隐马尔可夫模型(2)

(HMM)

徐静

2018-06-24

上节回顾

0.马尔可夫链

1.隐马尔可夫模型的基本概念

2.概率计算算法

3.参考文献

4.下次预告

5.问题与讨论

本节内容

1.学习算法

- 问题2的解决办法

2.预测算法

- 问题3的解决办法

3.HMM代码实现2

4.后退，我要实现一个GMM-HMM语音识别系统

- 放松一下: 看看我的偶像
- 愉快的读paper
- GMM-HMM语音识别系统
- 其他语音识别框架

5.参考文献与问题讨论

1.学习算法

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

HMM的学习，根据训练数据是包括观测序列和对应的状态序列还是只有观测序列，可以分别由监督学习与非监督学习实现。

本部分先介绍监督学习算法，再介绍非监督学习算法--Baum-Welch算法（也就是EM算法）

1.监督学习方法

假设已给的训练数据包含 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HMM的参数。

1.转移概率 a_{ij} 的估计

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

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

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

$$\hat{b}_j(k)=\frac{B_{jk}}{\sum_{k=1}^{M}B_{jk}},j=12,\dots,N;k=1,2,\dots M$$

3.初始状态概率 π 的估计 $\hat{\pi}$ 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，有时候就会利用非监督学习的方法

2.非监督学习算法--Baum-Welch算法

假设给定训练数据中包含 S 个长度为 T的观测序列 $\{O_1, O_2, ..., O_s\}$ 而没有对应的状态序列，目标是学习HMM模型 $\lambda = (A, B, \pi)$ 的参数。我们将观测序列数据看做观测数据 O, 状态序列看做不可观测的因数据 I, 那么HMM事实上是一个含有隐变量的概率模型

$$\Pr(O|\lambda) = \sum \Pr(O|I, \lambda) \Pr(I|\lambda)$$

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

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

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

其中, λ 是HMM参数的当前估计值, λ 是要极大化的HMM参数。

$$\Pr(O, I|\lambda) = \pi_{I_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i-2}(o_2) \ldots a_{i_{T-1} i_T} b_{i_T}(o_T)$$

于是函数 Q 可以写成：

$$Q(\lambda, \bar{\lambda}) = \sum_I log\pi_{i_1}\Pr(O, I|\bar{\lambda}) + \sum_I (\sum_{t=1}^{T-1} loga_{i_t i_{t+1}})\Pr(O, I|\bar{\lambda}) + \sum_I (\sum_{t=1}^{T} logb_{i_t}(o_t))\Pr(O, I|\bar{\lambda})$$

式中求和都是对所有新联数据的序列总长度 T 进行的。

[1].按照 Q函数的定义 $Q(\lambda, \bar{\lambda}) = E_I[log\Pr(O, I|\lambda)|O, \bar{\lambda}]$

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分别出现在上式的3个独立部分，所以只需要对各项分别极大化

第一项:使用拉格朗日数乘法(注意约束条件: $\sum_{i=1}^{N} \pi_i = 1$),最终可以得到

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

第二项:使用拉格朗日数乘法(注意约束条件: $\sum_{j=1}^{N} a_{ij} = 1$),最终可以求得

$$a_{ij} = \frac{\sum_{t=2}^{T-1} \Pr(O, i_t = i. i_{t+1} = j|\bar{\lambda})}{\sum_{t=1}^{T-1} \Pr(O, i_t = i|\bar{\lambda})}$$

第三项:同样使用拉格朗日书乘法(约束条件: $\sum_{k=1}^{M} b_j(k) = 1$)

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

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(这是当然了)

3.Baum-Welch模型参数估计

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}; b_j(k) = \frac{\sum_{t=1,o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}; \pi_i = \gamma_1(i)$$

负责人的告诉大家：上述三个式子就是Baum-Welch算法

Baum-Welch算法：

输入：观测数据 $O=(o_1,o_2,\ldots o_T)$

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 $n=1,2,\ldots,$ $a_{ij}^{(n+1)}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}; b_j(k)^{(}n+1); \pi_i^{(}n+1)$

右端各值按观测 $O=(o_1,o_2,\ldots,o_T)$和模型 $\lambda^{(n)=(A^{(n)},B^{(n)},\pi^{(n)})}$ 计算

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

2.预测算法

- 预测问题，也称为解码(decoding)问题,已知模型和观测序列，求给定观测序列条件概率 $\Pr(I|O)$ 最大的状态序列 $I=(i_1,i_2,\dots,i_T)$. 即给定观测序列，求最有可能对应的状态序列。
- 近似算法与维特比算法(Viterbi)

(1)近似算法

近似算法的思想是：在每个时刻 t 选择在该时刻最有可能出现的状态 $i_t^\star$, 从而得到一个状态序列 $I^\star=(i_1^\star,...,i_T^\star)$,将他们作为预测结果

- 给定HMM模型 λ 和观测序列 O,在时刻 t 处于状态 q_i 的概率 $\gamma_t(i)$(这个是可以计算出来的,式子我就不列了)
- 在每一个时刻 t 最有可能的状态 $i_t^\star$ 是：

  $$i_t^\star = argmax_{1\le i\le N}[\gamma_t(i)], t=1,2,...,T$$

  从而得到状态序列 $I^\star$
- 特点：
  - 计算简单
  - 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
  - 尽管如此，近似算法仍有用

(2)维特比算法

威特比算法实际上是用动态规划解隐马尔可夫模型的预测问题，既用动态规划求概率最大路径(最优路径)，一条路径就对应了一个状态序列

最优路径的意思：如果最优路径在时刻 t 通过节点 $i_t^\star$, 那么这一路径从节点 $i_t^\star$ 到终点 $i_T^\star$ 的部分路径，在所有可能的部分路径中是最优的。

这样我们就可以递归的计算 t 时刻状态为 i 各条部分路径的最大概率。这样时刻 $t=T$ 的最大概率即为最优路径的概率 $P^\star$, 最优路径的终结点 $i_T^\star$ 也同时得到。

之后，为了找出最优路径的各个节点，从终结点 $i_T^\star$ 开始，由后向前逐步求得节点，得到最优路径

以上就是维特比算法的过程

引入两个变量 δ 和 ψ.

定义在时刻 t 状态为 i 的所有单个路径中概率最大值：

$$\delta_t(i) = max_{i_1,\ldots,i_{t-1}} \Pr(i_t = i, i_{t-1}, \ldots, i_1, o_1, \ldots, o_t|\lambda), i = 1, 2, \ldots N$$

很容易的得到 δ 的递推公式

$$\delta_{t+1}(i) = max_{1\leq jleqN}[\delta_t(j)a_{ji}]b_i(o_{t+1})$$

定义在 t 时刻状态为i 的所有单个路径 $(i_1, \ldots, i_{t-1}, i)$ 中概率最大的路径的第 $t-1$ 个节点为

$$\psi_t(i) = argmax_{1\leq j\leq N}[\delta_{t-1}(j)a_{ji}]$$

维特比算法：

输入：HMM模型和观测

输出：最优路径 $I^\star = (i_1^\star, \ldots, i_T^\star)$

(1)初始化 $\delta_1(i) = \pi_i b_i(o_1); \psi_1(i) = 0$

(2)递推. $\delta_t(i); \psi_t(i)$

(3)终止 $P^\star = max_{1 \le i \le N} \delta_T(i); i_T^\star = argmax_{1 \le i \le N}[\delta_T(i)]$

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

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

求得最优路径 $I^\star = (i_1^\star, \ldots, i_T^\star)$

3.HMM代码实现2

```python
def EM(self, observation, criterion=0.05):
    """EM算法进行参数学习"""
    n_state = self.A.shape[0]
    n_sample = len(observation)
    done = 1
    while done:
        Alpha = self.forward(observation)
        Beta = self.backward(observation)
        xi = np.zeros((n_state, n_state, n_sample-1))
        gamma = np.zeros((n_state, n_sample))
        for t in range(n_sample-1):
            denom = np.sum(np.dot(Alpha[:, t].T, self.A) * self.B[:,
            sum_gamma1 = np.sum(Alpha[:, t] * Beta[:, t])
            for i in range(n_state):
                numer = Alpha[i, t] * self.A[i, :] * self.B[:, observa
                 xi[i, :, t] = numer/denom
            gamma[i, t] = Alpha[i, t] * Beta[i, t] / sum_gamma1
        last_col = Alpha[:, n_sample-1].T * Beta[:, n_sample-1]
        gamma[:, n_sample-1] = last_col / np.sum(last_col)
        # 更新参数
        n_pi = gamma[:, 0]
        n_A = np.sum(xi, 2) / np.sum(gamma[:, :-1], 1)
        n_B = np.copy(self.B)
        num_level = self.B.shape[1]
        sum_gamma = 0
        a = 0
        for lev in range(num_level):
```

```python
def viterbi(self, obs_seq):
     """viterbi算法预测状态序列"""
     N = self.A.shape[0]
     T = len(obs_seq)
     P = np.zeros((N, T))
     prev_point = np.zeros((N, T))
     prev_point[:, 0] = 0
     P[:, 0] = self.pi * self.B[:, obs_seq[0]]
     for t in range(1, T):
         for n in range(N):
             P[n, t] = np.max(P[:, t - 1] * self.A[:, n]) * self.B[n,
             prev_point[n, t] = np.argmax(P[:, t - 1] * self.A[:, n]
    return P, prev_point
```

```python
def build_path(self, obs_seq):
     """return the optimal path"""
     P, prev_point = self.viterbi(obs_seq)
     T = len(obs_seq)
     opt_path = np.zeros(T)
     last_state = np.argmax(P[:, T-1])
     opt_path[T-1] = last_state
     for t in reversed(range(T-1)):
         opt_path[t] = prev_point[int(opt_path[t+1]), t+1]
    last_path = reversed(opt_path)
    return last_path
```

4.后退，我要实现一个GMM-HMM语音识别系统

- 放松一下: 看看我的偶像
- 愉快的读paper
- GMM-HMM语音识别系统
- 其他语音识别框架

放松一下: 看看我的偶像

愉快的读paper

请看：Single Word Speech Recognition using Hidden Markov Models with Gaussian Mixture emissions

GMM-HMM语音识别系统

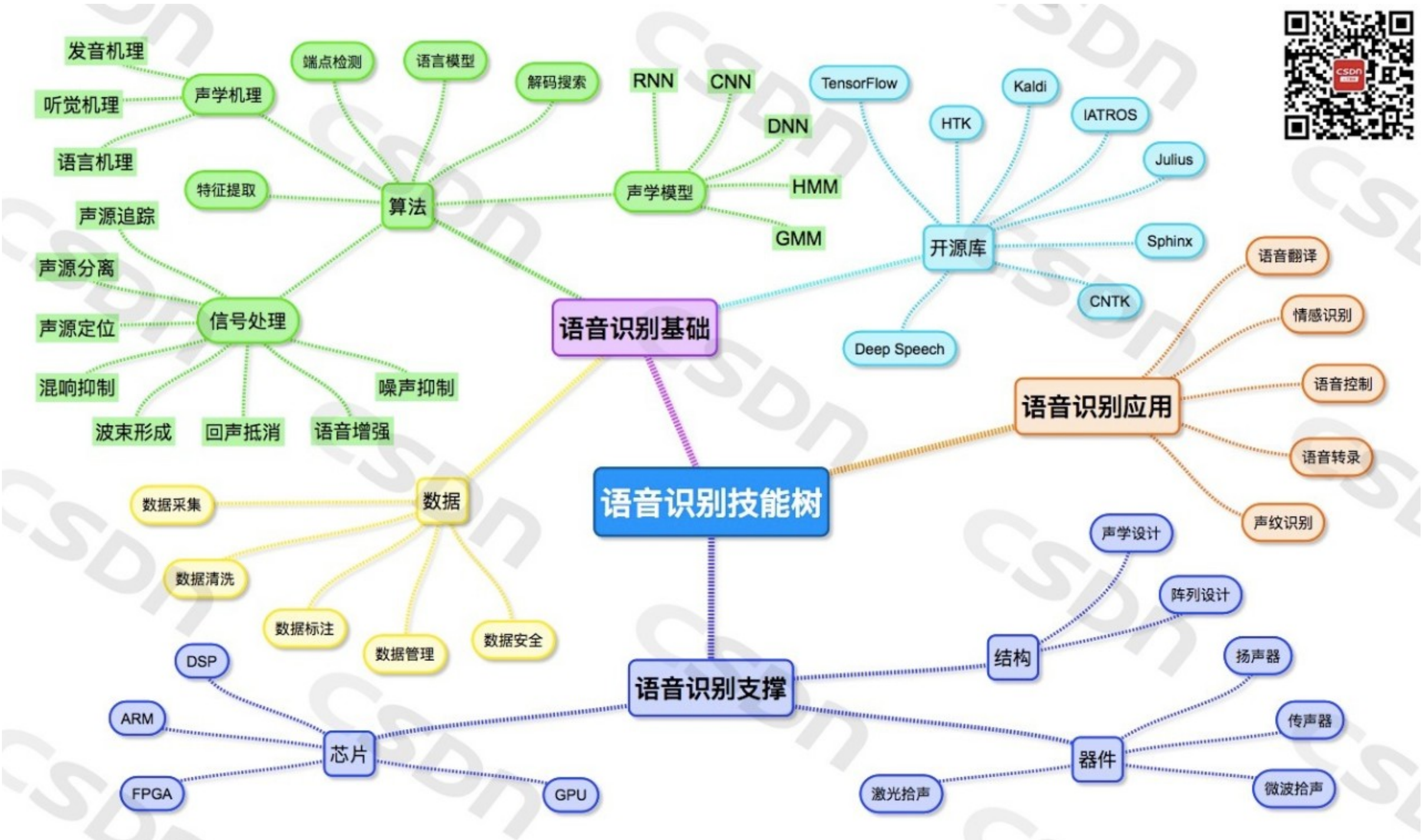
基于这篇paper,复现他GMM_HMM模型实现单个词的语音识别[1]

[1].代码我已打印，想索取代码可问徐静询要

其他语音识别框架

- 常用的语言模型
  - 基于GMM-HMM
  - 基于DNN-HMM
- 常用的框架
  - kaldi
  - htk

我们还需要这些技能


语音识别技能树
语音识别基础
算法
声学机理
发音机理
听觉机理
语言机理
端点检测
语言模型
解码搜索
特征提取
声学模型
RNN
CNN
DNN
HMM
GMM
信号处理
声源追踪
声源分离
声源定位
混响抑制
波束形成
回声抵消
语音增强
噪声抑制
开源库
TensorFlow
HTK
Kaldi
IATROS
Julius
Sphinx
CNTK
Deep Speech
语音识别应用
语音翻译
情感识别
语音控制
语音转录
声纹识别
数据
数据采集
数据清洗
数据标注
数据管理
数据安全
语音识别支撑
结构
声学设计
阵列设计
器件
扬声器
传声器
微波拾声
激光拾声
芯片
DSP
ARM
FPGA
GPU

5.参考文献

[1]. 机器学习[M],周志华

[2]. 统计学习方法[M],李航

[3]. 解析深度学习语音识别实践[M],俞凯，钱彦旻

[4]. 深度学习导论及案例分析[M],李玉鑑，张婷

[5]. Single Word Speech Recognition using Hidden Markov Models with Gaussian Mixture emissions[J],Ronak Panchal, Jayaram Kuchibhotla

问题讨论

1. HMM的3个基本问题
2. 第一节中HMM模型的例子

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

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 $I^\star$

Thanks For Your Attention!