## 朴素贝叶斯（native Bayes）

将输入x分到后延概率最大的类y

#### 注意点：

1 朴素贝叶斯对条件概率分布做了条件独立性的假设。

#### 相关知识点：

1 先验概率与后验概率

事情还没有发生,要求这件事情发生的可能性的大小,是先验概率.

事情已经发生,要求这件事情发生的原因是由某个因素引起的可能性的大小,是后验概率.

先验概率是指根据以往经验和分析得到的概率，如全概率公式，它往往作为“由因求果”问题中的“因”出现。后验概率是指在得到“结果”的信息后重新修正的概率，如贝叶斯公式中的，是“执果寻因”问题中的“因”。先验概率与后验概率有不可分割的联系，后验概率的计算要以先验概率为基础。

2 利用极大似然估计计算的时候可能出现概率值为0的情况，会影响到后验概率的结果，使分类产生偏差。因此采取增加的方式，称为拉普拉斯平滑。

#### 优点：

1 因为独立性的假设，模型包含的条件概率的数量大为减少，因此高效，且易于实现。

2 对小规模的数据表现很好，适合多分类任务，适合增量式训练。

#### 缺点：

1 因为独立性的假设，牺牲了一定的分类准确率，分类的性能不一定高。

2 对输入数据的表达形式很敏感。

## KNN(k-nearest neighbor)

#### 注意点：

1 K值的减小意味着整体模型变得复杂，容易发生过拟合。K值变大意味着整体模型变得简单。 K值通常选取一个比较小的数据，采用交叉验证法来选取最优的K值。常用的分类决策规则是多数表决，对应于经验风险最小化。

2 KNN的基本做法是：对给定的训练实例点和输入实例点，首先确定输入实例点的k个最近邻训练实例点，然后利用这k个训练实例点的类的多数来预测输入实例点的类。

3 KNN模型对应于基于训练数据集对特征空间的一个划分。KNN中，当训练集，距离度量，k值和分类决策规则确定后，其结果唯一确定。

4 KNN的实现需要考虑如何快速搜索k个最近邻点。kd树是一种便于对k维空间中的数据进行快速检索的数据结构。kd树是二叉树，表示对k维空间的一个划分，其每个节点对应于k维空间划分的一个超矩形区域。利用kd树可以省去对大部分数据点的搜索，从而减少搜索的计算量。kd树的平均计算复杂度是O(logN).

#### 相关知识点：

#### 优点：

1 思想简单，理论成熟，既可以用来做分类也可以用来做回归。

2 可用于非线性分类。

3 训练时间复杂度为O(n).

4 准确度高，对数据没有假设，对outlier不敏感。

#### 缺点：

1 计算量大

2 样本不平衡（即有些类别的样本数量很多，而其他样本的数量很少）

3 需要大量的内存

## 决策树（decision tree）

熵（entropy）的定义：表示随机变量不确定性的度量。熵越大，随机变量的不确定性越大。

如果对数以2为底，那么熵的单位是比特（bit），如果以为底，那么单位是纳特（nat）。

信息增益（information gain）表示得知特征A的信息而使得训练数据集D的信息的不确定性减少的程度。信息增益打的特征具有更强的分类能力。

#### 注意点：

1 因为从可能的决策树中直接选取最有决策树是NP完全问题。现实中采用启发式方法学习次优的决策树。决策树的学习算法包含3部分，特征性选择，树的生成和树的剪枝、常用的算法有ID3，C4.5和CART。

2 特征选择的目的在于选取对训练数据能够分类的特征。特征选择的关键是其准则。常用的准则如下：

1. 样本集合D对特征A的信息增益（ID3）

其中，H(D)是数据集D的熵，H()是数据集的熵，H(D|A)是数据集D对特征A的条件熵。是D种特征A取第i个值的样本子集，是D中属于第k类的样本子集。|D|表示其样本容量，即样本个数。n是特征A取值的个数，K是类的个数。

1. 样本集合D对特征A的信息增益比（C4.5）

其中，g(D,A)是信息增益，H(D)是数据集D的熵。

1. 样本集合D的基尼指数（CART） CART的决策树是二叉树

特征A条件下集合D的基尼指数：

3 决策树的生成。通常使用信息增益最大，信息增益比最大或者基尼指数最小作为特征选择的准则。决策树的生成往往通过计算信息增益或其他指标，从根节点开始，递归地产生决策树。这相当于用信息增益或其他准则不断的选取局部最优的特征，或将训练集分割为能够基本正确分类的子集。

4 决策树的剪枝。由于生成的决策树存在过拟合的问题，需要对它进行剪枝，以简化学到的决策树。决策树的剪枝，往往从已生成的树上减掉一些叶节点或叶节点以上的子树，并将其父节点或根节点作为新的叶节点，从而简化生成的决策树。

决策树的减枝分为预剪枝和后减枝，预剪枝通过验证集的方式实现，后减枝可以通过验证集或者损失函数来实现。

#### 相关知识点：

#### 优点：

1 计算量简单，可解释性强，比较适合处理有缺失属性值的样本，能够处理不相关的特征。

#### 缺点：

1 容易过拟合（后续出现了随机森林，减小了过拟合的现象）

## 逻辑回归

## 线性回归

## 感知机（perceptron）

感知机是根据输入实例的特征向量x对其进行二类分类的线性分类模型：

感知机模型对应于输入空间（特徵空间）中的分离超平面

#### 注意点：

1 感知机学习算法是基于随机梯度下降法的对损失函数的最优化算法，有原始形式和对偶形式。原始形式中，首先任意选取一个超平面，然后利用梯度下降法不断极小化目标函数。在这个过程中一次随机选取一个误分类点使其梯度下降。

2 感知机的损失函数是根据误分类点到超平面的总距离

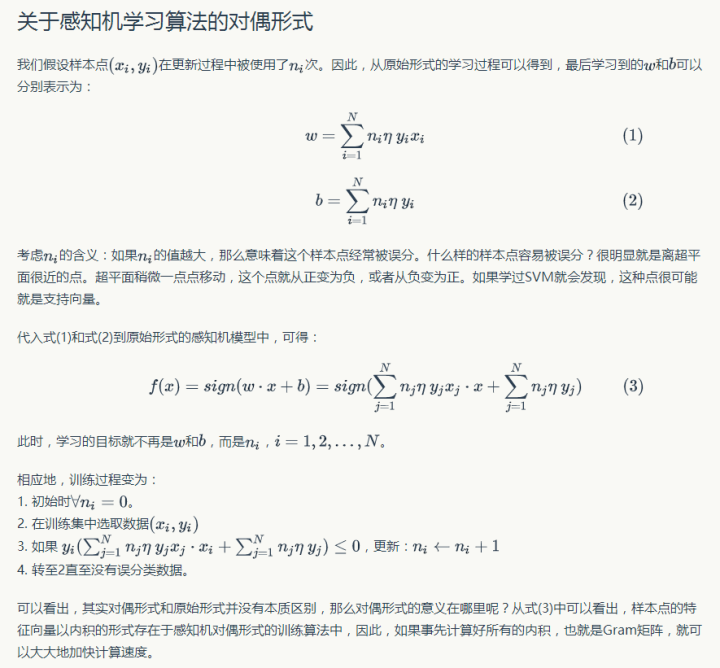
其中M是误分类点的集合。这个损失函数就是感知机学习的经验风险函数。

3 当训练数据集线性可分时，感知机学习算法是收敛的。感知机算法在训练数据集上的误分类次数k满足不等式：

当训练数据集线性可分时，感知机学习算法存在无穷多个解，其解由于不同的初值或不同的迭代顺序而可能有所不同。

4 采用随机梯度下降法进行优化，具体的实现要看书。

5 采用对偶形式的原因



#### 相关知识点：

#### 优点：

#### 缺点：

1 只能数据集线性可分时才能使用，只能用于二分类。

## 支持向量机（SVM）

#### 注意点：

1 支持向量机最简单的情况是线性可分支持向量机，或硬间隔支持向量机。构建它的条件是训练数据线性可分。其学习策略是最大间隔法。可以表示为凸二次规划问题，其原始最优化问题为

求的最优化问题的解为,得到线性可分支持向量机，分离超平面是

分类决策函数是

最大间隔法中，函数间隔与几何间隔是重要的概念。

点到分离超平面的距离是, 因此可用

超平面（w,b）对所有样本点（）函数间隔为

几何间隔是对函数间隔做规范化，

如果超平面参数w和b成比例的改变（超平面没有改变），函数间隔按此比例改变，而几何间隔不变。

间隔最大化就是对训练数据集找到几何间隔最大的超平面。

考虑几何间隔和函数间隔的关系，也可改写为

发现函数间隔对上面的最优化没有影响，因此取，然后注意到最大化是等价的。于是得到

线性可分支持向量机的最优解存在且唯一。位于间隔边界上的实例点为支持向量。最优分离超平面有支持向量完全决定。

引入拉格朗日函数

根据拉格朗日对偶性

先求对w,b的极小，再求对的极大

先求极小

将w的结果带入拉格朗日函数中，得到

然后求极大，将极大转换为极小，就得到如下的

二次规划问题的对偶问题是

通常，通过求解对偶问题学习线性可分支持向量机，即首先求解对偶问题的最优值，然后求最优值,得出分离超平面和分类决策函数。

并选择一个正分量，计算

2 现实中训练数据是线性可分的情形比较少，训练数据往往是近似线性可分的，这时使用线性支持向量机，或软间隔支持向量机。线性支持向量机是最基本的支持向量机。

对于噪声或者例外，通过引入松弛变量，使其“可分”，可以解释为某些样本点不能满足函数间隔大于等于1，因此就使函数间隔加上松弛变量大于等于1. C>0称为惩罚参数。得到线性支持向量机学习的凸二次规划问题，其原始最优化问题是

求解原始最优化问题的解，得到线性支持向量机，其分离超平面为

分类决策函数为

线性可分支持向量机的解唯一但不唯一。

对偶问题

线性支持向量机的对偶学习算法，首先求解对偶问题得到最优解，然后求原始问题最优解，得出分离超平面和分类决策函数。

原始问题的拉格朗日函数是

其中

先求对偶问题的极小

然后得到

然后再求极大得到

求解

并选择适合，计算

因为不唯一，实际计算时会取所有满足条件样本上的平均值。

对偶问题的解中满足的实例点称为支持向量。支持向量可在间隔边界上，也可在间隔边界与分离超平面之间，或者在分离超平面误分一侧。最优分离超平面由支持下向量完全决定。

线性支持向量机学习等价于最小化二阶范数正则化的合页函数

3 非线性支持向量机

对于输入空间中的非线性分类问题，可以通过非线性变换将它转化为某个高维特征空间中的线性分类问题，在高维特征空间中学习线性支持向量机。由于在线性支持向量机学习的对偶问题里，目标函数和分类决策函数都只涉及实例与实例之间的内积，所以不需要显示地指定非线性变换，而是用核函数来代替当中的内积。核函数表示，通过一个非线性转换后的两个实例间的内积。具体的，K(x,z)是一个核函数，或正定核，意味着存在一个从输入空间到特征空间的映射, 对任意 ,有

对称函数K(x,z)为正定核的充要条件如下：对任意,任意正整数m，对称函数K(x,z)对应的Gram矩阵是半正定的。

所以，在线性支持向量机学习的对偶问题中，用核函数K(x,z)代替内积，求解得到的就是非线性支持向量机

求解

求解最优解

并选择适合，计算

核函数

#### 线性核

线性核，主要用于线性可分的情况，我们可以看到特征空间到输入空间的维度是一样的，其参数少速度快，对于线性可分数据，其分类效果很理想，因此我们通常首先尝试用线性核函数来做分类，看看效果如何，如果不行再换别的

#### 多项式核

多项式核函数可以实现将低维的输入空间映射到高纬的特征空间，但是多项式核函数的参数多，当多项式的阶数比较高的时候，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度会大到无法计算。

#### 高斯（RBF）核

是高斯核的width

高斯径向基函数是一种局部性强的核函数，其可以将一个样本映射到一个更高维的空间内，该核函数是应用最广的一个，无论大样本还是小样本都有比较好的性能，而且其相对于多项式核函数参数要少，因此大多数情况下在不知道用什么核函数的时候，优先使用高斯核函数。

#### sigmoid核

tanh为双曲正切函数，

采用sigmoid核函数，支持向量机实现的就是一种多层神经网络。

因此，在选用核函数的时候，如果我们对我们的数据有一定的先验知识，就利用先验来选择符合数据分布的核函数；如果不知道的话，通常使用交叉验证的方法，来试用不同的核函数，误差最下的即为效果最好的核函数，或者也可以将多个核函数结合起来，形成混合核函数。在吴恩达的课上，也曾经给出过一系列的选择核函数的方法：

如果特征的数量大到和样本数量差不多，则选用LR或者线性核的SVM；

如果特征的数量小，样本的数量正常，则选用SVM+高斯核函数；

如果特征的数量小，而样本的数量很大，则需要手工添加一些特征从而变成第一种情况。

4 SMO算法

SMO算法是支持向量机学习的一种快速算法，其特点是不断地将原二次规划问题分解为只有两个变量的二次规划子问题，并对子问题进行解析求解，直到所有变量满足KKT条件为止。这样通过启发的方法得到的原二次规划问题的最优解。因为子问题有解析解，所以每次计算子问题都很快，虽然计算子问题次数很多，但在总体上还是很高效的。

#### 相关知识点：

### 1 拉格朗日对偶性

#### 用途：

对于约束最优化问题，有时原始问题的最优解不好求解，可以借助拉格朗日对偶性将原始问题转化为对偶问题，通过求解对偶问题的解来获得原始问题的最优解。在最大熵模型和支持向量机中用到了该类方法。

#### 描述

#### 原始问题：

假设f(x),是定义在上的连续可微函数。考虑约束最优化问题

称此约束最优化问题为原始最优化问题或者原始问题

引入广义拉格朗日函数

其中，,是拉格朗日乘子，。考虑x的函数

其中下标p表示原始问题。

#### 对偶问题

称为原始问题的对偶问题。

#### KKT条件：

假设函数f(x)和是凸函数，是仿射函数，并且不等式约束是严格可行的，则分别是原始问题和对偶问题的解的充分必要条件是满足KKT条件：

特别指出，式子A称为KKT的对偶互补条件，由此条件可知，若，则

#### 优点：

可用于线性/非线性分类，也可以用于回归；

低泛化误差；

容易解释；

计算复杂度较低；

#### 缺点：

对参数和核函数的选择比较敏感；

原始的SVM只比较擅长处理二分类问题；

## 神经网络

参考：http://blog.csdn.net/acdreamers/article/details/44657439

http://www.cnblogs.com/liuwu265/p/4696388.html

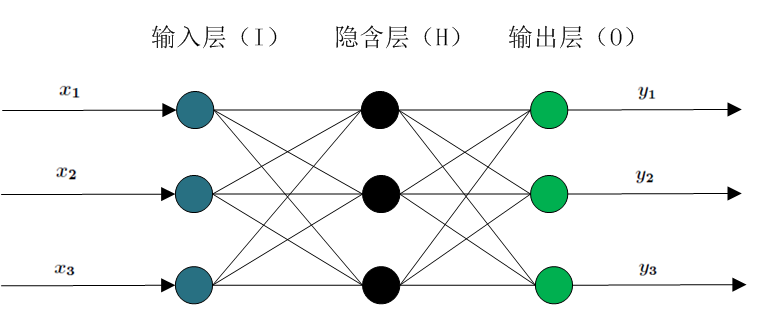
#### 注意点：

**BP（Back Propagation）神经网络**分为两个过程

     （1）工作信号正向传递子过程

     （2）误差信号反向传递子过程

   在BP神经网络中，单个样本有m个输入，有n个输出，在输入层和输出层之间通常还有若干个隐含层。实际上，**1989**年**Robert Hecht-Nielsen**证明了对于任何闭区间内的一个连续函数都可以用一个隐含层的BP网络来逼近，这就是**万能逼近定理**。所以一个三层的BP网络就可以完成任意的m维到n维的映射。即这三层分别是**输入层（I），隐含层（H），输出层（O）**。如下图示



**2. 隐含层的选取**

   在BP神经网络中，输入层和输出层的节点个数都是确定的，而隐含层节点个数不确定，那么应该设置为多少才合适呢？实际上，隐含层节点个数的多少对神经网络的性能是有影响的，有一个经验公式可以确定隐含层

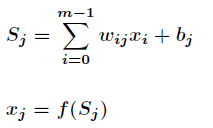
   节点数目，如下

http://images.cnitblog.com/blog/571227/201412/191145471412994.png

   其中h为隐含层节点数目，m为输入层节点数目，n为输出层节点数目，为1-10之间的调节常数。

**3. 正向传递子过程**

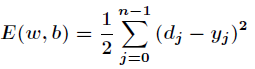
   现在设节点i和节点j之间的权值为，节点j的阀值为，每个节点的输出值为，而每个节点的输出值是根据上层所有节点的输出值、当前节点与上一层所有节点的权值和当前节点的阀值还有激活函数来实的。具体计算方法如下



 其中f为激活函数，一般选取s型函数或者线性函数。正向传递的过程比较简单，按照上述公式计算即可。在BP神经网络中，输入层节点没有阀值

**4. 反向传递子过程**

 在BP神经网络中，误差信号反向传递子过程比较复杂，它是基于**Widrow-Hoff**学习规则的。假设输出层的所有结果为，误差函数如下



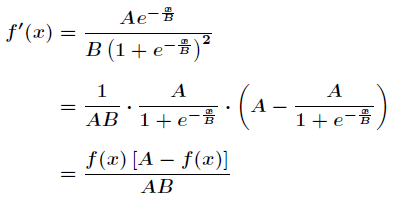
  而BP神经网络的主要目的是反复修正权值和阀值，使得误差函数值达到最小。**Widrow-Hoff**学习规则是通过沿着相对误差平方和的最速下降方向，连续调整网络的权值和阀值，根据梯度下降法，权值矢量的修正正比于当前位置上E(w,b)的梯度，对于第j个输出节点有

http://images.cnitblog.com/blog/571227/201412/191451343134484.png

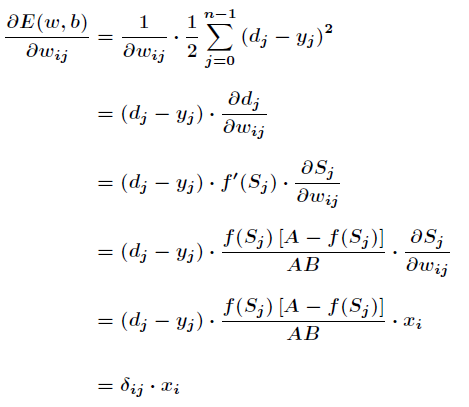
   假设选择激活函数为

http://images.cnitblog.com/blog/571227/201412/191459283446336.png

   对激活函数求导，得到



   那么接下来针对有



   其中有

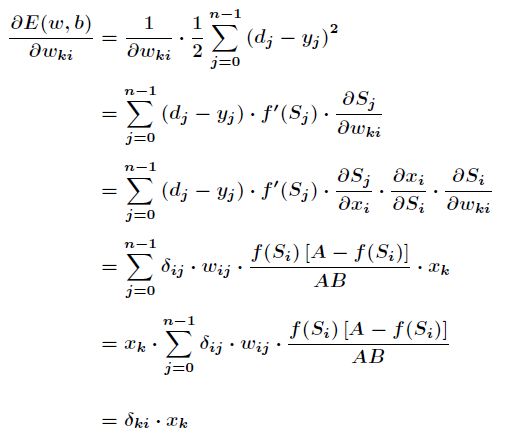
http://images.cnitblog.com/blog/571227/201412/191600406735354.png

   同样对于有

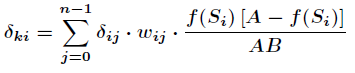
http://images.cnitblog.com/blog/571227/201412/191643504382508.png

   这就是著名的http://images.cnitblog.com/blog/571227/201412/191603072988848.png**学习规则，**通过改变神经元之间的连接权值来减少系统实际输出和期望输出的误差，这个规则又叫做**Widrow-Hoff**学习规则或者**纠错学习规则**。

   上面是对隐含层和输出层之间的权值和输出层的阀值计算调整量，而针对输入层和隐含层和隐含层的阀值调整量的计算更为复杂。假设是输入层第**k**个节点和隐含层第**i**个节点之间的权值，那么有

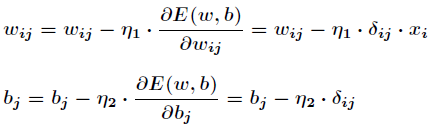


   其中有

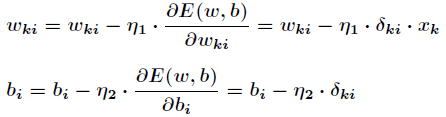


   这样对http://images.cnitblog.com/blog/571227/201412/191730562515010.png**学习规则**理解更为深刻了吧。

   有了上述公式，根据**梯度下降法**，那么对于隐含层和输出层之间的权值和阀值调整如下



   而对于输入层和隐含层之间的权值和阀值调整同样有



   至此BP神经网络的原理基本讲完。

**5. BP神经网络的注意点**

   BP神经网络一般用于分类或者逼近问题。如果用于分类，则激活函数一般选用Sigmoid函数或者硬极限函 数，如果用于函数逼近，则输出层节点用线性函数，即http://images.cnitblog.com/blog/571227/201412/191908593134685.png。

   BP神经网络在训练数据时可以采用增量学习或者批量学习。

**增量学习**要求输入模式要有足够的随机性，对输入模式的噪声比较敏感，即对于剧烈变化的输入模式，训练效果比较差，适合在线处理。**批量学习**不存在输入模式次序问题，稳定性好，但是只适合离线处理。

**标准BP神经网络的缺陷：**

   （1）容易形成局部极小值而得不到全局最优值。

       BP神经网络中极小值比较多，所以很容易陷入局部极小值，这就要求对初始权值和阀值有要求，要使得初始权值和阀值随机性足够好，可以多次随机来实现。

   （2）训练次数多使得学习效率低，收敛速度慢。

   （3）隐含层的选取缺乏理论的指导。

   （4）训练时学习新样本有遗忘旧样本的趋势。

过拟合

　　当网络的训练次数过多时，可能会出现过拟合的情况。解决过拟合主要两种方法：一种是使用权值衰减的方式，即每次迭代过程中以某个较小的因子降低每个权值；另一种方法就是使用验证集的方式来找出使得验证集误差最小的权值，对训练集较小时可以使用交叉验证等。

**BP算法的改进：**

   （1）增加动量项

       加入冲量项在一定程度上起到加大搜索步长的效果，从而能更快的进行收敛。另一方面，由于多层网络易导致损失函数收敛到局部极小值，但通过冲量项在某种程度上可以越过某些狭窄的局部极小值，达到更小的地方，即如下公式

http://images.cnitblog.com/blog/571227/201412/201532434541008.png

       动量因子http://images.cnitblog.com/blog/571227/201412/201534002048721.png一般选取http://images.cnitblog.com/blog/571227/201412/201534549071933.png。

   （2）自适应调节学习率

   （3）引入陡度因子

   通常BP神经网络在训练之前会对数据**归一化处理**，即将数据映射到更小的区间内，比如[0,1]或[-1,1]。

优点：

#### 缺点：

## GBDT和XGBOOST

## 随机森林

## kmeans

## EM