Interview-Kit

2022秋招的临阵磨枪

机器学习篇

1. Batch Normalization

基本公式

$$\mu = \frac{1}{m}\Sigma_{i=1}^{m} x_i \sigma^2 = \frac{\Sigma_{i=1}^{m}(x_i-\mu)^2}{m} \hat{x}_i = \frac{x_i-\mu}{\sqrt{\sigma^2+\epsilon}} y_i = \gamma\hat{x}_i + \beta \tag{1}$$

ϵ 是为了防止方差为0，γ 和 β 是可学习参数，为了使BN后的数据仍保留一定的原有特征，因为二者选择的比较好，可以使处理后的数据回归原始数据。

```
def Batchnorm_simple_for_train(x, gamma, beta, bn_param):
"""
param:x    : 输入数据，设shape(B,L)
param:gama : 缩放因子  γ
param:beta : 平移因子  β
param:bn_param   : batchnorm所需要的一些参数
    eps      : 接近0的数，防止分母出现0
    momentum : 动量参数，一般为0.9， 0.99， 0.999
    running_mean ：滑动平均的方式计算新的均值，训练时计算，为测试数据做准备
    running_var  : 滑动平均的方式计算新的方差，训练时计算，为测试数据做准备
"""
    running_mean = bn_param['running_mean']  #shape = [B]
    running_var = bn_param['running_var']    #shape = [B]
    results = 0. # 建立一个新的变量

    x_mean=x.mean(axis=0)  # 计算x的均值
    x_var=x.var(axis=0)    # 计算方差
    x_normalized=(x-x_mean)/np.sqrt(x_var+eps)       # 归一化
    results = gamma * x_normalized + beta            # 缩放平移

    running_mean = momentum * running_mean + (1 - momentum) * x_mean
    running_var = momentum * running_var + (1 - momentum) * x_var

    #记录新的值
    bn_param['running_mean'] = running_mean
    bn_param['running_var'] = running_var

    return results , bn_param

def Batchnorm_simple_for_test(x, gamma, beta, bn_param):
"""
param:x    : 输入数据，设shape(B,L)
param:gama : 缩放因子  γ
param:beta : 平移因子  β
param:bn_param   : batchnorm所需要的一些参数
    eps      : 接近0的数，防止分母出现0
    momentum : 动量参数，一般为0.9， 0.99， 0.999
    running_mean ：滑动平均的方式计算新的均值，训练时计算，为测试数据做准备
    running_var  : 滑动平均的方式计算新的方差，训练时计算，为测试数据做准备
"""
    running_mean = bn_param['running_mean']  #shape = [B]
    running_var = bn_param['running_var']    #shape = [B]
    results = 0. # 建立一个新的变量

    x_normalized=(x-running_mean )/np.sqrt(running_var +eps)       # 归一化
    results = gamma * x_normalized + beta            # 缩放平移

    return results , bn_param
```

2. 梯度消失和梯度爆炸问题

现象描述

梯度消失时，接近于输出层的参数有所更新，而远离输出层，靠近输入层的参数则几乎不变；

梯度爆炸时，一步走的太远，性能变化飘忽不定（猜测）

原因分析

总体原因是网络的反向传播算法需要不断地进行相乘。

- 梯度消失：网络层数过深，小梯度越乘越小；用了不合适的激活函数，比如sigmoid函数（sigmoid的梯度不可能超过0.5）
- 梯度爆炸：网络层数；初始化参数过大

解决方案

- 预训练加微调
- 梯度剪切，防止梯度爆炸
- 换损失函数，用relu及其变体
- BN

$$f_2 = f_1(w^T x + b)\frac{\partial f_2}{\partial w} = \frac{\partial f_2}{\partial f_1}x \tag{2}$$

可以看到，求梯度的时候有一项与输入x有关，而BN消除了x 缩放带来的影响

- ResNet，因为有跨层连接，所以梯度有直通管道，可以增加深度

3. 监督问题

- 监督：有标签
- 无监督：无标签，聚类，降维问题
- 半监督：少量有标签，大量无标签

  伪标签法，用少量标签先训一个模型，然后用该模型去预测无标注的数据标签，构成新的数据集，然后再去训练，以此类推；

  EM算法，元学习
- 自监督：输入就是标签，自编码器

4. domain adaption

迁移学习/元学习，样本迁移，特征迁移，模型迁移

5. 样本不平衡如何处理

- 欠采样，去除一些多数样本
- 过采样，增加小样本的数量；数据增强，增加少数样本，mix-up，旋转，反向，加噪，相位等等
- 加权重，给小样本赋予更多的权重，penalized-SVM
- 集成学习：通过训练多个模型的方式解决数据不均衡的问题，是指将多数类数据随机分成少数类数据的量N份，每一份与全部的少数类数据一起训练成为一个分类器，这样反复训练会生成很多的分类器
- 特征选择，选取好区分的特征

6. KNN相关

K近邻算法

k近邻算法输入的是样本的特征向量，对应特征空间的点，输出是样本所属的类别。k近邻实际上是利用训练数据对特征向量空间进行划分，并将其"分类"的模型；

k近邻模型的三要素是 k值的选取，距离的度量，分类决策规则；

具体而言，对于一个输入样本，选取距离样本最近的k个点，利用这k个点进行类别投票，选择票数最多的类别作为输入样本的类别。

距离度量

$P(x_1, y_1), Q(x_2, y_2)$为例

- 曼哈顿距离：坐标差的绝对值之和

$$dis(P, Q) = |x_1 - x_2| + |y_1 - y_2| \tag{3}$$

- 欧式距离：平方开根号

$$dis(P, Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4}$$

- Lp距离

$$dis(P,Q)=[(x_1-x_2)^p+(y_1-y_2)^p]^{1/p} \tag{5}$$

- 汉明距离：两个等长字符串中不同位置的数目

$$\begin{aligned} x&=x_1,x_2,x_3,\cdots,x_n\\ y&=y_1,y_2,y_3,\cdots,y_n\\ dis(x,y)&=\Sigma_{i=1}^n\delta(x_i-y_i) \end{aligned} \tag{6}$$

- 余弦距离：余弦相似度是通过测量两个向量夹角的度数来度量他们之间的相似度。0度的相似度是1，90度的相似度是0，180的相似度是-1，不是真正的距离，不满足三角不等式。

$$dis(x,y)=\frac{x^Ty}{\sqrt{||x||_2^2||y||_2^2}} \tag{7}$$

- 马氏距离：马氏距离是对特征按照主成分进行旋转，让维度间相互独立，然后进行标准化，让维度同分布之后的欧氏距离。这样可以消除量纲，特征分布分散的影响；由于多维空间中，不同的特征之间可能是相关的，所以单独在各个维度上消除量纲也不行

$$\begin{aligned} dis(x,y)&=\sqrt{(x-y)^TS^{-1}(x-y)}\\ S&=\frac{1}{n}\Sigma(x-\mu)(y-\mu)^T \end{aligned} \tag{8}$$

k值的选择

选取较小的k值，代表模型复杂，因此近似误差会比较小，但是估计误差会增大，容易发生过拟合，被噪声影响

选择较大的k值，代表模型简单，近似误差较大，估计误差较小，容易欠拟合，当k取最大N时，则代表每次的预测结果都是数量最多的类别。

分类决策规则

多数表决规则相当于损失函数为01函数时的经验风险最小化

kd树模型

这里假设k为维度，很简单，对一批数据有$(x_1,\cdots,x_k)$共k个维度的特征，首先对第一个维度，取其中位数，左子树为所有在第一个维度上小于中位数的样本，右子树为所有在第一个特征维度上大于中位数的样本，依次类推，对k个维度划分下去，这样便得到了一棵深度为k+1的二叉树。先后顺序上选择方差更大的维度在前。

当来了一个新样本时，首先找到叶子节点作为“当前最近点”，以新样本为球心，以距离“当前最近点”为半径画球，如果与同级另一颗子树相交，则有更近点，更新“当前最近点”集合，一路递归上去到根节点，就找到了最终的最近点。

7. LR推导

最大似然估计

最大似然估计的基本思想：拥有一组数据的样本观测值，并且已知其含参数概率分布的形式，选取合适的参数估计值，使得样本取到样本值的概率最大。

- 样本

设$(X_1,X_2,\cdots,X_n)$是来自总体的一个容量为n的样本，$(x_1,x_2,\cdots,x_n)$是相应的样本值。

- 离散型总体的似然函数

设分布律为$P(X=x)=p(x;\theta),\theta\in\Theta$的形式已知，$\theta$为待估参数，则样本的观察值为$(x_1,x_2,\cdots,x_n)$的概率为

$$P(X_1=x_1,X_2=x_2,\cdots,X_n=x_n)=\Pi_{i=1}^nP(X_i=x_i) \tag{9}$$

似然函数为

$$L(\theta)=\Pi_{i=1}^np(x_i;\theta) \tag{10}$$

对数似然函数为

$$lnL(\theta)=\Sigma_{i=1}^nlnp(x_i;\theta) \tag{11}$$

- 连续型总体的似然函数

设$Y\sim f(x;\theta)$，θ为待估参数，则上述样本的联合概率密度为

$$L(\theta)=L(x_1,\cdots,x_n;\theta)=\Pi_{i=1}^nf(x_i;\theta) \tag{12}$$

因为样本在样本点附近取值为大概率事件，所以要最大化上述似然函数

对数似然函数为

$$lnL(\theta) = \Sigma_{i=1}^{n} f(x_i;\theta) \tag{13}$$

逻辑回归原理

对于一个二分类问题，我们假设样本服从一个伯努利分布，即$X \sim B(1,p)$，现在有样本$(\mathbf{x_i}, y_i)$，其中，y_i 是类别，只有两种取值，即0和1，我们假设概率p 与样本点的关系如下：

$$p = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} \tag{14}$$

之所以这么选取是因为sigmoid函数的取值为$(0,1)$,则似然函数为

$$L(\mathbf{w}) = \Pi_{i=1}^{n}(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}})^{y_i}(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}})^{1-y_i} = \Pi_{i=1}^{n}(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}})^{y_i}(\frac{e^{-\mathbf{w}^T\mathbf{x}}}{1+e^{-\mathbf{w}^T\mathbf{x}}})^{1-y_i} \tag{15}$$

为了方便，对上述式子取对数

$$lnL(\mathbf{w}) = \Sigma_{i=1}^{n} - y_i * ln(1+e^{-\mathbf{w}^T\mathbf{x}}) + (1-y_i) * (-\mathbf{w}^T\mathbf{x} - ln(1+e^{-\mathbf{w}^T\mathbf{x}})) \tag{16}$$

因为梯度下降需要损失函数减小，所以可以对上述似然函数取负作为损失函数

$$Loss = -lnL(\mathbf{w}) = \Sigma_{i=1}^{n} y_i * ln(1+e^{-\mathbf{w}^T\mathbf{x}}) + (1-y_i) * (\mathbf{w}^T\mathbf{x} + ln(1+e^{-\mathbf{w}^T\mathbf{x}})) = (1-y_i) * \mathbf{w}^T\mathbf{x} + ln(1+e^{-\mathbf{w}^T\mathbf{x}})) \tag{17}$$

求梯度

$$\frac{dLoss}{d\mathbf{w}} = \Sigma_{i=1}^{n}(1-y_i) * \mathbf{x} + \frac{-e^{-\mathbf{w}^T\mathbf{x}}}{1+e^{-\mathbf{w}^T\mathbf{x}}} * \mathbf{x} = \Sigma_{i=1}^{n}(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} - y_i) * \mathbf{x} \tag{18}$$

运用梯度下降

$$w_{t+1} = w_t - \eta * \Sigma_{i=1}^{n}(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} - y_i) * \mathbf{x} \tag{19}$$

运用随机梯度下降

$$w_{t+1} = w_t - \eta * (\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} - y_i) * \mathbf{x} \tag{20}$$

- 决策边界

一般选取0.5为决策边界，即y_i 为1的概率p 超过了0.5即视作属于该类

- 损失函数的选取

在选择损失函数时，应尽量保证函数的凸性，即对一个凸函数求最小值，利用最大似然推导出来的损失函数恰好是凸函数，所以可用，在这里，如果用的是F范数作为损失函数，结果如何?

可以检验一下是不是凸函数，吴恩达说不是

8. 分类问题中的那些率

二分类问题的可能结果

	预测1	预测0	合计
真实1	TP	FN	TP+FN （Actual Positive）
真实0	FP	TN	FP+TN （Actual Negative）
合计	TF+FP （Predicted Positive）	FN+TN （Predicted Negative）	

上面说到，在逻辑回归训练完毕之后，测试时，输入特征$\mathbf{x}$ ，会输出一个概率p， 那么概率到达多少可以作为判断分类为1的依据呢，这是一个值得考虑的问题，选取的概率值可以称之为决策面，决策面的选取导致了第一类错误和第二类错误的发生。

- 第一类错误

漏检：原假设为真，缺排除了原假设，即真实1，预测0

- 第二类错误

虚警：原假设为假，缺认为为真，即真实0，预测1

在战争年代，第二类错误更可怕，容易引起世界大战。

各种率和ROC曲线

- 准确率（ACC）

$$准确率 = \frac{分类正确}{总样本数} \tag{21}$$

常见指标，最容易理解，但是在正负样本不均衡的时候说服力低，比如人群中奢侈品消费人数很少，数据收集过来是小样本，这样训练出来的广告推荐系统可能准确率很高，但对奢侈品销量没有影响，很可能是奢侈品人群数据在样本占比很低，导致输出推荐全为普通人群商品也可以获得很好的准确率。

- 精准率-查准率-精确率(Precision)

$$P = \frac{TP}{TP + FP} \tag{22}$$

预测为正的样本中，有多少比例是真的正

- 召回率，查全率(Recall)

$$TPR = \frac{TP}{TP + FN} \tag{23}$$

正样本数被查出来的比例，所以又叫查全率

- 虚警率(FP)

$$FPR = \frac{FP}{TN + FP} \tag{24}$$

输出为1，但实际为0的样本占所有0样本的比例

召回率和虚警率是对所有预测为正的样本的划分和分析。

- ROC曲线

ROC曲线的纵坐标是召回率，横坐标是虚警率，因此ROC曲线上有四个特殊的点：

（0,1）：所有样本分类正确

（1,0）：所有样本分类错误

（0,0）：所以样本都分类为负

（1:1）：所以样本都分类为正

绘制方法，不断地去移动决策边界，得到一组(recall,fpr)，在图上绘制一个点，从而得到一个阶梯ROC曲线

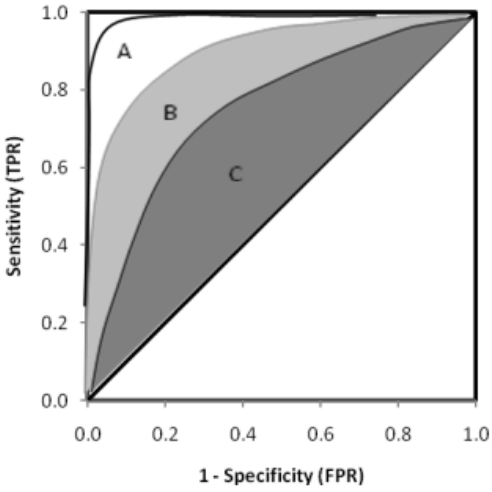
ROC曲线的评估方法

- AUC

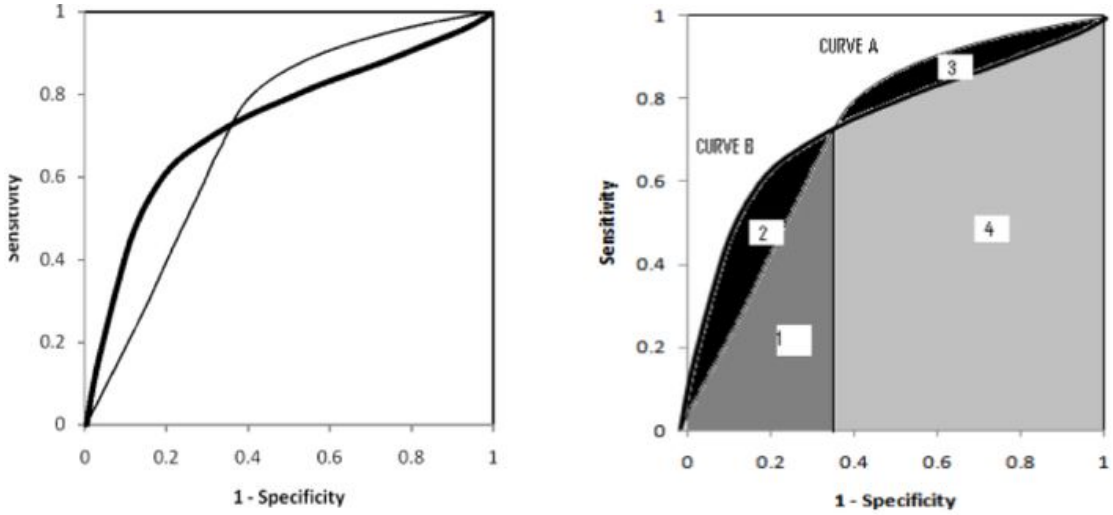
AUC是ROC曲线下的面积，物理意义可以按如下进行理解：**从所有1样本中随机选取一个样本， 从所有0样本中随机选取一个样本，然后根据你的分类器对两个随机样本进行预测，把1样本预测为1的概率为p1，把0样本预测为1的概率为p0，p1>p0的概率就等于AUC**

1. AUC = 1，是完美分类器，采用这个预测模型时，存在至少一个阈值能得出完美预测。绝大多数预测的场合，不存在完美分类器。
2. 0.5 < AUC < 1，优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。
3. AUC = 0.5，跟随机猜测一样（例：丢铜板），模型没有预测价值。
4. AUC < 0.5，比随机猜测还差；但只要总是反预测而行，就优于随机猜测。

不同分类器的ROC曲线无交叠时，AUC面积越大越好

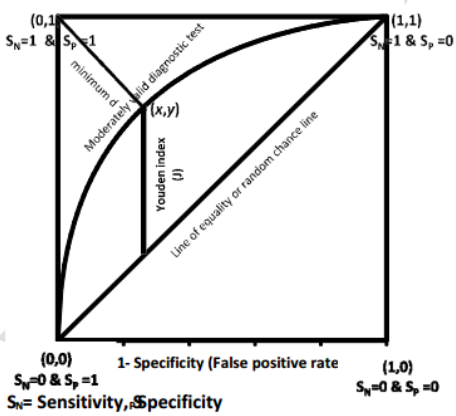


- 曲线比较方法



当AUC面积几乎一样，而ROC又有交叉时，可以根据需要进行选择，需要较高的Recall(sensitivity)，则选择A，需要较低的虚警，则选择B

- 最优临界点



ROC曲线上的最优临界点，让Recall尽量高的情况下，不显著增加TPR，这个点是距离(0,1)最近的点

9.多分类问题的解决模式

逻辑斯谛分布

$$
\begin{aligned}
F(x) = P(X \leq x) &= \frac{1}{1+e^{-(x-\mu)/\gamma}} \\
f(x) = F^{'}(x) &= \frac{e^{-(x-\mu)}}{\gamma(1+e^{-(x-\mu)/\gamma})}
\end{aligned}
\tag{25}
$$

假设样本服从二项逻辑斯谛分布，就是LR，用于二分类问题，假设样本服从多项逻辑斯谛分布，就是mlr，对应的激活函数就是softmax，损失函数就是交叉熵。

10. SVM的推导

线性可分支持向量机

对于一批训练数据$\mathrm{T}=\{(\mathbf{x}_i,y_i)\}_{i=1}^N$，其中$y_i=1(positive);y_i=-1(negative)$，现在想要找到一个最优的决策面，$\mathbf{w}^T\mathbf{x}+b=0$,使得

$$\begin{aligned}\mathbf{w}^T\mathbf{x_i}+b>=0,\ for\ y_i=1\\ \mathbf{w}^T\mathbf{x_i}+b<0,\ for\ y_i=-1\end{aligned}\tag{26}$$

如何利用已有的数据，得到最优的$\mathbf{w}$和b就是SVM的核心思想，定义如下的优化问题：

$$\begin{aligned}&\max\frac{2}{\|\mathbf{w}\|_2^2}\\ &s.t.\mathbf{w}^T\mathbf{x}_j+b\geq 1,\ y_j=1\\ &\quad\mathbf{w}^T\mathbf{x}_k+b\leq -1,\ y_k=-1\end{aligned}\tag{27}$$

这样是把点到直线的距离转化成了直线与直线的距离，最优的决策面仍然为$\mathbf{w}^T\mathbf{x}+b=0$，构造两条平行线作为“支撑向量”的边界，两条平行线之间的距离叫做间隔，上述问题等价于

$$\begin{aligned}&\min\ \frac{1}{2}\|\mathbf{w}\|_2^2\\ &s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1\ ,i=1,\cdots,N\end{aligned}\tag{28}$$

该问题是一个凸二次规划问题，可以直接求解，且具有唯一解，但是为了进一步分析，我们求它的对偶问题，该问题的拉格朗日函数为

$$L(\boldsymbol{\alpha},b,\mathbf{w})=\frac{1}{2}\|\mathbf{w}\|_2^2+\Sigma_{i=1}^N\alpha_i(1-y_i(\mathbf{w}^T\mathbf{x}_i+b))\tag{29}$$

为了求解拉格朗日函数的下确界，对$\mathbf{w},b$求偏导等于0，得到

$$\mathbf{w}=\Sigma_{i=1}^N\alpha_iy_i\mathbf{x}_i,\Sigma_{i=1}^N\alpha_iy_i=0\tag{30}$$

代入原式，得到对偶问题

$$\begin{aligned}\max\inf\ L(\boldsymbol{\alpha},b,\mathbf{w})&=\max-\frac{1}{2}\Sigma_{i=1}^N\Sigma_{j=1}^N\alpha_i\alpha_jy_iy_j{\mathbf{x_i}}^T\mathbf{x}_j+\Sigma_{i=1}^N\alpha_i\\ &=\min\ \frac{1}{2}\Sigma_{i=1}^N\Sigma_{j=1}^N\alpha_i\alpha_jy_iy_j{\mathbf{x_i}}^T\mathbf{x}_j-\Sigma_{i=1}^N\alpha_i\\ s.t.\ &\Sigma_{i=1}^N\alpha_iy_i=0,\\ &\alpha_i\geq 0,i=1,\cdots,N\end{aligned}\tag{31}$$

支持向量

对于线性可分的情况，支持向量有两种定义：

- 训练样本中与分离超平面距离最近的样本点的实例
- 训练样本中对应$\alpha_i^*>0$的点

由凸优化KKT中的互补松弛条件，二者等价。

线性支持向量机

对于一个问题，大部分样本点满足线性可分，少部分点不满足线性可分，我们采用软间隔方法，也称作线性支持向量机，定义优化问题如下：

$$\begin{aligned}&\min\ \frac{1}{2}\|\mathbf{w}\|_2^2+C\Sigma_{i=1}^N\xi_i\\ &s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i\ ,i=1,\cdots,N\\ &\quad\xi_i\geq 0,i=1,\cdots,N\end{aligned}\tag{32}$$

其中，C为超参数，代表对误分类的惩罚，可以证明，上述问题为凸二次规划问题，有最优解，其中$\mathbf{w}$是唯一的，b是一个区间。

上述问题的对偶问题为

$$\begin{aligned}&\min\ \frac{1}{2}\Sigma_{i=1}^N\Sigma_{j=1}^N\alpha_i\alpha_jy_iy_j{\mathbf{x_i}}^T\mathbf{x}_j-\Sigma_{i=1}^N\alpha_i\\ &s.t.\ \Sigma_{i=1}^N\alpha_iy_i=0,\\ &\quad 0\leq\alpha_i\leq C,i=1,\cdots,N\end{aligned}\tag{33}$$

线性支持向量机的支持向量指的是所有在间隔边界上以及其上方的点，对应对偶问题中$\alpha_i^*>0$的点：

- $\alpha_i<C,\xi_i=0$,落在了间隔边界上
- $\alpha_i=C,0<\xi<1$,落在间隔边界和分类边界之间
- $\alpha_i=C,\xi=1$, 落在了分类边界上
- $\alpha_i=C,\xi>1$，落在了分类错误的那边

非线性支持向量机

对于非线性问题，先利用一个非线性变换将输入空间对应到一个特征空间，使得输入空间的超曲面模型在特征空间中变成超平面模型，然后使用线性支持向量机求解。

核函数

设一个输入空间到特征空间的映射为$\Phi(x)$：$\mathcal{X}->\mathcal{H}$,对任意的$x,z\in\mathcal{X}$，函数$K(x,z)=\Phi(x)\Phi(z)$，则$K(x,z)$称为核函数，$\Phi(x)$称为映射函数。由于映射函数比较难以获得求解，在这里主要使用核函数，而核函数的选择较为简单，只要一个定义在$\mathcal{X}\times\mathcal{X}$上的对称函数满足核矩阵半正定即可，核矩阵定义为任意数据输入到对称函数的互相关矩阵。常用的核函数有：

- 线性核

$$K(\mathbf{x},\mathbf{z})=\mathbf{x}^T\mathbf{z} \tag{34}$$

- 多项式核

$$K(\mathbf{x},\mathbf{z})=(\mathbf{x}^T\mathbf{z}+1)^p \tag{35}$$

- 高斯核（最常用）

$$K(\mathbf{x},\mathbf{z})=exp(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}) \tag{36}$$

- sigmoid核

$$K(\mathbf{x},\mathbf{z})=tanh(\beta\mathbf{x}^T\mathbf{z}+\theta) \tag{37}$$

- 拉普拉斯核

$$K(\mathbf{x},\mathbf{z})=exp(-\frac{\|\mathbf{x}-\mathbf{z}\|}{\sigma}) \tag{38}$$

- 字符串核

问题形式

$$\begin{aligned}&\min\ \frac{1}{2}\Sigma_{i=1}^N\Sigma_{j=1}^N\alpha_i\alpha_jy_iy_jK(\mathbf{x_i},\mathbf{x}_j)-\Sigma_{i=1}^N\alpha_i\\&s.t.\ \Sigma_{i=1}^N\alpha_iy_i=0,\\&\quad 0\leq\alpha_i\leq C,i=1,\cdots,N\end{aligned} \tag{39}$$

支持向量机的求解——SMO算法

优化时固定其他变量不变，每次只优化两个变量，使问题变成两变量的二次优化问题，直到求得的解在误差范围内满足KKT条件。

分类问题的方法选取

- feature数目和样本数目差不多，线性可分，逻辑回归或者线性SVM
- 样本数量很多，高斯核运算慢，可以手动增加feature，使其变得线性可分
- 样本数量不算多，但feature少，用SVM+高斯核函数

11.1x1卷积的作用

- 基本原理

1x1的卷积核使得在一张feature map上，所有的特征（元素）等比例的扩大或者缩小：

1. 当channel = 1时，基本没有作用，因为只是所有特征乘了一个系数
2. 当channel>1时，输出的值相当于在channel维度上对所有的特征做了叠加

因此，常用来升维和降维，调整特征图的深度，有一个假设是特征图是过冗余的，所以可以在不丢失信息的前提下通过1x1卷积降维；具体与全连接的关系，在当前的深度学习框架下，对于channel_last的特征图，1x1卷积和全连接没有区别

12. 常用深度学习优化器原理

13. 信息熵、交叉熵、相对熵

- 信息熵

信息熵是随机变量的不确定度的度量

$$H(P)=-\Sigma_{i=1}^np_ilog(p_i) \tag{40}$$

- 交叉熵

用来衡量在给定的真实分布下，使用非真实分布所指定的策略消除系统的不确定性所需要付出的努力的大小

$$H(P,Q) = -\Sigma_{i=1}^{n} p_i log(q_i) \tag{41}$$

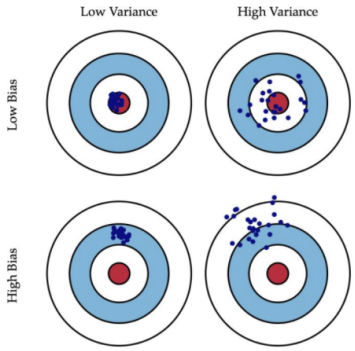
- 相对熵

用来衡量两个分布的差异，其中P是真实分布，Q是估计的分布

$$D(P||Q) = H(P,Q) - H(P) = \Sigma_{i=1}^{n} p_i log(\frac{p_i}{q_i}) \tag{42}$$

一般来说，相对熵是非对称的，即$D(P||Q)! = D(Q||P)$

14. MSE，方差，偏差



对一个估计量而言

- 方差：描述了预测值的离散程度，方差越大，离散程度越高

$$var(\hat{\theta}) = \mathbb{E}((\hat{\theta} - \mathbb{E}(\theta)^2) \tag{43}$$

- 偏差：描述了预测值的准确程度，偏差越大，预测越不准确

$$bias(\hat{\theta}) = \mathbb{E}(\hat{\theta}) - \theta \tag{44}$$

当偏差为0时，是一个无偏估计，对于两个无偏估计而言，方差越小越好，这样估计的结果较为稳定。

- MSE：均方误差，估计值与参数的差的平方取均值

$$MSE = \mathbb{E}((\hat{\theta} - \theta)^2) = var(\hat{\theta}) + bias^2(\theta) \tag{45}$$

15. 聚类方法

原型聚类

K-means

LVQ

高斯混合

密度聚类

层次聚类

16. 降维方法

17. 决策树

基本算法

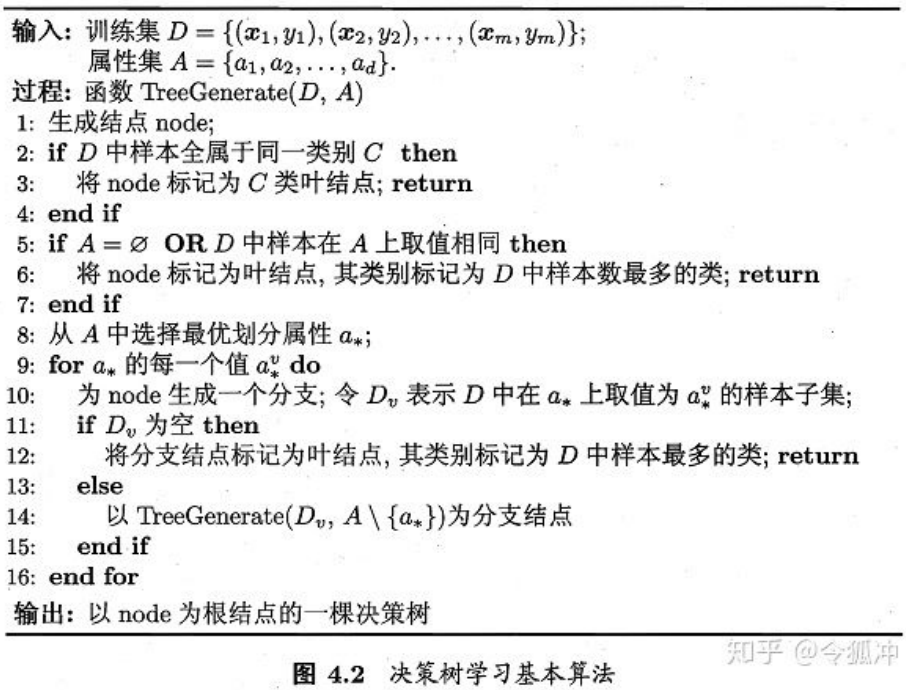


图 4.2 决策树学习基本算法

决策树是递归算法，递归地建立一棵树，有三个递归返回方式：

（1）所有的样本都是同一类C，无需继续划分，标记为C的叶子节点

（2）当前的属性集为空，或者所有样本在所有的属性上都是相同的，此时把当前节点标记为叶子节点，投票选出包含样本最多的类别C作为叶子节点的类别

（3）当前节点包含的样本集合为空，标记为叶子节点，将父亲节点中包含最多的类别C作为当前节点的类别

ID3算法——信息增益法：表示特征对于不确定性减少的程度

在第八步，选最优属性时，有不同的方法，对应了不同的算法。

$$Ent(D)=-\Sigma_{k=1}^{K}p_k log_2(p_k) \tag{46}$$

对于一个离散属性，$a=\{a^1,\cdots,a^V\}$共V个取值，用这些，取值对数据集做一个划分，对任意一个子集D^v,计算一下信息熵$Ent(D^v)$，则信息增益可以进行如下定义：

$$Gain(D,a)=Ent(D)-\Sigma_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v) \tag{47}$$

计算出当前数据集在不同的属性上的信息增益后

$$a^*=argmaxGain(D,a) \tag{48}$$

C4.5算法——信息增益率

C4.5算法选择了信息增益率作为了划分依据，信息增益率定义如下

$$\begin{aligned} Gain\_ratio&=\frac{Gain(D,a)}{IV(a)}\\ IV(a)&=-\Sigma_{v=1}^{V}\frac{|D^v|}{|D|}log_2\frac{|D^v|}{|D|}\end{aligned} \tag{49}$$

通常，a的取值越多，$IV(a)$的值偏向于越大，因此增益率对于可取属性值较少的属性有偏好，实际操作时，可以先选出IV较大的属性，再从中选择增益率最大的属性。

CART——基尼指数

$$Gini(D)=\Sigma_{k=1}^{K}\Sigma_{k'\neq k}p_k p_k'=1-\Sigma p_k^2 \tag{50}$$

直观上看，从一组样本中任取两个样本，二者不一样的概率越小，说明数据纯度越高，对属性D，基尼指数可以定义为

$$Gini\_index(D,a)=\Sigma_{v=1}^{V}\frac{|D^v|}{|D|}Gini(D^v) \tag{51}$$

$$a^*=argminGini\_index(D,a) \tag{52}$$

剪枝处理

剪枝是预防决策树算法过拟合的方法，当分支过多时，很可能出现训练集完全拟合，但测试时效果很差，此时便需要剪掉一些支路简化决策树，提高泛化性。剪枝利用验证集进行。

- 预剪枝

预剪枝是指每划分一个节点时，都计算一下划分前后的收益，即不划分该节点，验证集得到的准确率是a，划分该节点验证集得到的准确率是b，如果a>b，则不进行划分了，本质上是一种贪心算法。

预剪枝使得决策树没有展开，降低了过拟合风险，但由于是贪心，所以很可能不是最优的，类似viterbi译码和贪心译码。

- 后剪枝

后剪枝是对一棵决策树自底向上，对所有的非叶子节点进行考察，看它变成叶子节点能否提升泛化性，可以的话就变。

连续值的处理

连续值进行二分法处理，例如属性a在数据D中出现了n个值，则有一个候选分割点集合

$$T_a = \{\frac{a^i, a^{i+1}}{2}, 1 \leq i \leq n-1\} \tag{53}$$

$$Gain(D,a) = max\ Gain(D,a,t) = max\ Ent(D) - \Sigma\frac{D_t}{D}Ent(D_t) \tag{54}$$

缺失值的处理

其实思路很简单，对于某个特征，先找出没有缺失的数据集$\tilde{D}$,计算一个系数

$$\rho = \frac{|\tilde{D}|}{|D|} \tag{55}$$

然后在$\tilde{D}$上进行计算信息增益，计算完毕后，再乘上系数ρ，以此类推

18. 集成学习和Gbdt

集成学习通过构建并结合多个学习器来完成学习任务。可以证明，对于性能相同的相互独立的多个学习器进行简单投票集成，可以使错误率指数下降趋于0，但是独立这个假设很难达到。

个体学习器之间是串行关系，有强依赖，用Boosting算法集成；个体学习器之间是并行关系，用Bagging/随机森林算法集成。

20. 机器学习和深度学习的对比

- 数据依赖性

深度学习需要的数据更多，在大数据的情况下，达到的性能更好

- 硬件依赖

深度学习需要大量矩阵运算，可以用GPU加速

- 特征处理

机器学习大多是专家选取，深度学习大多是数据中提取的

- 解决方式

深度学习是端到端的，机器学习一般拆解为子问题

- 执行时间

深度学习训练时间很长，执行复杂度较低

- 可解释性差

深度学习很难解释，仅有的理论也只是收敛性上的，很少有能从参数、结构设计上有可以被证明的结论；机器学习的决策树，SVM等算法解释性很强

凸优化篇

0.凸问题

1. 对偶及其理解

原对偶问题

对一个优化问题

$$\begin{aligned} &\min\ f_0(x) \\ &s.t.\ f_i(x) \le 0\ \ for\ i = 1, \cdots, m \\ &\quad\ h_i(x) = 0\ for\ i = 1, \cdots, p \end{aligned} \tag{56}$$

其拉格朗日函数可以写作，

$$\begin{aligned} L = f_0(x) + \Sigma\lambda_i f_i(x) + \Sigma\mu_i h_i(x) \\ g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf L \end{aligned} \tag{57}$$

则对偶问题为

$$\begin{aligned} \max g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ s.t. \boldsymbol{\lambda} \ge 0 \\ dmog = \{(\lambda, \mu) | g(\boldsymbol{\lambda}, \boldsymbol{\mu}) > -\infty\} \end{aligned} \tag{58}$$

对偶性的强度

- 弱对偶

弱对偶 d^*为对偶问题的最优解，p^* 为原问题的最优解，若$d^* \le p^*$，则为弱对偶，$p^* - d^*$为对偶间距。

- 强对偶

强对偶 $d^* = p^*$

强对偶的条件很苛刻，目前已知有三种情况是成立的：

1）凸问题+slater条件成立

2）有可行域的线性规划问题

3）二次约束的二次目标问题+slater成立

- 对偶的几何性解释

2. Slater 和 KKT

3. 凸问题的求解

无约束优化

下降法

最速下降法

梯度下降法

牛顿法

不等式约束优化

对数障碍法

原对偶内点法

通信原理及通信信号处理篇

0. 信息论、信道容量

香农定理

对一个码本$\mathcal{C}$，共有$|\mathcal{C}|$ 个码字，每个码字的长度为N，$\mathcal{C} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{C}|}\}$, 信道可以定义为$P(\mathbf{y}|\mathbf{x})$

发送第i 个码字，检测结果为$\hat{i}$，误码率为$p_e = Pr(i \ne \hat{i})$，定义速率为$R = \frac{1}{N} log|\mathcal{C}|$

香农定理：当R小于信道容量C时，误码率可以任意小，编码方式选择卷积码，Turbo码，LDPC码

信道容量

- 信息熵：$H(X)=-\Sigma P(x)logP(x)$
- 联合熵：$H(X,Y)=-\Sigma\Sigma P(x,y)logP(x,y)$
- 条件熵：$H(X|Y)=-\Sigma\Sigma P(x,y)logP(x|y)$
- 链式法则：$H(X,Y)=H(Y)+H(X|Y)$
- 互信息：发送X，接收Y，$I(X;Y)=H(X)-H(X|Y)=\Sigma\Sigma P(x,y)log\frac{P(x,y)}{P(x)P(y)}$，表示通过接收信号消除的不确定性
- 离散信道容量：$C=maxI(X;Y)$，该优化问题的结果与输入X 的分布有关，对称信道的速率在离散信源等概率时达到最大为$C=log_2|X|-H(\mathbf{p})$
- 微分熵：人为定义的结果，没有信息的不确定性度量的功能，$h(x)=\int_{-\infty}^{\infty}f(x)logf(x)dx$,对于有限分布，均匀分布的微分熵最大，对于无限分布，高斯分布的微分熵最大，微分熵本身没有实际意义，但是微分熵的差可以定义为互信息
- 受输入功率限制的离散时间AWGN信道的容量：

$$Y_i=X_i+N_i \tag{59}$$

可以用大数定律得到，输出矢量$\mathbf{y}$ 落在了以$\mathbf{x}$ 为中心，$\sqrt{n(P+\sigma^2)}$ 的n 维球内，得出容量为

$$R=\frac{1}{2}log_2(1+\frac{P}{\sigma^2}) \tag{60}$$

注意该信号为实信号，如果采用复信号，还需×2

- 带限信道的信道容量，对于一个带宽受限于W 的信道，可以等效于发射波特率为$2W$的AWGN信道，输入功率限制为$P/2W$，噪声功率为$N_0/2$，因此连续时间信道容量为

$$C=2W*\frac{1}{2}log_2(1+\frac{P/2W}{N_0/2})=Wlog_2(\frac{P}{WN_0}) \tag{61}$$

  当$W\to\infty$时，$C=1.44\frac{P}{N_0}$

- 确定性MIMO系统的信道容量：$\mathbf{y}=\mathbf{Hx}+\mathbf{w}$,互信息为$\mathcal{I}=log|\mathbf{I}+\mathbf{HR}_x\mathbf{H}^H\mathbf{R}_w^{-1}|$,求解优化问题使互信息最大为信道容量，对$\mathbf{H}$ 做SVD分解，最后可以化为子信道的注水定理优化问题，用KKT求解，当信噪比比较大或者发射端未知信道时，对信号进行等功率分配，信道容量为

$$C=\Sigma_{i=1}^{r}log_2(1+\frac{P\lambda_i^2}{N_TN_0}) \tag{62}$$

- 随机MIMO系统的信道容量，采用遍历容量，如果信道是遍历的，则$\bar{C}=E\{C\}$,如果信道不是遍历的，则采用中断容量去描述，$P_{out}(R)=Pr(C(H)<R)$,即传输R速率的的数据，不能实现任意小的误码率的临界点

1. 无线通信的数字基带传输模型

基带传输模型

一个发射信号，具有如下的形式：

$$s(t)=\Sigma_{n=0}^{\infty}s_ng(t-nT) \tag{63}$$

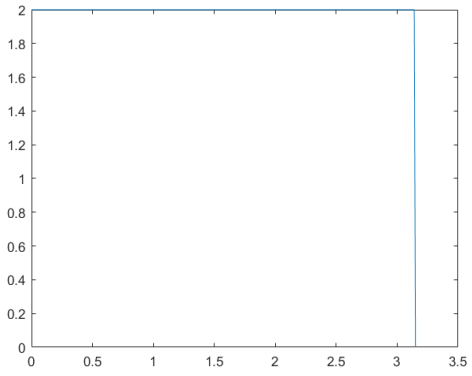
其中，s_n为调制符号，$g(t)$为脉冲波形，且是一个带限信号$|f|\le W$，同时，信道的频率响应也是带限的，则接收信号：

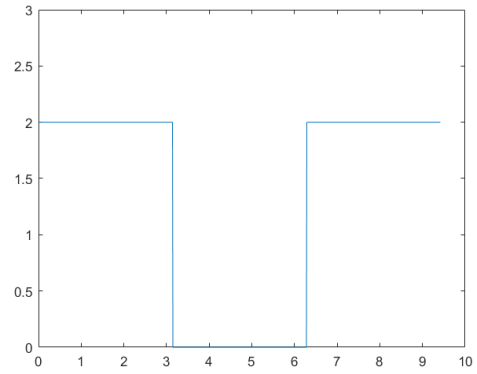
$$r(t)=\Sigma_{n=0}^{\infty}s_ng(t-nT)*h(t)+z(t) \tag{64}$$

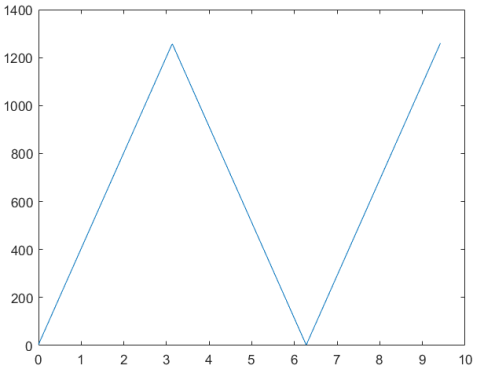
经过接收端匹配滤波后，

$$x(t)=g^*(-t)*h(t)*g(t) \tag{65}$$

$$y(t)=\Sigma_{n=0}^{\infty}s_nx(t-nT)+\mu(t) \tag{66}$$







以$g(t)$为一个矩形波信号为例，经过接收端的匹配滤波之后，变为一个三角波信号，此时，需要对接收的三角波信号进行定时采样，使其恰好采在三角波的尖上，对应了最大的接收信噪比：

$$y(kT) = \Sigma_{n=0}^{\infty} s_n x(kT - nT) + \mu(kT) \tag{67}$$

则等价于

$$y_k = \Sigma_{n=0}^{\infty} s_n x_{k-n} + \mu_k \tag{68}$$

所以，

$$y_k = s_k + \Sigma_{n=0,n\neq k}^{\infty} s_n x_{k-n} + \mu_k \tag{69}$$

则第一项为希望接收的符号，第二项是信道效应引起的符号间干扰，第三项是噪声,下面进一步分析x_k的结构，对于无线信道

$$h(t) = \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \delta(t - \tau_l) \tag{70}$$

所以

$$x(t_\delta - nT) = g^*(-t) * h(t) * g(t) = \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \int_0^{\infty} g(t - nT - \tau_l) g(t + t_\delta) dt \tag{71}$$

$$\begin{aligned} x(kT - nT) = g * (-t) * h(t) * g(t) =& \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \int_0^{\infty} g(t - nT - \tau_l) g(t + kT) dt \\ =& \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \int_0^{\infty} g((k-n)T - \tau_l) g(t) dt \\ =& \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \int_0^{T} g((k-n)T - \tau_l) g(t) dt \end{aligned} \tag{72}$$

即

$$x_{k-n} = \Sigma_{l=0}^{L} a_l e^{-j2\pi f \tau_l} \int_0^{T} g((k-n)T - \tau_l) g(t) dt \tag{73}$$

可以证明，此时x_k是一个有限长的序列，长度为L，所以

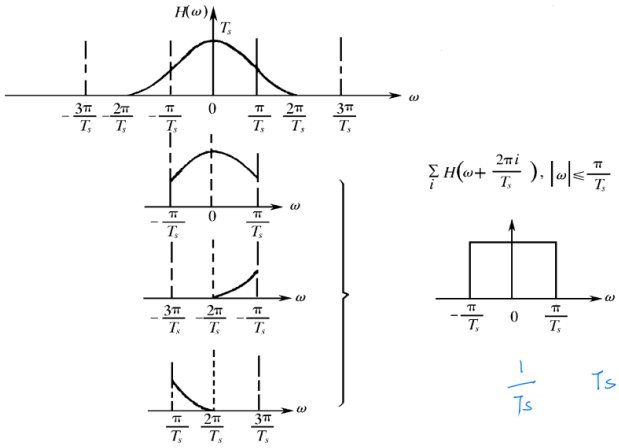
$$y_k = s_k + \Sigma_{l=0,l\neq k}^{L} s_l x_{k-l} + \mu_k \tag{74}$$

第一项为希望接收的符号，第二项是信道效应引起的符号间干扰，第三项是噪声。注意，以上系统均考虑的是基带系统，发射信号为复信号，信道为复信道，实现方式很简单，IQ调制即可。

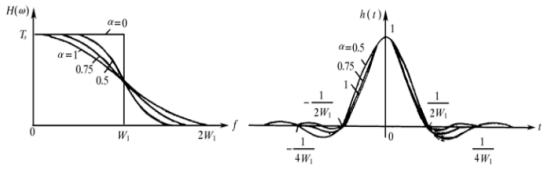
无符号间干扰的传输特性

从上述式子可以看出，为了使符号间没有干扰，我们需要保证，在抽样点k处有，$x_0=1,x_k=0(k\neq 0)$，该条件被称作奈奎斯特第一准则，频域条件是使得总传输特性满足

$$\Sigma_{-\infty}^{\infty}X(f+m/T)=T \tag{75}$$



满足的情况为理想低通特性和升余弦滚降两种，根据采样定理的搬移特性，对于$|f|\leq W$的总传输特性，传输的波特率最大为$2W$，即码元宽度最小为$1/2T$，但是理想低通特性是非因果且不可实现的，所以要设计升余弦滚降特性，此时最大的波特率为$2W/(1+\alpha)$



- **当α=0时，即为前面所述的理想低通系统；**
- **当α=1时，即为升余弦频谱特性，这时$H(\omega)$可表示为**

$$H(\omega)=\begin{cases}\frac{T_s}{2}(1+\cos\frac{\omega T_s}{2}), & |\omega|\leq\frac{2\pi}{T_s}\\ 0, & |\omega|>\frac{2\pi}{T_s}\end{cases}$$

2. OFDM

上文讲到，在带宽为W 的情况下，最大的波特率为$2W$，因此要实现更大的波特率需要更大的带宽，此时如果依然使用单载波系统，信道由多径效应引起的频率选择性衰弱更强了，引起了码间串扰，为了消除码间串扰，需要更复杂的均衡器，一般由FIR滤波器实现，需要的阶数更高，成本更高，不划算且难以实现，OFDM系统是用来解决上述单载波系统码间串扰的一种方法，提高系统的有效性和可靠性。

模拟OFDM

- 正交性原理

对于任意两个函数$S_1(t),S_2(t)$，如果$\int_0^T S_1(t)S_2^*(t)dt=0$，则二者在$(0,T)$上正交，可以证明，复指数信号$\{e^{j2\pi\frac{kt}{T}}\}_{k=0}^{N-1}$是一组正交函数,有

$$\frac{1}{T}R(k,i)=\begin{cases}1,k=i\\0,k\neq i\end{cases} \tag{76}$$

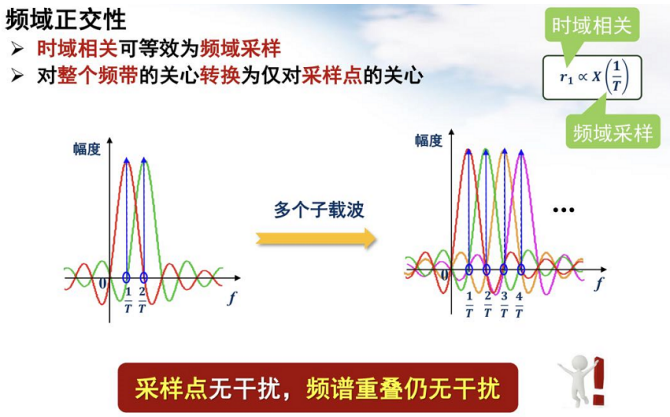
对一组正交载波的叠加传输有

$$x(t)=a_1sin(wt)+a_2sin(2wt) \tag{77}$$

分离时，由于sin函数的正交性，采用做相关的方式进行恢复

$$\begin{aligned}\int_0^T x(t)sin(wt)dt&=a_1\\ \int_0^T x(t)sin(2wt)dt&=a_2\end{aligned} \tag{78}$$

而又有时域相关的结果等效为频域采样，因此可以将对整个频带的关心转化为对采样点上的值的关心。



相比于FDMA，对频带的利用率更高了，模拟OFDM系统如下：

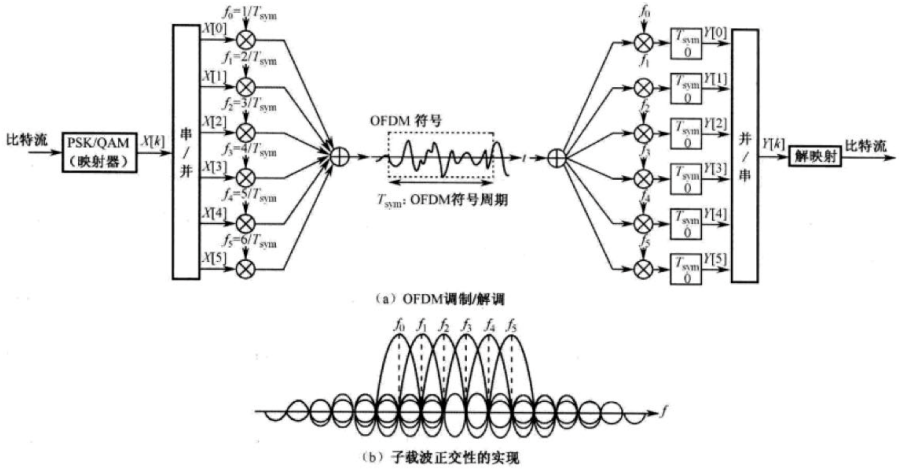


图 4.9　OFDM调制和解调的示意性框图：N=6

假设一个OFDM符号为T_{sym},则子载波带宽为$\frac{1}{T_{sym}}$,传输了N个符号，占用带宽$W=\frac{N}{T_{sym}}$,对带通系统，波特率为$B=\frac{N}{T_{sym}}$,等效的符号时间$T_s=\frac{1}{B}=\frac{T_{sym}}{N}$，所以有$T_{sym}=NT_s$的传闻，但这个概念只存在于模拟系统。

上面相当于用一个较宽的OFDM符号，持续周期T_{sym},实现了$\frac{N}{T_{sym}}$ 的波特率，增加了有效性的同时提高了可靠性，但是还是无法消除OFDM符号间的干扰，消除方式在数字OFDM里解释。

数字OFDM

$X_l[k]$表示第l个OFDM符号的第k个子载波上的发送符号，则总的OFDM基带信号为

$$x(t)=\Sigma_{l=0}^{\infty}\Sigma_{k=0}^{N-1}X_l[k]e^{j2\pi f_k(t-lT_{sym})} \tag{79}$$

在时刻$t=lT_{sym}+nT_s, T_s=T_{sym}/N, f_k=k/T_{sym}$ 进行采样，有

$$x_l[n]=\Sigma_{k=0}^{N-1}X_l[k]e^{j2\pi kn/N}, n=0,\cdots,N-1 \tag{80}$$

可以看到，$\{x_l[n]\}_{n=0}^{N-1}$恰好为$\{X_l[k]\}_{k=0}^{N-1}$的N点IDFT，可以用FFT算法快速实现，因此可以采用数字OFDM的方式，$x_l[n]$为一个OFDM符号。

得到OFDM符号后，还在数字域，要想实现传输，需要将其变到模拟域

$$s(t)=\Sigma_{n=1}^{N}x[n]g(t-nT_s) \tag{81}$$

其中，$g(t)$为成型脉冲，同理，在接收端，利用DFT，可以把时域信号变回频域信号进行信号检测，数字OFDM系统的组成如下：

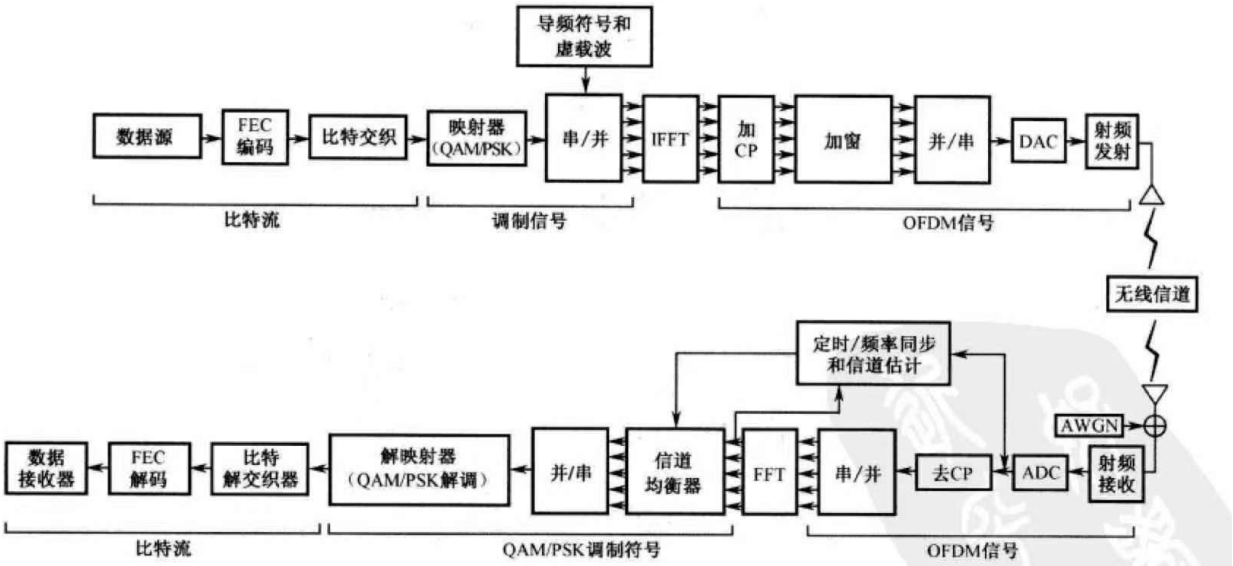
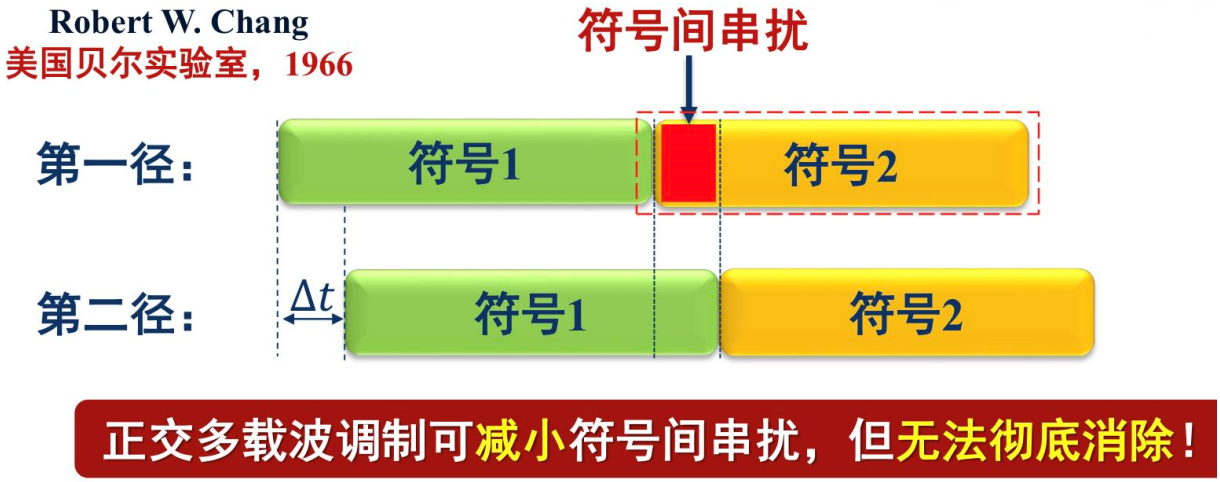


图 4.10　OFDM系统的发射机和接收机框图

OFDM收发机关键技术

- ISI和ICI的消除

通过前面对模拟OFDM和数字OFDM系统的分析，可以看到，通过OFDM的方式，在同等的波特率下，OFDM的时域符号更长，因此可以一定程度上减轻符号间干扰的问题，但是只要多径依然存在，必然还是会出现符号间干扰，即ISI。



为了消除ISI，一种直接的想法就是在每个符号之间加入保护间隔，即每个符号之间加入一段空白，但这样会破坏子载波间的正交性，引起载波间干扰(ICI)。

3. MIMO系统