**Machine Learning**

**Summary**

机器学习按算法类型区分：

（1）监督式学习supervised

算法输出期望是人为可以评判的(提供训练过程的)，模型可以根据人为评价进行调整；

比如： 回归模型，决策树，随机森林，knn，罗辑回归等。

（2）无监督学习un-supervised

无法预测或者评估结果；可以用再密度估计场景，异常判断场景中。

比如：聚类算法，关联规则等。

基于（1）（2）混合使用的，叫做半监督，常见于训练数据标记成本高的场景，如图像识别。

（3）强化学习reinforcement

算法训练程序来做决定，或者说算法通过反馈（惩罚或者奖励）来改进模型。比如线上广告系统，棋类游戏。通常也是online学习。

比如：马尔可夫决策过程。

机器学习按输出类型区分：

1. 分类classification

二分类(binary classification)，多分类

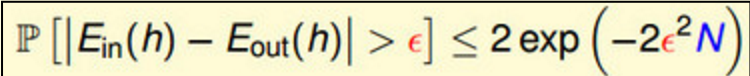
1. 回归分析

输出结果是连续数值，有界/无界

机器学习的资料类型区分：

1. online 线上学习，渐进学习
2. batch feeding 一次性资料
3. active 主动式学习，asking questions – deep learning

根据Hoeffdin定理可以证明，符合二项分布的事件是可以 被机器学习的（抽样越来越大的时候，预测结果会更接近真实结果）。公式如下：



常见的机器学习算法：

(1) 线性回归 linear regression

(2) 罗辑回归 logistic regression

(3) 决策树 decision tree

(4) 支持向量机 svm

(5) 朴素贝叶斯 naive bayes

(6) k邻近算法 knn

(7) k均值算法 k-means

(8) 随机森林 random forest

(9) 降低纬度 dimensionality reduction algorithms

(10) gradient boost & adaboost 算法

(11) gbdt

(12) 神经网络

mac os下安装python环境：

brew install gcc

brew install python

sudo easy\_install pip

pip install numpy

pip install scipy

pip install matplotlib

pip install scikit-learn --- 安装sclera

pip install pandas

材料：

【链接】scikit-learnTutorials

<http://scikit-learn.org/stable/tutorial/index.html>

【链接】Anintroductiontomachinelearningwithscikit-learn

<http://scikit-learn.org/stable/tutorial/basic/tutorial.html#loading-an-example-dataset>

数据集：

<http://archive.ics.uci.edu/ml/datasets.html>

其他：

Python中好用的的 中文切词 库有（ jieba —中文断词工具、 SnowNLP —中文文本处理库、 loso —另一个中文断词库）

声明：

总结和下述daily study参照：大数据分析，coursera机器学习基石台大课。

**Dailys:**

[1] 线性回归 －－ 隶属于广义线性模型

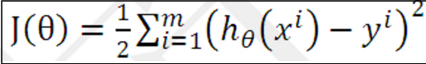
属于监督学习方法。

是对一个或者多个自变量和因变量之间的回归分析。－－－利用线性回归函数，这种函数是一个或者多个回归系数大模型参数的线性组合。

模型为线性方程：



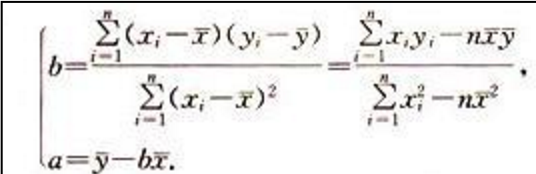
定义某个损失函数（lost function, cost function）：残差和最小，残差绝对值最小，残差平方和最小



求解参数：

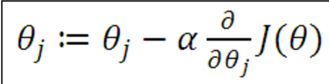
1. 最小二乘法 – 残差平方和最小

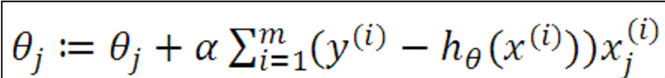
按残差和式子列出后，利用偏导=0求极值方法，可以为一元的线性回归确定参数：



（2）利用梯度下降法 求出参数。

批量梯度下降法：

 其中α是学习效率，可以根据收敛速度来判断是否要减小还是扩大。

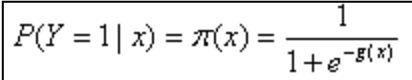


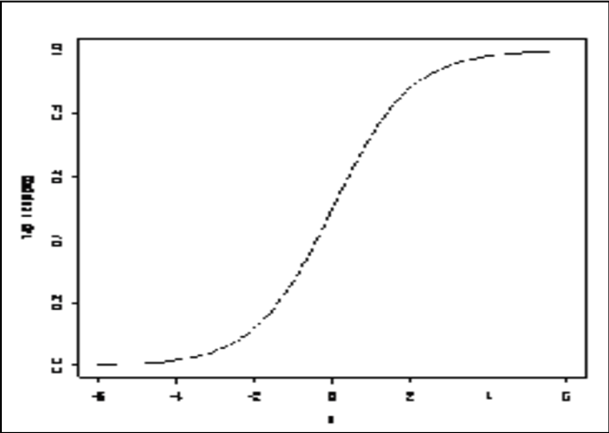
收敛快慢：lost function下降速度来判断。

[2] 罗辑回归 －－ 隶属于广义线性模型

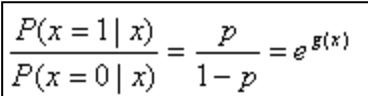
因变量是离散的分布 （如二项分布，泊松分布）。最常用来做二分类问题－－在多个自变量条件下属于因变量的概率多大。逻辑回归的特点是，因变量是概率，需要在[0,1]之间。

逻辑回归模型 – 即sigmoid函数：

， 



odds（事件发生概率/事件不发生概率）：



取对数后即为g(x)。

然后最终求解参数，为令似然函数最大—利用牛顿-拉菲森迭代法来求解。

似然函数解释：

ln 预测概率按真实情况出现的 概率

过拟合问题：过分拟合啦训练数据，是的模型复杂度提高，泛化能力差。

过拟合一般源自过多的特征

解决方法：（1）减少特征 （2）正则化

[3] 决策树

属于监督学习方法，常用于分类。

原理：根据增益信息最大选取特征作为属性节点，按照属性指分裂出子树。而后重复裂变过程。

主要特征选取方法：ID3和C4.5

ID3:

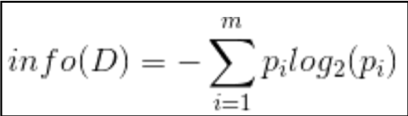
决策树的核心主要是需要在当前集合下找出一个确定性/区分度最强的属性特征。

熵 – 代表一个集合上的分布（label- 概率）的确定性/纯度 （熵越大，确定性越小，纯度越低）

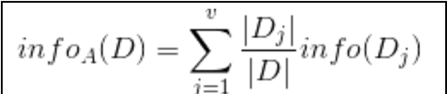
信息增益 = 原集合熵 – 分裂后集合熵

因此这里，我们需要计算每个特征对应的增益率。

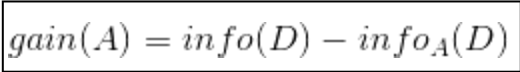
原集合熵（label - 概率）（不管选择哪个特征，该值都一样）：



按特征分裂数据集合后的熵（label – 概率）：

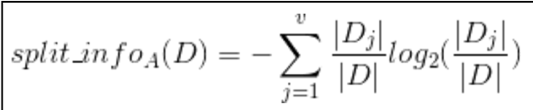
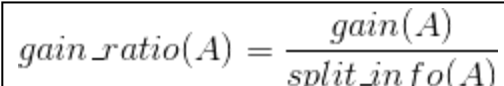


信息增益：



C4.5:

对于分布很散（但基本没有区分意义）的特征增益很大。因此我们需要对这种类型的节点进行处理。C4.5是对ID3 的改进 – 转化为信息增益率，即需要除以特征本身的熵值。

[4] 支持向量机 SVM

属于监督学习，用于分类。

模型为特征空间上的间隔最大的线性分类器，学习策略为令间隔最大化，最终转化为图二次规划问题求解。

[5]朴素贝叶斯

属于监督学习方法。

原理：利用贝叶斯公式，在先验概率下求后验概率。具体为，根据P(A/B) = P(B/A)\*P(A)/P(B)，求在B向量下，属于A类别的概率。其中，前提假设P(B1/A), P(B2/A),...,P(Bn/A)互相独立。

步骤：

（1）数据处理： 划分训练集和测试集－－留存交叉验证

（2）特征提取： 组合，平滑处理

（3）模型训练：统计各个先验概率 （稀疏数据处理方法：拉普拉斯平滑，或者利用聚类找出相关词对平均概率）

（4）模型预测：等式两边均转化为log求和来解决数据下溢或者舍入产生的误差

（5）评估：accuracy, precision, recall

|  |  |  |
| --- | --- | --- |
|  | 实际正样本 | 实际负样本 |
| 输出正样本 | A | B |
| 输出负样本 | C | D |

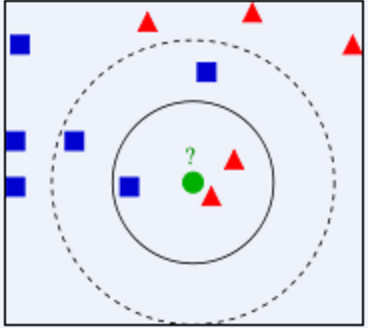
precision -- A/(A+B)

recall -- A/(A+C)

accuracy -- (A+D)/(A+B+C+D)

[6] knn

属于监督学习方法。用于分类或者回归。

简单说来就是从训练样本中找出K个与其最相近的样本，然后看这K个样本中哪个类别的样本多，则待判定的值（或说抽样）就属于这个类别。如图所示：

步骤：

（1）计算已知类别数据集中每个点与当前点的距离；

（2）选取与当前点距离最小的K个点；

（3）统计前K个点中每个类别的样本出现的频率；

（4）返回前K个点出现频率最高的类别作为当前点的预测分类。

距离：欧式距离(纬度差的平方和开根号)，曼哈顿距离（纬度差之和），闵氏距离（？），汉明距离（？离散特征）。

需要注意：

1. knn计算成本高 --- O(n\*m)
2. 特征标准化数量级，否则大数量级特征计算偏差 --- 特征值域范围压缩
3. 与处理数据，去除异常值，噪音

[7] k-means

属于无监督学习方法。

随机抽取k个点作为k个类别。每加入一个点，判断其距离谁最近，对该类别重新计算质心。依次类推。如图所示：

kmeans的计算方法如下：

1 随机选取k个中心点

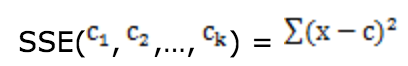
2 遍历所有数据，将每个数据划分到最近的中心点中

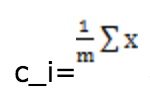
3 计算每个聚类的平均值，并作为新的中心点

4 重复2-3，直到这k个中线点不再变化（收敛了），或执行了足够多的迭代

时间和空间复杂度可以简化为O(n)，即线性的。

算法收敛

从kmeans的算法可以发现，SSE其实是一个严格的坐标下降（Coordinate Decendet）过程。设目标函数SSE如下：

采用欧式距离作为变量之间的聚类函数。每次朝一个变量的方向找到最优解，也就是求偏导数（一元导数，多元偏导数，这里对每个簇心求偏导数），然后令偏导数等于0－－－即可令整体值最小。可得

, 其中m是c\_i所在的簇的元素的个数。（对ci求偏导）（附：梯度－导函数在某一点的距离，代表这一点上升或者下降最快的方向。在一维中指斜率）。也就是当前聚类的均值就是当前方向的最优解（最小值），这与kmeans的每一次迭代过程一样。所以，这样保证SSE每一次迭代时，都会减小，最终使SSE收敛。

由于SSE是一个非凸函数（non-convex function－有鞍点的）（凸－任意两点不切边界），所以SSE不能保证找到全局最优解，只能确保局部最优解。但是可以重复执行几次kmeans，选取SSE最小的一次作为最终的聚类结果。

K值选取: 在实际应用中，由于Kmean一般作为数据预处理，或者用于辅助分类贴标签。所以k一般不会设置很大。可以通过枚举，令k从2到一个固定值如10，在每个k值上重复运行数次kmeans(避免局部最优解)，并计算当前k的平均轮廓系数，最后选取轮廓系数最大的值对应的k作为最终的集群数目。