机器学习笔记

**一 Supervised learning**

区别于监督学习与非监督学习最直接的方式是训练数据是否带有类标签（即，在分类中y表示类标签，回归问题中y表示连续值，分类问题y表示类别标签），下面是学习过程的一般形式：

训练数据

学习算法

h

预测 y

新数据

**一线性回归**

**注：上标表示第i样本，下标表示第j个特征或参数的第j个分量。**

同样，还是借助NG教学中的例子来引出线性回归。假设我们有以下数据，一套房子的价格可能与很多方面有关，比如面积，配置，地理位置等，这里我们选取其中之一面积作为特征来进行线性回归来预测房子的价格与面积的关系。

|  |  |
| --- | --- |
| 面积 | 价格（万元） |
| 120 | 160 |
| 200 | 300 |
| 160 | 200 |
| … | … |

假设估计房子的价格特征包括城市，地段，面积，室内等。那么采集的每个数据有城市，地段，面积，室内这些属性和价格，，其中j表示第j个数据。通过对已有的数据进行学习，采用线性回归来学习一个函数实现价格预测。首先假设一个线性的预测函数，其中x表示数据的属性特征，表示预测价格，，其中为参数。



值得注意的是对于上面的线性函数的构造，我们可以看出模型在以x0，x1..的坐标上是线性的，但是并不表示线性的模型就一定只能用于线性分布问题上。假如我们只有一个特征x，而实际上回归值是等，我们同样可以采用线性模型，因为我们完全可以把输入空间映射到高维的空间，其实这也是核方法以及PCA的一种思想，凡是对输入空间进行线性，非线性的变换，都是把输入空间映射到特征空间的思想，所以只需要把非线性问题转化为线性问题即可。

通过定义一个损失函数，对于这个损失函数的直观理解就是小学开始学的最小平方逼近，其实对于一个连续变量而言，这样一种损失定义也是合理的。至于为何这样定义，后面给出其概率上的解释。



其中表示第i个样本，我们的目标就是找到最优值使得所有样本的预测值与实际值的差值平方和最小，通过这样一个目标来确定优化的参数。

对于上式的求解，求解方法有很多，通用的最优化算法都可以，考虑到其无约束二次规划问题的特性，可以对其直接求导求出极值点，且极值点为全局最小值。

采用矩阵和向量的形式表示样本。同样上标表示第i个样本，每一行表示属性特征。



其中m表示样本数，是一个样本数据。

则有：



通过矩阵和向量的表示，结合最小二乘法即极小值在梯度为零处取得（凸规划），所以有：



上式是线性回归的一个显示解。

但是在实际问题中，我们往往会遇到这样一种情况，即高维数据的特性，而当我们样本数又不多情况，即，这个时候矩阵不是满秩，不可逆。此时最小化目标函数的（w，b）的解不唯一，且非常多，出于这样一种情况，我们可以考虑奥卡姆剃刀准则来简化模型复杂度，使其不必要的特征对应的w为0。所以引入正则项使得模型中w非0个数最少，同时又解决了的不可逆的问题，至于如何实现降低模型复杂度不必多说，岭回归如何解决不可逆问题可以看看岭回归的目标函数。

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，正则项采用二范数，使得模型更具有泛化性，同时解决一般的线性回归中不可逆问题。



对目标函数求导有：

三、梯度下降法与牛顿法

对于上式，由于是一个无约束的二次规划问题有显示解，对于一般的线性回归问题，不能求解显示解的情况下，我们可以采用迭代求解算法来求解。

最小二乘法与梯度下降法比较：

最小二乘法是通过线性函数或者非线性函数进行平方逼近，定义一个逼近函数，使其最小化，最小值必然在极小值处取到（凸规划），即导数为零。一般情况下线性的最小二乘法有解析解，即上式。对于无解析解问题求解其最小值同样也可以利用最小值点在导数为零处取得，前提是凸规划，其求解过程可以采用迭代算法求解，如梯度下降，牛顿法。而梯度下降法是求解参数一种迭代算法。所以说最小二乘法更像是一种拟合思想，而梯度下降法则是一种求解算法。

**三 最小均方法（LMS）**

最小均方法就是定义损失函数为预测值与实际值差的平方均值的形式，然后通过梯度下降法来求解参数，梯度下降法思想：负梯度方向作为搜素方向，一维搜索确定步长（实际中很少用一维精确搜索，一般采用固定选取一个很小的步长），使得每一次迭代都是下降最快的，下降由负梯度保证，最快由步长保证，最优解不能保证是全局最优解。



表示的第i个分量，为每次迭代的步长，这个可以由一维精确搜索确定，也可以手动设置，但是后者不保证最速下降，同时可能会错过局部极小点，徘徊在局部极小点。

当数据量大，每次可以用少量的样本来确定搜索方向。增量梯度和随机梯度法在实际运用中更有效， 其思想是每一次随机的采取少量的数据来确定搜索方向，但最终保证每个数据都有用。

牛顿法思想：函数的极值点必然在稳定点（导数为0点取到）而且当目标函数是一个凸函数的情况，则极值点必是全局极小点。这样求解最优解就是解方程。

由f（x）在处二阶Taylor展开有：



即：

所以对于上述问题的求解形式如下：



注意：牛顿法的收敛速度很快，但是局限性很多。

**三 局部加权线性回归**

考虑到一般的线性回归可能出现欠拟合和过拟合，局部加权线性回归通过对局部数据赋予不同的权重来更好的拟合数据（有分段拟合的效果），但局部加权线性回归并没有解决过拟合和欠拟合的问题，一般过拟合与欠拟合问题可以通过特征选择来缓和。

局部加权线性回归，即对样本的价值赋予不一样的权重，其表达形式如下：



其中权重函数一般选择下面的权重函数，权重函数选择并非因为其类似于高斯函数，而是根据数据分布的特性，但权重函数的选取并不一定依赖于数据特性。



有了上面的损失函数，我们的目标同样是求解使得损失函数最小化，可以看出损失函数的形式很像最佳平方逼近，可以采用最小二乘法来求线性最小二乘的解析解（显式解）。下面先介绍一种迭代法来求解。其求解思想与最小二乘不同。

**五概率解释**

**损失函数与最小二乘法采用最小化平方和来确定的概率解释。**设预测值与真实值的误差为，那么预测结果与之间有如下关系：



假设独立同分布于均值为0，方差为的高斯分布，这样假设是合理的，假设一个事件与很多独立随机变量有关，根据中心极限定理，该事件服从正态分布。那么就有：



即表示满足以均值为，方差为的高斯分布。

即：



由最大似然估计有：





最大化似然函数等价于最小化



**六分类与逻辑回归**

对于一般的二分类问题：

二分类图

分类问题同样可以看作是一种特殊的回归函数，对于多分类可以定义一个多值函数，一般情况下定义二值函数：



定义上述函数，那么分类问题可以借助回归思想来做，但是由于上述函数看似简单，但处理起来却不易，一般情况下，我们可以采用其一个凸近似来做

定义一个逻辑函数（sigmoid函数）



从函数图像我们能看出，该函数有很好的特性，适合二分类问题。

同样， 我们可以类似于线性回归一样，定义一个损失函数，然后使其最小化来求解，对于为何采用平方和最小方式来定义损失函数，我们给出了其概率解释，其假设是误差服从高斯分布，这里我们根据概率定义，导出其似然函数，然后通过最大化似然函数来求解，由于，假设随机变量y服从伯努利分布，该假设是合理的，因为y=1的概率非1即0，而y=0则是它的反面。这样一种概率模型服从伯努利分布，后面在讨论广义线性模型时，可以明白，伯努利分布与高斯分布假设都是广义线性模型的一种特例。



似然函数：



最大似然估计：



同样，利用梯度下降算法有：



这里需要注意是求最大值，那么可以沿梯度上升方向作为搜索方向。

线性判别分析LDA（fisher判别分析）

GDA:生成式模型，应该来说高斯，伯努利，泊松，指数分布族都可以用线性模型建立。生成模型。

LDA：线性判别分析中把线性模型中的θ看作一个向量，把我们的特征向量X投影到θ上，这样就可以理解类间距离与类内距离。

**七 广义线性模型**

线性回归和logistic回归，基本的形式都是先设定，然后求最最大似然函数,然后采用对数似然估计,然后用梯度上升法或其它方法求出，二种回归如此类似的原因就是在于它都都是广义线性模型里的一员。

如果一个概念分布可以表示成时，那么这个概率分布可以称之为指数分布

指数分布家族：



下面先看之前线性回归和逻辑回归对应指数家族中高斯分布和伯努利分布只是指数分布家族中的特例，下面先看常见的分布对应指数分布家族的形式。

伯努利分布：



由指数家族形式有：



高斯分布：

由于在假设高斯模型时，方差的选取与最后的目标无关，为了简化模型，这里取。则有：



由指数家族形式有：



下面再从广义线性模型出发来导出一般的线性回归和逻辑回归

指数家族的问题可以通过广义线性模型来解决。如何构建GLM呢？在给定x和参数后，y的条件概率需要满足下面三个假设：

        assum1)      y | x; θ ∼ ExponentialFamily(η).

        assum2)      h(x) = E[y|x]. 即给定x，目标是预测T(y)的期望，通常问题中T(y)=y

        assum3)       ，即η和x之间是线性的

第一条假设是为了能在指数分布范围内讨论y的概率，第二条假设是为了使得预测值服从均值为实际值的分布，第三条假设是为了设计决策函数为线性函数。

下面再来看线性回归和逻辑回归



所以有一般的线性回归函数为。

逻辑回归



 值得注意的是：从广义线性模型出发，则有，而。

Softmax regression

逻辑回归常用于二分类，对于（不一定理解为y的取值为k，更应该理解为y可以取k类）多分类问题，可以看出该分类问题服从二项式分布，从广义线性模型出发。

首先证明二项式分布属于指数分布家族，满足假设1,2,3..n类

定义，且满足概率和为1。



为了使得二项式分布满足指数分布家族，定义



即，而



定义一个指示函数

则有，y还是数值，则是把y映射为向量的一个函数，而指示函数表示T(y)的第i个下标是否为1,取决于分类中y所属的第几类k的值。那么这里。

再来看的概率



其中



，



由上面三式可推出：



即



再由广义线性模型的第二条假设有：



最后由最大似然估计有：



这里再来对比一下线性回归和逻辑回归，可以看出其都可以从广义线性模型出发来建立模型，而且都是从三条假设出发，但是由线性设计策略导到期望再到目标函数，而最终的目标函数是，其原因是属性到函数值的线性设计策略再到二分类是一个复合函数问题，而中间到二分类的映射可以有多种，而广义线性模型推导出来时是sigmoid函数。所以本质上从广义线性模型出发，逻辑回归和线性回归没区别。

**八 多分类问题**

对于多分类问题，同样可以借鉴二分类学习方法，在二分类学习基础上采用一些策略以实现多分类，基本思路是“拆解法”，假设N个类别，经典的拆分算法有“一对一”，“一对多”，“多对多”，

一对一的基本思想是从所有类别中选出两类来实现一个两分类学习器，即学习出个二分类器，然后对新样本进行预测时，对这个分类器进行投票最终决定属于那一类。

一对多的基本思想是把所有类别进行二分类，即属于类和非两类，这样我们就需要N个分类器，然后对新样本进行预测时，与每一个分类器比较，最终决定属于哪一类。

多对多的基本思想是

第二部 分生成式模型

**第五课-第七课**

**1生成式模型（Generative Learning Algorithms）**

生成式模型主要用于分类，其思想是学习不同的类别样本的具有的特征，分别建立匹配模型，当预测新样本时，可以与各类模型进行匹配。

用数学形式表示即：

学习一个模型与，其意义表示在样本类别为条件下，样本为的概率，以及所用样本中的概率。

利用贝叶斯公式有：



其中



在具体分类中，分母是相同的，所以有：



可以看出判别式学习模型是学习一个属性到标签值得函数或者，而生成式模型则是更具类别数来学习多个和的模型。

**2高斯判别分析（GDA）(\*\*\*与朴素贝叶斯类似完全基于样本统计，只是一个假设条件概率服从高斯分布，另一个是离散属性下的硬划分概率)**

一种生成式学习模型，其中一个重要的假设是服从多元高斯分布。其作用是能合理的用高斯分布模型来生成该类的模型。

多元高斯分布：，其参数：mean vector，：covariance matrix 。若服从多元高斯分布，则其概率密度函数为



高斯判别模型（GDA）

如果输入特征X是连续型随机变量，那么可以用高斯判别模型来建立。前面的假设服从高斯分布是理想的假设，一般从得到的样本中我们不能完全确定X服从正态分布，但可以用高斯分布建模。



为了简化模型，其中。

则有



由：



似然函数：



最大似然估计后参数：

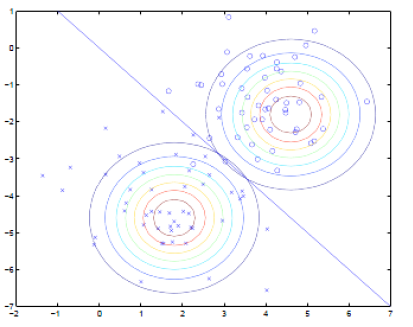


表示所有样本中y=1的概率

表示在所有y=0的样本中x的均值

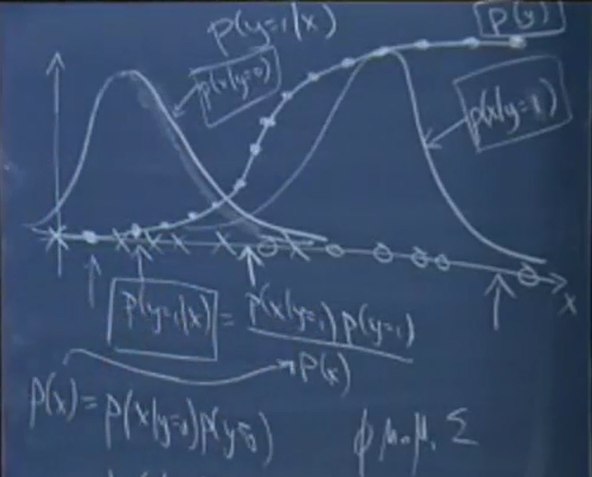
表示在所有y=0的样本中x的均值

表示样本特征的协方差



可以看出生成式模型通过建立两类样本的特征模型（常用高斯分布：随机变量为连续值），对于二分类问题，然后通过比较先验概率与的大小的大小，或者说后验概率与，类似于最小错误贝叶斯决策，同样可以得到一个决策边界，这与直接用判别式模型得到的的分类边界“一样”。

回过头来，再来看最小错误贝叶斯决策与一维高斯判别模型。有趣的是最后得到的决策函数也是sigmoid function。



高斯判别分析与逻辑回归比较

假如采用GDA来进行二分类，由贝叶斯公式导出后验概率等于似然函数与先验概率乘积，最后采用最大似然估计，最后导出后验概率的形式同样为：



下面比较下高斯判别分析与逻辑回归的区别：



不成立。所以GDA的假设比逻辑回归更强，但是不可以不想象，当的确是服从高斯分布的话，那么使用GDA建模在样本不是足够大的情况下同样能到达高斯分布的效果，而逻辑回归依靠的只有样本数据。所以当样本数据足够大的情况下，并且特征随机变量服从高斯分布的话，那么采用GDA则更方便，而逻辑回归则面临处理大量数据的问题。但是反过来，GDA模型的假设更强，模型的健壮性不如逻辑回归，而逻辑回归在样本上更具鲁棒性。出于这个原因，更多时候选择用逻辑回归比GDA更合适。但是假设特征随机变量服从泊松分布，而我们分别采用GDA和逻辑回归建模，那么GDA最后的预测结果则不如逻辑回归，而值得注意的是，属于指数分布族，那么可以推出是一个逻辑回归函数，逻辑回归同样适用，当然这个时候也可以用指数分布族服从服从来建模。

**3 Naive Bayes**

**在GDA，其特征向量x是连续，实值向量，下面讨论x是离散值。**考虑建立一个垃圾邮件过滤器，那么邮件的分类则对应与二分类问题。且邮件分类也是文本分类的一种。

目前代表性的做法是建立一个词典特征向量，其包含词典中出现的单词，进行如下表示，表示x的第i个分量，当表示第i个单词在邮件中出现，反之不出现,即。



可以看出x的取值范围为，假如把每一种取值看作一封邮件的特征，那么所有可能的邮件，特征个数为250000 个，其参数则有250000+1个。

而采用Naïve Bayes，首先需要如下假设：

每一个单词在所有邮件中出现的概率是独立的，即x的每一个分量在y类条件下都是独立的。

显然，上面假设在实际邮件中时不成立的，单词间出现的概率不可能是相互独立的，但是对于分类来说，还是可以。



下面针对上面的假设，我们定义如下参数：

对于每个单词在垃圾与非邮件中**出现**的概率表示为：







样本集为：m封邮件。

似然函数为：



最大似然估计有：



其中式1表示在所有的垃圾邮件中统计出现的次数所占的比例。式2式3同样。

现在假设来了一封信的邮件，其特征为x，用Naïve Bayes如下：



其中n表示邮件长度。

由二分类问题可知，所以上式与1/2比较即可。

拉普拉斯平滑：

为了使模型更具普适性，考虑当一封邮件中出现词典中有，但是在以往邮件中都没有的单词的情况，即如下情况：



显然上式无意义。

考虑这种情况，采用拉普拉斯平滑处理，其思想是对于分类问题考虑每一种情况出现的概率都是相等的，即，那么对应分子加1，分母加k。



同样，上述贝叶斯模型中只考虑单词是否出现，即单词服从伯努利分布，样本服从n次独立的伯努利分布（二项分布），而忽略了一个单词可能出现次数对邮件分类的影响。假设要统计某一单词出现的次数，那么有多项分布-multinational。只考虑单词是否出现的贝叶斯模型叫multi-variate Bernoulli event model，后者叫multinational event model。

下面还是邮件分类问题，引出multinational event model，在之前的模型中，我们首先建立词典，并且特征向量长度为词典长度，并且从词典出发，对于邮件出现过的单词，在对应词典的位置标记为1，反之标记为0产生一个特征向量X。对于线性回归，我们可能会直接从特征向量x出发求。朴素贝叶斯采用从某个单词在垃圾邮件与非垃圾邮件中出现的概率出发，并且假设每个单词出现是独立事件，从而有独立特性，进而得到先验概率和似然函数。采用贝叶斯公式得到后面概率，用最大似然估计来计算参数。而multinational event model则从邮件出发，表示邮件中第i个单词，其值表示第i个单词在字典中出现的位置，那么的取值则有其中V表示字典长度。这样一封邮件可以表示为，n表示一封邮件的长度。这相当于掷一枚有V面的骰子n次，将观测值记录下来形成一封邮件。

假设出现某一点的情况与第几次掷无关，也就是单词在邮件中出现的位置无关，而且每一次投掷都是独立的，即单词之间出现的事件是独立的。

参数有：







**注：的概率与j无关，j表示单词在邮件中的位置。**

这里可以看出参数同样不随邮件长度变化而变化，而是与词典长度一致（因为参数的值不依赖与单词在邮件中的位置，而是出现的次数，而下标表示的是出现在词典中的位置）。形式上与上面类似，但表达的意思却不同，上者表示词典中相应位置的单词是否在邮件中出现，出现记为1，参数则是在垃圾邮件与非垃圾邮件中出现的概率。下者表示的是邮件中从第一个单词直到最后一个单词出现在词典中的位置，并以其位置为下标表示该词在垃圾邮件和非垃圾邮件中出现的概率。M封邮件M个训练样本表示为：

，其中样本的特征长度由邮件的长度确定。

同样我们得到似然函数，先验概率如下：



最大似然估计有：



最后的决策函数为：



其中n表示邮件的长度。

从上面可以看出，朴素贝叶斯只适合处理随机变量离散取值的情况，对于连续值问题，可以采用随机变量离散化，即二分法产生多类，而对此产生的分类问题，这样问题不再是伯努利模型，而是二项式模型，同样可以采用上述模型，只是假设参数需要引入：



离散化处理如下：

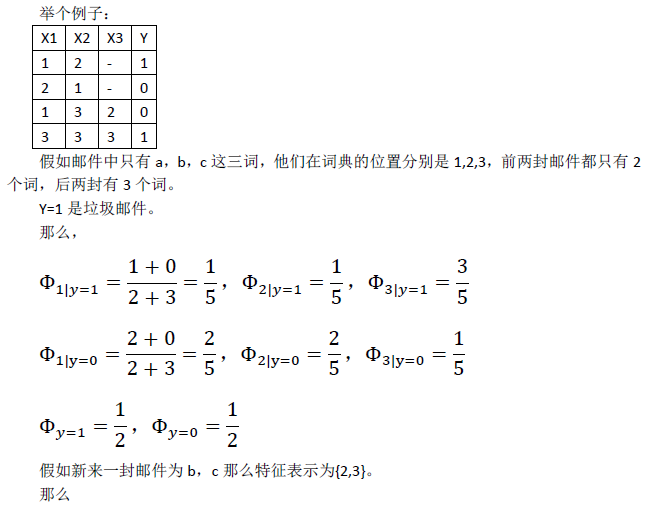
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | <400 | 400<x<800 | . | . |
| xi | 1 | 2 | 3 | . |

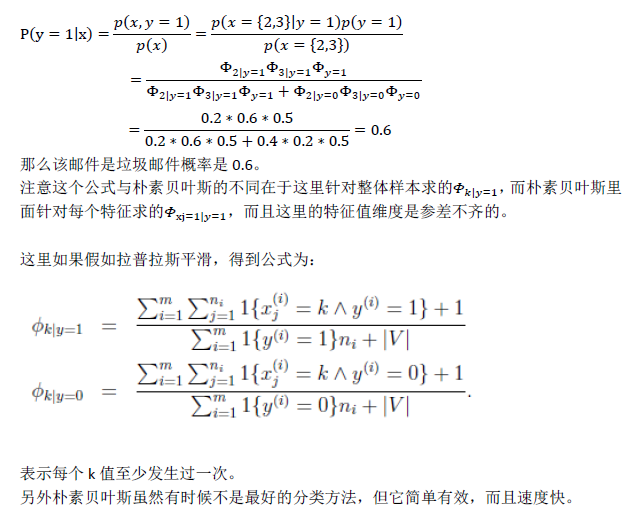
~~非线性分类器~~

~~神经网络- neurons network~~

~~在局部加权线性回归中，引入权重因子来更好的拟合非线性样本，神经网络算法类似在建立最终的决策函数中间加入隐藏层。~~

例如：





第三部分

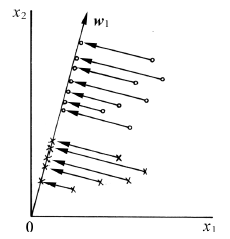
第六课-第七课

**支持向量机（SVM）**

1. 从线性模型到SVM

1.1线性模型与SVM

支持向量机中两个重要的概念是超平面和函数间隔/几何间隔，数学形式上也是在找一个（超平面方程），使得几何间隔最大化（线性可分）。这与线性模型中的目标很相似，但是两个的方法是不一样的，但有异曲同工其妙。前者是超平面方程，后者是线性函数，前者是为了直观的通过超平面来划分两类，使得样本到超平面的间隔距离最大，后者是将样本数据映射到一维数轴上进行分类。

![C:\Users\Administrator\AppData\Roaming\Tencent\Users\869233039\QQ\WinTemp\RichOle\NKKWBGZ)~D`]M](R3RCY71W.png](data:image/png;base64,)

从上面两个图可以看明白，线性分类与SVM（线性可分）都是一个线性模型，前者是在线性变换后的一维坐标上进行分类，后者是直观的给出超平面（线性变换）与样本的关系。超平面也可以理解为线性变换，变换后映射到与超平面垂直的一维坐标上（超平面的法向量）。所以说线性分类或者回归更为直观上的解释是SVM的线性可分特例。在逻辑回归中，当我们预测其为1，反之为0，或者预测为1，反之为0。从概率角度上讲，如果，那么预测为1的概率则越大，反之越接近0的，越难决策。同样，对应于SVM则是函数间隔，间隔越大越好决策，越靠近超平面，则越难进行分类。但是在SVM引入软间隔分类之后比线性分类更为灵活，在线性分类回归中目标函数考虑的是最小平方损失或者是对数几率损失加正则项，而SVM中考虑的是铰链损失加正则项。定义这样一种损失使得模型更加具有泛化性。而对数几率损失采用的是最大似然估计，铰链损失采用的是对于线性可分，对应于线性不可分情况，线性分类或者回归同样可以采用核方法对特征空间进行变换。

2.2支持向量机

在SVM中，采用如下定义:给定训练样本表示两类类别分别取值为1，-1。定义超平面与决策函数



可以看出是线性回归函数，而作为超平面，当新的样本代入时，不同的两类，z分别位于超平面两侧，可以作正负区别。即z的正负代入有两类之别。通过上述定义可知，超平面对应的函数也可以做分类函数。对于该超平面的方程可以看作是平面方程的点法式确定的，即平面上一点与平面法向量确定，即



函数间隔与几何间隔，定义函数间隔如下：



为了更好的表示函数间隔是一个标量，其正负只表示分类的正确性。函数间隔可以理解为样本实际类别与样本通过线性变换后的坐标值的乘积，坐标值的正负表示决策类别，值表示函数间隔值。这样，实际类别与决策类别的乘积的正负就表示分类的正确性。需要注意的函数间隔值会随w,b的放大而放大，同样函数间隔也就放大。这里的放大其实也可以理解为坐标轴的变换，但是几何距离是不变的。在SVM中，定义的函数间隔与几何间隔都是以间隔的下限来衡量，即取所有样本中最小的间隔来作为函数间隔或者几何间隔，而SVM的目标就是最大化最小间隔。可以看出当时，意味着线性可分。



下面，从下图中定义AB两点的几何距离为，当我们以超平面和法向量建立二维坐标系的话，已知A的坐标为,那么B的坐标可以用下式来表示：

![C:\Users\Administrator\AppData\Roaming\Tencent\Users\869233039\QQ\WinTemp\RichOle\NKKWBGZ)~D`]M](R3RCY71W.png](data:image/png;base64,)



由于B点在超平面上，那么就有



也就是说所有的样本点减去几何间隔就变成超平面上的点，也就是把样本点投影到超平面上。所以，几何间隔又可以表示为：



同样考虑分类结果的正确性，几何间隔引入实际类别标签，则有几何间隔如下：



当=1时，函数间隔等于几何间隔。

同样，考虑最小的几何间隔作为SVM的几何间隔：



最大化最小几何间隔就是SVM中对线性可分情况的一种直观理解。所以目标函数如下：



下面逐步优化该问题：



取



该式为SVM最终表达式，需要注意的是大于1不是表示线性可分，而是保证最小间隔大于1。

二、拉格朗日乘子法与对偶优化

对于SVM最终表达式，可以看出来是典型的带约束最优化问题，下面再来看拉格朗日乘子法，然后给出SVM的拉格朗日乘子法最终表达式。

考虑一般的约束的线性优化问题：



下面从下降方向和可行方向的几何意义来看上述问题。

下降方向：，表示在的下降方向。

可行方向：



其中I表示起作用约束，而在起作用约束上，可行方向就有范围，而对于不起作用约束，则无可行方向而言，对于等式约束，则是必须满足的。

所以，当点满足：时，表示必定为局部最优点。同时有Fretz John条件有代数表达形式如下：



左右两边同时约去有



反之不然。

需要注意的是对于不等式约束，一定是起作用约束才有，不起作用约束。对于等式约束，同样必须满足。至于为何要，是因为在不考虑等式约束的情况下，等值线上的点都是局部最优解，而考虑等式约束，那么局部最优解除了在等值线上相等以外，还需在等式约束直线上，所以最优点必然是等值线与等式约束直线相切的点。用代数表示为

下面对目标函数进行扩展为拉格朗日函数形式：



因为不等式约束中，把不起作用约束也放到拉格朗日函数中，所以当且仅当：



所以有：



其目标函数优化问题就是最小化拉格朗日函数最大值，其条件为：



可以看出扩展后的拉格朗日函数的条件与之前的目标函数的KK-T条件等价。

下面再来看其对偶问题：



其中



所以



对于上式求解，可以先把看作变量，对其求导在极值处取得极小值，即x用表示，然后再对目标函数求解最大值确定原问题的下限。

至于为何对偶问题为原问题提供下限：

对偶问题：



原问题：



又因为：



对偶问题为原问题提供一个下限，一定条件下对偶问题的最优值等于原问题的最优值，对偶问题的最优解可以推导出原问题的最优解x。

1. SVM拉格朗日函数的对偶问题

3.1硬间隔SVM

在了解完拉格朗日乘子法与对偶优化后，再来看最优间隔分类器：



其拉格朗日函数可以写成：



对拉格朗日函数，b求梯度有：



所以有原问题的下限：



所以原问题的对偶问题为：



可以看出对偶问题的导出也是原问题求解过程中的一步，即求解原问题的下限，有二次函数性质有，极值点为极小值，所以有下限在导数为0处取得，而下限用

表示，即为原问题的对偶问题，然后对其对偶问题求最大值即为原问题的下限。再次运用



在求解对偶问题后再求解：



其b的几何意义为正负样本中最小间隔的均值。

3.2软间隔与正则化

在上面所讨论的SVM方法很重要的一个假设就是样本是线性可分的，可以想象该假设与假设模型符合高斯分布不一样，高斯分布假设是“没有办法中最好的办法”，而且是合理的，尽管在假设不成立的情况下，高斯判别分析也能有起作用。而假设线性可分则是我们理想中的假设，不成立的话上述SVM方法就没有解，更不会有最优解。下面就引出一种软间隔SVM方法来解决该问题。引入允许样本分类误差，同时引入惩罚项，对其进行惩罚。那么SVM的原问题形式可以写成软间隔SVM如下：



同样对软间隔SVM求拉格朗日函数：



同样考虑其对偶问题，为确定其下限先求导：



代入原问题有其对偶问题：



这里可以对对偶问题做一个定性分析

当时，则有，那么，所以有

当时，则有，那么，所以有

当时，则有，那么，所以有

在求解对偶问题后再求解：



3.3损失函数加正则项的一般理解

可以看出是为了最优间隔-间隔最大化（同样有稀疏的作用，但稀疏性常采用L1范数），是惩罚项。关于一般情况下目标函数定义为损失函数加正则项的形式可以参考

<http://blog.csdn.net/zouxy09/article/details/24971995>

其中常用的惩罚项定义有下：

Hinge损失：SVM

指数损失:boosting

对率损失logistic

平方损失 liner 

而对于采用不同的损失函数，SVM又可以衍生出很多非常好的特性，也非常值得讨论。这里值得一提是一般实际问题上，我们并不希望一定要间隔最优，有时模型简单，健壮性强，分类正确就好，所以目标函数更多时候会选择下面形式：



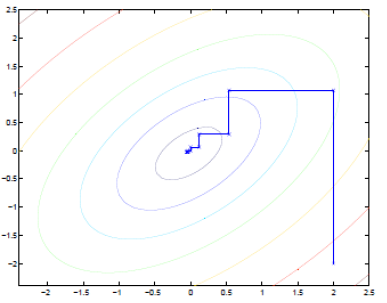
前一项是损失函数的定义，后一项是采用范数来使模型简单的约束。

四、SMO算法

在定义好SVM的优化函数之后，如何求解参数值便是一个优化问题了，常用的优化算法有梯度下降，牛顿法，共轭梯度法等，这些算法都是最优化理论中常用的求解优化问题的算法。而对于SVM，又有其针对性的优化方法SMO， SMO结合了SVM优化模型中的条件。在介绍SMO算法之前，先来看一下一个简单的坐标上升法，坐标上升算法的思想是对于多个参数的优化求解问题，可以每次只考虑一个变量，而固定其他所有变量，对一个变量进行目标优化，内循环每一个变量进行优化，外循环直到迭代到收敛。



如图是一个两个变量的一个优化问题，因为每一次只改变一个变量的值，所以优化路径与坐标轴平行，椭圆线表示二次目标函数的等值线。迭代过程会最终收敛到极小值。



不难发现，SVM优化是一个二次规划问题，可以使用二次规划算法来求解，然而，该问题的规模（参数）正比于训练样本数（引：周）。由于常用的优化算法一次迭代是求解所有参数，这样会遍历所有样本。导致计算开销大。当然迭代的收敛速度肯定是比坐标上升思想要快。而且SVM是一个带约束问题，一般的坐标上升法一次改变一个变量是不满足约束的。但是SMO算法中一次更新只遍历两个样本，即更新两个参数，这样使得计算开销是可接受的，同时启发式搜索也使得迭代速度提高，所以SMO算法在SVM中更为常用。这一点，也是SVM能支持无限维一个重要技巧。

SMO算法是platt提出针对SVM的一个高效的求解算法。其想法与坐标上升算法类似，唯一不同的是SVM的优化模型中有一个条件是，所以如果每次只改变一个变量的值，那么必然会导致解违反KKT条件。所以SMO算法每次取两个变量进行优化，一个主动变化，一个是被动变化的。

下面先看SVM的优化模型。



SMO采用一种启发式算法，每一次都选择违反KKT条件最严重的一个变量，并随之选择一个使得能够有最大变化的另一个变量同时进行优化，固定其他参数不变。优化的思想根据，用来表示，这样就得到了一个关于的一个二次规划问题。同样是取到极值点，即关于进行求导等于零。求解出，但这里需要注意的是每个同样受限于，所以在修改，时，还需要进行一个修剪，即保证，的上限和下限。取到极值点是对的主动修改，而随之的修改是为了满足条件。下面表达式中为主动修改变量，为随之修改变量。



通过上面的变换思想，由，有 



先对目标函数去复数变成最小化问题，那么目标函数可以化成：



这里用来表示有，并对进行求导有：



代入上式，并记

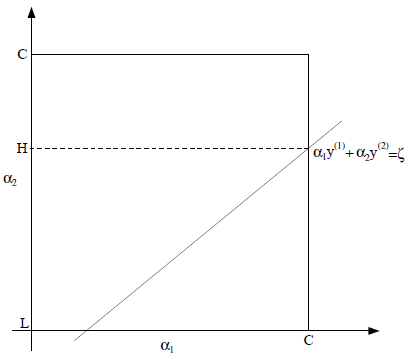
，。

，可以看出E1，E2就是预测误差。

则有：



又因为，所以有：



所以，都有一个上下限。



所以实际上对进行修改采用如下规则：



在对修改完之后，采用等式求解。

最后再来看b，再求解出之后，由KKT条件有：



且根据预测函数与实际类别差值为来确定b，需要注意的是，每一次修改了两个变量，这样b的确定依赖于两个变量，，同样，考虑KKT条件，如果，同时满足，那么就有选择的两个样本都处于支持向量上，即离超平面距离都为1，这是有b1=b2，如果其中有一个等于0或等于C，那么b1和b2之间的数都满足KKT条件，这时可以取中间值作为。需要注意到是，b的更新是在每一次确定，之后，而每次的更新都是迭代的过程，为的是让样本尽可能的分类正确，不是最终结果。而前面的



指的是最终结果选择决策正负样本中离超平面距离最近的两个取中间值。

五、核方法：

以上讨论的硬间隔SVM还是软间隔SVM，都是针对线性分类问题，如果分类问题是一个非线性的，那么可以采用核方法。核方法不仅仅应用于SVM，同样适合其他的学习问题，核方法更像是一种思想，对特征空间进行非线性变换。

在进入SVM核方法前，我们先看看核技巧的实现过程。

考虑到一个非线性问题，一般情况直接进行线性分类效果是不好的。但是如果还是想用线性分类器进行分类，那么可以对原始的输入空间进行非线性变换转换为线性问题，而这种非线性变换其实就可以理解类似于PCA的特征变换，只是PCA是一种线性变换，且主要目的是降维，而核技巧的非线性变换的目的是将原始输入的非线性空间变换到新的线性空间，对于这个新的空间也成为特征空间，数学上称为希尔伯特空间。

核函数的定义：设是输入空间（欧式空间）又设为特征空间（希尔伯特空间），如果存在一个从到的映射使得对所有的，函数满足条件，则称为核函数，为映射函数。

由核函数的定义可以看出，核函数的实现可以采用定义映射函数以及特征空间来求得。反之给定核函数是无法唯一确定映射函数和特征空间的，这也就说明了核函数的定义是一种隐式的定义映射函数和特征空间。当然核技巧更多的不是关心非线性映射和特征空间，而更多的是直接在核函数上进行运算。这样不仅减少了计算量，同时也解决了维度灾难问题，因为我们并不关心映射变换和映射变换后的特征空间，而是直接在核函数的基础上进行分类处理。下面借助核技巧在SVM上应用来解释上面两个优点，计算量和维灾问题。

首先观察优化问题的对偶形式可以看出内积操作，而在计算中，往往会因为特征向量的维度过高，而造成的运算量大的问题，而且在使用这些特征之前，常有特征映射这一步，即把在原空间上线性不可分映射到更高维使其线性可分。那么就存在一个特征映射函数，所以在计算内积前往往会有特征映射一步。而在我们优化问题中，我们并不关心映射后的样子，所以假如先进性特征映射计算后求内积，计算复杂度会增加，所以引出核函数的概念。

下面先看核函数的引入是如何在内积操作中忽略特征映射而简化计算。假设特征映射函数：



核函数：



现在假设特征映射函数，那么核函数则为：



如果，我们计算，先计算的话，那么必然有：



如果从核函数出发，先计算内积，在做特征映射，生成这样一个核函数只需要。

先内积后做特征映射运算



先特征映射运算后内积



再者，输入空间非线性的，假如不是直接定义核函数方式来进行线性分类，而是通过映射函数在输入空间上进行非线性变换，那么对于映射后的特征空间必然也是一个高维空间，这样可能会导致数据的稀疏以及维度灾难问题。而核函数则不考虑映射函数和特征空间，核技巧更像是一种相似性度量，且对于不同核，这种相似性度量的标准也不同。对于m个样本而言，核函数一般为m\*m的矩阵，其中x，z都表示一个样本。说了这么多核函数的优点，下面还得看看核函数的限制，不是随随便便就可以定义一个核函数的，这里给出一个直观理解核函数的性质，严格的数学证明可以参考统计学习方法-李航。

核函数表示两个样本进行非线性变换后内积表示核函数（矩阵形式）的第ij个位置的值，可以想象同样的映射作用于两个样本后内积，如果交换两个样本的顺序，对内积操作来说必然满足内积的定义，那么可想而知是一个对称的矩阵，再者，考虑主对角上元素，必然满足相同元素内积非负性，所以有矩阵K对称且主对角元素非负，可以导出K是半正定性，所以K的性质有对称半正定性，而这些特性都是内积操作的性质。



值得注意的，对于上述形式的优化问题通过研究其对偶问题，我们可以把



写成：



那么对于这一类优化问题，我们都可以采用核方法，所以这也就是引出了这一类方法-核方法。

下面再来讨论核函数在SVM中作用

1：应该说特征映射的作用是为了更好的拟合或者是为了映射到线性可分空间。但是可以从优化目标中可以看到只需要计算内积，而不关心映射变换和映射后的特征空间。而核函数就是代替特征映射在内积处理这两步的。而对于至于什么样的核函数好是没有固定的。而且先映射后内积是直观的理解，而核函数的先内积再映射是简化计算。

所以当特征映射确定后，那么与之对应的核函数必然是确定，同样核函数确定后与之对应的特征映射也唯一，但是却不能求出具体的特征映射值，而我们也不关心特征映射后的值。



也许我们并知道与高斯核对应的特征映射，但是高斯核确实有映射到高维后线性可分的作用。计算上直接运用高斯核肯定也会比用与之对应的特征映射函数再内积的方式更简便。如下图可见高斯核效果

六、支持向量机回归

支持向量机回归,SVM回归，我想这个时候可以抛开超平面和最大间隔这两个概念了，因为超平面对于二分类问题来说是非常直观有效的，但是对于回归来说确实不合适的。但是支持向量机回归还是借助了SVM的软间隔的概念。同样数学表达形式上与SVM分类一致。

结合线性回归中的lasso回归，其目标函数包括L1范式的正则项和平方损失函数。



在支持向量机回归中同样采用该目标函数，正则项采用二范数，首先抛开超平面的概念，定义一个线性映射对于每一个样本采用如上变换到一维，由于是软间隔，所以回归值与实际值之间允许一定的误差，这也就导出了SVM的铰链损失。这样一来就有如下表达式



将上式转化为SVM最终的形式有：



其中和是引入的松弛变量。再看其拉格朗日函数和对偶问题，具体参看周老师的机器学习。

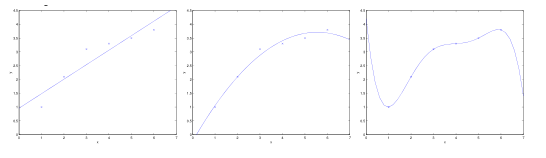
**学习理论**

**第九课-**

<http://www.cnblogs.com/zhizhan/p/5483574.html>

偏差与方差的概念

机器学习中偏差与方差这两个概念的定义在数学上讲还是非正式的，最直观的理解可以从下图理解，后面在经验风险最小化中给出偏差与方差的定义。

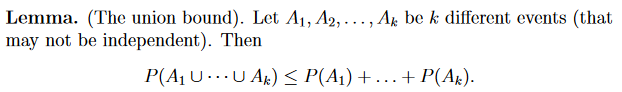


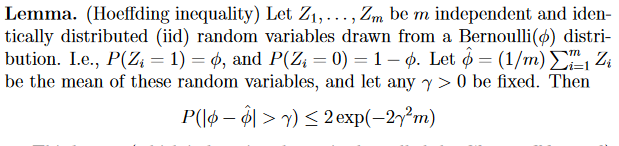
从图中可以看出当我们用不同的模型去拟合样本数据时，会发现第一个模型对样本的拟合程度不够，出现欠拟合问题，第三个模型对样本拟合的很好，但模型的复杂度也相对复杂，出现过拟合现象。而对偏差与方差最直观的理解就是，偏差与训练误差相关。第一个模型偏差过大，第三个模型方差过大，在训练误差上也许第三个模型更为准确，但是衡量一个模型的最终目标还是一般误差。而对于第三个模型的一般误差的预测不仅仅体现在训练误差-偏差上，还有方差，模型过于复杂不仅提高了算法的复杂度，而且增加了方差，所以最终的一般误差由偏差和方差决定的。

下面来看经验风险最小化ERM

在介绍ERM之前先来看下两个引理：

联合界引理：



该引理表示多个事件和的概率不大于多个事件概率的和，这是明显的。

该引理表示一个独立同分布的随机变量其偏离期望大于的概率有上限。可以从高斯分布出发其到均值距离大于的概率有上限（切比雪夫不等式）。

一致收敛概率界

对于二分类问题，同样，样本为，那么训练误差为。



一般误差则为：



现在我们以线性分类器为例：

在线性分类器中，我们要学习的就是：



可想而知，不同的，则对应不同的。现在，我们假设有这样一个假设类，假如是连续的，那么假设类必然包含n+1维下所有的。但是这样一来假设类就是无限的，如果我们离散化处理，使其有限化，且不失一般性。那么我们训练出来经验误差最小的一个必然也在假设类H中。所以经验最小化的目标就是在假设类中分析的一般误差，**更为重要的是分析模型的一般误差，从而选择合适的模型。**



下面给出一些符号的注解：

，其中是完成x到{0,1}的映射，也就是说是分类问题。

表示训练误差，表示一般误差，指示函数。

那么该假设类模型中具体一个的训练误差为：



独立同分布，在分类问题便是独立同分布于伯努利分布，由引理2有：



再由引理1有，整个模型的存在训练误差大于的概率为：



其对立面为：



也就是说整个模型的一般误差与训练误差的偏差在的范围内的概率有下限-一致收敛概率界。

**样本复杂度界：**

现在假设我们要求我们的一般误差与训练误差的偏差在的范围内的概率下限为1-。那么我们根据上式可以推出：



从这个式中可以分析，当我们需要一般误差与训练误差的偏差在的范围内的概率下限为1-时，需要的样本数有下限-样本复杂度界，作参考。同时可以看出训练样本的数量与一般误差与训练误差的偏差成反比，与假设类的复杂程度K成正比，与下限成反比。

误差界：



该式表示至少在的概率下，对假设类中任意的有：



再次说明：

：是假设类中训练误差最小的

：是假设类中一般误差最小的

那么有：



第一个不等式：对于在训练误差最小的一个假设类，其一般误差必然小于训练误差加偏差，由引理1.

第二个不等式放大偏差，是假设类中训练误差最小的一个，那么必然有。

第三个不等式，再次应用引理1，其训练误差必然小于一般误差加上偏差。

这样一来有-**定理：**令H为有限的假设类，|H|>k，令m和δ固定，至少在1-δ的概率下有：

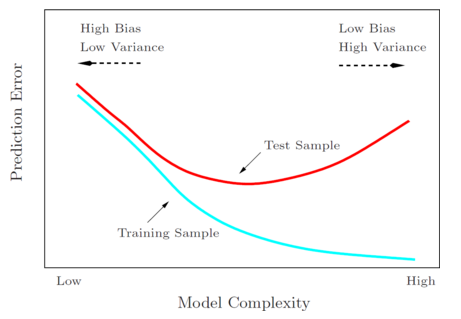


在假设类中，训练误差中最小的一个服从方差为，偏差为



的分布，这样就可以说明，在模型选择中，我们要权衡方差和偏差问题。

如果选择更复杂的目标函数或更多特征的类H’， 例如，将线性假设类换成二次假设类，从假设类中学习到训练误差最好的目标函数只可能更好，所以不等式右边的第一项（偏差bias）会减小，但代价是k会增加，从而第二项（方差variance）增加，这就是偏差方差权衡，可以用下图更具体的描述。



随着**模型复杂度**（如多项式的次数、假设类的大小等）的增长，**训练误差**逐渐降低，而**一般误差**先降低到最低点再重新增长。训练误差降低，是因为模型越复杂，对于训练集合的拟合就越好。对于一般误差，最左边的端点表示欠拟合（高偏差），最右边的端点表示过拟合（高方差），最小化一般误差时，一般倾向于选取中间的模型复杂度，最小一般误差的区域。

最后介绍上述定理的**Corollary推论**：

 令假设类含有k个假设，|H|=k，给定γ和δ，为了保证：



至少在1-δ的概率下，满足条件：



**注：更多时候，由于界的条件很宽，所以得出的界具备参考的价值不大，更多时候是直观的理解，需要样本数的大小与复杂度成正比，与误差范围成反比。**

VC维

前面已经介绍在假设类是有限的条件下，经验风险最小化可以得到假设类模型的一致收敛界，样本复杂度界，误差界。但这些都必须建立在有限假设类条件下，下面来从VC分析无限类情况。

考虑给定假设类，给定一个样本集，其VC维是能被H打散的最大示例集合的大小即：H存在最大能完全表示的所有样本集合的元素个数（与样本的分布无关），一般来说：等于假设类的参数个数。

VC定理：给定假设类，，那么至少以的概率满足：



进一步有：



这里拿SVM与线性模型作比较：

1假如我们用线性模型，其参数则为特征维度k+1,那么,其最大能打散的样本集合的元素个数为k+1,也就是说不能扩展到无限维。

2假如采用SVM，采用核函数方式，其参数个数可以映射到无限维，而且对于软间隔SVM中至少有样本数个参数（损失项），可见其VC维是无限维。因此也称SVM支持无限维分类吧。

正则化

这里再次强调一下：SVM中定义的损失项的惩罚函数的定义，也是学习理论中中很重要的一种方法，也称正则化过程。当然还可以把其他目标放到正则项中，如范数（模型复杂度）。

模型选择

在模型选择上，最常采用的一种方法就是训练-验证-评价。也就是说把训练数据划分为训练集-验证集-测试集。

训练集：获得模型及其训练误差，用来训练不同模型；

验证集：与训练集相对独立，获取训练模型在该集上的预测误差，用来做模型选择；

测试集：与训练集和验证集独立，获得一般误差和其他模型评价指标，用来评价已选择出的模型。

常用的验证方法有:

交叉验证法（hold-out cross validation）：

1随机的分割训练样本为和，一般和，前者为训练集，后者为验证集。

2为每一类模型在训练集上学习，每个假设类得到一个目标函数。

3对这一些目标函数在验证样本上进行验证，得出

K-fold cross validation：

1随机的把样本分为份，其中m表示样本数量。这样就得到了训练子集。

2对于每一个模型，从1到k选择留下一份作为验证集，其余的作为训练集，进行k次训练得到k个，记为，然后在上进行验证，求均值作为该模型的验证误差。

3选出模型中验证误差最小的。

一般情况下较常用，也称十重交叉验证。特别的在样本数非常少时，当时，称为留一法leave-one-out cross validation。

特征选择

模型的选择当理解为选择合适的假设类，那么具体是选择线性模型，多项式模，SVM等都是假设类中很重要的一部分，同样，特征选择直接决定模型所需的参数，同样也决定着假设类的复杂程度。所以特征选择也是模型选择的一部分。特征选择常用的方法有封装（wrapper）和过滤（filter）。

封装法是一种启发式规则搜索算法，有向前搜索和向后搜索。

Forward search：

1初始化特征子集合为

2从，设置，然后采用交叉验证法来评价，选择其中最好的，并重新设置。

3判断特征子集合是否满足要求，否则重复2。

同样，向后封装算法是初始化特征子集合为全集，然后不断的选择删除一个特征，同样采用交叉验证选出删除一个特征最好的子集合直到，或者满足结束条件。

过滤法是采用一个衡量标准对每一个特征进行计算其与y的相似度。



这里假设与y都是二值型。

可以看出，分子表示特征与类别的联合概率，分母表示特征与类别的独立概率积，由联合概率可知，当两个事件独立则有，则说明关联性越低，即该特征对分类的价值越低。

贝叶斯估计与正则化

<http://blog.csdn.net/pipisorry/article/details/51471222>

前面在概率模型上常用的参数估计法是最大似然估计，其目标是在使得在给定参数下，似然函数最大化，即统计所有的样本，使其概率最大，并且在这个估计过程中把参数看作是条件概率中的参数，不作随机变量。



可以看出，最大似然估计是在统计样本，所以参数的值完全由样本分布决定。这也是频率/统计学派的观点。

这里举例说明，同似然估计中一样，其形式与模型有关，假如采用贝叶斯逻辑回归，那么



其中。

另一个是贝叶斯学派，则认为是一个随机变量，并假设的先验概率为。

同样对于训练集，那么有：



到这里，MAP（最大化后验概率）在不考虑分母的情况下，直接对似然函数与先验概率乘积最大化，以近似达到后验概率最大化。



由最大化后验概率进一步推导出的BIC法则：



而贝叶斯估计则不然，其采用关于随机变量的全期望展开作为样本预测：



比较最大似然，最大化后验概率，贝叶斯估计，我们会发现，最大似然是把看作是一个不知道但确定的参数，所以不把其作为随机变量考虑其概率分布。在参数估计上只考虑样本的分布来使得似然函数最大化。而最大化后验概率与贝叶斯估计则把看作随机变量，作为先验知识处理，考虑其先验概率。但是为了计算方便，最大化后验概率是用后验概率近似的等于。而贝叶斯估计则采用全期望公式来进一步表示最后的概率。

C:\Users\Administrator\Desktop\gif.latex.gif

可以看出，贝叶斯估计是“先验知识”+“样本信息”=“后验概率”，贝叶斯估计中分子为似然函数与先验概率的乘积，分母是证据。

值得注意的是，不论是最大化后验概率还贝叶斯估计，其先验概率一般采用常用的分布的来给定，如高斯分布，拉普拉斯等。

比较考虑先验概率与不考虑先验概率，我们会发现最终的目标函数是在优化一个

最大似然估计的优化目标函数



贝叶斯估计的优化目标函数：



可以看得出贝叶斯估计的第二项与正则化项中简化模型一样，所以当我们采用最大似然估计求解时由于完全考虑样本的分布，会出现过拟合现象，拟合曲线护出现尖刺。而贝叶斯估计则把先验知识考虑进来，到达简化模型，有防止过拟合作用，拟合曲线会变得平滑。

**注：关于贝叶斯估计还有很多值得深入学习的知识，有些概念似懂非懂。**

在线学习

机器学习实际问题分析：

1学习算法的调试与诊断

常见的问题有训练误差与一般误差不匹配问题，可以尝试从偏差与方差权衡上调整，再比如训练误差与一般误差都达不到预期的效果，可以从优化算法上尝试，比如迭代收敛上不到最优值。再者目标函数定义不够准确，不能实现高检测率和低误检率。对于目标函数常用的调整如惩罚项系数。最后在考虑模型选择上是否合适，是否选择其他模型会更好。

1获得更多的训练样本

2增加更多的特征，或者选择更好的特征

3减少特征

4尝试不同的优化算法，梯度下降，牛顿法。

5调整优化的目标函数

6选择其他的模型，SVM。

2误差分析和销蚀分析

在使用机器学习算法时，要实现一个系统往往需要一系列的处理，对于每一部分称之为系统的一个组件，而要对系统进行误差分析时，需要对各个组件进行分析，最常用的方法是逐个增加与逐个去除的方式进行误差分析，以确定各个组件对系统的影响，那么最终对系统影响最大必然是后期优化的目标。

3机器学习设计过程

机器学习设计过程主要有两种方式：

1逐步实现，逐步分析，直到最后成型。

2先建立快速简单系统，然后对系统逐步优化。

需要注意的是，逐步实现的过程中不需要对每一步进行过早的优化，因为在系统还未出来前并不确定优化是否对最终的系统有效。

**无监督学习**

第十二讲-

在无监督问题中，给定数据集，并对数据进行聚类，其中，而没有类别标签。

K-means算法：

1初始化每一类的聚类中心，随机。

2重复下面步骤直到收敛

2.1遍历所有样本，根据距离哪一类类中心近划分到哪一类。



可以看出，表示第i个样本的所属类别，其值为离k类中心最近的一类j。

2.2遍历所有样本，对每一类，更新类中心为该类样本的均值。



可以看出，第二步是根据上一步类中心划分的类别重新更新类中心。

整个K-means算法可以看做是先随机选取类中心，然后根据类中心划分后的样本重新更新类中心，直到类中心不再变化，即收敛。

注：K-means聚类算法与之前的监督算法相比可以看做是先假设类标签，然后期望最大化再次更新类标签，更新类标签而后再期望最大化这样反复迭代直到类标签不变。

这里来看K-means算法的收敛问题，下面定义失真函数：



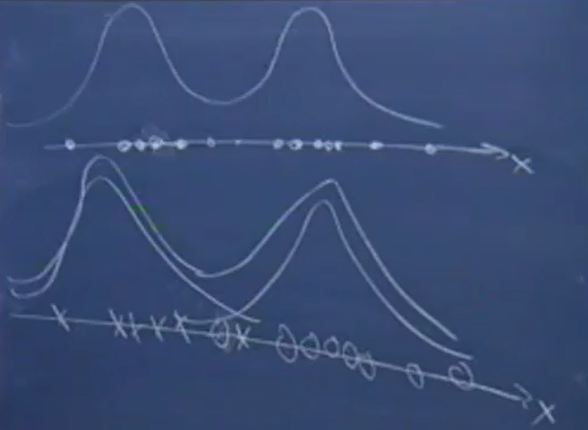
其中c是类别数，一般随机选择，当然也有学习算法自动选择。J函数表示每个样本到其类中心的距离。我们的目标就是最小化目标函数。首先初始化类中心之后，调整样本所属类别使得失真函数最小化，然后根据新的类别划分来调整类中心，两个过程相互循环使得失真函数最小化，c作为划分类别数在此过程中不变，但是样本所属类别在使得失真函数最小化过程中反复调整，同样随着样本类别的调整类中心也在不断更新，最终当失真函数达到局部最小，所属类别不变，类中心也不变。

由于失真函数J是非凸函数，所以最后的最小化也是局部最小，可以尝试通过初始化不同类中心，在多次的局部最优解中选其一。在初始化或迭代过程中出现类中心无一样本时，该类中心不起作用，可直接舍去。

K-means算法的思想与EM算法思想一致，E步：在假设参数（分布）下（假设类中心）期望最大化来求出最好的类别划分，M步：期望最大化来调整假设参数（假设类中心），混合高斯模型也是EM算法的一个特例，是在假设高斯分布的条件下使用期望最大化思想，后面在EM部分，继续比较K-means，混合高斯模型，EM算法，EM算法更为广义。

混合高斯模型：

K-means算法从距离的角度定义所有样本都是以1的概率属于某一类别。而高斯混合模型对于所有样本都是以一定的概率归为某一类高斯分布。在假设每一类的样本之间都独立，且服从多元高斯分布。在这个假设前提下的期望最大化来求解参数，该模型也称为混合高斯模型。以一维样本举例：



在混合高斯模型中，我们假设在样本中隐藏了样本类别标签，同时隐藏了每一类样本的分布。那么首先假设隐藏的类别标签服从二项式或者伯努利分布，那么有：



对于二类聚类，那么有：



那么样本与类标签的联合分布为：



同样对于每一类样本都服从多元高斯分布，那么有：



所以整个模型中的参数有，其中为每一类的概率，为每一类服从高斯分布的均值，为每一类服从高斯分布的协方差矩阵。而隐藏的类标签只是模型的一个假设，用于连接样本与类标签的联合分布概率。然后用最大似然估计：



可以看出最大似然估计左边是参数的似然函数，右边还是概率。而第二步是全期望公式。对于上式由于等式右边没有解释解，由于是聚类问题，只需要保证类别正确即可，这也就表示考虑最大似然时只需要考虑z的实际类别（假设已知，其实未知），当然也可以理解为考虑最相似的一类的进行最大似然估计，所以k类中只考虑最相似的一类（实际类）：



最后最大似然估计求解参数有：



可以看出为每一类样本所占的比例，为每一类高斯分布的均值，为每一类高斯分布的协方差。

结合EM思想，混合高斯模型算法的步骤如下：

循环下面步骤，直到收敛{

初始化参数，以及隐藏类别。

E-step：对所有样本，划分为 的概率



可以看出第二等式是贝叶斯公式，分子表示划分为的条件下的联合概率与的先验概率的乘积。而分母则表示每一类下所有样本在其所属类条件下概率的总和。

M-step，更新参数



}

实际上，在假设类标签之后，并初始化或者更新（也就是确定）之后，混合高斯模型与高斯判别分析模型近似。前者是类标签服从二项式分布，后者是类标签服从伯努利分布，前者是用多个多元高斯分布来拟合多类，后者一般用两个多元高斯分布来拟合二分类。所以最大似然估计求解处的参数形式表达上一致。

最后再来看EM算法，混合高斯模型只是EM的一个特例，其思想是期望最大化。

EM算法：

在讨论EM算法之前，先介绍Jensen inequality（由凸函数性质导出）

假设f是定义在实数上的凸函数，由凸函数的定义有：



严格凸函数则严格大于，凸函数的判定是其二阶可微的话，其Hesse矩阵半正定。

对凸函数的性质推广，存在



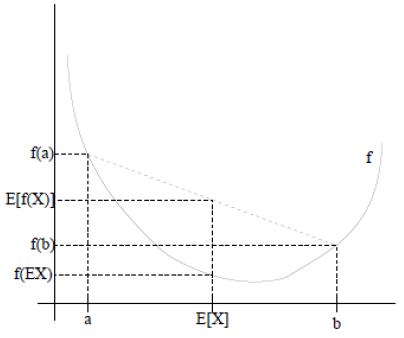
那么有：



当表示与之对应的的概率时，那么有：



当且仅当：，即为常数函数。



反之，凹函数不等式反之成立。

EM算法：

给定训练样本，引入隐含的类别标签，同样采用最大似然估计使得的联合概率最大：



由于在样本中，我们能观察到的只有样本特征X，对于隐藏的类别标签是我们假设的，在E-step，根据的联合概率来得到似然函数，在M-step，最大化似然函数来更新假设，如此迭代过程如下图。

EM算法的思想就是通过不断的迭代来尝试找到最优的类别标签假设，这也就完成了我们最终的聚类问题。

现在在不考虑实际分布的情况下，假设有



第二步从代数上理解，即分子分母同乘一个常数（常数由确定），由于该常数生成函数满足概率的定义，在把看作随机变量的情况下，那么，如果把等式1看作是随机变量的全期望，那么第二个等式就是在构造



即同乘一个常数生成函数相当于在原来随机变量的基础上进行如下映射，把

随机变量映射到随机变量，而把作为新随机变量的概率分布。

又：

又可以知：



由函数是凹函数，再根据Jessen inequality



可以推出最后的不等式。

这个过程也是在对似然函数求下界，而对于Qi的选择就对应不同的假设。

由Jessen inequality有当且仅当等式成立。

为了使等式成立即：



即

又



那么现在对于的假设就是要满足等式成立。也就是要说只是连接的中间件。对应于混合高斯模型中对样本预测为j类的概率。同混合高斯模型一样，函数（的概率分布）是由贝叶斯公式转换为似然函数和先验概率的乘积求得，其似然函数是对样本根据假设（更新）的类别标签（，在混合高斯模型中，有确定参数后的高斯模型）统计而得。

所以EM算法如下：

循环直到收敛{

初始化一个

E-step 对每一个i，计算



M-step 更新



}

接下来再看EM算法的收敛性，假如和是EM算法的第t次和t+1次迭代后的目标函数的最优解，假如我们证明了那么就有EM算法每循环一次就是似然函数非递减一次，最终达到最大似然的最大值。



下面通过分析EM算法的一次迭代来说明上式证明，从上一次的M步，固定，对求导得到最大值，并把最大值对应的解赋值给，所以是在该次下（固定）使得最大的解，即有也就是对应第二个不等式成立。在本次E步之前由Jessen不等式有：



在本次的E步下（根据样本分布期望最大化来更新），使得等式成立。也就是对应第一个不等式成立。

所以说EM算法E是借助样本期望最大化，使得为



常数函数，Jessen不等式成立，在M步最大似然估计求解参数。然后下一次E步在新的参数下期望最大化，使得



为常数函数，Jessen不等式成立，再M步似然估计求解参数，如此反复迭代，必然收敛。

最后，如果定义：



从前面的推导中我们知道 ℓ(θ)≥J(Q, )，EM 可以看作是 J的坐标上升法， 的坐标上升法， E步固定， 优化 Q，M步固定 步固定 Q优化。

下面再来从EM思想来看混合高斯模型，并给出混合高斯模型最大似然估计计算出参数的推导过程：对于混合高斯模型，我们的假设是隐藏的类别标签服从二项式或者伯努利分布，伯努利分布（样本中包含标签信息）对应高斯判别分析，混合高斯模型中假设类别服从二项式分布，而具体每一类的样本则独立服从高斯分布：



而在EM算法框架下

混合高斯模型E步中：



该式可以通过对样本的统计与混合高斯模型的假设参数求得。

在M步中，固定后，进行最大似然估计：



对于上的最大似然估计就像是带隐含类别标签的分类问题的最大似然估计，似然函数对求梯度有:



有



再来看，是假设类别服从二项式分布的参数，在不考虑条件概率的高斯分布，和，即固定，的条件下：最大似然函数最大化有：



考虑到约束条件，那么有其拉格朗日函数为（最大化目标函数）：



对其求导有：



有：



即



再由：



有：



所以最后有：



最后同样可以求得。

在总结EM思想之前，再介绍一种混合朴素贝叶斯方法，其可以用于文本聚类

与朴素贝叶斯相比，同混合高斯模型一样先对隐含类别标签进行假设，然后借助EM思想求解模型参数。

同样样本为，样本特征，表示词典中第j个单词是否在该样本邮件中出现。即服从伯努利分布，当然反过来表示一封邮件中第j个单词对应词典中单词的位置，那么就是服从二项式分布。同样可以借助EM思想进行聚类，只是模型比伯努利模型复杂。而对于假设聚类标签可以定义为服从伯努利分布，也可以定义为二项式分布。

采用EM算法则有：

E-step：



M-step：

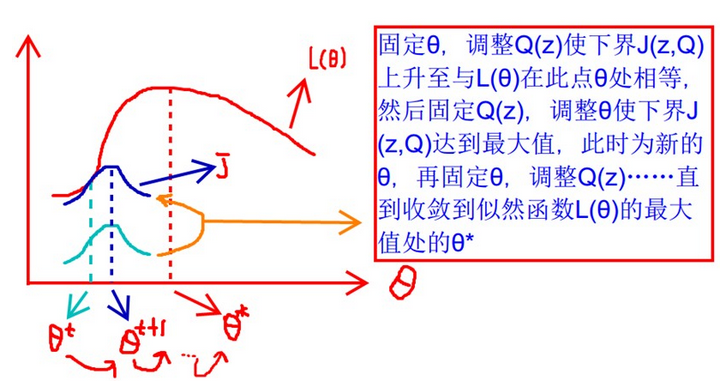


最后求解处参数有：



可以对比朴素贝叶斯分类器，最后参数的形式上非常相似。

总结EM算法

EM算法更像是一种思想，在EM思想下，常用的分类算法可以借助EM假设隐含类标签的方式，进行聚类，在EM算法中E步和M步中反复迭代使得目标函数达到最大值，其过程就是在E步期望最大化，使得假设函数为常数函数，这样目标函数为似然函数提供的下界等式成立，在M步固定类标签假设参数进行最大似然估计更新模型参数，再次根据更新后的参数结合样本分布期望最大化求。如此反复迭代使得目标函数达到最大值。**其中EM算法中E步是让 Jessen不等式取等号，取等式的方式有两种，在混合高斯模型与混合贝叶斯模型中使等式成立即基于样本统计和固定的参数期望最大化使等式成立。在因子分析中则通过映射（多维），直接求得后验概率****所以EM算法更为广义。**

因子分析：

同混合高斯模型一样，因子分析的是借助EM算法的思想，因子分析出现的目的是为了解决在某种情况下混合高斯模型不适用的情形。假设样本特征维度为n，样本数量为m，一般情况下 ，我们采用混合高斯模型，其最大似然估计的期望和方差为：



当或者时，发现其协方差矩阵为奇异阵，即，不可逆。所以问题就来了。

至于为何协方差矩阵为奇异阵，举个简单的例子，当样本数m与特征维数n都为2时，我们采用混合高斯模型，那么其高斯模型如下：

现在我们对进行如下约束，

假如我们没有足够的数据对参数进行估计，我们可以通过加限制来解决为奇异的问题。假设为对角阵（因子分析中实现该矩阵对角化的方式是找一个映射函数），那么显然对角线上元素即为所有样本在各特征维度的期望的二范数和，忽略特征之间的关系，即非主对角线上元素为0：



这样限制的话，必然会失去特征与特征之间的关系。

更强的约束，不仅为对角阵，且为对角线元素相等，即

这样一来，对其进行最大似然估计有：



下面在继续因子分析模型前，先来看一个高斯分布的边缘分布于条件分布。

假设有随机变量



假设，其中



由，协方差性质有

现在用x与来表示协方差矩阵



同样，



因为高斯分布的边缘分布也是高斯分布，由，同样

而且有条件概率分布

其中



关于上式，具体回头再看数学推导。

下面我们来看因子分析模型，在因子分析模型中，假设一个联合分布其中随机变量x和z满足下面分布：



其中：

与之等价的定义是：



也就是说在训练样本集的特征x外定义一个，**其中关键的处理是随机变量x与随机变量z的关系有。**

由之前给出的高斯分布特性，的联合分布为：



由，，可知



所以联合期望为：



再来看协方差矩阵，由，可知，，且同样



那么有的联合分布为：



所以最大似然估计为最大化：



以上为纯数学推导，其构造因子z满足，就是为了满足随机变量的参数-协方差矩阵可逆，对于其**直观理解**如下：

由前面高斯分布推导中有：即服从均值和方差的正态分布其中均值和方差为：



再来看其EM思想：

在E-step：



M-step 更新



同前面一样，上式也可以看做是再求



同样在固定z的分布之后，最大化似然函数等价于最大化最后一式。

展开求梯度有最后参数的结果：



具体不再讨论过程，这里再次分析一下EM算法，对于EM算法更广义上的理解应该是构造一个



其中满足随机变量概率分布函数的三条基本性质。对于其构造的方式有很多，比如混合高斯模型时通过贝叶斯估计转换为似然与先验乘积的形式，通过对样本的统计来求期望，而因子分子则是通过构造因子z满足，那么这里的就不需要通过贝叶斯公式转换，而直接通过



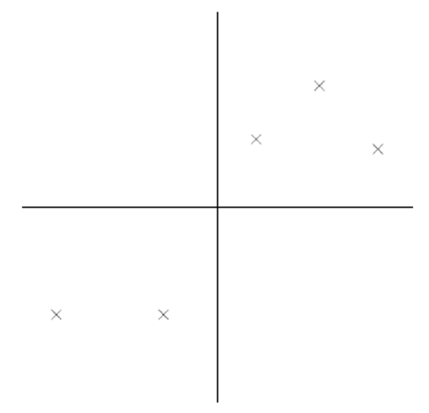
来求得，所以广义上讲EM算法在构造上还是通过，如果说x是之前的特征维度，那么就是联合特征或者是特征z。

主成分分析PCA

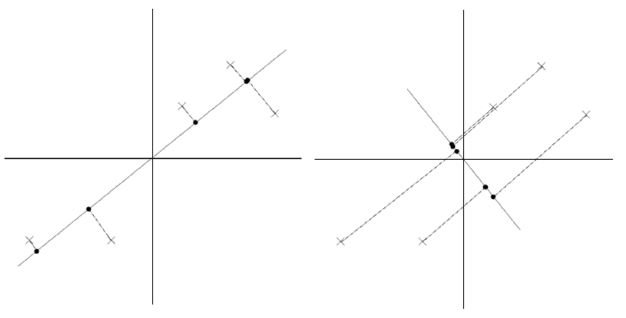
首先从问题出发，假设样本维度很高，我们无法直观的从高维的数据中看出样本的分布，其中可能有1厘米-米这样等价的属性同时存在，也可能有2学号-成绩这样无关联的属性同时存在，也可能有3学习兴趣与成绩这样关联性强的属性同时存在，还有可能有噪声属性存在，还有可能对于样本少特征多的情况。那么我们如何从高维特征属性中把数据给“解救”出来。

PCA理论基础。

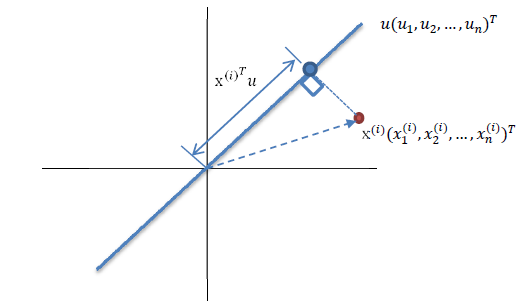
假设，我们有如下样本分布：



由于我们并没有类别标签，我们希望做的，能够做的就是希望找到一个方向，使样本点投影到该方向上的距离越大（从回归问题上讲，就是找到合适来使得的值的分布距离越大），那么对于上图的样本分布，我们可以观察下面两个投影方向后的样本分布



可以看出，要想找到使得投影后样本在该方向（一维）离散度越大，也就是找到一个方向使得点到该方向上的距离最小。



对于一个样本点在该方向上投影上距离可以表示为。

所以对于所有样本，我们希望其投影后的距离和的平方（忽略方向）和平均值最大，写成数学形式如下：



可以看出，假设x的期望为0，那么中间那个表达式就是x随机变量的协方差矩阵，用表示。

假设上式等于，则有：

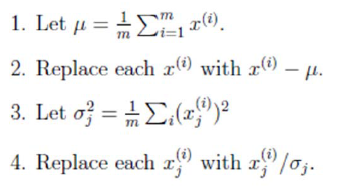
，由为单位向量，所以下式成立：



可以看出该协方差矩阵是由所有样本的特征向量构成，x的每一列就是一个样本，每一行就是所有样本的一维特征。

所以我们最大化目标函数的值则为协方差矩阵的特征值，而该特征值对应的特征向量就是投影方向，即确定目标函数。不过需要注意的是前提假设是均值为0，且特征属性在方差上进行归一化。

所以在使用PCA之前一般要对样本做预处理：



下面分析下预处理的作用，1-2步做的是对样本进行平移，使得样本的均值为0,3-4步做的是对于样本方差在度量上进行归一化，因为，对于采用米与厘米进行记录的样本，在这两个特征属性上做方差比较，不归一化必然是不合适的。

在进行完预处理之后，下面就可以对数据进行PCA分析，也就是求样本数据的协方差矩阵，然后对其进行求特征向量，当然有时一个特征向量并不能很好的反映样本的分布特性，可以选择较大的几个特征值对应的特征向量最为投影方向，并把样本投影到该方向上的值最为新的特征属性，而且用这些特征值属性来分析样本。写成如下表达式：



**可以看出PCA的思想并不是简单的从n维属性中挑选出k维主要的属性，PCA的思想更像是在高维的分布中找出一些投影方向使得高维的数据在这些方向上的距离（距离的概念由于均值归零化，那么距离的概念就等价于离散度最大化，分布不集中）最大，从PCA的目标可以看出PCA在数据降维上有很好的作用。**

下面再来看看PCA适合那些应用场景

1数据可视化：对于高维数据，无法直观的观察样本的分布，可以采用PCA方法提取2-3个主成分作为新的特征维度，观察样本的分布。

2压缩：当样本特征维度较高，可以采用PCA选择适当的主成分来描述样本，通过降维的方式实现数据的压缩。

3PCA减少过拟合：即当由于特征维度过高产生过拟合现象，同样可以通过降维来防止过拟合。

4匹配，距离计算：在对于两个本进行相似匹配时，可以不直接采用欧式距离，而是采用PCA投影后的点进行距离计算。当然不一定是样本匹配，对于特征距离计算也可以。

SVD奇异值分解

接着PCA分析继续数据降维，在面对求解高维的样本数据的特征协方差矩阵的特征向量，在求解时，其中



奇异值分解是：若，A有若干正奇异值，则存在m阶的酉矩阵U和n阶的酉矩阵V，满足：



那么有：

由上式可知U酉矩阵是满足酉相似对角矩阵，同理也可以推出V是酉相似对角矩阵。进一步可以推出：



所以有U是的所有特征特征向量构成，而且前r个较大的特征值对应的特征向量就是的投影方向。所以PCA中所求的投影方向就是SVD分解中U酉矩阵的前几个向量。结合PCA分析的话，采用SVD分解表示X矩阵，分解来的U酉矩阵就是其所有主成分的投影方向构成的矩阵，选择适当的投影方向对数据进行投影产生新的分布，而在新的样本分布上进行下一步分析。也就大大简便了计算，然后求解特征值分解过程。而对SVD分解本身来说，假如一个样本信息可以由矩阵A表示，而矩阵SVD分解，且把特征值从小到大排序后，选择合适的特征值数k就可以很好的表示的话，约等于的原因是假如把左边和右边做减法，则由于选择特征值最大的前几个，而U和V又是酉矩阵，所以特征值小的与与矩阵的乘积也小，所以左边和右边的差值也就小那么也就实现了对A的降维表示，也达到了降维的目的。

独立主成分分析ICA

前面提到的PCA方法，其目的是为了找出一个投影方向使得投影后样本的方差最大，但这并不一定表示该方向是最有利与分类的方向，仅仅是使得样本更离散化，当然PCA对不同样本分布效果也不一样，其也需要建立在一些假设条件之上。

独立主成分分析同样，需要假设主成分之间是独立的，且不能是高斯分布，下面先从一个具体的样例思考ICA。

假设在party中有n个人，他们可以同时说话，我们也在房间中一些角落里共放置了n个声音接收器（Microphone）用来记录声音。宴会过后，我们从n个麦克风中得到了一组数据\*x(𝑖)(𝑥1(𝑖),𝑥2(𝑖),…,𝑥𝑛(𝑖));𝑖=1,…,+，i表示采样的时间顺序，也就是说共得到了m组采样，每一组采样都是n维的。我们的目标是单单从这m组采样数据中分辨出每个人说话的信号。

那么我们得到的数据就是n个声音接收器记录下来的数据，我们用

，假设每时刻所有声音接收器记录的数据构成一个n维样本，那么我们一个样本的定义则为，m表示0-t时刻离散化记录了m个样本。也就是说我们的样本是一些n维的向量。同样，假设有n个或者个人在说话，而说话人声音是独立的，每个人的声音我们用，同由声音接收器一样每时刻所有人的声音构成一个维的样本， 那么原说话者构成一组样本。而原说话人构成的样本是不知道的，我们直接观察到的是。由生活经验可知，那么每个声音接收器应该是个人声音的组合，这里假设是线性组合，那么就有：



这里可以看出，n表示维度，m表示样本数，即样本的每一列都是某一时刻上的样本，每一行都是一个接收器在0-t时刻记录的一维数据。这么一来，有了上面的表达式，而我们知道的是观察样本X，我们目标是知道S，换一种思考方式，假如我们知道n个接收器是如何组合每个人的声音，即知道A的话，也就可以求解S。

这里假设，那么有。

下面针对上面样例来分析ICA的不确定性，

由于w和s都不确定，那么在没有先验知识的情况下，无法同时确定这两个相关参数。比如上面的公式s=wx。当w扩大两倍时，s只需要同时扩大两倍即可，等式仍然满足，因此无法得到唯一的s。同时如果将人的编号打乱，变成另外一个顺序，那么只需要调换A的列向量顺序即可同样可以组合出X，因此也无法单独确定s。这两种情况称为原信号不确定。另外，从矩阵角度来看，线性组合就是一个旋转，平移操作，假如原始的S是一个多元正态分布，那么不论如何线性组合，我们改变不了X的分布，这么一来我们就无法从X的分布中学习出原始原本的分布。从数学推到上来说，可以有下面解释：

在求解W之前，我们再来了解一下线性变换后的密度函数的变化。

前面我们有，现在假设随机变量S的概率密度函数为，那么X的概率密度函数呢？

从概率分布函数与概率密度函数出发，，



有了上面的公式，我们继续来看ICA算法：

还是从出发，我们观察得到的是X样本，不知道的有S，W，虽然目标是得到S，但是直接从S出发是不可取的，我们可以通过求解W后再来解出S。对于案例来说也就是学习出声音接收器接受多个人声音的一种组合。假如我们从先验经验出发，假设每一个n维的样本每一个维度都是独立同分布的，即在某一时刻每一个人的声音都是独立且同分布。而这个假设的分布很关键，选择不同的分布进行分析效果也不一样，这里假设使用分布函数为sigmoid函数，可以看出sigmoid函数也满足概率分布函数的特性，单调不减，值域[0-1]，这么一来，概率分布函数则为分布函数的导数：



有了某一时刻的每一个有概率密度为，由之间独立，那么s联合概率就可以写成如下形式：



根据线性变换后的概率密度公式，结合上面的假设服从sigmoid分布有：



最后，根据采样后的训练样本，采样最大似然估计有：



其中大括号里面的表示，需要注意的是是基于样本每一维是独立的，而最大似然估计是所有样本概率的成绩最大，所以大括号外的为相加形式，而这也需要建立在每个样本之间是独立的，即每一维度所有样本之间独立，但是这不符合实际情况的，因为一个人在0-t时间内声音数字化后一定是连续有关联的，但是当训练样本很大时，该条独立假设影响不大。

接下来就是对W求导，这里再给出对矩阵W的求导方法：



最终似然函数对W求导后的表达式如下：



需要注意的

再来分析假设意义：这里假设一个说话声音的数字化是服从sigmoid分布，那么一个样本一维分布，即一个声音接收器记录的一个维度就是



用图像表示即为，（像高斯分布，但又与高斯分布不一样）：

线性判别分析LDA

**增强学习**

在机器学习算法中，有一类增强学习算法，又称奖励学习、评价学习，是一种重要的机器学习方法，在智能控制机器人及分析预测等领域有许多应用。

假如要设计一个下象棋的AI，AI的最终目标就是赢棋，要赢棋显然需要AI下棋的每一步进行决策。而每一次决策都对最终结果起着累积作用。所以，如何进行每一步决策就是设计象棋AI的关键。一般的监督或者非监督学习算法大都是一次性的决定最终结果，无法刻画这样一个过程。增强学习就是为了实现这样一个学习过程的，每次决策的优劣用一个回报函数来刻画，直到最终结果结束该过程。马尔科夫决策过程一种常用的增强学习算法。

同样，机器行走也是一样，如何以一条最优的路径到达目标位置也是一个需要决策的过程，每一次决策都使得回报函数最大。

一个马尔科夫决策过程由一个五元组组成

例如如下问题：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | +1 |
|  | 无 |  | -1 |
|  |  |  |  |

则有：

S表示状态集，如上面的每一个方格表示所处的一个状态

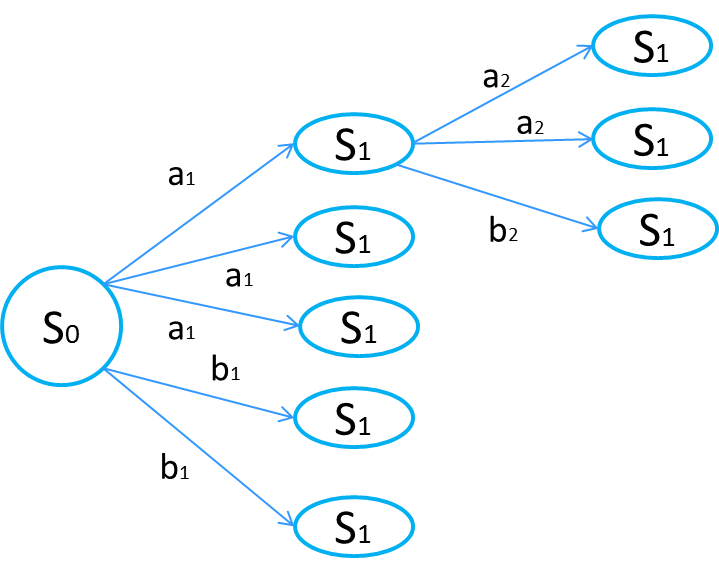
A表示一组动作，在上面的每个状态下都有四个动作上下左右{N,S,E,W}

表示在某一个状态下，采取某一个动作之后，到达的概率，即表示在某一状态下由决策函数采用动作之后可能由于实际情况，并不能以1的概率到达下一状态，而是以不同的概率到达几种状态。

阻尼系数，其作用是前一时刻与后一时刻到达同一状态得到的回报应该有所区别，即使得快速收敛。

R：R是回报函数，在不考虑动作的情况下，回报函数只与状态有关，而与其采取的动作无关。所以又常定义为，一般情况下，到了奖励状态，定义回报函数，到了惩罚状态，定义回报函数，而对于一般的状态定义的回报函数。

其中

 有了如上的定义，那么马尔科夫决策过程在某一个决策函数（决策集）下，就有如下过程：

该过程表示，从状态出发，由决策函数来定义在选择对应的动作，当采用动作后按的概率转移到下一个状态，。然后重复该步骤，在不同的状态下，根据决策函数一直走下去，直到结束。

根据回报函数的定义，那么统计整个过程的累积回报则为：



当回报函数与动作无关时：



可以看出，在t时刻回报值被打折，也就是说对于同样的回报值，由于越晚到达，所得到的回报衰减的越多，所以我们的决策函数必然要把回报值大的放到前面。

也就是说，我们的目标就是选择一组最佳的动作（由决策函数所确定），使得累积回报加权和期望最大。



在定义好马尔科夫决策过程五元组的基础上由决策函数来决定最后的累积回报的，上面的一系列动作就是在状态s下根据决策函数选择动作，也就是说决策函数是一个从状态集到动作集的映射函数。下面给出决策函数的定义：。

给定之后，动作。

在定义决策函数之后，那么对应一个决策函数的累积回报就可以表示为：



值得注意的是，上式是表示从出发，决策函数为下的累积回报，同样，若在下出发，决策函数同样为，其累积回报则是，所以必须牢记的是R是表示每一步的回报，而则是由从发状态和决策函数下的累积回报。可以看出，累积回报函数只与出发状态和决策函数有关。

直接最大化上面的表达式是不易的，从递推的角度来看，贝尔曼方程有：



拿下象棋来说，每一步决策除了要考虑当前局势s，下棋人往往会再脑子里构思后面几步的局势，这里刻画局势的就是回报函数，当前局势用立即回报R(s)表示，后面的局势用来刻画。

由于在某一状态下采用指定动作到下一状态是以一定的概率形式，所以写成最终形式。可以看出第一项是状态的立即回报，第二项可以看作是从下一状态出发累积回报的期望，其中下一状态服从分布。

有了上面的数学表达形式，我们的目标就很清楚了，就是要定义一个决策函数使得整个马尔科夫决策过程的累积回报值最大，值得注意的是，我们不仅是要求以为初始状态的累积回报最大，而是在该决策函数下，任意状态为初始状态整个累积回报都最大化，也就是用下式表示：



由贝尔曼方程，上式也可以转换为：



可以看出，最优决策累积回报值在初始状态已知的情况下，其回报值由最优决策所确定，也就是说根据最优决策可以确定最优回报（其实在初始状态已知的前提下，任意决策函数都可确定最终累积回报值）。至于在确定决策函数下，求解不同初始状态下的累积回报就是一个解线性方程组的问题，其方程个数与未知量个数等于状态数。（这样一种过程有点类似于马尔科夫链的极限分布求解过程）

定义了最优累积回报，再定义最优决策：:



有了最优决策与最优累积回报的定义，那么就有下式：



也就是说最优决策对应的累积回报一定大于一般的决策，而最优决策所产生的累积回报也一定是最优回报。

这里值得注意有几点：

1最优回报和最优决策一一对应，显然，决策可以确定累积回报，反过来累积回报是否能确定决策，也就是说等价性。答案是肯定的。

2在决策函数确定下，最优决策是否能保证在不同的初始状态下得到的累积回报都是最大的，这个值得思考，一般情况下决策函数最优出于全局考虑的，并不一定最优决策在每一个状态为初始态下决策累积回报都大于一般的决策。

下面举个最优决策与最优累积回报的例子来说明其一一对应性，且说明累积回报函数本身就是一个过程的累积回报，回报函数R才是每一步的回报。

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | +1 |
|  | 无 |  | -1 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 0.85 | 0.90 | 0.93 | +1 |
| 0.82 | 无 | 0.69 | -1 |
| 0.78 | 0.75 | 0.71 | 0.49 |

**其中就是以s为初始状态沿着决策函数走到结束状态的累积回报。**

下面再来看求解上述最优问题

值迭代

1将每一个初始状态为s的初始化为0

2循环直到收敛{

对于每一个初始状态s，对进行更新



}

可以看出，更新第一次所有的，也就是说都只看眼下的立即回报，然后由于奖励状态和惩罚状态的分布不同，由靠近奖励状态和惩罚状态的状态决策逐渐导向到初始状态的决策，这也就是累积回报不断更新的原因（动力）。但是值得思考的还是最终会不会收敛到最优累积回报（暂时不作讨论）。

内循环迭代的的处理方法有两种：

1同步迭代：即在一次循环过程中，累积回报不更新，而是计算完所有的累积回报之后，再统一更新。

2异步迭代，即在一次循环过程中，每计算完一个初始状态下累积回报就立即更新，不需要等到所有的累积回报都计算出来之后再更新。

可以看出两种迭代方式造成不同的原因是第二项，因为立即更新之后，再计算下一个初始状态下的累积回报与暂时不更新得到的累积回报肯定不一样，拿第一次更新为例，同步更新第一次，而异步更新则第一次内循环中，除了第一个更新的s会出现，剩下的都有，值得肯定的是异步迭代的收敛速度肯定是快于同步迭代。

策略迭代

值迭代是使累积回报值最优为目标进行迭代，而策略迭代是借助累积回报最优即策略最优的等价性，进行策略迭代。

1随机指定一个策略：。

2循环直到收敛{

a:令V：= 

b对于每一个状态s，对做更新



}

这里要说明的是a步是通过前面的贝尔曼方程，以解方程的形式求解出每一个状态下的累积回报：



在b步则是根据累积回报值，从新更新决策。

同样，收敛性也是值得探讨的，这里简单的思考一下，由于奖励状态和惩罚状态的分布，以及累积回报唯一确定决策函数，那么未达到最优决策，必然累积回报和决策函数处于不稳定的状态，而只有当到达最优决策时，才有



所以该过程就是在a步由决策函数确定累积回报，然后最大化累积回报来更新决策，如此反复，则有最优决策。

值迭代和策略迭代比较：可以看出策略迭代涉及从决策函数到累积回报的解线性方程组的步骤，值迭代则是反复的，所以策略迭代更适合处理少量状态的情况，一般10000以内还是可以接受的。

MDP中的参数估计

回过头来再来看前面的马尔科夫决策过程的定义是一个五元组，一般情况下，五元组应该是我们更加特定的问题建立马尔科夫决策模型时该确定的，并在此基础上来求解最优决策。所以在求解最优决策之前，我们还需更加实际问题建立马尔科夫模型，建模过程就是确定五元组的过程，其中我们仅考虑状态转移概率，那么也就是一个参数估计过程。（其他参数一般都好确定，或设定）

假设，在时间过程中，我们有下面的状态转移路径：



其中表示i步，第j条转移路径对应的状态，是状态下执行的动作，每一条转移路径中状态数都是有限的，在实际过程中，每一个状态转移路径都要进入终结状态。如果我们获得了很多上面的转移路径，那么我们就可以来估计参数



分子是在s状态下采取a动作都转移到s的次数，分母是在a状态下采取s动作的次数。为了避免的情况，同样采用拉普拉斯平滑，用来代替。也就是说当到达的状态是样本中为为到达过的状态，那么在该状态下的执行的动作达到下一状态的概率均分。上面的这种估计方法是从历史数据中进行统计的，同样该方法适合于在线更新。对于立即回报函数的估计，一般根据实际情况学习或者设定。

所以整个马尔科夫决策过程流程如下（以策略迭代为例）：

1 随机初始化策略：。

2循环直到收敛{

  a在样本上统计该策略下每个状态转移的次数，来估计和R

b使用估计到参数来更新对应决策函数下的累积回报V

c根据更新的累积回报V重新进行决策，即更新

}

整个流程就是在策略迭代的基础上，同时进行了参数估计。

总结：

MDP值得说明的地方有迭代算法，以及迭代算法收敛性的证明，关于迭代算法可以与动态规划问题进行对比。

K近邻算法kNN

介绍：

K近邻算法是一种简单易于理解的学习算法，其思想是采用一种相似性度量来选择出最相似的K个样本点，然后根据这K个样本的确定类别（数值）来确定新样本。K近邻可以用来做分类和回归，分类时采用K个样本点投票机制对新样本进行估计，回归是采用K个样本的回归值进行加权求平局值。一种极端思考是，当K=1时，即最近邻算法。当K等于样本数m时，可以说就是全匹配了，由K=m可知，K近邻算法是一种懒惰的学习算法，因为K近邻算法没有训练过程，到测试时只选取与测试样本最近的K个样本点来进行估计。这也就导致K近邻算法在测试时开销大。由此，可以看出K近邻算法中三个基本的要素：相似性度量，K值选取，决策规则。

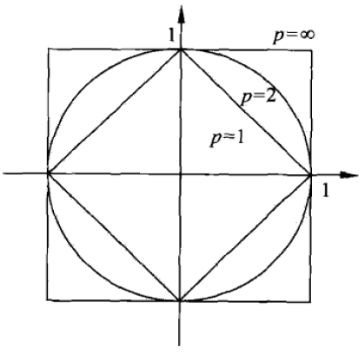
1相似性度量

由于K近邻算法没有什么对数据预处理的步骤（这里如果采用某种映射对样本进行特征空间映射，KNN可以在这个映射后的特征空间进行K近邻），所以最直接的相似度量就是距离的概念了，这里距离可以采用Lp距离。



其中，下标表示样本，上标表示特征属性。

可以看出，p=2，即为欧式距离，p=1即为曼哈顿距离，p=∞，即为个特征距离最大的一个。



由图，我们可以看出各种距离的适用范围，类似于损失函数的选择一样，选择哪种距离依赖于样本的分布。一般来讲（从概率角度），连续变量适合用欧式距离，离散变量适合曼哈顿距离或者极限距离。

2 K值得选择

K值得选择会对K近邻算法结果有重大影响，当K值选择较小时，那么就是用较小的邻域来预测新样本，而决策就会变得更依赖于样本的分布，预测结果对近邻实例比较敏感。即预测结果过分的拟合样本，易发生过拟合，即偏差虽然会减少，模型复杂度增加导致方差增大。当K值选取的较大时，即模型复杂度降低，那么偏差就会变大，方差会较小。在实际应用中，K值得选取通常采用交叉验证法来选取。

3决策规则

在确定了K个近邻之后，如何根据这K个样本来预测新样本就是最终的决策。由经验可知，K个近邻如果一视同仁是不合适，所以一般可以对每个近邻赋予一个权重。同样，考虑到样本类别均衡问题，对于每一类一视同仁也是不合适，一般会对样本少的类别赋予较大的权重。

4 K近邻算法实现步骤

1收集数据，根据数据的特性选择合适的距离度量

2 计算预测样本与每个标记样本的距离

3选出K个距离最小（相似性最大）的样本

4根据数据的分布，选择合适的决策规则

5根据决策规则预测测试样本

决策树

决策树是一种基本的分类和回归方法，其用于分类则主要借助每一个叶子节点对应一种决策属性的思想，而用于回归则是类似于局部线性回归一样，对线性回归函数进行分段拟合，而不同的叶子节点对应不同的分段函数。决策树算法中，主要的步骤有：特征选择，建树，剪枝。下面对三种典型的决策树进行分析。

**1.1特征选择**

对于决策树而言，每一个非叶子节点都是在进行一次属性的分裂，把不同属性值的样本划分到不同的子树当中，而对于分类属性的选择就是建立子树的关键。常见的选择方式有ID3的信息增益，C4.5的信息增益率，CART的最小均方差。

这里分别介绍这三种决策树的特征选择标准。

1. 信息增益

为了更清楚的理解信息增益，需要了解信息论中，信息，信息熵，以及条件熵的概念，，，熵则是随机变量的一种期望概念，信息熵越大，则表示随机变量的不确定性越高。所以在进行特征选择时当然是尽可能的选择分裂的属性使得分裂后的子集的信息熵和最小。也就是不分裂下的信息熵与分裂后的信息熵的差值最大，即信息增益最大。所以信息增益表示的就是在得知特征X的信息而使得类Y的信息的不确定性减少的程度。在决策树中，即为特征A对训练数据集D的信息增益为Gain（D,A），定义为集合D的信息熵H(D)与特征A给定条件下的D的条件熵H(D|A)之差。

设X是一个取有限值的离散型随机变量（对应决策属性的取值），其概率分布为



信息熵：



条件熵：



信息增益：



对于用信息增益来进行特征选择，的确可以选择出那些对于类别敏感的特征。但是需要注意的是，信息增益无法缓解某一特征尽可能多的取值的情况（即倾向于选择取值较多的特征），如对于数据表中主键这样的属性，用信息增益进行属性选择，那么必然会导致信息增益率最大，因为主键与样本是一一对应关系，而这样做分类是无意义的。即信息增益不考虑分裂后子集的数目问题。

（2）信息增益比

  针对信息增益偏向于选择特征值取值较多的特征，信息增益比则把分裂后的条件熵做为分母来限制这种倾向。即采用信息增益比上信息熵来定义信息增益比。



（3）最小均方差

最小均方差只适用于CART，是一种回归树的一种特征选择方式。在特征选择上分类树与回归树不同的是，回归树的特征选择是可以重复使用的，分类树在一个分支上只能一次。回归树在进行特征选择时不仅需要选择合适的特征还需要确定合适的阈值，一般来讲回归树是一个不断二分的过程。回归树在进行特征选择时会对每一个特征选择不同的阈值来测试其最小的最小均方差。最后会选择最小均方误差对应的特征和阈值。

最小均方差的定义



其中，i表示样本，j表示特征，s表示阈值。

**1.2建树**

ID3

1若D中所有样本属于同一类，则建立叶子节点T，并将该叶子节点的类标记为该类，返回上一次递归。

2若A为空集，即所有的属性都使用完了，则建立叶子节点T，并把剩余子集中最多一类标记为该叶子节点的类别，返回上一次递归。

3否则，进行特征选择，选择信息增益最大，信息增益比最大，最小均方误差的进行建树（递归）。

4如果选择出来特征，其最大的信息增益（）小于预设的最小信息增益或者是子树中样本数小于预设的最小数时，同样建立叶子节点T，并把剩余子集中最多一类标记为该叶子节点的类别，返回上一次递归。

5如果选择出来的特征符合要求，则可以进行分裂递归建立子树，这里可以将子树根据样本数的多少从左往右排序，并依次递归进行建树，直到返回T。

6对第i个子树再递归建树之前，需要把刚分裂用过的特征排除在接下来能继续使用的特征集之外。

C4.5

同ID3

CART

由于树回归不存在剩余样本属于一类或者特征用完的情况，所以CART没有前面ID3的前面两步。

1 对每一个特征进行循环选择合适的阈值求其最小的均方误差，并把所有特征可能的阈值下的最小均方误差记录作为当前的最小均方误差，与之对应的特征和阈值作为当前分裂的特征和阈值。

2如果当前的最小均方误差小于预设的最小误差，或者分裂后的子集存在小于预设的最小值则进行建立叶子节点T，对于整个集合的样本进行线性回归建立回归模型。返回上一次递归。

3对于分裂后的左右子树递归建树，直到返回。

4生成CART决策树。

**1.3剪枝**

首先对样本集进行划分，分为训练集和验证集，为了防止过拟合，以至于把训练集自身的一些特点当作所有数据都具有的一般性质而导致过拟合。

**1.3.1预剪枝**

预剪枝是在构造决策树的过程中，在进行完属性划分后，首先根据训练集样本对划分后的子集Dv类别进行生成树，然后在验证集上进行验证该划分是否提高了预测精度，否则进行剪枝。其过程是一个贪心过程，即每一次划分都必须使得决策精度有所提高。当然也可以对建树进行约束，比如信息增益小于一定值的情况与建树之后其中一个子集的样本数小于一定数量则禁止建树来达到预剪枝的效果。

**1.3.2后剪枝**

后剪枝是在决策树建立后以后，自底向上的对决策树在验证集上对每一个非叶子节点判断剪枝前和剪枝后的验证精度，若剪枝后对验证精度有所提高，则进行剪枝。（不同预剪枝的是，预剪枝是对划分前后的精度进行比较，而后剪枝是对剪枝前和剪枝后的验证精度进行比较），相对于预剪枝，后剪枝决策树的欠拟合风险小，泛化能力优于预剪枝，但时间开销大。

**1.4连续值和缺损值**

连续值

决策树是基于离散属性生成的，对于连续值属性需要离散化才能生成决策树的分枝，最简单的划分方法是二分法，即选取一个阈值对样本的连续型属性值进行0,1划分，而关键就是阈值的选取，该阈值的选取采用下面函数表达式来刻画：



即最大化信息增益，同样对于连续型属性，在进行属性选择度量时，选择该信息增益。

缺损值

对于缺损值得处理我们有两个问题需要解决，一是在进行计算属性选择度量时，此时在当前样本集D下定义一个D1，D1是去除缺损值样本后的D的一个子集，在该子集上计算信息增益。即把上面定义计算信息增益的表达式中的D换成D1。二是属性选择之后，对于缺损值的样本该如何划分到由属性A划分后的子集Dv中，此时的处理是把其同时划分到每个子集Dv中。

决策树比较

树回归与ID3，C4.5相同之处在于都是决策树的一种，其思想是结合了决策树和线性回归。在ID3中属性一般是离散型（当然连续型可以离散化处理，但是这是极其无奈的一种方法）。而在树回归中属性则是连续性的，而且在选择分裂属性时，需要考虑的不仅仅是选择合适的属性，更包括选择合适的属性的一个合适的分裂阈值。而且，一个属性可以进行多次分裂（这也缓和了ID3中决策过快操作）。之所以说树回归与线性回归类似，是树回归的结果上其实是把线性回归进行了分段处理，即在一定的范围内使用不同的线性方程来拟合数据，这与局部线性回归很是类似，但是值得一提的是，凡是对连续性属性进行分段必然是不合理的。这与现实中的平滑经验一致。在局部线性加权回归中，。

同样，树回归的过程与ID3一致，选择合适的分裂属性和阈值进行分裂，选择的标准常用的有最小均方差。对于叶子节点的建立不再是决策属性了，而一般是线性回归模型。剪枝的操作同样有预剪枝和后剪枝，剪枝的标准与ID3一致。

提升算法

提升学习算法的理论基础源于，kearns和valiant首次提出了“强可学习”和弱可学习的概念中指出：在概率近似正确（PAC）学习框架中，一个概念（一个类），如果存在一个多项式学习算法能够学习它，并且正确率很高，那么就称这个概念示强可学习的。一个概念，如果存在一个多项式学习算法能够学习它，学习的正确率仅比随机猜测略好，那么就称这个概念示弱可学习的。而且Schapire证明强可学习与弱可学习是等价的，也就是说，在PAC学习框架下，一个概念是强可学习的必要条件是弱可学习的。

对于提升方法来说，两个关键点是：每一轮结束之后，如何来更新数据的权重或者说概率分布，二是弱分类器如何组合成强分类器。这里主要总结下Adaboost算法。而且Adaboost巧妙的使用了上面两个关键点。

Adaboost算法设计流程

1初始化训练样本的权重分布为



2 对m=1,2，M个弱分类器进行如下训练

1. 使用当前权重分布的样本进行训练得到最好的第m个弱分类器



1. 计算G（x）上的带权分类误差率



1. 计算当前弱分类的投票系数



1. 更新训练样本的权重分布



其中，Z为归一化因子



3构建基本分类器的线性组合



这里值得说是样本权重的更新，而且在样本权重的更新还涉及到了弱分类器的投票权重的确定。这两点是实现Adaboost的关键，关于它的介绍，可以来看看Adaboost的误差分析。



由于当时，则有，因此。由此可以推出不等式成立。

对于等式的成立，需要用到



到这里可以解释为什么权重的更新使用指数的形式，即采用指数的形式可以产生连乘的等式，而且，假如Z小于1，那么必然有限个Z相乘最后必然会使得误差不断减少。



这里怎么把Z和e联系起来，需要用到



而可以证明的是，所以有上式的不等式成立。到这里也就说明了为什么弱分类器的投票权重为什么采用了，也是为了不等式的延续。

至于为何权重需要归一化，而弱分类器不需要归一化问题。其实权重的归一化主要是因为如果不归一化的话那么求分类误差的时候分母也就不再是N个样本了，而是所有权重之和。所以假如权重不归一化的话，在计算分类器误差时在归一化也是可以的，但不方便。对于投票权重则没有归一化的要求，无论归不归一化，最后永远是权重大的弱分类器的投票权越大。

关于提升方法的理论部分大致如上，提升方法的关键点在权重和弱分类器的组合。而理论基础就是弱分类（学习的正确率仅比随机猜测略好）的组合可以得到强分类器。根据这个理论，提升树就是一个很好的应用例子。在提升树中，每一颗树只有一个分裂属性，即一次分裂，但是可以实现的就是总能找到合适的分类到达弱分类器的要求。而对于多个弱分类器的组合就可以实现提升方法了。提升树的设计过程和Adaboost的设计思想一致，每一个弱分类的实现则类似于决策树的实现，只是建树过程只分裂一次。

HMM

1隐马尔科夫模型定义

隐马尔科夫模型是关于时序的概率模型，描述由一个隐藏的马尔科夫链随机生成的不可观察的状态序列，再由每一个状态随机产生观察值构成一个可观测的随机序列。其中关键是状态序列是满足马尔科夫性质的，且可观测序列是由隐藏的状态序列随机生成的，即观测值由状态确定。

由此，隐马尔科夫模型由初始概率，转移概率，以及发射概率分布可确定。

形式化定义为：

所有可能的隐藏状态Q，所有可能的观察值V



其中N是可能的状态数，M是可能的观察数。

假设I是长度为T的状态序列，O是其对应的观测值序列。



A是状态转移概率矩阵



其中



表示在t时刻处于i状态到t+1时刻转移到j状态的概率。

B是发射概率矩阵



其中



表示在t时刻处于j状态发射出Vk观测值的概率。

是初始状态的概率分布



其中



表示在t=1时刻处于qi状态的概率。



由此，整个隐马尔科夫模型就定义完了，那么第一次接触隐马尔科夫模型的时候就会奇怪，为什么这样定义？

明显，如此定义是因为这样的模型有其应用的场景，而且应用场景还非常广泛，如语音识别，自然语言处理，生物信息，模式识别。下面以自然语言处理为例。

首先，在很多场景中，我们观察到的序列是有前后（上下文）关系的，但是这个关系比不是由观测序列决定的，而是由不可观测的隐状态关系决定（且该隐状态序列构成马尔科夫链）。这种场景下，为了很好的表示这种上下文关系，我们可以通过定义一个隐状态的转移概率矩阵来表示这种关系，用发射概率矩阵来表示隐状态到观测值的关系。

在自然语言中文分词中，由于自然语言是由明显的上下文关系的，即当前字与其前后出现的字都是有关系的。为了表示前一个字对当前字的影响，我们有一个隐状态来表示前一个字，用在前一个状态下转移到发射出当前字的隐状态的概率表示前一个字对当前字的影响。整个来说就是把上下文字对字的影响转化成状态对状态的影响。而用发射概率来表示状态到字的关系。

有了隐马尔科夫模型应用场景，那么接下来就要看用隐马尔科夫模型来做什么？

1 给定一个确定的隐马尔科夫模型（参数确定）和观察序列O，计算在该参数下观察序列的输出概率。

概率计算，由于观测序列的产生于隐状态是相关的，所以需要从隐状态的转移概率入手，通过发射概率间接的转化到观察序列。一般情况下该观测序列对应的隐状态序列有多个，把所有隐状态可能的序列结合观察序列求概率，再求和。

2 学习问题，已知观察序列O，估计模型参数，使得在该模型下观测序列的概率最大。

学习问题，假设在不知道模型参数的情况下，而我们有大量的观察序列，那么这些大量的观察序列一定不是偶然是这样，而不是那样的。从概率的角度来讲，是这样，而不是那样的原因就是，是这样的概率大于是那样的概率。如果有大量的观察序列，那么其中必然隐藏了模型的信息。

3预测问题，已知模型的参数和观察序列O，求解一条使得该观测序列概率最大的隐状态序列。这样概率计算类似，只需要求最大的即可。

好了，对应上面的三个问题，分别有三个算法求解对应的问题。

1概率计算-前向后向算法

2 参数学习-最大似然估计（有监督），Baum-Walch（无监督）

3 预测-Viterbi算法

最大熵模型

最大熵原理是概率模型学习中的一个准则。最大熵原理认为，学习概率模型时，在所有可能的概率模型分布中，熵最大的模型是最好的模型。

熵的定义有，熵，条件熵。

熵最大的情况对应的即是均匀分布的情况。