

ASR实战

第一课 语音识别(ASR)基础

第一课 语音识别(ASR)基础

- 知识点1：什么是ASR：介绍语音识别应用场景、技术发展及难点
- 知识点2：基础模型框架： HIERARCHICAL MODEL FOR SPEECH RECOGNITION
- 知识点3：声音的基本处理：声音信号处理， 以及声音特征提取(频谱图， F-BANK特征， MFCC特征等)
- 知识点4：HMM基础：隐马尔可夫模型原理介绍， 以及前向后向算法
- 知识点5：GMM基础：高斯混合模型原理介绍

第二课 声学模型 GMM/DNN-HMM

- 知识点1：单音素声学模型：MONOPHONE GMM-HMM语音模型
- 知识点2：基于EM的模型训练
- 知识点3：考虑上下文的多音素声学模型：CONTEXT-DEPENDENT TRIPHONE GMM-HMM
- 知识点4：神经网络DNN原理
- 知识点5：从传统GMM-HMM到经典CD-DNN-HMM语音模型

第三课 语言模型与解码对齐

- 知识点1：语言模型概述（LEXICON AND LANGUAGE MODEL）：N-GRAM, WORD2VEC, EMBEDDING
- 知识点2：解码与对齐（DECODING, ALIGNMENT）：从孤立词识别到连接词/词序列之CONNECTED WORD RECOGNITION与TIME ALIGNMENT
- 知识点3：利用WFST (WEIGHTED FINITE STATE TRANSDUCERS)实现语言模型计算：WFST介绍、WFST基本操作：COMPOSITION/DETERMINISATION/MINIMISATION、WFST在ASR中的应用：HCLG、基于WFST的BEAM SEARCH

什么是ASR

- 自动语音识别技术(AUTOMATIC SPEECH RECOGNITION, ASR)是一种将人的语音转换为文本的技术。语音识别作为一个多学科交叉的领域，它与声学、语音学、语言学、数字信号处理理论、信息论、计算机科学等众多学科紧密相连。
- 语音识别近年来受关注度不断提升，相关技术广泛用于家用电器和电子设备，如智能音箱、声控遥控器，移动应用上的各种声控操作、语音助手等；也可用于个人、呼叫中心，以及电信级应用的信息查询与服务等领域。

什么是ASR

- 应用场景

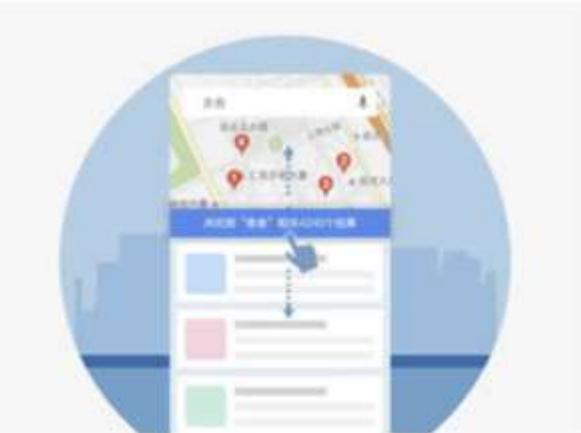
语音工具



语音输入法



语音助手



语音导航

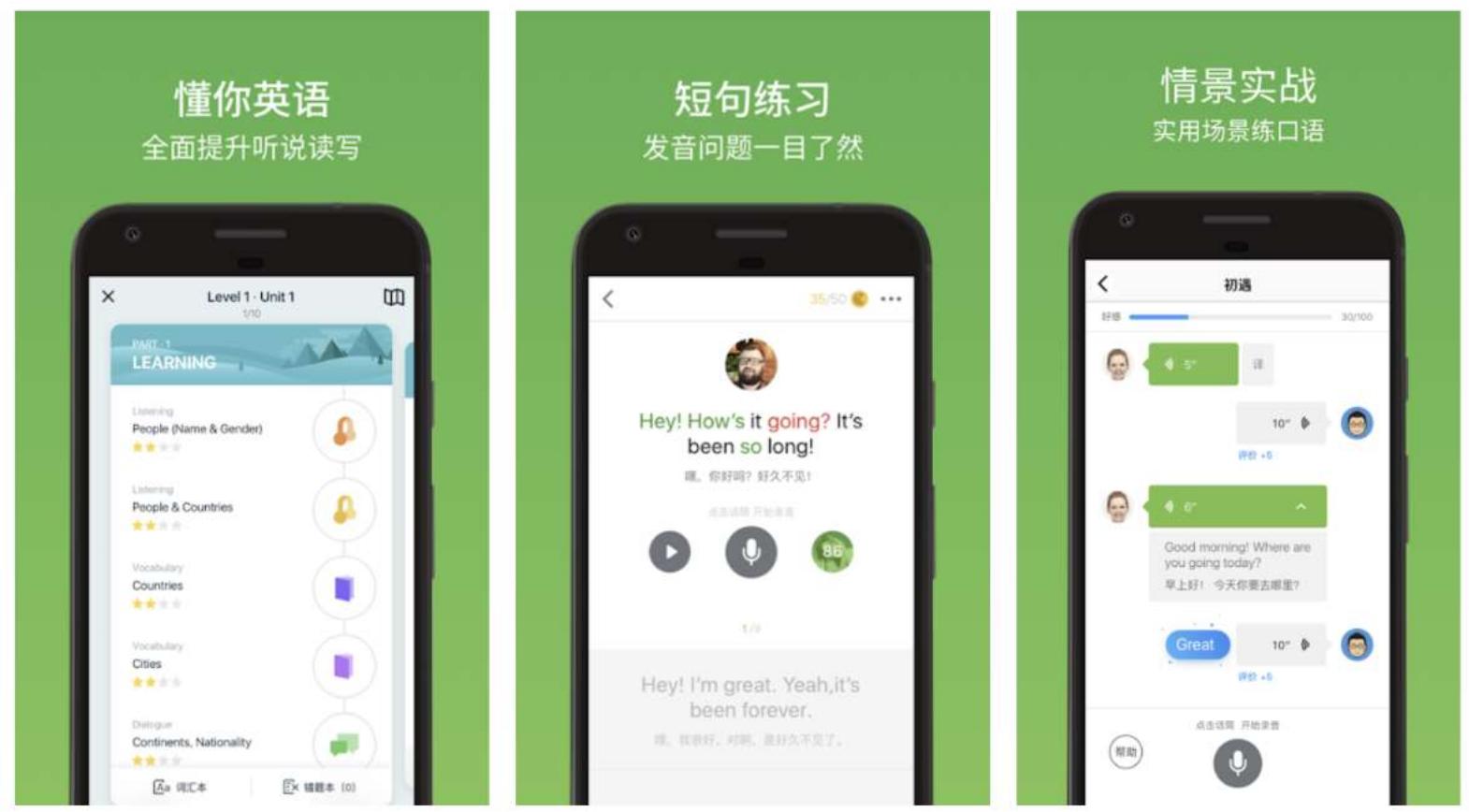


聊天机器人

什么是ASR

语言教学

- 应用场景



什么是ASR

智能客服

- 应用场景



第一课 语音识别(ASR)基础

- 知识点1：什么是ASR：介绍语音识别应用场景、技术发展及难点
- 知识点2：基础模型框架： HIERARCHICAL MODEL FOR SPEECH RECOGNITION
- 知识点3：声音的基本处理：声音信号处理， 以及声音特征提取(频谱图， F-BANK特征， MFCC特征等)
- 知识点4：HMM基础：隐马尔可夫模型原理介绍， 以及前向后向算法
- 知识点5：GMM基础：高斯混合模型原理介绍

基础模型框架

HIERARCHICAL MODEL FOR SPEECH RECOGNITION

Fundamental Equation of Statistical Speech Recognition

If \mathbf{X} is the sequence of acoustic feature vectors (observations) and \mathbf{W} denotes a word sequence, the most likely word sequence \mathbf{W}^* is given by

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} P(\mathbf{W} | \mathbf{X})$$

Applying Bayes' Theorem:

$$P(\mathbf{W} | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{W}) P(\mathbf{W})}{p(\mathbf{X})}$$

$$\propto p(\mathbf{X} | \mathbf{W}) P(\mathbf{W})$$

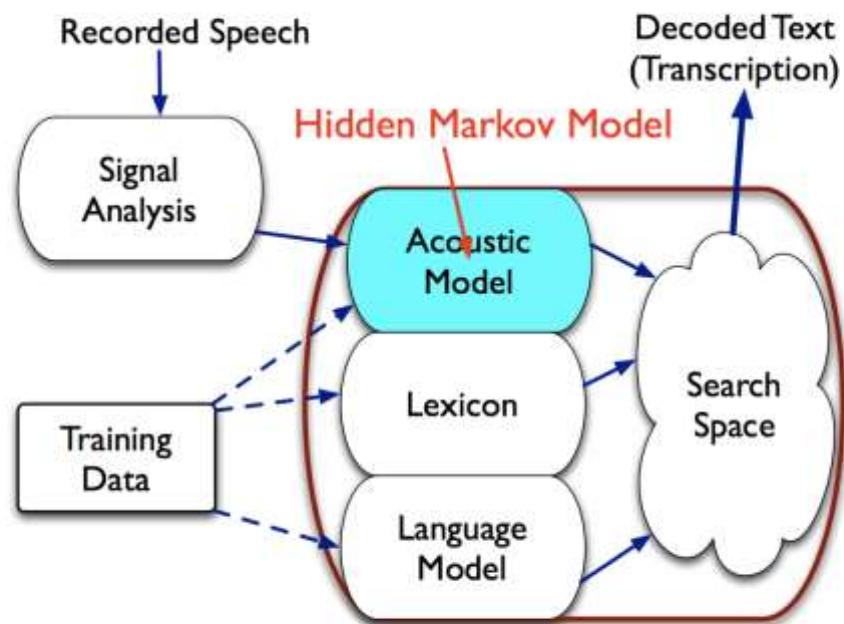
$$\mathbf{W}^* = \arg \max_{\mathbf{W}} \underbrace{p(\mathbf{X} | \mathbf{W})}_{\text{Acoustic model}} \underbrace{P(\mathbf{W})}_{\text{Language model}}$$

NB: \mathbf{X} is used hereafter to denote the output feature vectors from the signal analysis module rather than DFT spectrum.

基础模型框架

HIERARCHICAL MODEL FOR SPEECH RECOGNITION

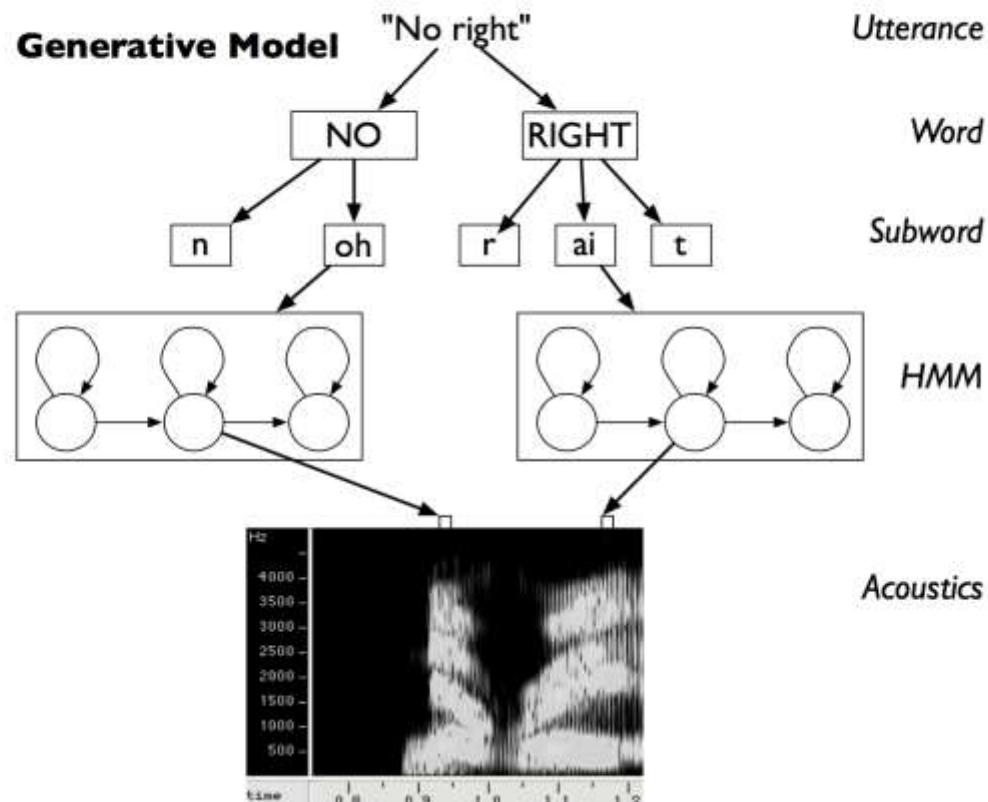
Acoustic Modelling



基础模型框架

HIERARCHICAL MODEL FOR SPEECH RECOGNITION

Hierarchical modelling of speech

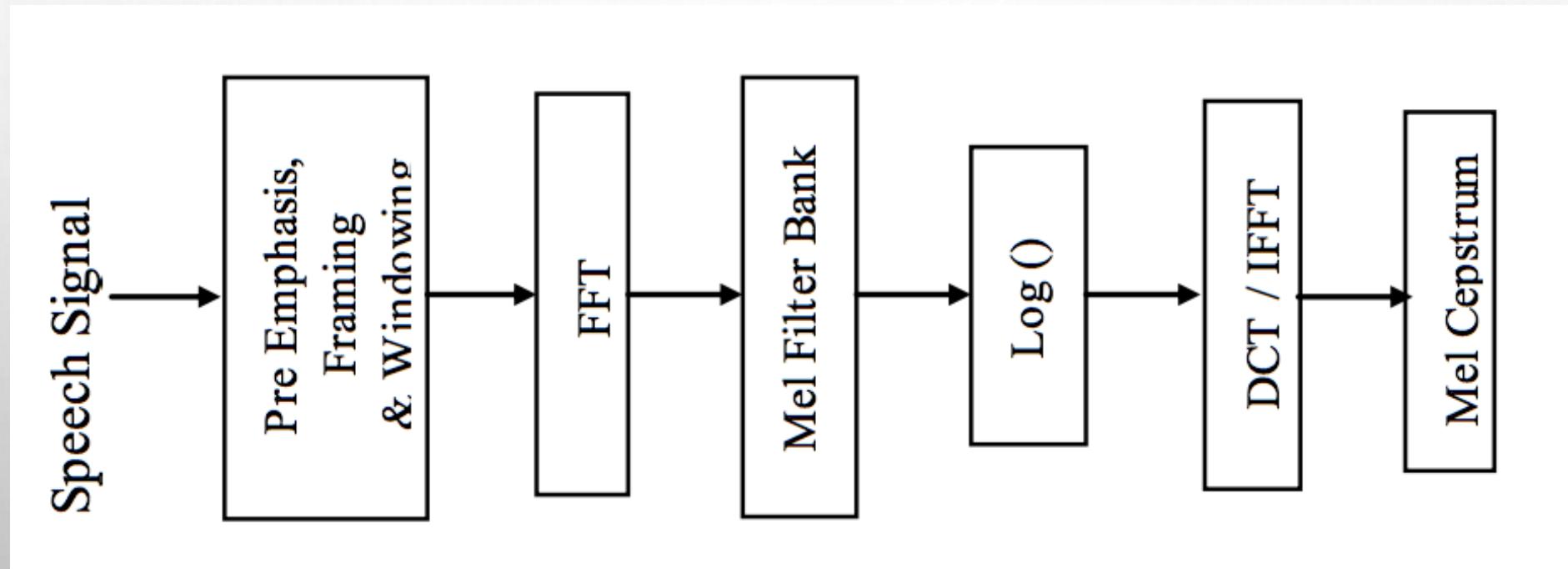


第一课 语音识别(ASR)基础

- 知识点1：什么是ASR：介绍语音识别应用场景、技术发展及难点
- 知识点2：基础模型框架： HIERARCHICAL MODEL FOR SPEECH RECOGNITION
- 知识点3：声音的基本处理：声音信号处理， 以及声音特征提取(频谱图， F-BANK特征， MFCC特征等)
- 知识点4：HMM基础：隐马尔可夫模型原理介绍， 以及前向后向算法
- 知识点5：GMM基础：高斯混合模型原理介绍

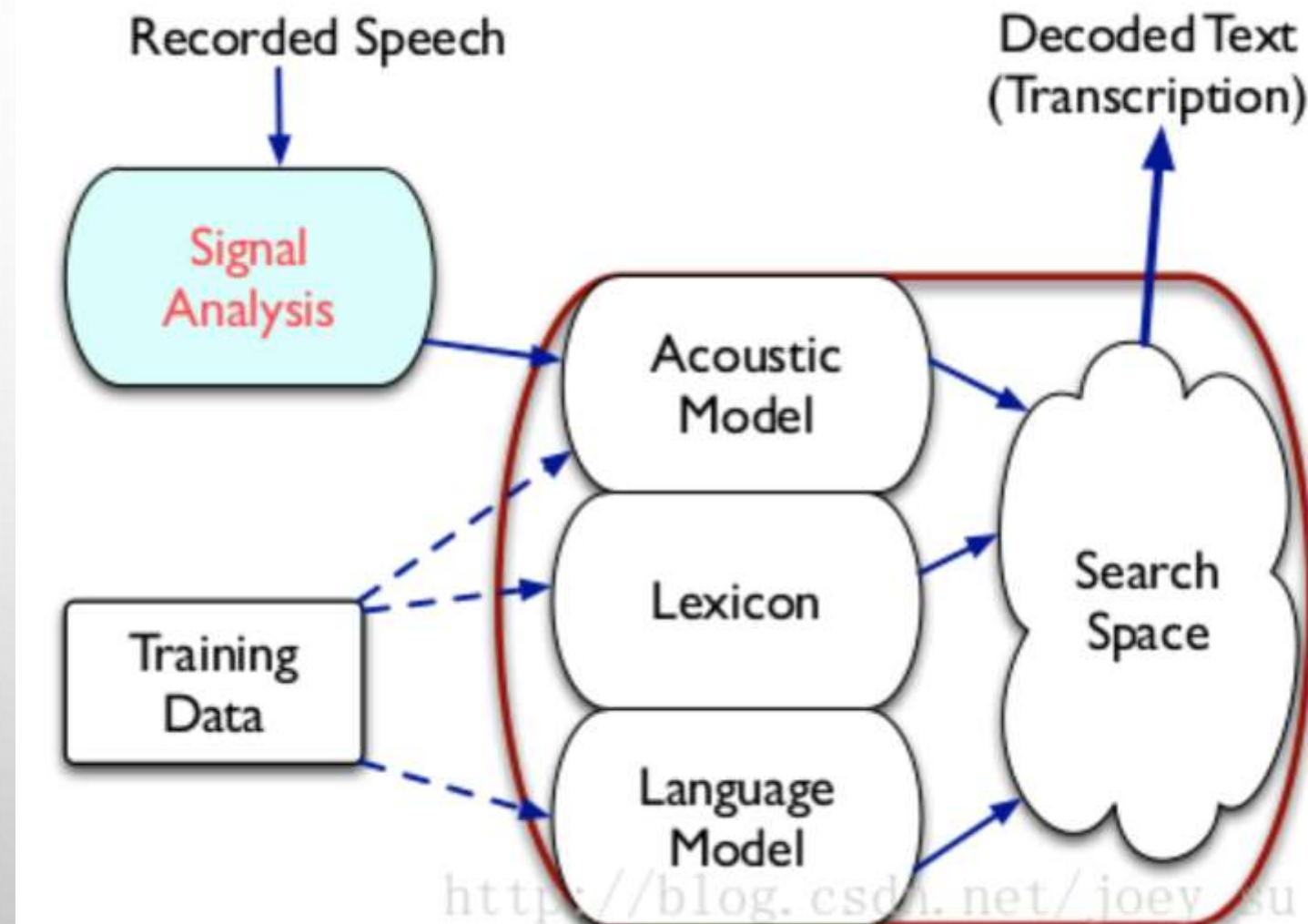
MFCC

MEL FREQUENCY CEPSTRUM COEFFICIENTS

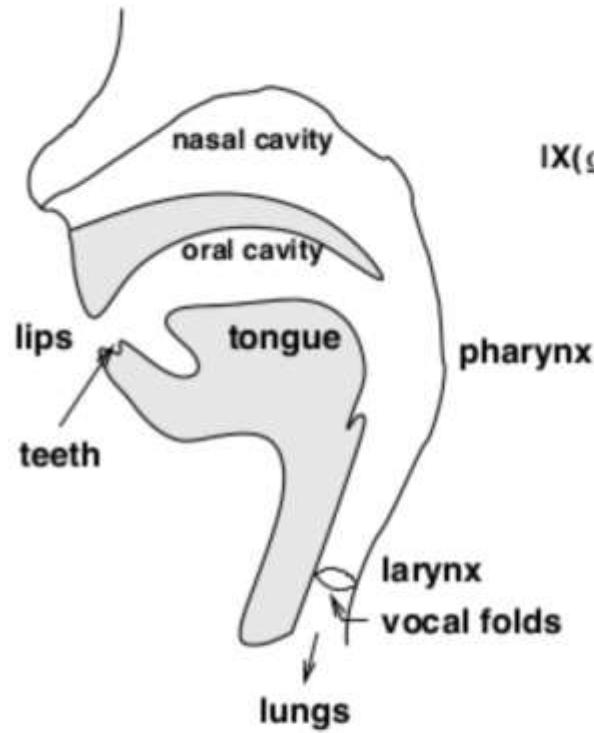


<https://pdfs.semanticscholar.org/0b44/265790c6008622c0c3de2aa1aea3ca2e7762.pdf>
<https://blog.csdn.net/huashui2009120/article/details/80450062>

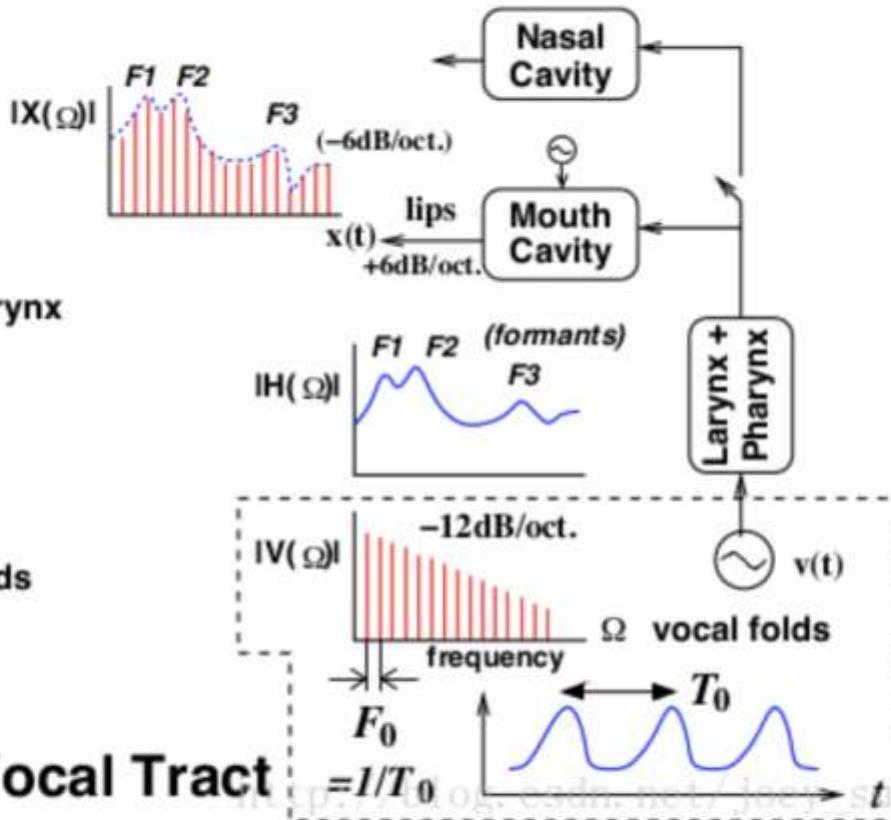
MFCC



我们先来认识下语音的产生过程：



Vocal Organs & Vocal Tract



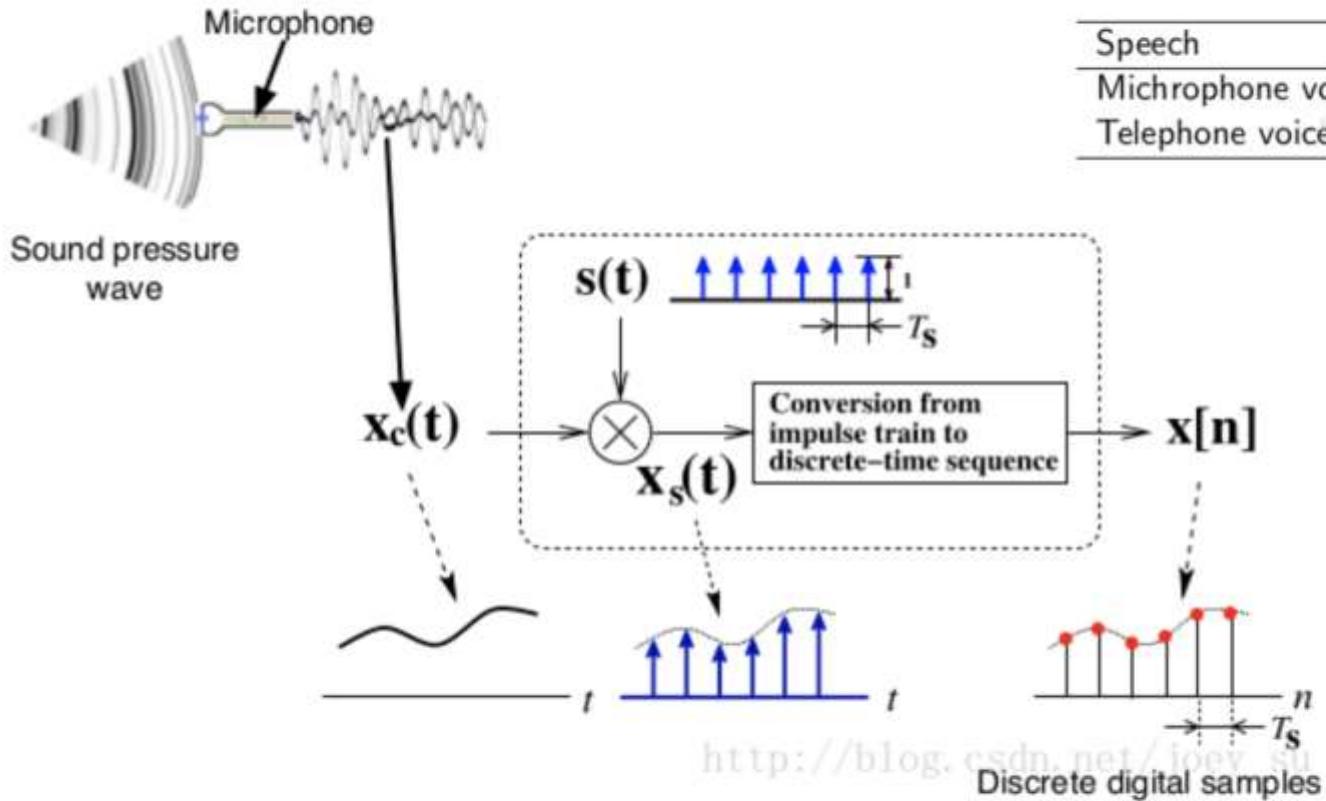
语音是在发音器官和声道共同作用下产生的。说话时，声带振动发出具有一定周期特性（基音周期 T_0 ）的声音，通过喉，咽，鼻腔，口腔等发音器官，以及在嘴唇的摩擦作用下形成语音信号 $x(t)$ ，对 $x(t)$ 进行傅立叶变换，得到频谱 $X(\Omega)$ ， $X(\Omega)$ 由共振峰（ F_1, F_2, F_3 ）组成。

发出的语音属于模拟信号，为了对语音信号进行分析和处理，需要进行模数转换。

采样，即把模拟信号转换为数字的形式：

其中，采样频率为 $F_s = 1/T_s$ ，根据奈奎斯特采样定理，采样频率要大于或等于信号最高频率，实际应用中，采样频率选取如下：

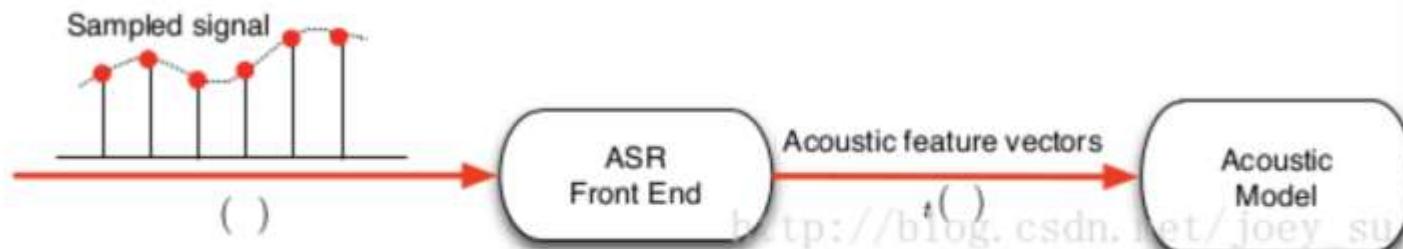
Speech	Sufficient F_s
Microphone voice (< 10kHz)	20 kHz
Telephone voice (< 4kHz)	8 kHz



语音引起空气振动，是一种声压波，用麦克风进行录制。

$x_c(t)$ 是经过麦克风录制后的语音信号，用一个周期为 T_s ，幅度为1的冲击函数与 $x_c(t)$ 相乘，得到 $x_s(t)$ ，将脉冲转换为离散时间序列即得到周期为 T_s 的 $x[n]$ 。

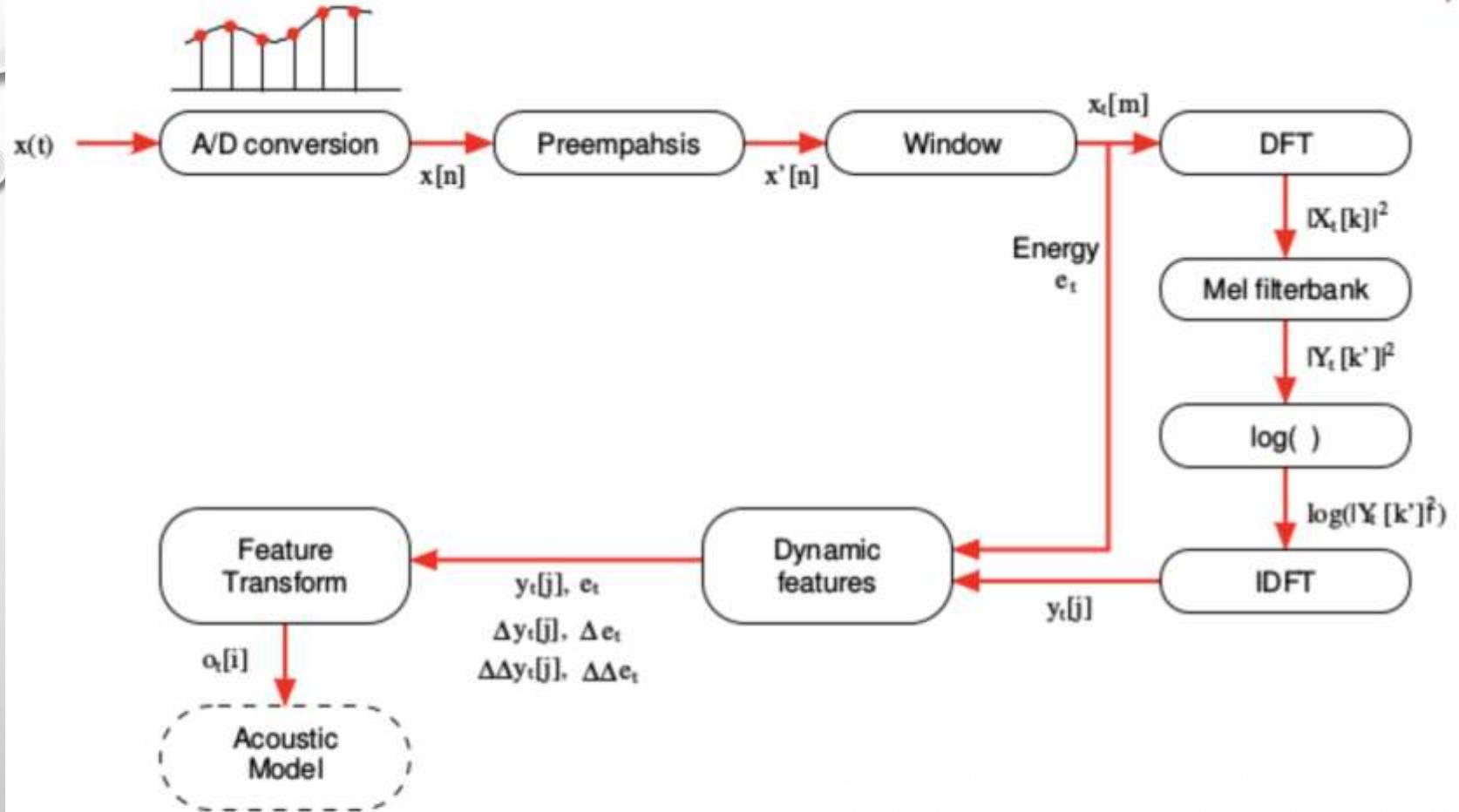
数字化后，下一步的工作是提取语音信号的声学特征：



采样后的信号通过前处理后进行声学特征向量提取，得到声学模型。

用于语音识别的声学特征应包含以下特性：

- 特征应包含区分音素与音素之间的有效信息
 - 良好的时间分辨率 (10ms)
 - 良好的频率分辨率 (~20 channels)
- 分离基音频率 F_0 以及它的谐波成分
- 对不同说话人具有鲁棒性
- 对噪音或者信道失真具有鲁棒性
- 有着良好的模式识别特性
 - 低维特征
 - 特征独立



原始语音信号经过A/D转换得到数字信号， 经过预加重提升高频成分， 接着是加窗， 对加窗后的信号进行两个方面的处理， 一个方面是提取倒谱特征， 即经过离散傅立叶变换后， 对频谱幅度进行平方， 通过梅尔滤波器组， 再进行对数变换， 最后进行离散傅立叶变换的逆运算得到倒谱特征； 另一方面是求加窗后信号的能量， 将这两个方面结合起来形成动态特征， 最后再进行特征变换得到声学模型。

下面对每个步骤进行分析

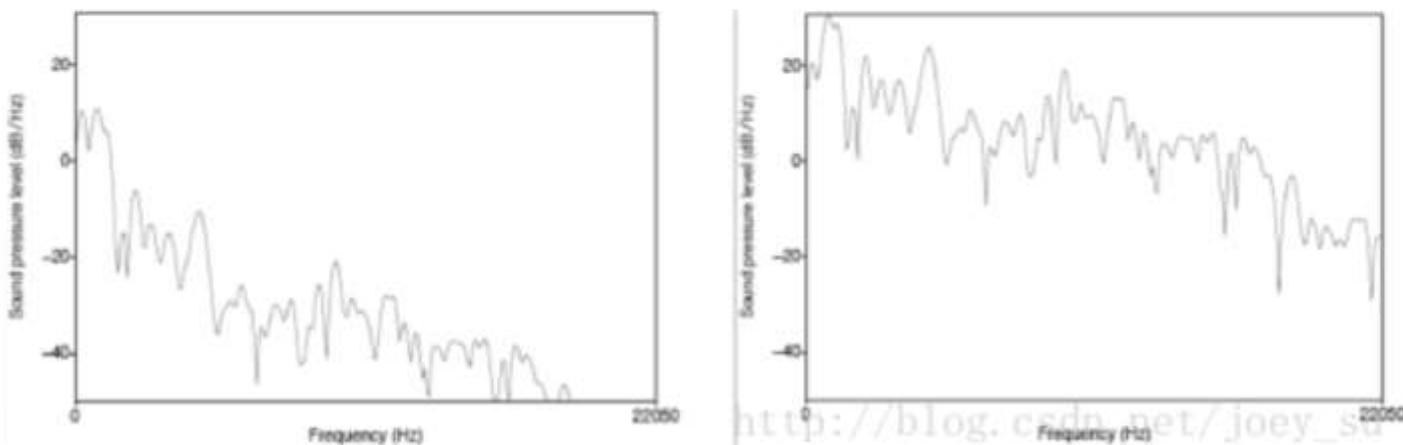
A/D转换在前面已经讲过了，在这里就不赘述。我们从预加重开始。

我们知道，语音是由声门激励通过系统（声道等）产生的，声门激励属于低频，所以语音的能量主要集中在低频，相对于低频来说，高频的能量较低，提升高频分量有助于提高信噪比，可采用预加重的方法，这种方法在通信系统中经常使用。

预加重（第一级）滤波器提升高频，公式如下：

$$x'[n] = x[n] - \alpha x[n-1] \quad 0.95 < \alpha < 0.99$$

下图对元音 /aa/ 进行预加重操作：



在语音信号中，由于声门气流波的影响，每倍频衰减是**12dB**，而唇腔辐射是每倍频增加**6dB**，所以总的效果是每倍频衰减**6dB**，为了弥补这**6dB**我们采取预加重。

由于每倍频会衰减**6dB**，高频部分的能量一般比较低，提高高频部分的能量使得高频能量和低频能量大致相等，尽量弥补每倍频损失的**6dB**

从图中可看到，高频部分有一定的提升。

分帧

- 因为语音信号是快速变化的，而**FOURIER TRANSFORM**适用于分析平稳的信号，利用语音的短时平稳性（在每一时刻所有阶差分都是一样的）
- 在语音识别中一般去帧长为**20MS~50MS(一般取25MS)**，这样一帧内既有足够的周期，又不会变化很剧烈
- 一般帧移取**10MS**，也就是说帧与帧之间有**15MS**是重复的
- $(S - 15) / 10 = \text{帧数}$ ，其中 S 为一段语音的毫秒数。

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx,$$

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi,$$

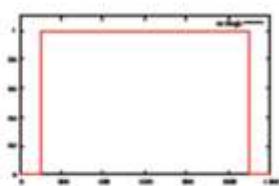
加窗

- 因为之后要做FFT，而一个信号的FFT与这个信号的周期信号的FFT相同，所以如果这个信号边缘不平滑，那么这个信号的周期信号在现实中是很少遇到的，这样就没有意义了，
- 所以每帧信号通常要与一个平滑的窗函数相乘，让帧两端平滑的衰减到零，这样可以降低傅立叶变换后旁瓣的强度(主瓣是变换为频谱之后振幅最大的那个波峰部分，而周围的小的波峰部分叫旁瓣)，取得更高质量的频谱，
- 通常选用的窗函数：**HAMMING WINDOW, HANNING WINDOW**。好的窗函数也能减弱频谱泄漏。
- 这里也解释了为什么要帧移是**10MS**, 有**15MS**的OVERLAP, 由于帧与帧连接处的信号因为加窗而弱化，如果没有OVERLAP,这部分信息就丢失了. (补充一点，这里其实跟**CONTEXT-FRAMES**并不冲突，其实 $(T-5, T, T+5)$ 11帧数据的话，去掉重叠部分能表达的时长是 $(10*11+15) = 125MS$, 通常对于一个音节来说足够了

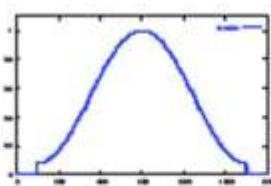
加窗：在时域上，波形乘以窗函数即可得到加窗后的波形，公式为 $x[n] = w[n]s[n]$

如果我们简单地将语音信号分成很多小段，那么这些小段（帧）就是矩形窗，而矩形窗的边缘是陡峭的，即不连续的，所以应该选取边缘连续的窗函数，使得相邻两帧可以平滑过渡。

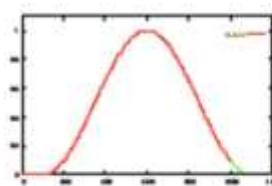
时域上的加窗效果如下：



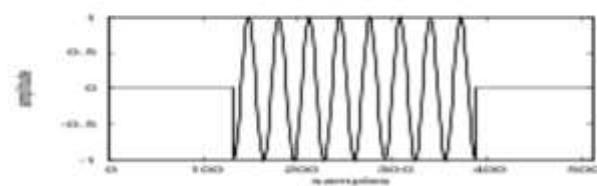
Rectangular



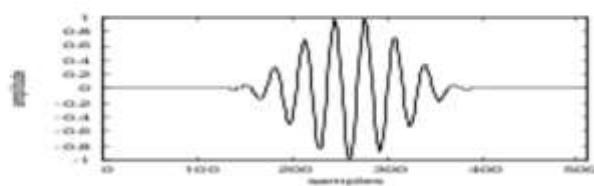
Hamming



Hanning

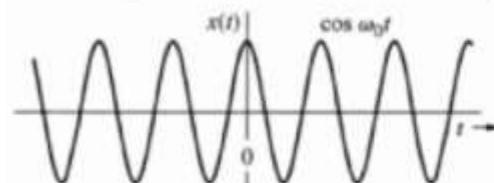


(a) Rectangular window

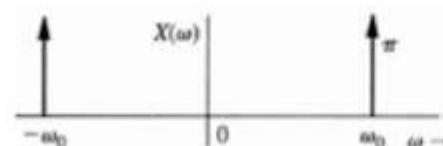


(b) Hanning window

- Extracting a segment of a signal in time is the same as multiplying the signal with a rectangular window:

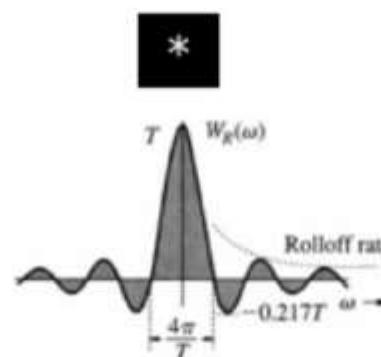
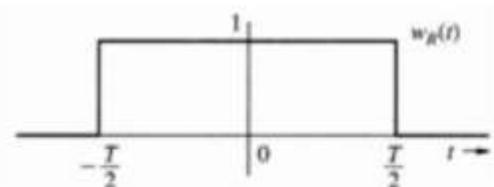


X



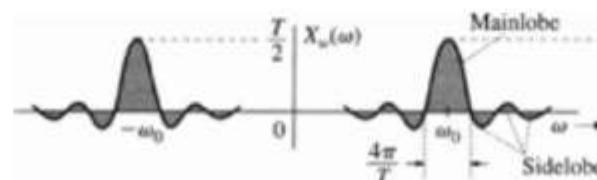
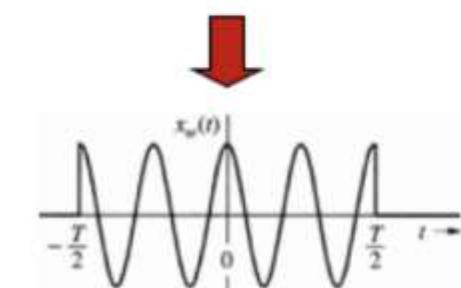
Spectral spreading

Energy spread out from ω_0 to width of $2\pi/T$ – reduced spectral resolution.



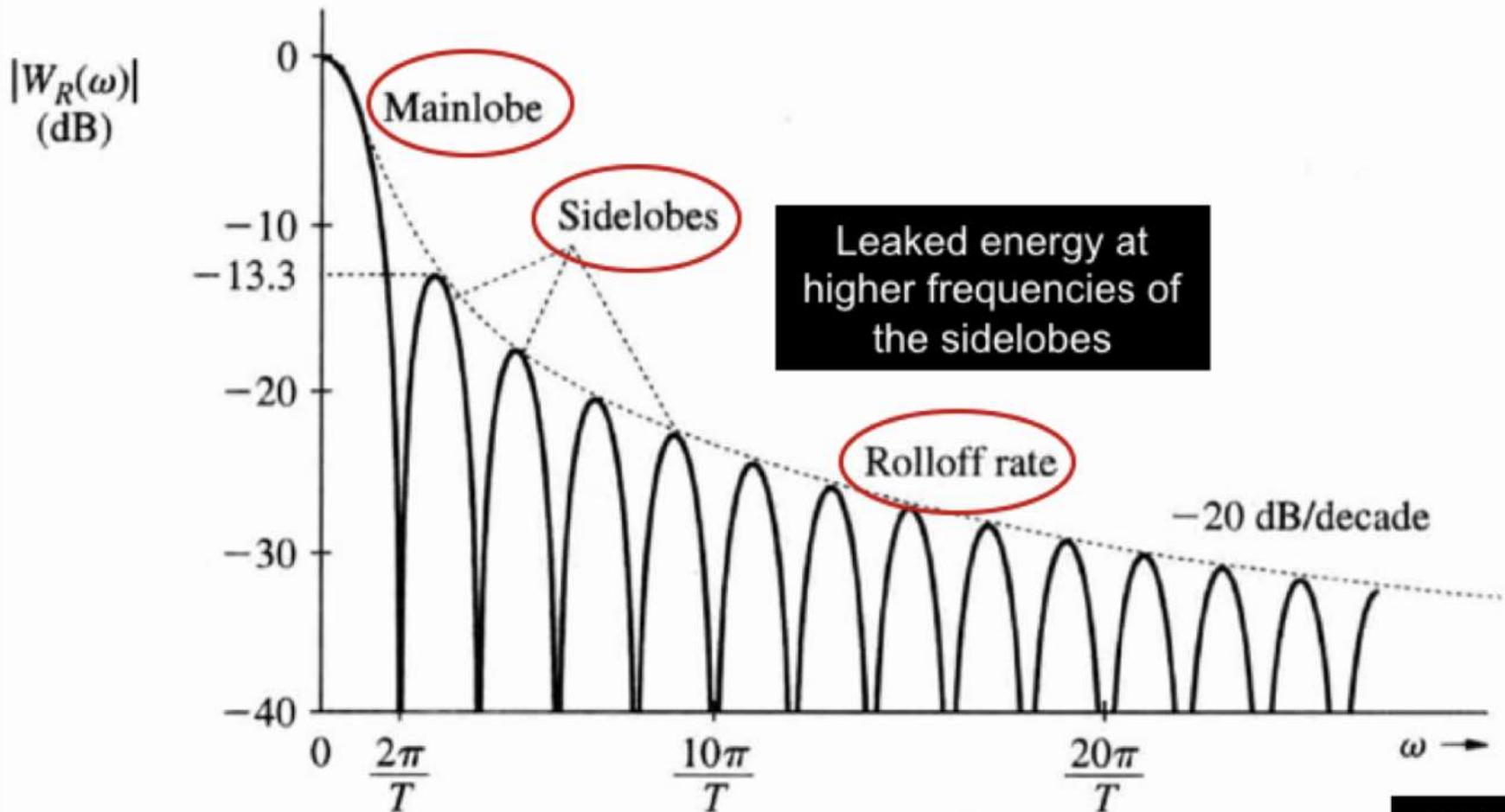
Leakage

Energy leaks out from the mainlobe to the sidelobes.

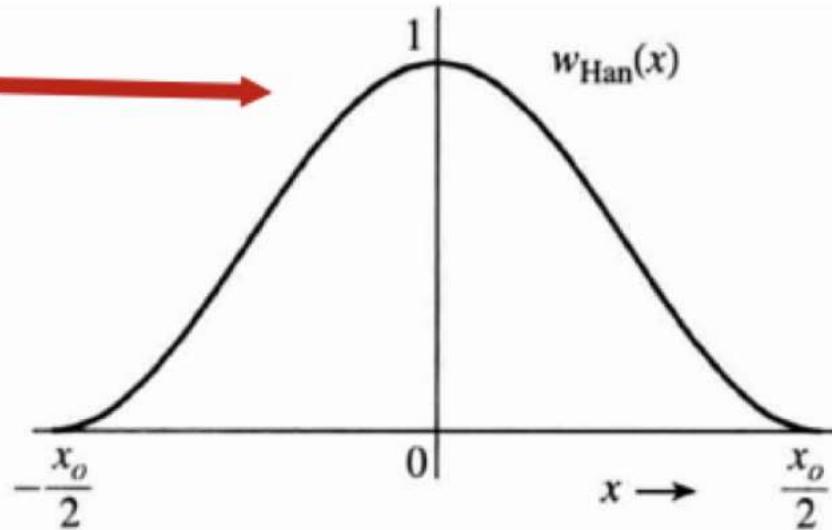


L7.8 p746

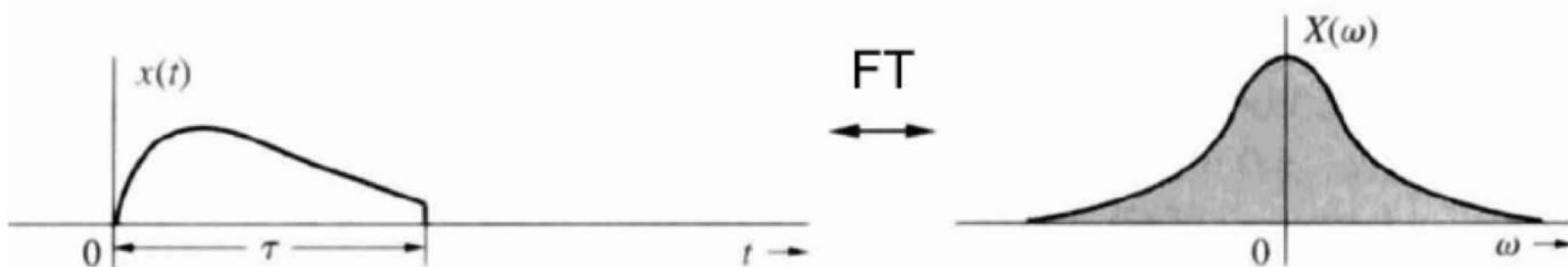
- ◆ Detail effects of windowing (rectangular window):



1. Make mainlobe width as narrow as possible → implies as wide a window as possible.
 2. Avoid big discontinuity in the windowing function to reduce leakage (i.e. high frequency sidelobes).
 3. 1) and 2) above are incompatible – therefore needs a compromise.
- ◆ Commonly replace rectangular window with one of these:
- Hamming window
 - Hanning window
 - Barlett window
 - Blackman window
 - Kaiser window

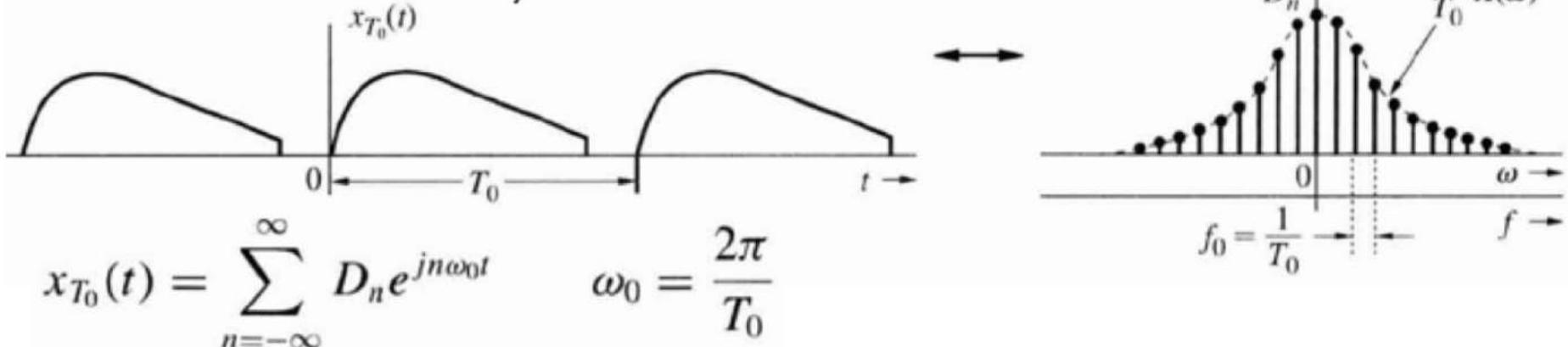


- ◆ As expected, time domain sampling has a dual: spectral sampling.
- ◆ Consider a time limited signal $x(t)$ with a spectrum $X(\omega)$.



$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt = \int_0^{\tau} x(t)e^{-j\omega t} dt$$

- If we now CONSTRUCT a periodic signal $x_{T_0}(t)$, we will expect the spectrum of this signal to be discrete (expressed as Fourier series).

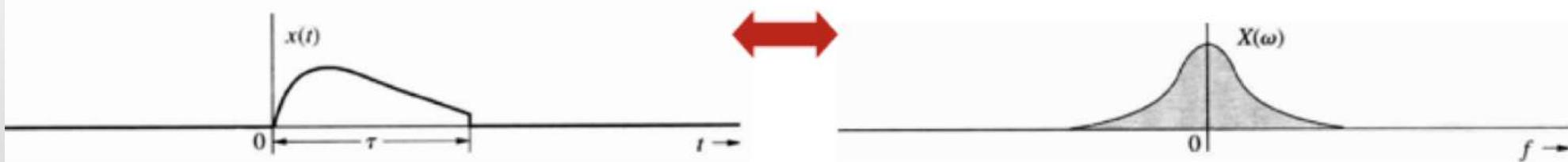


where $D_n = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-j n \omega_0 t} dt = \frac{1}{T_0} \int_0^\tau x(t) e^{-j n \omega_0 t} dt$

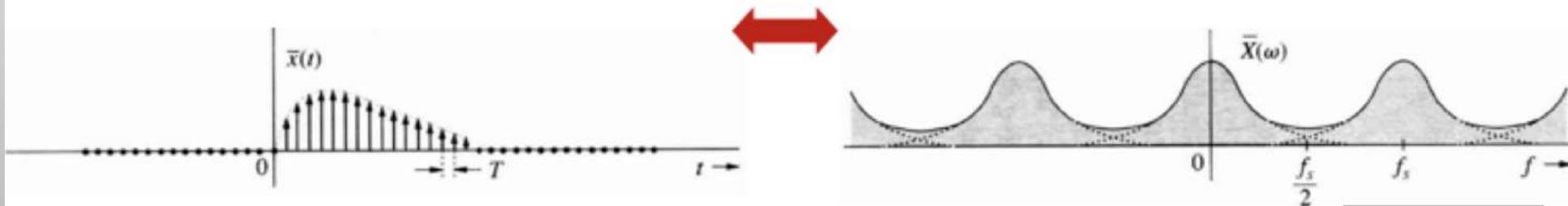
therefore

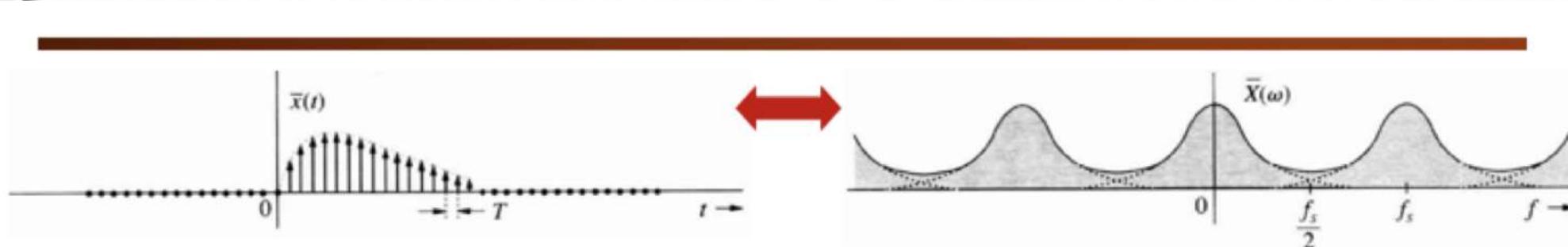
$$D_n = \frac{1}{T_0} X(n\omega_0)$$

- ◆ Fourier transform is computed (on computers) using discrete techniques.
- ◆ Such numerical computation of the Fourier transform is known as Discrete Fourier Transform (DFT).
- ◆ Begin with time-limited signal $x(t)$, we want to compute its Fourier Transform $X(\omega)$.

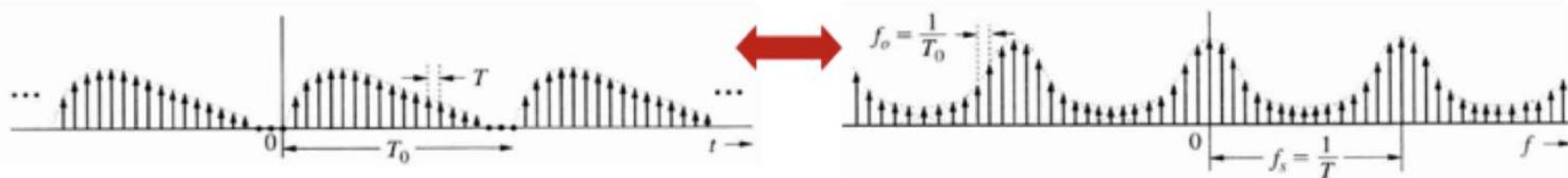


- ◆ We know the effect of sampling in time domain:





- ◆ Now construct the sampled version of $x(t)$ as repeated copies. The effect (from slides 6-8) is sampling the spectrum.



Number of time samples in T_0

$$N_0 = \frac{T_0}{T}$$

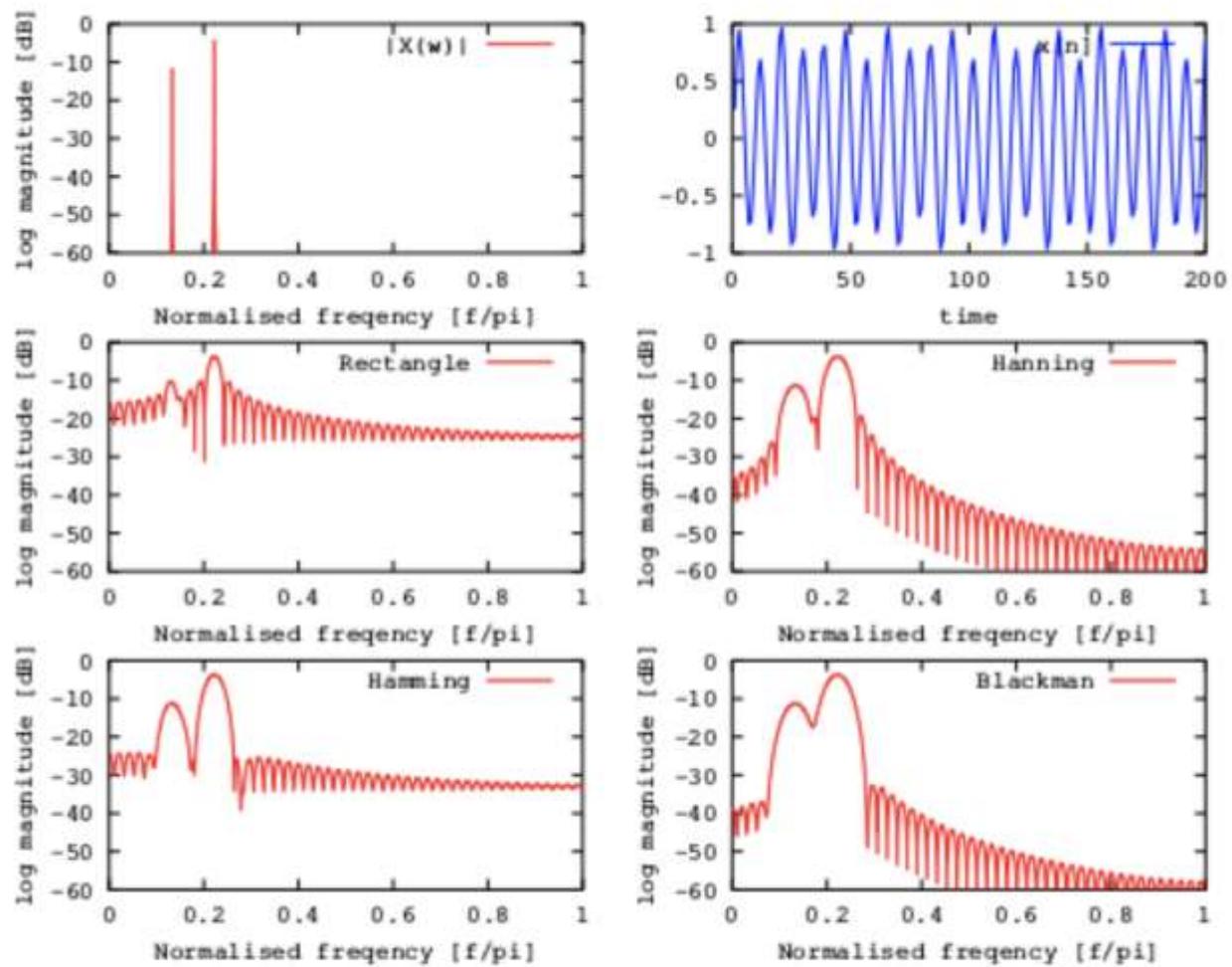
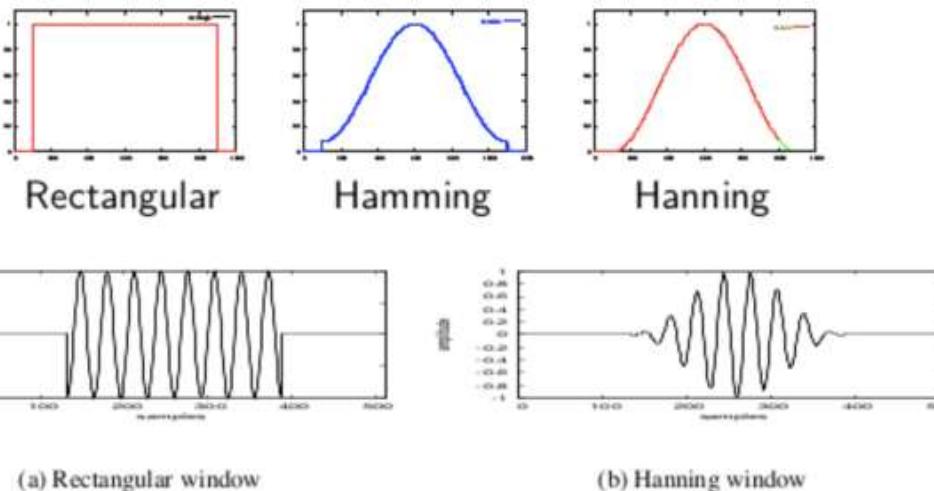
Number of frequency samples in f_s

$$N'_0 = \frac{f_s}{f_0}$$

$$N_0 = \frac{T_0}{T} = \frac{1/f_0}{1/f_s} = \frac{f_s}{f_0} = N'_0$$

频域上的加窗效果如下：

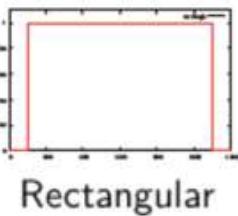
时域上的加窗效果如下：



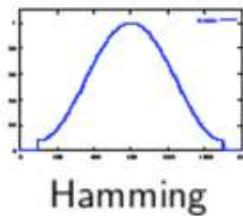
$$x(t) = 0.15 \sin(\pi f_1 t) + 0.85 \sin(\pi f_2 t + 0.3)$$

$f_1 = 0.13, f_2 = 0.22$

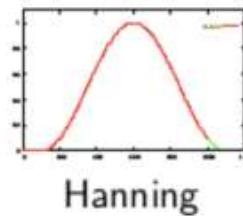
时域上的加窗效果如下：



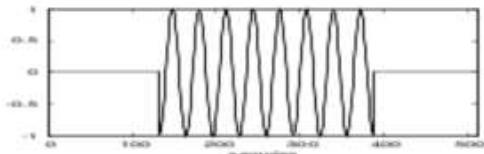
Rectangular



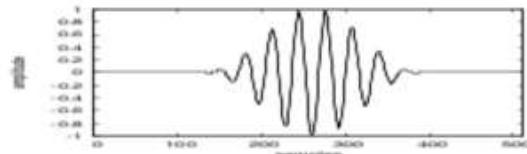
Hamming



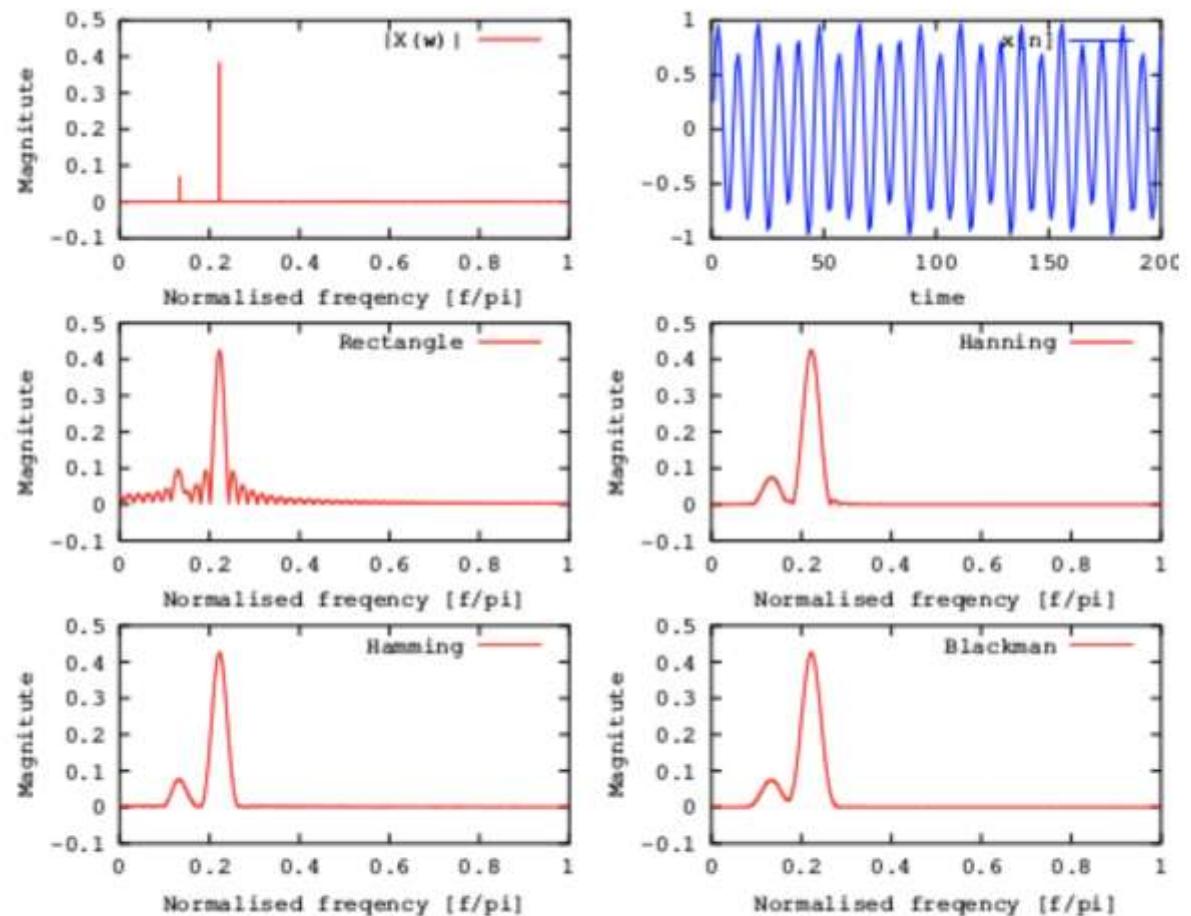
Hanning



(a) Rectangular window



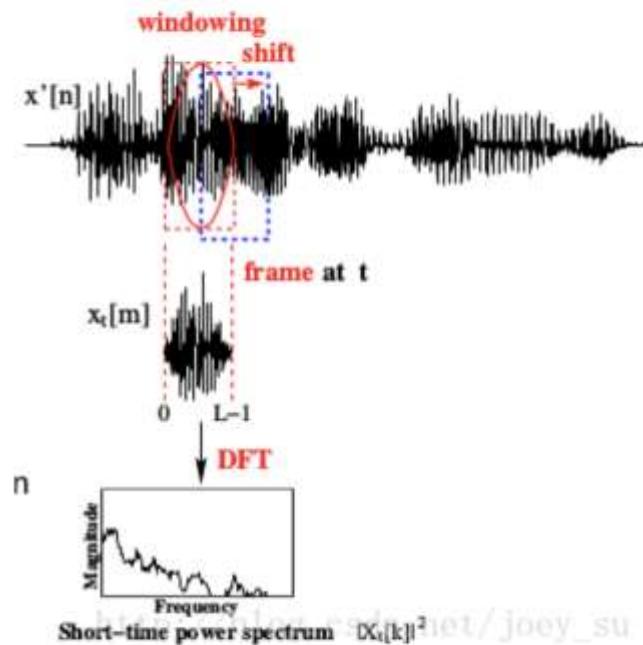
(b) Hanning window



$$x(t) = 0.15 \sin(\pi f_1 t) + 0.85 \sin(\pi f_2 t + 0.3)$$

$$f_1 = 0.13, f_2 = 0.22$$

http://blog.csdn.net/joey_su



首先对语音信号 $x[n]$ 加窗，加窗后的信号为 $x_t[m]$ ， t 为时域信号的时间点， m 表示第 m 个窗，然后对每帧进行傅立叶变换，得到短时功率谱 $|X_t[k]|^2$ 。

补零(zero-padding)

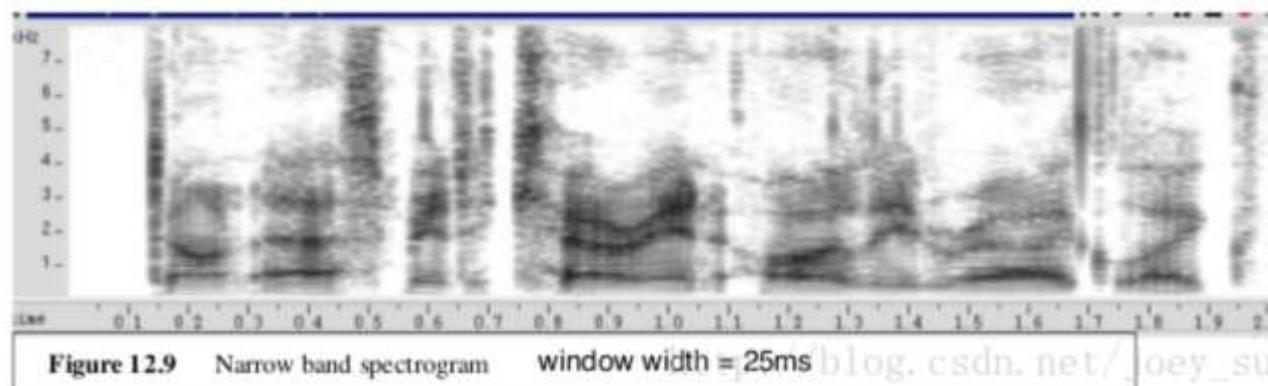
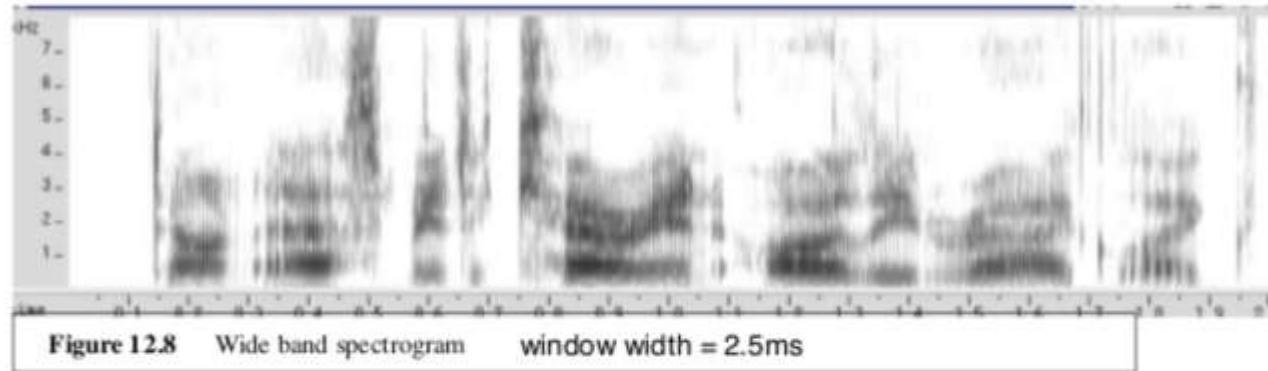
- 因为做FFT(快速傅里叶变化)要求信号长度为 2^n ，所以如果采样率为 16000Hz, $16000 * 0.025 = 400$ ，要补 0 使长度为 512。

FFT

- 将时域谱转化为频率谱，纵坐标变为能量。fourier transform 逐帧进行的，为的是取得每一帧的频谱。一般只保留幅度谱，丢弃相位谱

这个过程中，需要注意的有两点，之一是帧长，对于使用较短的帧，其具有较宽的频带，较高的时间分辨率和较低的频率分辨率，而对于较长的帧，则具有较窄的频带，较低的时间分辨率和较高的频率分辨率；另一点要注意的是为了使帧与帧之间的过渡更加平稳，采用了帧移的方法，即两帧之间有个重叠区域。

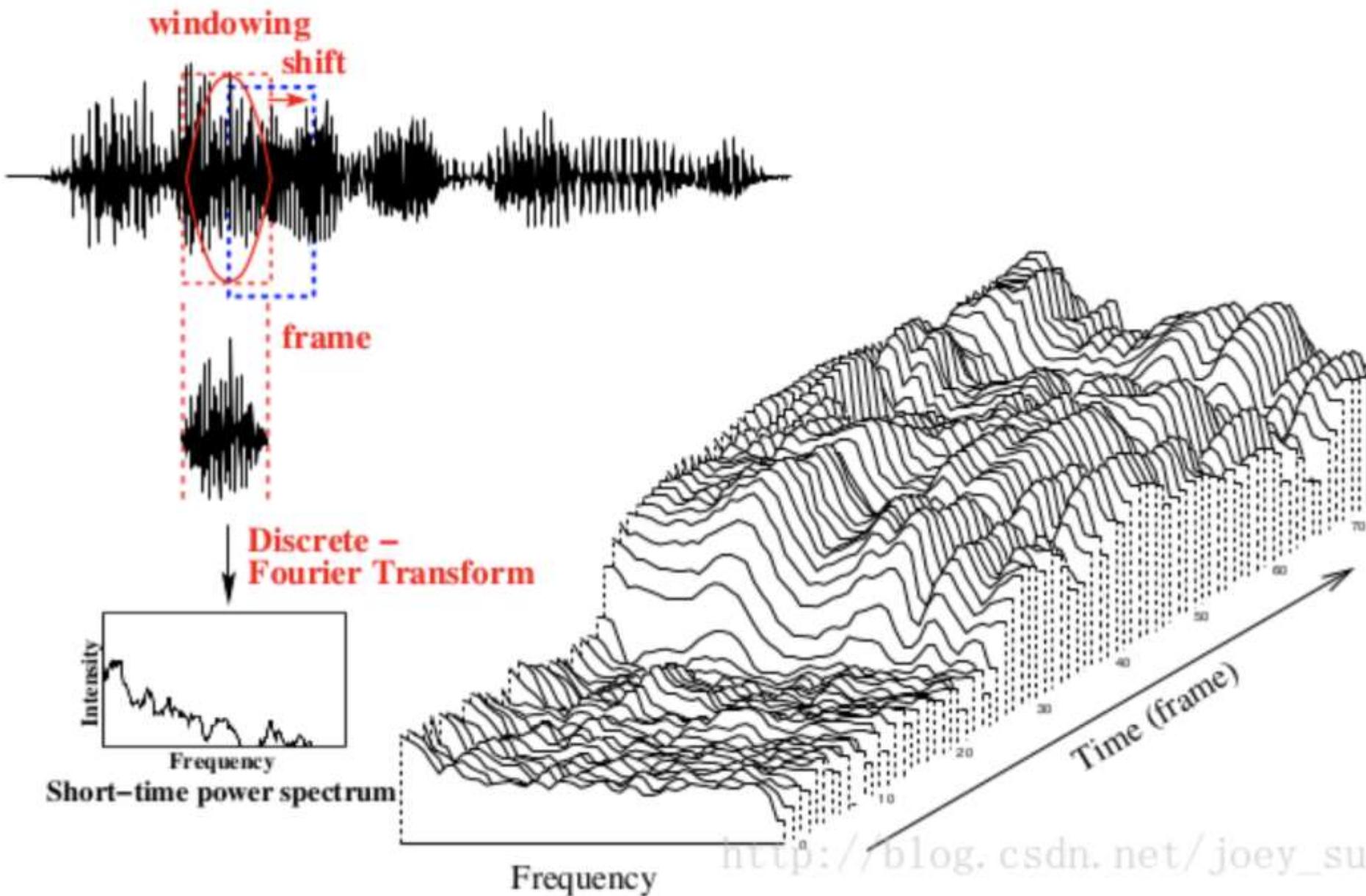
下图为宽带和窄带的语谱图对比：



语谱 (spectrogram)

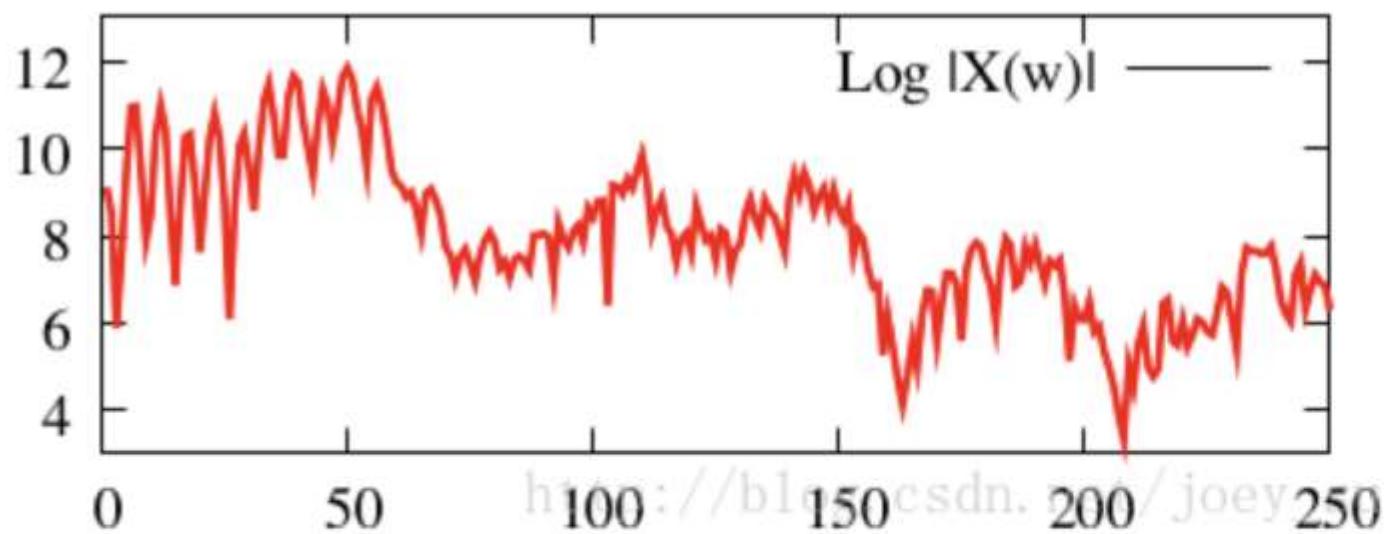
实际上就是把每帧的频谱图向左旋转90度，用颜色的深浅表示幅度的大小，幅度越大颜色越深，然后把每帧的颜色信息按照时间（帧）的顺序列出来，所以，语谱的横坐标为时间（帧），纵坐标为频率，颜色为频率的幅度

短时频谱分析



DFT频谱特征

从前面的介绍中，我们看到频带是等间隔的，但是我们知道，人类的耳朵其实是一个超级强大的语音识别系统，我们研究语音识别时，很大程度是从人类自身来寻找答案的，从人类听觉系统上看，我们的耳朵对声音的获取是有选择性的，对于大于1000Hz的声音，人类的听觉敏感度会降低，具体为什么是1000Hz，应该是跟耳朵的生理构造有关吧。



功率谱包含F0的基频（前面讲过的），正因为这样，使得估计频谱的包络变得困难，但还是有办法的。

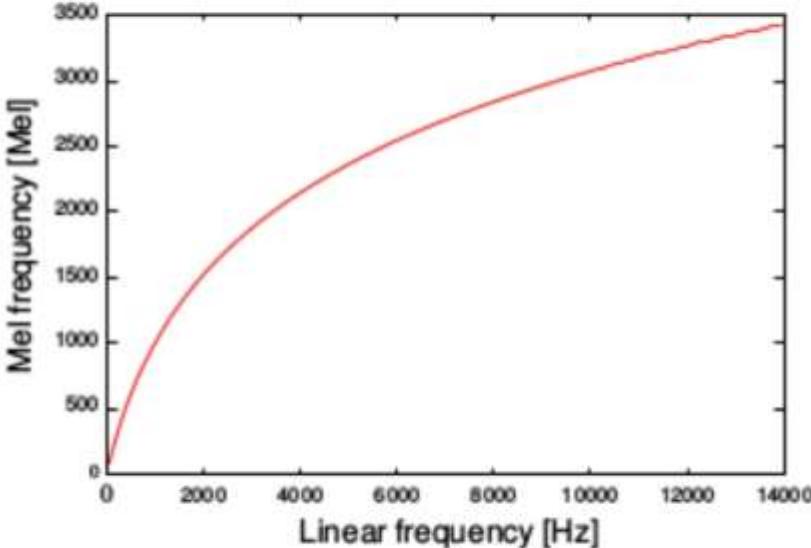
非线性频率刻度

前面提到人类的听觉系统对越高频率的敏感度越低，这就说明了人类的频率的感知是非线性的，也就是说人耳自身对声音的频率有所划分，并且这些划分的频段是非线性的（不是等间隔的）。

下面是三种非线性刻度，分别是Mel刻度和Bark刻度和Ln刻度，实际语音处理中常用到Mel刻度：

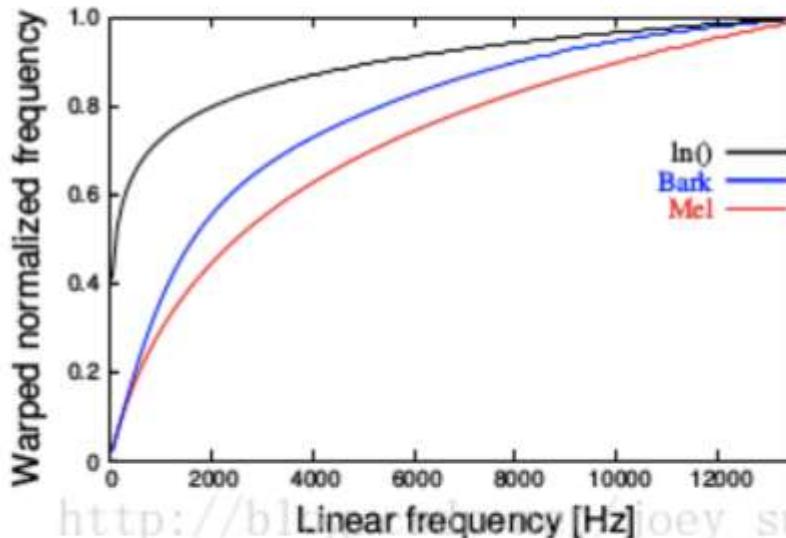
Mel scale

$$M(f) = 1127 \ln(1 + f / 700)$$



Bark scale

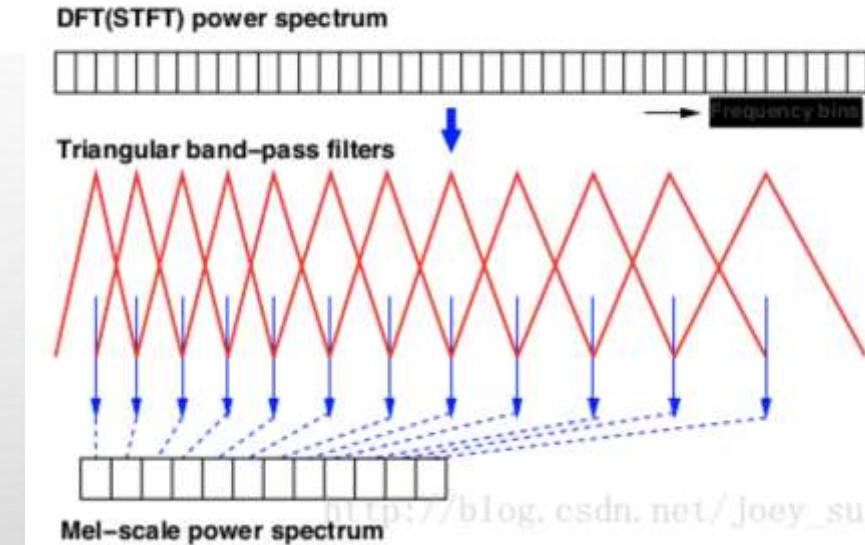
$$b(f) = 13 \arctan(0.00076f) + 3.5 \arctan((f/7500)^2)$$



实际语音处理中常用到Mel刻度。

首先要明白为什么要设置Mel滤波器组。在这里，我们用到了若干个间隔不等的三角低通滤波器构成的滤波器组，由上面的介绍中我们了解到可以使用Mel刻度来代替线性的频率刻度，以满足人类的听觉特性。所以，我们需要对频率刻度的频点（Frequency bins）进行分类，分类是按顺序进行的，这个分类就需要Mel滤波器组来实现，上图一共12个三角形，所以可以理解为将一大段频点分成了12类，也就是12中Mel刻度功率谱。需要注意的是小于1000Hz的部分为线性间隔，而大于1000Hz的部分为对数间隔。

Mel滤波器组



频谱有包络和精细结构，分别对应音色与音高。

对于语音识别来讲，音色是主要的有用信息，音高一般没有用。

在每个三角形内积分，就可以消除精细结构，只保留音色的信息。

当然，对于有声调的语言来说，音高也是有用的，所以在MFCC特征之外，还会使用其它特征刻画音高。

对数能量

为什么要计算对数能量呢？

- 可以使用对数来压缩动态范围
- 人类对于信号能量的敏感度是呈对数的，例如，人类对于高能量中的小变化表现出相对于低能量更低的敏感度，也就是说人类低能量区的变化更敏感
- 对数使得声学耦合的变化对于特征来说是不可变的，也就是对数使得声学耦合的变化在特征提取中变得可有可无。
- 移除相位信息，相位信息对于语音识别来说不是很重要（不过不是所有的人都认同这一点）

可通过计算每个Mel滤波器组输出的对数功率谱的平方来得到对数能量。

什么是倒谱？倒谱倒谱，倒过来的频谱，将频谱（Spectrum）的前四个字母倒过来就变成了倒谱（Cepstrum）。所以我们可以近似理解为倒谱是频谱的一种特殊的逆变换，专业说法是同态处理。

语音产生模型可看成是发声源-滤波器（Source-Filter）模型：

发声源（Source）：声带振动产生声门源波形

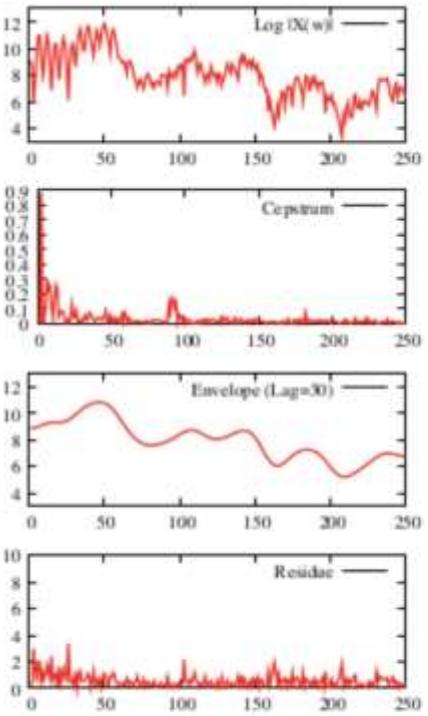
滤波器（Filter）：发声源波形通过声道：舌头的位置，下巴等。给定一个特定的形状就会有一个特定的滤波特性（你往我脸上打一拳，脸上的某个部位就会肿出来，求别打脸）。

需要注意的是发声源的特性（F0, 动态的声门脉冲）对区分音素并没有帮助；

而滤波器指定了发音器官的位置，这些都是固定的，所以可以区分音素。

说了那么多，倒谱究竟能干嘛呢？

倒谱分析可以帮助我们分离发声源和滤波器！这样，我们就可以把滤波器分离出来，就可以区分音素了，这就是非常重要的倒谱特征，语音识别最基本的单元就是音素，即phone（不是手机啊囧），后面就会有Bi-phone, Tri-phone...要是我们能把最基本的单元区分（识别）了，就可以做后面的工作了。



Log Spectrum (freq domain)

↓ Inverse Fourier Transform

Cepstrum (time domain) (quefrency)

↓ Liftering to get low/high part
(lifter: filter used in cepstral domain)

↓ Fourier Transform

Smoothed-spectrum (freq. domain)

[low-part of cepstrum]

Log spectrum

[high-part of cepstrum]

http://blog.csdn.net/joey_su

姗姗来迟的图——将功率谱分离成频谱包络和F0谐波。

对数频谱（频域）通过傅立叶变换的逆变换变成倒谱（时域）（逆频率），同态滤波得到高低两部分（其实就是在倒谱域中加入两个滤波器来完成），然后再进行傅立叶变换即可得到平滑的频谱（频域），这个是倒谱中较低的部分，还得到对数频谱，也就是倒谱中较高的部分。

从图中我们可知第三个图就是我们想要的，它就是原功率谱的包络（Envelope），而第四个图的Residue呢就是分离出来的发声源了，这部分可扔掉。就好比榨橙汁，把渣渣过滤掉就是我们可口的橙汁啦。

给出一个平滑频谱 (smoothed spectrum) 的概念：变换到倒谱域，截断，再变换回频域。

那么，MFCCs有什么用呢？

- 作为声学特征被广泛用于基于HMM的语音识别系统
- 前12个MFCCs通常被用作特征向量（也就是移除FO的信息）
- 相对频谱特征有着更小的相关性，也就是说比频谱特征更容易建立模型
- 它的表示非常紧凑（挤挤更健康），因为这12个特征描述了一段语音数据中的一个20ms的帧，再回去看看上面的语谱图就明白了
- 对于标准的基于HMM的系统，MFCCs在语音识别的性能比滤波器组或者语谱特征更优越
- 可惜的是MFCCs抵抗噪声的鲁棒性不强

第一课 语音识别(ASR)基础

- 知识点1：什么是ASR：介绍语音识别应用场景、技术发展及难点
- 知识点2：基础模型框架： HIERARCHICAL MODEL FOR SPEECH RECOGNITION
- 知识点3：声音的基本处理：声音信号处理， 以及声音特征提取(频谱图， F-BANK特征， MFCC特征等)
- 知识点4：HMM基础：隐马尔可夫模型原理介绍， 以及前向后向算法
- 知识点5：GMM基础：高斯混合模型原理介绍

GMM + HMM

- <HTTP://59.80.44.98/WWW.INF.ED.AC.UK/TEACHING/COURSES/ASR/2018-19/ASR03-HMMGMM-HANDOUT.PDF>

第一课 语音识别(ASR)基础

- 知识点1：什么是ASR：介绍语音识别应用场景、技术发展及难点
- 知识点2：基础模型框架： HIERARCHICAL MODEL FOR SPEECH RECOGNITION
- 知识点3：声音的基本处理：声音信号处理， 以及声音特征提取(频谱图， F-BANK特征， MFCC特征等)
- 知识点4：HMM基础：隐马尔可夫模型原理介绍， 以及前向后向算法
- 知识点5：GMM基础：高斯混合模型原理介绍

隐马尔科夫模型 (HIDDEN MARKOV MODELS)

- 隐马尔可夫模型(HIDDEN MARKOV MODEL, HMM)作为一种统计分析模型，创立于20世纪70年代，80年代得到了传播和发展并成功应用于声学信号的建模中，到目前为止，它仍然被认为是实现快速精确语音识别系统最成功的方法。

* HIDDEN MARKOV MODELS (HMMS) WERE INITIALLY DEVELOPED IN THE 1960' S BY BAUM AND EAGON AT THE INSTITUTE FOR DEFENSE ANALYSES (IDA). IN THE 1970' S, BAKER AT CARNEGIE-MELLON UNIVERSITY (CMU), JELINEK AT IBM, AND OTHER APPLIED HMMS TO THE PROBLEM OF SPEECH RECOGNITION.

* IN 1980, IDA INVITED A NUMBER OF RESEARCH ORGANIZATIONS IN SPEECH RECOGNITION, AMONG THEM WERE AT&T AND BBN, FOR A WORKSHOP ON HMMS. IN THE MID 1980' S, SEVERAL HMM-BASED SPEECH RECOGNITION SYSTEMS FROM AT&T, BBN, AND CMU SHOWED SUPERIOR RESULTS.

* THE SUCCESS OF THESE SYSTEMS DRAMATICALLY INCREASED INTEREST IN APPLYING HMMS TO CONTINUOUS SPEECH RECOGNITION AND OTHER DIFFICULT PATTERN RECOGNITION PROBLEMS.

- 作为信号处理的一个重要方向，HMM广泛应用于图像处理，模式识别，语音人工合成和生物信号处理等领域的研究中，并取得了诸多重要的成果。近年来，很多研究者把HMM应用于计算机视觉、金融市场的波动性分析和经济预算等新兴领域中。
- 用来描述一个含有隐含未知参数的马尔可夫过程。其难点是从可观察的参数中确定该过程的隐含参数。

HMM – 栗子

A concrete example [edit]

Consider two friends, Alice and Bob, who live far apart from each other and who talk together daily over the telephone about what they did that day. Bob is only interested in three activities: walking in the park, shopping, and cleaning his apartment. The choice of what to do is determined exclusively by the weather on a given day. Alice has no definite information about the weather where Bob lives, but she knows general trends. Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.

Alice believes that the weather operates as a discrete Markov chain. There are two states, "Rainy" and "Sunny", but she cannot observe them directly, that is, they are hidden from her. On each day, there is a certain chance that Bob will perform one of the following activities, depending on the weather: "walk", "shop", or "clean". Since Bob tells Alice about his activities, those are the observations. The entire system is that of a hidden Markov model (HMM).

Alice knows the general weather trends in the area, and what Bob likes to do on average. In other words, the parameters of the HMM are known. They can be represented as follows in Python:

```
states = ('Rainy', 'Sunny')

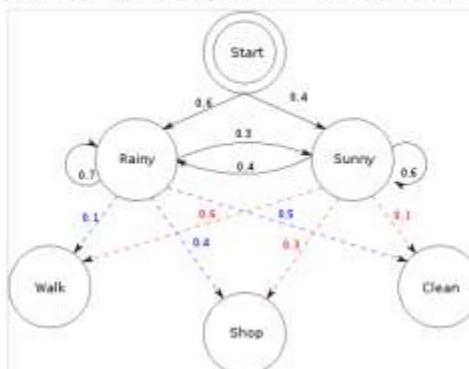
observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
    'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
    'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
}

emission_probability = {
    'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
    'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
}
```

In this piece of code, `start_probability` represents Alice's belief about which state the HMM is in when Bob first calls her (all she knows is that it tends to be rainy on average). The particular probability distribution used here is not the equilibrium one, which is (given the transition probabilities) approximately `{'Rainy': 0.57, 'Sunny': 0.43}`. The `transition_probability` represents the change of the weather in the underlying Markov chain. In this example, there is only a 30% chance that tomorrow will be sunny if today is rainy. The `emission_probability` represents how likely Bob is to perform a certain activity on each day. If it is rainy, there is a 50% chance that he is cleaning his apartment; if it is sunny, there is a 60% chance that he is outside for a walk.



A similar example is further elaborated in the Viterbi algorithm page.

本文链接地址

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/HIDDEN_MARKOV_MODEL#A_CONCRETE_EXAMPLE](https://en.wikipedia.org/wiki/Hidden_Markov_model#A_concrete_example)

[HTTP://WWW.52NLP.CN/HMM-CONCRETE-EXAMPLE-ON-WIKI/](http://www.52nlp.cn/HMM-CONCRETE-EXAMPLE-ON-WIKI/)

HMM – 原理 – 基本概念

定义 10.1 (隐马尔可夫模型) 隐马尔可夫模型是关于时序的概率模型，描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列，再由各个状态生成一个可观测而产生观测随机序列的过程。隐藏的马尔可夫链随机生成的状态的序列，称为状态序列 (state sequence)；每个状态生成一个观测，而由此产生的观测的随机序列，称为观测序列 (observation sequence)。序列的每一个位置又可以看作是一个时刻。

隐马尔可夫模型由初始概率分布、状态转移概率分布以及观测概率分布确定。隐马尔可夫模型的形式定义如下：

设 Q 是所有可能的状态的集合， V 是所有可能的观测的集合。

```
states = ('Rainy', 'Sunny')          Q = {q1, q2, ..., qN} ,   V = {v1, v2, ..., vM}      observations = ('walk', 'shop', 'clean')
```

其中， N 是可能的状态数， M 是可能的观测数。

I 是长度为 T 的状态序列， O 是对应的观测序列。

$$I = (i_1, i_2, \dots, i_T), \quad O = (o_1, o_2, \dots, o_T)$$

I = ('Rainy', 'Sunny', 'Sunny', 'Sunny')
= 4

O = ('clean', 'walk', 'shop', 'clean') T

摘自：《统计学习方法》李航，第10章 隐马尔科夫模型

HMM - 原理 - 基本概念

- 状态转移概率矩阵

A 是状态转移概率矩阵:

$$A = [a_{ij}]_{N \times N}$$

其中,

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i), \quad i=1,2,\dots,N; j=1,2,\dots,N$$

是在时刻 t 处于状态 q_i 的条件下在时刻 $t+1$ 转移到状态 q_j 的概率.

```
transition_probability = {
    'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
    'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
}
```

- B 是观测概率矩阵:

$$B = [b_j(k)]_{N \times M}$$

其中,

$$b_j(k) = P(o_t = v_k | i_t = q_j), \quad k=1,2,\dots,M; j=1,2,\dots,N$$

```
emission_probability = {
    'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
    'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
}
```

是在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率.

HMM – 原理 – 基本概念

- 初始状态概率

π 是初始状态概率向量:

```
start_probability = {'Rainy': 0.6, 'Sunny': 0.4}
```

$$\pi = (\pi_i)$$

其中,

$$\pi_i = P(i_1 = q_i), \quad i = 1, 2, \dots, N$$

是时刻 $t=1$ 处于状态 q_i 的概率.

隐马尔可夫模型由初始状态概率向量 π 、状态转移概率矩阵 A 和观测概率矩阵 B 决定. π 和 A 决定状态序列, B 决定观测序列. 因此, 隐马尔可夫模型 λ 可以用三元符号表示, 即

$$\lambda = (A, B, \pi) \tag{10.7}$$

- A, B, π 称为隐马尔可夫模型的三要素.

HMM – 原理 – 基本概念

- 齐次独立

从定义可知，隐马尔可夫模型作了两个基本假设：

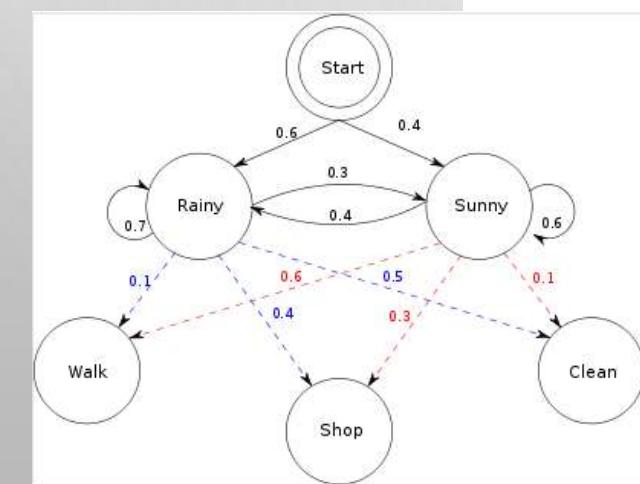
(1) 齐次马尔可夫性假设，即假设隐藏的马尔可夫链在任意时刻 t 的状态只依赖于其前一时刻的状态，与其他时刻的状态及观测无关，也与时刻 t 无关。

$$P(i_t | i_{t-1}, o_{t-1}, \dots, i_1, o_1) = P(i_t | i_{t-1}), \quad t = 1, 2, \dots, T \quad (10.8)$$

(2) 观测独立性假设，即假设任意时刻的观测只依赖于该时刻的马尔可夫链的状态，与其他观测及状态无关。

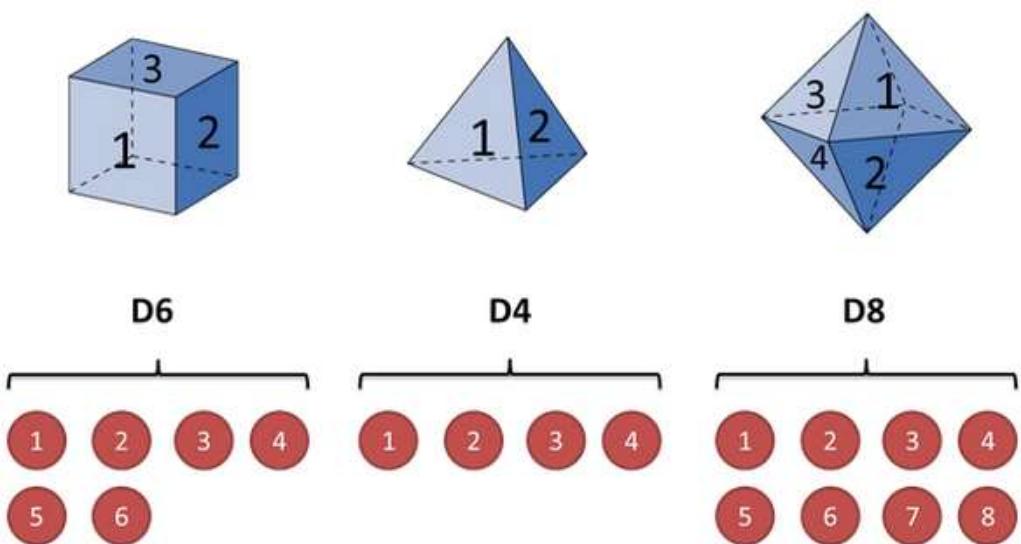
$$P(o_t | i_T, o_T, i_{T-1}, o_{T-1}, \dots, i_{t+1}, o_{t+1}, i_t, i_{t-1}, o_{t-1}, \dots, i_1, o_1) = P(o_t | i_t) \quad (10.9)$$

```
transition_probability = {
    'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
    'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
}
```

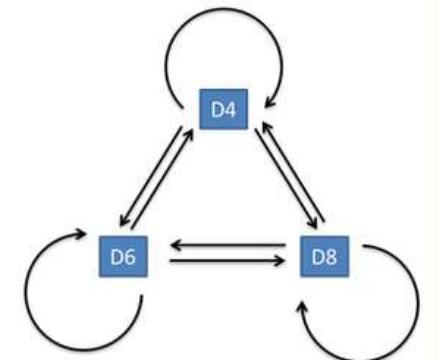


HMM – 其他栗子

假设我手里有三个不同的骰子。第一个骰子是我们平常见的骰子（称这个骰子为D6），6个面，每个面（1, 2, 3, 4, 5, 6）出现的概率是 $1/6$ 。第二个骰子是个四面体（称这个骰子为D4），每个面（1, 2, 3, 4）出现的概率是 $1/4$ 。第三个骰子有八个面（称这个骰子为D8），每个面（1, 2, 3, 4, 5, 6, 7, 8）出现的概率是 $1/8$ 。



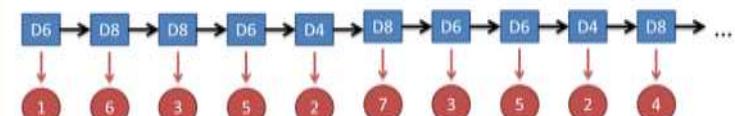
隐含状态转换关系示意图



假设我们开始掷骰子，我们先从三个骰子里挑一个，挑到每一个骰子的概率都是 $1/3$ 。然后我们掷骰子，得到一个数字，1, 2, 3, 4, 5, 6, 7, 8中的一个。不停的重复上述过程，我们会得到一串数字，每个数字都是1, 2, 3, 4, 5, 6, 7, 8中的一个。例如我们可能得到这么一串数字（掷骰子10次）：1 6 3 5 2 7 3 5 2 4

这串数字叫做可见状态链。但是在隐马尔可夫模型中，我们不仅仅有这么一串可见状态链，还有一串隐含状态链。在这个例子里，这串隐含状态链就是你用的骰子的序列。比如，隐含状态链有可能是：D6 D8 D8 D6 D4 D8 D6 D6 D4 D8

隐马尔可夫模型示意图



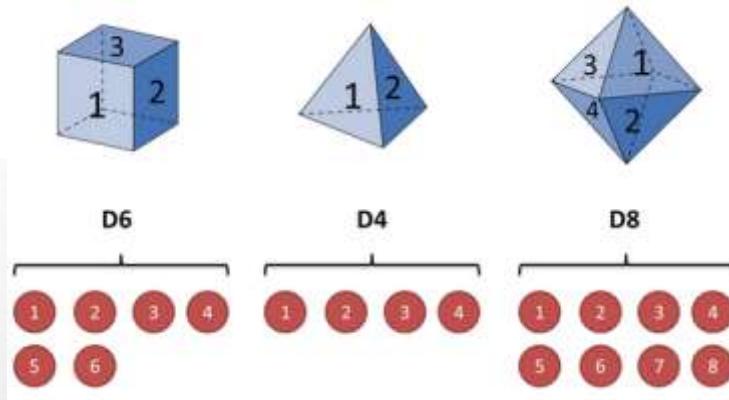
HMM – 其他栗子

隐马尔可夫模型由初始概率分布、状态转移概率分布以及观测概率分布确定。隐马尔可夫模型的形式定义如下：

设 Q 是所有可能的状态的集合， V 是所有可能的观测的集合。

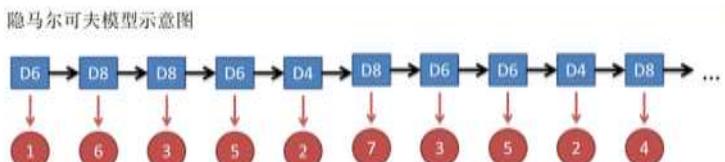
$$Q = \{q_1, q_2, \dots, q_N\}, \quad V = \{v_1, v_2, \dots, v_M\}$$

其中， N 是可能的状态数， M 是可能的观测数。



I 是长度为 T 的状态序列， O 是对应的观测序列。

$$I = (i_1, i_2, \dots, i_T), \quad O = (o_1, o_2, \dots, o_T)$$



本文链接地址

[HTTP://WWW.CNBLOGS.COM/SKYME/P/4651331.HTML](http://www.cnblogs.com/skyme/p/4651331.html)

HMM - 原理 - 基本概念

- 状态转移概率矩阵

A 是状态转移概率矩阵:

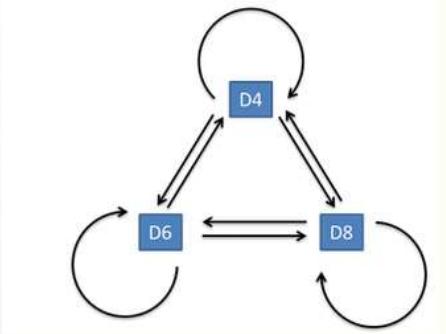
$$A = [a_{ij}]_{N \times N}$$

其中,

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i), \quad i=1,2,\dots,N; j=1,2,\dots,N$$

是在时刻 t 处于状态 q_i 的条件下在时刻 $t+1$ 转移到状态 q_j 的概率.

隐含状态转换关系示意图



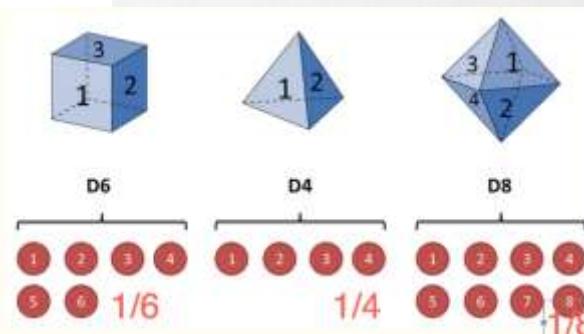
B 是观测概率矩阵:

$$B = [b_j(k)]_{N \times M}$$

其中,

$$b_j(k) = P(o_t = v_k | i_t = q_j), \quad k=1,2,\dots,M; j=1,2,\dots,N$$

是在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率.



HMM – 原理 – 基本概念

- 初始状态概率

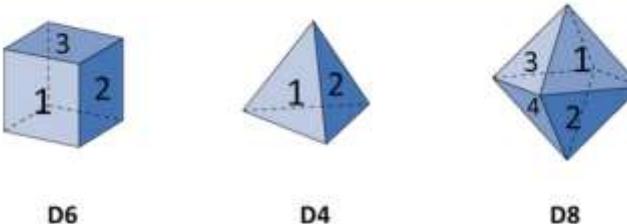
π 是初始状态概率向量:

$$\pi = (\pi_i)$$

其中,

$$\pi_i = P(i_1 = q_i), \quad i = 1, 2, \dots, N$$

是时刻 $t=1$ 处于状态 q_i 的概率.



挑到每一个骰子的概率都是 $1/3$.

隐马尔可夫模型由初始状态概率向量 π 、状态转移概率矩阵 A 和观测概率矩阵 B 决定. π 和 A 决定状态序列, B 决定观测序列. 因此, 隐马尔可夫模型 λ 可以用三元符号表示, 即

$$\lambda = (A, B, \pi) \tag{10.7}$$

- A, B, π 称为隐马尔可夫模型的三要素.

HMM – 原理 – 基本概念

• 判别模型 (DISCRIMINATIVE MODEL)

关注的核心点是数据的条件概率 $p(label \mid features)$

就开始时候给的例子，建立模型的目标是：

$$\arg \max_y p(y|x)$$

这个公式的解释是，已知特征求类标，使得类标y最靠谱（即使得其正确的概率最大）

• 生成模型 (GENERATIVE MODEL)

关注的核心点是数据的联合概率 $p(label, features)$ ，其中 $p(label, features) = p(features \mid label) * p(label)$

* $p(label)$

就开始时候给的例子，建立模型的目标是：

$$\begin{aligned} \arg \max_y p(y|x) &= \arg \max_y \frac{p(x|y)p(y)}{p(x)} \\ &= \arg \max_y p(x|y)p(y). \end{aligned}$$

这个模型的终极目标和判别模型一致，即已知特征求类标，使得类标y最靠谱（即使得其正确的概率最大）

区别在于模型的目标被换做了另一种形式，公式变换中 $p(x)$ 被去掉是因为对于正负例来说， $p(x)$ 是个一致恒定的概率，相当于常量。

HMM – 原理 – 基本概念

- 三个基本问题

(1) 概率计算问题. 给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \dots, o_T)$, 计算在模型 λ 下观测序列 O 出现的概率 $P(O | \lambda)$.

Alice和Bob通了三天电话后发现第一天Bob去散步了，第二天他去购物了，第三天他清理房间了。

Alice现在有两个问题：这个观察序列“walk, shop, clean”的总的概率是多少？

(2) 学习问题. 已知观测序列 $O = (o_1, o_2, \dots, o_T)$, 估计模型 $\lambda = (A, B, \pi)$ 参数, 使得在该模型下观测序列概率 $P(O | \lambda)$ 最大. 即用极大似然估计的方法估计参数.

模型参数 (A, B, pi)

(3) 预测问题, 也称为解码 (decoding) 问题. 已知模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \dots, o_T)$, 求对给定观测序列条件概率 $P(I | O)$ 最大的状态序列 $I = (i_1, i_2, \dots, i_T)$. 即给定观测序列, 求最有可能的对应的状态序列.

能解释这个观测的状态序列是什么? “Sunny, Sunny, Rainy” or “Sunny, Rainy, Rainy” ?

HMM – 原理 – 概率计算问题

(1) 概率计算问题. 给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \dots, o_T)$, 计算在模型 λ 下观测序列 O 出现的概率 $P(O | \lambda)$.

Alice 和 Bob 通了三天电话后发现第一天 Bob 去散步了, 第二天他去购物了, 第三天他清理房间了。
Alice 现在有两个问题: 这个观察序列 “walk, shop, clean”的总的的概率是多少?

10.2.1 直接计算法

给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \dots, o_T)$, 计算观测序列 O 出现的概率 $P(O | \lambda)$. 最直接的方法是按概率公式直接计算. 通过列举所有可能的长度为 T 的状态序列 $I = (i_1, i_2, \dots, i_T)$, 求各个状态序列 I 与观测序列 $O = (o_1, o_2, \dots, o_T)$ 的联合概率 $P(O, I | \lambda)$, 然后对所有可能的状态序列求和, 得到 $P(O | \lambda)$.

状态序列 $I = (i_1, i_2, \dots, i_T)$ 的概率是

$$P(I | \lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdots a_{i_{T-1} i_T} \quad (10.10)$$

对固定的状态序列 $I = (i_1, i_2, \dots, i_T)$, 观测序列 $O = (o_1, o_2, \dots, o_T)$ 的概率是 $P(O | I, \lambda)$,

$$P(O | I, \lambda) = b_{i_1}(o_1) b_{i_2}(o_2) \cdots b_{i_T}(o_T) \quad (10.11)$$

O 和 I 同时出现的联合概率为

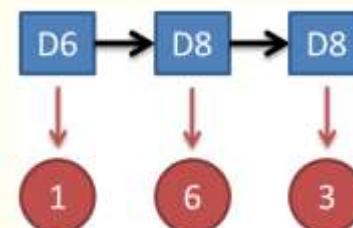
$$\begin{aligned} P(O, I | \lambda) &= P(O | I, \lambda) P(I | \lambda) \\ &= \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T) \end{aligned} \quad (10.12)$$

然后, 对所有可能的状态序列 I 求和, 得到观测序列 O 的概率 $P(O | \lambda)$, 即

$$\begin{aligned} P(O | \lambda) &= \sum_I P(O | I, \lambda) P(I | \lambda) \\ &= \sum_{i_1, i_2, \dots, i_T} \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T) \end{aligned} \quad (10.13)$$

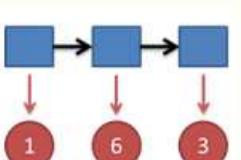
但是, 利用公式 (10.13) 计算量很大, 是 $O(TN^T)$ 阶的, 这种算法不可行.

知道骰子有几种, 每种骰子是什么, 每次掷的都是什么骰子. 根据掷骰子掷出的结果, 求产生这个结果的概率。



解法无非就是概率相乘:

$$\begin{aligned} P &= P(D6) * P(D6 \rightarrow 1) * P(D6 \rightarrow D8) * P(D8 \rightarrow 6) * P(D8 \rightarrow D8) * P(D8 \rightarrow 3) \\ &= \frac{1}{3} * \frac{1}{6} * \frac{1}{3} * \frac{1}{8} * \frac{1}{3} * \frac{1}{8} \end{aligned}$$



HMM – 原理 – 概率计算问题

- 前向算法

定义 10.2 (前向概率) 给定隐马尔可夫模型 λ , 定义到时刻 t 部分观测序列为 o_1, o_2, \dots, o_t 且状态为 q_i 的概率为前向概率, 记作

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda) \quad (10.14)$$

可以递推地求得前向概率 $\alpha_t(i)$ 及观测序列概率 $P(O | \lambda)$.

算法 10.2 (观测序列概率的前向算法)

输入: 隐马尔可夫模型 λ , 观测序列 O ;

输出: 观测序列概率 $P(O | \lambda)$.

(1) 初值

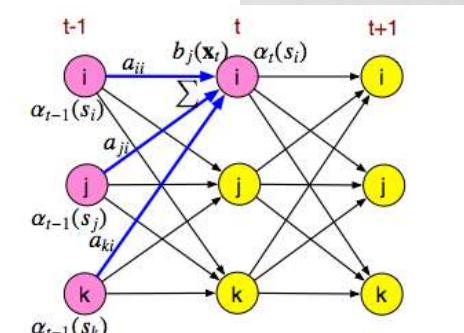
$$\alpha_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \dots, N$$

(2) 递推 对 $t = 1, 2, \dots, T - 1$,

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^N \alpha_t(j) a_{ji} \right] b_i(o_{t+1}), \quad i = 1, 2, \dots, N$$

(3) 终止

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (10.17) \blacksquare$$



HMM – 原理 – 概率计算问题

- 前向算法

$$\alpha_T(i) = P(o_1, o_2, \dots, o_T, i_T = q_i | \lambda)$$

所以

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

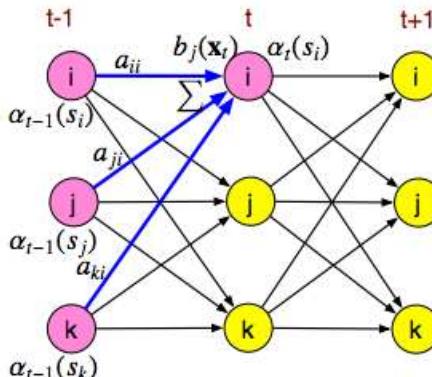


图 10.1 前向概率的递推公式

$P(O | \lambda)$ 的计算量是 $O(N^2T)$ 阶的，而不是直接计算的 $O(TN^T)$ 阶.

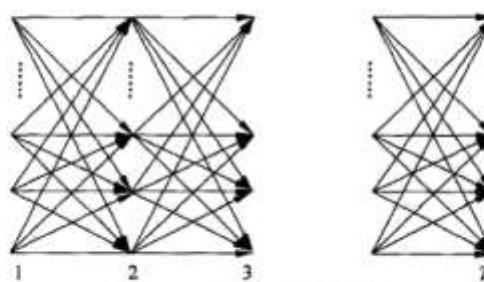


图 10.2 观测序列路径结构

HMM – 原理 – 概率计算问题

- 前向算法

例 10.2 考虑盒子和球模型 $\lambda = (A, B, \pi)$, 状态集合 $Q = \{1, 2, 3\}$, 观测集合 $V = \{\text{红, 白}\}$,

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \quad \pi = (0.2, 0.4, 0.4)^T$$

设 $T = 3$, $O = (\text{红, 白, 红})$, 试用前向算法计算 $P(O | \lambda)$.

解 按照算法 10.2

(1) 计算初值

$$\alpha_1(1) = \pi_1 b_1(o_1) = 0.10$$

$$\alpha_1(2) = \pi_2 b_2(o_1) = 0.16$$

$$\alpha_1(3) = \pi_3 b_3(o_1) = 0.28$$

(2) 递推计算

$$\alpha_2(1) = \left[\sum_{i=1}^3 \alpha_1(i) a_{i1} \right] b_1(o_2) = 0.154 \times 0.5 = 0.077$$

$$\alpha_2(2) = \left[\sum_{i=1}^3 \alpha_1(i) a_{i2} \right] b_2(o_2) = 0.184 \times 0.6 = 0.1104$$

$$\alpha_2(3) = \left[\sum_{i=1}^3 \alpha_1(i) a_{i3} \right] b_3(o_2) = 0.202 \times 0.3 = 0.0606$$

$$\alpha_3(1) = \left[\sum_{i=1}^3 \alpha_2(i) a_{i1} \right] b_1(o_3) = 0.04187$$

$$\alpha_3(2) = \left[\sum_{i=1}^3 \alpha_2(i) a_{i2} \right] b_2(o_3) = 0.03551$$

$$\alpha_3(3) = \left[\sum_{i=1}^3 \alpha_2(i) a_{i3} \right] b_3(o_3) = 0.05284$$

(3) 终止

$$P(O | \lambda) = \sum_{i=1}^3 \alpha_3(i) = 0.13022$$

HMM – 原理 – 概率计算问题

- 后向算法

定义 10.3 (后向概率) 给定隐马尔可夫模型 λ , 定义在时刻 t 状态为 q_i 的条件下, 从 $t+1$ 到 T 的部分观测序列为 $o_{t+1}, o_{t+2}, \dots, o_T$ 的概率为后向概率, 记作

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | i_t = q_i, \lambda) \quad (10.18)$$

可以用递推的方法求得后向概率 $\beta_t(i)$ 及观测序列概率 $P(O|\lambda)$.

算法 10.3 (观测序列概率的后向算法)

输入: 隐马尔可夫模型 λ , 观测序列 O ;

输出: 观测序列概率 $P(O|\lambda)$.

(1)

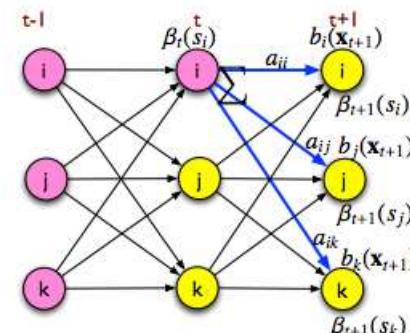
$$\beta_T(i) = 1, \quad i = 1, 2, \dots, N \quad (10.19)$$

(2) 对 $t = T-1, T-2, \dots, 1$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad i = 1, 2, \dots, N \quad (10.20)$$

(3)

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i) \quad (10.21) \blacksquare$$



HMM – 原理 – 概率计算问题

- 前后向概率的统一形式

利用前向概率和后向概率的定义可以将观测序列概率 $P(O | \lambda)$ 统一写成

$$P(O | \lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad t = 1, 2, \dots, T-1 \quad (10.22)$$

此式当 $t=1$ 和 $t=T-1$ 时分别为式(10.17)和式(10.21).

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (10.17)$$

$$P(O | \lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i) \quad (10.21)$$

HMM – 原理 – 概率计算问题

- 一些概率和期望的计算

利用前向概率和后向概率，可以得到关于单个状态和两个状态概率的计算公式。

- 给定模型 λ 和观测 O ，在时刻 t 处于状态 q_i 的概率。记

$$\gamma_t(i) = P(i_t = q_i | O, \lambda) \quad (10.23)$$

可以通过前向后向概率计算。事实上，

$$\gamma_t(i) = P(i_t = q_i | O, \lambda) = \frac{P(i_t = q_i, O | \lambda)}{P(O | \lambda)}$$

由前向概率 $\alpha_t(i)$ 和后向概率 $\beta_t(i)$ 定义可知：

$$\alpha_t(i)\beta_t(i) = P(i_t = q_i, O | \lambda)$$

于是得到：

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O | \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \quad (10.24)$$

HMM – 原理 – 概率计算问题

- 一些概率和期望的计算

2. 给定模型 λ 和观测 O , 在时刻 t 处于状态 q_i 且在时刻 $t+1$ 处于状态 q_j 的概率. 记

$$\xi_t(i, j) = P(i_t = q_i, i_{t+1} = q_j | O, \lambda) \quad (10.25)$$

可以通过前向后向概率计算:

$$\xi_t(i, j) = \frac{P(i_t = q_i, i_{t+1} = q_j, O | \lambda)}{P(O | \lambda)} = \frac{P(i_t = q_i, i_{t+1} = q_j, O | \lambda)}{\sum_{i=1}^N \sum_{j=1}^N P(i_t = q_i, i_{t+1} = q_j, O | \lambda)}$$

而

$$P(i_t = q_i, i_{t+1} = q_j, O | \lambda) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

HMM – 原理 – 概率计算问题

- 一些概率和期望的计算

3. 将 $\gamma_t(i)$ 和 $\xi_t(i, j)$ 对各个时刻 t 求和，可以得到一些有用的期望值：

(1) 在观测 O 下状态 i 出现的期望值

$$\sum_{t=1}^T \gamma_t(i) \quad (10.27)$$

(2) 在观测 O 下由状态 i 转移的期望值

$$\sum_{t=1}^{T-1} \gamma_t(i) \quad (10.28)$$

(3) 在观测 O 下由状态 i 转移到状态 j 的期望值

$$\sum_{t=1}^{T-1} \xi_t(i, j) \quad (10.29)$$

HMM – 原理 – 学习算法

(2) 学习问题. 已知观测序列 $O = (o_1, o_2, \dots, o_T)$, 估计模型 $\lambda = (A, B, \pi)$ 参数, 使得在该模型下观测序列概率 $P(O | \lambda)$ 最大. 即用极大似然估计的方法估计参数.

模型参数 (A, B, π)

10.3.1 监督学习方法

假设已给训练数据包含 S 个长度相同的观测序列和对应的状态序列 $\{(O_1, I_1), (O_2, I_2), \dots, (O_S, I_S)\}$, 那么可以利用极大似然估计法来估计隐马尔可夫模型的参数. 具体方法如下.

1. 转移概率 a_{ij} 的估计

设样本中时刻 t 处于状态 i 时刻 $t+1$ 转移到状态 j 的频数为 A_{ij} , 那么状态转移概率 a_{ij} 的估计是

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^N A_{ij}}, \quad i=1, 2, \dots, N; \quad j=1, 2, \dots, N \quad (10.30)$$

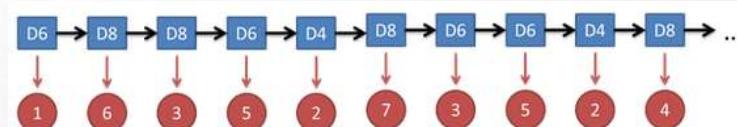
2. 观测概率 $b_j(k)$ 的估计

设样本中状态为 j 并观测为 k 的频数是 B_{jk} , 那么状态为 j 观测为 k 的概率 $b_j(k)$ 的估计是

$$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^M B_{jk}}, \quad j=1, 2, \dots, N; \quad k=1, 2, \dots, M \quad (10.31)$$

3. 初始状态概率 π_i 的估计 $\hat{\pi}_i$ 为 S 个样本中初始状态为 q_i 的频率

由于监督学习需要使用训练数据, 而人工标注训练数据往往代价很高, 有时就会利用非监督学习的方法.



HMM – 原理 – 学习算法

Expectation-Maximization (EM) algorithm 

"MLE (or MAP) when some data is missing"

Given: $x = (x_1, \dots, x_n)$ ^{typ exp fam} $X = (X_1, \dots, X_n)$ ^{exp fam}

Model: $(X, z) \sim p_\theta$ for some (unknown) $\theta \in \Theta$.

Goal: $\theta_{\text{MLE}} \in \arg\max_{\theta} p_\theta(x)$ 

Issue: $p_\theta(x) = \prod_i p_\theta(x_i | z)$ difficult to maximize.

Alg: Initialize $\theta_0 \in \Theta$.
For $t=0, 1, 2, \dots$ (until convergence):
• E-step: $Q(\theta, \theta_t) = E_{\theta_t}(\log p_\theta(X, z) | X=x)$.
• M-step: $\theta_{t+1} \in \arg\max_{\theta} Q(\theta, \theta_t)$.

Pros: • $p_{\theta_{t+1}}(x) \geq p_{\theta_t}(x)$.
• Works well in practice.
Cons: • Not guaranteed to give θ_{MLE} .
• MLE may overfit, \rightarrow EM.
• Convergence can be slow.
• Specialized to exp fam

HMM – 原理 – 学习算法

- BAUM-WELCH算法

假设给定训练数据只包含 S 个长度为 T 的观测序列 $\{O_1, O_2, \dots, O_s\}$ 而没有对应的状态序列，目标是学习隐马尔可夫模型 $\lambda = (A, B, \pi)$ 的参数。我们将观测序列数据看作观测数据 O ，状态序列数据看作不可观测的隐数据 I ，那么隐马尔可夫模型事实上是一个含有隐变量的概率模型

$$P(O | \lambda) = \sum_I P(O | I, \lambda) P(I | \lambda) \quad (10.32)$$

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

它的参数学习可以由 EM 算法实现.

1. 确定完全数据的对数似然函数

所有观测数据写成 $O = (o_1, o_2, \dots, o_T)$, 所有隐数据写成 $I = (i_1, i_2, \dots, i_T)$, 完全数据是 $(O, I) = (o_1, o_2, \dots, o_T, i_1, i_2, \dots, i_T)$. 完全数据的对数似然函数是 $\log P(O, I | \lambda)$.

2. EM 算法的 E 步: 求 Q 函数 $Q(\lambda, \bar{\lambda})$ ^①

$$Q(\lambda, \bar{\lambda}) = \sum_I \log P(O, I | \lambda) P(O, I | \bar{\lambda}) \quad (10.33)$$

其中, $\bar{\lambda}$ 是隐马尔可夫模型参数的当前估计值, λ 是要极大化的隐马尔可夫模型参数.

① 按照 Q 函数的定义

$$Q(\lambda, \bar{\lambda}) = E_I [\log P(O, I | \lambda) | O, \bar{\lambda}]$$

式 (10.33) 略去了对 λ 而言的常数因子 $1/P(O | \bar{\lambda})$.

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

2. EM 算法的 E 步: 求 Q 函数 $Q(\lambda, \bar{\lambda})$ ^①

$$Q(\lambda, \bar{\lambda}) = \sum_I \log P(O, I | \lambda) P(O, I | \bar{\lambda}) \quad (10.33)$$

其中, $\bar{\lambda}$ 是隐马尔可夫模型参数的当前估计值, λ 是要极大化的隐马尔可夫模型参数.

$$P(O, I | \lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T)$$

于是函数 $Q(\lambda, \bar{\lambda})$ 可以写成:

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_I \log \pi_{i_1} P(O, I | \bar{\lambda}) \\ &\quad + \sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}} \right) P(O, I | \bar{\lambda}) + \sum_I \left(\sum_{t=1}^T \log b_{i_t}(o_t) \right) P(O, I | \bar{\lambda}) \end{aligned} \quad (10.34)$$

式中求和都是对所有训练数据的序列总长度 T 进行的.

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

3. EM 算法的 M 步：极大化 Q 函数 $Q(\lambda, \bar{\lambda})$ 求模型参数 A, B, π

由于要极大化的参数在式 (10.34) 中单独地出现在 3 个项中，所以只需对各项分别极大化。

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_I \log \pi_{i_1} P(O, I | \bar{\lambda}) \\ &\quad + \sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}} \right) P(O, I | \bar{\lambda}) + \sum_I \left(\sum_{t=1}^T \log b_{i_t}(o_t) \right) P(O, I | \bar{\lambda}) \end{aligned} \quad (10.34)$$

HMM – 原理 – 学习算法

- BAUM-WELCH算法

(1) 式 (10.34) 的第 1 项可以写成:

$$\sum_I \log \pi_{i_0} P(O, I | \bar{\lambda}) = \sum_{i=1}^N \log \pi_i P(O, i_1 = i | \bar{\lambda})$$

注意到 π_i 满足约束条件 $\sum_{i=1}^N \pi_i = 1$, 利用拉格朗日乘子法, 写出拉格朗日函数:

$$\sum_{i=1}^N \log \pi_i P(O, i_1 = i | \bar{\lambda}) + \gamma \left(\sum_{i=1}^N \pi_i - 1 \right)$$

对其求偏导数并令结果为 0

$$\frac{\partial}{\partial \pi_i} \left[\sum_{i=1}^N \log \pi_i P(O, i_1 = i | \bar{\lambda}) + \gamma \left(\sum_{i=1}^N \pi_i - 1 \right) \right] = 0 \quad (10.35)$$

得

$$P(O, i_1 = i | \bar{\lambda}) + \gamma \pi_i = 0$$

对 i 求和得到 γ

$$\gamma = -P(O | \bar{\lambda})$$

代入式 (10.35) 即得

$$\pi_i = \frac{P(O, i_1 = i | \bar{\lambda})}{P(O | \bar{\lambda})} \quad (10.36)$$

HMM – 原理 – 学习算法

- BAUM-WELCH算法

(2) 式(10.34)的第2项可以写成

$$\sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}} \right) P(O, I | \bar{\lambda}) = \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} \log a_{ij} P(O, i_t = i, i_{t+1} = j | \bar{\lambda})$$

类似第1项，应用具有约束条件 $\sum_{j=1}^N a_{ij} = 1$ 的拉格朗日乘子法可以求出

$$a_{ij} = \frac{\sum_{t=1}^{T-1} P(O, i_t = i, i_{t+1} = j | \bar{\lambda})}{\sum_{t=1}^{T-1} P(O, i_t = i | \bar{\lambda})} \quad (10.37)$$

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

(3) 式 (10.34) 的第 3 项为

$$\sum_I \left(\sum_{t=1}^T \log b_{i_t}(o_t) \right) P(O, I | \bar{\lambda}) = \sum_{j=1}^N \sum_{t=1}^T \log b_j(o_t) P(O, i_t = j | \bar{\lambda})$$

同样用拉格朗日乘子法, 约束条件是 $\sum_{k=1}^M b_j(k) = 1$. 注意, 只有在 $o_t = v_k$ 时 $b_j(o_t)$ 对 $b_j(k)$ 的偏导数才不为 0, 以 $I(o_t = v_k)$ 表示. 求得

$$b_j(k) = \frac{\sum_{t=1}^T P(O, i_t = j | \bar{\lambda}) I(o_t = v_k)}{\sum_{t=1}^T P(O, i_t = j | \bar{\lambda})} \quad (10.38)$$

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

10.3.3 Baum-Welch 模型参数估计公式

将式 (10.36) ~ 式 (10.38) 中的各概率分别用 $\gamma_t(i)$, $\xi_t(i, j)$ 表示, 则可将相应的公式写成:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (10.39)$$

$$b_j(k) = \frac{\sum_{t=1, o_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (10.40)$$

$$\pi_i = \gamma_1(i) \quad (10.41)$$

其中, $\gamma_t(i)$, $\xi_t(i, j)$ 分别由式 (10.24) 及式 (10.26) 给出. 式 (10.39) ~ 式 (10.41) 就是 Baum-Welch 算法 (Baum-Welch algorithm), 它是 EM 算法在隐马尔可夫模型学习中的具体实现, 由 Baum 和 Welch 提出.

HMM – 原理 – 学习算法

- BAUM-WELCH 算法

算法 10.4 (Baum-Welch 算法)

输入：观测数据 $O = (o_1, o_2, \dots, o_T)$ ；

输出：隐马尔可夫模型参数.

(1) 初始化

对 $n=0$ ，选取 $a_{ij}^{(0)}$, $b_j(k)^{(0)}$, $\pi_i^{(0)}$ ，得到模型 $\lambda^{(0)} = (A^{(0)}, B^{(0)}, \pi^{(0)})$.

(2) 递推. 对 $n=1, 2, \dots$,

$$a_{ij}^{(n+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j(k)^{(n+1)} = \frac{\sum_{\substack{t=1, o_t=v_k \\ t=1}}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}$$
$$\pi_i^{(n+1)} = \gamma_1(i)$$

右端各值按观测 $O = (o_1, o_2, \dots, o_T)$ 和模型 $\lambda^{(n)} = (A^{(n)}, B^{(n)}, \pi^{(n)})$ 计算. 式中 $\gamma_t(i)$, $\xi_t(i, j)$ 由式 (10.24) 和式 (10.26) 给出.

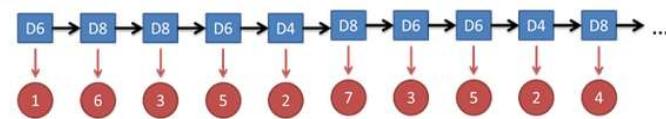
(3) 终止. 得到模型参数 $\lambda^{(n+1)} = (A^{(n+1)}, B^{(n+1)}, \pi^{(n+1)})$. ■

HMM – 原理 – 预测算法

(3) 预测问题, 也称为解码 (decoding) 问题. 已知模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \dots, o_T)$, 求对给定观测序列条件概率 $P(I|O)$ 最大的状态序列 $I = (i_1, i_2, \dots, i_T)$. 即给定观测序列, 求最有可能的对应的状态序列.

能解释这个观测的状态序列是什么? "Sunny, Sunny, Rainy" or "Sunny, Rainy, Rainy"?

10.4.1 近似算法



近似算法的想法是, 在每个时刻 t 选择在该时刻最有可能出现的状态 i_t^* , 从而得到一个状态序列 $I^* = (i_1^*, i_2^*, \dots, i_T^*)$, 将它作为预测的结果.

给定隐马尔可夫模型 λ 和观测序列 O , 在时刻 t 处于状态 q_i 的概率 $\gamma_t(i)$ 是

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} \quad (10.42)$$

在每一时刻 t 最有可能的状态 i_t^* 是

$$i_t^* = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \quad t = 1, 2, \dots, T \quad (10.43)$$

从而得到状态序列 $I^* = (i_1^*, i_2^*, \dots, i_T^*)$.

近似算法的优点是计算简单, 其缺点是不能保证预测的状态序列整体是最有可能的状态序列, 因为预测的状态序列可能有实际不发生的部分. 事实上, 上述方法得到的状态序列中可能存在转移概率为 0 的相邻状态, 即对某些 i, j , $a_{ij} = 0$ 时. 尽管如此, 近似算法仍然是有用的.

HMM – 原理 – 预测算法

- VITERBI算法

维特比算法实际是用动态规划解隐马尔可夫模型预测问题，即用动态规划 (dynamic programming) 求概率最大路径 (最优路径). 这时一条路径对应着一个状态序列.

首先导入两个变量 δ 和 ψ . 定义在时刻 t 状态为 i 的所有单个路径 (i_1, i_2, \dots, i_t) 中概率最大值为

$$\delta_t(i) = \max_{i_1, i_2, \dots, i_{t-1}} P(i_t = i, i_{t-1}, \dots, i_1, o_t, \dots, o_1 | \lambda), \quad i = 1, 2, \dots, N \quad (10.44)$$

由定义可得变量 δ 的递推公式:

$$\begin{aligned} \delta_{t+1}(i) &= \max_{i_1, i_2, \dots, i_t} P(i_{t+1} = i, i_t, \dots, i_1, o_{t+1}, \dots, o_1 | \lambda) \\ &= \max_{1 \leq j \leq N} [\delta_t(j) a_{ji}] b_i(o_{t+1}), \quad i = 1, 2, \dots, N; \quad t = 1, 2, \dots, T-1 \end{aligned} \quad (10.45)$$

定义在时刻 t 状态为 i 的所有单个路径 $(i_1, i_2, \dots, i_{t-1}, i)$ 中概率最大的路径的第 $t-1$ 个结点为

$$\psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], \quad i = 1, 2, \dots, N \quad (10.46)$$

HMM – 原理 – 预测算法

- VITERBI算法

算法 10.5 (维特比算法)

输入：模型 $\lambda = (A, B, \pi)$ 和观测 $O = (o_1, o_2, \dots, o_T)$ ；

输出：最优路径 $I^* = (i_1^*, i_2^*, \dots, i_T^*)$.

(1) 初始化

$$\delta_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \dots, N$$

$$\psi_1(i) = 0, \quad i = 1, 2, \dots, N$$

(2) 递推. 对 $t = 2, 3, \dots, T$

$$\delta_t(i) = \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] b_i(o_t), \quad i = 1, 2, \dots, N$$

$$\psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], \quad i = 1, 2, \dots, N$$

(3) 终止

$$P^* = \max_{1 \leq i \leq N} \delta_T(i)$$

$$i_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

(4) 最优路径回溯. 对 $t = T-1, T-2, \dots, 1$

$$i_t^* = \psi_{t+1}(i_{t+1}^*)$$

求得最优路径 $I^* = (i_1^*, i_2^*, \dots, i_T^*)$.

HMM – 原理 – 预测算法

- VITERBI算法

例 10.2 考虑盒子和球模型 $\lambda = (A, B, \pi)$, 状态集合 $Q = \{1, 2, 3\}$, 观测集合 $V = \{\text{红}, \text{白}\}$,

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}, \quad \pi = (0.2, 0.4, 0.4)^T$$

已知观测序列 $O = (\text{红}, \text{白}, \text{红})$, 试求最优状态序列, 即最优路径 $I^* = (i_1^*, i_2^*, i_3^*)$.

解 如图 10.4 所示, 要在所有可能的路径中选择一条最优路径, 按以下步骤处理:

(1) 初始化. 在 $t=1$ 时, 对每一个状态 i , $i=1, 2, 3$, 求状态为 i 观测 o_1 为红的概率, 记此概率为 $\delta_1(i)$, 则

$$\delta_1(i) = \pi_i b_i(o_1) = \pi_i b_i(\text{红}), \quad i=1, 2, 3$$

代入实际数据

$$\delta_1(1) = 0.10, \quad \delta_1(2) = 0.16, \quad \delta_1(3) = 0.28$$

记 $\psi_1(i) = 0$, $i=1, 2, 3$.

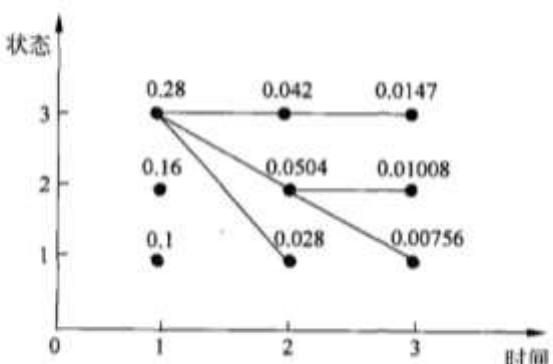


图 10.4 求最优路径

(2) 在 $t=2$ 时, 对每个状态 i , $i=1, 2, 3$, 求在 $t=1$ 时状态为 j 观测为红并在 $t=2$ 时状态为 i 观测 o_2 为白的路径的最大概率, 记此最大概率为 $\delta_2(i)$, 则

$$\delta_2(i) = \max_{1 \leq j \leq 3} [\delta_1(j)a_{ji}]b_i(o_2)$$

同时, 对每个状态 i , $i=1, 2, 3$, 记录概率最大路径的前一个状态 j :

$$\psi_2(i) = \arg \max_{1 \leq j \leq 3} [\delta_1(j)a_{ji}], \quad i=1, 2, 3$$

计算:

$$\begin{aligned} \delta_2(1) &= \max_{1 \leq j \leq 3} [\delta_1(j)a_{j1}]b_1(o_2) \\ &= \max_j \{0.10 \times 0.5, 0.16 \times 0.3, 0.28 \times 0.2\} \times 0.5 \\ &= 0.028 \end{aligned}$$

$$\begin{aligned} \psi_2(1) &= 3 \\ \delta_2(2) &= 0.0504, \quad \psi_2(2) = 3 \\ \delta_2(3) &= 0.042, \quad \psi_2(3) = 3 \end{aligned}$$

同样, 在 $t=3$ 时,

$$\begin{aligned} \delta_3(i) &= \max_{1 \leq j \leq 3} [\delta_2(j)a_{ji}]b_i(o_3) \\ \psi_3(i) &= \arg \max_{1 \leq j \leq 3} [\delta_2(j)a_{ji}] \\ \delta_3(1) &= 0.00756, \quad \psi_3(1) = 2 \\ \delta_3(2) &= 0.01008, \quad \psi_3(2) = 2 \\ \delta_3(3) &= 0.0147, \quad \psi_3(3) = 3 \end{aligned}$$

(3) 以 P^* 表示最优路径的概率, 则

$$P^* = \max_{1 \leq i \leq 3} \delta_3(i) = 0.0147$$

最优路径的终点是 i_3^* :

$$i_3^* = \arg \max_i [\delta_3(i)] = 3$$

(4) 由最优路径的终点 i_3^* , 逆向找到 i_2^*, i_1^* :

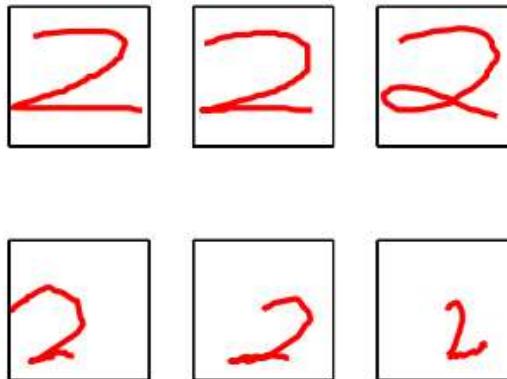
$$\text{在 } t=2 \text{ 时}, \quad i_2^* = \psi_3(i_3^*) = \psi_3(3) = 3$$

$$\text{在 } t=1 \text{ 时}, \quad i_1^* = \psi_2(i_2^*) = \psi_2(3) = 3$$

于是求得最优路径, 即最优状态序列 $I^* = (i_1^*, i_2^*, i_3^*) = (3, 3, 3)$.

HMM – 应用 – 在线手写

Top row: examples of on-line handwritten digits. Bottom row: synthetic digits sampled generatively from a left-to-right hidden Markov model that has been trained on a data set of 45 handwritten digits.



The most natural unit of handwriting is a letter. A letter is represented by a 7-state left-to-right HMM. The HMM model is illustrated in Figure 2-3.

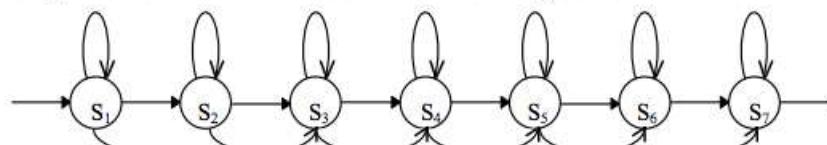


Figure 2-3: A 7-state HMM for a letter.

The left-to-right type of HMM, a special class of HMMs, have an additional property that the state index is non-decreasing as the time increases, i.e.

$$a_{ij} = P(Q_n=S_j|Q_{n-1}=S_i) = 0, \quad i > j. \quad (2.31)$$

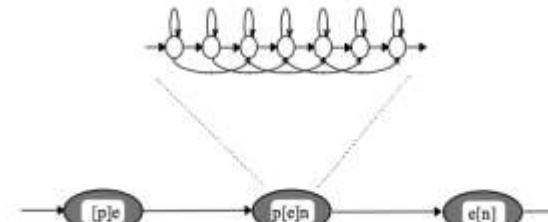


Figure 2-5: Illustration of an HMM modeling of the word "pen", consists of three 7-state HMMs for the trigraph "/p/e/", "p/e/n", and "e/n/" .

HMM – 应用 – 中文分词

本质上讲，分词可以看做一个为文本中每个字符分类的过程，例如我们现在定义两个类别：E代表词尾词，B代表非词尾词，于是分词“你/现在/应该/去/幼儿园/了”可以表达为：你E现B在E应B该E去E幼B儿B园E了B，分类完成后只需要对结果进行“解读”就可以得到分词结果了。

那么如何找到这个分类序列EBEBEEBBEB呢？我们可以求得所有可能出现的分类序列的出现概率然后选择其中概率最大的一个，用数学表达为：

嗷嗷、

$$\operatorname{argmax}_C P(C_1, C_2, C_3 \dots C_i) \quad (1)$$

其中C代表一个分类标识，这里，C属于{E,B} 进一步的：

[用HMM做中文分词：模型准备](#)

[用HMM做中文分词：前向算法和Viterbi算法的开销](#)

HMM – 应用 – 词性标注

我们以Brown语料库中的句子为例，词性标注的任务指的是，对于输入句子：

The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced " no evidence " that any irregularities took place .

需要为句子中的每个单词标上一个合适的词性标记，既输出含有词性标记句子：

The/at-tl Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta's/np\$ recent/jj primary/jj election/nn produced/vbn "/ no/at evidence/nn "/" that/cs any/dti irregularities/nns took/vbd place/nn ./.

关于词性标注歧义问题，对Brown语料库进行统计，按歧义程度排列的词型数目 (The number of word types in Brown corpus by degree of ambiguity) DeRose(1988)给出了如下的标记歧义表：

无歧义 (Unambiguous) 只有1个标记： 35,340

歧义 (Ambiguous) 有2-7个标记： 4,100

2个标记： 3,764

3个标记： 264

4个标记： 61

5个标记： 12

6个标记： 2

7个标记： 1

嗷、

HMM – 应用 – 其他资源

[52nlp HMM系列](#)

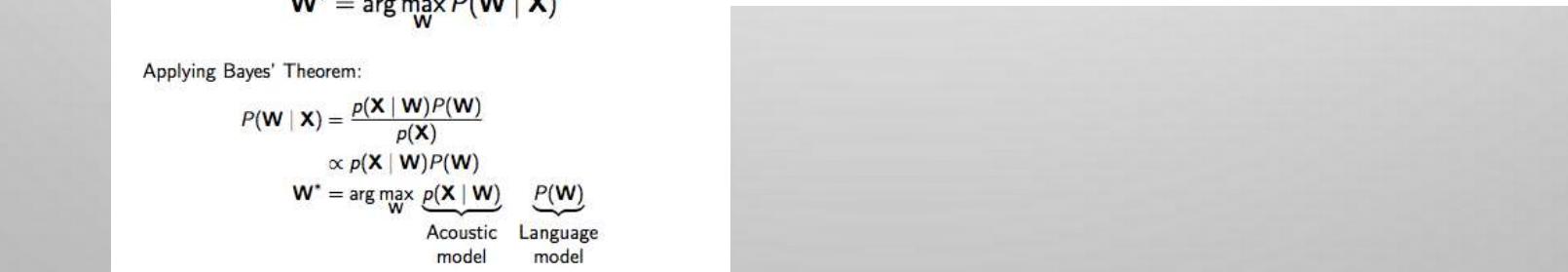
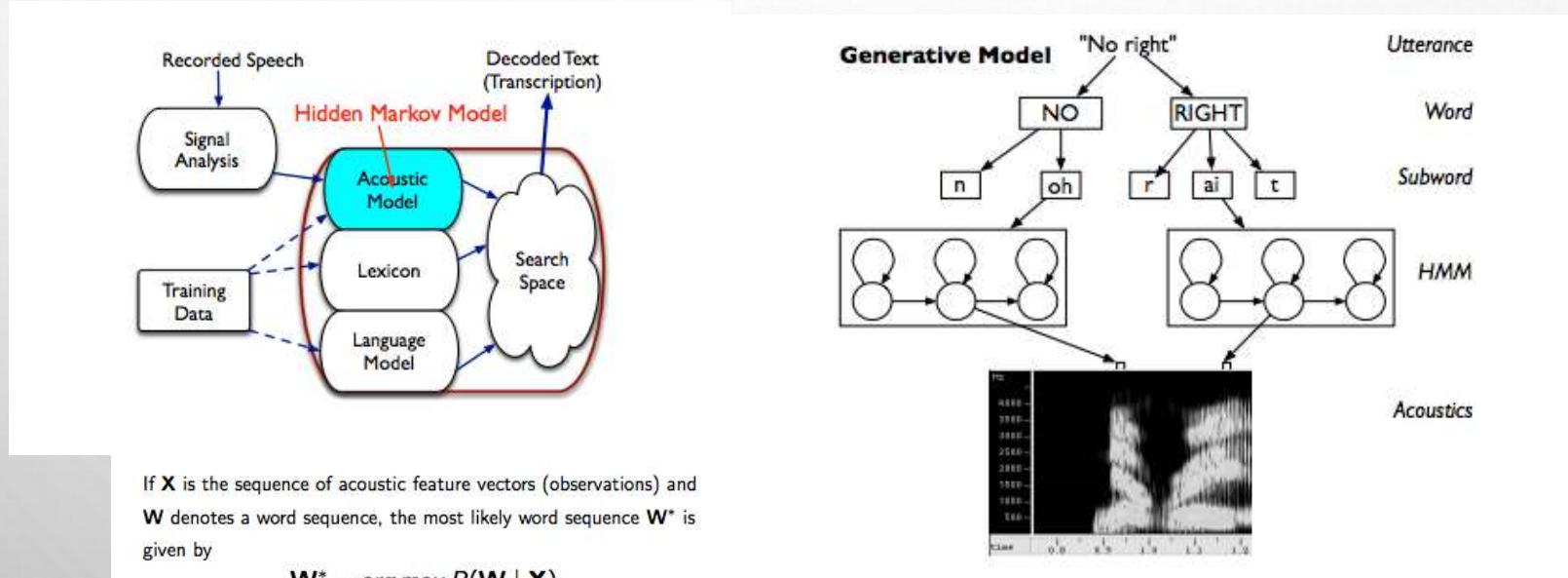
[Exercise: Using a Hidden Markov Model](#)

[开源实现](#)

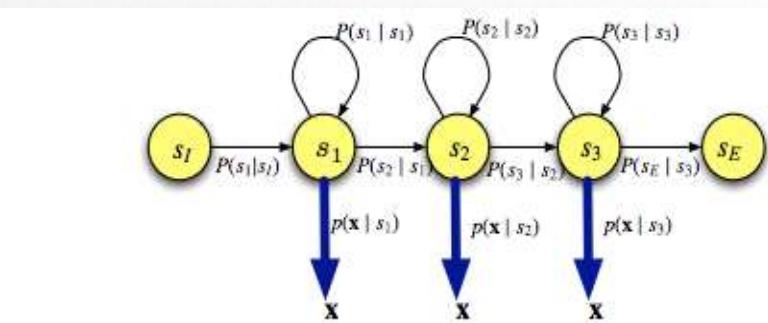
嗷嗷、

HMM – 应用 – 语音识别

- HIDDEN MARKOV MODELS AND GAUSSIAN MIXTURE MODELS
- DEEP NEURAL NETWORKS FOR ACOUSTIC MODELING IN SPEECH RECOGNITION



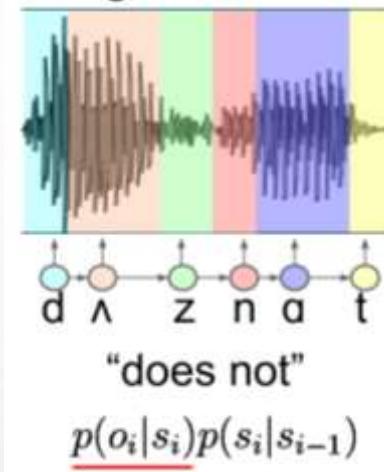
HMM – 应用 – 语音识别



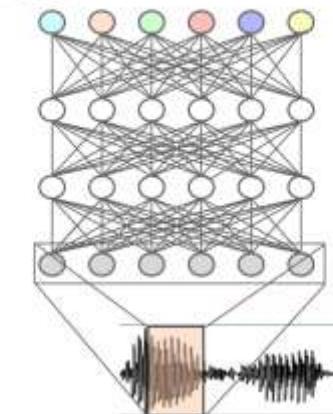
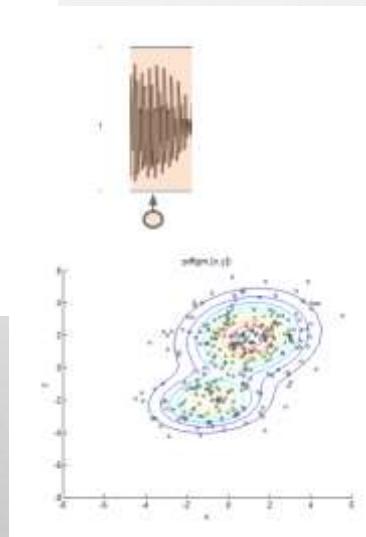
Speech Recognition Problem

Viterbi Search
 $\underset{w}{\operatorname{argmax}} \quad p(w) \quad p(o|w)$

Language Model Acoustic Model

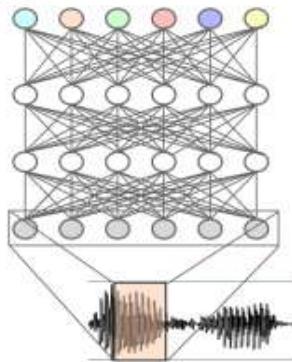


Acoustic Model - 2011-Today - Neural Nets

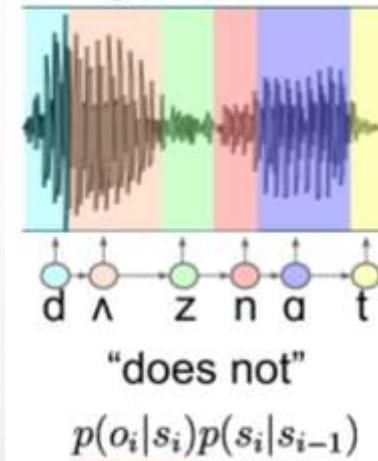
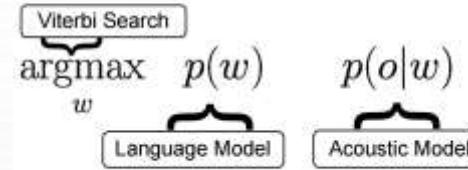


HMM – 应用 – 语音识别

Acoustic Model - 2011-Today - Neural Nets



Speech Recognition Problem



Tweaks - Maximum Mutual Information Training

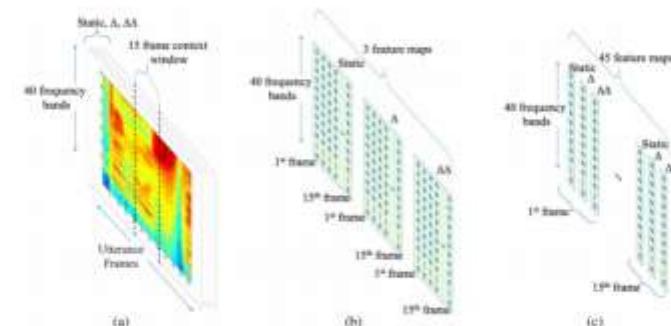
$$p(l_{1:T}|v_{1:T}) = p(l_{1:T}|h_{1:T}) \\ = \frac{\exp\left(\sum_{t=1}^T \gamma_{ij} \phi_{ij}(l_{t-1}, l_t) + \sum_{t=1}^T \sum_{d=1}^D \lambda_{l_t, d} h_{td}\right)}{Z(h_{1:T})}$$

transition probabilities (HMM)

agreement between activations + hidden units

more closely related to objective (sequence labeling)
~5% relative gain in accuracy

Tweaks - Convolutional Nets



~5% relative gain in accuracy

Source: "Convolutional Neural Networks for Speech Recognition" O. Abdel-Hamid et al, IEEE Transactions on Audio, Speech, and Language Processing, Oct 2014

实战

- KALDI
 - [HTTPS://GITHUB.COM/KALDI-ASR/KALDI.GIT](https://github.com/kaldi-asr/kaldi.git)
 - [HTTPS://KALDI-ASR.ORG/DOC/](https://kaldi-asr.org/doc/)
 - [HTTP://BLOG.GEEKIDENTITY.COM/ASR/KALDI/KALDI_TUTORIAL/](http://blog.geekidentity.com/asr/kaldi/kaldi_tutorial/)
- OPEN SLR
 - [HTTP://WWW.OPENSLR.ORG/RESOURCES.PHP](http://www.openslr.org/resources.php)

THANKS