第一章

1.求版本空间。 习题 1.1

第二章

过拟合：将训练样本本身的特点　　当做所有样本的一般性质，导致泛化性能下降

- 优化目标加正则项
- early stop

欠拟合：对训练样本的一般性质尚未学好

- 决策树:拓展分支
- 神经网络:增加训练轮数

线性模型：不易过拟合

决策树：剪枝处理

神经网络：对于过拟合：早停（有一个验证集）或者 正则化（在**误差目标函数**中增加一个用于描述网络复杂度的部分）

正则化：可理解为一种"罚函数法,"即对不希望得到的结果施以惩罚，从而使得优化过程逐渐趋向于希望目标，从贝叶斯估计的角度来看，正则化项可认为是提供了模型的先验概率。

支持向量机：过拟合（核函数过于 powerful，软间隔时候，C 的参数过大时）

AUC 衡量了样本预测的排序质量。

偏差-方差分解"可以用来帮助解释泛化性能

- 偏差度量了学习算法期望预测与真实结果的偏离程度; 即**刻画了学习算法本身的拟合能力**;
- 方差度量了同样大小训练集的变动所导致的学习性能的变化; 即**刻画了数据扰动所造成的影响**;
- 噪声表达了在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; 即**刻画了学习问题本身的难度**。

偏差-方差窘境:

在训练不足时，学习器拟合能力不强，训练数据的扰动不足以使学习器的拟合能力产生显著变化，此时偏差主导泛化错误率;

随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导泛化错误率;

训练充足后，学习器的拟合能力非常强，训练数据的轻微扰动都会导致学习器的显著变化，若训练数据自身非全局特性被学到则会发生过拟合。

第三章

- 各任务下（回归、分类）各个模型优化的目标

最小二乘法：最小化均方误差

对数几率回归：最大化样本分布似然

线性判别分析：投影空间内最小（大）化类内（间）散度

- 参数的优化方法

最小二乘法：线性代数

对数几率回归：凸优化梯度下降、牛顿法

线性判别分析：矩阵论、广义瑞利商

第四章

Algorithm 1 决策树学习基本算法

输入:

- 训练集 $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A = \{a_1, \ldots, a_d\}$.

过程: **函数** TreeGenerate(D, A)

生成结点 node;
if D 中样本全属于同一类别 C **then**
  将 node 标记为 C 类叶结点; **return**
end if
if $A = \emptyset$ **OR** D 中样本在 A 上取值相同 **then**
  将 node 标记叶结点，其类别标记为 D 中样本数最多的类; **return**
end if
从 A 中选择最优划分属性 a_*;
for a_* 的每一个值 a_*^v **do**
  为 node 生成每一个分枝; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
  if D_v 为空 **then**
    将分枝结点标记为叶结点, 其类别标记为 D 中样本最多的类; **return**
  else
    以 TreeGenerate(D_v, $A - \{a_*\}$) 为分枝结点
  end if
end for

输出: 以 node 为根结点的一棵决策树

划分选择:

- 信息增益
- 增益率
- 基尼指数

第四章

感知机由两层神经元组成，输入层接受外界输入信号传递给输出层，输出层是 M-P 神经元

多层前馈神经网络：每层神经元与下一层神经元全互联，神经元之间不存在同层连接也不存在跨层连接

输入： 训练集 $D = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{m}$;
学习率 η.

过程：

在$(0,1)$范围内随机初始化网络中所有连接权和阈值
repeat
  for all $(\boldsymbol{x}_k, \boldsymbol{y}_k) \in D$ **do**
    根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\boldsymbol{y}}_k$;
    根据式(5.10) 计算输出层神经元的梯度项 g_j;
    根据式(5.15) 计算隐层神经元的梯度项 e_h;
    根据式(5.11)-(5.14) 更新连接权 w_{hj}, v_{ih} 与阈值 θ_j, γ_h
  end for
until 达到停止条件

输出： 连接权与阈值确定的多层前馈神经网络

图 5.8　误差逆传播算法

如何跳出局部极小达到全局最小:

1. 多组不同的初始参数优化，选取误差最小解
2. 模拟退火技术：每一步都以一定的概率接受比当前解更差的结果，从而有助于跳出局部极小。
3. 随机梯度下降：计算梯度的时候加入了随机因素。
4. 遗传算法:

第六章

核函数 接受两个低维空间里面的向量，能够计算出经过某种变换后的在高维空间的向量内积值。

替代损失函数数学性质较好，一般是 0/1 损失函数的上界

无论是支持向量机还是支持向量回归，学得的模型总可以表示成核函数的线性组合.

支持向量机与对率回归:

两者的优化目标相近，通常性能也相近。

对率回归的优势主要在于其输出具有自然的概率意义即在预测标记的同时也给出了概率，而支持向量机没有

对率回归可直接用于多分类任务，支持向量机则需要推广

支持向量机解具有稀疏性，对率损失是光滑的单调递减函数，对率回归的解依赖于更多训练样本其预测开销更大

第七章

贝叶斯判定准则 为最小化总体风险，只需在每个样本上选择那个能使条件风险最小的类别标记。

估计类条件概率的常用策略 先假定其具有某种确定的概率分布形式，再基于训练样本对概率分布参数估计。

极大似然估计：是试图在参数值 所有可能的取值中，找到一个使数据出现的“可能性”最大值。

EM 算法:

- 当参数　已知　－> 根据训练数据推断出最优隐变量　　的值(E 步)
- 当　已知　－> 对　做极大似然估计(M 步)

第八章

同质：只包含同种类型的个体学习器

异质：包含不同算法生成的。

串行，并行。

Booting——AdaBoot 算法:

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的受到更多的关注，然后基于调整之后的样本分布来训练下一个基学习器；如此反复进行到 T 次。

算法步骤:

输入: 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:

1: $\mathcal{D}_1(\boldsymbol{x})=1/m$.
2: **for** $t=1,2,\ldots,T$ **do**
3: $\quad h_t=\mathfrak{L}(D,\mathcal{D}_t)$;
4: $\quad \epsilon_t=P_{\boldsymbol{x}\sim\mathcal{D}_t}(h_t(\boldsymbol{x})\neq f(\boldsymbol{x}))$;
5: $\quad$ **if** $\epsilon_t>0.5$ **then break**
6: $\quad \alpha_t=\frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
7: $\quad \mathcal{D}_{t+1}(\boldsymbol{x})=\frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t}\times\begin{cases}\exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x})=f(\boldsymbol{x})\\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x})\neq f(\boldsymbol{x})\end{cases}$
$\qquad\qquad =\frac{\mathcal{D}_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x})h_t(\boldsymbol{x}))}{Z_t}$
8: **end for**

输出: $H(\boldsymbol{x})=\operatorname{sign}\left(\sum_{t=1}^{T}\alpha_t h_t(\boldsymbol{x})\right)$

对特定的数据分布进行学习:

重赋值法

重采样法

注意一直检查是否满足基本条件，若不满足则停止，可以“重启法”

Bagging:

从偏差-方差的角度 降低方差，在不剪枝的决策树、神经网络等 易受样本影响的学习器上效果更好

随机森林:

· RF 的主要优点有:

1.训练可以高度并行化，对于大数据时代的**大样本训练速度有优势**。 个人觉得这是的最主要的优点。

2.由于可以随机选择决策树节点划分特征，这样在样本特征**维度很高 的时候，仍然能高效的训练模型。**

3.在训练后，可以给出**各个特征对于输出的重要性**

4.**方差小，泛化能力强**

5.对部分特征缺失不敏感。

缺点:

.在噪音比较大的样本集上面，容易过拟合。

取值划分比较多的特征容易对 RF 决策产生更大的影响从而影响拟合结果。

结合算法:

平均（用于回归）

投票

学习法

个体学习器精确性越高、多样性越大，则集成 效果越好。称为误差-分歧分解

第九章

Kmeans 算法分析

π 主要优点:

λ 是解决聚类问题的一种经典算法，**简单、快速**。

λ **对处理大数据集，该算法是相对可伸缩和高效率的**。其时间复杂度为 O(nkt)，n 是 对象数目，k 是簇的数目，t 是迭代次数。通常 k<<

λ 当结果**簇是密集的，而簇与簇之间区别明显**时，它的效果较好。

π 主要缺点:

λ 必须事**先确定 k 值**，很多情况下并不清楚类别个数。但是聚类结果对初始值敏感， 对于不同的初始值，可能导致不能结果。

λ **初始聚类中心的选择对聚类结果有较大的影响**。初值选择不当，可能无法得到好 的聚类结果。可以多设一些不同初值，对比最后的运算结果，一直到结果趋于稳 定来解决这一问题，但比较耗时耗资源。

λ 在**簇的平均值被定义的情况下才能使用**，这对于处理符号属性的数据不适用。

λ 对于**噪声和孤立点数据是敏感**的，少量的该类数据能够对平均值产生较大的影响

密度聚类:

能通过样本分布的紧密程度来确定。

π 优点:

λ 形成的簇可以具有**任意形状和大小**;

λ 可以**自动确定形成的簇数目**;

λ 可以**分离簇和环境噪声**;

λ 可以被**空间索引结构所支持**;

λ **效率高**，即使对大数据集也是如此；
λ **一次扫描数据即可完成聚类**。
π 缺点：
λ 只能发现**密度相仿的簇**；
λ **对用户定义的参数（Eps 和 MinPts）是敏感的**，参数难以确定（特别 是对于高维数据），设置的细微不同可能导致差别很大的聚类；
λ 所使用参数 Eps 和 MinPts 是两个全局参数，**不能刻画高维数据内在的 聚类结构**，因为真实的高维数据常常具有非常倾斜的分布；
λ 计算复杂度至少为 O(n2)，若采用空间索引，**计算复杂度**为 O(nlog n)。

降维的作用
π **降低时间复杂度和空间复杂度** π **节省提取不必要特征的开销** π **去掉数据集中夹杂的噪声** π **较简单的模型在小数据集上有更强的鲁棒性** π 当数据能有**较少的特征**进行解释，可以**更好的解释数据**，使得可 以提取知识。 π 实现**数据可视化**

PCA 算法步骤 π 设有 m 条 n 维数据。 λ 1）将原始数据按列组成 n 行 m 列矩阵 X λ 2）将 X 的每一行（代表一个属性字段）进行零均值化，即减去这一行的 均值 λ 3）求出协方差矩阵 λ 4）求出协方差矩阵的特征值及对应的特征向量 λ 5）将特征向量按对应特征值大小从上到下按行排列成矩阵，取前 k 行组 成矩阵 P λ 6）Y=PX 即为降维到 k 维后的数据