机器学习往年题&习题整理

名词解释

1.机器学习

是一种让机器通过样本自动学习规则的方法。

ML目标：使学得的模型能够很好地适用于新的样本

学习：指对于某类任务T和性能度量P，一个计算机程序在T上以P衡量的性能随着经验E而自我完善，则称这个计算机程序在从经验E学习

步骤：数据处理，特征提取、转化，算法学习，测试，实用。

2.主动学习

主动学习(active learning)主要针对数据标签较少或标注代价较高的场景而设计的。主动学习主动地提出标注请求，将经过筛选的数据提交给专家进行标注，以期在较少的训练样本下获得较好的模型。

3. ID3

ID3是一种决策树算法，采用的是利用信息增益来判断以哪个特征作为分类依据。

4.神经网络

人工神经网络，简称神经网络，在计算机领域中，是一种模拟生物神经网络的结构功能和计算的模型， 目的是模拟大脑某些机理与机制，实现某个方面的功能，如图像的识别，语音的识别等。

5.多层感知器

多层感知器（Multi Layer Perceptron，即 MLP）包括至少一个隐藏层（除了一个输入层和一个输出层以外）。

6.线性可分

指一组二分类样本可以通过线性函数作为界限完整地分为两类。

7.KNN

knn算法是一种分类算法，具体为选取样本点最近的k个邻居，以多数者的类型作为样本的类型。

8.独立同分布

指随机过程中，任何时刻的取值都为随机变量，如果这些随机变量服从同一分布，并且互相独立，那么这些随机变量是*独立同分布*。

9.激活函数

是在人工神经网络的神经元上运行的函数，负责将神经元的输入映射到输出端。

10.支持向量 margin

距离超平面最近的这几个训练样本点使w x +b=1或w x +b=-1成立，它们被称为“支持向量”(support vector)，也就是就是支持平面上把两类类别划分开来的超平面的向量点。

margin（分类间隔）假设H代表分类线，H1和H2是两条平行于分类线H的直线，并且它们分别过每类中离分类线H最近的样本， H1和H2之间的距离叫做分类间隔。

11.奥卡姆剃刀原理

如果有两套理论都可以解释一件事情，用那个简单的理论。空洞无物的普遍性要领都是无用的累赘，应当被无情地“剃除”。即复杂问题简单化。

12.间隔 软间隔

边距定义为边界在到达数据点之前可以增加的宽度。

允许训练的模型中，部分样本（离群点或者噪音点）不必满足该约束，同时在最大化间隔时，不满足约束的样本应该尽可能的少。

13.强化学习

强化学习是智能体（Agent）以“试错”的方式进行学习，通过与环境进行交互获得的奖赏指导行为，目标是使智能体获得最大的奖赏

简答题

1.parzen窗简述。

Parzen窗法是指定V_N，求取包含在以待估计点x_0为中心，区域R_N内的（体积V_N）内的样本数k_N，从而得到该点概率密度的方法。区域R_N的函数就叫做窗函数。窗函数的形式有多种，主要分为4种：①超球窗②超立方体窗③正态分布窗④指数分布窗。在这里我们只介绍最常用的正态分布窗。样本点计为x,待求点中心记为x_0，令$d = x - x_0$，设球半径为h，距离尺度$u = d/h$。其窗函数$\phi(u)$如下：

$$\phi(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$

我们可以认为每个样本对于空间上的每个点的概率密度都有一定的贡献，但是随着空间点距离样本点的距离增大，这个贡献率会减小。一个样本点对其所在位置的概率密度贡献最大，随着距离样本点的距离会依次减小，**这个贡献率分布函数就是由窗函数定量给出的。**将每个样本点对所有空间点的概率密度贡献累加，就是空间上的概率密度分布函数，这便是Parzen窗的算法原理。

注意将每个样本的窗函数累加只是代表k的值，即距离空间点最近的k个元素，要想求出概率密度，还需要知道R区域的体积（样本数N是已知的）。在Parzen窗法中，体积V是正比于$\frac{1}{\sqrt{N}}$的一个函数，它需要跟距离尺度u保持一致，使估计概率密度的极限与窗函数系数h无关。因此$V = \frac{h}{\sqrt{N}}$，这样对于空间点x_0

$$k = \sum_{i=1}^{N} \phi\left(\frac{X_i - x_0}{h}\right)$$

$$\hat{p}(x_0) = \frac{k}{VN} = \frac{\sum_{i=1}^{N} \phi\left(\frac{X_i - x_0}{h}\right)}{\frac{h}{\sqrt{N}} N} = \sum_{i=1}^{N} \frac{1}{h\sqrt{2\pi N}} e^{-\frac{1}{2}\frac{(X_i - x_0)^2}{h}}$$

2.梯度下降算法与牛顿法的基本思想和区别。证明为什么负梯度是下降最快的方向。

梯度下降算法：通过搜索方向和步长来对参数进行更新。其中搜索方向是目标函数在当前位置的负梯度方向。因为这个方向是最快的下降方向。步长确定了沿着这个搜索方向下降的大小。

牛顿法：是通过求解目标函数的一阶导数为0时的参数，进而求出目标函数最小值时的参数。其迭代过程是在当前位置x0求该函数的切线，该切线和x轴的交点x1，作为新的x0,重复这个过程，直到交点和函数的零点重合。此时的参数值就是使得目标函数取得极值的参数值。

区别：**梯度下降的目的是直接求解目标函数极小值，而牛顿法则变相地通过求解目标函数一阶导为零的参数值，进而求得目标函数最小值。**牛顿法收敛速度相比梯度下降法很快，缺点就是计算海森矩阵的逆比较困难，消耗时间和计算资源。

证明：

设单位向量 $I = (cos\theta, sin\theta)$ ，其中 θ 是此向量与x轴正向夹角。单位向量 I 可以表示对任何方向导数的方向，如下图：

函数$z = f(x, y)$沿单位向量 I 方向的导数为：

$$D_I f = f_x(x,y)\cos\theta + f_y(x,y)\sin\theta$$

想要找到使函数$z = f(x, y)$下降最快的方向，我们将上述方向导数最小化，

$$\min D_l f = \min[f_x(x,y)\cos\theta + f_y(x,y)\sin\theta]$$

设向量（这个向量其实就是梯度，只是大家给它起了一个“小名”而已）

$$\nabla f = (f_x(x,y), f_y(x,y))$$

$$\begin{aligned}\min D_l f &= \min[f_x(x,y)\cos\theta + f_y(x,y)\sin\theta] \\ &= \min \nabla f \cdot I \\ &= \min|\nabla f| \cdot |I| \cos\alpha \\ &= \min|\nabla f| \cos\alpha\end{aligned}$$

其中 α 为单位向量 I 与梯度向量的夹角。单位向量的模为1。

当 $\alpha = 180$ 度时，取最小值。180度时其实就是梯度的负方向。

所以说，**梯度的负方向是函数值下降最快的方向。**

3.什么是过拟合？模型为什么会出现过拟合？如何避免过拟合？

过拟合：训练的模型过于复杂，导致在训练集上运行正确率高，但是在测试集上正确率严重下降。

为什么会出现过拟合：

训练集的数量级和模型的复杂度不匹配。训练集的数量要小于模型的复杂度；
训练集和测试集特征分布不一致；
样本里的噪音数据干扰过大，大到模型过分记住了噪音特征，反而忽略了真实的输入输出间的关系；
权值学习迭代次数足够多（overtraining），拟合了训练数据中的噪声和训练样例中没有代表性的特征。

如何避免：

1.增加训练数据。

2.数据处理-清洗数据

3.dropout，通过修改隐藏层神经元的个数，使其部分故障来防止过拟合。

4.在模型对训练数据集迭代收敛之前停止迭代来防止过拟合。

5.正则化

6.决策树剪枝

7.集成学习

4.贝叶斯的基本思想和过程

已知类条件概率密度参数表达式和先验概率

★利用贝叶斯公式转换成后验概率

★根据后验概率大小进行决策分类

过程：

> 1、估计类条件概率密度$P(x|\omega_i)$
>
> 2、估计类先验概率$P(\omega_i)$(一般从训练数据中统计)
>
> 3、决策代价λ_{ij}。（除非面向特定应用，否则一般用0/1损失，即最大后验决策）
>
> 4、计算错误率
>
> 5、判断大小

5.Adaboost、bagging的基本思想，并比较他们的异同点

Bagging：利用Bootstrap思想产生多个伪样本集，然后采取多种复杂的模型来预测数据类型，把最终得到的结果做平均或者投票得到最后结果。

AdaBoost（Adaptive Boosting），即自适应增强。主要思想是：对分错的样本提高权重，对分错的样本降低权重，用全新的加权样本去训练下一个分类器，直到达到某个预定的足够小的错误率或达到预先指定的最大迭代次数。

相同点：都是集成学习，都是采用多个模型来提高最终决策精确度

不同点：

1）样本选择上：

Bagging：训练集是在原始集中有放回选取的，从原始集中选出的各轮训练集之间是独立的。

Boosting：每一轮的训练集不变，只是训练集中每个样例在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整。

2）样例权重：

Bagging：使用均匀取样，每个样例的权重相等

Boosting：根据错误率不断调整样例的权值，错误率越大则权重越大。

3）预测函数：

Bagging：所有预测函数的权重相等。

Boosting：每个弱分类器都有相应的权重，对于分类误差小的分类器会有更大的权重。

4）并行计算：

Bagging：各个预测函数可以并行生成

Boosting：各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。

6.欠拟合概念，原因及解决方法

概念：泛化能力差，训练样本集准确率低，测试样本集准确率低

原因：1.训练样本数量少
2.模型复杂度过低
3.参数还未收敛就停止循环

解决方法：1.增加样本数量
2.增加模型参数，提高模型复杂度
3.增加循环次数
4.查看是否是学习率过高导致模型无法收敛

7.二分类：混淆矩阵、查准率、查全率、ROC曲线横纵坐标、AUC

混淆矩阵的每一列代表了预测类别，每一列的总数表示预测为该类别的数据的数目；每一行代表了数据的真实归属类别，每一行的数据总数表示该类别的数据实例的数目。

查准率（precision）：所有预测为阳性的样本中真正为阳性的比例 p=TP/（TP+FP）

查全率（recall）：所有真正为阳性的样本中预测为阳性的比例 r=TP/（TP+FN）

ROC曲线 横坐标：FPR（False Positive Rate）假阳率 FPR=FP/(TN+FP)

纵坐标：TPR真阳率=TP/(TP+FN)

AUC:AUC（Area Under Curve）被定义为ROC曲线下的面积，显然这个面积的数值不会大于1

当横坐标越小，纵坐标数值越大时，分类器效果好，因此AUC越大效果越好

8.Kmeans思想，K的初始化

思想：1.随机选取k个中心点

2.按照距离中心点最近的原则将所有样本归类到最近的中心点形成聚类。

3.计算新聚类的中心值作为新的聚类中心点。

4.重复23步直到没有点归类为新的簇

k的初始化：肘部法：肘部法所使用的聚类评价指标为：数据集中所有样本点到其簇中心的距离之和的平方。每次计算聚类完成后的距离之和平方就是SSE，但是肘部法选择的并不是**误差平方和（SSE）**最小的k，而是误差平方和突然变小时对应的值，轮廓系数法

9.有监督学习和无监督学习的机制，代表算法

有监督学习：**输入的数据有标签。**监督学习是指数据集的正确输出已知情况下的一类学习算法。因为输入和输出已知，意味着输入和输出之间有一个关系，监督学习算法就是要发现和总结这种“关系”。

代表算法：

- 线性回归
- 神经网络
- 决策树
- 支持向量机
- KNN
- 朴素贝叶斯算法

无监督学习：无监督学习是指对无标签数据的一类学习算法。因为没有标签信息，意味着需要从数据集中发现和总结模式或者结构。

代表算法：

- 主成分分析法（PCA）
- 聚类算法 k-means

10.简述线性回归，比较岭回归和lasso回归的区别

线性回归：线性回归采用一个高维的线性函数来尽可能的拟合所有的数据点

岭回归的目标函数在一般的线性回归的基础上加入了正则项，在保证最佳拟合误差的同时，使得参数尽可能的“简单”，使得模型的泛化能力强（即不过分相信从训练数据中学到的知识）。正则项一般采用一，二范数，使得模型更具有泛化性，同时可以解决线性回归中不可逆情况。

Lasso回归采用一范数来约束，使参数非零个数最少。

岭回归与Lasso回归最大的区别在于岭回归引入的是L2范数惩罚项，Lasso回归引入的是L1范数惩罚项，Lasso回归能够使得损失函数中的许多θ均变成0，这点要优于岭回归，因为岭回归是要所有的θ均存在的，这样计算量Lasso回归将远远小于岭回归。

综合题

1.从期望损失角度解释adaboost，如分布和分类器权重更新的依据。

AdaBoost的策略是：我们先得到一个学习能力较弱的$f_1(x)$，接下来要找一组与$f_1(x)$互补的$f_2(x)$，那么我们**就可以先找到一个新的训练集，让原来的$f_1(x)$在这个上面的拟合能力很烂，然后再用这个训练集来训练产生新的$f_2(x)$，这样$f_2(x)$就与$f_1(x)$互补了。**。。。真会玩。那么我们该如何找到这个可以让$f_1(x)$表现很烂的训练集呢?

然后调整权值，如此往复。

最后整合有两种方法，一种是分类相加看哪种最多就选择哪种，一种是相加前乘以一个系数来解决分类器表现不同的问题，系数为：

$$\alpha_t = \ln \sqrt{(1-\varepsilon)/\varepsilon}$$

2.（1）从VC维和结构风险角度分析为什么margin要最大化。

因为你离得越近，比如说刚好有一些点在分界线上，那么就容易产生噪声。（风险上界最小）既是在保证风险最小的子集中选择经验风险最小的函数。从决策边界到各个training example的距离越大，在分类操作的差错率就会越小

VC维：将N个点进行分类，如分成两类，那么可以有2^N种分法，即可以理解成有2^N个学习问题。若存在一个假设H，能准确无误地将2^N种问题进行分类。那么这些点的数量N，就是H的VC维。 这个定义真生硬，只能先记住。一个实例就平面上3个点的线性划分的VC维是3. 而平面上 VC维不是4，是因为不存在4个样本点，能被划分成2^4 = 16种划分法，因为对角的两对点不能被线性划分为两类。更一般地，在r 维空间中，线性决策面的VC维为r+1。

置信风险的影响因素有： 训练样本数目和分类函数的VC维。训练样本数目，即样本越多，置信风险就可以比较小；VC维越大，问题的解的种类就越多，推广能力就越差，置信风险也就越大。因此，增加样本数，降低VC维，才能降低置信风险。而一般的分类函数，需要提高VC维，即样本的特征数据量，来降低经验风险，如多项式分类函数。如此就会导致置信风险变高，结构风险也相应变高。过度学习即overfit，就是置信风险变高的缘故。

结构风险最小化SRM(structured risk minimize)就是同时考虑经验风险与结构风险。在小样本情况下，取得比较好的分类效果。 保证分类精度（经验风险）的同时，降低学习机器的 VC 维，可以使学习机器在整个样本集上的期望风险得到控制，这应该就是SRM的原则。

当训练样本给定时，分类间隔越大，则对应的分类超平面集合的 VC 维就越小。（分类间隔的要求，对VC维的影响）

根据结构风险最小化原则，前者是保证经验风险（经验风险和期望风险依赖于学习机器函数族的选择）最小，而后者使分类间隔最大，导致 VC 维最小，实际上就是使推广性的界中的置信范围最小，从而达到使真实风险最小。

训练样本在线性可分的情况下，全部样本能被正确地分类（咦这个不就是传说中的yi(*w*xi+b)）>=1的条件吗），即经验风险Remp 为 0 的前提下，通过对分类间隔最大化（咦，这个就是Φ（w） = (1/2)*w*w嘛），使分类器获得最好的推广性能。

（2）推导优化函数的对偶形式。

给个结果 $$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

（3）简述SVM线性不可分的情况下如何求解

用核函数将样本映射到高维空间，使其线性可分再进行计算。

2.加松弛变量

3.什么是训练误差，什么是泛化误差

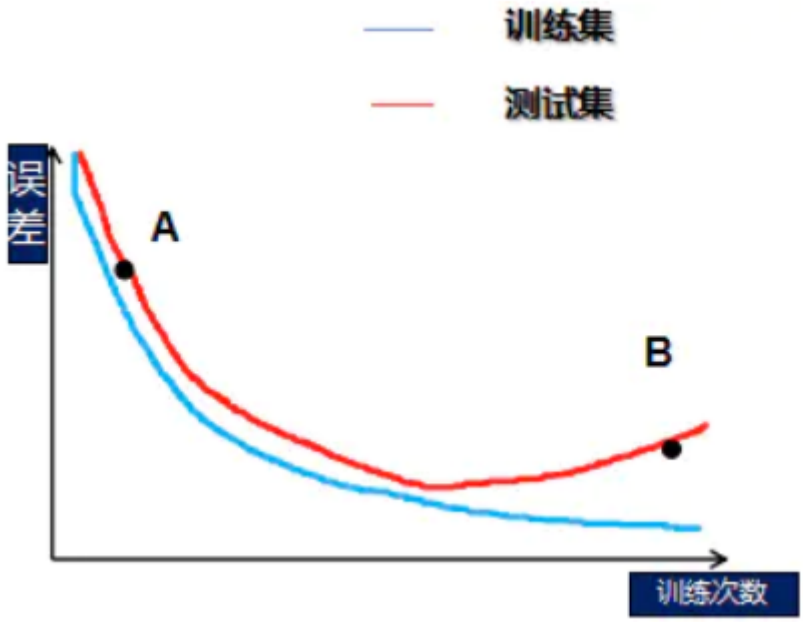
训练误差就是在训练集上期望输出的结果与实际输出的结果的差值平方和，泛化误差就是在测试集和其他数据集上期望输出的结果与实际输出的结果的差值平方和。

4.画图说明误差，指出过拟合和欠拟合区域

其实，模型在训练集上的**误差**来源主要来自于**偏差**（和1比较），在测试集上**误差**来源主要来自于**方差**（和训练集比较）。

	训练集	测试集	偏差	方差
欠拟合	80%	79%	20%	1%
过拟合	99%	80%	1%	19%

上图表示，如果一个模型在训练集上正确率为 80%，测试集上正确率为 79% ，则模型欠拟合，其中 20% 的误差来自于偏差，1% 的误差来自于方差。如果一个模型在训练集上正确率为 99%，测试集上正确率为 80% ，则模型过拟合，其中 1% 的误差来自于偏差，19% 的误差来自于方差。

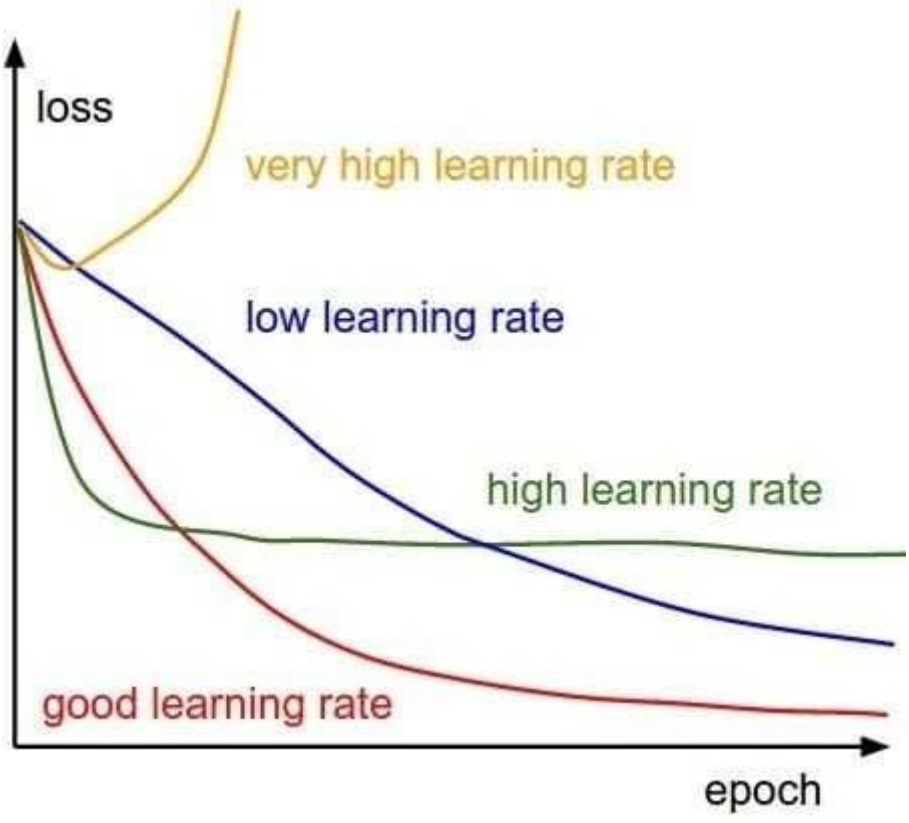


A区域为欠拟合，B区域为过拟合，中间为合适。

5.怎么选择学习率，过大过小的影响?

首先我们设置一个非常小的初始学习率，比如1e-5，然后在每个batch之后都更新网络，同时增加学习率，统计每个batch计算出的loss。最后我们可以描绘出学习的变化曲线和loss的变化曲线，从中就能够发现最好的学习率。

如果学习率太小，会导致网络loss下降非常慢，如果学习率太大，那么参数更新的幅度就非常大，就会导致网络收敛到局部最优点，或者loss直接开始增加



6.两个一模一样的碗，一号碗有30颗水果糖和10颗巧克力糖，二号碗有水果糖和巧克力糖各20颗。现在随机选择一个碗，从中摸出一颗糖，发现是水果糖。请问这颗水果糖来自一号碗的概率有多大?

p=0.6

较易。