本文简明讲述GMM-HMM在语音识别上的原理，建模和测试过程。这篇blog只回答三个问题：

1. 什么是Hidden Markov Model？
    HMM要解决的三个问题:
    1) Likelihood
    2) Decoding
    3) Training
2. GMM是神马？怎样用GMM求某一音素（phoneme）的概率？

3. GMM+HMM大法解决语音识别
    3.1 识别
    3.2 训练
        3.2.1 Training the params of GMM
        3.2.2 Training the params of HMM

首先声明我是做视觉的不是做语音的，迫于**需要24小时速成语音。上网查GMM-HMM资料中文几乎为零，英文也大多是paper。苦苦追寻终于貌似搞懂了GMM-HMM，感谢语音组老夏（http://weibo.com/ibillxia）提供资料给予指导。本文结合最简明的概括还有自己一些理解应运而生，如有错误望批评指正。

==

1. 什么是Hidden Markov Model？

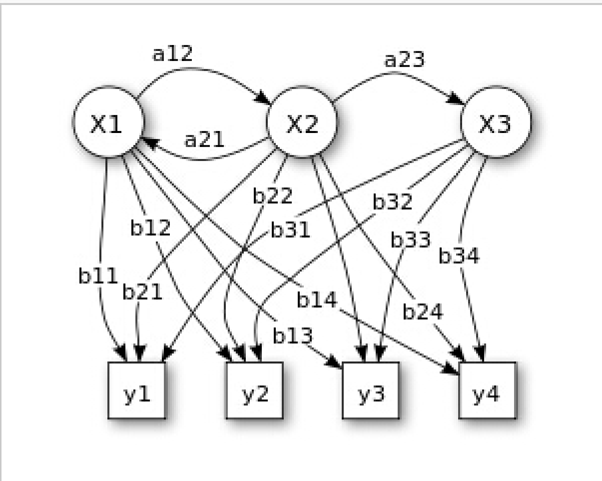


Figure 1. Probabilistic parameters of a hidden Markov model (example)
x — states
y — possible observations
a — state transition probabilities
b — output probabilities

ANS：一个有隐节点（unobservable）和可见节点（visible）的马尔科夫过程（见详解）。
隐节点表示状态，可见节点表示我们听到的语音或者看到的时序信号。
最开始时，我们指定这个HMM的结构，训练HMM模型时：给定n个时序信号y1...yT（训练样本）, 用MLE（typically implemented in EM） 估计参数：
1. N个状态的初始概率
2. 状态转移概率a
3. 输出概率b

- 在语音处理中，一个word由若干phoneme（音素）组成；
- 每个HMM对应于一个word或者音素（phoneme）
- 一个word表示成若干states，每个state表示为一个音素

用HMM需要解决3个问题：

1）. Likelihood: 一个HMM生成一串observation序列x的概率< the Forward algorithm>

- Initialization

$$\alpha_0(s_I) = 1$$
$$\alpha_0(s_j) = 0 \qquad \text{if } s_j \neq s_I$$

- Recursion

$$\alpha_t(s_j) = \sum_{i=1}^{N} \alpha_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$

- Termination

$$p(\mathbf{X} \mid \boldsymbol{\lambda}) = \alpha_T(s_E) = \sum_{i=1}^{N} \alpha_T(s_i) a_{iE}$$

其中，αt（sj）表示HMM在时刻t处于状态j，且observation = {x1,...,xt}的概率

$$\alpha_t(s_j) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, S(t) = s_j \mid \boldsymbol{\lambda})$$

，

aij是状态i到状态j的转移概率，

bj（xt）表示在状态j的时候生成xt的概率，

2）. Decoding: 给定一串observation序列x，找出最可能从属的HMM状态序列< the Viterbi algorithm>

在实际计算中会做剪枝，不是计算每个可能state序列的probability，而是用Viterbi approximation：

从时刻1：t，只记录转移概率最大的state和概率。

记Vt(si)为从时刻t-1的所有状态转移到时刻t时状态为j的最大概率：

$$V_t(s_j) = \max_i V_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$

记

$$bt_t(s_i)$$

为：从时刻t-1的哪个状态转移到时刻t时状态为j的概率最大；

进行Viterbi approximation过程如下：

- Initialization

$$V_0(s_I) = 1$$
$$V_0(s_j) = 0 \qquad \text{if } s_j \neq s_I$$
$$bt_0(s_j) = 0$$

- Recursion

$$V_t(s_j) = \max_{i=1}^{N} V_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$
$$bt_t(s_j) = \arg\max_{i=1}^{N} V_{t-1}(s_i) a_{ij} b_j(\mathbf{x}_t)$$

- Termination

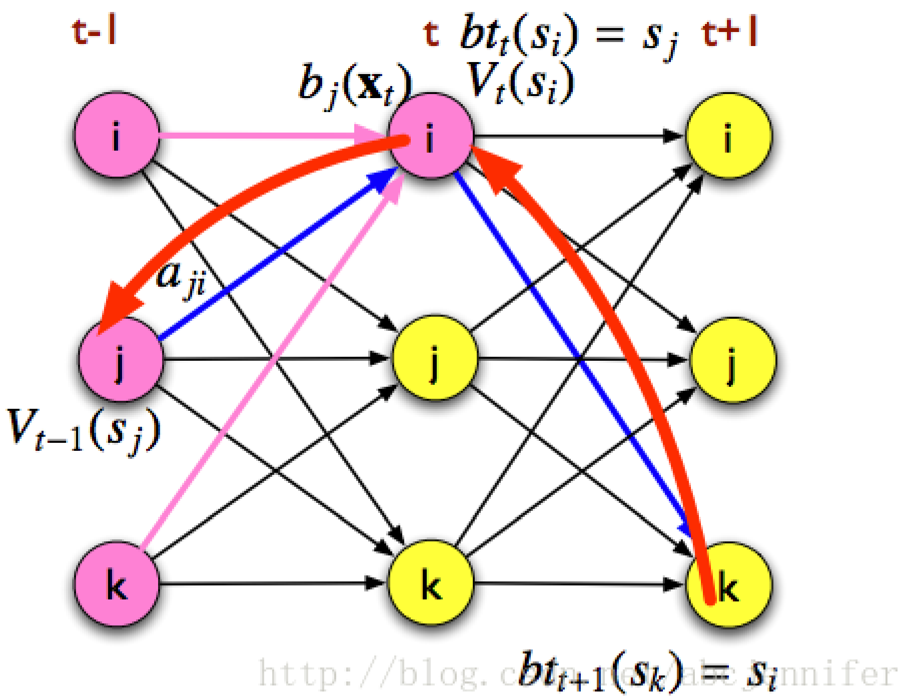
$$P^* = V_T(s_E) = \max_{i=1}^{N} V_T(s_i) a_{iE}$$
$$s_T^* = bt_T(q_E) = \arg\max_{i=1}^{N} V_T(s_i) a_{iE}$$

然后根据记录的最可能转移状态序列

$$bt_t(s_i)$$

进行回溯：

Backtrace to find the state sequence of the most probable path

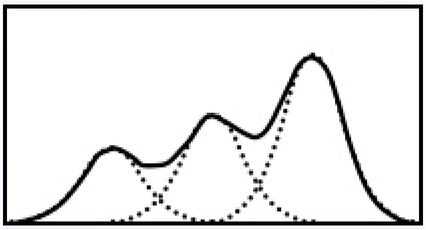


3）. Training: 给定一个observation序列x，训练出HMM参数λ = {aij, bij} the EM (Forward-Backward) algorithm
这部分我们放到“3. GMM+HMM大法解决语音识别”中和GMM的training一起讲

----

2. GMM是神马？怎样用GMM求某一音素（phoneme）的概率？

2.1 简单理解混合高斯模型就是几个高斯的叠加。。。e.g. k=3



$$p(\mathbf{x}) = \sum_{j=1}^{P} P(j)p(\mathbf{x}|j) = \sum_{j=1}^{P} P(j)N_j(\mathbf{x};\boldsymbol{\mu}_j,\sigma_j^2)$$

fig2. GMM illustration and the probability of x

2.2 GMM for state sequence

每个state有一个GMM，包含k个高斯模型参数。如"hi“（k=3）：

PS：sil表示silence（静音）

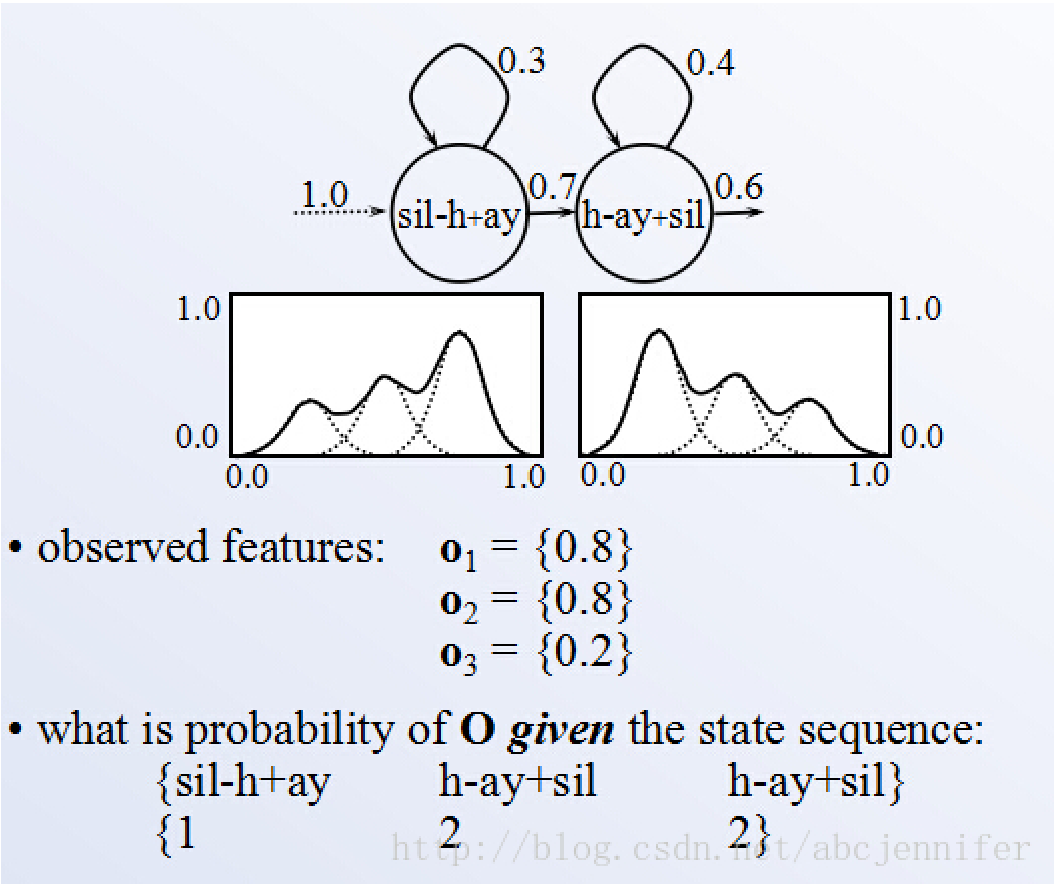


fig3. use GMM to estimate the probability of a state sequence given observation {o1, o2, o3}

其中，每个GMM有一些参数，就是我们要train的输出概率参数

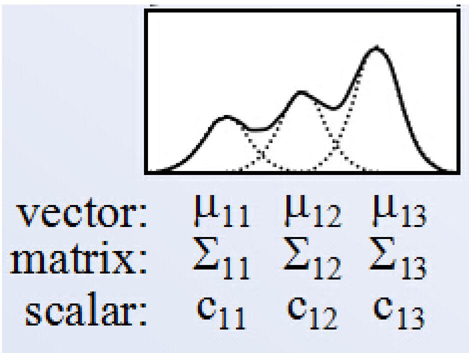


fig4. parameters of a GMM

怎么求呢？和KMeans类似，如果已知每个点x^n属于某每类 j 的概率p(j|x^n)，则可以估计其参数:

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_n P(j|\mathbf{x}^n)\mathbf{x}^n}{\sum_n P(j|\mathbf{x}^n)} = \frac{\sum_n P(j|\mathbf{x}^n)\mathbf{x}^n}{N_j^*}$$

$$\hat{\sigma}_j^2 = \frac{\sum_n P(j|\mathbf{x}^n)||\mathbf{x}^n - \boldsymbol{\mu}_k||^2}{\sum_n P(j|\mathbf{x}^n)} = \frac{\sum_n P(j|\mathbf{x}^n)||\mathbf{x}^n - \boldsymbol{\mu}_k||^2}{N_j^*}$$

$$\hat{P}(j) = \frac{1}{N}\sum_n P(j|\mathbf{x}^n) = \frac{N_j^*}{N}$$

，其中

$$N_j^* = \sum_{n=1}^{N} P(j|\mathbf{x}^n)$$

只要已知了这些参数，我们就可以在predict（识别）时在给定input sequence的情况下，计算出一串状态转移的概率。如上图要计算的state sequence 1->2->2概率：

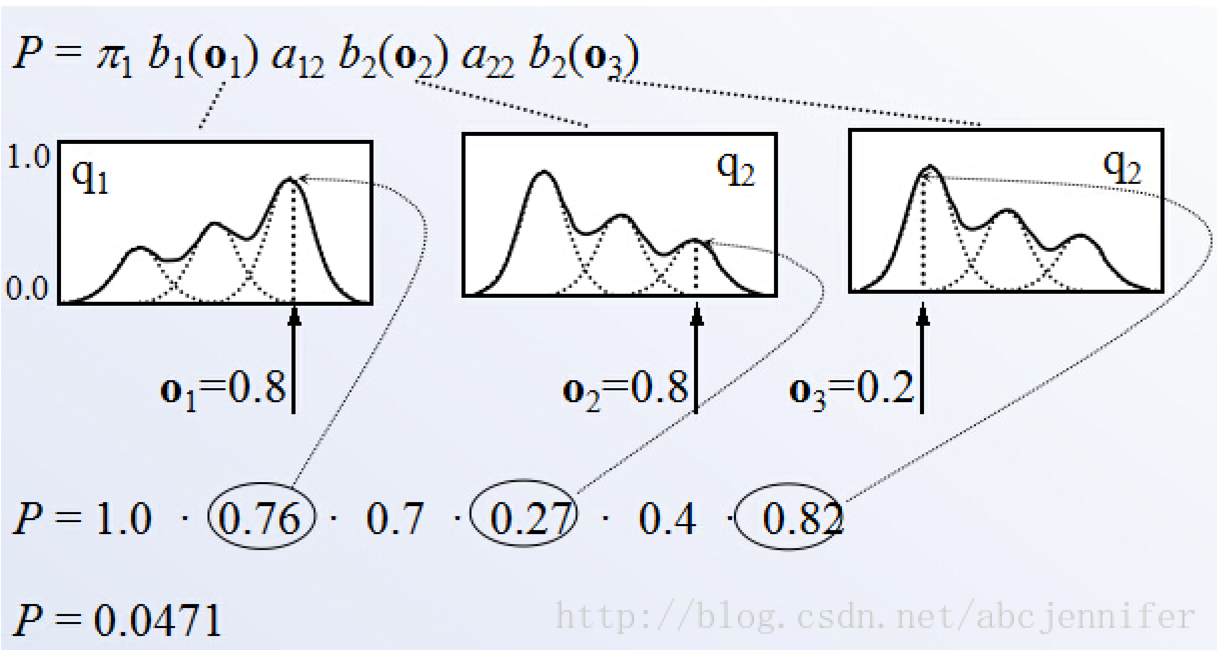


fig5. probability of S1->S2->S3 given o1->o2->o3

----

3. GMM+HMM大法解决语音识别

<!--识别-->

我们获得observation是语音waveform, 以下是一个词识别全过程：

1). 将waveform切成等长frames，对每个frame提取特征（e.g. MFCC），

2).对每个frame的特征跑GMM，得到每个frame(o_i)属于每个状态的概率b_state(o_i)

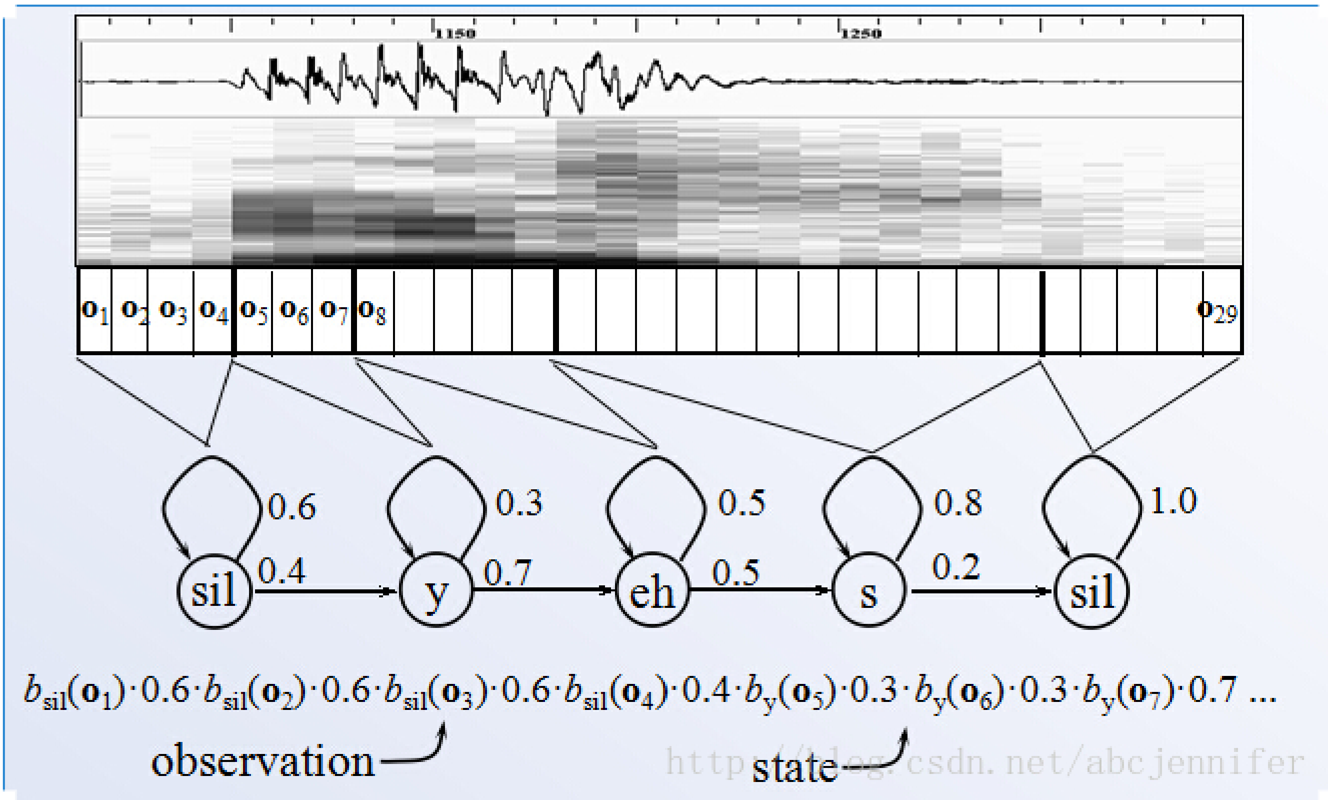


fig6. complete process from speech frames to a state sequence

3). 根据每个单词的HMM状态转移概率a计算每个状态sequence生成该frame的概率; 哪个词的HMM 序列跑出来概率最大，就判断这段语音属于该词

宏观图：

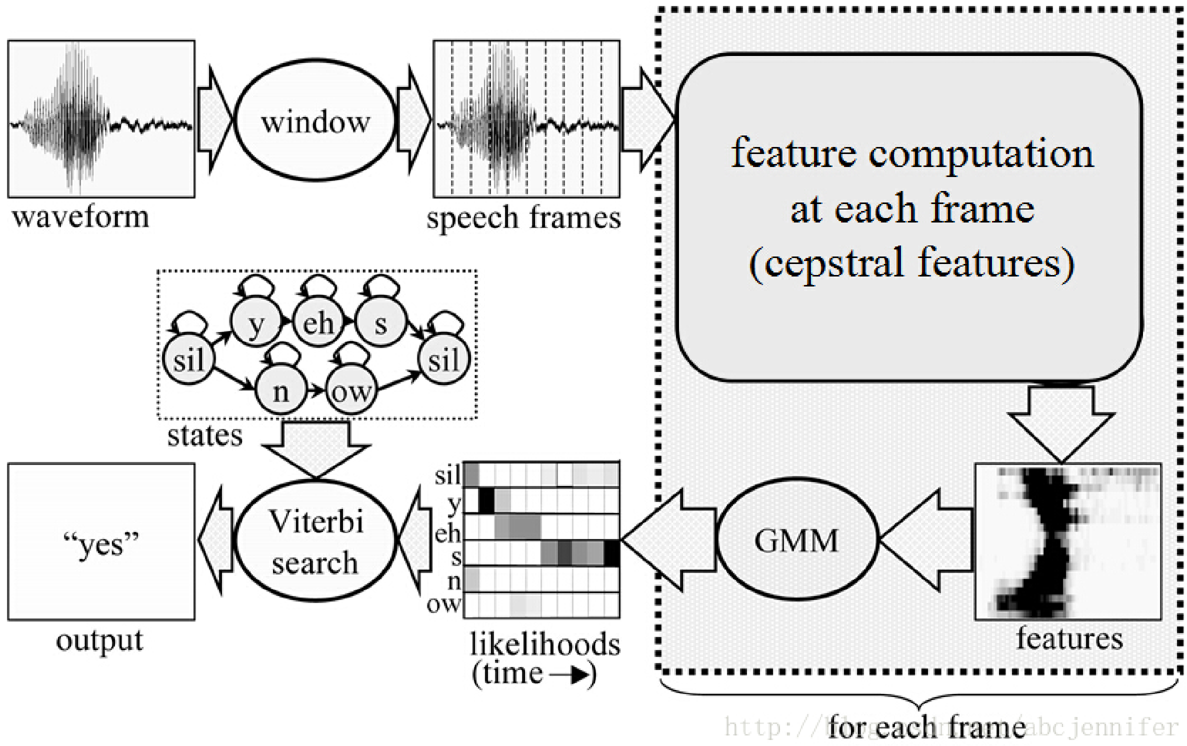


fig7. Speech recognition, a big framework
(from Encyclopedia of Information Systems, 2002)

<!--训练-->

好了，上面说了怎么做识别。那么我们怎样训练这个模型以得到每个GMM的参数和HMM的转移概率什么的呢？

①Training the params of GMM

GMM参数：高斯分布参数：

mean vector μ^j; covariance matrix Σ^j

从上面fig4下面的公式我们已经可以看出来想求参数必须要知道P(j|x),即，x属于第j个高斯的概率。怎么求捏？

$$P(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)P(j)}{p(\mathbf{x})}$$

fig8. bayesian formula of P(j | x)

根据上图 P（j | x）, 我们需要求P(x|j)和P（j）去估计P(j|x).

这里由于P(x|j)和P（j）都不知道，需要用EM算法迭代估计以最大化P(x) = P(x1)*p(x2)*...*P(xn)：

A. 初始化（可以用kmeans）得到P(j)

B. 迭代

  E（estimate）-step: 根据当前参数 (means, variances, mixing parameters)估计P(j|x)

  M（maximization）-step: 根据当前P(j|x) 计算GMM参数（根据fig4 下面的公式：）

$$\hat{\mu}_j = \frac{\sum_n P(j|\mathbf{x}^n)\mathbf{x}^n}{\sum_n P(j|\mathbf{x}^n)} = \frac{\sum_n P(j|\mathbf{x}^n)\mathbf{x}^n}{N_j^*}$$

$$\hat{\sigma}_j^2 = \frac{\sum_n P(j|\mathbf{x}^n)||\mathbf{x}^n - \boldsymbol{\mu}_k||^2}{\sum_n P(j|\mathbf{x}^n)} = \frac{\sum_n P(j|\mathbf{x}^n)||\mathbf{x}^n - \boldsymbol{\mu}_k||^2}{N_j^*}$$

$$\hat{P}(j) = \frac{1}{N}\sum_n P(j|\mathbf{x}^n) = \frac{N_j^*}{N}$$

，其中

$$N_j^* = \sum_{n=1}^{N} P(j|\mathbf{x}^n)$$

②Training the params of HMM

前面已经有了GMM的training过程。在这一步，我们的目标是：从observation序列中估计HMM参数λ；

假设状态->observation服从单核高斯概率分布：

$$b_j(\mathbf{x}) = p(\mathbf{x} \mid s_j) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^j, \boldsymbol{\Sigma}^j)$$

，则λ由两部分组成：

Parameters $\boldsymbol{\lambda}$:

- Transition probabilities a_{ij}:

$$\sum_j a_{ij} = 1$$

- Gaussian parameters for state s_j:
  mean vector $\boldsymbol{\mu}^j$; covariance matrix $\boldsymbol{\Sigma}^j$

HMM训练过程：迭代

E（estimate）-step: 给定observation序列，估计时刻t处于状态sj的概率

$\gamma_t(s_j)$

M（maximization）-step: 根据

$\gamma_t(s_j)$

重新估计HMM参数aij.

其中，

E-step: 给定observation序列，估计时刻t处于状态sj的概率

$\gamma_t(s_j)$

为了估计

$\gamma_t(s_j)$

, 定义

$\beta_t(s_j)$

: t时刻处于状态sj的话，t时刻未来observation的概率。即

$$\beta_t(s_j) = p(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \mathbf{x}_T \mid S(t) = s_j, \boldsymbol{\lambda})$$

这个可以递归计算：β_t（si）=从状态 si 转移到其他状态 sj 的概率aij * 状态 i 下观测到x_{t+1}的概率 bi(x_{t+1}) * t时刻处于状态sj的话{t+1}后observation概率β_{t+1}（sj）

即：

- Initialisation
$$\beta_T(s_i) = a_{iE}$$
- Recursion
$$\beta_t(s_i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(s_j)$$
- Termination
$$p(\mathbf{X} \mid \boldsymbol{\lambda}) = \beta_0(s_I) = \sum_{j=1}^{N} a_{Ij} b_j(\mathbf{x}_1) \beta_1(s_j) = \alpha_T(s_E)$$

定义刚才的

$\gamma_t(s_j)$

为state occupation probability，表示给定observation序列，时刻t处于状态sj的概率P(S(t)=sj | X,λ) 。根据贝叶斯公式p(A|B,C) = P(A,B|C)/P(B|C)，有：

$$P(S(t) = s_j \mid \mathbf{X}, \boldsymbol{\lambda}) = \frac{p(\mathbf{X}, S(t) = s_j \mid \boldsymbol{\lambda})}{p(\mathbf{X}|\boldsymbol{\lambda})}$$

由于分子p(A,B|C)为

$$\begin{aligned}\alpha_t(s_j)\beta_t(s_j) &= p(\mathbf{x}_1, \ldots, \mathbf{x}_t, S(t) = s_j \mid \boldsymbol{\lambda}) \\ &\qquad p(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \mathbf{x}_T \mid S(t) = s_j, \boldsymbol{\lambda}) \\ &= p(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_T, S(t) = s_j \mid \boldsymbol{\lambda}) \\ &= p(\mathbf{X}, S(t) = s_j \mid \boldsymbol{\lambda})\end{aligned}$$

其中，αt（sj）表示HMM在时刻t处于状态j，且observation = {x1,...,xt}的概率

$$\alpha_t(s_j) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, S(t) = s_j \mid \boldsymbol{\lambda})$$

；

$\beta_t(s_j)$

: t时刻处于状态sj的话，t时刻未来observation的概率；

且

$$p(\mathbf{X}|\boldsymbol{\lambda}) = \alpha_T(s_E)$$

finally, 带入

$\gamma_t(s_j)$

的定义式有：

$$\boxed{\gamma_t(s_j) = P(S(t) = s_j \mid \mathbf{X}, \boldsymbol{\lambda}) = \frac{1}{\alpha_T(s_E)} \alpha_t(j)\beta_t(j)}$$

好，终于搞定！对应上面的E-step目标，只要给定了observation和当前HMM参数 λ，我们就可以估计

$\gamma_t(s_j)$

了对吧 (*^__^*)

M-step：根据

$\gamma_t(s_j)$

重新估计HMM参数λ：

对于λ中高斯参数部分，和GMM的M-step是一样一样的（只不过这里写成向量形式）：

$$\hat{\mu}^j = \frac{\sum_{t=1}^T \gamma_t(s_j) x_t}{\sum_{t=1}^T \gamma_t(s_j)}$$

$$\hat{\boldsymbol{\Sigma}}^j = \frac{\sum_{t=1}^T \gamma_t(s_j)(x_t - \hat{\mu}^j)(x - \hat{\mu}^j)^T}{\sum_{t=1}^T \gamma_t(s_j)}$$

对于λ中的状态转移概率aij, 定义C(Si->Sj)为从状态Si转到Sj的次数，有

$$\hat{a}_{ij} = \frac{C(s_i \rightarrow s_j)}{\sum_k C(s_i \rightarrow s_k)}$$

实际计算时，定义每一时刻的转移概率

$\xi_t(s_i, s_j)$

为时刻t从si->sj的概率：

$$\begin{aligned}\xi_t(s_i, s_j) &= P(S(t) = s_i, S(t+1) = s_j \mid \mathbf{X}, \boldsymbol{\lambda}) \\ &= \frac{P(S(t) = s_i, S(t+1) = s_j, \mathbf{X} \mid \boldsymbol{\lambda})}{p(\mathbf{X}|\Lambda)} \\ &= \frac{\alpha_t(s_i) a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(s_j)}{\alpha_T(s_E)}\end{aligned}$$

那么就有：

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \xi_t(s_i, s_j)}{\sum_{k=1}^{N} \sum_{t=1}^{T} \xi_t(s_i, s_k)}$$

把HMM的EM迭代过程和要求的参数写专业点，就是这样的：

E step For all time-state pairs

1. Recursively compute the forward probabilities $\alpha_t(s_j)$ and backward probabilities $\beta_t(j)$
2. Compute the state occupation probabilities $\gamma_t(s_j)$ and $\xi_t(s_i, s_j)$

M step Based on the estimated state occupation probabilities re-estimate the HMM parameters: mean vectors $\boldsymbol{\mu}^j$, covariance matrices $\boldsymbol{\Sigma}^j$ and transition probabilities a_{ij}

PS：这个训练HMM的算法叫 Forward-Backward algorithm。

一个很好的reference：点击打开链接