隐式马尔科夫模型在信息提取中的应用

严春伟

2013 年 1 月 15 日

1 前言

信息提取 (Information Extraction)，也就是从非结构的数据中提取出结构化数据，如从自然语言文本中提取出相关的信息，然后存储到结构化的数据库中。

如何能够从庞大的互联网自然语言文本化的数据中提取出我们所感兴趣的数据一直是自然语言处理的重点研究方向，也是我比较感兴趣的方面。

隐马尔科夫模型 (Hidden Markov model) 在传统的自然语言处理（如统计机器翻译、以及语音识别方面）用的非常成熟，而将 HMM 用于 IE 还是一个比较新的尝试。目前，大部分 IE 的实现仍旧需要建立模板，通过模板从自然语言文本中提取数据，如果人工建立模板的话，是一个非常繁琐的事情，而且模板永远是滞后的。通过 HMM 模型以及机器学习的方式来训练出模板是一件非常有前景的事情。前人通过 HMM 模型来进行一些相对简单的信息提取实验，实现的模型的精度甚至超出了人工的水平。可见 HMM（以及条件随机场）等在 IE 领域前途非常广阔。

隐马尔科夫模型是一种统计学习模型，可以用于标注问题。隐马尔科夫模型在序列化的数据的处理中有比较成熟的应用，如:

- 语音识别
- 自然语言处理
- 生物信息
- 模式识别

本文将会论述隐马尔科夫的基本问题，如定义，解决问题，基本算法等，此外，本文还会论述 HMM 在 IE 方面的一个应用的思想。

2 基本概念

2.1 定义

隐马尔科夫模型 (HMM) 描述一个隐藏的马尔科夫链生成不可观察的状态 (state) 随机序列，在由各个状态生成一个观测而产生观测随机序列的过程。

HMM 的确定因素：

1. 初始状态分布 π
2. 状态转移概率分布 A
3. 观测概率 B

隐马尔科夫的要素

1. N, 表示模型中的状态数目
2. M, 表示模型中每个状态不同的观察符号
3. A, 状态转移概率分布。$A = \{a_{ij}\}$

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_j) \quad 1 \le j \le N \tag{1}$$

4. B, 观察字符在状态 j 时的概率分布,$B = \{b_j(k)\}$, 其中

$$b_j(k) = P(v_k | q_t = S_j) \quad 1 \le j \le N, 1 \le k \le M \tag{2}$$

2.2 三个基本问题

1. 给定一个模型 $\gamma = N, M, A, B, \pi$, 如何高效地计算某一输出字符序列 $O = O_1 O_2 \cdots_n$, 如何找到产生这一序列概率的最大的概率序列.
2. 给定一个模型 $\gamma = N, M, A, B, \pi$ 和一个输出字符序列 $O = O_1 O_2 \cdots O_n$, 如何调整模型的成熟使得产生这一序列的概率最大的状态系列 $\varrho = s_i s_j \cdots s_T$.
3. 给定一个模型 $\gamma = N, M, A, B, \pi$ 和一个输出字符序列 $O = O_1 O_2 \cdots O_n$, 如何调整模型的参数使得产生这一序列的概率最大.

2.3 条件性假设

1. t 时刻的状态 $q_t = s_i$ 只依赖于 $t-1$ 时刻的状态 $q_{t-1} = s_j$, 即

$$P(q_1 | q_{t-1} \cdots q_1, \gamma) = P(q_t | q_{t-1}, \gamma) \tag{3}$$

2. t 时刻生成的值 $b_i(O_t)$ 只依赖与 t 时刻的状态 $q_t = s_i$, 即

$$P(O_1 O_2 \cdots O_T | q_1 q_2 \cdots q_T, \gamma) = \Pi_{t=1}^{T} P(O_t | q_t) \tag{4}$$

3. 状态与具体的时间无关

$$P(q_i | q_{i-1}, \gamma) = P(q_j | q_{j-1}, \gamma) \tag{5}$$

3 HMM 的概率计算

假如给定一个评论问题, 给定一个模型 γ 和一个观察序列 $O = O_1 O_2 \cdots O_T$, 如何计算由这一模型产生观察序列的概率.

由于状态是不可见的, 要计算观察序列的概率, 必定需要回到状态. 不妨假设其中一个状态序列是

$$\varrho = q_1 q_2 \cdots q_T \tag{6}$$

其中 q_1 为初始状态, 则此时序列 O 的概率为

$$P(O | \varrho, \gamma) = \Pi_{t=1}^{T} P(O_t | q_t, \gamma) = \Pi_{t=1}^{T} b_{q_t}(O_t) \tag{7}$$

如此, 知道在某一特定状态下产生观察序列的概率, 最简单的方式就是利用穷举法, 但是其复杂度达到了 $2TN^T$, 其中 N 为模型中的状态数,T 为观察序列的长度.

HMM 模型是一个比较成熟的模型, 在漫长的发展中, 已经有许多成熟的方法, 本文将会简单介绍比较经典的 forward 和 backward 算法.

3.1 forward 算法

forward 以及后面的 backward 算法的核心都是定义了一个核心的公式形式, 通过这个公式形式完成递推迭代.

定义 forward 变量 $\alpha_t(i)$ 为

$$\alpha_t(t) = P(O_1O_2\cdots O_t, q_t = s_i|\gamma) \tag{8}$$

那么递推起来就是

$$\begin{aligned}\alpha_j(t+1) &= P(O_1O_2\cdots O_{t+1}, q_{t+1} = s_j|\gamma) \\ &= \left[\sum_{i=1}^{N}\alpha_i(t)a_{ij}\right]b_j(O_{t+1})\end{aligned}$$

利用这个特殊的递推形式, 最终的概率为

$$P(O|\gamma) = \sum_{i=1}^{N}\alpha_T(i) \tag{9}$$

forward 算法的计算量是 N^2T, 远小于穷举法的 $2TN^T$.

3.2 backward 算法

backward 算法与 forward 算法思想基本一致, 特性也相同

其核心递推变量:

$$\beta_t(i) = P(O_{t+1}O_{t+2}\cdots O_T|q_t = s_i, \gamma) \tag{10}$$

$$\beta_i(j) = \sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1, \quad 1 \le i \le N. \tag{11}$$

3.3 Viterbi 算法

给定一个解码问题, 从 N^T 个可能的状态序列中找到一个" 最优" 的状态序列，其中 N 是 HMMs 模型的状态个数，T 是观察序列的长度。

定义一个变量 $\gamma_i(i)$ 表示给定观察序列 O 和模型 γ 条件下，在 t 时刻，状态为 $q_t = s_t$ 的概率：

$$\gamma_i(i) = P(q_t = s_t|O, \gamma) \tag{12}$$

可以用之前介绍的 forward　backward 算法来表示：

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\gamma)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{t=1}^{N}\alpha_{t(i)}\beta_t(i)} \tag{13}$$

如此，利用 $\gamma_t(i)$ 就可以找出 t 时刻单个概率最大的状态 q_t：

$$q_t = arg\max_{1\le i\le N}\gamma_i(i), 1 \le t \le T \tag{14}$$

Viterbi 算法在实现中，最核心的一个思想就是类似图理论中最短路径，根据动态规划理论，最优路径的特性：如果最优路径在时刻 t 通过节点 i_t^*，那么这一路径从节点 i_t^* 到终点 i_T^* 的部分路径，对于从 i_t^* 到 i_T^* 的所有可能的部分路径来说，必须是最优的。

Viterbi 算法运行时，会按照步骤动态搜索和保存局部最优的路径，直至最终最优路径生成。计算复杂度为 NT^2。

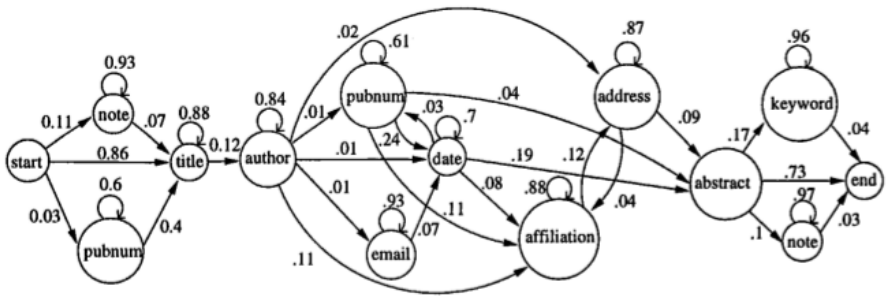

Figure 1: Example HMM. Each state emits words from a class-specific multinomial distribution.

图 1: 状态转移示意图

4 HMM 在 IE 中的一种应用思路

现在有一个任务，有一大堆论文，需要从论文的头部提取出作者名称、邮箱地址、地址、单位、年份等信息。

常规的做法是采用正则表达式人工构建模板，或者利用简单机器学习的方式，构建硬性组合模板。前者是工作繁琐；后者尽管降低了人工负担，但适应性比较弱。

由于论文头部信息的格式千变万化，但是不变的是内容。人工如何去识别，那就是通过理解的方式。那么如何让机器理解论文头部的内容，从语义的层次来识别出论文头不同的部分，这就是 HMM 模型的应用之处。

设定 HMM 隐含的状态为相应的标注标签，如：标题、作者、邮箱地址等。而观测就是具体的内容。我们的目标是训练出一个模型，能够通过处理了论文头的内容，还原出最优可能的状态序列。

首先需要进对数据集进行标注，然后进行模板训练

$$\hat{P}(q \to q') = \frac{c(q \to q')}{\sum_{\delta \in Q} c(q \to s)} \tag{15}$$

$$\hat{P}(q \uparrow \delta) = \frac{c(q \uparrow \delta)}{\sum_{\rho \in \sum} c(q \uparrow p)} \tag{16}$$

其中 $c(q \to q')$ 表示从状态 q 转移到状态 q' 的数量。

$c(q \uparrow \delta)$ 表示从状态 q 得到观测 δ 的数量。

通过这些，学习出 HMM 的模型 γ，然后就可以进行标注了。传入一个论文头，HMM 模型转化出其状态序列，通过状态序列，就可以得到各种信息了。

参考文献

[1] 李航, 统计学习方法, 清华大学出版社.2012

[2] 基于 CRFs 的中文分词算法研究与实现

[3] Andrew McCallum, Dayne Freitag, and Fernando Pereira, Maximum Entropy Markov Models for Information Extraction and Segmentation