

漫谈 HMM: Forward-Backward Algorithm

<http://freemind.pluskid.org/machine-learning/hmm-forward-backward-algorithm>

上一次我们介绍了 HMM 模型以及它相关的三个 Task，这一次我们来介绍一下第一个 Task，也就是 Scoring，或者说是 Inference。为了说明 HMM 的多才多艺，我们这次也换一个完全不同的应用场景。

现在将镜头切换到火星探测车上。我们需要做的是，比如说，让火星车移动到五十米开外的那个小山坡（比如，绝对位置坐标为 z^* ）上去。听上去似乎是再简单不过的任务，可是这个任务会碰到什么问题呢？



posted on [Free Mind](#) on May 18, 2013
generated with pandoc on December 3, 2015
category: Machine Learning

tags: Algorithm, Probabilistic Graphical Model

Figure 1:

首先，你不是很清楚火星车目前的位置 z_0 ，话说回来，我们究竟怎么知道一个东西的位置呢？在地球上我们或许可以用 GPS 或者 WIFI 信号定位之类的，亦或者我们通常可以通过观察周围的环境啊建筑物之类的来得到一个大概的方位。不过，无论通过什么方法，有一个共同点就是：得到的坐标都是有误差的。比如说，我们假设火星车可以通过某个在火星周围的卫星来对自己进行定位，通常我们将定位误差（好吧，其实基本上啥误差都这么搞）通过 Gaussian 分布来建模，设火星车真实的位置为 z 的话，那么通过定位系统读出来的位置 x 满足如下 Gaussian 分布：

$$x \sim \mathcal{N}(z, \sigma^2)$$

当然， σ^2 越大，定位就越不准。除了定位不准之外，还有另一个问题：比如我们假设现在定位是 100% 精确的，我已经测量出目的地就在我正前方五米处，然后我决定向前走五米，然而在我向前走的期间，突然碰到一颗石头绊了一跤，或者突然刮来一阵火星风，或者我突然没电

了，或者我轮子上的某颗螺丝松了.....总而言之，当我（自己认为）完成了“向前走五米”的动作之后，我所处的位置通常并不 **exactly** 在正前方五米处，而是不知道跑到哪个星球上去子只能实现在某个范围内的精确度。比如，我们再次将它建模为一个 **Gaussian** 分布。设当前的位置为 z ，假设我进行了某个移动操作 $f(z)$ ，那么我的结果位置 z' 满足如下 **Gaussian** 分布：

$$z' \sim \mathcal{N}(f(z), \sigma^2)$$

这些模型建立起来之后，是不是已经俨然一个 **HMM** 模型出现了？只是和之前的例子有所不同的是，现在状态 z 也好，观察值 x 也好，都变成了连续的 \mathbb{R}^n 上的值，而不再是简单的离散变量。不过我们依然假设时间仍然是一个离散的具有一个一个“时刻”的量。

有了模型之后，终于要进入正题了，最简单的一种 **Inference** 就是计算“我现在在哪”的问题。简单来说，现在是 T 时刻，我们有从第 0 时刻到目前为止每一时刻的定位观测值 x_0, \dots, x_T ，并且我们知道每一步执行的移动操作。现在我要问的是，我目前的真实位置 z_T 是什么，或者更确切地说，满足什么分布。也就是计算 $P(z_T | x_0, \dots, x_T)$ ，随机变量 z_T 的（条件）**Marginal** 分布。根据 **Marginal** 分布的概念，我们可以有如下的公式：

$$\begin{aligned} P(z_T | x_0, \dots, x_T) &= \int_{z_{T-1}} \dots \int_{z_0} P(z_0, \dots, z_T | x_0, \dots, x_T) dz_0 \dots dz_{T-1} \\ &= \int_{z_{T-1}} \dots \int_{z_0} \frac{P(z_0, \dots, z_T, x_0, \dots, x_T)}{P(x_0, \dots, x_T)} dz_0 \dots dz_{T-1} \end{aligned}$$

为了方便起见，我们用 x_0^T 这样的下标上标组合表示一个 **sequence** x_0, \dots, x_T 。注意到在求 $P(z_T)$ 这个分布的时候分母上的 $P(x_0^T)$ 是个常数，所以我们不用专门对它进行计算，只要在最后对我们算出来的分布进行重新 **normalize** 即可。

接下来就是 **Graphical Model** 的内容，从图 2 中的条件独立性关系可以看出来，**HMM** 的联合概率分布可以展开成如下形式：

$$P(z_0^T, x_0^T) = P(z_0)P(x_0|z_0)P(z_1|z_0)P(x_1|z_1) \prod_{t=2}^T P(z_t|z_{t-1})P(x_t|z_t)$$

这对于原来式子中的积分计算非常有帮助，例如，我们如果先计算针对 z_0 的积分，会发现上面的展开式中只有红色的三项和 z_0 是有关系的，所以其他的项可以直接拿到这个积分号外面：

$$M_{0 \rightarrow 1}(z_1) = \int_{z_0} P(z_0)P(x_0|z_0)P(z_1|z_0) dz_0$$

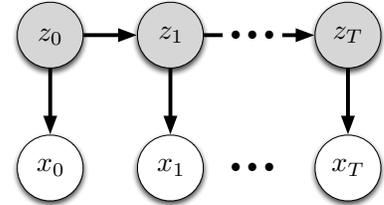


Figure 2: 火星车的 HMM。

将该式子替换回原来的大积分式之后，变成这个样子：

$$P(z_T|x_0^T) \propto \int_{z_{T-1}} \dots \int_{z_1} M_{0 \rightarrow 1}(z_1) P(x_1|z_1) P(z_2|z_1) P(x_2|z_2) \prod_{t=3}^T P(z_t|z_{t-1}) P(x_t|z_t) dz_1 \dots dz_{T-1}$$

类似地，注意到对于 z_1 的积分，只有红色的三项是有关的，其余的可以直接拿到该积分号外面：

$$M_{1 \rightarrow 2}(z_2) = \int_{z_1} M_{0 \rightarrow 1}(z_1) P(x_1|z_1) P(z_2|z_1) dz_1$$

如果我们将 $P(z_0)$ 记为 $M_{-1 \rightarrow 0}(z_0)$ 的话，可以很容易得到如下的递推公式：

$$M_{t \rightarrow t+1}(z_{t+1}) = \int_{z_t} M_{t-1 \rightarrow t}(z_t) P(x_t|z_t) P(z_{t+1}|z_t) dz_t$$

以及：

$$P(z_T|x_0^T) \propto M_{T-1 \rightarrow T}(z_T) P(x_T|z_T)$$

每一个 $M_{t \rightarrow t+1}(z_{t+1})$ 看起来就像是节点 t 到节点 $t+1$ 传递的消息，整个算法的工作流程就是从最左边开始，从左到右计算消息，并依次传递到右边。所以这种算法又叫做 **Message Passing**，或者 **Belief Propagation**。然后 HMM 里也将这样的算法称为 **Forward Algorithm** —— 更确切地说，还有一些细节上的小差别。**Forward Algorithm** 里从左到右依次计算的是另一个量

$$\alpha(z_{t+1}) = P(x_0^{t+1}, z_{t+1}) = M_{t \rightarrow t+1}(z_{t+1}) P(x_{t+1}|z_{t+1})$$

并且递推公式可以改为

$$\begin{aligned} \alpha(z_{t+1}) &= M_{t \rightarrow t+1}(z_{t+1}) P(x_{t+1}|z_{t+1}) \\ &= \int_{z_t} M_{t-1 \rightarrow t}(z_t) P(x_t|z_t) P(z_{t+1}|z_t) dz_t P(x_{t+1}|z_{t+1}) \\ &= \int_{z_t} \alpha(z_t) P(z_{t+1}|z_t) dz_t P(x_{t+1}|z_{t+1}) \end{aligned}$$

这样有一个好处就是，我们所求的 z_T 的分布直接就是

$$P(z_T|x_0^T) \propto \alpha(z_T)$$

对于之前介绍的离散的 HMM 的例子，全部推导完全一样，只要把积分号换成求和符号即可。这样一来，我们的问题也就解决了。可是不是说 HMM 的 Inference 算法叫做 Forward-Backward 吗？怎么缺了一半？实际上，Backward 算法被用于解决另一个问题：假设我们在移动的过程中做了一些测量或者记录工作，比如火星车所处位置的温度啊、土壤啊之类的，然后我们想把之前所记录下来的数据和该处的位置坐标一起保存下来以供以后参考。一个办法就是使用刚才的 Forward 算法直接计算 $P(z_t|x_0^t)$ ，然而这个分布只是参考了从出发点到 t 时刻的位置观察值得到的，仔细想一下的话，如果我们把 t 时刻以后的观察值 x_{t+1}, \dots, x_T 全部考虑进去的话，应该会得到一个更加精确的真实位置估计才对。于是，为了计算

$$P(z_t|x_0^T) = \frac{P(z_t, x_0^T)}{P(x_0^T)} \propto P(z_t, x_0^T)$$

只需要注意到

$$\begin{aligned} P(z_t, x_0^T) &= P(z_t, x_0^t, x_{t+1}^T) \\ &= P(z_t, x_0^t)P(x_{t+1}^T|z_t, x_0^t) \\ &= \alpha(z_t)P(x_{t+1}^T|z_t, x_0^t) \\ &= \alpha(z_t)P(x_{t+1}^T|z_t) \end{aligned}$$

其中最后一步是根据 HMM 的图模型所具备的条件独立性得到的。在 HMM 中红色的部分被定义为 $\beta(z_t)$ 。由于 $P(z_T, x_0^T) = \alpha(z_T)$ ，所以 $\beta(z_T)$ 就直接等于 1。然后和计算 α 的时候类似地，我们可以得到一个关于 β 的递推公式：

$$\begin{aligned} \beta(z_t) &= P(x_{t+1}^T|z_t) \\ &= \int_{z_{t+1}} P(x_{t+1}^T, z_{t+1}|z_t) dz_{t+1} \\ &= \int_{z_{t+1}} P(x_{t+1}^T|z_{t+1})P(z_{t+1}|z_t) dz_{t+1} \\ &= \int_{z_{t+1}} P(x_{t+1}|z_{t+1})P(x_{t+2}^T|z_{t+1})P(z_{t+1}|z_t) dz_{t+1} \\ &= \int_{z_{t+1}} P(x_{t+1}|z_{t+1})\beta(z_{t+1})P(z_{t+1}|z_t) dz_{t+1} \end{aligned}$$

这里又多次用到了 HMM 中的条件独立性，所以说条件独立性可以算是 Graphical Model 中的核心。然后因为这个 β 的递推式是从后面往前面递推的，所以计算 β 的过程叫做 Backward 算法。当然，如果放在 Message Passing 的框架下面看的话，Forward 和 Backward 没有任何本质上的不同，只是消息传递的方向不一样而已。通过一轮 Forward 和一轮 Backward 算法的计算，我们就可以得到在任意时刻 t 的位置估计：

$$P(z_t | x_0^T) \propto \alpha(z_t) \beta(z_t)$$

不过话说回来，我们最开始介绍 HMM 的三个 Task 的时候，所说的 Scoring 似乎并不是这样的，而是说给定一串 observation x_0^T ，计算它们的 likelihood $P(x_0^T)$ 。其实这个东西和我们这里所做的计算几乎是一回事，因为我们只要再在任意一个时刻 t 处做一次 Marginalization 就可以得到：

$$P(x_0^T) = \int_{z_t} P(z_t, x_0^T) dz_t = \int_{z_t} \alpha(z_t) \beta(z_t) dz_t$$

到这里为止，Forward-Backward 算法可以说是讲完了，特别是对于离散的情况，只要正向反向各遍历一遍求和就可以了。但是对于连续的情况我们要算的是一个积分，任意的积分一般是没法算的，所以大家通常使用 Gaussian 分布来作为其中的概率模型（并以“Gaussian 分布是最通用的用于建模误差的分布”作为借口）。最后实在是和 Gaussian 扯不上什么关系，或者是必须要用其他不好计算积分的分布的时候，也可以使用 Approximation 的方法。不过详情又要等下次再来细说了。^ ^