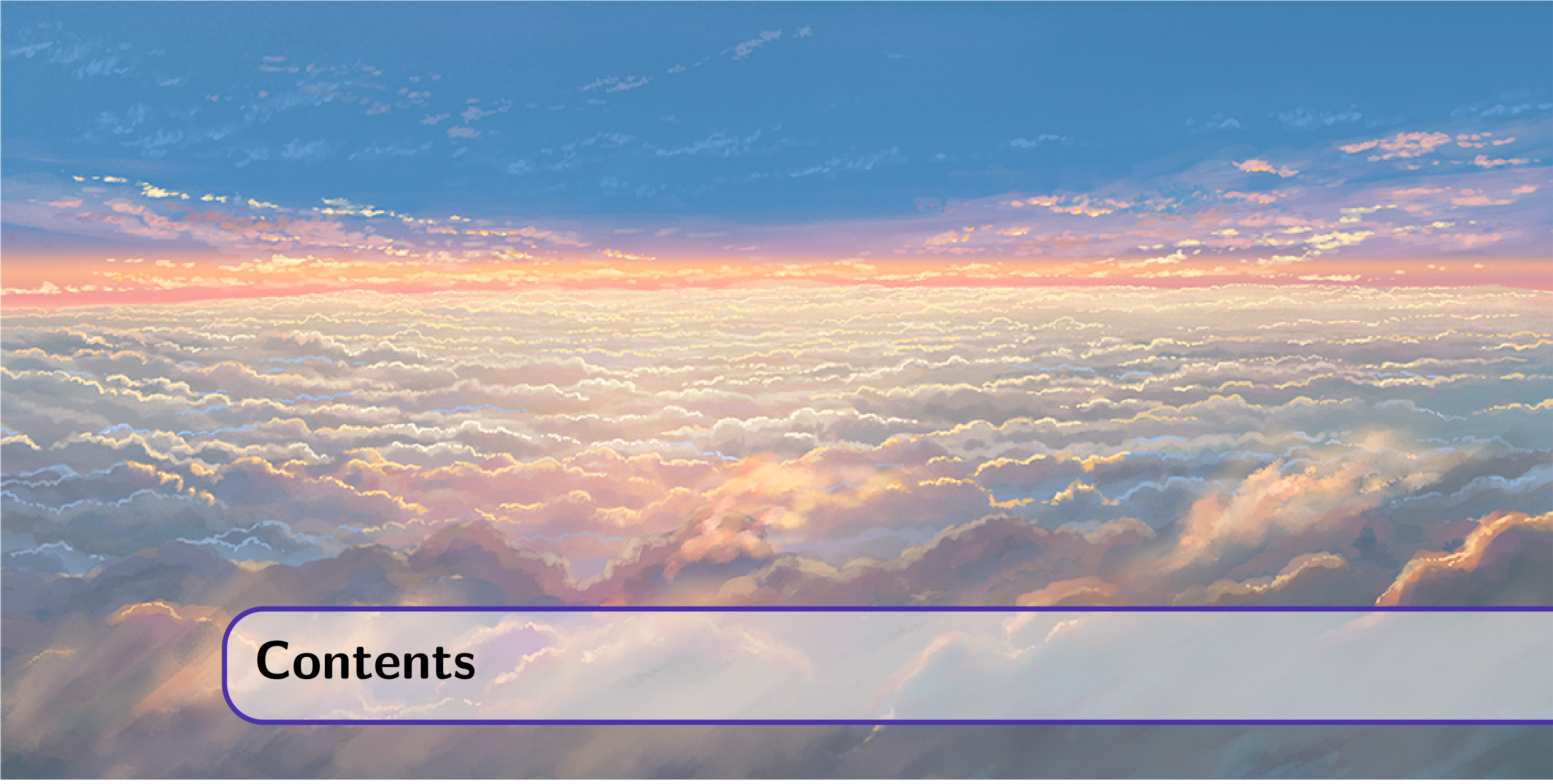
Written by Xia Wenxuan, 2021

Published by myself

`https://github.com/gegeji`

First printing, October 12, 2021



Contents

I Map Matching

II 论文相关概念

I Map Matching

1. Hidden Markov Map Matching

Source: [2].

1.1 论文动机

1.1.1 原始数据

the raw input data consists of vehicle *locations* measured by GPS, Each measured point consists of a time-stamped latitude/longitude pair.

The *roads* are also represented in the conventional way, as a graph of nodes and edges. The *nodes* are at intersections, dead ends, and road name changes, and the edges represent road segments between the nodes. Some *edges* are directional to indicate one-way roads. Each node has an associated latitude/longitude to indicate its location, and each edge has a polyline（折线）of latitude/longitude pairs to represent its geometry.

1.2 其他论文的方法

平滑曲线匹配：create a (possibly smoothed) curve from the location measurements and attempt to find matching roads with similar geometry

■ **Example 1.1** White et al. present four algorithms, starting with the simple, nearest match scheme. Their second algorithm **adds orientation information to the nearest match approach**, comparing the measured heading to the angle of the road. Their third algorithm evolves the second algorithm to **include connectivity constraints**, and their fourth algorithm does **curve matching**.

> their most sophisticated algorithm, the fourth one, was outperformed by the simpler second algorithm.

通过拓扑结构建模：builds up a topologically feasible path through the road network. Matches are determined by a similarity measure that weights errors based on distance and orientation. The algorithm was found to perform flawlessly, even though the GPS data was collected while *Selective Availability* was turned on, leading to noisier location measurements than are available now.

模糊匹配策略：Kim and Kim look at a way to measure **how much each GPS point belongs to any given road**, taking into account its distance from the road, the shape of the road segment, and the continuity of the path. The measure is used in a **fuzzy matching scheme** with learned parameters to optimize performance.

Brakatsoulas et al. Their algorithm uses variations of the *Fréchet distance* to match the curve of the GPS trace to candidate paths in the road network.

!

One potential problem with purely geometric approaches is **their sensitivity to measurement noise and sampling rate.** Connecting the dots of a set of noisy measurements sampled at a slow rate would not match well with the road geometry, especially **direction information**.

基准方法：将 GPS 点匹配到最近邻的路上

!

result in extremely unreasonable paths involving strange U-turns, inefficient looping, and overall bizarre driving behavior.

1.3 论文贡献

- maintaining a principled approach to the problem while simultaneously making the algorithm robust to location data that is both **geometrically noisy and temporally sparse**
- a test of our map matching algorithm where we vary the levels of noise and sparseness of the sensed location data over a 50 mile urban drive

1.4 论文模型-Hidden Markov Model

The HMM models processes that involve a path through many possible states, where some state transitions are more likely than others and where the state measurements are uncertain.

1.4.1 HMM

- HMM 状态：N_r individual road segments $r_i, i = 1 \dots N_r$
- 状态的测量：每次带噪声的位置数据 Z_t
- 候选路径：有很多，可能有很曲折的
- 目标：将每个 GPS 点匹配到合适的路段上

1.4.2 已知在这个路段得到这个 GPS 点位置的概率估计

对于给定的 z_t, r_i 有 *emission probability*$x_{t,i}$。

估计方法：The *great circle distance* on the surface of the earth between the measured point and the candidate match is $\|z_t - x_{t,i}\|_{great\_circle}$. For the correct match, this difference is due to GPS noise. 噪声认为是零均值的高斯噪声。

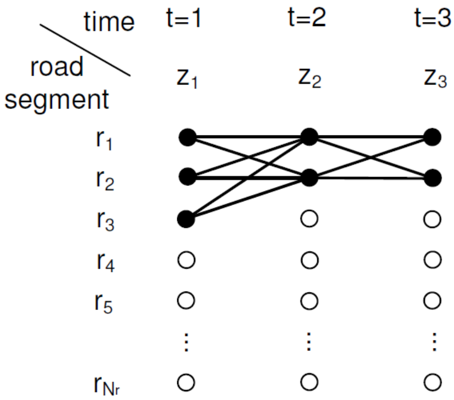


Figure 3: For each measurement z_t, the HMM considers all the road segments r_i as well as all the transitions between the road segments.

$$p\left(z_t \mid r_i\right)=\frac{1}{\sqrt{2 \pi} \sigma_z} e^{-0.5\left(\frac{\left\|z_t-x_{t, i}\right\|_{\text {great circle }}}{\sigma_z}\right)^2}$$

σ_z 是 GPS 测量的方差（需要估计）。

对于初始状态 $\pi_i, i=1 \ldots N_r$（指定一开始车辆在所有路段上的可能性），为了简化使用

$$\pi_i=p\left(z_1 \mid r_i\right)$$

1.4.3 转移概率

Each measurement z_t has a list of possible road matches, as does the next measurement z_{t+1}.

作者认为**测量距离和大圆距离相近的状态转移才是比较好的**，否则会绕路（概率下降）。

计算发现大圆距离与路径距离的差值绝对值近似于：

$$p\left(d_t\right)=\frac{1}{\beta} e^{-d_t / \beta}$$

路径距离 $d_t=\left|\left\|z_t-z_{t+1}\right\|_{\text {great circle }}-\left\|x_{t, i^*}-x_{t+1, j^*}\right\|_{\text {route }}\right|$ 是动态规划 (*Viterbi algorithm*) 得到的路线距离。对于 z_t 和候选路段 r_i，匹配出在路段上的点是 $x_{t,i}$。

!

the transition probability dependence on the current and previous observations violates the properties of an ideal HMM

the great-circle distance between the observed locations may introduce significant errors in estimating the circuitousness of a path (在定位误差大的情况下)

transition probabilities computed based on the above measure of circuitousness vary greatly for equally plausible transition paths depending on the sampling interval

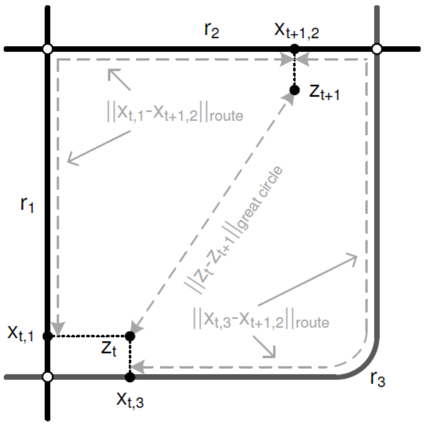


Figure 4: This shows an example of our notation. There are three road segments, r_1, r_2, and r_3, and two measured points, z_t and z_{t+1}. The first measured point, z_t, has candidate road matches at $x_{t,1}$ and $x_{t,3}$. Each match candidate results in a route to $x_{t+1,2}$, which is a match candidate for the second measured point, z_{t+1}. These two routes have their own lengths, as does the great circle path between the two measured points. Our data shows that the route distance and great circle distance are closer together for correct matches than for incorrect matches.

1.5 算法的实现

- 数据预处理的时候对于离上一个 GPS 点位置超过 $2\sigma_z$ 的点进行剔除
- 将离 GPS 位置观测值 200 米以外路段的观测概率设为 0
- 对于大圆距离和路径距离超过 2 公里路径的观察概率设为 0
- 对于明显超速的路径的概率设为 0
- 当明显无法匹配的时候，移除一些点，尝试重新连接，如果仍旧无法连接，引入匹配断点

1.5.1 参数估计

估计 GPS 误差：σ_z using the *median absolute deviation (MAD)*

$$\sigma_z = 1.4826\mathrm{median_t}\left(\|z_t - x_{t,i}\|_{\text{great circle}}\right)$$

the point on r_i^* （人工匹配的路段）nearest z_t is $x_{t,i}^*$.
估计路径距离与大圆距离之间的差值：

$$\beta = \frac{1}{\ln(2)}\,\mathrm{median}_t\Big(\big|\ \|z_t - z_{t+1}\|_{\text{great circle}} \\ - \|x_{t,i^*} - x_{t+1,j}*\|_{\text{route}}\ \big|\Big)$$

1.6 实验

数据：真实驾车 GPS 数据、高斯噪声模拟数据
评价指标：reported error

$$(\mathrm{d}- + \mathrm{d}_+)/\mathrm{d}_0$$

d_- =length erroneously subtracted, d_+=length erroneously added
其他两种指标：
Locations on Road.

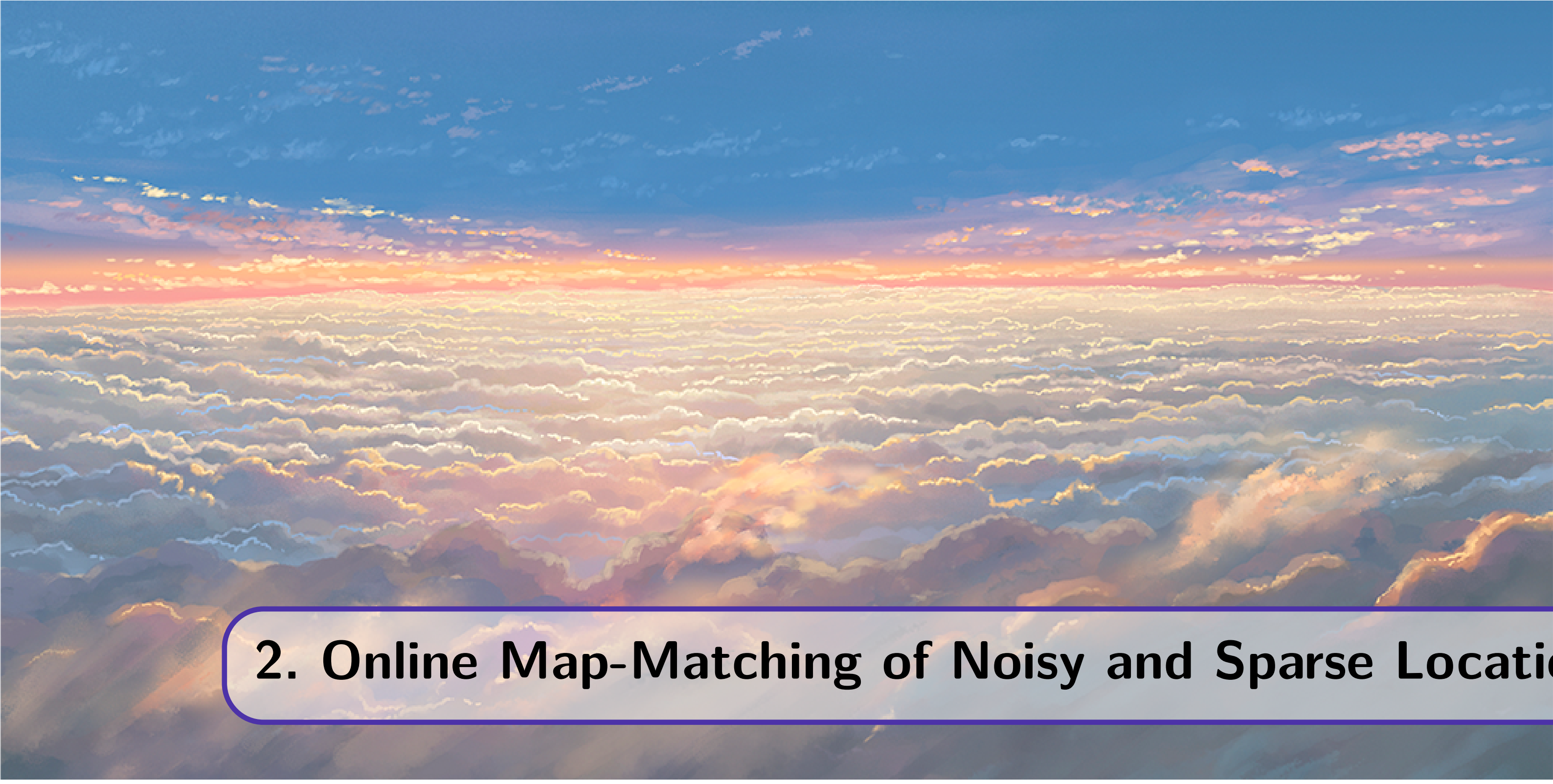
!

This accuracy measure says that the matched point should be in the same location as the actual vehicle. Since we measured the vehicle's location with inherently noisy GPS, we do not know its actual location.

Road Segment.

!

This accuracy measure says that the matched point should be on the same road segment as the actual vehicle. While the correct road segment is easier to guess than the correct location, it is still ambiguous at intersections, where a noisy measurement could match to any of the roads converging at that point.



2. Online Map-Matching of Noisy and Sparse Locati

Source: [1]

2.1 论文动机

蜂窝网络/WiFi 定位不精确性更高，现有算法对于这两种定位的处理不足够。

利用司机的驾驶选择信息 (*route choice*). 最简单办法：决定性的路径选择准则

■ **Example 2.1** Koutsopoulos generated **a number of candidate paths** in a search graph and selected the path with the minimum length from among a subset of paths whose **travel times are within a threshold**.

Sarlas replaced the deterministic route choice criteria with a **probabilistic logit route choice model**.

Miwa et al. applied a logit route choice model in a **piecewise manner for short portions of the location sequence** to select the best among several candidate paths. ■

在线匹配：Instead of generating an unique best fitting path, their solution produces **a set of potential paths along with a likelihood** that the location measurements are recorded from each path.

[2] 用到了 HMM，此外 *Conditional Random Fields (CRF)*也用于地图匹配

2.2 本文贡献

拓展了 HMM 的地图匹配算法、引入了司机路径选择的模型

2.3 论文模型-HMM Based Online Map-Matching

2.3.1 基本概念

A *location observation* (or simply *observation*) o_t consists of the measured latitude, longitude and timestamp corresponding to a mobile user's location at time step t.

A *road network* is a directed graph $G = (V, E)$, where V is a set of nodes corresponding to intersections or endpoints of the road segments and E is a set of edges representing *road segments*. Each road segment has a number of attributes including road class, length and *free-flow travel time*.

A *path* p between nodes u and v is a sequence of connected road segments (edges) $e_1, \ldots, e_n$ such that u is the start node of e_1 and v is the end node of e_n.

!

本文假设：a path does not necessarily start and end at nodes. It could start or end at any point that lies along the centerline of any road segment.

denote the k^{th} state at time step t as $s_{t,k}$. The hidden true state at time step t is denoted as s_t^*.

Given a sequence of N observations $o_{1:N} = \{o_1, \ldots, o_N\}$ and a road network G, the map-matching problem is to find the path p in G corresponding to $o_{1:N}$.

2.3.2 HMM 模型的建立

The location observations are subject to measurement noise and therefore the true on-road locations corresponding to them are unknown.

在实际中只考虑在 GPS 测量点附近的路段（在 4 倍标准差以内）

HMM 性质：

- the observation o_t at time step t depends only on the hidden state s_t^* at that time.
- (*Markov property*) states that the hidden state s_t^* at time step t depends only on the hidden state s_{t-1}^* at time step $t-1$ and is not influenced by the history of hidden states before that.

对于一个 GPS 记录点 o_t, each state $s_{t,k}$ is assigned an *emission probability* $P(o_t \mid s_{t,k})$。状态转移概率是 $P(s_{t,k} \mid s_{t-1,j})$

2.3.3 Emission Probability

(同 [2])

For a given state $s_{t,k}$ and a location observation o_t, the emission probability is given as

$$P(o_t \mid s_{t,k}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{g(o_t, s_{t,k})^2}{2\sigma^2}}$$

!

the location measurement error may not strictly conform to the above model, especially in dense urban networks. 但是它模型简单而且在前人工作中性能还不错。

2.3.4 Transition Probability

the path with the minimum *free-flow travel time*, found using the well-known Dijkstra's algorithm, as the optimal path.

[2] 用的是大圆距离与路径距离之间的差值。(有问题 1.4.3)

对于空间，本文提出由两个时间状态之内大圆距离与路径距离之间的差值除以时间的比率

$$y\left(s_{t-1,j}, s_{t,k}\right) = \frac{d\left(s_{t-1,j}, s_{t,k}\right) - g\left(s_{t-1,j}, s_{t,k}\right)}{\Delta T}$$

the driving distance between two points on the road network is the length of the minimum-travel-time path between them found by Dijkstra algorithm

对于时间，定义时间上的难以置信性。(**考虑到了时间上的转移概率**)

$$z\left(s_{t-1,j}, s_{t,k}\right) = \frac{\max\left(\left(f\left(s_{t-1,j}, s_{t,k}\right) - \Delta T\right), 0\right)}{\Delta T}$$

where $f\left(s_{t-1,j}, s_{t,k}\right)$ is the *free-flow travel time*, in seconds, of the optimal path between states $s_{t-1,j}$ and $s_{t,k}$.

define the transition probability of moving from state $s_{t-1,j}$ to state $s_{t,k}$ as

$$P\left(s_{t,k} \mid s_{t-1,j}\right) = \lambda_y e^{-\lambda_y y\left(s_{t-1,j}, s_{t,k}\right)} \lambda_z e^{-\lambda_z z\left(s_{t-1,j}, s_{t,k}\right)}$$

2.4 Online Viterbi Inference

维特比算法通过递归算法可以找出 HMM 中最可能的序列。

$$\begin{aligned} V_{1,k} &= P\left(o_1 \mid s_{1,k}\right) \\ V_{t,k} &= P\left(o_t \mid s_{t,k}\right) \max_j \left(V_{t-1,j} P\left(s_{t,k} \mid s_{t-1,j}\right)\right) \end{aligned}$$

$V_{t,k}$ is the most likely state sequence ending at state $s_{t,k}$ based on the observations $o_1, \ldots, o_t$.

j maximizes $V_{t,k}$ is stored as the back pointer for state $s_{t,k}$. It points to the predecessor state $s_{t-1,j}$ of the state $s_{t,k}$ in the most likely sequence ending at the latter.

离线模式: 完整的最优路径可以通过回溯法获得.

Viterbi algorithm needs to be performed in an online manner such that portions of the most likely state sequence are generated from time to time without waiting for all the observations to be received.

In some cases, partial, optimal state sequences can be generated using the concept of *transitive closure*, which defines the reachability relations in a directed graph [20].

启发式策略: 由于当每个时间步的状态很大时计算最短路径的时间可能很大, 因此只保留 k (= 测量误差的方差) 个最高可能性的状态.

2.5 短期路径选择模型

2.5.1 纯 HMM 模型的缺点

计算下一个状态的概率分布时失去了上下文信息.

不满足马尔科夫性 (即受到司机路径选择的影响), 而司机路径选择没有过多地被 HMM 模型所考虑, 有可能会产生在 HMM 模型中较为合理, 但是实际并不合理的路径

概率模型如 CRF 不受马尔可夫独立性假设的影响, 但是高阶 CRF 的参数难以推导. 现有 CRF 方法只考虑一阶 (相邻状态) 的依赖性. 条件随机场?

2.6 多项 Logit 模型 (Multinomial Logit Model)

假设: 路径选择通过 *utility*进行衡量, 司机会选择最高 utility 的路径, 假设所有司机都是相同的. (部分文献会考虑司机的特征, 本文不考虑) 参考 logistic 回归

Given a set of alternative paths, called the choice set, C, the utility associated with alternative path $p_i \in C$ is given by

$$U_i = V_i + \varepsilon_i$$

where V_i is a deterministic term and ε_i is a random term for capturing the uncertainty involved. The deterministic term V_i is modelled as a linear-in-parameters function of the attributes of alternative path p_i.

$$V_i = \beta_{FTT} FTT_i + \beta_{NTS} NTS_i + \beta_{ARC} ARC_i + \beta_{NCC} NCC_i$$

可以化简为:$V_i = \boldsymbol{\beta}' \boldsymbol{x}_i$

FTT: Free-flow travel time (in seconds); NTS: Number of traffic signals; ARC: Average road class (road classes are numbered from 1, starting with the highest);NCC: Number of class changes. In a multinomial logit model, the random term ε_i is assumed to be independent and identically *Gumbel distributed*. (随机因子已被标准化的, 同时假设随机因子/未观察的因素对于所有已观察因素的影响是相等的)

$$P(p_i \mid C) = \frac{e^{\boldsymbol{\beta}' x_i}}{\sum_{p_j \in C} e^{\beta' x_j}}$$

2.7 生成可选路段

上述模型需要计算可选的路段, 可能会非常庞大.

2.7.1 2 种基本方法

k 最短路径

!

考虑整段行程中某几分钟之内的行程.

II 论文相关概念

3. Hidden Markov Model

3.1 隐马尔可夫模型定义

隐马尔可夫模型是关于时序的概率模型，描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列，再由各个状态生成一个观测而产生观测随机序列的过程，隐藏的马尔可夫链随机生成的状态的序列，称为状态序列 (*state sequence*); 每个状态生成一个观测，而由此产生的观测的随机序列，称为观测序列 (*observation sequence*). 序列的每一个位置又可以看作是一个时刻。

Definition 3.1.1 — 隐马尔可夫模型. 设 Q 是所有可能的状态的集合, V 是所有可能的观测的集合.

$$Q=\{q_1,q_2,\cdots,q_N\},\quad V=\{v_1,v_2,\cdots,v_M\}$$

I 是长度为 T 的状态序列, O 是对应的观测序列.

$$I=(i_1,i_2,\cdots,i_T),\quad O=(o_1,o_2,\cdots,o_T)$$

A 是状态转移概率矩阵:

$$A=[a_{ij}]_{N\times N}$$

其中,

$$a_{ij}=P\left(i_{t+1}=q_j\mid i_t=q_i\right),\quad i=1,2,\cdots,N;j=1,2,\cdots,N$$

是在时刻 t 处于状态 q_i 的条件下在时刻 $t+1$ 转移到状态 q_j 的概率.

B 是观测概率矩阵:

$$B=[b_j(k)]_{N\times M}$$

其中,

$$b_j(k)=P\left(o_t=v_k\mid i_t=q_j\right),\quad k=1,2,\cdots,M;j=1,2,\cdots,N$$

是在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率.

π 是初始状态概率向量:

$$\pi = (\pi_i)$$

其中,

$$\pi_i = P(i_1 = q_i), \quad i = 1, 2, \cdots, N$$

是时刻 $t = 1$ 处于状态 q_i 的概率.

隐马尔可夫模型由初始状态概率向量 π 、状态转移概率矩阵 A 和观测概率矩阵 B 决定. π 和 A 决定状态序列, B 决定观测序列. 因此, 隐马尔可夫模型 λ 可以用三元符号表示, 即

$$\lambda = (A, B, \pi)$$

A, B, π 称为隐马尔可夫模型的三要素.

从定义可知, 隐马尔可夫模型作了两个基本假设:

（1）齐次马尔可夫性假设, 即假设隐藏的马尔可夫链在任意时刻 t 的状态只依赖于其前一时刻的状态, 与其他时刻的状态及观测无关, 也与时刻 t 无关.

$$P(i_t \mid i_{t-1}, o_{t-1}, \cdots, i_1, o_1) = P(i_t \mid i_{t-1}), \quad t = 1, 2, \cdots, T$$

（2）观测独立性假设, 即假设任意时刻的观测只依赖于该时刻的马尔可夫链的状态, 与其他观测及状态无关.

$$P(o_t \mid i_T, o_T, i_{T-1}, o_{T-1}, \cdots, i_{t+1}, o_{t+1}, i_t, i_{t-1}, o_{t-1}, \cdots, i_1, o_1) = P(o_t \mid i_t)$$

3.2 观测序列的生成过程

Algorithm 1: 观测序列的生成过程

Input: 隐马尔可夫模型 $\lambda = (A, B, \pi)$, 观测序列长度 T

Output: 观测序列 $O = (o_1, o_2, \cdots, o_T)$

```
按照初始状态分布 π 产生状态 i_1;
t := 1;
while t ≤ T do
    按照状态 i_t 的观测概率分布 b_i(k) 生成 o_t;
    按照状态 i_t 的状态转移概率分布 {a_{i_t i_{t+1}}} 产生状态 i_{t+1}, i_{t+1} = 1, 2, ···, N;
    t := t + 1;
end
```

3.3 隐马尔可夫模型的 3 个基本问题

1. 概率计算问题. 给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \cdots, o_T)$, 计算在模型 λ 下观测序列 O 出现的概率 $P(O \mid \lambda)$.
2. 学习问题. 已知观测序列 $O = (o_1, o_2, \cdots, o_T)$, 估计模型 $\lambda = (A, B, \pi)$ 参数, 使得在该模型下观测序列概率 $P(O \mid \lambda)$ 最大. 即用极大似然估计的方法估计参数.
3. 预测问题, 也称为解码 (decoding）问题. 已知模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \cdots, o_T)$, 求对给定观测序列条件概率 $P(I \mid O)$ 最大的状态序列 $I = (i_1, i_2, \cdots, i_T)$. 即给定观测序列, 求最有可能的对应的状态序列.

3.4 概率计算算法

3.4.1 直接计算法

状态序列 $I=(i_1,i_2,\cdots,i_T)$ 的概率是

$$P(I\mid\lambda)=\pi_{i_1}a_{i,i_2}a_{i_2i_3}\cdots a_{i_{T-1}i_T}$$

对固定的状态序列 $I=(i_1,i_2,\cdots,i_T)$, 观测序列 $O=(o_1,o_2,\cdots,o_T)$ 的概率是 $P(O\mid I,\lambda)$,

$$P(O\mid I,\lambda)=b_{i_1}(o_1)\,b_{i_2}(o_2)\cdots b_{i_T}(o_T)$$

O 和 I 同时出现的联合概率为

$$\begin{aligned}P(O,I\mid\lambda)&=P(O\mid I,\lambda)P(I\mid\lambda)\\&=\pi_{i_1}b_{i_1}(o_1)\,a_{ii_2}b_{i_2}(o_2)\cdots a_{i_{i-1}\}}b_{i_t}(o_T)\end{aligned}$$

然后, 对所有可能的状态序列 I 求和, 得到观测序列 O 的概率 $P(O\mid\lambda)$, 即

$$\begin{aligned}P(O\mid\lambda)&=\sum_{l}P(O\mid I,\lambda)P(I\mid\lambda)\\&=\sum_{i_1,i_2,\cdots,i_T}\pi_{i_1}b_{i_1}(o_1)\,a_{i,i_2}b_{i_2}(o_2)\cdots a_{i_{r-1}i_T}b_{i_T}(o_T)\end{aligned}$$

! 计算量很大, 是 $O\left(TN^T\right)$ 阶的, 这种算法不可行.

3.4.2 前向算法

Definition 3.4.1 — 前向概率. 给定隐马尔可夫模型 λ, 定义到时刻 t 部分观测序列为 $o_1,o_2,\cdots,o_t$ 且状态为 q_i 的概率为前向概率, 记作

$$\alpha_t(i)=P\left(o_1,o_2,\cdots,o_t,i_t=q_i\mid\lambda\right)$$

可以递推地求得前向概率 $\alpha_t(i)$ 及观测序列概率 $P(O\mid\lambda)$.

如图所示, 前向算法实际是基于“状态序列的路径结构”递推计算 $P(O\mid\lambda)$ 的算法. 前向算法高效的关键是其局部计算前向概率, 然后利用路径结构将前向概率“递推”到全局, 得到 $P(O\mid\lambda)$. 具体地, 在时刻 $t=1$, 计算 $\alpha_1(i)$ 的 N 个值 $(i=1,2,\cdots,N)$; 在各个时刻 $t=1,2,\cdots,T-1$, 计算 $\alpha_{t+1}(i)$ 的 N 个值 $(i=1,2,\cdots,N)$, 而且每个 $\alpha_{t+1}(i)$ 的计算利用前一时刻 N 个 $\alpha_t(j)$. 减少计算量的原因在于每一次计算直接引用前一个时刻的计算结果, 避免重复计算，这样，利用前向概率计算 $P(O\mid\lambda)$ 的计算量是 $O\left(N^2T\right)$ 阶的, 而不是直接计算的 $O\left(TN^T\right)$ 阶.

3.4.3 后向算法

Definition 3.4.2 — 后向概率. 给定隐马尔可夫模型 λ, 定义在时刻 t 状态为 q_i 的条件下, 从 $t+1$ 到 T 的部分观测序列为 $o_{t+1},o_{t+2},\cdots,o_T$ 的概率为后向概率, 记作

$$\beta_t(i)=P\left(o_{t+1},o_{t+2},\cdots,o_T\mid i_t=q_i,\lambda\right)$$

可以用递推的方法求得后向概率 $\beta_t(i)$ 及观测序列概率 $P(O\mid\lambda)$.

步骤 (1) 初始化后向概率, 对最终时刻的所有状态 q_i 规定 $\beta_T(i)=1$. 步骤 (2) 是后向概率的递推公式. 如图 10.3 所示, 为了计算在时刻 t 状态为 q_i 条件下时刻 $t+1$ 之后的观测序

Algorithm 2: 观测序列概率的前向算法

Input: 隐马尔可夫模型 λ, 观测序列 O

Output: 观测序列概率 $P(O \mid \lambda)$

1

$$\alpha_1(i)=\pi_i b_i\left(o_1\right), \quad i=1,2,\cdots,N$$

;

2 对 $t=1,2,\cdots,T-1$,

$$\alpha_{t+1}(i)=\left[\sum_{j=1}^N \alpha_t(j) a_{ji}\right] b_i\left(o_{t+1}\right), \quad i=1,2,\cdots,N$$

;

3

$$P(O \mid \lambda)=\sum_{i=1}^N \alpha_T(i)$$

;

Algorithm 3: 观测序列概率的后向算法

Input: 隐马尔可夫模型 λ, 观测序列 O

Output: 观测序列概率 $P(O \mid \lambda)$

1

$$\beta_T(i)=1, \quad i=1,2,\cdots,N$$

;

2 对 $t=T-1,T-2,\cdots,1$

$$\beta_t(i)=\sum_{j=1}^N a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j), \quad i=1,2,\cdots,N$$

;

3

$$P(O \mid \lambda)=\sum_{i=1}^N \pi_i b_i\left(o_1\right) \beta_1(i)$$

;

列为 $o_{t+1}, o_{t+2}, \cdots, o_T$ 的后向概率 $\beta_t(i)$, 只需考虑在时刻 $t+1$ 所有可能的 N 个状态 q_j 的转移概率（即 a_{ij} 项), 以及在此状态下的观测 o_{t+1} 的观测概率 (即 $b_j\left(o_{t+1}\right)$ 项), 然后考虑状态 q_j 之后的观测序列的后向概率 (即 $\beta_{t+1}(j)$ 项). 步骤（3) 求 $P(O \mid \lambda)$ 的思路与步骤（2）一致, 只是初始概率 π_i 代替转移概率.

利用前向概率和后向概率的定义可以将观测序列概率 $P(O \mid \lambda)$ 统一写成

$$P(O \mid \lambda)=\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j), t=1,2, \cdots, T-1$$

3.4.4 概率与期望值的计算

Theorem 3.4.1 给定模型 λ 和观测 O, 在时刻 t 处于状态 q_i 的概率.

记 $\gamma_t(i)=P\left(i_t=q_i \mid O, \lambda\right)$

$$\gamma_t(i)=\frac{\alpha_1(i) \beta_t(i)}{P(O \mid \lambda)}=\frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

Proof. 可以通过前向后向概率计算. 事实上,

$$\gamma_t(i)=P\left(i_t=q_i \mid O, \lambda\right)=\frac{P\left(i_t=q_i, O \mid \lambda\right)}{P(O \mid \lambda)}$$

由前向概率 $\alpha_t(i)$ 和后向概率 $\beta_t(i)$ 定义可知:

$$\alpha_t(i) \beta_t(i)=P\left(i_t=q_i, O \mid \lambda\right)$$

所以

$$\gamma_t(i)=\frac{\alpha_t(i) \beta_t(i)}{P(O \mid \lambda)}=\frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

■

Theorem 3.4.2 给定模型 λ 和观测 O, 在时刻 t 处于状态 q_i 且在时刻 $t+1$ 处于状态 q_j 的概率.

记 $\xi_t(i, j)=P\left(i_t=q_i, i_{t+1}=q_j \mid O, \lambda\right)$

$$\xi_t(i, j)=\frac{\alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{i=1}^N \alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}$$

Proof. 可以通过前向后向概率计算:

$$\xi_t(i, j)=\frac{P\left(i_t=q_i, i_{t+1}=q_j, O \mid \lambda\right)}{P(O \mid \lambda)}=\frac{P\left(i_t=q_i, i_{t+1}=q_j, O \mid \lambda\right)}{\sum_{i=1}^N \sum_{j=1}^N P\left(i_t=q_i, i_{t+1}=q_j, O \mid \lambda\right)}$$

而

$$P\left(i_t=q_i, i_{t+1}=q_j, O \mid \lambda\right)=\alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)$$

所以

$$\xi_t(i, j)=\frac{\alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{i=1}^N \alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}$$

■

将 $\gamma_t(i)$ 和 $\xi_i(i, j)$ 对各个时刻 t 求和, 可以得到一些有用的期望值;

Theorem 3.4.3 在观测 O 下状态 i 出现的期望值

$$\sum_{t=1}^{T} \gamma_t(i)$$

Theorem 3.4.4 在观测 O 下由状态 i 转移的期望值

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

Theorem 3.4.5 在观测 O 下由状态 i 转移到状态 j 的期望值

$$\sum_{t=1}^{T-1} \xi_t(i,j)$$

3.5 学习算法

3.5.1 监督学习方法

隐马尔可夫模型的学习，根据训练数据是包括观测序列和对应的状态序列还是只有观测序列，可以分别由监督学习与非监督学习实现。

假设已给训练数据包含 S 个长度相同的观测序列和对应的状态序列

$$\{(O_1, I_1), (O_2, I_2), \cdots, (O_s, I_s)\}$$

那么可以利用*极大似然估计法*来估计隐马尔可夫模型的参数. 具体方法如下.

Algorithm 4: 极大似然估计法

Input: 包含 S 个长度相同的观测序列和对应的状态序列 $\{(O_1, I_1), (O_2, I_2), \cdots, (O_s, I_s)\}$

1 转移概率 a_{ij} 的估计：设样本中时刻 t 处于状态 i 时刻 $t+1$ 转移到状态 j 的频数为 A_{ij}, 那么状态转移概率 a_{ij} 的估计是

2
$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}, \quad i = 1, 2, \cdots, N; j = 1, 2, \cdots, N$$
;

3 观测概率 $b_j(k)$ 的估计: 设样本中状态为 j 并观测为 k 的频数是 B_{jk}, 那么状态为 j 观测为 k 的概率 $b_j(k)$ 的估计是

4
$$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}, \quad j = 1, 2, \cdots, N; k = 1, 2, \cdots, M$$
;

5 初始状态概率 π_i 的估计 $\hat{\pi}_i$ 为 S 个样本中初始状态为 q_i 的频率;

3.5.2 Baum-Welch 算法

假设给定训练数据只包含 S 个长度为 T 的观测序列 $\{O_1, O_2, \cdots, O_s\}$ 而没有对应的状态序列, 目标是学习隐马尔可夫模型 $\lambda = (A, B, \pi)$ 的参数. 我们将观测序列数据看作观测数据 O, 状态序列数据看作不可观测的隐数据 I, 那么隐马尔可夫模型事实上是一个含有隐变量的概率模型

$$P(O \mid \lambda) = \sum_I P(O \mid I, \lambda) P(I \mid \lambda)$$

它的参数学习可以由 EM 算法实现。

Theorem 3.5.1 将式3.3、3.4、3.5中的各概率分别用 $\gamma_t(i), \xi_t(i,j)$ 表示, 则可将相应的公式写成:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j(k) = \frac{\sum_{t=1, o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\pi_i = \gamma_1(i)$$

其中, $\gamma_t(i)$ 由式3.4.1给出, $\xi_t(i,j)$ 由式 3.4.2 给出

Proof. 1. 确定完全数据的对数似然函数: 所有观测数据写成 $O = (o_1, o_2, \cdots, o_T)$, 所有隐数据写成 $I = (i, i_2, \cdots, i_T)$, 完全数据是 $(O, I) = (o_1, o_2, \cdots, o_T, i_1, i_2, \cdots, i_r)$. 完全数据的对数似然函数是 $\log P(O, I \mid \lambda)$

EM 算法的 E 步: 求 Q 函数 $Q(\lambda, \bar{\lambda})$

$$Q(\lambda, \bar{\lambda}) = \sum_I \log P(O, I \mid \lambda) P(O, I \mid \bar{\lambda})$$

其中, $\bar{\lambda}$ 是隐马尔可夫模型参数的当前估计值, λ 是要极大化的隐马尔可夫模型参数.

$$P(O, I \mid \lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_r}(o_T)$$

于是函数 $Q(\lambda, \bar{\lambda})$ 可以写成:

$$\begin{aligned} Q(\lambda, \bar{\lambda}) = & \sum_I \log \pi_{i_1} P(O, I \mid \bar{\lambda}) \\ & + \sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}} \right) P(O, I \mid \bar{\lambda}) + \sum_I \left(\sum_{t=1}^{T} \log b_{i_t}(o_t) \right) P(O, I \mid \bar{\lambda}) \end{aligned} \tag{3.1}$$

式中求和都是对所有训练数据的序列总长度 T 进行的.

3. EM 算法的 M 步: 极大化 Q 函数 $Q(\lambda, \bar{\lambda})$ 求模型参数 A, B, π 由于要极大化的参数在式中单独地出现在 3 个项中, 所以只需对各项分别极大化.

(1) 式 3.1 的第 1 项可以写成;

$$\sum_I \log \pi_{i_0} P(O, I \mid \bar{\lambda}) = \sum_{i=1}^{N} \log \pi_i P(O, i_1 = i \mid \bar{\lambda})$$

注意到 π_i 满足约束条件 $\sum_{i=1}^{N}\pi_i=1$, 利用拉格朗日乘子法, 写出拉格朗日函数:

$$\sum_{i=1}^{N}\log\pi_i P\left(O, i_1=i \mid \bar{\lambda}\right)+\gamma\left(\sum_{i=1}^{N}\pi_i-1\right)$$

对其求偏导数并令结果为 0

$$\frac{\partial}{\partial\pi_i}\left[\sum_{i=1}^{N}\log\pi_i P\left(O, i_1=i \mid \bar{\lambda}\right)+\gamma\left(\sum_{i=1}^{N}\pi_i-1\right)\right]=0 \tag{3.2}$$

得

$$P\left(O, i_1=i \mid \bar{\lambda}\right)+\gamma\pi_i=0$$

对 i 求和得到 γ

$$\gamma=-P(O \mid \bar{\lambda})$$

代入式 3.2 即得

$$\pi_i=\frac{P\left(O, i_1=i \mid \bar{\lambda}\right)}{P(O \mid \bar{\lambda})} \tag{3.3}$$

(2) 式 3.1 的第 2 项可以写成

$$\sum_I\left(\sum_{t=1}^{T-1}\log a_{i_t i_{t+1}}\right)P(O, I \mid \bar{\lambda})=\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}\log a_{ij}P\left(O, i_t=i, i_{t+1}=j \mid \bar{\lambda}\right)$$

类似第 1 项, 应用具有约束条件 $\sum_{j=1}^{N}a_{ij}=1$ 的拉格朗日乘子法可以求出

$$a_{ij}=\frac{\sum_{t=1}^{T-1}P\left(O, i_t=i, i_{t+1}=j \mid \bar{\lambda}\right)}{\sum_{t=1}^{T-1}P\left(O, i_t=i \mid \bar{\lambda}\right)} \tag{3.4}$$

(3) 式 3.1 的第 3 项为

$$\sum_I\left(\sum_{t=1}^{T}\log b_{i_t}\left(o_t\right)\right)P(O, I \mid \bar{\lambda})=\sum_{j=1}^{N}\sum_{t=1}^{T}\log b_j\left(o_t\right)P\left(O, i_t=j \mid \bar{\lambda}\right)$$

同样用拉格朗日乘子法, 约束条件是 $\sum_{k=1}^{M}b_j(k)=1$. 注意, 只有在 $o_t=v_k$ 时 $b_j\left(o_t\right)$ 对 $b_j(k)$ 的偏导数才不为 0 , 以 $I\left(o_t=v_k\right)$ 表示. 求得

$$b_j(k)=\frac{\sum_{t=1}^{T}P\left(O, i_t=j \mid \bar{\lambda}\right)I\left(o_t=v_k\right)}{\sum_{t=1}^{T}P\left(O, i_t=j \mid \bar{\lambda}\right)} \tag{3.5}$$

■

Algorithm 5: Baum-Welch 算法

Input: 观测数据 $O=(o_1,o_2,\cdots,o_T)$
Output: 隐马尔可夫模型 $\lambda^{(n+1)}=\left(A^{(n+1)},B^{(n+1)},\pi^{(n+1)}\right)$

1 初始化：对 $n=0$, 选取 $a_{ij}^{(0)},b_j(k)^{(0)},\pi_i^{(0)}$, 得到模型 $\lambda^{(0)}=\left(A^{(0)},B^{(0)},\pi^{(0)}\right)$;

2 递推. 对 $n=1,2,\cdots$,

$$a_{ij}^{(n+1)}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$b_j(k)^{(n+1)}=\frac{\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

3

$$\pi_i^{(n+1)}=\gamma_1(i)$$

右端各值按观测 $O=(o_1,o_2,\cdots,o_T)$ 和模型 $\lambda^{(n)}=\left(A^{(n)},B^{(n)},\pi^{(n)}\right)$ 计算. 式中 $\gamma_t(i)$, $\xi_t(i,j)$ 由式 3.4.1 和式 3.4.2 给出;

4 终止. 得到模型参数 $\lambda^{(n+1)}=\left(A^{(n+1)},B^{(n+1)},\pi^{(n+1)}\right)$

3.6 预测算法

3.6.1 近似算法

近似算法的想法是, 在每个时刻 t 选择在该时刻最有可能出现的状态 i_t^*, 从而得到一个状态序列 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$, 将它作为预测的结果.

给定隐马尔可夫模型 λ 和观测序列 O, 在时刻 t 处于状态 q_i 的概率 $\gamma_t(i)$ 是

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{P(O\mid\lambda)}=\frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

在每一时刻 t 最有可能的状态 i_t^* 是

$$i_t^*=\arg\max_{1\leqslant i\leqslant N}\left[\gamma_t(i)\right],\quad t=1,2,\cdots,T$$

从而得到状态序列 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$.

!

近似算法的优点是计算简单，其缺点是不能保证预测的状态序列整体是最有可能的状态序列，因为预测的状态序列可能有实际不发生的部分. 方法得到的状态序列中有可能存在转移概率为 0 的相邻状态, 即对某些 i,j, $a_{ij}=0$ 时. 尽管如此, 近似算法仍然是有用的.

3.6.2 维特比算法

根据动态规划的性质。

首先导入两个变量 δ 和 ψ.

Definition 3.6.1 — 在时刻 t 状态为 i 的所有单个路径 $(i_1,i_2,\cdots,i_t)$ 中概率最大值. 定义在时刻 t 状态为 i 的所有单个路径 $(i_1,i_2,\cdots,i_t)$ 中概率最大值为

$$\delta_t(i)=\max_{i_1,i_2,\cdots,i_{t-1}}P\left(i_t=i,i_{t-1},\cdots,i_1,o_t,\cdots,o_1\mid\lambda\right),\quad i=1,2,\cdots,N$$

Corollary 3.6.1 由定义可得变量 δ 的递推公式:

$$\begin{aligned}\delta_{t+1}(i) &= \max_{i_1,i_2,\cdots,i_t} P\left(i_{t+1}=i, i_t,\cdots,i_1,o_{t+1},\cdots,o_1 \mid \lambda\right) \\ &= \max_{1\leqslant j\leqslant N}\left[\delta_t(j)a_{ji}\right] b_i\left(o_{t+1}\right), \quad i=1,2,\cdots,N; t=1,2,\cdots,T-1\end{aligned}$$

Definition 3.6.2 — 在时刻 t 状态为 i 的所有单个路径 $(i_1,i_2,\cdots,i_{t-1},i)$ 中概率最大的路径的第 $t-1$ 个结点. 定义在时刻 t 状态为 i 的所有单个路径 $(i_1,i_2,\cdots,i_{t-1},i)$ 中概率最大的路径的第 $t-1$ 个结点为

$$\psi_t(i) = \arg\max_{1\leqslant j\leqslant N}\left[\delta_{t-1}(j)a_{ji}\right], \quad i=1,2,\cdots,N$$

Algorithm 6: 维特比算法

Input: 模型 $\lambda=(A,B,\pi)$ 和观测 $O=(o_1,o_2,\cdots,o_T)$; 输出: 最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$

Output: 最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$

1 初始化

$$\begin{aligned}\delta_1(i) &= \pi_i b_i\left(o_1\right), & i=1,2,\cdots,N \\ \psi_1(i) &= 0, & i=1,2,\cdots,N\end{aligned}$$

;

2 递推. 对 $t=2,3,\cdots,T$

$$\begin{aligned}\delta_t(i) &= \max_{1\leqslant j\leqslant N}\left[\delta_{t-1}(j)a_{ji}\right] b_i\left(o_t\right), & i=1,2,\cdots,N \\ \psi_t(i) &= \arg\max_{1\leqslant j\leqslant N}\left[\delta_{t-1}(j)a_{ji}\right], & i=1,2,\cdots,N\end{aligned}$$

;

3 终止

$$\begin{aligned}P^* &= \max_{1\leqslant i\leqslant N}\delta_T(i) \\ i_T^* &= \arg\max_{1\leqslant i\leqslant N}\left[\delta_T(i)\right]\end{aligned}$$

;

4 最优路径回溯. 对 $t=T-1,T-2,\cdots,1$

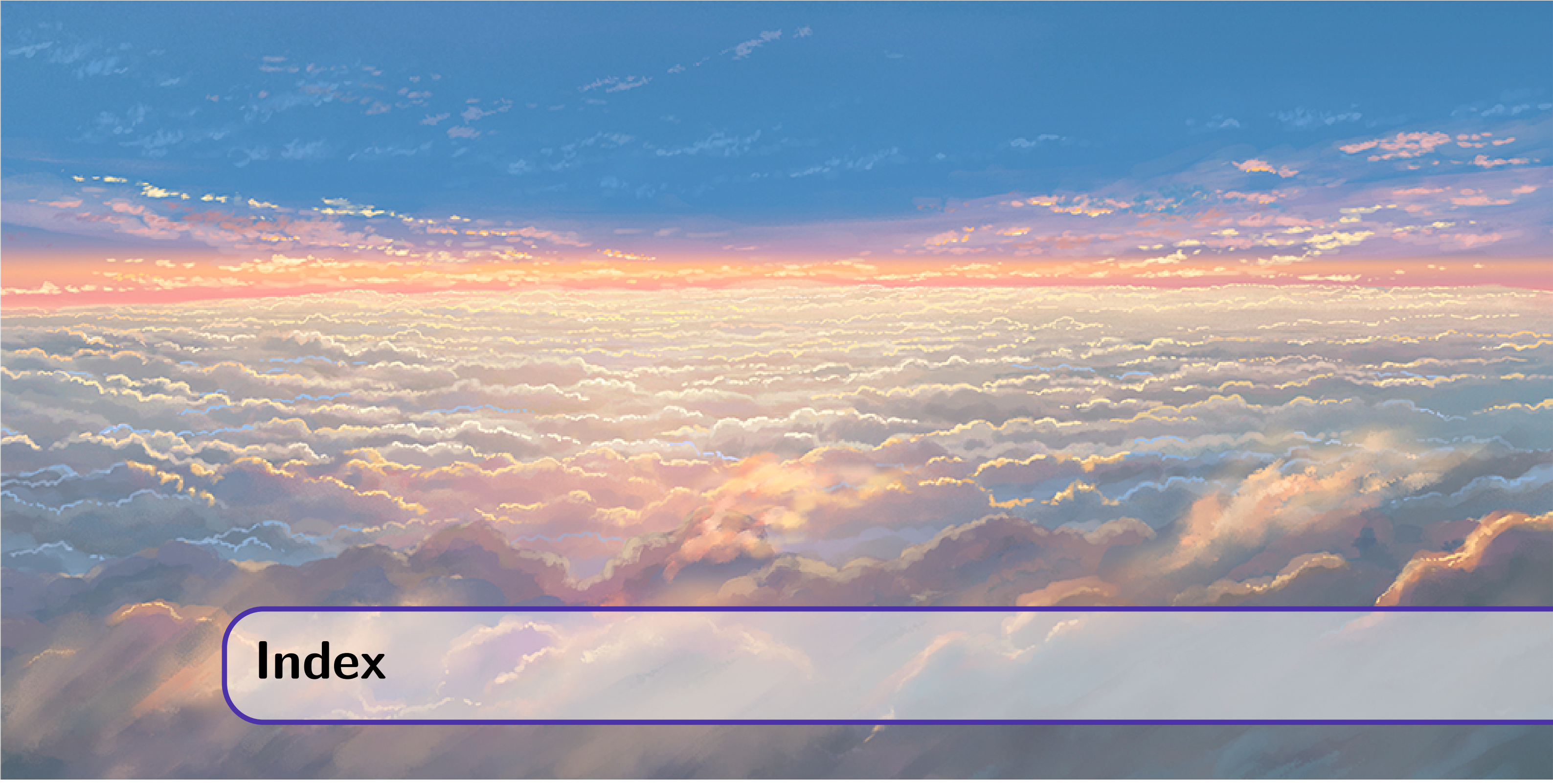
$$i_t^* = \psi_{t+1}\left(i_{t+1}^*\right)$$

求得最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$;

Bibliography

[1] George R. Jagadeesh and Thambipillai Srikanthan. "Online Map-Matching of Noisy and Sparse Location Data with Hidden Markov and Route Choice Models". In: *IEEE Transactions on Intelligent Transportation Systems* 18.9 (2017), pages 2423–2434. ISSN: 15249050. DOI: `10.1109/TITS.2017.2647967` (cited on page 11).

[2] Paul Newson and John Krumm. "Hidden Markov Map Matching Through Noise and Sparseness". In: *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), November 4-6, Seattle, WA*. 2009, pages 336–343. URL: `https://www.microsoft.com/en-us/research/publication/hidden-markov-map-matching-noise-sparseness/` (cited on pages 6, 11–13).



Index