一、 统计学习概念方法

监督学习、非监督学习、半监督学习、强化学习

监督学习的基本概念

监督学习如分类，回归，标注，即学习一个模型，使得模型对于给定的输入，能对其输出做出一个较好的预测。

训练数据——模型（属于某个函数集合，即假设空间）——评价准则——选出最优模型（选取方式即算法）——进行预测分析等

输入空间——特征空间（特征向量存在的空间，模型是定义在特征空间上的）——输出空间——假设空间

模型：即所要学习的条件概率分布（决策函数），模型的假设空间包含了所有可能的条件概率分布，例如所有线性函数构成的函数集合。

策略：有了模型的假设空间，就需要按照一个准则去选择最优的模型，即策略。常见的是通过引入损失函数/ /代价函数（一次预测的好坏），风险函数（平均意义下的模型预测好坏）的概念来度量模型的好坏。

算法：具体采用什么样的计算方法去求解最优的模型。例如若有显示的解析解，就比较方便直接求解，若不存在解析解，往往就需要数值方法求解。算法的目的就是使得求解过长高效准确。

Cost Function/Loss function——选择模型

$$L(Y, f(X))$$

（1）0-1 损失函数

（2）平方损失函数

（3）绝对损失函数

（4）对数损失函数

学习的目标就是选择期望风险最小的模型，但是由于联合分布位置，所以可以通过样本数据计算经验风险来作为替代，根据大数定律，当样本数量趋于无穷时，经验风险趋于期望风险。

经验风险：

$$\frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i))$$

当样本容量足够大时，经验风险最小化能有较好的效果，但是样本容量较小的时候，可能就会出现过拟合的现象，从而就会通过结构风险最小化的方式。

结构风险：

$$\frac{1}{N}\sum_{i=1}^{N} L\big(y_i, f(x_i)\big) + \lambda C(f)$$

$C(f)$*是模型的复杂度*

模型选择——最大熵原理&最大熵模型

学习概率模型时，在所有可能（满足约束条件）的概率模型中认为熵最大的模型时最好的模型。

模型选择——极大似然估计

目标函数最优化求解方法

迭代尺度法、梯度下降法、牛顿法、拟牛顿法

模型评估

可以采用训练时所用的损失函数来作为评估标准。

训练误差 测试误差（评判模型预测能力的指标）

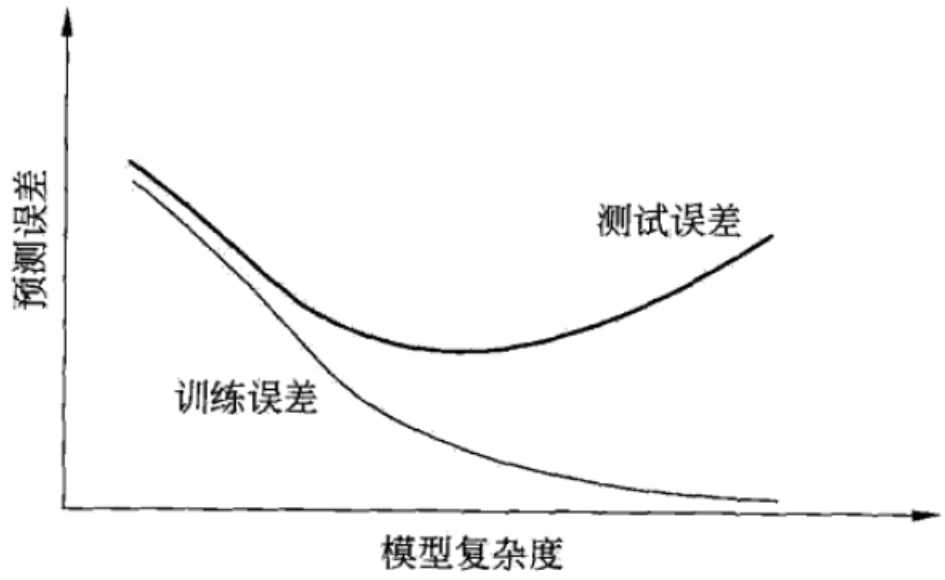
过拟合问题

过拟合问题：模型过于复杂，使得训练数据表现很好，但在测试数据上可能就表现很差。

解决方式：

（1） 选择合适正确的特征：手工选择；PCA 等模型。

（2） 正则化，即保留所有的特征，但是减少参数的大小。



正则化

正则化是对模型复杂度加以惩罚的一种模型选择方式，例如对多项式拟合目标函数中的参数（or 选择性地）添加惩罚项，加以惩罚，一定程度上减少高次项参数的值。

一般形式：

$$\frac{1}{N}\sum_{i=1}^{N} L\big(y_i, f(x_i)\big) + \lambda C(f)$$

λ被称作正则化参数

例如，若取正则化项取 L2 范数：

$$J(\theta) = \frac{1}{2n} * \sum_{i=1}^{n} (f_\theta(x_i) - y_i)^2 + \lambda \sum_{j=1}^{p} {\theta_j}^2$$

λ的选取方式：

通常是在 0-10 之间呈现 2 倍关系的值，可以取不同的值，分别使用训练集训练处若干个不同程度的正则化模型，利用交叉验证集分别计算交叉验证误差，选取交叉验证误差最小的一个模型。

交叉验证

在样本数据充足时，可以考虑数据集切分方式来进行模型选择，将数据切分成训练集、验证集、测试集，利用验证集来进行模型选择。

但是当样本数据不够充足时，即可以考虑利用交叉验证的方法，其主要思想是重复利用数据，反复进行训练、测试以及模型的选择。

（1）简单交叉验证

随机地将已有数据分为两部分：训练集，测试集。用训练集在不同条件下（如不同的参数个数）训练模型，得到不同的模型，再在测试集上评价测试误差，选出测试误差最小的模型。

（2）S-fold 交叉验证

随机将数据切分为 S 个互不相交的大小相同的子集，利用 S-1 个子集的数据训练模型，剩下的子集测试模型，重复 S 次这个过程，选出 S 次测评中平均测试误差最小的模型。

（3）留一交叉验证

即 S-fold 的特殊情形，取 S=数据及容量 N

泛化能力

指由某种方法学习到的模型对位置数据的预测能力，现实中可以利用测试误差来评价学习方法的泛化能力。

特征/样本选择

获取更多示例/减少特征数量/获取更多特征/正则化程度

通过学习曲线判断是否存在高偏差/高方差的问题

误差分析

二、线性回归

线性回归模型表示：

$$Y = \theta_0 + \sum \theta_i x_i = \theta^T X$$

Cost function：

$$J(\theta) = J(\theta_0, \theta_1, \theta_2 \dots)$$

Batch Gradient Descent：

$$\theta_j = \theta_j - \alpha * \frac{\partial J(\theta)}{\partial \theta_j}$$

α是学习率，决定了让 cost function 下降的步幅，过小时会是的搜索的速度太慢，过大的话容易越过最优解，甚至无法收敛。当接近局部最优解时，导数变小，使得梯度下降的幅度自动变小。通过绘制 Cost function 与迭代次数的图来直观感受何时收敛(或者设置某个阈值)。

Normal Equation：

求解方程使得

$$\frac{\partial J(\theta)}{\partial \theta_j}$$

$$\theta = (X^T X)^{-1} X^T y$$

但是当(1)特征之间不独立 或 （2）特征数量>样本数量时，逆矩阵可能会不存在，则无法直接通过正规方程得到参数。

梯度下降法与正规方程法的比较：

梯度下降法需要选择学习率，且需要多次迭代，而正规方程法直接通过一次计算得到；

当特征数量p特别大（如>10000）时，正规方程法的计算会比较复杂，因为逆矩阵的时间复杂度是$O(p^3)$;

梯度下降法适用于多种类型模型的求解，而正规方程法不适合 logistic 回归等。

特征缩放（feature scaling)：

对于多维的特征，应尽量使得不同的特征具有相似的尺度，从而使得收敛速度更快。对于不同尺度的特征，就需要通过特征缩放使得所有特征的尺度尽量保持一致（如-1~1 之间)。交简单的方法是可以用:

$$x_n = \frac{x_n - \mu_n}{s_n} \quad \mu_n是样本均值 \ \ s_n是样本方差$$

多项式回归前特征缩放尤为必要。

三、Logistic 回归

当分类问题的预测变量是离散值时，可以考虑采用 logistic regression

线性回归的输出变量（预测值）范围会很大，而我们希望一个假设函数其预测值在 0,1 之间（当预测值>0.5 时可以归为 1，<0.5 时归为 0）。
Logistic 回归模型假设：

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

当$h_\theta(x) > 0.5 \Leftrightarrow \theta^T x > 0$时，预测$y$ =1
当$h_\theta(x) < 0.5 \Leftrightarrow \theta^T x < 0$时，预测$y$ =0

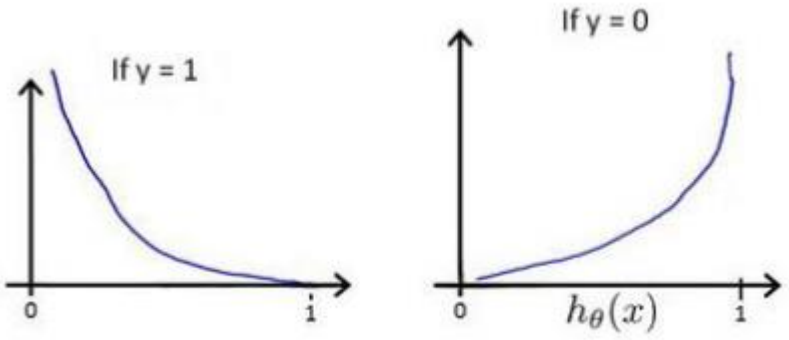
Cost function：
如果采用线性回归的的误差平方和作为 Cost function，其实非凸的，则 Cost function 存在很多局部极小值点，会影响梯度下降法。

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} cost(h_\theta(x_i), y_i)$$

$$cost(h_\theta(x_i), y_i) = \begin{cases} -\log\big(h_\theta(x)\big) & y_i = 1 \\ -\log\big(1-h_\theta(x)\big) & y_i = 0 \end{cases}$$

即：$cost(h_\theta(x), y) = -y * \log\big(h_\theta(x)\big) - (1-y) * \log\big(1-h_\theta(x)\big)$

实际上，利用极大似然估计法估计模型参数所得到的对数似然函数与上面表达式一致·



梯度下降法迭代求解：

$$\theta_j = \theta_j - \alpha * \sum_{i=1}^{n} \big(h_\theta(x^i) - y^i\big) * x_j^i$$

其他求代价函数最小的算法：共轭梯度；局部优化法（BFGS）；有限内存局部优化法（LBFGS）。
大致思路：给出计算导数项和代价函数的方法，算法自动尝试不同的学习速率α（内部的线性搜索算法），并自动选择一个好的学习速率α，每次迭代甚至可以选择不同的学习速率

一对多的分类问题：
训练时：依次遍历每个类 i，将 i 标记为正向类，剩余的类标记为负向类，进而得到$h_\theta^i = p(y = i|x, \theta)$
预测时：对于一个输入变量，将每个分类机都运行一遍，选择最高可能性输出变量所对应的分类。

四、感知机

二分类的线性分类模型，根据输入的特征向量，输出为取值+1/-1 的值，几何上感知机就是将输入空间划分为政府两类的分离超平面，感知机的学习过程就是求出将训练数据进行划分的超平面。
模型：

$$f(x) = sign(\omega x + b)$$

损失函数：

$$J(\omega, b) = -\frac{1}{||\omega||} \sum_{x_i属于误分类集合} y_i(\omega x_i + b)$$

使得损失函数最小的求解算法利用梯度下降法。

$$\frac{\partial J}{\partial w} = -\sum_{误分类} y_i x_i$$

$$\frac{\partial J}{\partial b} = -\sum_{误分类} y_i$$

极小化过程是随机挑选一个误分类点 (x_i, y) 使得梯度下降，即：

$$\omega = \omega + \lambda y_i x_i$$
$$b = b + \lambda y_i$$

对偶形式

五、K 近邻法

分类时，对于新的实例，根据其 k 个最近的训练实例的类别来预测（多数表决）新实例的类别，可以取多累，几何上就是利用训练数据集对特征向量空间进行划分。
K 值的选择：K 值过小时，模型复杂，容易过拟合；K 值过大时，模型会过于简单。
Kd 树：
K 近邻法没有显示的学习过程，主要考虑的问题是如何快速地进行 K 近邻搜索（尤其是样本数量大、特征维数大）

六、朴素贝叶斯

后验概率

$$P(Y=c_k|X=x)=\frac{P(X=x|Y=c_k)P(Y=c_k)}{\sum P(X=x|Y=c_i)P(Y=c_i)}$$

分类器后验概率最大化，表达式为：

$$y=argmaxP(X=x|Y=c_k)P(Y=c_k)=argmaxP(Y=c_k)\prod_j P(X^j=x^j|Y=c_k)$$

利用极大似然估计估计先验概率：

$$P(Y=c_k)=\frac{\sum_{i=1}^{n}I(y_i=c_k)}{n},\ k=1,2,\dots,K$$

$$P\left(X^j=x^j\middle|Y=c_k\right)=\frac{\sum_{i=1}^{n}I(X^j=x^j,y_i=c_k)}{\sum_{i=1}^{n}I(y_i=c_k)}$$

但是利用极大似然估计估计先验概率可能会出现所要估计的概率值为 0 的情况，进而影响后验概率的计算结果。

$$P\left(X^j=x^j\middle|Y=c_k\right)=\frac{\sum_{i=1}^{n}I\left(X^j=x^j,y_i=c_k\right)+\lambda}{\sum_{i=1}^{n}I(y_i=c_k)+\lambda S_j},S_j\text{是第}j\text{个特征的可能取值个数}$$

$$P(Y=c_k)=\frac{\sum_{i=1}^{n}I(y_i=c_k)+\lambda}{n+K\lambda}$$

$\lambda=1$时即拉普拉斯平滑

七、决策树

决策树学习通常包含 3 个步骤：特征选择，决策树生成，决策树的修剪。

特征选择：

信息增益/信息增益比。计算训练数据集每个特征的信息增益，选择信息增益最大的特征。

决策树生成算法（ID3 C4.5）

ID3 算法：在决策树每个节点上应用信息增益准则选择特征，递归地选择特征构建决策树，若信息增益小于给定的阈值，则设置为单节点。此算法只有树的生成，容易产生过拟合。

C4.5 算法：利用信息增益比来选择特征。

剪枝：

对于一个给定的决策树，叶节点个数为|T|，t 是树的叶节点，该叶节点有N_t个样本点，其中 k 类的有N_{tk}个，则决策树的损失函数定义为：

$$J=C(T)+\alpha|T|$$

$$C(T)=\sum_t N_tH_t(T)=-\sum_t N_t\sum_k\frac{N_{tk}}{N_t}\log\left(\frac{N_{tk}}{N_t}\right)=-\sum_t\sum_k N_{tk}\log\left(\frac{N_{tk}}{N_t}\right)$$

$C(T)$反映了模型对训练数据的拟合程度，$|T|$反映了模型的复杂程度

剪枝即是在给定α的情况下，选择损失函数最小的模型。

具体算法：

（1） 生成算法产生树，设置好α

（2） 递归地向上回缩，如果损失函数值变小，则进行剪枝。

（3） 迭代，直至不能继续

CART 算法：给定输入随机变量生成输出变量的条件概率分布。

回归树利用平方误差最小化准则，分类数利用基尼指数最小化准则。

回归树：每个类别对应输出值为该类的样本均值；对每个变量可以找到最优切分点；遍历所有变量，找到最优的切分变量。（最小二乘回归树）

分类树：基尼指数。对每个特征，依次选取其可能的取值，计算分割后的基尼指数。遍历所有的特征和可能的取值，选择基尼指数最小的特征和取值作为拆分点作为最优特征和最优拆分点，从现节点生成两个子节点。递归调用。

CART 的剪枝：对不同的α产生最优子树序列，利用交叉验证集选择最优的。

八、支持向量机

SVM 是一种二分类模型，对于一个超平面，点距离超平面的距离越远，则分配的确定性就越高。SVm 的想法就是求解能够正确划分训练数据集并且集合间隔最大的分离超平面（即超平面不仅能够区分训练数据集，而且要有充分大的确信度来进行区分）。

线性可分支持向量机——线性支持向量机——非线性支持向量机

核方法

$$\max \quad \gamma$$

$$s.t. \quad y_i(\frac{\omega}{||\omega||}x_i + \frac{b}{||\omega||}) \geqq \gamma$$

进而改写为：

$$\max \quad \frac{\gamma}{||\omega||}$$

$$s.t. \quad y_i(\omega x_i + b) \geqq \gamma$$

可以看出，上式中γ的取值对结果没有影响，可以直接取值为 1，问题转换为

$$\min \ \frac{1}{2}||\omega||^2$$

$$s.t. \quad y_i(\omega x_i + b) \geqq 1$$

线性可分时，满足上面凸优化问题的解是存在且唯一的。决定超平面的只有支持向量，而在边界外移动其他点不会影响模型的求解。

通过拉格朗日乘数子将问题转换为对偶问题后进行求解。

线性不可分（软间隔最大化）：对于每个样本点引入一个松弛变量

$$\min \ \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.t. \quad y_i(\omega x_i + b) \geqq 1 - \xi_i$$

非线性问题 kernel trick
利用一个变换将原空间的数据映射到新空间；在新空间里用线性分类学习方法学习分类模型。函数映射是隐式的，而核函数是显示的。
求解算法：SMO 算法

九、提升方法

Adaboost 算法
主要思想：从弱学习器出发，反复学习（每一轮改变训练样本的权重），得到一些列弱分类器，然后组合这些分类器。Adaboost 的做法是前一轮弱分类器生成后，提高前一轮中分类错误的样本权重，降低准确分类的样本的权值，然后对弱分类器以多数表决的方法组合。

（1） 初始化权值，训练分类器
（2） 计算误差，更新权值，训练分类器
（3） 迭代多次后线性组合分类器

十、EM 算法

主要解决含有隐变量的问题，通过迭代计算
可以用来解决混合高斯模型的参数求解

十一、 隐马尔可夫模型

十二、 条件随机场

十三、 神经网络

原始特征只是输入层，输出层利用的特征是上一层的特征。
实现逻辑运算
计算预测结果：正向传播算法
计算偏导数，每层误差：反向传播算法

步骤：
（1） 选择网络结构：层数，每层的单元个数
（2） 参数初始随机化
（3） 正向传播算法计算 f(x)
（4） 计算代价函数
（5） 利用反向传播方法计算偏导数
（6） 利用数值检验方法检验偏导数
（7） 优化算法来最小化代价函数

十四、 启发式算法

（1） 遗传算法
（2） 蚁群算法
（3） 粒子群算法
（4） 模拟退火算法

十五、 非监督学习——聚类分析

K-均值算法
初始化选择 K 个随机地点，作为聚类中心
对于数据集中的每一个数据，将其归类到距离最近的一个聚类中心中
将每个聚类中心中的点作为一类，计算每一组的均值作为这一组的新的聚类中心。
重复迭代直到中心点不再变化。
代价函数:
为了避免局部最小值的情况（依赖于初始化的选取），可以运行多次 K-均值算法，每次都随机初始化，比较多次的结果，选择代价函数最小的结果。

聚类数 K 的选择:
肘部法则

降维算法
PCA 主成分分析

异常检测