HMM for Chiniese Corpus

段续光, 2018310786; 2018/12/28

Abstract

1 引言

本文给出了使用 EM 算法优化无监督 HMM 算法的推导和实现，以及其再中文数据集上的性能。

2 推导

HMM 模型假设：1) 对应观测序列存在一个隐状态序列；2) 隐状态序列之间满足马尔可夫性；3) 且观测状态仅依赖于当前的隐状态。因此，一个 HMM 需要的参数有：隐状态的初始分布 π，隐状态之间的转移矩阵 A，隐状态到观测状态的观测矩阵 B，记作 $\theta = (\pi, A, B)$。

我们训练 HMM 一般使用的算法是 BaumWelch 算法，其本质是 EM 算法在 HMM 中的应用，我们的目标是，最大化训练数据在 HMM 下的概率：

$$\theta = \arg\max_{\theta} \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} \log P(X, Z; \theta), \tag{1}$$

其中 $\mathcal{X}$ 表示所有的训练语料，$\mathcal{Z}$ 表示所有可能的隐状态。在 HMM 假设下 $P(X, Z|\theta)$ 具有以下形式：

$$P(X, Z; \theta) = P(z_0; \pi) \prod_i P(z_i|z_{i-1}; A) \prod_i P(x_i|z_i; B) \tag{2}$$

其中 $X = (x_0, x_1, x_2, ...x_{N_X})$ 表示观测序列，$Z = (z_0, z_1, z_2, ...z_{N_X})$ 表示隐状态序列。

BaumWelch 算法中，有如下变形：

$$\begin{aligned}
\mathcal{L}(\theta) = \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} \log P(X, Z; \theta) &= \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} \log P(X, Z; \theta^{'}) \frac{P(X, Z; \theta)}{P(X, Z; \theta')} \\
&\geq \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} P(X, Z; \theta^{'}) \log \frac{P(X, Z; \theta)}{P(X, Z; \theta')} = Q(\theta, \theta^{'})
\end{aligned}$$

我们的最终目标是最大化 $\mathcal{L}(\theta)$，而 $\mathcal{L}$ 的最大化可以通过间接最大化 $Q(\theta, \theta^{'})$ 实现，而 Q 的最大化可以利用 EM(Expactation-Maximum) 算法。

E-Step. E-Step(Expactation step) 的目的是求 $Q(\theta, \theta^{'})$ 的值，$Q = Q(\theta, \theta^{'})$

M-Step. M-Step(Maximization step) 的目的是求使 $Q(\theta, \theta^{'})$ 最大的 θ 值。即

$$\theta = \arg\max_{\theta} Q(\theta, \theta^{'})$$

因为 Q 可以进行以下展开:

$$Q(\theta,\theta^{'}) = \underbrace{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\log P(X,Z;\theta)}_{U(\theta,\theta^{'})} - \underbrace{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\log P(X,Z;\theta^{'})}_{V(\theta^{'})} \tag{3}$$

其中 $V(\theta^{'})$ 对 EM 算法 M-Step 中寻找最大化参数没有帮助，所以我们只需要关注 $U(\theta,\theta^{'})$。

将公式2 代入公式3，我们有：

$$\begin{aligned} U(\theta,\theta^{'}) = \sum_{X\in\mathcal{X}} \Bigg(& \sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\log\pi(z_0) \\ & + \sum_{Z\in\mathcal{Z}} \left(P(X,Z;\theta^{'}) \sum_{i=1}^{N_X} \log A(z_{i-1},z_i) \right) \\ & + \sum_{Z\in\mathcal{Z}} \left(P(X,Z;\theta^{'}) \sum_{i=0}^{N_X} \log B(z_i,x_i) \right) \Bigg) \end{aligned}$$

用 $\{\hat{z}_i\}$ 表示所有可能的隐状态。令 π_i 表示初始隐状态为第 i 个隐状态的概率，则 $\sum\pi_i = 1$，利用拉格朗日乘子法，得到:

$$\frac{\partial}{\partial\pi_i}\left(U(\theta,\theta^{'}) + \gamma\left(\sum\pi_i - 1\right)\right) = \sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z},z_0=\hat{z}_i} P(X,Z;\theta^{'})\frac{1}{\pi_i} + \gamma = 0 \tag{4}$$

对所有 i 求和，得到 $\gamma = -\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'}) = \sum_{X\in\mathcal{X}} P(X;\theta^{'})$，代入4，得到:

$$\pi_i = \frac{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z},z_0=\hat{z}_i} P(X,Z;\theta^{'})}{\sum_{X\in\mathcal{X}} P(X;\theta')} \tag{5}$$

同理，用 A_{ij} 表示隐状态从 $\hat{z}_i$ 转移到 $\hat{z}_j$ 的概率（即 $A(\hat{z}_i,\hat{z}_j)$），我们有 $\sum_j A_{ij} = 1$，即:

$$\frac{\partial}{\partial A_{ij}}\left(U(\theta,\theta^{'}) + \gamma\left(\sum_j A_{ij} - 1\right)\right) = \sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=1}^{N_x}\frac{1}{A_{ij}}\mathbb{1}(z_{k-1}=\hat{z}_i, z_k=\hat{z}_j) + \gamma = 0 \tag{6}$$

同样，对 j 求和，我们得到 $\gamma = -\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=1}^{N_x}\frac{1}{A_{ij}}\mathbb{1}(z_{k-1}=\hat{z}_i)$，则:

$$A_{ij} = \frac{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=1}^{N_x}\mathbb{1}(z_{k-1}=\hat{z}_i, z_k=\hat{z}_j)}{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta')\sum_{k=1}^{N_x}\mathbb{1}(z_{k-1}=\hat{z}_i)} \tag{7}$$

同理，取所有可能的观测集合为 $\{\hat{x}_i\}$，令状态 i 转移到观测 j 的概率为 B_{ij}，我们有：

$$B_{ij} = \frac{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=0}^{N_x}\mathbb{1}(z_k=\hat{z}_i, x_k=\hat{x}_j)}{\sum_{X\in\mathcal{X}}\sum_{Z\in\mathcal{Z}} P(X,Z;\theta')\sum_{k=0}^{N_x}\mathbb{1}(z_k=\hat{z}_i)} \tag{8}$$

2.1 计算简化

之前的部分里，推导过程已经完全完成，但是随着隐状态和观测状态的增加，我们的推导结果仍然是难以计算的，回顾 Viterbi 算法计算 HMM 观测概率，我们引入前向概率 $\alpha_t(i)$ 和后向概率 $\beta_t(i)$（参考图1）。**具体地，给定观测值** X：

$$\alpha_t(i) = B_i(X_t)\sum_j \alpha_{t-1}(j)A_{ji},\ \alpha_0(i) = \pi_i B_i(x_0) \tag{9}$$

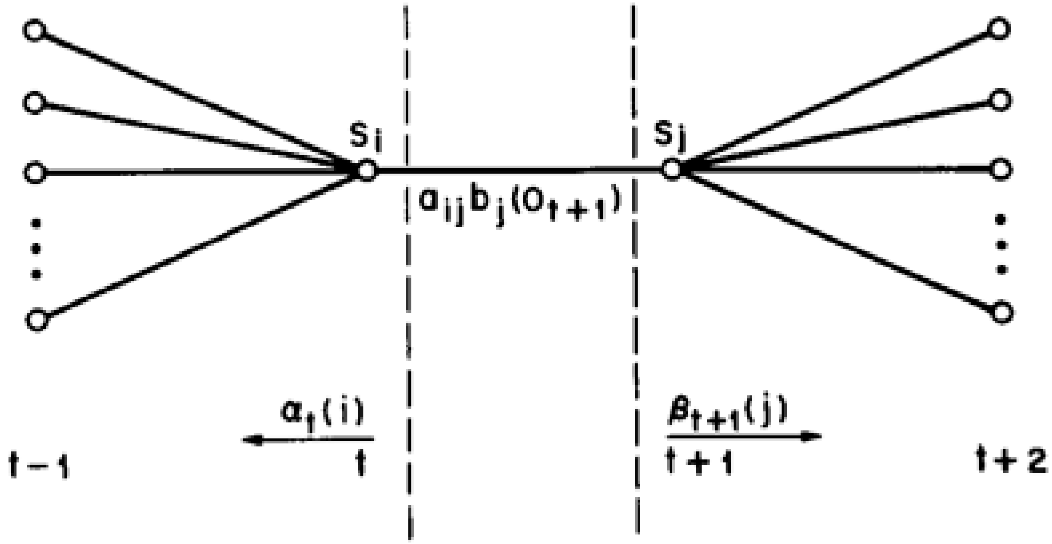


图 1: Viterbi 算法计算前向后向概率

$$\beta_t(i) = \sum_j \beta_{t+1}(j)A_{ij}B_j(x_{t+1}),\ \beta_{N_X}(i) = 1 \tag{10}$$

则我们可以得到 $P(X;\theta) = \sum_i \alpha_{N_X}(i) = \sum_i \pi_i\beta_0(i)B_i(x_0) = \sum_i \alpha_t(i)\beta_t(i)$。

考虑 $\sum_{Z\in\mathcal{Z}} P(X,Z;\theta)\sum_{k=0}^{N_x} \mathbb{1}(z_k = \hat{z}_i)$，相当于对所有满足 $z_k = \hat{z}_i$ 的 Z 进行遍历，可以写成如下的形式:

$$\sum_{Z\in\mathcal{Z}} P(X,Z;\theta)\sum_{k=0}^{N_x} \mathbb{1}(z_k = \hat{z}_i) = \sum_{k=0}^{N_X} \alpha_k(i)\beta_k(i) \tag{11}$$

同样，以下式子也可以直接写出:

$$\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=0}^{N_x} \mathbb{1}(z_k = \hat{z}_i, x_k = \hat{x}_j) = \sum_{k=0}^{N_X} \alpha_k(i)\beta_k(i) * \mathbb{1}(x_k = \hat{x}_j) \tag{12}$$

$$\sum_{Z\in\mathcal{Z}} P(X,Z;\theta^{'})\sum_{k=1}^{N_x} \mathbb{1}(z_{k-1} = \hat{z}_i, z_k = \hat{z}_j) = \sum_{k=0}^{N_X-1} \alpha_k(i)\beta_{k+1}(j)A_{ij}B_j(X_{k+1}) \tag{13}$$

$$\tag{14}$$

即:

$$\pi_i^{(new)} = \frac{\sum_{X\in\mathcal{X}} \alpha_0(i)\beta_0(i)}{\sum_{X\in\mathcal{X}}\sum_j \alpha_0(j)\beta_0(j)}$$

$$A_{ij}^{(new)} = \frac{\sum_{X\in\mathcal{X}}\sum_{k=0}^{N_x-1} \alpha_k(i)\beta_{k+1}(j)A_{ij}B_j(x_{k+1})}{\sum_{X\in\mathcal{X}}\sum_l\sum_{k=0}^{N_x-1} \alpha_k(i)\beta_{k+1}(l)A_{il}B_l(x_{k+1})}$$

$$B_{ij}^{(new)} = \frac{\sum_{X\in\mathcal{X}}\sum_{k=0}^{N_x} \alpha_k(i)\beta_k(i)\mathbb{1}(x_k = \hat{x}_j)}{\sum_{X\in\mathcal{X}}\sum_{k=0}^{N_x} \alpha_k(i)\beta_k(i)}$$

需要注意的是，上面的公式中，我们一开始用 $\theta^{'}$ 表示旧的参数，之后为了书写方便省略了 '$'$'。最后的结果中为了区分，用 '$^{(new)}$' 表示新的参数。

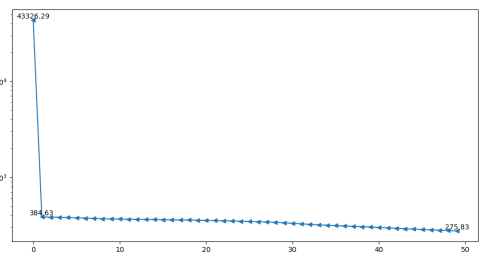
3 实验

3.1 数据预处理

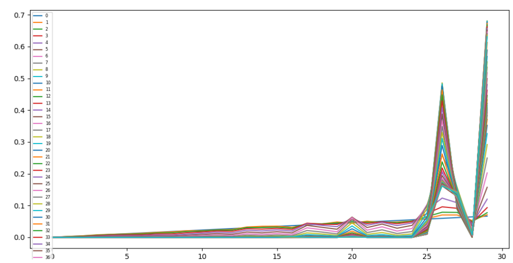
1. 句子以换行符、句号、问号、叹号为分界符；
2. 移除训练数据中句子长度大于 200 词或者小于 5 个词的句子；
3. 词频低于 20 或者未在训练数据中出现的词均用 ‘oov’ 表示。

3.2 训练过程

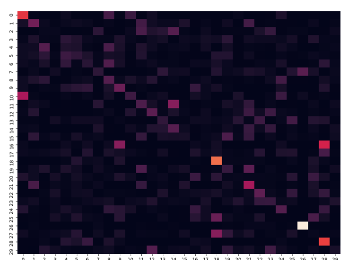
我们对模型进行了 50 次迭代，以下是 ‘hidden_size=30’ 时在测试集上的初始状态概率变化图；困惑度变化曲线（所有句子困惑度的几何均值）；以及最终状态下的状态转移矩阵 A，观测矩阵 B。



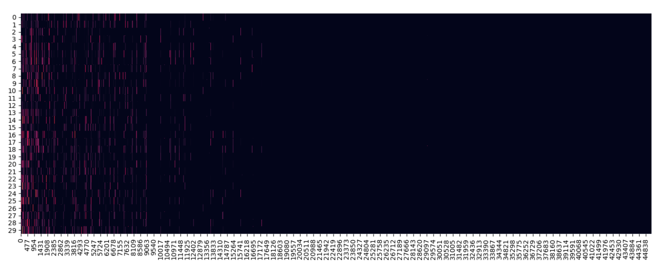
(a)



(b)



(c)



(d)

图 2: (a) 测试集上 ‘困惑度-训练步数’ 变化曲线，(b)‘初始状态分布-训练步数’ 变化曲线，可以看出最终初状态收敛到在某两个状态之间 (c) 训练 50 轮 (最终轮) 时的状态转移矩阵，颜色偏黑概率较小，偏红概率较大，(d) 训练 50 轮时的观测矩阵，横轴为观测值，纵轴为状态

3.3 最终结果

dataset	hidden_size			baseline
	10	20	30	
test	288.76	278.19	274.76	159.66
valid	307.99	295.89	293.36	180.53

从结果来看，我们训练的隐马尔科夫模型并没有 basleine 使用词频得到的结果好，我们认为原因主要有:

1. 隐状态利用率太低，因为我们不对隐状态之间的差异性进行约束，因此可能导致不同的隐状态之间差异较小，从而导致最终效果较差，这也可以从不同的 hidden size 对应的结果差别不是很大得出相似的结论；
2. 训练时常不够，我们训练 50 轮时因为时间原因并没有进一步实验，但是模型仍有继续收敛的倾向；
3. 数据不够，我们其实发现测试数据中出现了大量的没有出现在训练集中的词，这些词都用 'oov' 代替了；同时训练出来的隐马尔科夫模型倾向于不预测低频词，因此在数据不充足的情况下，会过拟合到训练集。
4. 多项式分布本身导致了模型参数太多 (与上一条原因某种层次上等价)，容易过拟合。

A 代码

所有的代码和结果图均已上传到`https://github.com/XgDuan/n_gram_hw`