漫谈 HMM：Definition

http://freemind.pluskid.org/machine-learning/hmm-definition

posted on Free Mind on May 4, 2013
generated with pandoc on December 3, 2015
category: Machine Learning

tags: Algorithm, Probabilistic Graphical Model, Speech Recognition

坊间有流传过这么一段《胡适留学日记》:

> *7* 月 *4* 日: 新开这本日记，也为了督促自己下个学期多下些苦功。先要读完手边的莎士比亚的《亨利八世》。
>
> *7* 月 *13* 日: 打牌。
>
> *7* 月 *14* 日: 打牌。
>
> *7* 月 *15* 日: 打牌。
>
> *7* 月 *16* 日: 胡适之啊胡适之！你怎么能如此堕落！先前订下的学习计划你都忘了吗？子曰：“吾日三省吾身。”不能再这样下去了！
>
> *7* 月 *17* 日: 打牌。
>
> *7* 月 *18* 日: 打牌。

且不论真假，突然觉得倒是很合适用来作为 Hidden Markov Model (HMM) 的例子来讲的，因为和书上课上讲的例子，天气呀遛狗啊还是马克杯啊什么的，果然还是这个比较好玩一点啊。

假设小明有很严重的拖延症，在每一天他会处于没有拖延症的正常状态 Normal、以及不同程度的拖延症 Light、Heavy 和 Critical 状态中的一种。每天的状态会随着前一天所处的状态不同而发生改变，转移方式如图 1 所示。

简单来说：小明一开始会处于正常状态，不过由于他拖延症非常严重，第二天毫无悬念地会进入轻度拖延症状态。在轻度拖延症状态中有很大的概率 (0.7) 会进入重度拖延症状态或者以 0.3 的概率维持在轻度拖延症状态中。一旦进入到重度拖延症状态，他会以 0.8 的概率一直保留在那个状态，或者有比较小的几率 (0.2) 进入 “致命拖延” 状态。在 “致命拖延” 状态中度过一天之后小明会幡然醒悟，下定决心重新做人，并在第二天成功回复正常状态。然后.....周而复始、世袭罔替.....

不过，小明的拖延症状态是 “隐藏” 在他大脑里的（这也是 HMM 中 Hidden 的由来），他自己也搞不清楚。但是我们知道他在不同的状态下会做什么样的事情。

Table 1: 小明在不同状态下打牌的概率

状态	打牌的概率	不打牌的概率
Normal	0	1

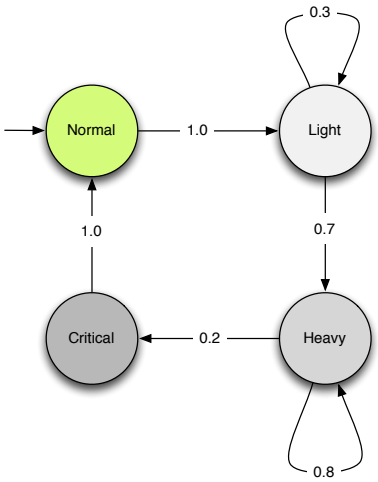

Figure 1: 小明的拖延症状态转移图

状态	打牌的概率	不打牌的概率
Light	0.3	0.7
Heavy	0.8	0.2
Critical	1	0

虽然我们没法把小明的脑袋打开看看里面的寄存器是什么状态，但是我们可以~~偷看小明的日记~~观察小明的日常生活。通过这些历史数据，我们可以做这样一些事情：

- 给定小明某一段时间的日记（打牌、不打牌），计算该日记是否是伪造的概率。更确切地说，计算该日记所记录的日常生活是来自于小明的拖延症模型的概率。
- 给定小明某一段时间的日记，推断出每一天小明最有可能处在什么状态。

另外，如果我们并不事先知道小明的拖延症模型（状态转移和不同状态下的行为），如果有足够多的历史数据（日记），我们还可以做的第三件事情就是：

- 估计小明的拖延症模型参数。

这三件事正好对应了 HMM 中的三个任务，分别是 Scoring、Matching（或者 Decoding）、Traing （或者 Learning）。对应这三个任务分别有三个算法：

- Scoring: Forward-Backward 算法，是 Graphical Model 里的 Sum-Product 算法的特例。
- Matching: Viterbi 算法，是 Graphical Model 里的 Max-Product 算法的特例。
- Training: Baum-Welch 算法，是 EM 算法的特例。

熟悉 Graphical Model 的同学肯定一下子能看出来，前两个问题属于 Inference 问题，分别就是算 Marginal 和 MAP 推断；而最后一个问题则是 Learning 问题，之所以 EM 算法也是因为状态是没有观察到的隐变量。由于三个问题都定义得非常清楚，而且也有非常高效的算法进行计算，加之 HMM 模型的适用范围非常广，所以在诸如语音识别、自然语言处理、机器翻译、基因序列对齐、机器人定位等等各种各样的应用中得到广泛采用。虽然它提出来也已经有相当年头了，并且也有自己的局限和问题，但是，比如说，现在的 state-of-the-art 的语音识别系统仍然主要是基于 HMM 框架的。

具体来说，HMM 可以定义为一个三元组 π, A, B。其中 π 用于描述初始状态的分布，在小明的例子里初始状态是 deterministic 地位于 “Normal” 状态的，General 的 HMM 并不要求这样。A 则是状态转移矩阵，a_{ij} 表示在 t 时刻处于状态 s_i 的情况下，$t+1$ 时刻转移到状态 s_j 的概率：

$$a_{ij} = \mathbb{P}(q_{t+1} = s_j | q_t = s_i)$$

注意 $t+1$ 时刻的状态分布只依赖于 t 时刻的状态，而与更之前的状态无关 (independent)[1]，这类的性质通常叫做 Markov Property，这也是为什么 HMM 叫做 "Markov Model" 的原因。除了状态有这个性质之外，每个时刻所得到的观察值也有类似的性质：t 时刻的观察值（打牌还是不打牌）o_t 只依赖于 t 时刻的状态 q_t，并由三元组中的 B 来决定，具体来说

[1] 更确切地说，是 q_{t+1} 在给定了 q_t 的情况下与更早的 q_{t-1} 等状态无关，亦即是 Conditional Independence。条件独立性是 Graphical Model 里的核心概念。下面关于 "观察值" 的独立性类似。

$$b_{ij} = \mathbb{P}(o_t = v_j | q_t = s_i)$$

对于小明的例子，我们实际上是用状态转移图来给出了 A 和 π，并用表格的形式给出了 B。由于 HMM 总是对应一个 sequence （的状态或者观察值），所以另一个非常常见的 HMM 的表达方式是用每个时刻 t 的对应的状态 q_t 和观察值 o_t 的随机变量的 Graphical Model 来表示，例如我们直接从 Wikipedia 上盗用一个图（它这里用 x 表示状态，y 表示观察值）：

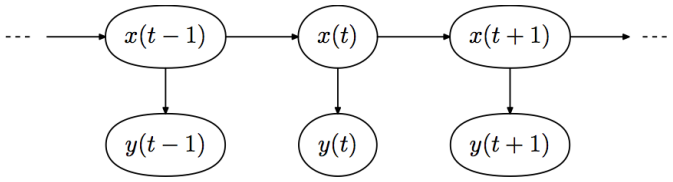

Figure 2: .....

从这个 Graphical Model 上能更加清楚地看出各个随机变量之间的条件独立性，也就是 Markov 性质。不过不要忘记的是在这个图模型中每个横向箭头所对应的转移概率函数全都是一样的，同样，纵向箭头所对应的观察值生成概率分布也是所有时刻公用的。

最后，假设胡适先生上面的日记日期是连续的话，我用 Viterbi 算法算了一下，最可能的状态 sequence 是：他进入了 "重度拖延症" 状态之后就再也没能出来过。`^_^bb` 嘛，也许是模型参数设得不够好呢？

注：由于现在越来越忙，长篇大论的博客能写出来的频率也越来越低，为了让 blog 不止于荒废掉，我决定尝试着写一些比较短的文章，在需要的情况下把一些长文分成系列文章分期完成。当然风险就是有可能烂尾，不过写出一部分来似乎也总比整个憋不出来要好啊？`^_^bb` 所以，关于 HMM 的三种算法的细节以及相关的应用的介绍之类的，就下次再说了。