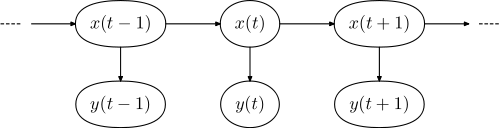
**隐马科夫(HMM)模型**

HMM已经有很多人研究过了，而我要研究它的目的只是为了那个让我我纠结多天的Hidden-state Variable。说起这个Hidden-state Variable，真是惭愧，应该是我很笨或者很懒的问题，从JMLMIL理论开始，我就一直开始搞不懂Hidden-state Variable究竟是怎么样来的，虽然我知道它的物理意义和怎么使用。可能很多人看论文都会有这样的感觉吧，即使当你清楚了某个东西的框架，也认同这个框架，但是当你细细深入并打算实现的时候，才会发现现实跟框架是有很多差距的啊！

不好意思，扯远了。我从JMLMIL开始追寻这个Hidden-state variable，到HCRF，到CRF，到HMM。而如今对HMM的理解算是对这个Hidden-state Variable的理解的一个开始吧。当我完全搞懂这个Hidden的东西之后，我会详细的把这些东西都串起来。现在，先看看HMM吧。

隐马科夫(HMM)模型全称：Hidden Markov model，是一种统计学的模型，是马科夫链与无法观察的状态的结合。在这里，我假设看这篇文章的人已经对马科夫过程([**Markov Process**](http://en.wikipedia.org/wiki/Markov_process))有了初步的认识，对概率论有过初步的学习。

用一句话描述马科夫过程，就是后一个事件发生的概率只与当前时间发生的概率相关。在下面的图里面，我们可以清楚看到”马科夫过程”这一属性，注箭头意x(t-1), x(t), x(t+1)之间的的指向，就用这个箭头表示他们之间的关系，这个箭头是单向的，对于t这一个时间的状态x(t)，只有t-1时刻的状态x(t-1)指向它，说明影响t时刻的状态x(t)的，只有t-1时刻的状态x(t-1)。

**[](http://www.suzker.cn/wp-content/uploads/2010/05/500px-Hmm_temporal_bayesian_net.svg_.png)**

Hmm temporal bayesian net

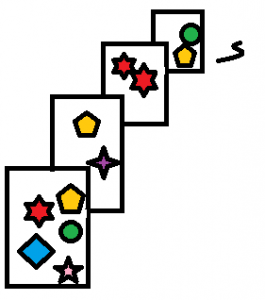
那么这个HMM模型，隐马模型到底“隐”在哪里呢？其实解释起来也不难，这里…t-1, t, t+1…各个时刻的x状态是一个随机过程，试想一下，你总不能确定一个随机过程中的每个状态吧？所以，上图的x状态对于我们来说，在没有到达t时间之前，他仍然是未知的，隐藏的（Hidden）。这些x被称为“unobserved state”。

**Turn to the next page for more.**

了解了这个“隐”的概念之后，我们可以深入HMM的一个整体结构和过程了。

为了更好理解，我引用一个实例来说明HMM的结构和过程。

假设我们有5个筐装着很多不同水果，水果的种类有6种（下图中用不同颜色的图案表示）。

**[](http://www.suzker.cn/computervision/the-hmm-model.html/attachment/hmm_1)**

HMM模型的实例\_01

HMM过程是，我从任意一个筐开始选水果，我去到那一个框那里，随机拿起里面的一个水果，然后把这个水果记录下来，然后再随机地去另外一个筐里面选水果，不断地重复这个过程，知道我选够了L个水果，我就停止。

在我选水果的过程中，我将遵循以下准则：

* 我从某一个框选完水果后移到另外一个框的概率用一个 Transition-probabilty Matrix来表示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 筐     |     筐 | 1 | 2 | 3 | 4 | 5 |
| 1 | 0.5 | 0.1 | 0.2 | 0.2 | 0 |
| 2 | 0.3 | 0.4 | 0.1 | 0.1 | 0.1 |
| 3 | 0.1 | 0.5 | 0.1 | 0 | 0.3 |
| 4 | 0.3 | 0.3 | 0.2 | 0.1 | 0.1 |
| 5 | 0 | 0.1 | 0.2 | 0.1 | 0.6 |

* 而我从第i个框中选6仲水果中的概率也用一定的概率分布表示，称为这个框中水果的概率分布(Emission-probability Matrix)。（以下概率值省略）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 筐   |   水果 | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | … |  |  |  |  |  |
| 2 |  | … |  |  |  |  |
| 3 |  |  | …. |  |  |  |
| 4 |  |  |  | … |  |  |
| 5 |  |  |  |  | … | …. |

* 我们还必须牢记一点，因为下一次转移到哪一个框是不确定的，所以当我还没选够水果的之前，你还是不会准确得知我的路径的。我选的水果你也将不会准确得知。

实例的一个结果是这样的： 我选完水果之后，发现我的路径是[筐2>筐1>筐1>筐5>筐2>筐3>筐4]，即我选了7个水果，L=7。而我最终选出来的水果依次是[f2, f1, f3, f5, f5, f2, f4]。

之前已经假设过大家已经了解了一点Markov process和随机过程，所以这里你们应该会很自然地想到其实这个HMM就是这两个过程的结合：

[Markov process：选框] > [普通随机过程：选水果]；

假设我们的模型参数是 lambda(N, M, pi, A, B);

* L: 将会观察的长度。对应上面例子中的我将要选取的水果个数。
* N: 状态数目。对应上面例子中，水果筐的个数。
* M:每个状态可能的观察值数。对应上面例子中，水果的种类。
* pi: 初始状态空间的分布。对应上面例子中，哪个水果筐将会成为我选的第一个筐的概率。
* A: 代表状态转移矩阵(Transition-probability matrix)。对应上面例子中，我从某个筐移到另外一个筐的概率。
* B: 代表观察值的概率分布(Emission-probability matrix)。对应上面例子中，某个筐中，水果的分布概率。

状态的序列标记为S。对应上面例子中，我最终选筐的移动路径[筐2>筐1>筐1>筐5>筐2>筐3>筐4]。

观察到的序列标记为O。对应上面例子中，我最终选出来的水果序列[f2> f1> f3> f5> f5> f2> f4]。

HMM将涉及以下三个问题：

1. 给定一个观察到得序列O，及参数lambda，求出P(O|lambda)，即发生这种观察序列的可能性。对应上面例子中，即是我给定一个最终确定了的水果序列S，求我选到这样的水果的可能性。
2. 给定一个观察到的序列O，及参数lambda，求出最有可能产生这种序列的状态序列S。对应上面例子中，即我给定一个最终确定了的水果序列S，求我最可能的选水果筐路径。
3. 同样给定一个观察得到的序列O，求如何调整参数lambda，使P(O|lambda)最大。

由于涉及具体算法和论证，这里不再详述这三个问题的解决算法。只简单地标出：

1. 问题一的一种有效解决算法是前向法([**forward algorithm**](http://en.wikipedia.org/wiki/Forward_algorithm))，是一种**[动态规划](http://en.wikipedia.org/wiki/Dynamic_programming" \t "_blank)**的方法，用来减少程序运算的复杂度。
2. 问题二的一种有效解决算法是维特比算法([**Viterbi algorithm**](http://en.wikipedia.org/wiki/Viterbi_algorithm))，也是一种动态的规划方法，用来找出最可能的状态路径。
3. 问题三的一种有效解决算法是[**Baum-Welch**](http://en.wikipedia.org/wiki/Baum-Welch_algorithm)算法，通过给定一个O，不断估算一个适合的lambda参数，使发生这个O的概率P(O|lambda)最大。