变量}$, $s_{ij是sigma中的元素}$, $s_{ij} = E([x_i-E(x_i)][x_j-E(x_j)])$
    - $sigma = 求和_x((x-u)*(x-u)^T)$
- 类内散度矩阵/within-class scatter matrix
    - $S_w = sigma_0+sigma_1$

- 类间散度矩阵/between-class scatter matrix
  - $S_b = (u_0-u_1)*(u_0-u_1)^T$
- 广义瑞利商/generalized Rayleigh quotient
  - J是S_b和Sw的广义瑞利商
- 拉格朗日乘子法
  - 是一种寻找多元函数在一组约束下的极值的方法. 通过引入拉格朗日乘子, 可将d个变量和k个约束条件的最优化问题转化为具有d+k个变量的无约束优化问题求解.
  - 拉格朗日函数: L(x, lambda, mu) = f(x) + 求和($lambda_i*h_i(x)$) + 求和($mu_i*g_j(x)$)
    - 其中h(x)是等式约束, g(x)是不等式约束
    - 等式约束为拉格朗日函数求极值
    - 不等式约束转化为KKT条件: g(x)<=0, mu>=0, mu*g(x)=0
- 特征值分解
  - alpha*A = alpha*x
- 奇异值分解
  - 任意实矩阵A属于R_m*n都可以分解成: $A = U*Sigma*V^T$
    - U属于R^m*m, 是满足$U^T*U=I$的m阶酉矩阵(unitary matrix)
    - V属于R^n*n, 是满足$V^T*V=I$的n阶酉矩阵
    - Sigma属于R^m*n, 其中$sigma_{ii}$未非负实数, 其中位置为0, $sigma_{11}>=sigma_{22}>=\ldots>=0$
  - 非零奇异值的个数为A的秩
  - U列向量为左奇异向量, V列向量为右奇异向量
- 低秩矩阵近似/low-rank matrix approximation
  - 可用奇异值分解求解的问题
  - 给定秩为r的矩阵A, 欲求最优k秩近似矩阵A~, k<=r
    - min({$||A-A~||_F$, A~属于R^m*n}), s.t. rank(A~)=k: $A_k = U_k*Sigma_k*V_k^T$, $Sigma_{k为r}$-k个最小奇异值置0的Sigma, U_k, $V_{k为U}$, V只保留对应的列奇异向量
- 多分类LDA
  - 过程省略, 可记忆其可监督降维

3.5 多分类学习

- 拆解法
  - 将多分类任务拆为若干个二分类任务求解, 再将结果集成成最终结果
- 一对一/OvO
  - N个类别任意两两配对, 共N(N-1)/2个分类器
- 一对其余/OvR
  - N个类别每个类别与余下类别配对成二分类, 共N个分类器
- 多对多/MvM
  - 每次抽若干类作为正类,若干其他类为反类, 需要特殊设计.
- 纠错输出码/Error Correcting Output Codes, ECOC
  - 分为编码,解码两步: 编码是划分分类器, 使每个类别有唯一编码, 解码是将预测编码同每个类别的编码比较, 距离最小的类别为最终结果
- 编码矩阵/coding matrix
  - 为所有类别编码构成的矩阵, 常见形式有二元码和三元码

- 二元码
  - 矩阵元素有两种, 正例和反例
- 三元码
  - 矩阵元素有三种, 正例,反例和停用例
- 海明距离
  - 常用在信息编码中求距离
  - 两个代码在对应位上编码不同的位数称为海明距离/码距, 如10101和00110从第一位开始依次有第一位、第四、第五位不同，海明距离为3.

3.6 类别不平衡问题

- 类别不平衡/class-imbalance
  - 指分类任务中不同类别的训练样例数目差别很大的情况
- 再缩放/rescaling/再平衡/rebalance
  - 假设训练集是真实样本的无偏采样, 将观察几率设备预测几率的阈值
- 欠采样/undersampling/下采样/downsampling
  - 除掉一些样例多的类别的数据
- 过采样/oversampling/上菜样/upsampling
  - 对样例少的类别多采样一些
- 阈值移动/threshold-moving
  - 使用原始数据集, 但对应调整阈值

3.7 阅读材料

- 稀疏表示/sparse representation
  - 对问题获得稀疏性的解
- DAG/Directed Acyclic Graph
- 闭式解
  - 解析解
- 多标记问题/multi-label learning
  - 一个样本有多个标记

4 第4章 决策树

4.1 基本流程

- 决策树
  - 基于树形结构决策, 每个子结点是对某个属性的"测试"(单变量决策树)
- 分而治之/divide-and-conquer
  - 决策树流程遵循的策略
- 递归过程
  - 原理: 1调用了自身, 2退出机制
  - 思考方法: 参考归纳过程
  - 决策树的生成过程属于递归过程

- 先验分布/prior distribution
  - 是概率分布的一种. 与试验结果无关，或与随机抽样无关，反映在进行统计试验之前根据其他有关参数口的知识而得到的分布
- 后验分布/posterior distribution
  - 根据样本 X 的分布Pθ及θ的先验分布π(θ)，用概率论中求条件概率分布的方法,可算出在已知X=x的条件下,θ的条件分布 π(θ|x)。因为这个分布是在抽样以后才得到的，故称为后验分布

4.2 划分选择

- 纯度/purity
  - 决策树中结点包含的样本类别越集中, 纯度越高
  - 随着决策树划分的深入,希望分支结点的纯度越高

4.2.1 信息增益

- 信息熵/information entropy
  - 度量样本合集纯度最常用的一种指标
  - Ent(D) = -对k求和(p_k*$\log_2(p_k)$), p_k为第k类样本所占的比例, D为数据集合
  - Ent(D)值越小, D的纯度越高
- 信息增益/information gain
  - 利用属性a对样本集D进行划分可得
  - Gain(D, a) = Ent(D) - 对v求和$|D^v|/|D|$*Ent(D^v), 其中v是a的属性值, D^v是属性a为v的集合
  - 一般而言, 信息增益越大, 意味着使用属性a进行划分所获得的"纯度提升"越大
  - 对含较多属性值的属性有所偏好
- ID3决策树学习算法
  - 基于信息增益为准则来划分属性

4.2.2 增益率

- 增益率/gain ratio
  - 利用属性a对样本集D进行划分可得
  - $Gain_{ratio}$(D, a) = Gain(D,a)/IV(a)
    - IV(a) = -对v求和($|D^v|/|D|$*$\log_2(|D^v|/|D|)$)
    - IV(a)称为属性a的固有值/intrinsic value.
    - 属性a的可能取值越多, IV(a)的值通常会越大
  - 增益率准则对可取值数目较少的属性有所偏好
- C4.5算法
  - 先从候选划分属性中找到信息增益高于平均水平的属性, 再从中选择增益率最高的

4.2.3 基尼指数

- 基尼值
  - Gini(D) = 对k求和(对k'!=k的k'求和(p_k*$p_{k'}$)) = 1-对k求和(p_k^2)

    - 反映了从数据集D中随机抽取两个样本,其类别标记不一致的概率
    - Gini(D)越小, 则数据集D的纯度越高
- 属性a的基尼指数/gini index
    - $Gini_{index}(D, a)$ = 对v求和($|D^v|/|D|*Gini(D^v)$)
- CART决策树
    - 候选属性集合A中选择划分后基尼指数最小的属性作为最优划分属性

4.3 剪枝处理

- 剪枝/pruning
    - 决策树学习算法对付"过拟合"的主要手段
    - 剪掉过多的决策树分支
- 预剪枝/prepruning
    - 决策树生成过程中, 对每个结点在划分前先进行估计, 若当前结点的划分不能带来决策树泛化性能提升,则停止划分并将当前结点标记为叶结点
- 后剪枝/post-pruning
    - 先从训练集生成一颗完整的决策树, 然后自底向上地对非叶结点进行考察,若将该结点对应的子树替换成叶结点能带来决策树泛化能力提升,则替换

4.3.1 预剪枝

- 决策树桩/decision stump
    - 仅有一层划分的决策树
- 贪心
    - 每一步利用局部最优

4.3.2 后剪枝

4.4 连续与缺失值

4.4.1 连续值处理

- 二分法/bi-partition
    - 一种连续属性离散化的技术

4.4.2 缺失值处理

- 处理两个问题
    - 属性值缺失情况下进行划分属性选择
        - 利用没有缺失属性值的集合计算指标, 但缺失属性集合有权重(指标已经被规范化)
    - 给定划分属性, 该样本在该属性上缺失, 如何划分
        - 分配到所有子树, 以训练集缺失集合权重的分配概率

4.5 多变量决策树

- 轴平行/axis-parallel
  - 决策树分类边界由若干个与坐标轴平行的分段组成
- 多变量决策树/multivariate decision tree/斜决策树/oblique decision tree
  - 一个非叶结点是对属性的 **线性组合** 进行测试
  - 划分边界可为斜线
- 单变量决策树/univariate decision tree
  - 非叶结点是对单个属性进行测试

4.6 阅读材料

- 增量学习/incremental learning
  - 接收到新样本后可对已学得的模型进行调整,而不用完全重新学习

5 第5章 神经网络

5.1 神经元模型

- 神经网络/neural networks
  - 神经网络是由 **具有适应性** 的 **简单单元** 组成的 **广泛并行互联的网络**, 它的组织能模拟生物神经系统对真实世界物体所做出的交互反映.
- 神经网络学习
  - 机器学习和神经网络这两个学科领域的交叉部分
- 神经元/neuron
  - 神经网络中的简单单元
  - 与其他神经元相连, 神经元内电位超过阈值, 将向相连的神经元传递化学物质, 改变它们的电位
- M-P神经元模型/阈值逻辑单元/threshold logic unit
  - 输入: n个其他神经元的加权输出
  - 输出: 输入同阈值比较经激活函数
- 激活函数/activation function
  - 将神经元输出映射到 激活/1, 抑制/0, 两种状态
  - 理想激活函数为 阶跃函数
- 挤压函数/squashing function/Sigmoid函数
  - 阶跃函数不连续,不光滑, 为了神经网络有更好的数学性质, 使用Sigmoid函数作为常用激活函数

5.2 感知机与多层网络

- 感知机/perceptron
  - 两层神经元组成, 输入接外界输入信号, 输出层是M-P神经元
- 哑结点/dummy node
  - ??
- 学习率/learning rate
- 收敛/converge

- 振荡/fluctuation
- 隐含层/hidden layer
  - 输入层和输出层之间的神经元层
- 多层前馈神经网络/multi-layer feedforwad neural networks
  - 每层神经元与下一层神经元互连, 神经元之间不存在同层连接, 也不存在跨层连接
- 连接权/connection weight

5.3 误差逆传播算法

- 误差逆传播算法/error backPropagation/BP
  - 基于梯度下降
  - 基于链式法则, 从输出反向计算梯度
- 标准BP算法
  - 每次仅对一个训练样例更新连接权和阈值
- 累计误差逆传播算法/accumulated error backpropagation
  - 每次对整个训练集样例更新连接权和阈值
- 一轮/one round/one epoch
  - 读取训练集一遍
- 随机梯度下降/stochastic gradient descent/SGD
  - 每次读取样本数少于训练集样本数
- 标准梯度下降
  - 每次读取整个训练集
- 早停/early stopping
  - 将数据集分成训练集和验证集, 训练集用于计算梯度, 更新连接权和阈值, 验证集用来估计误差, 若训练集误差降低但验证集误差升高, 则停止训练, 同时返回具有最小验证集误差的连接权和阈值.
- 正则化/regularization
  - 基本思想是在误差目标函数中增加一个用于描述网络复杂度的部分, 例如连接权和阈值的平方和, 以达到优化时降低网络复杂度, 使网络输出更加"光滑"的目的.

5.4 全局最小与局部最小

- 局部最小/local minimum
- 全局最小/global minimum
- 跳出局部最小的技术(缺乏理论保障)
  - 以不同初始点训练多个模型
  - 模拟退火
  - 随机梯度下降
  - 遗传算法
- 模拟退火/simulated annealing
  - 在每一步都以一定概率接受比当前解更差的结果, 但随着迭代的推进, 接受"次优解"的概率要变低, 保证算法稳定.
- 遗传算法/genetic algorithms
  - 父,母,遗传,变异

5.5 其他常见的神经网络

5.5.1 RBF网络

- RBF网络/radial basis function networks/径向基函数网络
  - 结构
    - 前馈神经网络
    - 隐层神经元激活函数为径向基函数
      - 径向基函数为: p(x, c), 样本x到数据中心c之间的欧式距离的单调函数, c为必然存在的参数, 还可以有距离的线性参数
      - 常用的高斯径向基函数形如: $p(x, c) = \exp(-\beta*||x-c||^2)$
    - 输出层是隐层神经元输出的线性组合
  - 训练
    - 第一步, 确定神经元c, 常用的方法包含随机采样, 聚类等
    - 第二步, 利用BP算法等来确定参数(连接权, 径向基线性参数)

5.5.2 ART网络

- 竞争型学习/competitive learning
  - 神经网络中一种常用的无监督学习策略, 使用该策略时, 网络的输出神经元相互竞争, 每一时刻仅有一个竞争获胜的神经元被激活, 其他神经元的状态被抑制.
- 胜者通吃/winner-take-all
- ART/adaptive reasonance theory/自适应谐振理论
  - 竞争学习的代表
  - 由比较层, 识别层, 识别阈值和重置模块构成
    - 比较层负责接收输入样本, 并将其传递给识别层神经元
    - 识别层每个神经元对应一个模式类, 神经元数目可在训练过程中动态增长以增加新的模式类
      - 识别层神经元需要产生获胜神经元, 竞争最简单的方式是, 计算输入向量与每个识别层神经元所对应的模式类之间的距离, 距离最小获胜.
      - 获胜神经元发送信号抑制其他神经元
      - 若输入向量与获胜神经元所对应的向量大于阈值, 则网络的连接权更新, 接收到类似输入时获胜神经元的相似度更大.
      - 若不大于阈值, 识别层新增神经元, 其代表向量就设为当前输入向量.
  - 识别阈值影响重大, 当识别阈值较高时, 输入样本分类较多, 模式精细. 反之, 较粗.
- 可塑性-稳定性窘境/stability-plasticity dilemma
  - 竞争型学习常见的窘境
  - 可塑性: 神经网络要有学习新知识的能力
  - 稳定性: 神经网络在学习新知识时要保持对旧知识的记忆
  - ART算法比较好的缓解了竞争型学习中的"可塑性-稳定性窘境"
- 在线学习/online learning
- 批模式/batch-mode

5.5.3 SOM网络

- SOM/self-organizing map/自组织映射/self-organizing feature map/自组织特征映射
  - 一种竞争学习型的无监督神经网络, 它能将高维输入数据映射到低维空间(通常二维),同时保持输入数据在高维空间的拓扑结构, 即高维空间中相似的样本点映射到网络输出层中的临近神经元
  - 输出层神经元以矩阵方式排列在二维空间中, 每个神经元都拥有一个权向量, 网络在接收向量后, 将会确定输出层获胜神经元, 它决定了该输入向量在低维空间中的位置.
  - 训练目的: 为每个输出层神经元找到合适的权向量, 以达到保持拓扑结构的目的
  - 训练过程: 在接收到一个训练样本后, 每个输出层神经元会计算该样本与自身携带的全向量之间的距离, 距离最近的神经元成为竞争获胜者, 称为最佳匹配单元. 然后, 最佳匹配单位及其临近神经元的权向量将被调整, 以使得这些权向量与当前输入样本的距离缩小. 这个过程不断迭代, 直至收敛.

5.5.4 级联相关网络

- 结构自适应神经网络/构造性神经网络/constructive networks
  - 将网络结构也当作学习的目标之一, 并希望能在训练过程中找到最符合数据特点的网络结构.
- 级联相关网络/cascade-correlation networks
  - 两个主要成分: 级联, 相关
    - 级联: 建立层次连接的层次结构
    - 相关: 通过最大化新神经元的输出和网络误差之间的相关性来训练相关的参数
  - 训练: 开始时, 网络只有输入层和输出层, 处于最小拓扑结构；随着训练的进行, 新的隐层神经元逐渐加入, 从而建立起层级结构. 当新的隐层神经元加入时, 其输入端连接权值是冻结固定的.
  - 特点: 无需设置网络层数, 隐层神经元数目；训练速度快；数据较少时易陷入过拟合

5.5.5 Elman 网络

- 递归神经网络/recurrent neural networks/recursive neural networks
  - 允许网络中出现环形结构, 从而可让一些神经元的输出反馈回来作为输入信号.
  - 这样的结构与信息反馈过程, 使得网络在t时刻的输出状态不仅和t时刻的输入有关, 还与t-1时刻的网络状态有关, 从而能处理与时间有关的动态变化
- Elman网络
  - 结构与多层前馈网络很相似, 但隐层神经元的输出被反馈回来, 与下一时刻输入层神经元提供的信号一起, 作为隐层神经元在下一时刻的输入. 隐层神经元通常采用sigmoid激活函数, 网络的训练通过推广的BP算法进行

5.5.6 Boltzmann机

- 能量/energy
  - 神经网络中有一类模型是为网络状态定义一个"能量", 能量最小化时网络达到理想状态, 而网络的训练就是在最小化这个能量函数

- 基于能量的模型/energy-based model
- Boltzmann机
  - 一种基于能量的模型
  - 常见结构: 神经元分为两层: 显层和隐层
    - 显层: 用于数据的输入和输出
    - 隐层: 被理解成数据的内在表达
  - 神经元都是布尔型的, 只有0,1两种状态
  - 能量定义为: E(s) = -对i求和$_i$(0->n-1)(对j求和$_j$(i+1->n)(w_{ij}*s_i*s_j)) - 对i求和$_i$(1->n)($sita_i$*s_i)
    - s表示n个神经元的状态{0, 1}
    - w_{ij}表示神经元i和j之间的连接权
    - $sita_i$表示神经元i的阈值
    - 若网络中的神经元以任意不依赖于输入值的顺序进行更新, 则网络最终将达到Boltzmann分布/平衡态, 此时状态向量s出现的概率将仅由其能量与所有可能状态向量的能量确定:
      - 概率/P(s) = exp(-E(s))/对t求和(exp(-E(t)))
    - 训练: 将每个训练样本视为一个状态向量, 使其出现的概率尽可能大.
    - 难点: 如何得到所有可能的状态向量
    - 受限的Boltzmann机/restricted boltzmann machine/RBM
      - 结构中仅隐层和显层之间连接
    - 对比散度/contrastive divergence/CD算法
      - 令v和h分别表示显层和隐层的状态向量, 则由于同一层内不存在连接, 有
        - P(v|h) = 对i求乘积(P($v_{i|h}$))
        - P(h|v) = 对j求乘积(P($h_{j|v}$))
      - 对每个训练样本v, 先计算出隐层神经元状态的概率分布, 然后根据这个概率分布采样得到h, 从h产生v',可以从v'产生h'
        - 连接权的更新公式: $delta_w$ = sigma*(v*h^T - v'*h'^T)
        - 阈值更新

5.5.7 深度学习

- 容量/capacity
  - todo 参见12章
- 深度学习/deep learning
  - 典型: 很深层的神经网络
- 发散/diverge
  - 与收敛相对的概念
  - 训练无法收敛, 训练次数n –> 无穷, 训练误差 -/-> 某个值
- 无监督逐层学习/unsupervised layer-wise training
  - 基本思想: 预训练+微调
- 预训练/pre-training
  - 每次训练一层的隐结点, 训练时将上一层的隐结点的输出作为输入, 而本层隐结点的输出作为下一层隐结点的输入.
- 微调/fine-tuning

- 深度信念网络/deep belief network/DBN
  - 每一层是一个受限的Boltzmann机
  - 整个网络可视为若干个RBM堆叠
  - 利用无监督逐层学习
- 预训练+微调
- 权共享/weight sharing
  - 让一组神经元使用相同的连接权
  - 代表: 卷积神经网络
- 卷积神经网络/convolutional neural network/CNN
  - 包含: 卷积层和采样层对输入信号进行加工, 连接层实现与输出目标之间的映射
  - 每个卷积层包含多个特征映射/feature map
    - 每个特征映射是由多个神经元构成的"平面", 通过一种卷积滤波器提取输入的一种特征
  - 采样层亦称汇合层/pooling, 其作用是基于局部相关性进行亚采样, 从而减少数据量的同时保留有用信息.
- 特征学习/feature learning/表示学习/representation learning
  - 深度神经网络经过多层处理, 逐渐将初始的"低层"特征表示转化为"高层"表示后, 用"简单模型"即可完成复杂的分类等学习任务. 可将这一过程理解为特征学习
- 特征工程/feature engineering
  - 描述样本的特征由人类专家来设计

6 第6章 支持向量机

6.1 间隔与支持向量

- 超平面
  - 在数学中，超平面(Hyperplane)是n维欧氏空间中，余维度为1的子空间。即超平面是n维空间中的n-1维的子空间。它是平面中的直线、空间中的平面之推广。
  - n维空间中的超平面可定义为：线性函数 $w^T*x=b$，其中w，x为n维向量，w不全为零
- 欧氏空间
  - 欧几里得空间
  - 一句话总结：欧几里得空间就是在对现实空间的规则抽象和推广（从n<=3推广到有限n维空间）。
  - 欧几里得几何就是中学学的平面几何、立体几何，在欧几里得几何中，平行线任何位置的间距相等。
- 划分超平面
  - 即超平面, 样本空间中任意点到超平面的距离：$r = |w^T*x+b|/||w||$
  - 可用于二分类划分
    - $w^T*x_i + b >= +1, y_i = +1$
    - $w^T*x_i + b <= -1, y_i = -1$
- 支持向量/support vector
  - 距离划分超平面最近的样本，即满足上述等式的样本
- 间隔/margin

    - 两个异类支持向量到超平面的距离：gama = 2/||w||
- 支持向量机/Support Vector Machine/SVM基本型
    - 满足训练样本超平面可分，最大化间隔，求解w，b
    - 即
        - min $1/2*||w||^2$
        - s.t. $y_i(w^T*x_i + b) >= 1$, i=1,2,…,m

6.2 对偶问题

- 线性规划/Linear Programming问题
    - 研究线性约束条件下线性目标函数的极值问题的数学理论和方法
- 二次规划问题
    - 特殊类型的优化问题
    - 一个有n个变数与m个限制的二次规划问题可以用以下的形式描述。
        - 首先给定：
            - 一个n 维的向量 c
            - 一个n × n 维的对称矩阵Q
            - 一个m × n 维的矩阵A
            - 一个m 维的向量 b
        - 则此二次规划问题的目标即是在限制条件为
            - A*x <= b
        - 找一个n 维的向量 x
        - 最小化
            - $f(x) = (1/2)*x^T*Q*x + c^T*x$
- 凸二次规划/convex quadratic programming问题
    - 如果Q是半正定矩阵，那么f(x)是一个凸函数, 此时为凸二次规划问题
    - 此时若约束条件定义的可行域不为空，且目标函数在此可行域有下界，则该问题有全局最小值。
- 对偶问题/dual problem
    - 对偶问题：每一个规划问题都伴随有另一个规划问题，称为对偶问题。
    - 原来的线性规划问题则称为原始线性规划问题，简称原始问题。
    - 对偶问题有许多重要的特征, 它的变量能提供关于原始问题最优解的许多重要资料，有助于原始问题的求解和分析。
    - 对偶问题与原始问题之间存在着下列关系：
        - ①目标函数对原始问题是极大化，对偶问题则是极小化。
        - ②原始问题目标函数中的收益系数是对偶问题约束不等式中的右端常数，而原始问题约束不等式中的右端常数则是对偶问题中目标函数的收益系数。
        - ③原始问题和对偶问题的约束不等式的符号方向相反。
        - ④原始问题约束不等式系数矩阵转置后即为对偶问题的约束不等式的系数矩阵。
        - ⑤原始问题的约束方程数对应于对偶问题的变量数，而原始问题的变量数对应于对偶问题的约束方程数。
        - ⑥对偶问题的对偶问题是原始问题，这一性质被称为原始和对偶问题的对称性。
- SMO/Sequential Minimal Optimization

6.3 核函数

- 核技巧/kernel trick
  - 将样本映射到特征空间后，其内积可用样本输入到核函数中计算。即$k(x_i, x_j) = phi(x_i)^T * phi(x_j)$
- 核函数/kernel function
  - k(.,.)
- 支持向量展式/support vector expansion
  - 使用核函数带入到支持向量求解问题
- 核函数定理
  - 令X为输入空间, k(.,.)是定义在X * X上的对称函数, 则k是核函数，当且仅当
    - 任意输入数据$D = [x_1, …, x_n]$, 核矩阵是半正定的
- 对称函数
  - f(x,y) = f(y,x)
- 核矩阵/kernel matrix
  - $K = matrix\{k_i,j\}$
    - $k_i,j = k(x_i, x_j)$
- 再生核希尔伯特空间/Reproducing Kernel Hilbert Space/RKHS的特征空间
  - 希尔伯特空间
    - 在数学裡，希尔伯特空间（英語：Hilbert space）即完备的内积空间，也就是一個帶有內積的完備向量空間。
    - 希尔伯特空间是有限维欧几里得空间的一个推广，使之不局限于實數的情形和有限的维数，但又不失完备性（而不像一般的非欧几里得空间那样破坏了完备性）
  - 由核函数隐式定义
- 常见核函数
  - 线性核：$k(x_i, x_j) = x_i^T*x_j$
  - 多项式核：$k(x_i, x_j) = (x_i^T*x_j)^d$
  - 高斯核：$k(x_i, x_j) = exp(-||x_i-x_{j||}^2/(2*sigma^2))$, sigma > 0
  - 拉普拉斯核： $k(x_i, x_j) = exp(-||x_i-x_{j||}/sigma)$, sigma >0
  - Sigmoid核：$k(x_i, x_j) = tanh(beta*x_i^T*x_j + sita)$
- 核函数性质
  - k1,k2是核函数，则对于任意正数gama1，gama2，其线性组合，gama1*k1+gama2*k2也是核函数
  - k1,k2是核函数，则核函数的直积k1(.)k2(x,z) = k1(x,z)*k2(x,z)
  - k1为核函数，则对于任意函数g(x), k(x,z) = g(x)*k1(x,z)*g(z)也是核函数

6.4 软间隔与正则化

- 软间隔/soft margin
  - 功能：允许支持向量机在一些样本上出错
  - 优化目标：min（$1/2||w||^2 + C*l_{01}(y_i*(w^T*x_i+b)-1)$),
    - 其中$l_{01}(z)$
      - 1, if z < 0
      - 0, otherwize

        - 当C为无穷大时，软间隔同硬间隔，C为有限值时，允许一些样本不满足约束
- 硬间隔/hard margin
    - 要求所有样本都必须划分正确
- 代替损失/surrogate loss函数
    - 因为l_{01}非凸，非连续，数学性质不太好，则人们使用其他数学性质较好，同l01同功能(惩罚划分错误)的函数
- 常见代替损失
    - hinge损失：$l_{hinge}(z) = max(0, 1-z)$
    - 指数损失/exponential loss: $l_{exp}(z) = exp(-z)$
    - 对率损失/logistic loss: $l_{log}(z) = log(1+exp(-z))$
- 松弛变量/slack variable
    - 损失函数改写成一个变量
- 软间隔支持向量机
    - 优化目标：min（$1/2*||w||^2$ + C对i求和($sigma_i$)）
    - s.t. $y_i*(w^T*x_i + b) >= 1 - sigma_i$, $sigma_i >= 0$
- 结构风险/structural risk
    - 优化函数中，用于描述函数f的某些性质
    - 类似于正则化的功能，引入领域知识和用户意图，减少过拟合风险
- 经验风险/empirial risk
    - 优化函数中， 用于描述模型与训练数据的契合程度
- 正则化/regularization

6.5 支持向量回归

- 支持向量回归/Support Vector Regression/SVR
    - 与传统回归问题的不同
        - 容忍f(x)与y之间最多有sigma的偏差，即仅当f(x)与y之间的差别绝对值大于sigma时才计算损失
    - 优化问题
        - min（$1/2*||w||^2$ + C*对i求和($l_{sigma}(f(x_i)-y_i)$)
- sigma-不敏感损失/sigma-insensitive loss
    - l_{sigma}
        - 0, if |z|<=sigma
        - |z|-sigma, otherwise

6.6 核方法

- 表达定理/representer theorem
    - 条件
        - 令H为核函数k对应的再生核希尔伯特空间
        - $||h||_H$表示H空间中关于h的范数
        - 对于任意单调递增函数g：[0, 正无穷] –> R
        - 任意非负损失函数l：R^m –> [0, 正无穷]

    - 优化问题
        - min F(h) = g($||h||_H$) + l(h(x_1),h(x_2),…,h(x_m))
    - 解
        - $h^*(x)$ = 对i求和($alpha_i$*k(x,x_i))
- 核方法/kernel methods
    - 基于核函数的学习方法
    - 常见，通过核函数将线性学习器拓展为非线性学习器
- 核化
    - 引入核函数
- 核线性判别分析/Kernelized Linear Discriminant Analysis/KLDA
    - 假设
        - g：X–>F 将样本映射到特征空间F
        - 在F中执行线性判别分析，求h(x) = w^T*g(x)
    - 利用线性判别分析和表达定理，求解alpha和h

6.7 阅读材料

- LIBSVM
    - SVM著名的软件包

7 第7章 贝叶斯分类器

7.1 贝叶斯决策论

- 贝叶斯决策论/Bayesian decision theory
    - 概率框架下实施决策的基本方法
    - 对分类任务来说，在所有相关概率都已知的理想情况下，贝叶斯决策论考虑如何基于这些概率和误判损失来选择最优的类别标记
- 期望损失/expected loss/风险/risk
    - 将x分类为c_i所产生的期望损失
    - R($c_{i|x}$) = 对j求和($lambda_{ij}$ * P($c_{j|x}$))
        - 有N中可能的类别，Y = {c_1, c_2, …, c_N}
        - $lambda_{ij}$是将一个真实标记为c_j的样本误分类为c_i产生的损失
- 总体风险
    - R(h) = $E_{x[R}$(h(x)|x)]
        - 判定准则h：X –> Y
- 贝叶斯判定准则/Bayes decision rule
    - 为最小化总体风险，只需在每个样本上选择哪个能使条件风险R(c|x)最小的类别标记，即
        - $h^*(x) = argmin_c(R(c|x))$
    - 使用此准则最小化决策风险，首先要获得后验概率P(c|x)
- 贝叶斯最优分类器/Bayes optimal classifier
    - 上面的h^*
- 贝叶斯风险/Bayes risk

    - 贝叶斯最优分类器对应的总体风险$R(h^*)$
    - 机器学习所能产生模型的风险下限
- 判别式模型/discriminative models
    - 估计$P(c|x)$的方法
    - 直接建模$P(c|x)$来预测c
- 生成式模型/generative models
    - 估计$P(c|x)$的方法
    - 先对联合概率分布$P(x, c)$建模，然后再由此获得$P(c|x)$
        - $P(c|x) = P(x,c)/P(x)$
        - $P(c|x) = P(c)*P(x|c)/P(x)$
- 先验/prior概率
    - $P(c)$
- 条件概率/class-conditional probability/似然/likelihood
    - $P(x|c)$，样本x相对于类标签c的类条件概率
- 证据/evidence因子
    - $P(x)$, 用于归一化

7.2 极大似然估计

- 参数估计/parameter estimation
    - 概率模型的训练过程
        - 估计类条件概率的常用策略是先假设其固有某种确定的概率分布形式，再基于训练样本对概率分布的参数进行估计
- 频率主义学派/Frequentist
    - 认为参数虽然未知，但却是客观存在的固定值；因此，可以通过优化似然函数等准则来确定参数值
- 贝叶斯学派/Bayesian
    - 认为参数是未观察到的随机变量，其本身也可有分布；因此，可假定参数服从一个先验分布，然后基于观测到的数据来计算参数的后验分布
- 极大似然估计/Maximum Likelihood Estimation/MLE
    - 步骤
        - 假设样本独立同分布，得到对数似然函数
        - 通过最大化对数似然函数，求参数
    - 优点
        - 简单
    - 缺点
        - 结果准确性严重依赖假设的概率分布
- 似然函数
    - $P(D_{c|sitac})$ = 对属于D_c的x求乘积$(P(x|sita_c))$
- 对数似然/log-likelihood
    - 对似然函数求对数
    - $LL(sita_c) = log(P(D_{c|sitac}))$

7.3 朴素贝叶斯分类器

- 朴素贝叶斯分类器/naive Bayes classifier
  - 假设
    - 因为，条件概率P(x|c)是所有属性上的联合概率，难以从有限的训练样本中直接估算
    - 所以，采用“属性条件独立性假设”
  - 表达式
    - $h_{nb}(x)$ = argmax(P(c)*对i求乘积($P(x_{i|c})$))
      - P(c) = $|D_c|/|D|$, 当独立同分布样本充足
      - 离散属性
        - $P(x_{i|c}) = |D_c,x_i|/|D_{c|}$
      - 连续属性
        - 假设$p(x_{i|c})$ ~ 高斯分布$N(u_c,i,sigma^2_c,i)$
        - 高斯分布参数使用统计方式求解
- 属性条件独立性假设/attribute conditional independence assumption
  - P(c|x) = P(c)*P(x|c)/P(x) = P(c)/P(x)*对i求乘积($P(x_{i|c})$)
- 平滑/smoothing
  - 为了避免其他属性携带的信息被训练集中 **未出现的属性值** ”抹去“，在估计概率时通常要进行“平滑”
  - 常用拉普拉斯修正
- 拉普拉斯修正/Laplacian correction
  - 修正结果
    - P(c) = ($|D_{c|}$+1)/(|D|+N)
      - N表示训练集D中可能的类别数
    - $P(x_{i|c})$ = ($|D_c$,xi|+1)/($|D_{c|}$+N_i)
      - N_i表示第i个属性可能的取值数
- 懒惰学习/lazy learning
  - 先不进行任何训练，待收到预测请求时再根据当前数据集进行估值

7.4 半朴素贝叶斯分类器

- 半朴素贝叶斯分类器/semi-naive Bayes classifiers
  - 解决问题
    - 属性条件独立性假设，通常难以成立
  - 基本想法
    - 适当考虑一部分属性间的相互依赖信息，从而既不需要进行完全联合概率计算，又不至于彻底忽略了比较强的属性依赖关系
- 独依赖估计/One-Dependent Estimator/ODE
  - 半朴素贝叶斯分类器中最常用的一种策略
  - 每个属性在类别之外最多仅依赖一个其他属性
    - P(c|x) 同比 P(c)*对i求乘积($P(x_{i|c},pa_i)$)
      - pa_i为属性xi所依赖的属性，成为xi的父属性
- 超父/super-parent
  - 最直接的做法是假设所有属性依赖同一属性，此属性成为超父

- SPODE/Super-Parent ODE
  - 利用交叉验证等模型选择方法来确定超父属性
- TAN/Tree Augmented naive Bayes
- 最大带权生成树/maximum weighted spanning tree
- 条件互信息/conditional mutual information
- AODE/Averaged One-Dependent Estimator

7.5 贝叶斯网

- 贝叶斯网/Bayesian network/信念网/belief network
  - 借助有向无环图来刻画属性之间的依赖关系, 并使用条件概率表(假定所有属性均为离散型)来描述属性的联合概率分布
  - 由结构G和参数sita组成, 即B=<G,sita>
    - G是一个有向无环图
      - 两个属性有直接依赖关系则两者相连
      - 有效地表达了属性间的条件独立性.
        - 给定父结点集, 贝叶斯网假设每个属性与它的非后裔属性独立
        - $P_B(x_1, x_2, \ldots, x_d)$ = 对i求乘积($P_B(x_{i|pii})$) = 对i求乘积($sita_{xi|pii}$)
    - sita是定量描述这种依赖关系
      - 假设属性$x_{i在G中的父结点集是pii}$, 则sita包含了每个属性的条件概率表$sita_{xi|pii} = P_B(x_{i|pii})$
- 有向无环图/Directed Acyclic Graph/DAG
- 条件概率表/Conditional Probability Table/CPT

7.5.1 结构

- 同父/common parent结构
  - $x_1 \rightarrow x_3$, $x_1 \rightarrow x_4$
    - 给定$x_{1的取值}$, 则x_3,$x_{4独立}$
- 顺序结果
  - $z \rightarrow x$, $x \rightarrow y$
    - 给定x的值, y与z条件独立
- V型结构/V-structure
  - $x_1 \rightarrow x_4$, $x_2 \rightarrow x_4$
    - 给定子结点$x_{4的取值}$,x_1,$x_{2必不独立}$
    - $x_{4的取值完全未知}$, 则V型结构下$x_{1与x2是相互独立的}$
- 边界独立性/marginal independence
  - 对变量做积分或求和亦称为边际化
  - 通过边际化得到的独立关系
    - 比如
      - V型结构x_1,$x_{2对x4积分得到独立关系}$
      - 同父结构, x_3, $x_{4无法对x1积分得到独立关系}$
- 有向分离/D-separation
  - 把有向图转化为无向图

        - 找到有向图中所有的V型结构, 在V型结构的两个父结点之间加上一条无向边
        - 将所有的有向边改为无向边
- 道德图/moral graph
    - 有向分离得到的无向图称为道德图
    - 基于道德图能直观,迅速地找到变量间的条件独立性.
        - 假定道德图中有变量x,y和变量集合$z=\{z_i\}$, 若变量x和y在图上被z分开, 即从道德图中将变量集合z去除后, x和y分属两个连通分支, 则称变量x和y被z有向分离, x独立y|z成立
- 道德化/moralization
    - 将父结点相连的过程称为道德化

7.5.2 学习

- 评分搜索
    - 根据训练数据来找出结构最恰当的贝叶斯网
- 评分函数/score function
    - 用于评估贝叶斯网与训练数据的契合程度
    - 通常基于信息论准则, 将学习问题看作一个数据压缩任务
        - 学习的目标是找到一个能以 **最短编码长度** 描述训练数据的模型
            - 此时编码长度包含了描述模型自身所需的编码位数 和 使用该模型描述数据所需要的编码位数
    - 给定训练集$D=\{x_1,\dots x_m\}$, 贝叶斯网B=<G,sita>在D上的评分函数可写为
        - s(B|D) = f(sita)|B|-LL(B|D)
            - |B|是贝叶斯网的参数个数
            - f(sita)表示描述每个参数sita所需要的编码位数
            - LL(B|D) = 对i求和($\log(P_B(x_i))$), 贝叶斯网的对数似然
            - 第一项是计算编码贝叶斯网B所需要的编码位数
            - 第二项是计算B所对应的概率分布$P_{B对D描述得有多好}$
- 信息论准则
- 最小描述长度/Minimal Description Length/MDL准则
    - 选择综合编码长度(描述网络和编码数据)最短的贝叶斯网
        - 对贝叶斯学习而言, 模型就是一个贝叶斯网
        - 同时, 每个贝叶斯网描述了一个在训练数据上的概率分布, 自有一套编码机制能使那些经常出现的样本有更短的编码
- AIC/Akaike Information Criterion评分函数
    - f(sita) = 1, 即每个参数用1编码位描述
- BIC/Bayesian Information Criterion评分函数
    - f(sita) = 1/2*log(m), 即每个参数用1/2*log(m)编码位描述

7.5.3 推断

- 查询/query
    - 通过一些属性变量的观测值来推测其他属性变量的取值

- 推断/inference
  - 通过已知变量观测值来推测待查询变量的过程
- 证据/evidence
  - 已知变量观测值
- 吉布斯采样/Gibbs sampling
  - 输入
    - 贝叶斯网 B=<G,sita>
    - 采样次数T
    - 证据变量E及其值e
    - 待查询变量Q及其值q
  - 过程
    - n_q=0
    - q^0=对Q的随机赋初值
    - for t=1,2,…,T do
      - for $Q_{i属于Q}$ do
        - Z=E并Q\{Q_i}
        - z=e并q^t-1\q_i^t-1
        - 根据B计算分布$P_B(Q_{i|Z}=z)$
        - q_i^t=根据$P_B(Q_{i|Z}=z)$采样所获$Q_{i取值}$
        - q^t=将q^t-1中的q_i^t-1用$q_i^{t替换}$
      - end for
      - if q^t-q then
        - n_q=n_q+1
      - end if
    - end for
  - 输出
    - P(Q=q|E=e) ~=n_q/T
- 随机漫步/random walk
  - 每一步仅依赖前一步的状态
- 马尔科夫链/Markov chain
- 平稳分布/stationary distribution

7.6 EM算法

- 隐变量/latent variable
  - 未观测变量
- 边际似然/marginal likelihood
  - LL(sita|X,Z)=lnP(X,Z|sita)
    - sita模型参数, X已观测变量集, Z隐变量集
  - 边界似然: LL(sita|X) = lnP(X|sita)=ln对Z求和(P(X,Z|sita))
- EM/Expectation-Maximization算法
  - 常用估计参数隐变量的利器, 一种迭代方法
  - 基本想法
    - 若参数sita已知, 则可根据训练数据推断出最优隐变量Z的值(E步)

        - 若Z的值已知, 则可方便地对参数sita做做大似然估计(M步)
    - 原型
        - 以初始值$sita^{0为起点}$, 迭代执行一下步骤直至收敛
            - 基于$sita^{t推断隐变量Z的期望}$,记为Z^t
            - 基于已观测变量X和$Z^{t对参数sita做极大似然估计}$,记为$sita^t+1$
    - 如果基于$sita^{t计算隐变量Z的概率分布P}(Z|Z,sita^t)$, 而不是取Z的期望
        - 以当前参数$sita^{t推断隐变量分布P}(Z|X,sita^t)$, 并计算对数似然估计LL(sita|X,Z)关于Z的期望
            - $Q(sita|sita^t) = E_{Z|X,sita^t}(LL(sita|X,Z))$
        - 寻找参数最大化期望似然
            - $sita^t+1 = argmax(sita)(Q(sita|sita^t))$
- 坐标下降法

8 第8章 集成学习

8.1 个体与集成

- 集成学习/ensemble learning/多分类器系统/multi-classifier system/基于委员会的学习/committee-based learning
    - 假设
        - 个体学习器相互独立, 随着集成中个体分类器数目T的增大, 集成的错误率将指数下降, 最终归于零.(现实情况, 无法相互独立)
    - 目标
        - 个体学习器好而不同
    - 两类
        - 个体学习器间存在强依赖关系, 必须串联生成的序列化方法
            - 代表: Boosting
        - 个体学习器件不存在强依赖关系, 可同时生成的并行化方法
            - 代表: Bagging和随机森林
- 个体学习器/individual learner
- 同质/homogeneous
    - 集成学习只包含相同类型的个体学习器
- 基学习器/base learner
    - 同质集成中的个体学习器
- 基学习算法/base learning algorithm
    - 同质集成中的对应算法
- 异质/heterogeneous
    - 集成中包含不同类型的个体学习器
- 组件学习器/component learner
    - 异质集成中的个体学习器
- 弱学习器/weak learner
    - 常指泛化性能略优于随机猜想的学习器
- 投票法/voting

- 好而不同
  - 个体学习器有一定的准确性
  - 个体学习器有多样性, 即有差异性

8.2 Boosting

- Boosting
  - 工作机制
    - 先从初始训练集训练出一个基学习器, 再根据基学习器的表现对训练样本分布进行调整, 使得先前基学习器做错的训练样本在后续受到更多关注, 然后基于调整后的样本分布来训练下一个基学习器
    - 重复上步, 直至基学习器数目达到事先指定的值T, 最终将T个基学习器进行加权结合
- AdaBoost
  - 加性模型/additive model
    - 基学习器的线性组合
    - H(x) = 对t求和($alpha_t$*h_t(x))
  - 指数损失函数/exponential loss function
    - l_{exp}(H|D) = E_x~D[exp(-f(x)*H(x))]
  - 算法
    - 输入
      - 训练集: D = {(x1, y1), …, (xm, ym)}
      - 基学习算法Hl
      - 训练轮数T
    - 过程
      - D1(x) = 1/m
      - for t = 1,2,…,T do
        - h_t = Hl(D, Dt)
        - $epsion_t$ = P_x~Dt(h_t(x)!=f(x))
        - if $epsion_t$ > 0.5 then break
        - $alpha_t$ = 1/2*ln((1-$epsion_t$)/$epsion_t$)
        - Dt+1(x) =
          - Dt(x)/Zt * ?
            - ? = exp(-$alpha_t$), if h_t(x) = f(x)
            - ? = exp($alpha_t$), if h_t(x) != f(x)
          - Dt(x)*exp(-$alpha_t$*f(x)*h_t(x))/Zt
        - end for
    - 输出
      - F(x) = sign(对t求和($alpha_t$*h_t(x)))
    - 说明
      - Dt 是分布
      - Zt 是规范化因子
  - 重赋权法/re-weighting
    - 训练过程中的每一轮, 根据样本分布为每个训练样本重新赋予一个权重
  - 重采样法/re-sampling
    - 每一轮学习中, 根据样本分布对训练集重新进行采样
    - 可避免训练早停

    - 特点
        - 主要关注降低偏差, 能基于泛化性能相当弱的学习器构建出很强的集成

8.3 Bagging和随机森林

8.3.1 Bagging

- Bagging
    - 过程
        - 基于自助采样法, 获取T个m大小的数据集, 训练T个个体学习器
        - 个体学习器结合: 通常: 分类任务采用简单投票法; 回归任务使用简单平均法
    - 特点
        - 利用外包估计, 减小过拟合风险
        - 主要关注降低方差

8.3.2 随机森林

- 随机森林/random forest
    - 使用决策树为基学习算法
    - 以Bagging为基础
    - 基决策树学习过程中, 随机选择包含k个属性的子集, 然后再从这个子集中选择一个最优属性进行划分, k推荐log2(d)
    - 不仅样本扰动, 属性也扰动

8.4 结合策略

- 学习器结合的3方面好处(感觉表述不合理)
    - 从统计的方面来看, 由于学习任务的假设空间往往很大, 可能有多个假设在训练集上达到同等性能, 此时若使用单学习器可能因误选而导致泛化性能不佳, 结合多个学习器则会减小这一风险
    - 从计算的方面来看, 学习算法往往会陷入局部极小, 有的局部极小点所对应的泛化性能可能很糟, 通过多次运行之后进行结合, 可降低陷入糟糕局部极小点的风险
    - 从表示的方面来看, 某些学习任务的真实假设可能不再当前学习算法所考虑的假设空间中, 此时若使用单学习器则肯定无效, 二通过结合多个学习器, 由于相应的假设空间有所扩大, 有可能学得更好的近似

8.4.1 平均法

- 平均法/averaging
    - 个体学习器性能相差较大时宜使用加权平均法
    - 个体学习器性能相近时宜使用简单平均法
- 简单平均法/simple averaging
    - $H(x) = 1/T$*对i求和$(h_i(x))$
- 加权平均法/weighted averaging

- H(x) = 对i求和(w_i*h_i(x)), w_i>=0,对i求和(w_i)=1

8.4.2 投票法

- 投票法/voting
- 绝对多数投票法/majorityvoting
  - 某标记得票数超过半数, 则预测为该标记, 否则拒绝
- 相对多数投票法/plurality voting
  - 预测为得票数最多的标记, 若同时有多个标记获最高票, 则从中随机选取一个
- 加权投票法/weighted voting
  - 考虑权重的相对多数投票法/绝对多数投票法
- 硬投票/hard voting
  - 个体学习器对某个标记只能投{0,1}, 即类标记
- 软投票/soft voting
  - 个体学习器对某个标记可以投[0,1], 即类概率

8.4.3 学习法

- 学习法
  - 当训练数据很多时, 通过另一个学习器来结合的策略
- Stacking
  - 从初始数据集训练出初级学习器, 然后"生成"一个新的数据集用于训练次级学习器
  - 新数据集中, 初级学习器的输出被作为样例输入特征, 而初始样本的标记仍被当作样例标记
  - 利用初级学习器未使用的数据训练次级学习器, 避免过拟合
  - 将初级学习器的输出类概率作为次级学习器的输入属性, 用多响应线性回归作为次级学习算法效果较好
- 初级学习器
  - 个体学习器
- 次级学习器/元学习器/meta-learner
  - 用于结合的学习器
- 多响应线性回归/multi-response linear regression/MLR
  - 对每一个类分别进行线性回归,属于该类的训练样例所对应的输出被置于1,其他类置于0
  - 测试示例将被分给输出值最大的类
- 贝叶斯模型平均/Bayes Model Averaging/BMA
  - 基于后验概率来为不同的模型赋予权重, 可视为加权平均法的一种特殊实现
  - 同Stacking的比较
    - 理论上, 若数据生成模型恰在当前考虑的模型中, 且数据噪声很少,则BMA不差于Stacking
    - 现实中, 前提难以满足, Stacking通常优于BMA, 鲁棒性比BMA更好, 而且BMA对模型近似误差非常敏感

8.5 多样性

8.5.1 误差-分歧分解

- 分歧/ambiguity
  - $A(h_{i|x}) = (h_i(x)-H(x))^2$
  - 集成分歧
    - 考虑加权平均: **A** = 对i求和($w_i*A(h_{i|x})$)
- 误差
  - $E(h_{i|x}) = (f(x)-h_i(x))^2$
  - $E(H|x) = (f(x)-H(x))^2$
  - 加权平均: **E** = 对i求和($w_i*E(h_{i|x})$)
- 误差-分歧分解/error-ambiguity decomposition
  - E = **E** - **A**
  - 表明个体学习器准确性越高, 多样性越大, 则集成越好
  - 但推导结果目前只适用于回归学习, 难以直接推广到分类学习任务上去

8.5.2 多样性度量

- 多样性度量/diversity measure
  - 用于度量集成中的个体分类器的多样性.
  - 典型做法是考虑个体分类器的两两相似/不相似
- 结果列联表/contingency table
  - 假设二分类任务
  - | | hi = +1 | hi = -1 |
  - | -——— | -——— | -——— |
  - | hj = +1 | a | c |
  - | hj = -1 | b | d |
    - a 为hi,hj均预测为正类的样本数目, b,c,d以此类推, a+b+c+d = m
- 不合度量/disagreement measure
  - $dis_{ij} = (b+c)/m$
  - dis_{ij}值域为[0,1], 值越大则多样性越大
- 相关系数/correlation cofficient
  - $p_{ij} = (ad-bc)/sqrt((a+b)(a+c)(c+d)(b+d))$
  - p_{ij}的值域为[-1,1], 若hi和hj无关,则值为0;若hi与hj正向关则值为正,否则为负
- Q-统计量/Q-statistic
  - $Q_{ij} = (ad-bc)/(ad+bc)$
  - Q_{ij}与相关系数pij的符号相同, 且$|Q_{ij}|>=|p_{ij}|$
- k-统计量/k-statistic
  - $k = (p_1-p_2)/(1-p_2)$
  - 其中,p_1是两个分类器取得一致的概率;p_2是两个分类器偶然达成一致的概率,它们可由数据集D估算:
    - $p_1 = (a+d)/m$
    - $p_2 = [(a+b)(a+c)+(c+d)(b+d)]/m^2$
- k-误差图
  - 每一对分类器作为图上的一个点
  - 横坐标是这对分类器的k值, 纵坐标是这对分类器的平均误差

8.5.3 多样性增强

- 数据样本扰动
  - 给定初始数据集, 可从中产生不同的数据子集, 再利用不同的数据子集训练出不同的个体学习器.
  - 对"不稳定基学习器"很有效, 例如,决策树,神经网络等
  - "稳定基学习器"对数据扰动不敏感,例如,线性学习器,支持向量机,朴素贝叶斯,k临近学习器等,需要别的方法.
- 输入属性扰动
  - 从不同的属性"子空间"训练出个体学习器, 著名算法, 随机子空间/random subspace
- 输出表示扰动
  - 对输出表示进行操纵以增强多样性
    - 训练样本的类标记稍作变动, 如"翻转法/flipping output", 随机改变一些训练样本的标记
    - 对输出表示进行转化, 如"输出调制法/output smearing",将分类输出转化为回归输出后构建个体学习器
    - 将原任务拆解成多个可同时求解的子任务,如ECOC法, 利用纠错输出码将多分类任务拆解成一系列二分类任务来训练基学习器
- 算法参数扰动
  - 随机设置算法参数得到差异比较大的个体学习器, 例如负相关法/negative correlation显示地通过正则化相来强化个体神经网络使用不同的参数

8.6 阅读材料

- 集成修剪/ensemble pruning/选择性集成/selective ensenble/集成选择/ensemble selection
  - 在集成产生之后再试图通过去除一些个体学习器来获得较小的集成
    - 序列化集成, 减小集成规模后常导致泛化性能下降
    - 并行化集成在减小规模的同时可提升性能

9 第9章 聚类

9.1 聚类任务

- 无监督学习/unsupervised learning
- 聚类/clustering
- 簇/cluster

9.2 性能度量

- 性能度量/聚类"有效性指标"/validity index
- 簇内相似度/intra-cluster similarity
- 簇间相似度/inter-cluster similarity
- 外部指标/external index
- 内部指标/internal index

- Jaccard系数/Jaccard Coefficient/JC
- FM指数/Fowlkes and Mallows Index/FMI
- Rand指数/Rand Index/RI
- 考虑聚类结果的簇划分
- DB指数/Davies-Bouldin Index/DBI
- Dunn指数/Dunn Index/DI

10 TO-DO

- 过拟合处理技术
- 平均
- 常见距离计算方法及应用场景
- 常见凸函数
- 类别不平衡处理策略
- hessian矩阵
- 对偶问题(拉格朗日乘子法)
- 降维方法总结
- 连续值离散化技术
- 纯度度量的比较
- 跳出局部最小的技术
- 构造学习算法的策略
- 构造无监督学习的策略
- 收集更多的神经网络结构
- 节省训练开销的策略
- 无监督的常见策略
- Boltzmann能量什么含义
- 支持向量机与对率回归的比较
    - 使用$l_{\log}$作为替代函数
        - 支持向量机与对率回归目标相近，通常性能相当。
        - 对率回归其输出有自然的概率意义
        - 对率回归可直接多分类
    - hinge损失作为替代函数
        - 支持向量机的解具有稀疏性
        - 训练样本需求更少，避免过拟合
- 数据挖掘十大算法

Author: iwos-ml

Created: 2021-03-09 二 19:56

Emacs 25.3.1 (Org mode 8.2.10)