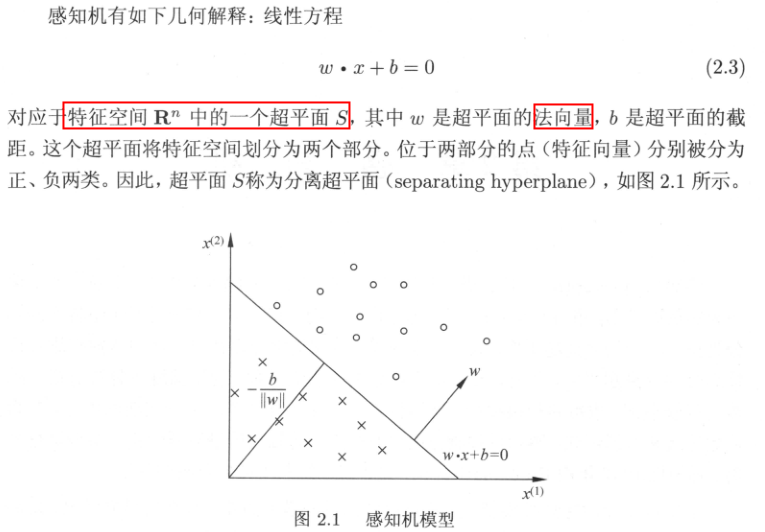
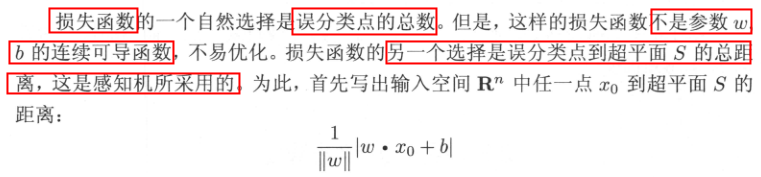
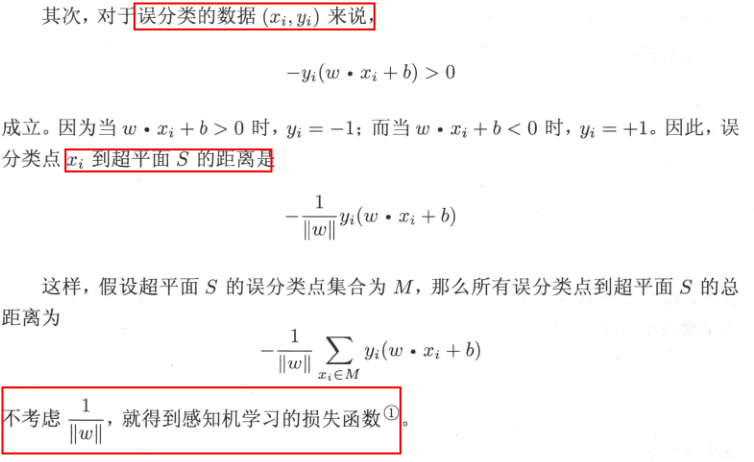
# 2.感知机

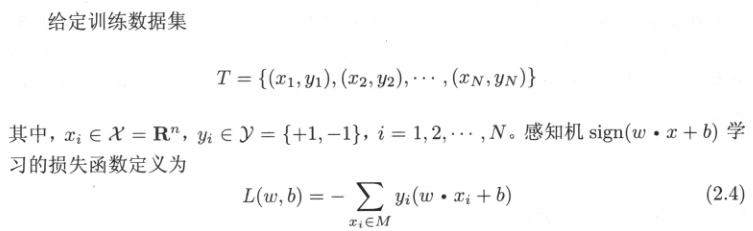
## 2.1、感知机的几何解释



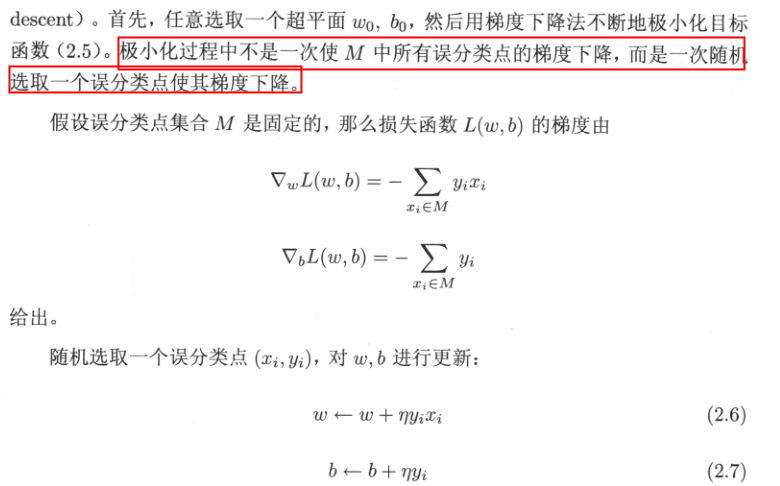
## 2.2、学习策略





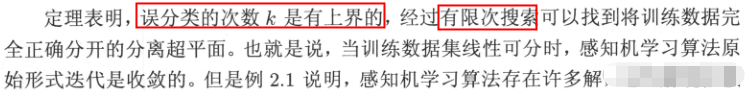


## 2.3、梯度学习算法

****

****

## 2.4、为什么可以收敛



另外，算法是误分类驱动的，一旦没有误分类的样本点，算法就停止了，所以不会像SVM那样找到全局最优解。

## 2.5、多解因素

初值的选择+误分类点的选择(顺序)

**-----------------------------------------------------------------------------------------**

# 4.朴素贝叶斯

最直接的，假如目标是求P(y|x)，可以根据贝叶斯公式转化为：p(y)\*p(x|y)/p(x)进一步的，假如想求哪一个y相对较大，可以只考虑分子部分。所以直接的目标就是

1.求出y不同取值的*先验概率*(边缘概率)

2.求y为某值对应的*条件概率*

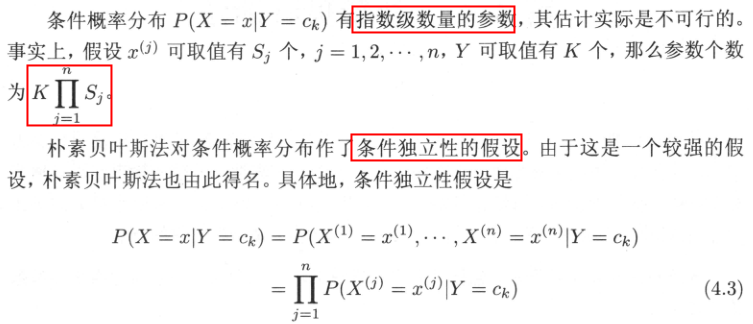
然后求一个最大的类别概率作为当前样本的分类。大概意思是说：别问我你属于哪一类，让我先看下，哪一类更可能生成你。

符号说明：

n:特征x的维度；N：样本个数；Sj：第j个特征的取值个数

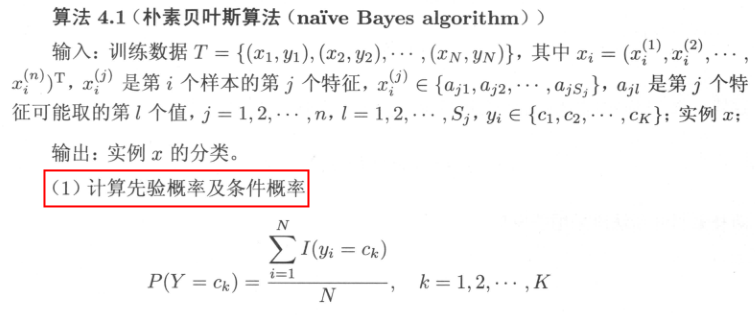
ajl：第j个特征的第l个取值。可以理解为一个具体的数。

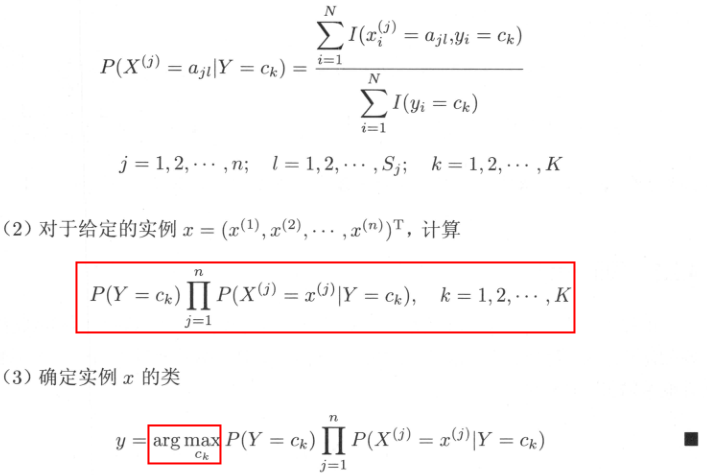
**问题：为什么叫朴素贝叶斯？or什么是特征条件独立假设？**



上图中讲参数的个数：实际上这个算法没有w，b，只是计算先验概率，可以理解为要计算先验概率的个数。

**参数估计-最大似然估计：**





**参数估计方法2：贝叶斯估计-拉普拉斯平滑技巧**

****

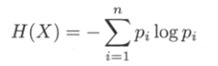
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# 5.决策树

## 前置知识：

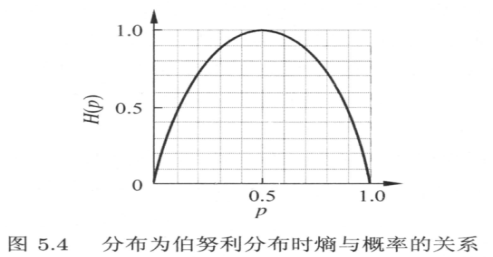
1、熵、条件熵、信息增益

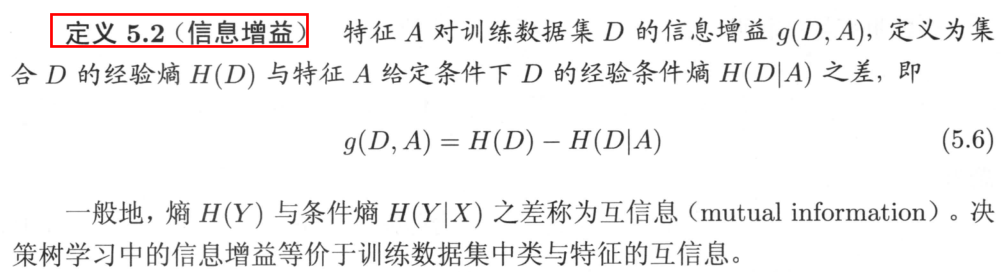
熵与条件熵的定义为：

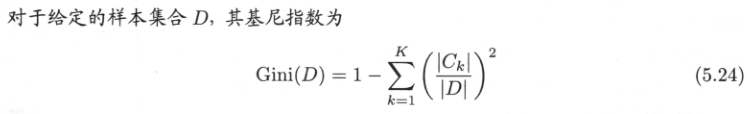
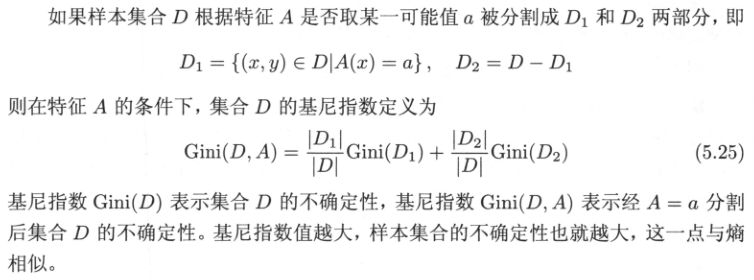




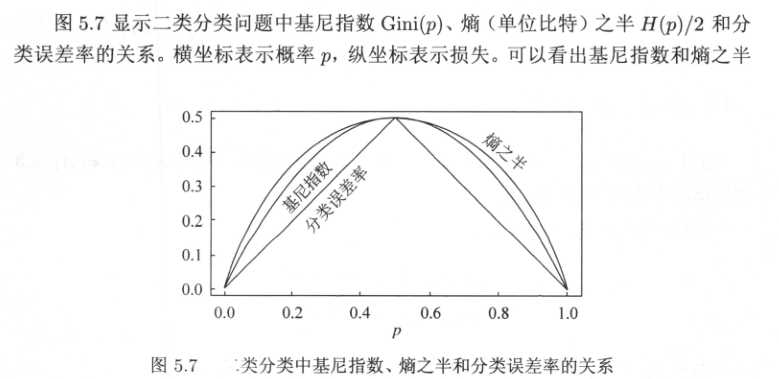
熵能表示随即***变量的不确定性***。根据等价无穷小的有：logx->x-1,xlogx->x(x-1)。基于此来验证下面的公式可以看出，当p为0或者1的时候，熵最小=0。这符合我们对不确定性的理解。实际上，对于一个二分类问题有：





2、基尼指数  

熵表示变量的不确定性，基尼指数表示集合的不确定性，假如我们将集合的划分用变量体现，那么也可以计算集合的熵，所以两者是相似的。



## 5.1、经典分类决策树

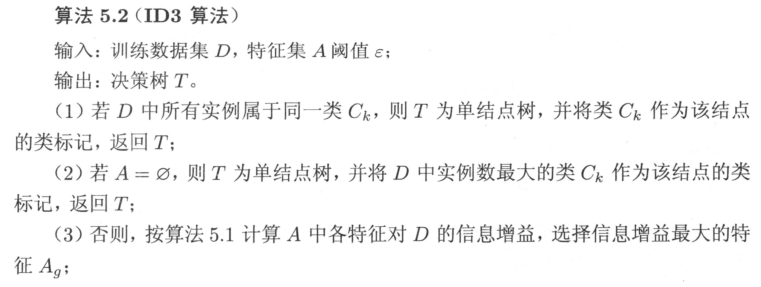
### 5.1.1 特征选择

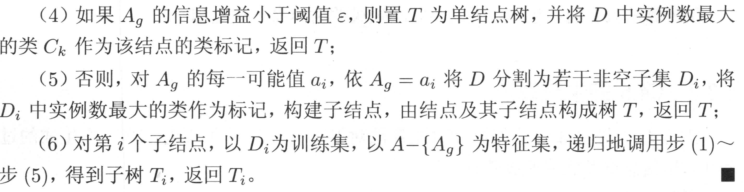
ID3：最大化信息增益。

C4.5：最大化信息增益比。

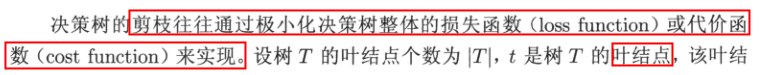
### 5.1.2 决策树的生成

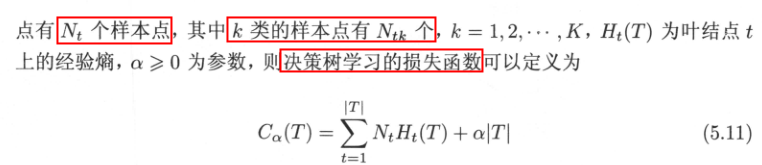
递归的进行特征选择

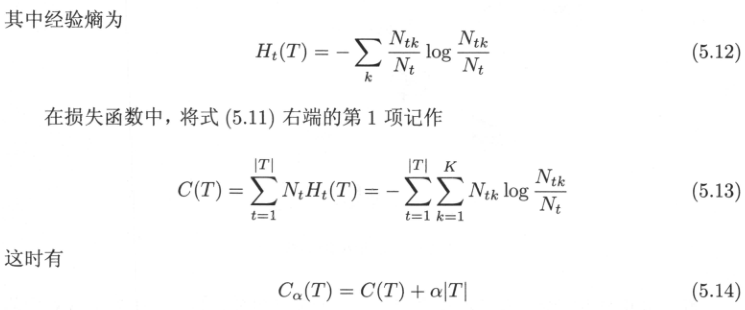
****

****

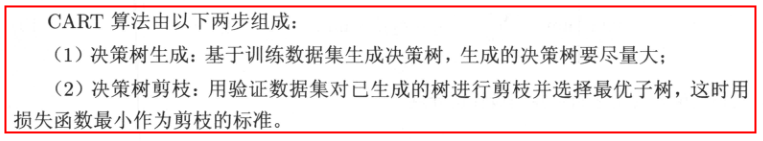
### 5.1.3 剪枝

****

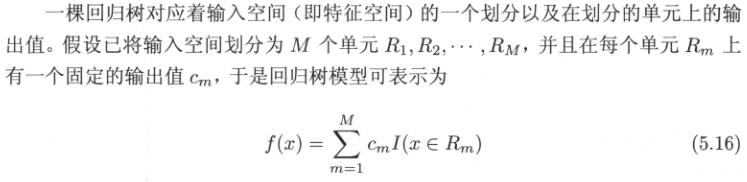
****

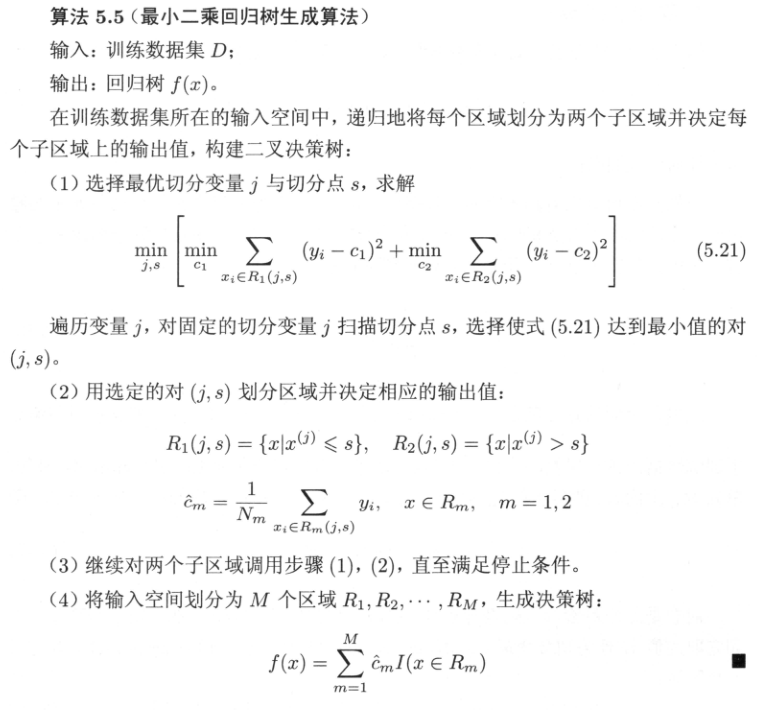
****

## 5.2、CART



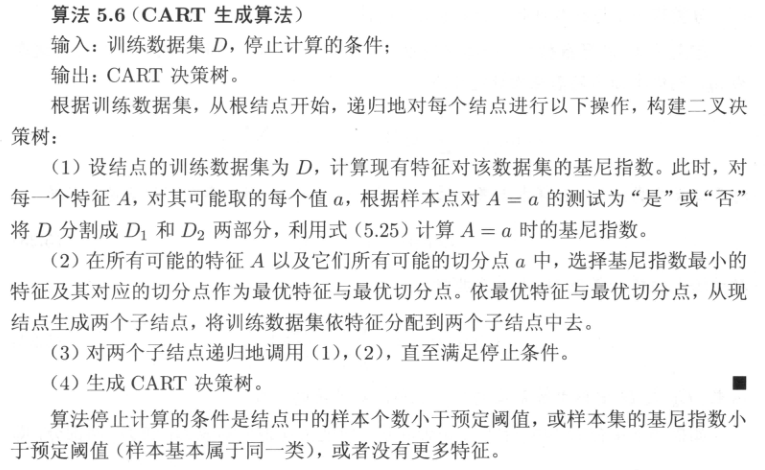
### 5.2.1 回归树





**算法理解：**遍历所有的切分变量与切分点，每一次确定一种划分后：根据公式计算响应的平方差的和，最小值对应的切分变量与切分点就是最佳值。继续在子集上进项相同的操作，知道满足停止条件。

**5.2.2 CART分类树**



算法理解：类似于会归属，需要选择最佳的切分变量与切分点，区别在于回归树用最小二乘法选择，而分类树用基尼指数选择。

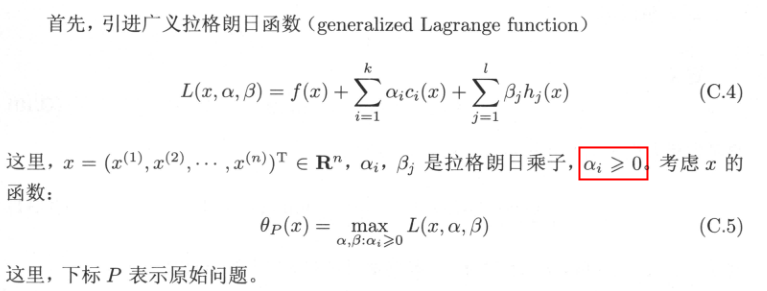
**小结**：决策树算法考量不同特征划分数据集带来的信息增益，不会考虑切分点，所以假如特征有M个候选值，那么就会有M叉，但是CART算法考虑切分变量+切分点的最佳组合，所以生成的是二叉树。

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

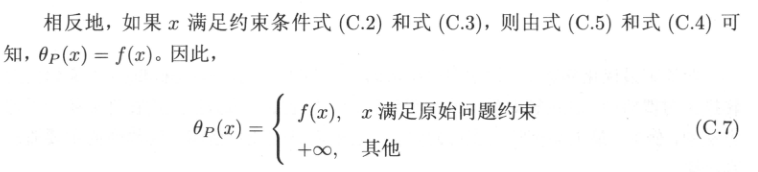
# 7.支持向量机

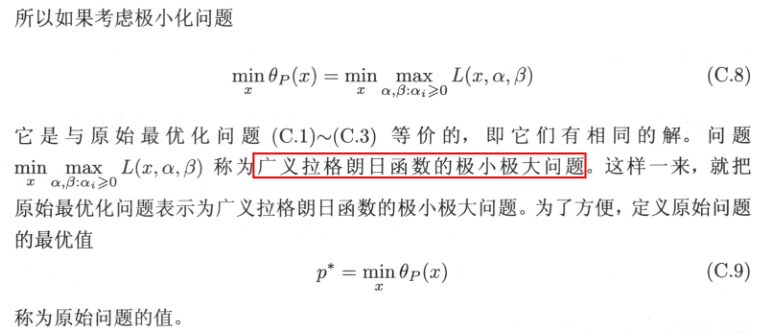
## 7.1．拉格朗日对偶性

****

****

**关键：面向拉格朗日函数，原始问题首先对其求了最大。**

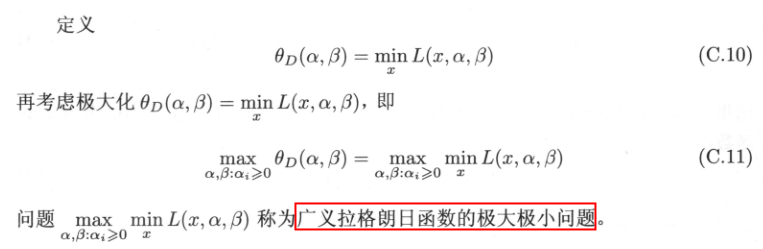
****

****

小结一下：

在引入拉格朗日函数之前，是一个约束最优化问题，变量是x；在朗格朗日函数中：x，α，β都是自变量；拉格朗日函数关于α，β求最大后又变成只是x的函数即为C.5所示；在原始问题的定义域内问题等价；改变x，函数能取到的最小值是P\*。

对偶问题

****

关键：面向拉格朗日函数，对偶问题首先对其求了极小。

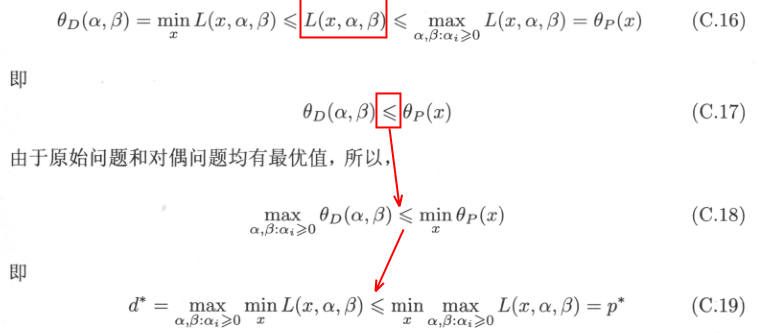
1.3.两者关系

原始问题：先极大再极小

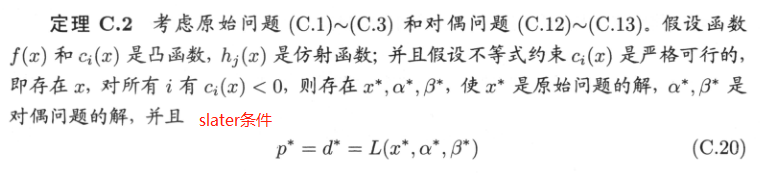
对偶问题：先极小再极大

所以那个大呢？一个是阆苑仙葩，一个是美玉无瑕，若说没奇缘，今生偏又遇见他，若说有奇缘，如何心事终虚化？

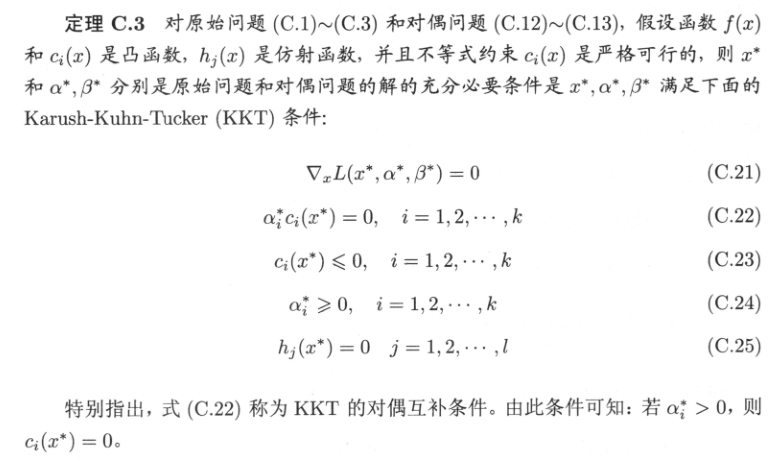
言归正传：当然是原始问题的大，为什么？因为第一步真的很重要！不信你看：

****

1.4.最优解存在的充分条件

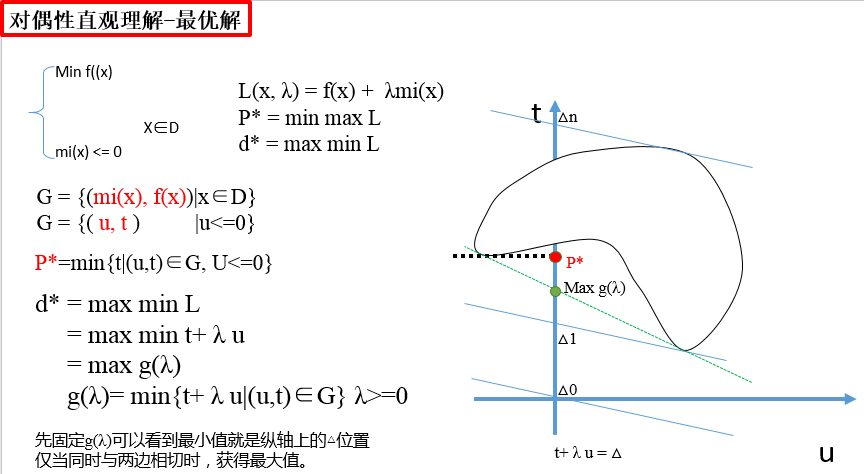
****

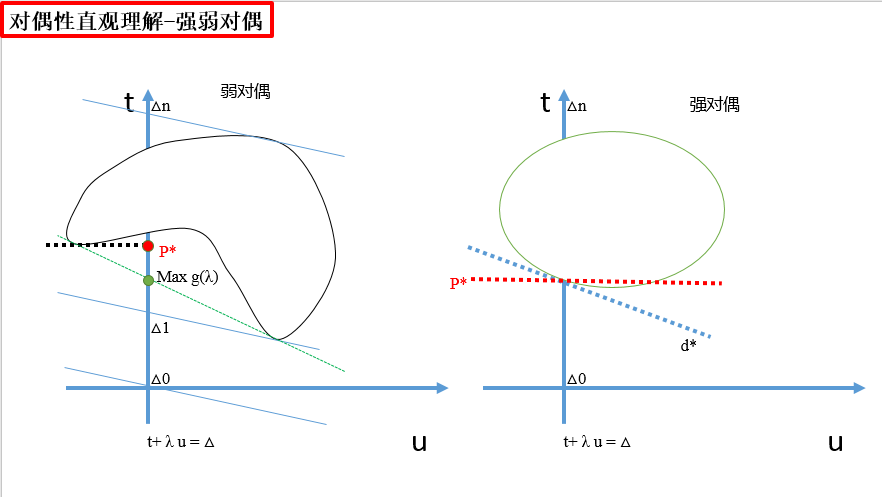
1.5.最优解充要条件

****

看了1.4和1.5可能有点懵逼吧.tell me why

## 7.2、拉格朗日对偶性可视化

****

****

2.2 slatter条件

至少存在一点x，使得所有mi(x)<0，也就是满足原问题约束条件。两点note:

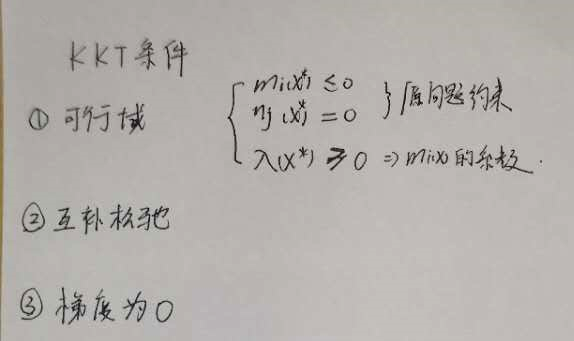
1、对于大多数凸优化问题，slatter是满足的；2、放松的slatter：假如mi(x)中有K个仿射函数，只用校验M-K个条件。

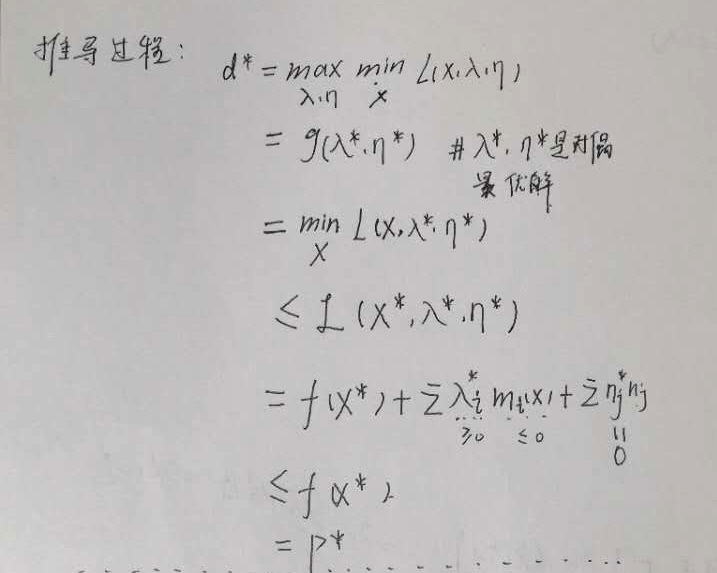
假如一个问题是凸二次规划问题，即：

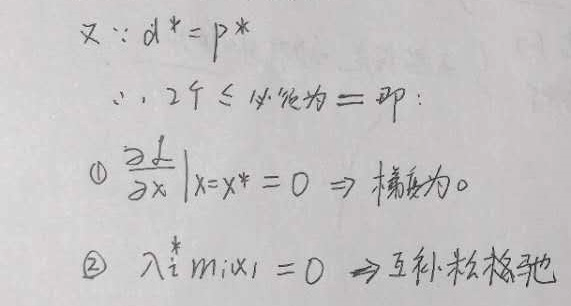
1、f 是凸的；2、约束条件mi(x) 仿射函数

那么天然的满足slatter条件，即满足强对偶关系，原问题和对偶问题的最优解是相同的。Svm天然满足强对偶关系。

2.3 KKT条件

****

****

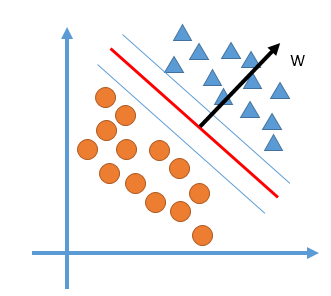
****

**从0到1支持向量机**

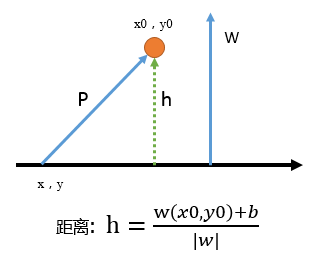
# 预备知识

1、【法向量】Wx+b=0，w是什么？

Wx+b=0是直线方程，其中w表示法向量，法向量的指向由具体值确定。例如x+y-2=0，法向量为(1,1)，指向右上方。



2、【距离公式】



3、【函数间隔】

当w确定的时候，距离的远近可以比较分子，也就是说wx+b的绝对值表示样本点到分类平面的相对距离，显然点在w指向的方向距离为正值，规定正类为这一侧，同样的分析应用到负类，那么y(wx+b)即可以表征分类准确性，又可以表征分类的置信度，具体而言绝对值的大小表示置信度，正负分别表示分类正确、错误，在svm中这被称作函数间隔。

4、【几何间隔】

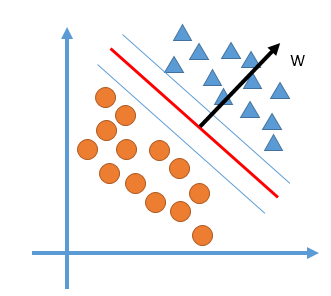
可以理解为真正的距离，与函数间隔是正比关系。

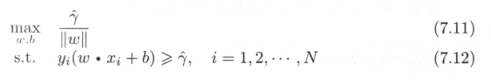
5、【核技巧】

6、【拉格朗日对偶性】

7、【hingelossfunction】

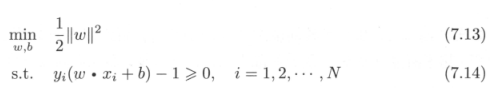
# 7.1 线型可分支持向量机





**解释**：γ是函数距离，γhat所有样本点中距离分类面最小的函数距离。由前置知识可以知道，目标函数是几何间隔最大化。

其中γ hat为到超平面最小的函数间隔，这时的超平面还是一个任意的平面，显然这是一个极大极小问题。因为优化的目标是找到最佳的自变量w,b的比例关系以确定分离超平面，而与变量的绝对大小无关，基于此，我们把函数间隔设置为1.优化目标变为：



满足所有点的函数间隔大于1，最小化w的模长，这时的w和b就是我们的目标值。其实是1还是2都没关系，这个数值是为支持向量准备的函数间隔的值，相当于设定了参考点，别的样本点的函数间隔都应该大于等于该值。

假如未能满足的话就会形成梯度对w和b优化。直到所有样本点满足约束。

能满足约束条件的参数组不止一组，例如w1，b1满足，现在将他们全部放大2倍，仍然满足，为了避免无意义的多解，目标函数对参数的2范数进行约束。使得解是唯一的。

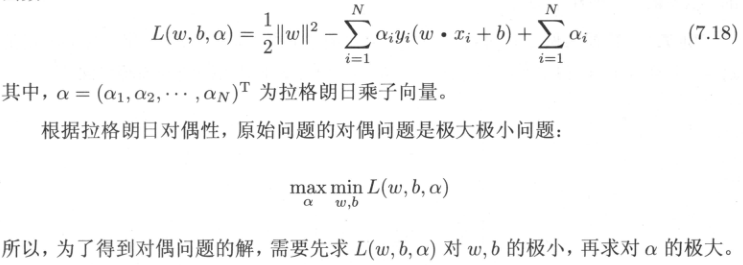
考虑一种比较有意思的情况：假如初始化已经满足了所有的约束条件，但是并没有处在分类中间，接下来会怎样呢？假如两边的支持距离分别为1和2,2说麻烦把b调一下，让我的距离小一点，之后两边的距离变成呢1.2,1.5，指挥部发现这样的话可以把参数缩小1.2倍，正和我意，于是距离变成了1，1.25，之后repeat直到两边的距离都是1。

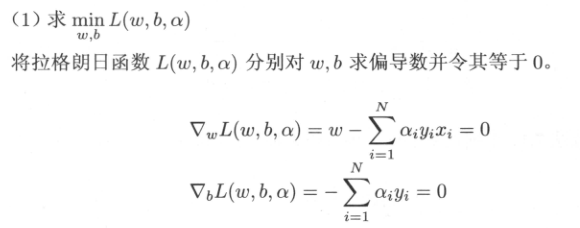
上面是一种情况，还有其他很多种，但是无论如何，最终都是只要w可以优化模型就会不断朝间隔最大化移动。因为：

**几何距离1\*|w1|=函数距离（都是1） =几何距离2 \* |w2|**

几何距离越大，才能使得w的模长越小。

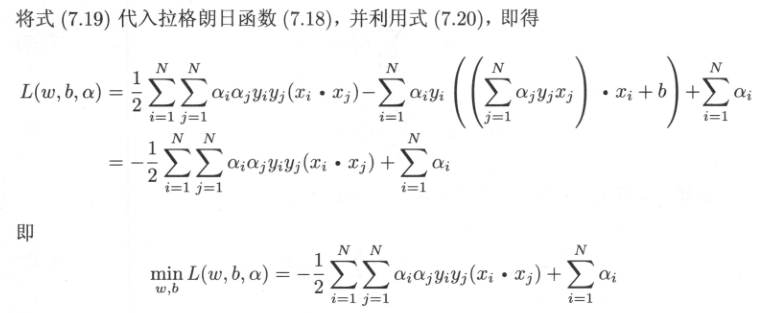
不太严谨的讲：在模型没有满足约束的时候，约束条件使得模型不得不调整到可以正确分类的位置，之后基于最小化范数，模型逐渐调整到中间的位置。以上只是帮助理解的，下面是解决方案：基于拉格朗日函数与对偶算法：构建target：







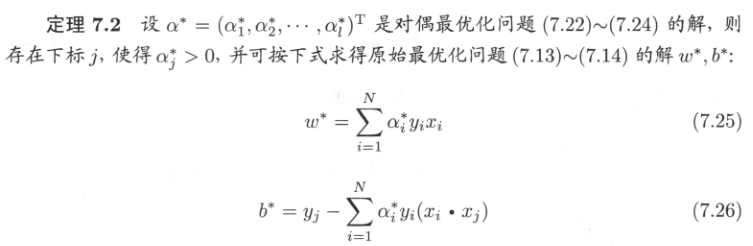




问题转化为：



基于当前的优化目标和约束条件求出α。就相当于在一个高维空间求最小值，然后附带了很多条件。



也就是说，现在问题的焦点在于α的求取。这已经是单变量了，可以直接求解（解析法，梯度法*（需要max（0，步进后的值）*）

## 7.2 线性支持向量机与软间隔最大化

由于数据集已经线型不可分，所以如果仍然要求所有点的函数间隔大于等于某一个正数已经不可能，也就是说每一个样本点需要设置一个专属的函数间隔，ξ为引入的松弛变量（可能大于1），约束条件为：



对松弛变量的理解：引入松弛变量在数学上等价于改变样本点的坐标。

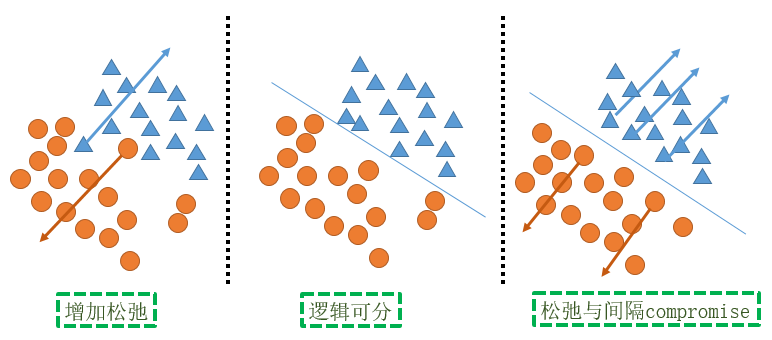
yi（w \* xi + b）+ yi（w \* Δx+ b）>=1，

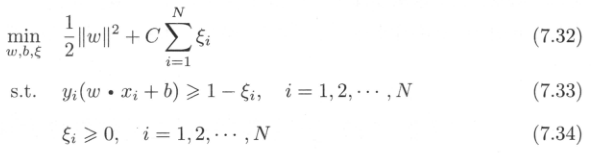
ξ =yi（w \* Δx+ b）

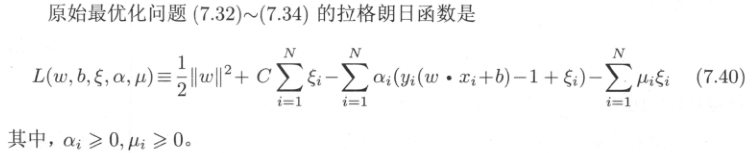
本来样本是不可分的，引入松弛后，变成可分的数据集。至于那些引入，这个不用我们管，模型优化的过程会自动的确定。所以目标函数仍然包含极小化w2，那么对ξ有没有要求呢？当然有，比如抗战时期，国共敌三者人员都是相互渗透的，现在想对敌我分分类，就需要对国军和敌后工作者发放松弛变量，不然就把它们错分为敌人了，但是如果对所有人发放松弛，那就会导致敌我不分。从这个角度，我们希望松弛绩极小。

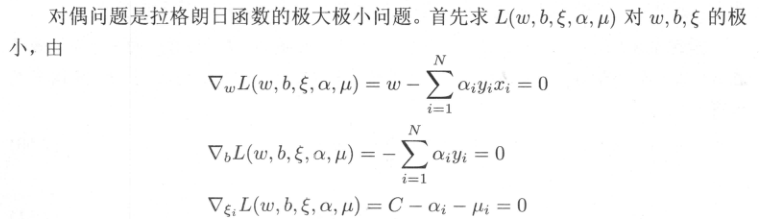
这里要区分逻辑分类结果和实际分类结果，实际上模型是不可分的，无论分类面在哪里，都没有分开，但是引入松弛后在逻辑上，实际上已经分开了。

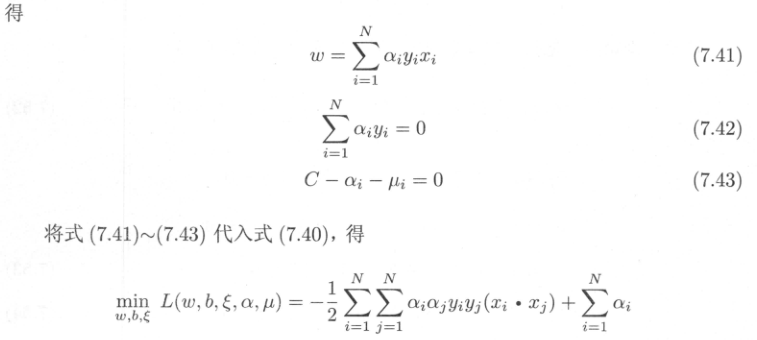
在逻辑可分的情况下，松弛发放的越少，分类间隔越小。继续发放松弛，逻辑上的几何间隔可以变大。即为了追求间隔最大，我们希望扩大松弛，但是松弛又是我们想极小化的，所以，现在不再单纯的追求间隔最大化或者松弛最小化，而是两者的和最小。引入松弛的过程可以理解为改变真正sv的过程，而sv决定间隔。



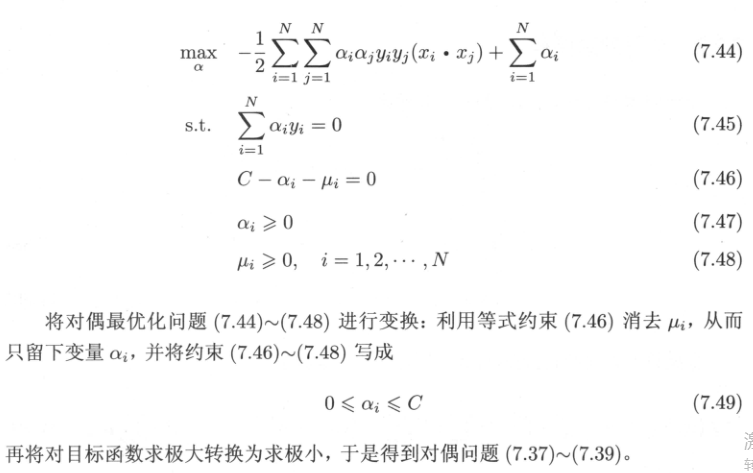


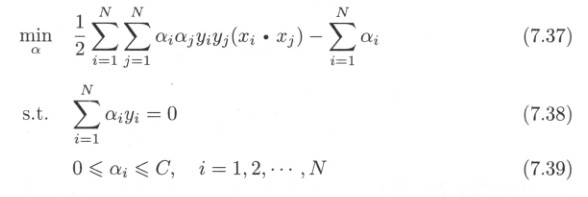


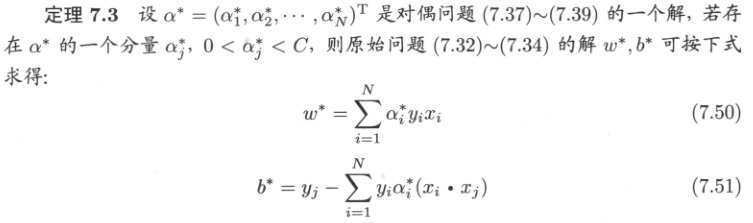












## 7.3 非线性支持向量机与核函数

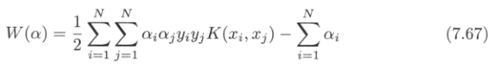


卧槽，发生了什么？怎么就非线性了？

在线型支持向量机的背景下，无论是目标函数，还是决策函数。关键部分都是两个向量的内积。但是在非线性的背景下，这些是不成立的，本质上是因为线型不可分，没有之前讨论的函数间隔，自然没有办法给出优化目标。

但是，设想一下，把现在的空间做一次映射，变成了线型可分的情况，而直接借用之前的目标函数的形式，那么目标函数仍然是向量内积的形式（映射后的向量），即特征空间的向量内积，但是没关系核函数的存在为两个空间建立了通道。

以下式为例进行理解：



可以理解为这是特征空间的目标函数。刚开始看这一部分的时候可能会很疑惑为什么直接从输入空间的内积变成特征空间的内积了？这会相等吗？其实，只是特征空间套用了输入空间的目标函数的模板而已。所以不存在相不相等的问题。而这个模板最关键的部分就是内积，核函数就是特征内积。

利用这个目标函数求α。

另外，从决策函数可以看出，分两类结果也要映射到特征空间进行判定。在整个过程中，我们可以这样看待：

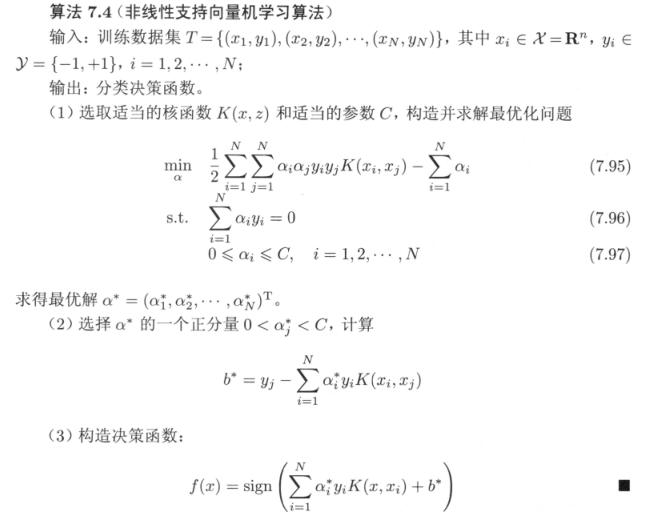
特征空间是进行判定的空间，输入空间是数据源，而核函数决定了输入向量映射到特征空间所处的位置。

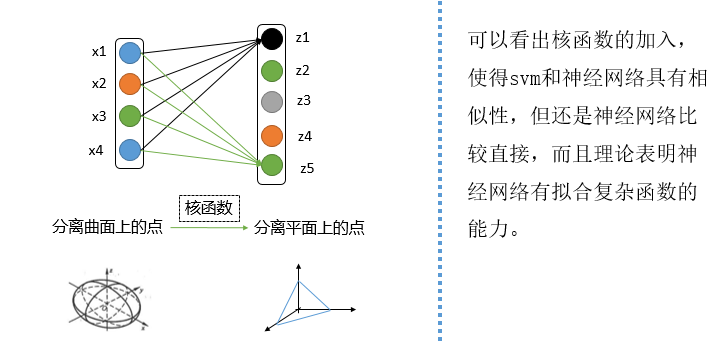
特征空间是前端，输入空间是后端，核函数是控制层

为什么映射函数可以不显式的给出？

因为我们关注的不是特征空间向量的具体形式，而是其内积，而内积有时等于输入空间的某种运算，所以直关注输入向量基于核函数在输入空间的运算结果即可。

怎样根据问题背景选择核函数，不都是用来非线性分类吗，难道是考虑输入空间的分布？目前的想法：假如我们能知道映射函数的表达式，就可以根据映射函数写出核函数。如果不能确定映射，可能就采用实验的方法测试。





**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# 9. EM算法

## 引例

一个袋子中有红球和白球，被抽到的概率分别为p和1-p，放回抽样得到观测序列为：x1,x2,x3...xn,求p的似然估计。

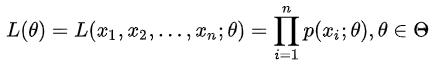
l(p) = log

然后求函数的极值点，可以解出：p=|{xi=红}|/n，也就是说：利用最大似然方法，估计值就等于红色朝上的次数/总的实验次数。

## 极大似然估计

假设一致分布的模型，但是模型的参数不知道，根据观测序列来估计模型的方法，极大似然估计就是极大化观测序列出现概率求模型参数。

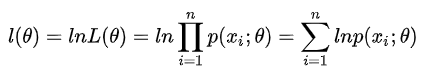
假设观测序列为X，待估计的参数为θ。



目标就是：



然后：



然后求函数的零点即为所求。

## Jensen不等式

定义：如果函数时凹函数(二阶导>=0)，那么E(f(x)) >= f(E(x))。函数的期望大于等于期望的函数。

应用1：KL散度(相对熵)非负



## 9.1 EM算法

EM应用案例：

观测序列：x1,x2,...xn；总的先验参数θ=(π,p,q)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 红色 先验 | 白色 先验 | 红色 联合 | 白色 联合 |
| 袋子1（π） | P | 1-p | πp | π(1-p) |
| 袋子2（1-π） | q | 1-q | (1-π)q | (1-π)(1-q) |

假如我们已知所有的先验参数(π0,P0,q0)，那么可以计算后验概率表为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 后验概率  (条件概率密度) | x1 | x2 | ...... | xn |
| P(袋子1|红色) | =a1 | a2 | ...... | an |
| P(袋子1|百色) | =b1 |  |  |  |
| P(袋子2|红色) | =c1 |  |  |  |
| P(袋子2|白色) | =d1 |  |  |  |
| Sum | 1 | 1 | 1 | 1 |

问题1：x1取自第一个袋子的概率是多少？

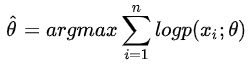
答案1：a1+b1

模型参数的估计值：(xi=1代表实际为红色)

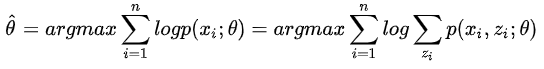
思想：将上一轮的后验作为下一轮的先验。迭代到收敛。

## 9.2 EM理论

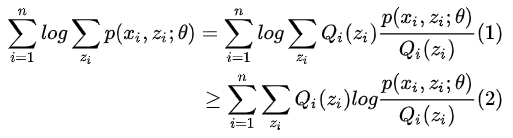
对于n个观测x1,x2,...xn，找出样本的模型参数θ, 极大化模型分布的对数似然函数如下：



加入隐变量后为：

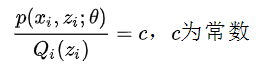


由于包含和的对数以及因变量，所以无法直接给出解析解，尝试jensen缩放：



这在数学上是成立的，至于Q(zi)是什么，后面会分析。

至此，我们找到了一个难解函数的下界，但是下界增大，是原函数增大既不必要也不充分条件，所以有必要进一步增加约束：就是两个函数至少有一个交点，那么这对Q有什么要求呢？根据jensen不等式，相等意味着变量为常数，即：

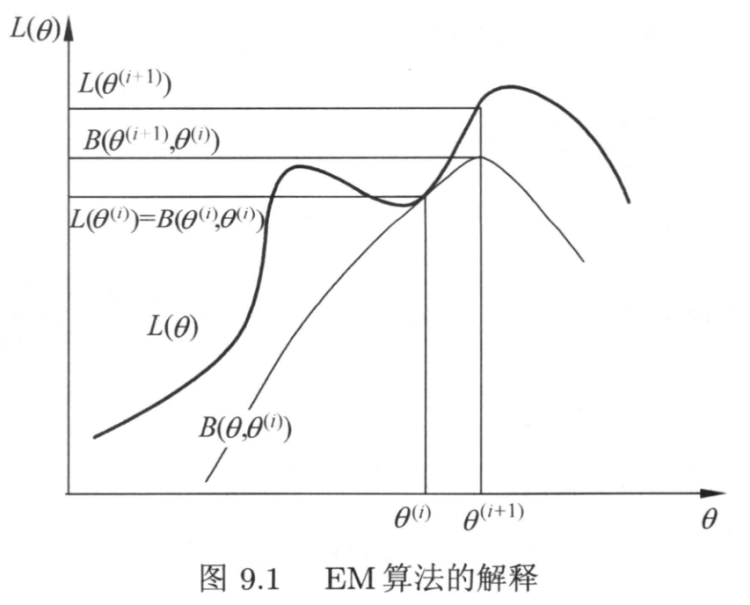




为什么不直接用联合概率？因为不知道zi是什么。**至此，小结一下：**

1、当Q为隐变量的后验概率的时，序列似然概率的下界与之至少交于一点

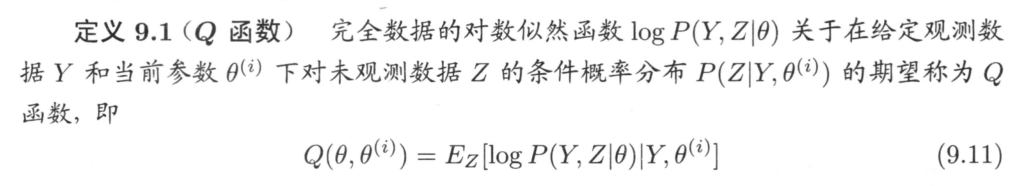
2、下界函数可以直接优化求解

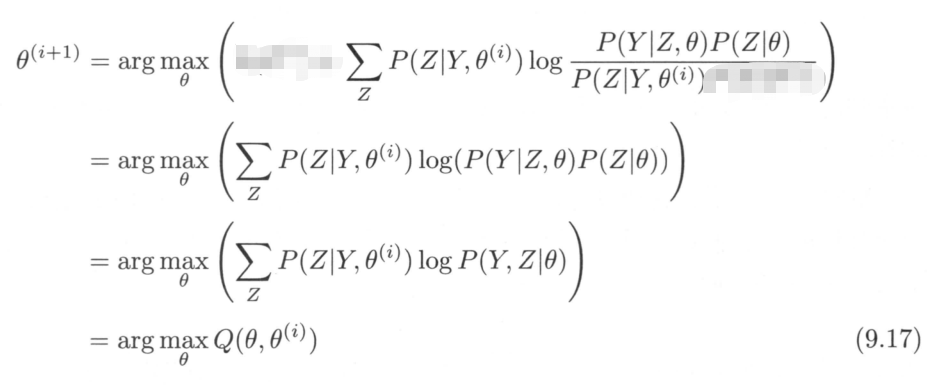


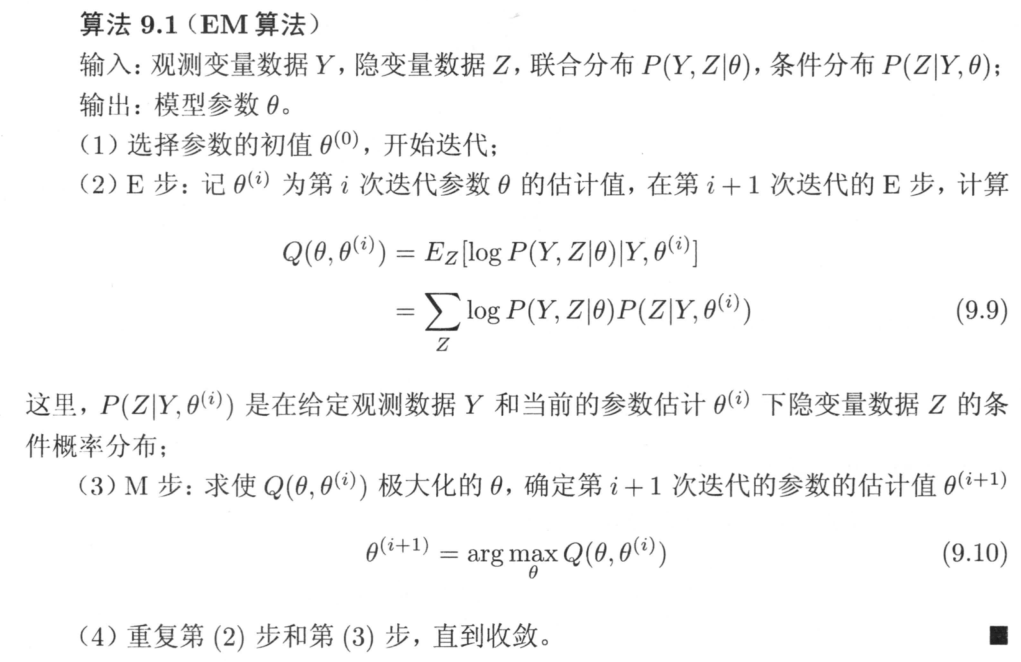
可以看到，下界函数收敛代表EM算法会收敛，单数不一定是极值点，更不用说最值点。

## 9.3 Q函数与EM导出

下面是小蓝书上的，符号有一点不一样，Y对应前面分析的X(观测序列)







最后，理论层融合工程层面，θi+1是什么？

基于公式，求似然函数的极值，这个是可以计算出来的，但是实际应用的时候很简单， 从3.1的案例可以看出，更新的方法就是，逐个的遍历观测序列然后就可以在线更新各个公式的分子分母，便利完毕，参数就更新完毕了。

参考1: <https://zhuanlan.zhihu.com/p/40991784>(EM算法详解)

参考2：《统计学习导论》

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 概率图模型

## 概率图基础

1、将随机变量作为结点，若两个随机变量相关或者不独立，则将二者连接一条边；若给定若干随机变量，则形成一个有向图，即构成一个网络。

2、如果该网络是有向无环图，则这个网络称为贝叶斯网络。

3、如果这个图退化成线性链的方式，则得到马尔科夫模型；因为每个结点都是随机变量，

将其看成各个时刻(或空间)的相关变化，以随机过程的视角，则可以看成是马尔科夫过程。4、若上述网络是无向的，则是无向图模型，又称马尔科夫随机场-马尔科夫网络。

5、如果在给定某些条件的前提下，研究这个马尔科夫随机场，则得到条件随机场。

6、如果使用条件随机场解决标注问题，并且进一步将条件随机场中的网络拓扑变成线性的，则得到线性链条件随机场。

贝叶斯网络总的来说就是两句话：从单一到混合，从有限到无限。单一到混合比较好理解，就直接看上图就好，有限到无限具体指的是空间和时间两者。

隐马尔科夫模型HMM是动态贝叶斯网络，是有向图；马尔科夫随机场，条件随机场是无向图模型。可以通过道德图的方式，把有向图转化为无向图-head2head的话，把两个双亲节点添加连线，其他的情况，直接把尖头去掉。

概率图模型的核心问题：基于高维随机变量求边缘概率或者条件概率。对于高维随机变量，两个基本法则分别对应这两个基本问题：

**加法**（积分）求**边缘概率**：

https://pic3.zhimg.com/80/v2-1f99b4cbadaa82866ea4baa3118b8323_720w.png

乘法求条件概率：

https://pic4.zhimg.com/80/v2-563d4852cc96f327d98a575c1b4acea7_720w.png

根据两个基本法则可以推出两个常用的法则：

**链式法则**：

https://pic3.zhimg.com/80/v2-34f4356a1735f2797818be6dde57676b_720w.png

**贝叶斯法则：**

https://pic4.zhimg.com/80/v2-e86a7c08e96c0fb7d2b964a45bbd5e4c_720w.png

研究**高维随机变量**会造成一个问题—>**计算量巨大,**在面对这样的一个问题是，我们通常都是简化它，**假设他们是相互独立的，彼此之间互不相干**，有一个典型的算法：**朴素贝叶斯算法**，也就是假设各个维度相互独立。

https://pic3.zhimg.com/80/v2-6bd6eb3a5df5f155db6a655296814147_720w.png

这个假设过于强了，因此，需要重新假设，那么就假设**它只与一个状态有关，其他状态无关**（马尔可夫假设）：

https://pic2.zhimg.com/80/v2-62ca1d453915cce7fc19219d4601ec86_720w.png

最后这个假设也比较强，因此重新假设，将所有变量分为三个互不相交的集合A,B,C，这样就得到了条件独立性假设：

https://pic3.zhimg.com/80/v2-ec9e998aa92baed4975b7239ab39ce70_720w.png

至此整个就得到了概率图中两个要素，研究的问题是：高维随机变量，解决办法是：条件独立性假设。整个概率图模型需要解决的问题就是：表示、推断、学习以及决策。

## 2.贝叶斯网络

1.三种基本的拓扑结构：

概率图就是将图赋予了概率定义，可以很直观的根据图结构寻找到概率之间的独立性。根据独立性假设，我们就可以将复杂的计算简化。

链式法则：

https://pic3.zhimg.com/80/v2-34f4356a1735f2797818be6dde57676b_720w.png

条件独立性：

https://pic3.zhimg.com/80/v2-ec9e998aa92baed4975b7239ab39ce70_720w.png

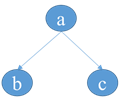
因子分解：

https://pic4.zhimg.com/80/v2-ad986bfe687859e8bfa1bdf07654243a_720w.jpg

其中：pa(i)表示节点i的前驱节点，根据***概率图模型就可以很直观的写出因子分解***，图结构的作用就在于很好的表达条件独立性。首先将箭头符号定义如下：

## 1).tail—tail

当事件a发生的时候，事件b，c被阻塞，两者独立。



根据**因子分解**可以很快写出表达式：

https://pic4.zhimg.com/80/v2-325eedae130d6ed9c38c995be51676ea_720w.png

根据**链式法则**可以得到：

https://pic1.zhimg.com/80/v2-dfd95772b2d3ce285f6cd11f4ee2888e_720w.png

两个公式一对比可以得到：

https://pic2.zhimg.com/80/v2-45142dda083dc13ba1a7dd51669c4ba9_720w.png

这也就意味着**在给定a的前提下，b和c相互独立**。如果觉得不够直观，可以继续推导：

https://pic3.zhimg.com/80/v2-72329a52f6f3121ffc7b3f0cbab05c73_720w.png

这也就直接得到了条件独立的定义：

https://picb.zhimg.com/80/v2-b58c5659a5f661d3a4f48cdf21e53c77_720w.png

根据这样就得出一条规律：在tail-tail（尾—尾）的拓扑结构中，若a被观察，则路径阻塞，也就是说，b和c相互独立。

## 2).head—tail

在b发生的情况下，ac被阻塞，两者是独立的。

https://pic3.zhimg.com/80/v2-55b6f1f51dc0c5ad907e1b96dbf63337_720w.jpg

根据**因子分解**可以写出：

https://pic2.zhimg.com/80/v2-a8a70c56361940d596c51bc192504cd2_720w.png

根据**链式法则**写出：

https://pic1.zhimg.com/80/v2-dfd95772b2d3ce285f6cd11f4ee2888e_720w.png

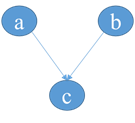
两式对比可得：

https://pic1.zhimg.com/80/v2-3f43bc93cf43f8a848dd8364c874de95_720w.jpg

根据这样就得出一条规律：在head-tail（首—尾）的拓扑结构中，若b被观察，则路径阻塞，也就是说，a和c相互独立。

## 3).head—head

ab本来是独立的，但是c发生的话，两者相遇，不再独立。



根据**因子分解**可以写出：

https://pic2.zhimg.com/80/v2-112956bd2b073180deaff831907c46bd_720w.png

根据**链式法则**可以写出：

https://pic1.zhimg.com/80/v2-dfd95772b2d3ce285f6cd11f4ee2888e_720w.png

两式对比，可以直接得到：https://pic3.zhimg.com/80/v2-8786213c4d1a7e703d79658005485554_720w.png

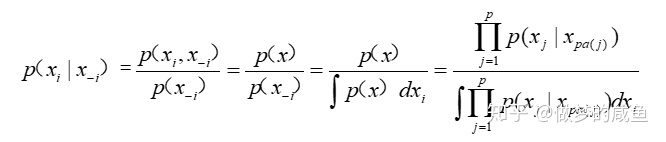
a发生与否对b没有影响，即ab独立。根据这样就得出一条规律：在head-head（首—首）的拓扑结构中，若c未被观察，则路径阻塞，也就是说，a和b相互独立。a和b两者是天然独立的，但是一旦c被观察了，那么两者的独立性就被打破了。当c的后继节点被观察，同样会打破a和b的独立性。

## 3.D划分

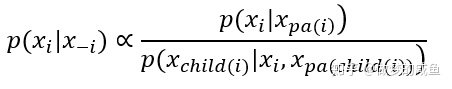
可以看做是一种完备划分，如果AC，在给定B的条件下是独立的。那么所有可以阻塞AC的节点都要划分到B中。D划分其实就是三种基本拓扑节点关系的一个推广，将节点关系推广到集合关系。

假定集合ABC相互之间不相交，若在A和C之间存在路径节点b，节点b和集合B之间需要满足的关系为：1. 当b节点的拓扑关系为tail—tail, head—tail这两种类型是，b节点一定存在于B集合当中；2. 当b节点的拓扑关系为head—head结构时，b节点及其后继节点一定不在B集合当中。在满足这一关系时，称作 D划分的意义在于帮助我们找到了**条件独立性**，这样也就便于简化计算。

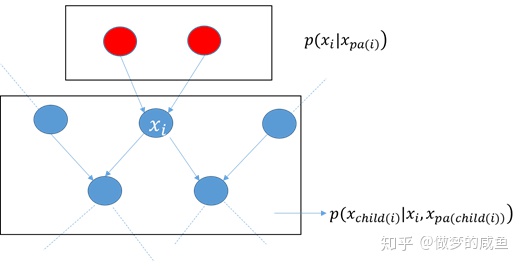
在基于全局节点条件下，求某一个节点的概率问题可以写为：



可以分解为与 Xi有关的和无关的两个部分之乘积，那么对于下面的积分，就可以将与无关的部分提到积分号前，直接约掉，这样就只剩下与有关的部分了。最终可以化为：

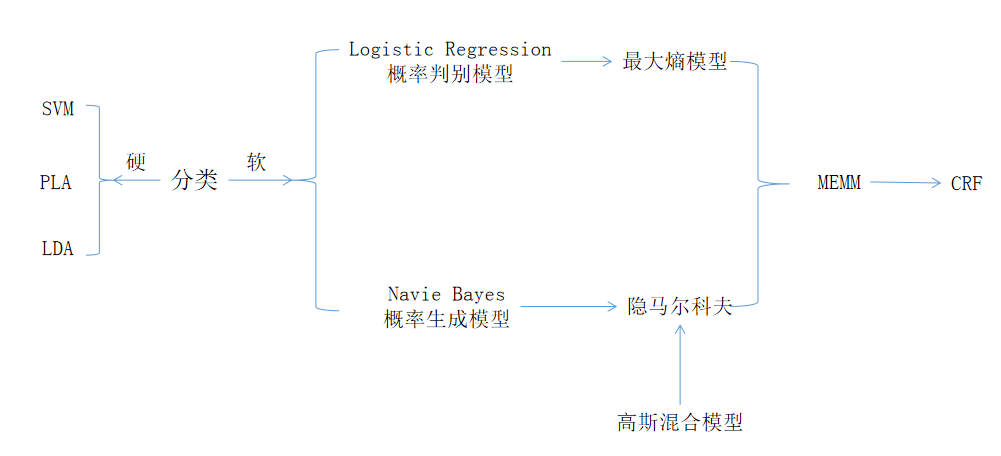


可以用图来表示这种关系，该图又被称为马尔可夫毯：



# 条件随机场

## 背景知识：



分类问题可以划分为硬分类和软分类：

**1、硬分类：**输出即为确切值（对于二分类就是0 or 1）

**SVM（支持向量机）**：最大支持向量的几何间隔驱动的；

**PLA（感知机模型）**：误分类驱动的f(w)=sign(wTx+b)。

**2、软分类：**输出是概率，这里可以再进一步细分为：

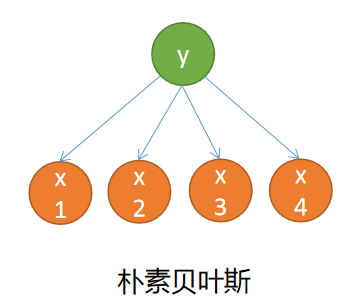
概率生成模型：是对联合概率建模：NB，HMM

概率判别模型：对条件概率建模：LR，ME，MEMM，CRF

逻辑回归（Logistics Regression）/SoftMax Regression。这类问题是对于P(y|x)进行建模的，利用最大熵思想（Maximum Entropy Model）驱动模型。最大熵模型的解是指数族分布。在均值、方差给定的条件下，最大熵模型就是高斯分布。逻辑回归、softmax回归是最大熵模型的特例。

HMM可以看做是NB的推广，也可以看做是高斯混合的推广，GMM中y是离散分布，y->x是高斯分布。MEMM结合ME和HMM模型的优点，是一个判别模型，解决了观测独立性假设带来的问题，但是存在局部归一化带来的标注偏置问题，CRF无向图的全局归一化特点，解决了标记偏置问题。下面具体的分解下各个模型：

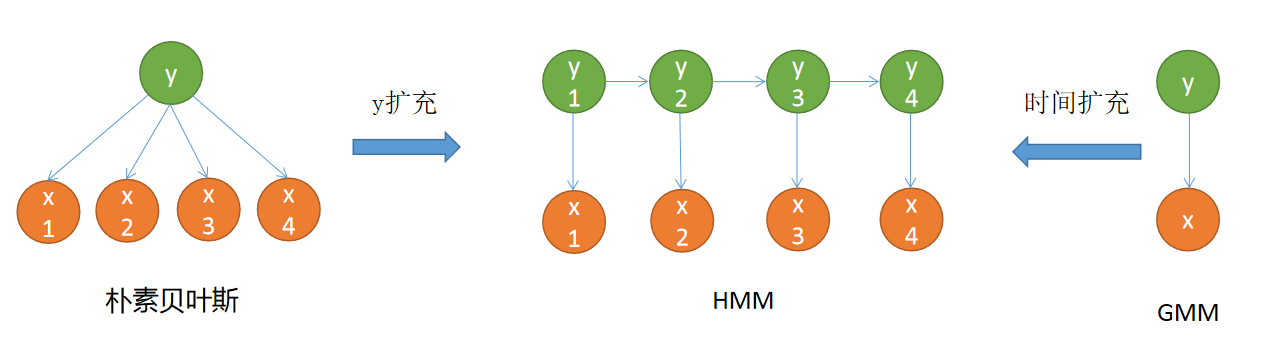
**朴素贝叶斯（Naive Bayes）**，朴素贝叶斯这里之所以叫“朴素”，是因为这里做了一个比较强的假设，即xi⊥xj|y（i≠j），就是在给定y的情况下，xi与xj是相互独立的，因此可以简写NB假设为：P(X|y=1/0) =∏P(xi|y=1/0)，根据这种假设，可以构建tail2tail的概率图结构如下：



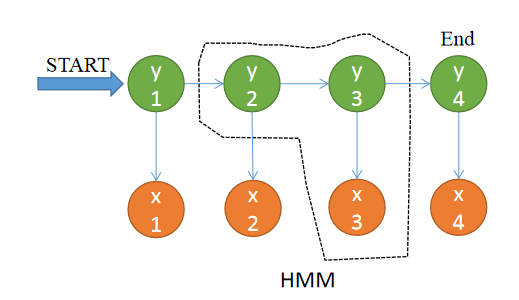
**隐马尔科夫（Hidden Markov Model）**，当朴素贝叶斯模型中的Y从0/1扩展到Seq(序列)，那么，模型便扩展为了隐马尔科夫模型（HMM）。这里面也有2个假设：

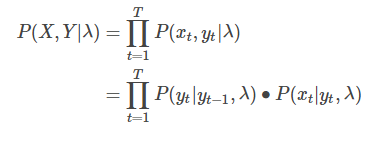
1、齐次一阶Markov假设，所谓一阶，是指当给定yt-1状态的情况下yt与之前的状态之间条件独立，齐次是指不同隐状态之间转移概率服从同一分布即P(yt|yt+1)这个转移概率与时间t无关。

2、观测独立假设；

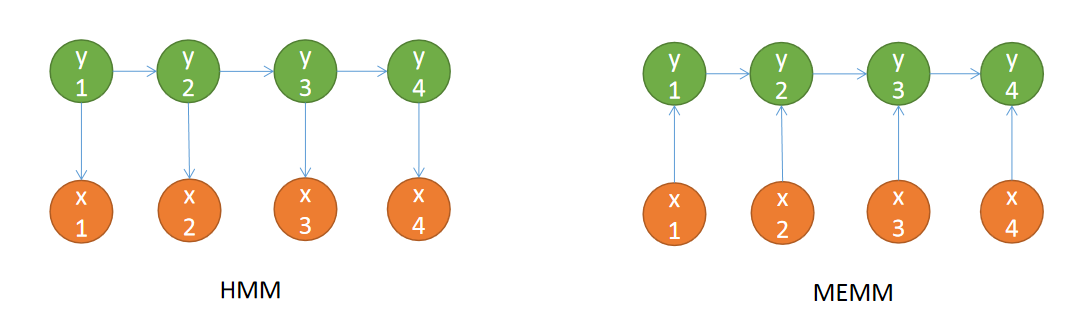


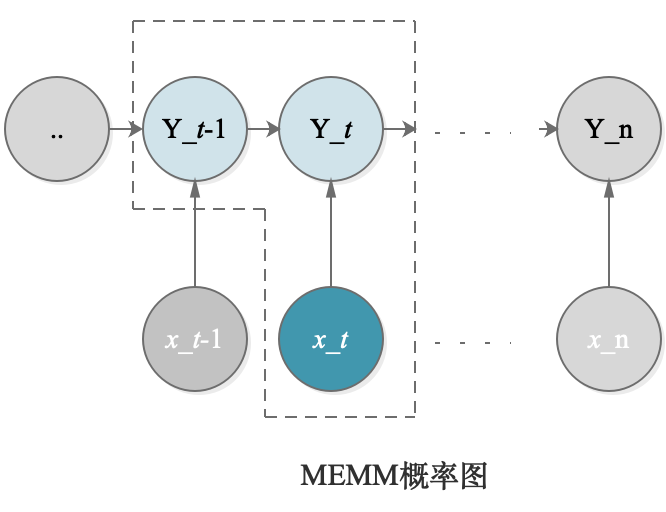
HMM的建模对象为： P(X,Y|λ)——注意：生成模型建模的是**联合概率分布**





**MEMM（Maximum Entropy Markov Model）最大熵马尔科夫模型**。这是结合了最大熵+HMM的优点的一个模型，属于概率**判别模型**。可以直接从概率图看出，每一次确定y的时候，x之间不再独立，实际上y会考虑整个上下文的信息。例如，在确定y2的时候，显然x1和x2都会发挥作用。





**P(Y|X,λ)=∏P(yt|yt−1,x1:T,λ)**

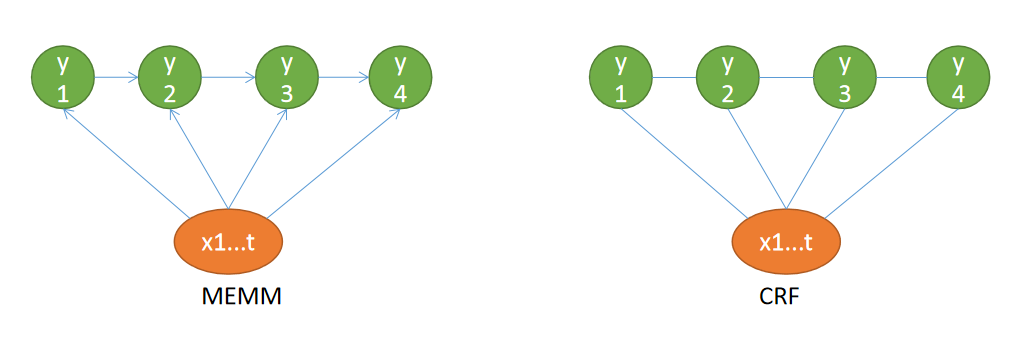
从上面的建模对象可以看出，yt可以看做是一个多分类的逻辑回归，即做了局部的归一化，正是由于这一点，造成了标记偏置问题。HMM和MEMM对比如如下：

1、HMM是生成模型，是对P(X,Y)进行建模；MEMM是判别模型，是对P(Y|X)进行建模；判别模型更容易求解。

2、在HMM中，观测变量是隐变量的输出；在MEMM中，观测变量变成输入了；

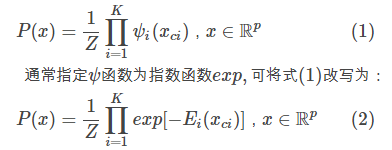
3、HMM中是有观测独立假设的；在MEMM中观测不再独立；

**随机场：**随机场是由若干个位置组成的整体，当给每一个位置中按照某种分布（或者是某种概率）随机赋予一个值之后，其全体就叫做随机场。

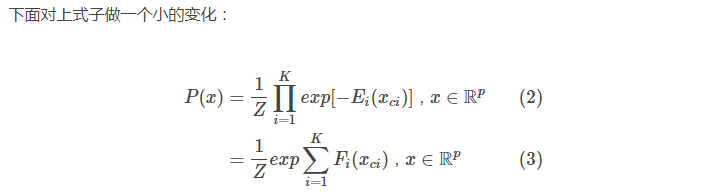


CRF用无向图代替有向图天然的具有全局归一化的特点，解决标记偏置问题。MEMM到CRF，实际上是从动态贝叶斯网络过度到了马尔科夫网络(随机场)。

**MRF(马尔科夫随机场因子分解定理)**：为了简化计算，必须进行因子分解，就像NB。





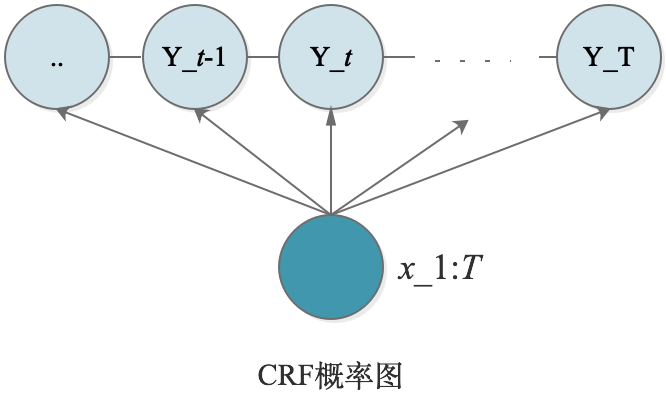


## 条件随机场

CRF是马尔科夫随机场的特例，**它假设马尔科夫随机场中只有𝑋和𝑌两种变量**，𝑋一般是给定的，而𝑌一般是在给定𝑋的条件下我们的输出。这样马尔科夫随机场就特化成了条件随机场。对于CRF，我们给出准确的数学语言描述：设X与Y是随机变量，P(Y|X)是给定X时Y的条件概率分布，若随机变量Y构成的是一个马尔科夫随机场，则称条件概率分布P(Y|X)是条件随机场。

## 线性连条件随机场

注意在CRF的定义中，我们并没有要求𝑋和𝑌有相同的结构。当𝑋和𝑌有相同结构，即：X=(x1,x2,...,xT),Y=(y1,y2,...,yT)这个时候，𝑋和𝑌有相同的结构的CRF就构成了线性链条件随机场。

[](https://raw.githubusercontent.com/anxiang1836/FigureBed/master/img/CRF_local.png)

概率模型和判别模型各有优劣，但是在标注问题上，判别足矣。在MEMM和CRF中，我们经常可以看到两种概率图，一种是线性链结构，一种是XY具有相同结构的线性链，其实无论是MEMM还是CRF都会利用特征函数建模，我觉得这两种模型的区别在于我们根据数据先验，设计的特征函数，在特征函数中，我们可以考虑前一步，也可以考虑整个序列，但是考虑一步的感觉更容易实现。

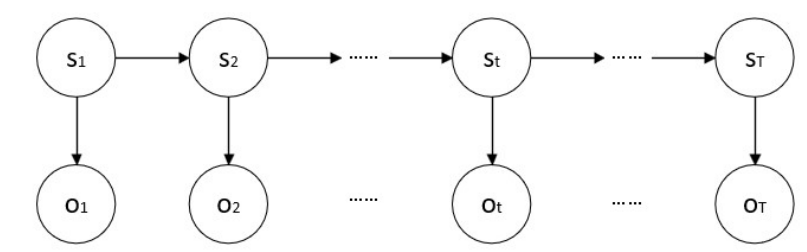
参考：

https://anxiang1836.github.io/2019/11/05/NLP\_From\_HMM\_to\_CRF/

https://zhuanlan.zhihu.com/p/95994825

# 10.HMM-隐马尔科夫模型

具有马尔科夫属性的概率有向图



## 10.1 两个空间三组参数

状态空间：隐状态S的取值范围

观测空间：观测状态O的取值范围

初始概率：初始状态的概率分布，记做π

转移概率：不同状态之间的转移概率，可以用转移矩阵表示，记做a

发射概率：基于当前状态，不同输出的概率分布，记做b

模型参数λ =(a, b, π)

两个假设

齐次假设：即马尔科夫假设

观测独立性假设：观测值只取决于对应的状态值，与其他状态无关

三个问题

概率计算：已知模型参数λ求某个观测序列O的概率

预测：已知模型参数和观测序列O，求最有可能的隐状态序列

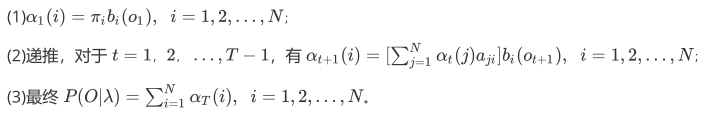
学习：已知观测序列O(或者还要状态序列S)，对参数进行估计

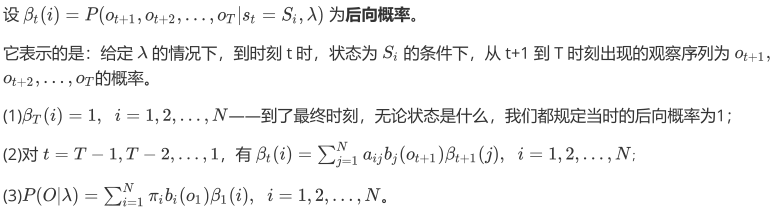
## 10.2、三个问题的计算方法

### 10.2.1概率计算-前、后向算法

两种方法从本质上讲都是**动态规划**，他们可以分别用来求解p(O|λ) 前向概率：下标表示时刻，i表示t时刻的状态，并且观测序列包括t时刻。







后向概率：下标表示时刻，i表示t时刻的状态，但是观测序列不包括t时刻。

**结合前后向算法：**

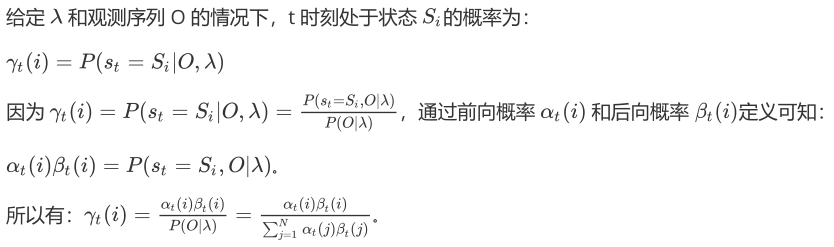


用三者其一都可以计算，而且复杂度是一样的，因为前后向算法都是动规。

### 10.2.2 预测算法

已知观测序列，求最有可能的状态序列

10.3.1拼凑法：求出每个时刻不同隐状态的概率分布，选取概率最大的状态作为S\*(i)，然后将不同时刻的状态拼在一块即可。缺点是：可能不是全局最优解。



分母的理解：加法公式P(A)=P(AB1)+P(AB2)+...+P(ABn)

10.3.2维特比算法

### 10.2.3学习算法

监督学习：当只有观测序列O，用无监督方式，当有状态序列S和观测序列O，用监督方式.基于O/S序列来估计模型参数，采用统计的思想，用频数来逼近频率(包括发射概率、转移概率、输出概率)

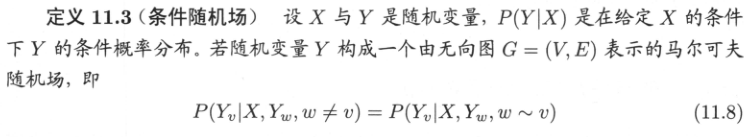
无监督学习：Baum-Welch算法

# 11.条件随机场

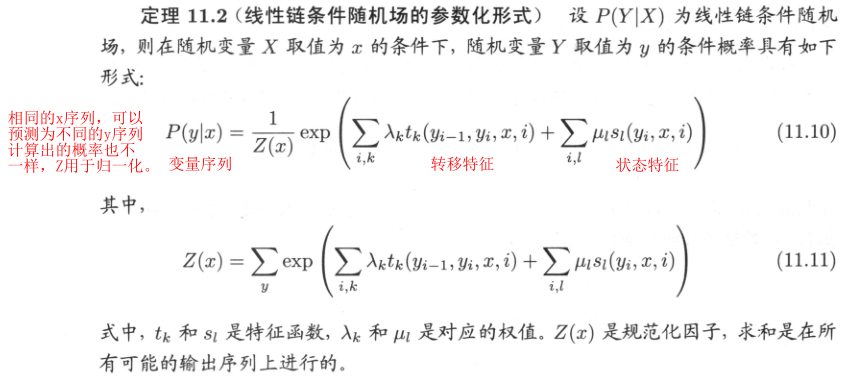
## 11.1. 概率无向图模型的因子分解

将概率无向图模型的联合概率分布表示为最大团上的随机变量函数乘积形式的操作，称为概率无向图模型的因子分解。

## 11.2 条件随机场的定义



### 11.2.2 条件随机场的参数化形式



理解：求已知观测序列x的条件下，状态(标记)序列为y的概率：

1、主题部分为特征函数值的计算

2、以转移特征为例，在每一个时间步，可以计算k个特征值

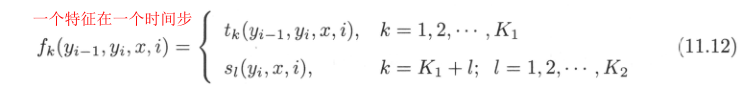
3、对特征值加权求和，在求其指数并归一化即为该条件概率

### 11.2.3 向量形式(时间维求和)

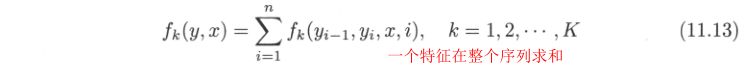
将参数形式的权重和特征函数统一符号描述，为此有：

1、假设K1个转移特征，K2个状态特征，令K=K1+K2

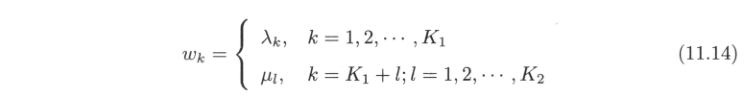
2、统一的特征描述为：



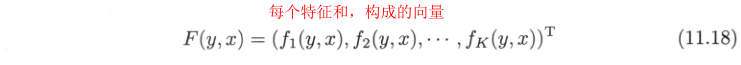
3、一个特征在所有的时间步的斩获(数量值)



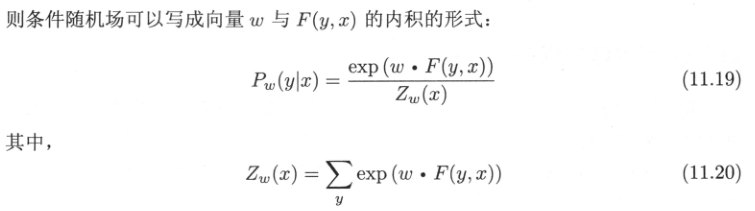
4、权值统一化：



5、特征向量(每个元素，表示一个特征在整个序列上的斩获)



6、内积形式

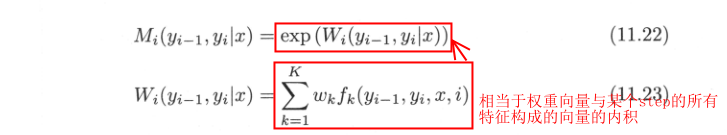


参数形式：求和

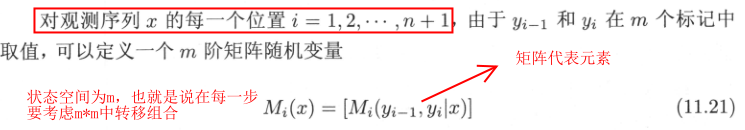
内积形式：内积(等价于求和)

### 11.2.4 矩阵形式(特征维求和)

1、单步特征和：单个时间步基于所有特征的斩获



2、单步矩阵：



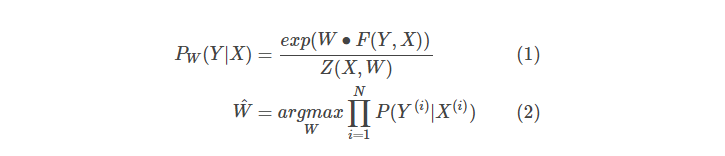
为什么是m阶呢？因为考虑不同的标记组合带入到M(x)中。基于此可以求所有y序列的条件概率

## 11.3 参数学习算法

input: (xi, yi) N对，最大似然函数

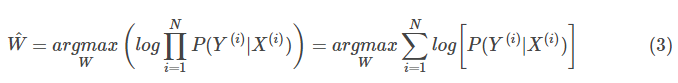
output: 参数λ(特征权重),η(状态权重)

向量形式：

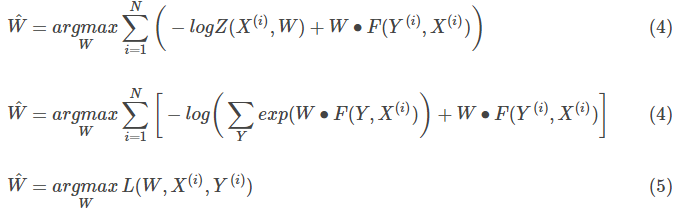


其中，F为K(特征总数)维特征向量，W为K为权重，目标就是最大似然。

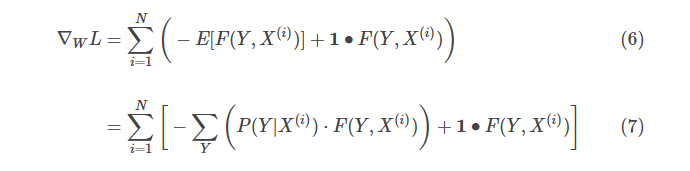
1、对数似然为：



2、带入条件概率，并将归一化因子展开为：



3、梯度计算



由于F为特征函数，第二项可以直接求解。现在考虑第一项z(x)：

将特征按照时间步展开：注意此时f为单步K维特征向量

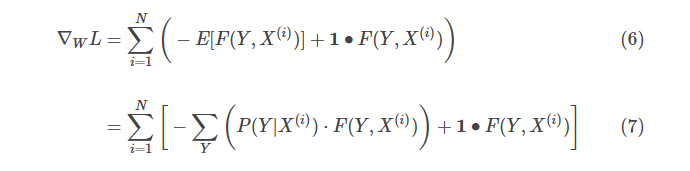
将时间步前提：

将Y拆分：

将pdf通过"积分"化为"边缘概率密度"

由于p可以由前后向算法计算得出，f可以直接计算，至此已经没有变量。

综上：



其中：F(Y,X)=∑tf(yt-1, yt, Xi)

参数更新：

W=step\*

注：求充分统计量的均值用到了：log-partition Function,具体理论来源于指数族分布相关知识。

参考1：<https://www.bilibili.com/video/BV19t411R7QU>

参考2：<https://anxiang1836.github.io/2019/11/05/NLP_From_HMM_to_CRF/#%E7%BA%BF%E6%80%A7%E9%93%BE%E6%9D%A1%E4%BB%B6%E9%9A%8F%E6%9C%BA%E5%9C%BA-Linear-CRF>

## 12实践文档：

### 12.1 真实路径得分

从Bi-lstm中可以得到每一个step标记为某个label的分值，然后相加。就是真实路径的发射得分。从转移矩阵中可以查询出响应的转移分数。

### 12.2所有可能序列的得分

归结为对



的计算，其中s就是序列的发射转移得分。下面举一个例子。

假设我们的观测序列和状态集合为：





lstm输出的发射分数，以及当前学习到的转移概率为：



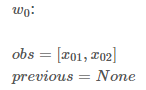


接下来的计算会涉及到两个变量：obs和previous

其中obs：当前step的信息

previous：前一步的计算结果(动态规划计算出的分数)

1、wo step

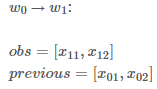


obs是当前step的两个可能状态的发射分数：X01、X02，那么到w0步的总分：



2、w1 step

前向分数：previous



计算到当前step的分数：

假如手写的话就是：

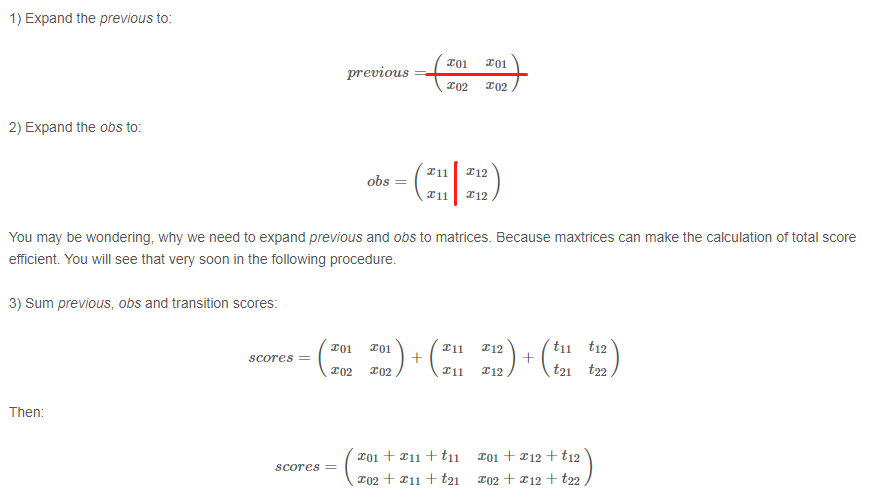
x01+t11+x11 # 前一步 转移 后一步

x01+t12+x12

x02+t21+x11

x02+t22+x12

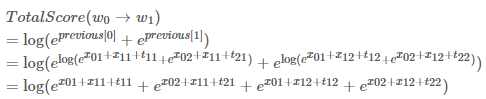
这可以对应处理为矩阵形式如下：

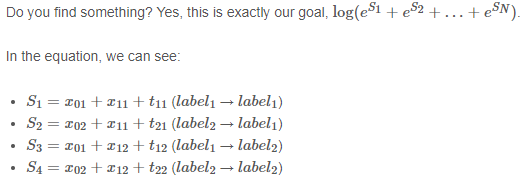


令N为状态集合的长度，则每一个step都要N种可能，T步就是NT。那么到第二步就有4种序列，更新previous为：(以状态1或者状态2结束)



总得分：

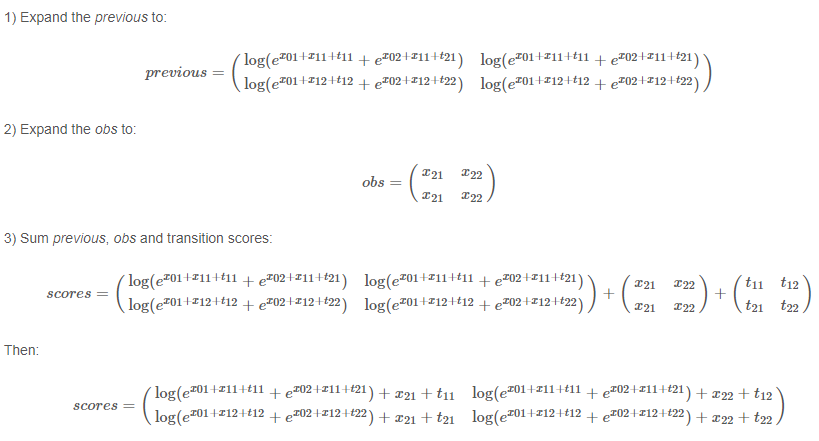


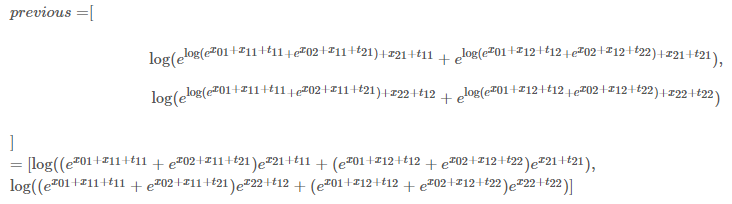


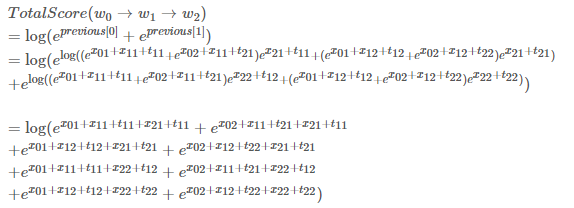
3、 w2 step



前面已经有4个状态序列，所以这一步可以转移为8个状态序列。







所以，哪里体现了动态规划呢？previous中有两项，分别代表到当前步，以状态1、2结尾的分数。虽然看上去很复杂，但是就是两个float。

所以归纳一下就是：previous expand + 转移分数 + 发射分数

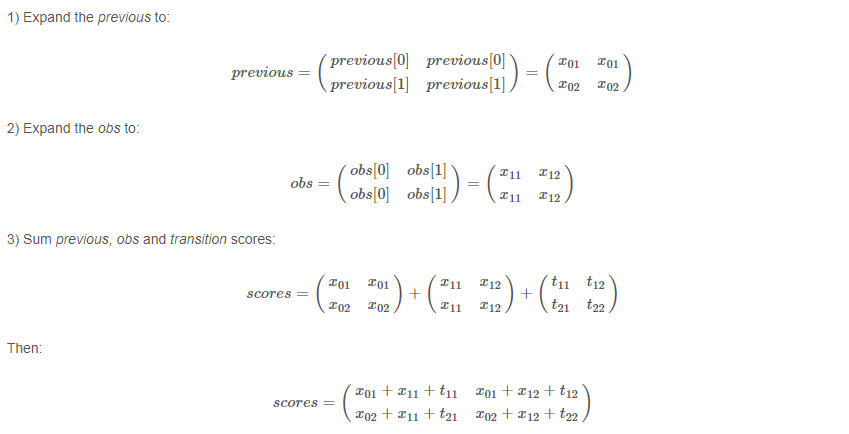
上面算出总分，然后更新当前previous

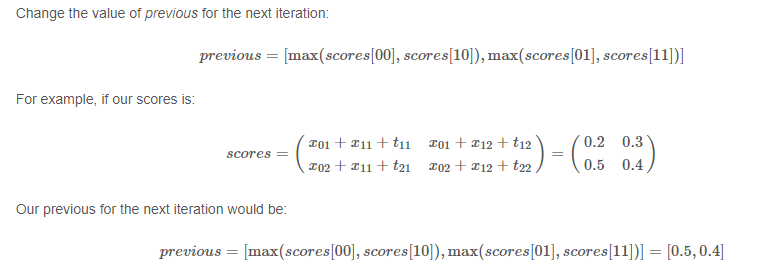
如果是最后一步，可以直接算出总分。

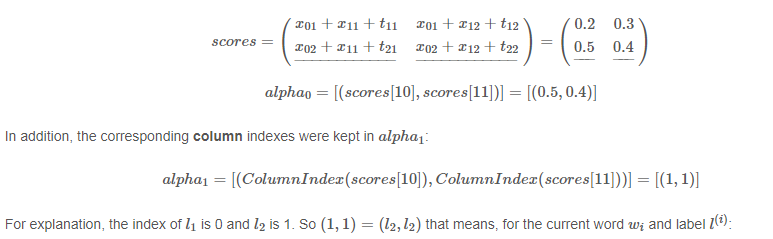
预测算法-维特比

第一步：因为不涉及到转移等，那么最好的label就是发射分数最高的。

第二步：分数求和贪心，并做记忆score&index(行对应label)：







参考：<https://createmomo.github.io/2017/11/11/CRF-Layer-on-the-Top-of-BiLSTM-5/>

<https://zhuanlan.zhihu.com/p/97676647>