**Regression**

1. 假设函数
2. Mean Squared Error 均方误差
3. Gradient Descent 梯度下降
4. 几种梯度下降

Batch gradient descent：批量梯度下降

对于θ的更新操作，进行n次

Stochastic Gradient Descent：随机梯度下降

仅仅对随机的某一个进行梯度下降，而不是全部

区别：

批量梯度下降权值的更新需要对每一步多个样例进行求和，需要更多计算

当目标函数是凸函数的时候，BGD一定能达到全局最优，SGD准确度下降，即使目标函数为强凸函数时，仍然无法做到线性收敛

1. Minimizing Squared Error：除了梯度下降的另一种方法（矩阵）

令

1. 均值和方差的估计
   1. 均值mean

根据每个xi分组，每个xi出现fi次

,

* 1. 方差 variance

根据xi分组

1. Covariance and Correlation协方差和相关性

correlation

相关系数介于-1到1之间，接近1时表示正相关，接近0表示没有相关关系，接近-1表示负相关

相关性是快速检验两个变量之间是否存在线性相关的快速方法

对于一元线性回归：

容易受残差（测量错误的点）的影响

1. 多元线性回归：拟合多次函数

将 看作x1x2，进行一元线性回归

1. regularization 正则化：解决过拟合问题

1. 模型选择：采用所有的特征有时会使得模型变得很复杂

解决方法：

* 1. 子集选择：简而言之就是去掉对模型提升不大的特征
  2. 通过优化使系数收缩或正则化为0
  3. 通过将点投影到低维空间来进行降维

1. Local (nearest-neighbor) regression

最近邻：对于xq，首先找到它的最近邻xn，然后yq=f(xn)

k近邻：

距离函数：

1. Bias and Variance 偏差和方差

高偏差意味着欠拟合，高方差意味着过拟合

如果模型过于简单，参数很少，那么它有可能会有高偏差和低方差（欠拟合）

如果模型过于复杂，参数很多，那么它有可能会有低偏差和高方差（过拟合）

**Classification**

1. 全概率公式 条件概率P(A|B) B发生的条件下A发生的概率

如果，x就属于y=0，否则x属于y=1

1. 线性分类

用一条直线进行划分，直线上面的点一类，直线下面的点另一类

1. 交叉验证

handout method：在所有数据中选择一部分做测试集，剩下的为训练集

留一法：在所有数据中每次循环留下一个数据做测试集

k折交叉验证：将所有数据分为k组，每次循环取一组数据作为测试集

1. 数据类型

第一种分类：数值型和其他

第二种分类

irrelevant：字符串或者数字，与结果无关（参赛者的名字或代码）

nominal：不同类别之间没有数字关系的离散值（动物类型，颜色，国籍）

binary：只有两种可能性的离散数据

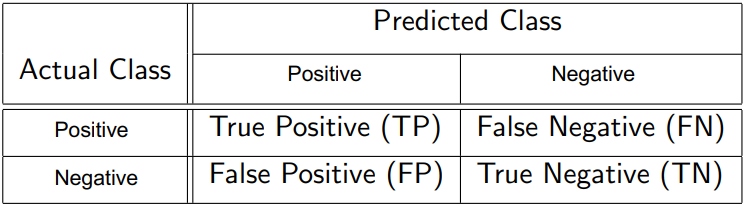
ordinal：可以排序的离散整数（学生排名）

count：计数

time：周期性的，重复的，连续的数据形式（时间）

interval：可以测量他们之间距离的数据（温度、收入）

1. Contingency table



1. 分类准确性
2. 其他评价指标

F1 score：

AUC-ROC： 当AUC->1时模型分类情况最好

1. 缺失值

填补缺失值的方法：

1. 删除缺失值的样本

缺失值很多的时候不适用

1. 用平均值、中位数等替换缺失的值

增加了模型的bias（减少了变量的方差）

相比其他方法效果不佳

1. 指定唯一的类别

添加一个类别可能会使模型出现较大的差异

添加最常见的类别可能会导致模型偏差

1. 预测缺失值

同样也会出现偏差

1. 使用支持缺失值的算法

其中一些算法非常耗时，在需要提取大型数据库的数据挖掘中可能至关重要

1. Nearest Neighbor最近邻算法
2. Distance距离

Minkowski Distance明可夫斯基距离

Euclidean Distance 欧式距离

Manhattan Distance 曼哈顿距离

如果p趋近于∞

如果p为0

1. mean and distance均值和距离

在欧几里得空间中，一组数据点D的算术平均值是唯一能使到这些数据点的欧几里得距离的平方和最小化的点

最小化一组给定点的欧式距离平方和和最小化平均欧氏距离的平方和是一样的

1. Centroids and medoids质心和中间点

质心有可能在点集之外，中间点必须在点集上

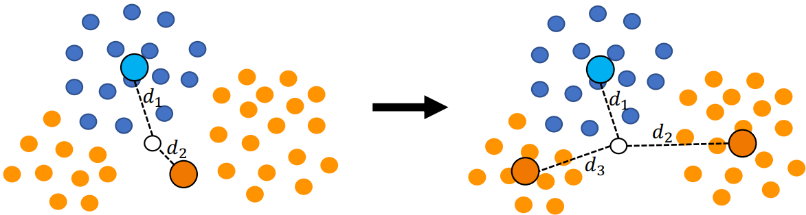
1. 基于距离的模型

有两点：第一是距离指标，即使用上述哪种距离公式，第二是使用质心还是中间点

1. Nearest Centroid Classifier最近质心分类器

类的质心公式：

如果一个测试集向量与质心i距离最近，那么他就被分类为质心i的类



类似于这样的很难被分类，除非在两边同时都有质心

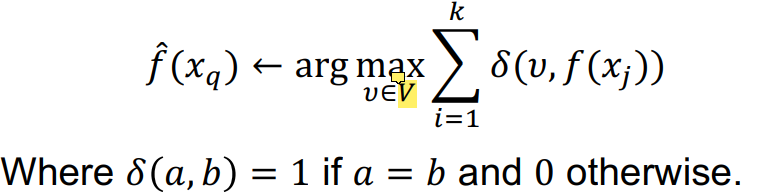
好处：简单，快速，在类紧凑并且相距较远时效果很好

坏处：不能处理复杂的分类，不能很好地处理异常值和噪声数据，无法处理丢失的数据

1. Nearest neighbor classification 最近邻分类器

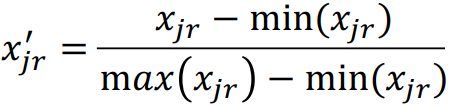
最近邻：对于向量xq，找到离他最近的训练集数据xn，将xn的分类赋值给它

k近邻：对于向量xq，找到k个离他最近的训练集数据进行投票，f(xq)的值为得票最多的那个分类标签

其中V是类别集合

Normalization

不同的特征有不同的范围，我们需要将他们进行normalization

 这是某一列（r）特征

什么时候考虑使用kNN算法？

少于20个属性，训练数据非常多，不需要学习“解释型”模型

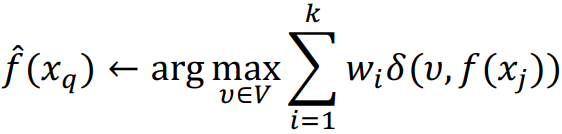
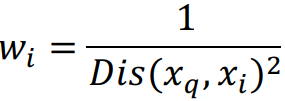
好处：训练速度快、能够学习复杂的目标函数、非常准确

坏处：查询较慢，最基本的算法要扫描整个训练集、假设所有属性同样重要，会被不相关属性所干扰（加权重）、噪音（移除）、需要相同的特征类型和规模

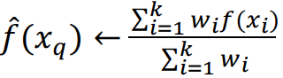
随着k的不断增加，bias减少，variance增加，

1. Distance-Weighted KNN

使用距离构造权值

对于实值数据



1. Curse of Dimensionality维度诅咒

随着维度的增加，距离的有效性下降

多项式更难估计许多参数

数据可视化变得更加困难

训练模型需要大量的数据

会导致过度拟合

解决方法：交叉验证、特征选择

1. 处理噪声

使用交叉验证

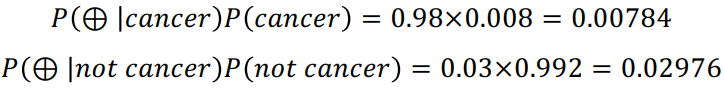
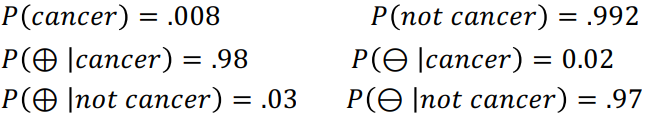
丢弃表现不佳的实例

1. 贝叶斯理论

最大后验假设

最大似然假设

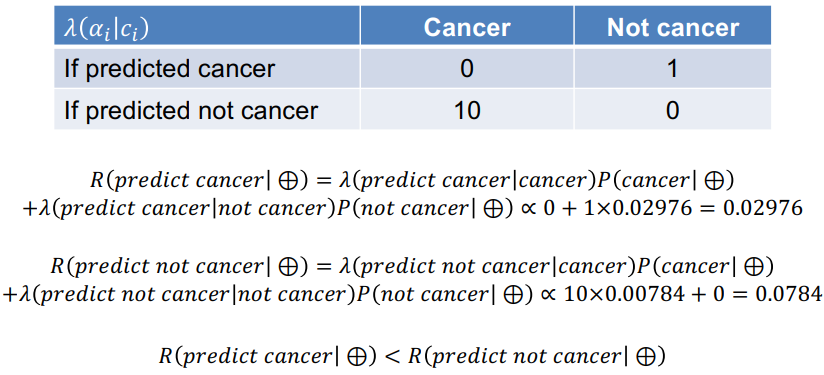
最大后验假设例子：



因此预测为not cancer

1. Bayesian Expected Loss 贝叶斯预期损失

例子：



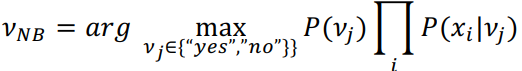
1. 贝叶斯最优分类器

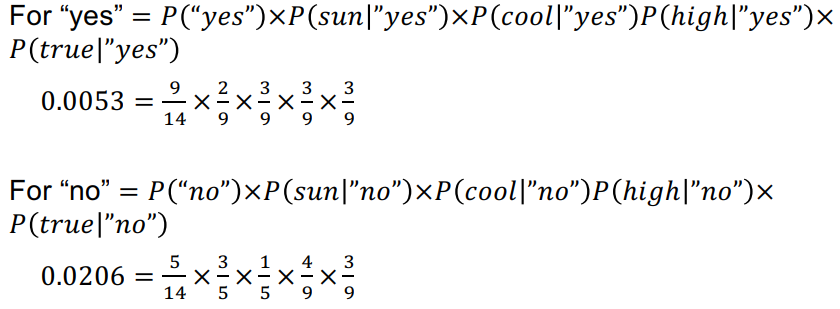
其中是a) 中的λ，我们要最小化刚刚的贝叶斯预期损失

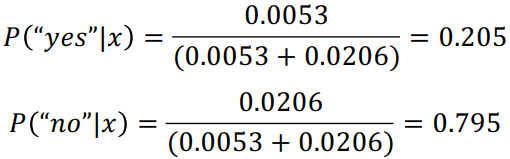
1. Bayes Error 贝叶斯错误
2. Naïve Bayes Classifier朴素贝叶斯分类器

例子：



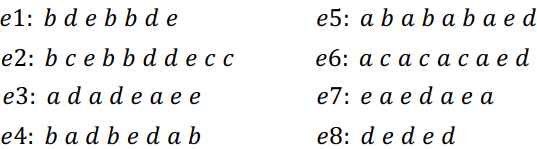






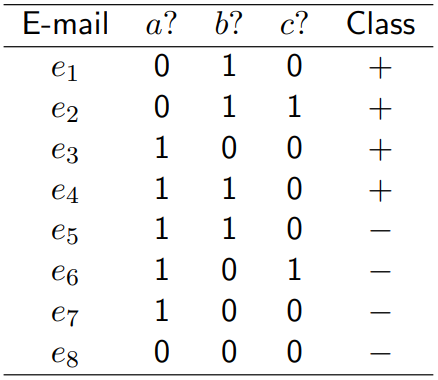
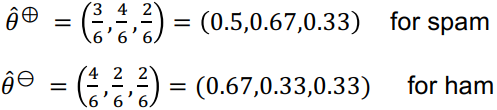
1. Text文本的分类

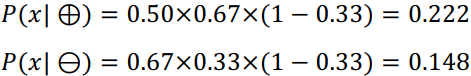
使用计数向量描述电子邮件数据集例子：



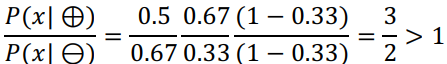
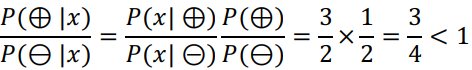
第一种方法：

在原先的基础上增加一个abc都有的和abc都没有的向量

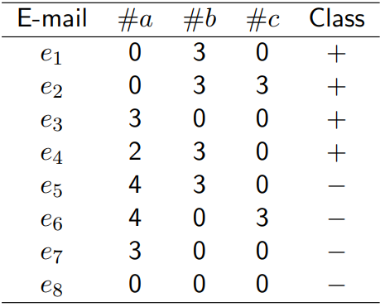
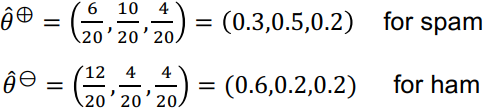
 

 ab是正的c是负的，系数(1,1,0)

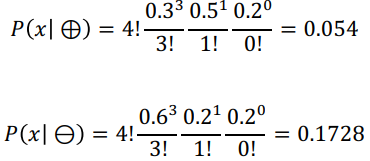
正和负的比率是0.33：0.67

第二种方法：

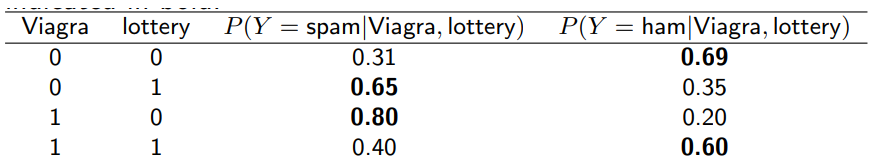
 

在原先的基础上增加一个abc都有的向量

比率是5/16所以选ham

1. Missing Value缺失值



对于这个例子，如果我们确定看清楚了email里面的lottery但是没有看清楚viagra

所以我们可以用这个公式

例子： 其中P(v=0)=0.90, P(V=1)=0.10

1. 贝叶斯理论的好处和坏处

好处：简单快捷，在多类预测中也有较好效果

当独立性假设成立时，朴素贝叶斯比其他模型表现更好

与数值变量相比，分类变量效果更好，对于数值变量，假设为正态分布

坏处：零出现很频繁的时候无法进行预测

贝叶斯分类器也被认为是很糟糕的分类器，输出概率不能被太认真地对待

强烈的独立性假设，现实生活中几乎不可能

1. Logistic Regression逻辑回归

优点和缺点：

优点：相对容易实现

易于解读

训练速度快，测试快

可以轻松拓展到多个类

提供概率预测

缺点：高维数据容易过拟合

它提供了线性决策边界，对于非线性决策边界，需要进行特征变换

对异常值敏感

要求输入变量之间中度或无相关性，可能导致模型不佳（降维）

**Tree Learning**

1. Decision Tree决策树
2. 什么时候考虑使用决策树？

由数字特征和离散属性值混合描述的实例

目标函数是离散值（否则使用回归树）

可能有噪声的数据

可解释性是一个优势

例子：设备或医疗诊断、信用风险分析、偏好

1. 决策树的多种算法

ID3、CaRT

接下来说的都是ID3算法

1. Entropy熵-一种检查纯度的方法

当存在多个类别时

熵的值越接近1表示越不纯，越接近0表示越纯

1. Information Gain信息增益

其中S代表的是父亲结点，Sv代表的是儿子结点

选择信息增益高的属性

局限性：信息增益更倾向于具有大量类别的数据，这样子集更有可能是纯的，这可能导致过拟合

Gain ratio：

1. overfitting 决策树的过拟合

对于假设，如果存在假设满足并且，那么假设h就是过拟合的

解决方法 – 剪枝pruning：

前剪枝：数据分裂时停止增长

后剪枝：长出完整的树，然后去掉过拟合的子树

前剪枝可能导致早停，阻止树木生长，经典问题：XOR，但是XOR类问题并不常见，前剪枝速度比后剪枝更快

后剪枝：验证剪枝前后验证集的精确度，提高剪枝，不提高不剪枝

连续值属性：n个属性分成n-1次，每次分解

attribute with cost：

误分类代价：

1. 好处和坏处

好处：轻松处理无关属性，能处理分类和数字数据，可以处理丢失数据，非常紧凑，测试速度快

坏处：只对数据进行轴向切分，倾向于过度拟合、greedy、以指数形式存在很多可能的树

1. Regression Tree回归树 – CaRT算法

运用树模型做回归问题，叶子结点不再是分类而是预测值（数值）

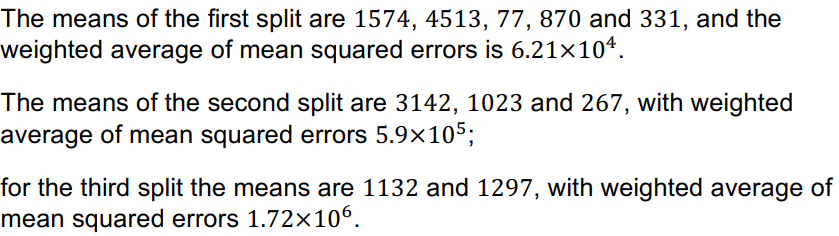
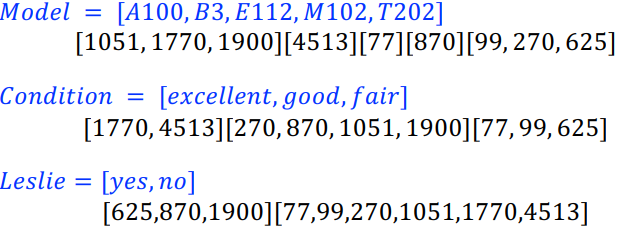
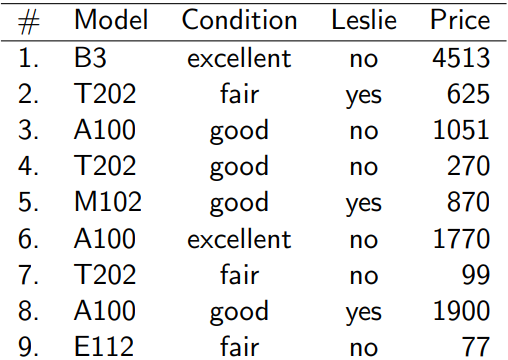
1. 原理

对于划分好的每一个子集计算它的MSE

计算方差加权平均数

遍历每一个特征，然后对于这个特征找到使得weight average variance最小的一个切分点

例子：



所以按照model分割

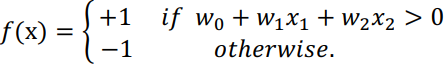
1. 如何避免过拟合

限制树的最大深度和叶子的最大数量

**Kernel Methods**

1. Perceptron感知机：是一种使用线性预测函数的二进制分类算法

如果我们有两个x1和x2的特征，那么我们可以用以下方法预测目标函数：



1. 损失函数

输入空间任意一点x0到超平面的距离

对于误分类的点，

因此误分类点到超平面的距离为

因此所有误分类点到超平面的距离总和为

不考虑，就得到了损失函数

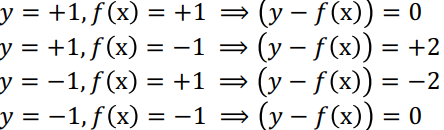
1. 感知机算法的原始形式

最小化损失函数

采用随机梯度下降法

随机选取一个误分类点，对w，b进行更新

1. 感知机算法
2. 将wi的初值都设置为0
3. 遍历训练数据，对每个训练样本进行分类
4. 更新 其中有四种情况



即

1. 重复步骤2几次，直到没有误分类点
2. 感知机算法的对偶形式 dual

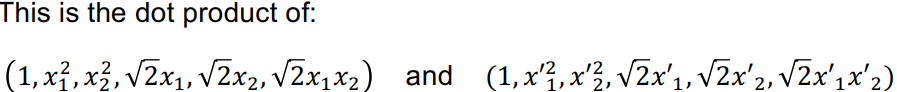
将w和b表示为xi和yi线性组合的形式

感知机模型

1. α=0，b=0
2. 在训练集选取
3. 如果

1. 重复2直到没有误分类点
2. Kernel Trick核技巧

证明函数是否为合理的核，其中，

所以它是合理的核

其中K被称为核函数或者核

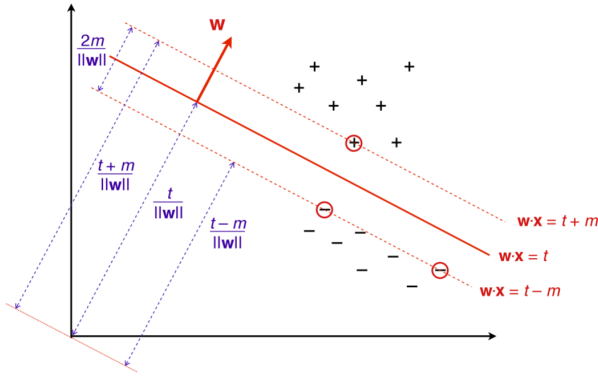
一些有用的核函数：

多项式核：

RBF核（高斯核）：

所以我们就可以使用这些核函数来代替上述感知机模型中的

1. Support Vector Machine 支持向量机SVM



在样本空间中，划分超平面可通过线性方程描述

样本空间的任意点到超平面的距离为

图中画圆圈的点就是支持向量，两个支持向量到超平面的距离之和为

所以要想找到拥有最大间隔的超平面，就要满足下式：

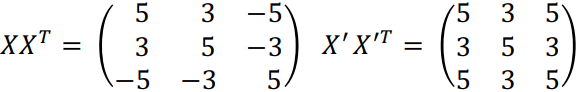
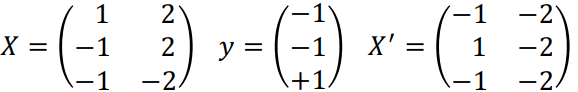
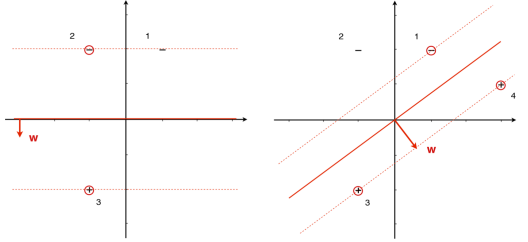
即：

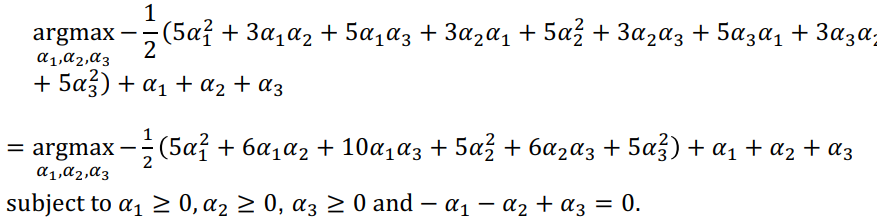
所以该问题的拉格朗日函数为

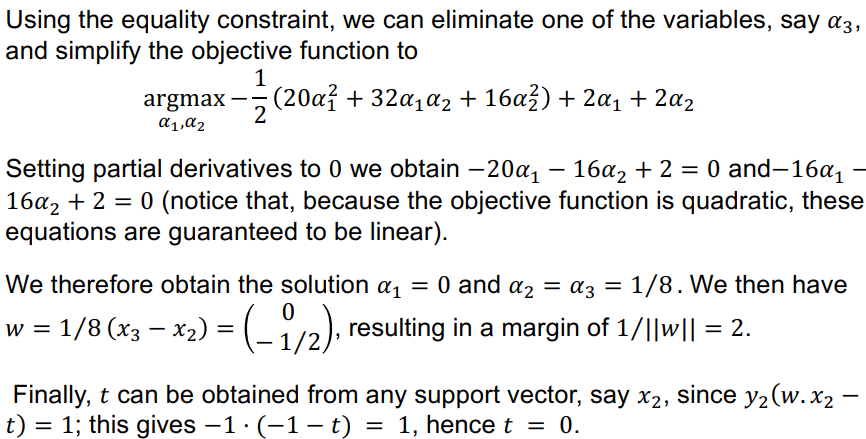
对w和b求偏导得到

带入L得到

例题：







软间隔 soft-margin

硬间隔要求间隔之间不存在任何点，这就使得硬间隔SVM对异常值非常敏感，噪声可能会导致误分类

而软间隔svm允许部分点分布在间隔内部

软间隔表达式：

位于间隔内部的点，我们引入一个松弛变量，对于间隔内部的点，满足

其中，C越大，位于间隔内的点越少

**Ensemble Learning**

1. 偏差与方差

偏差描述的是算法的平均值和真实值的关系，即算法的拟合能力如何

方差描述的是同一个算法在不同数据集上的预测值和所有数据集上的平均预测值之间的关系，即算法的稳定性如何

* 1. kNN算法

当k=1时，模型拟合能力比较强，当包含噪声样本时，容易过拟合，即低偏差高方差

当k值增大时，偏差上升，方差下降

当k=n时，无论输入实例是什么，都将简单地预测它属于训练实例中最多的类

一般选择，使用交叉验证确定k的值

* 1. bagging and boosting

bagging对应了降低模型的方差

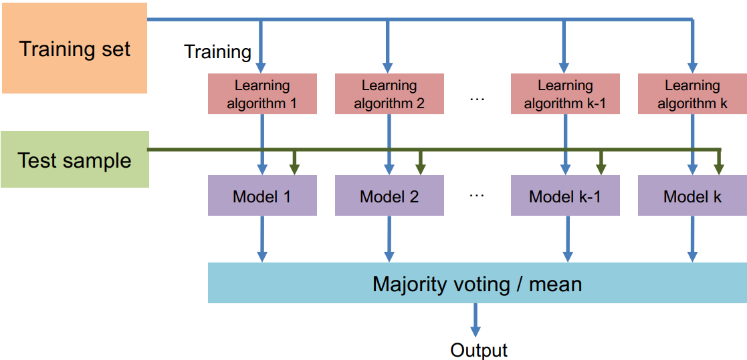
boosting对应了降低模型的偏差

1. Ensemble Methods 集成方法

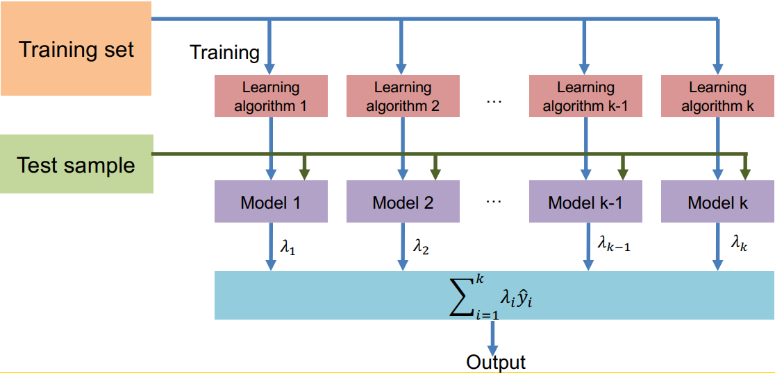
集成方法是将不同模型组合成一个模型的元算法，他们可以降低方差、偏差，提高性能

* 1. 简单集成

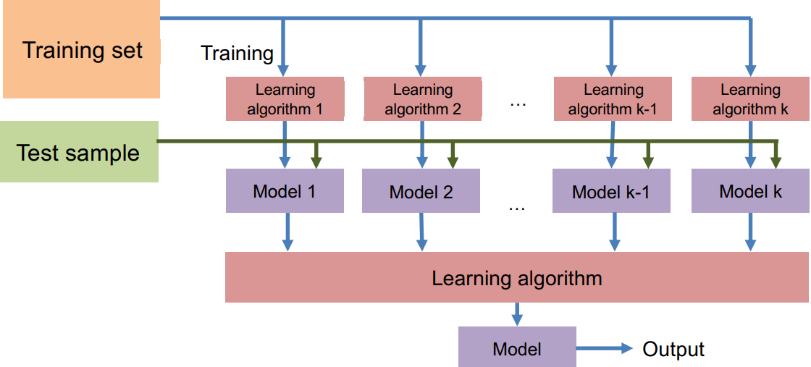
未加权的投票法或平均法



加权投票法或平均法



将每个模型的输出当作模型的输入



* 1. bagging methods – 用于降低方差，应用于高方差低偏差模型

它基于自助抽样法：集合D中有m个元素，每次从集合D中每次抽取一个放到D’中，再放回D，重复m次。这样操作之后D中约有63.2%的样本出现在采样集中。D’作为训练集，{D-D’}作为测试集

我们使用自助抽样法可以采样出T个含有m个训练样本的采样集，基于每个采样集训练出一个基学习器，再将这些学习器结合

bagging常对分类任务采用简单投票法，对回归任务采用简单平均法

优点：减少过拟合（降低方差），更好的推广

缺点：稳定模型不受bagging的影响，降低了可解释性，对于大量的数据，我们通常学习相同的分类器，所以平均他们没有帮助

* 1. 随机森林 Random Forest – 决策树bagging的一种变体

随机森林在以决策树为基学习器构建bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。即修改了决策树的划分属性的算法

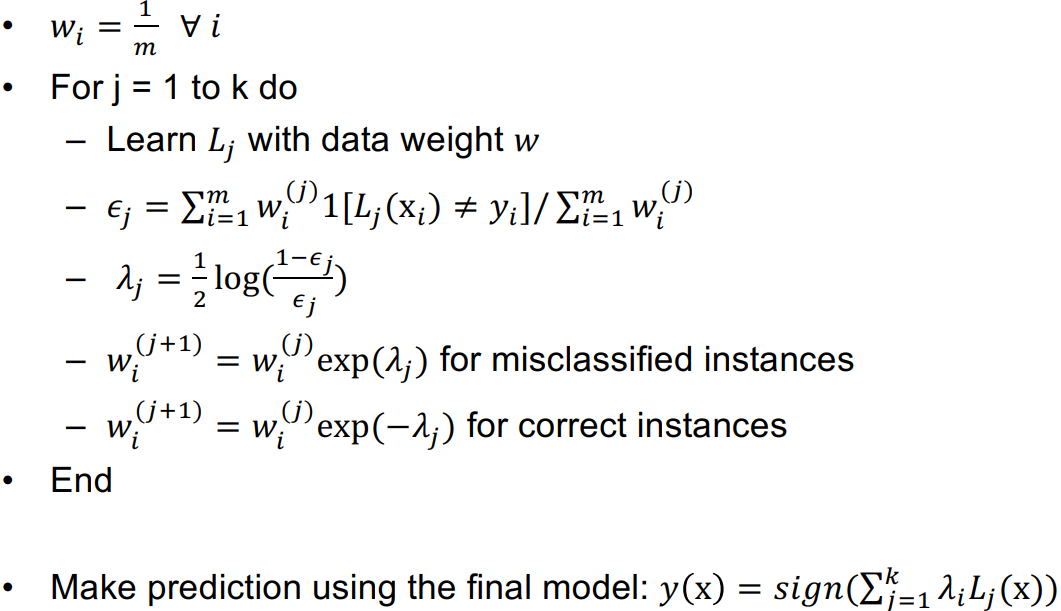
传统决策树在划分属性时是在当前结点的属性集合（d个属性）中选择一个最优属性，而在RF中，对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分

当k=d时，为传统决策树

当k=1时，则是随机选择一个属性用于划分

收敛性：RF的收敛性与bagging相似。随机森林的起始性能往往相对较差

* 1. boosting methods – 用于降低偏差，应用于高偏差模型

-

其中是权值，是误差率，是的系数，是基学习器

好处：不需要使用复杂模型，可以提高任何弱学习器的表现，非常简单容易实现，降低偏差，减小方差

坏处：缺乏可解释性，在训练和测试过程中速度缓慢

**Neural Learning**