Unsupervised Learning

Christophe Ambroise

2021-2022

Section 1

Markov and hidden Markov models

Introduction

Models for sequences of observations, $X_1, ..., X_T$, of arbitrary length T.

Such models have applications in - computational biology, - natural language processing, - time series forecasting, etc.

Focus

- Focus on the case where we the observations occur at discrete "time steps",
- "time" may also refer to locations within a sequence.

Markov models

Markov chain assumes that X_t captures all the relevant information for predicting the future (i.e., we assume it is a sufficient statistic).

If we assume discrete time steps, the joint distribution is a **Markov chain** :

$$p(X_{1:T}) = p(X_1)p(X_2|X_1)p(X_3|X_2)... = p(X1)\prod_{t=2}^{T} p(X_t|X_{t-1})$$

- X_t captures all relevant information for prediction the future
- X_t is a sufficient statistic

Homogeneous discrete Markov Chain

if $p(X_t|X_{t-1})$ is independant fo time then the chain is called homogeneous, stationary or time invariant.

We assume in the follwing that the observed variables are discrete, so $X_t \in \{1, ..., K\}$, this is called a finite-state Markov chain.

Transition matrix

When X_t is discrete $X_t \in \{1, ..., K\}$, the conditional distribution $p(X_t|X_{t-1})$ can be written as a $K \times K$ matrix, known as the transition matrix A

where $A_{ij} = p(X_t = j|X_{t-1} = i)$ is the probability of going from state i to state j. $\forall j, \sum_i A_{ij} = 1$

Example 2 states Markov Chain

$$A = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$

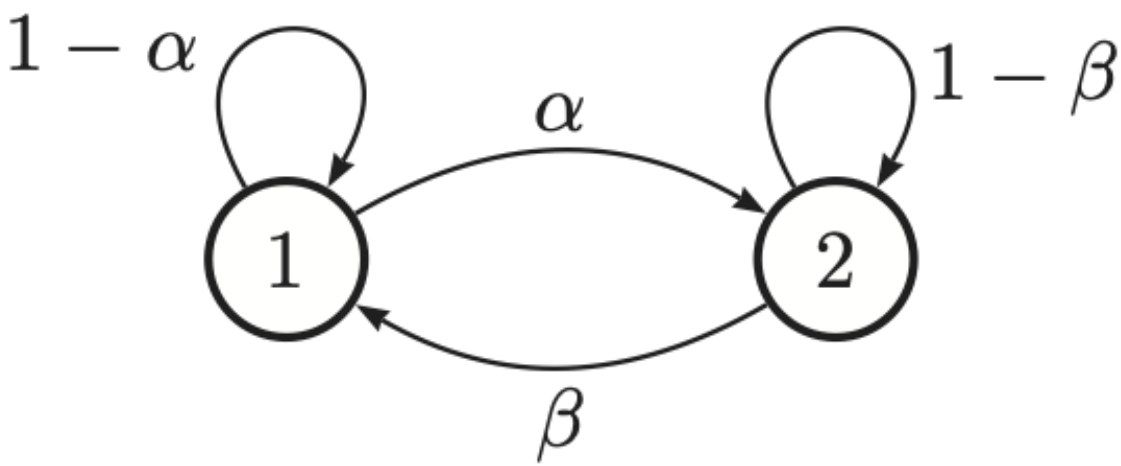


Figure 1: 2 states Markov chain

Example of DNA mutation Markov Chain

$$A = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} & & \vdots & \\ & & P(X_t = G \mid X_{t-1} = C) & \cdots \\ & & & \end{bmatrix} \end{array}$$

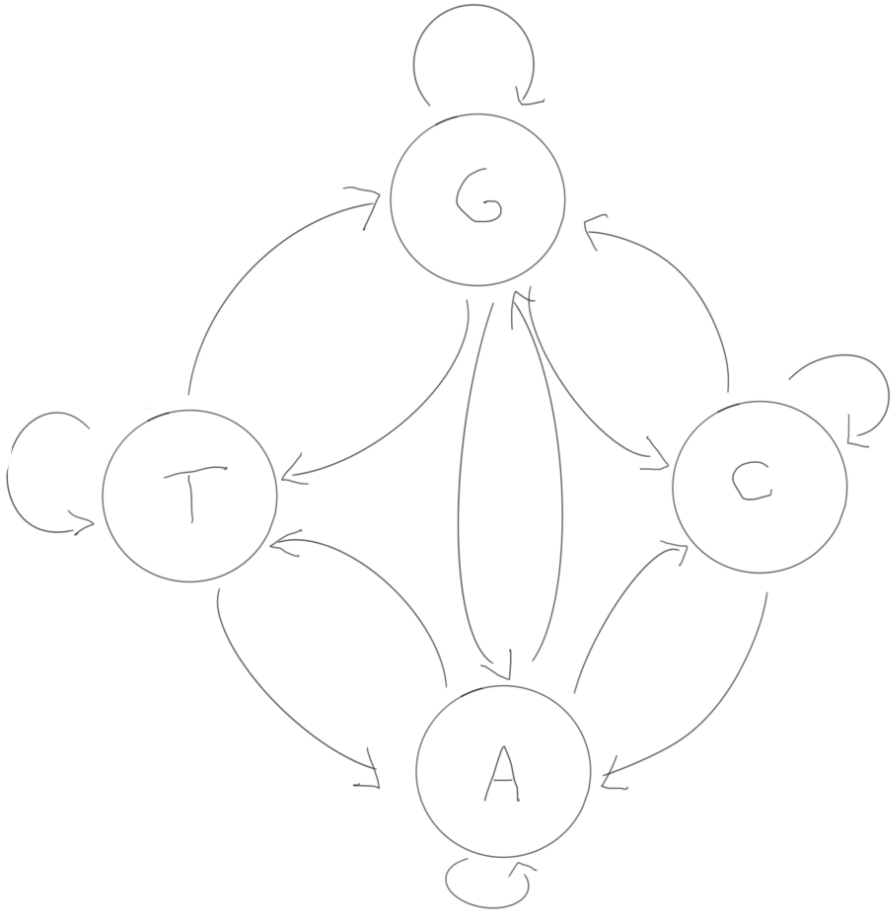


Figure 2: 4 states Markov chain

The n-step transition matrix $A(n)$

$$A_{ij} \triangleq p(X_{t+n} = j | X_t = i)$$

which is the probability of getting from i to j in exactly n steps.

$A(1) = A$

Chapman-Kolmogorov equations

$$A_{ij}(m+n) = \sum_k A_{ik}(m) A_{kj}(n)$$

We can write the above as a matrix multiplication

$$A(m+n) = A(m)A(n)$$

And thus $A(n) = A^n$

Language modeling

One important application of Markov models is to make statistical language models, which are probability distributions over sequences of words.

We define the state space to be all the words of some language.

n-gram models

- The marginal probabilities $p(X_t = k)$ are called unigram statistics.
- using first-order Markov model $p(X_t = k|X_{t-1} = j)$ is called a bigram model.
- using second-order Markov model $p(X_t = k|X_{t-1} = j, X_{t-2} = i)$ is called a trigram model.
- ...

Exercice

- Estimate unigram and bigram models for the letters $\{a,\ldots,z,-\}$ from a set of Leonard Cohen songs

Language models

can be used for several things, such as the following:

- **Sentence completion** A language model can predict the next word given the previous words in a sentence. This can be used to reduce the amount of typing required, which is particularly important for disabled users or uses of mobile devices.
- **Data compression** Any density model can be used to define an encoding scheme, by assigning short codewords to more probable strings.
- **Text classification** Any density model can be used as a class-conditional density and hence turned into a (generative) classifier.
- **Automatic essay writing** One can sample from $p(x_{1:t})$ to generate artificial text.

MLE for Markov language models

The probability of any particular sequence of length T is given by

$$\begin{aligned} p(x_{1:T}|\theta) = & \quad \pi(x_1)A(x_1, x_2)...A(x_{T-1}, x_T) \\ = & \quad \prod_j \pi_j^{\mathbb{I}_{x_1=j}} \prod_{t=2}^{T} \prod_{jk} (A_{jk})^{\mathbb{I}_{(x_t=k, x_{t-1}=j)}} \end{aligned}$$

Hence the log-likelihood of a set of N sequences $\mathcal{D} = (x_1, ..., x_N)$, where $xi = (x_{i,1}, ..., x_{i,Ti})$ is a sequence of length Ti, is given by

$$logp(\mathcal{D}|\theta) = \log p(x_i|\theta) = N_j^1 \log \pi_j + N_{jk} \log A_{jk}$$

MLE estimates are normalized Counts

$N_j^1 \triangleq \sum_{i=1}^{N} \mathbb{I}_{(x_{i1}=j)}$, $N_{jk} \triangleq \sum_{i=1}^{N} \sum_{t=1}^{Ti-1} \mathbb{I}_{(x_t=k, x_{t-1}=j)}$

$\hat{\pi}_j = \frac{N_j^1}{\sum_j N_j^1}$, $\hat{A}_{jk} = \frac{N_{jk}}{\sum_k N_{jk}}$

The problem of zero-counts

very acute whenever

- the number of states K,
- and/or the order of the chain, n, is large.

A simple solution

use add-one smoothing, where we simply add one to all the empirical counts before normalizing (Bayesian interpretation. . .)

- However add-one smoothing assumes all n-grams are equally likely, which is not very realistic.

Empirical Bayes version of deleted interpolation

Deleted interpolation (Chen and Goodman 1996)

Defines the transition matrix as a convex combination of the bigram requencies $f_{jk} = \frac{N_{jk}}{N_j}$ and the unigram frequencies $f_k = \frac{N_k}{N}$:

$$A_{jk} = (1 - \lambda) f_{jk} + \lambda f_k$$

The term λ is usually set by cross validation.

An equivalent simple hierarchical Bayesian model

Prior $A_j \sim Dir(\alpha_0 m_1, ..., \alpha_0 m_K) = Dir(\alpha_0 m) = Dir(\alpha)$

- A_j is row j of the transition matrix,
- m is the prior mean (satisfying $\sum_k m_k = 1$) and
- α_0 is the prior strength.

Posterior $A_j \sim Dir(\alpha + N_j)$ where $N_j = (N_{j1}, ..., NjK)$ is the vector that records the number of times we have transitioned out of state j to each of the other states.

Empirical Bayes version of deleted interpolation

the posterior predictive density is
$p(X_{t+1} = k|X_t = j, \mathcal{D}) = \bar{A}_{jk} = \frac{N_{jk}+\alpha_0 m_k}{\sum_k Njk+\alpha_0} = (1-\lambda_j)f_{jk} + \lambda_j m_k$ where
$\bar{A}_{jk} = E[A_{jk}|\mathcal{D}, \alpha]$ and $\lambda_j = \frac{\alpha_j}{N_j+\alpha_0}$

Remark

We have to choose α_0 and m...

Stationary distribution of a Markov chain

We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain

Distribution over states

let $\pi_t(j) = p(X_t = j)$

- Assume that π_t is a row vector.
- If we have an initial distribution over states of π_0, then at time 1 we have $\pi_1(j) = \sum_i \pi_0(i) A_{ij}$, which can be written as $\pi_1 = \pi_0 A$

Stationary distribution

If we ever reach a stage where

$$\pi = \pi A$$

then we say we have reached the stationary distribution (also called the invariant distribution or equilibrium distribution).

Computing the stationary distribution

Eigenvector associated with unit eigenvalue

To find the stationary distribution, we can just solve the eigenvector equation $A^T v = v$, and then to set $\pi = v^T$, where v is an eigenvector with eigenvalue 1.
Eigenvalues of A and A^T are the same.

A more general approach

We have K constraints from $\pi(I - A) = 0_{Kx1}$ and 1 constraint from $\pi \mathbb{1}_{Kx1} = 1$.
Since we only have K unknowns, this is overconstrained.
Let us replace any column (e.g., the last) of $I - A$ with 1, to get a new matrix, call it M. Next we define $r = [0, 0, ..., 1]$, then solve

$$\pi M = r$$

.

Exercise

Let

$$A^T = \begin{pmatrix} 0 & 1/2 & 1 \\ 1 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix}$$

be the transpose of a transition matrix. Compute the stationary distribution.

Exercise

Solution 1

```
A <- matrix(c(0, 1/2,1,1,0,0,0,1/2,0),3,3)
eigen.res<-eigen(t(A))
pi<-eigen.res$vectors[,1]
pi<-t(pi)%*%A
pi/sum(pi)
```

```
##          [,1]   [,2]   [,3]
## [1,] 0.4+0i 0.4+0i 0.2+0i
```

Solution 2

```
M<-diag(rep(1,3))-A
M[,3]<-rep(1,3)
solve(t(M),b=c(0,0,1))
```

```
## [1] 0.4 0.4 0.2
```

When does a stationary distribution exist?

Irreducible chain

A necessary condition to have a unique stationary distribution is that the state transition diagram be a singly connected component, i.e., we can get from any state to any other state. Such chains are called irreducible.

Aperiodic chain

Define the period of state i: $d(i) = gcd\{t : A_{ii}(t) > 0\}$
We say a state i is aperiodic if $d(i) = 1$. (A sufficient condition to ensure this is if state i has a self-loop, but this is not a necessary condition.) A chain is aperiodic if all its states are aperiodic.

Irreducible, aperiodic finite state Markov chain

Every irreducible (singly connected), aperiodic finite state Markov chain has a limiting distribution, which is equal to π, its unique stationary distribution. Every regular finite state chain has a unique stationary distribution, where a regular chain is one whose transition matrix satisfies $A_{ij}^n > 0$ for n and all i,j. Consequently, after n steps, the chain could be in any state, no matter

Application: Google's PageRank algorithm for web page ranking

The web is a giant directed graph, where nodes represent web pages (documents) and edges re We can get a refined search by storing the location of each word in each document.

Basic Ranking

For each word, we store a list of the documents where this word occurs.

Taking the web structure into account

But the link structure of the web provides an additional source of information.

Each incoming link is weighted by the source's authority. Thus we get the following recursive definition for the authoritativeness of page j, also called its **PageRank**:

$$\pi_j = \sum_i \pi_i A_{ij}$$

where A_{ij} is the probability of following a link from i to j. We recognize a stationary distribution of a Markov chain.

Mixture models and the EM algorithm

Latent variable models

Assume that the observed variables are correlated because they arise from a hidden common "cause". Model with hidden variables are also known as latent variable models or LVMs.

Latent variables

In general there are L latent variables, $z_{i1}, ..., z_{iL}$, and D visible variables, $x_{i1}, ..., x_{iD}$, where usually $D >> L$. If we have $L > 1$, there are many latent factors contributing to each observation, so we have a many-to-many mapping. If $L = 1$, we we only have a single latent variable; in this case, z_i is usually discrete, and we have a one-to-many mapping.

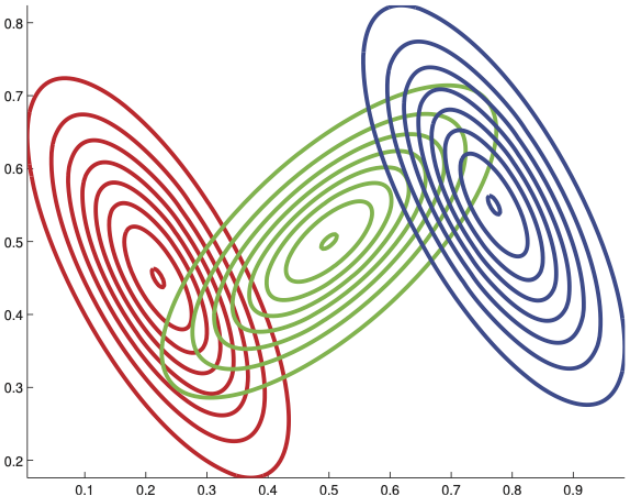
Mixture models

The simplest form of LVM is when $z_i \in \{1, ..., K\}$, representing a discrete latent state:

- $p(z_i) = Cat(\pi)$ (proportions)
- $p(x_i|z_i = k) = p_k(x_i)$ (class densities, components),

$$p(x_i|\theta) = \sum_k \pi_k p_k(x_i)$$

π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.



Mixtures of Gaussians

In this model, each base distribution in the mixture is a multivariate Gaussian with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$:

$$p(x_i|\theta) = \sum_k \pi_k \mathcal{N}(x_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Mixture of multinoullis

class- conditional density is a product of Bernoullis:

$$p(x_i|z_i = k, \theta) = \prod_j Ber(x_{ij}|\mu_{jk}).$$

EM algorithm

Data

- Observed data : $x_{1:n}$
- Missing (or hidden) data : : $z_{1:n}$

Principle

- Starting from θ^0
- At step q
  - E(xpectation) step: $Q(\theta, \theta^q) = E_{Z_{1:n}|\boldsymbol{x}_{1:n}}[\log P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta)]$
  - M(aximisation) step: $\theta^{q+1} = argmax_\theta Q(\theta, \theta^q)$

EM algorithm

At each iteration the log-likelihood of the parameters increase

$$
\begin{aligned}
Q(\theta^{q+1}, \theta^q) &\geq Q(\theta^q, \theta^q) \\
0 &\leq Q(\theta^{q+1}, \theta^q) - Q(\theta^q, \theta^q) \\
0 &\leq E_{Z_{1:n}|\boldsymbol{x}_{1:n}}[\log \frac{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^{q+1})}{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^q)}] \\
E_{Z_{1:n}|\boldsymbol{x}_{1:n}}[\log \frac{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^{q+1})}{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^q)}] &\underset{Jensen}{\leq} \log E_{Z_{1:n}|\boldsymbol{x}_{1:n}}[\frac{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^{q+1})}{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^q)}] \\
0 &\leq \log \int \frac{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^{q+1})}{P(\boldsymbol{x}_{1:n}, \boldsymbol{z}_{1:n}, \theta^q)} P(\boldsymbol{z}_{1:n}|\boldsymbol{x}_{1:n} \\
0 &\leq \log \frac{P(\boldsymbol{x}_{1:n}, \theta^{q+1})}{P(\boldsymbol{x}_{1:n}, \theta^q)}
\end{aligned}
$$

Exercice

Programmer un algorithme EM pour les mélange de Poisson univariés

Hidden Markov models

Model

1. a discrete-time, discrete-state Markov chain, with hidden states $z_t \in \{1, ..., K\}$
2. plus an observation model (emission) $p(\boldsymbol{x}_t|z_t)$

Joint distribution

$$p(z_{1:T}, x_{1:T}) = p(z_{1:T})p(x_{1:T}|z_{1:T}) = p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})\prod_{t_1}^{T} p(x_t|z_t)$$

discrete or continuous observations in HMM

Discrete

If observations are discrete, it is common for the observation model to be an observation matrix:
$p(x_t = l|z_t = k, \theta) = B(k, l)$

Continuous

If the observations are continuous, it is common for the observation model to be a conditional Gaussian:
$p(\boldsymbol{x}_t|z_t = k, \theta) = \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Applications

- Automatic speech recognition
- Activity recognition
- Gene finding
- Protein sequence alignment. x_t represents an amino acid, and z_t represents whether this matches the latent consensus sequence at this location. This model is called a profile HMM. The HMM has 3 states, called match, insert and delete.

	x	x	.	.	.	x
bat	A	G	-	-	-	C
rat	A	-	A	G	-	C
cat	A	G	-	A	A	-
gnat	-	-	A	A	A	C
goat	A	G	-	-	-	C
	1	2	.	.	.	3

Figure 4: DNA

Types of Inference in HMMs

Filtering

compute the belief state $p(z_t|x_{1:t})$ online, or recursively, as the data streams in.

Prediction

compute $p(z_{t+h}|x_{1:t})$, where $h > 0$ is called the prediction horizon.

Smoothing

compute $p(z_t|x_{1:T})$ offline, given all the evidence.

MAP estimation

computing $\arg\max_{z_{1:T}} p(z_{1:T}|x_{1:T})$, which is a most probable state sequence (**Viterbi decoding**)

EM algorithm for HMMs I

Complete log-likelihood:

$$\begin{aligned}\log P(x_{1:T}, z_{1:T}; \theta) &= \sum_k \mathbb{I}_{(z_1=k)} \log \pi_k \\ &+ \sum_{t=2}^{T} \sum_{j,k} \mathbb{I}_{(z_{t-1}=j; z_t=k)} \log A_{jk} \\ &+ \sum_{t=1}^{T} \sum_{k} \mathbb{I}_{(z_t=k)} \log \Psi_t(k)\end{aligned}$$

where $\Psi_t(k) = p(x_t | z_t = k)$

EM algorithm for HMMs II

Expectation

$$
\begin{aligned}
Q(\theta, \theta q) &= \sum_k P_{\theta_q}(z_1 = k | x_{1:T}) \log \pi_k \\
&+ \sum_{t=2}^{T} \sum_{j,k} P_{\theta_q}(z_{t-1} = j; z_t = k | x_{1:T}) \log A_{jk} \\
&+ \sum_{t=1}^{T} \sum_{k} P_{\theta_q}(z_t = k | x_{1:T}) \log \Psi_t(k)
\end{aligned}
$$

Principle

- **E-step**. Compute $P_{\theta_q}(z_t = k | x_{1:T})$ and $P_{\theta_q}(z_{t-1} = j; z_t = k | x_{1:T})$,
  - forward
  - backward equations
- **M-step**. $\theta^{q+1} = arg \max_\theta Q(\theta, \theta^q)$

The forwards-backwards algorithm

Computing $P_{\theta_q}(z_t = k|x_{1:T})$ can be achieved via the forward-backward algorithm.

The key decomposition relies on the fact that we can break the chain into two parts, the past and the future, by conditioning on z_t:

$$p(z_t = j|x_{1:T}) \propto p(z_t = j, x_{t+1:T}|x_{1:t}) \propto p(z_t = j|x_{1:t})p(x_{t+1:T}|z_t = j, x_{1:t})$$

The forward algorithm

Compute the filtered marginals, $p(z_t|x_{1:t})$ in an HMM

$$\begin{aligned}\alpha_t(j) &\triangleq p(z_t = j|x_{1:t}) = p(z_t = j|x_t, x_{1:t-1}) \\ &= \frac{p(x_t|z_t = j, x_{1:t-1})p(z_t = j|x_{1:t-1})}{p(x_t|x_{1:t-1})}\end{aligned}$$

where

$$p(z_t = j|x_{1:t-1}) = \sum_l p(z_t = j, z_{t-1} = l|x_{1:t-1}) = \sum_l p(z_t = j|z_{t-1} = l)p(z_{t-}$$

$$p(x_t|x_{1:t-1}) = \sum_k p(x_t, z_t = k|x_{1:t-1}) = \sum_k p(z_t = k|x_{1:t-1})p(x_t|z_t = k)$$

The distribution $p(z_t|x_{1:t})$ is called the (filtered) belief state at time t, and is a vector of K numbers $\boldsymbol{\alpha}_t$ where

$$\boldsymbol{\alpha}_t \propto \psi_t \odot (A^T \boldsymbol{\alpha}_{t-1})$$

The backward algorithm

Compute the conditional likelihood of future evidence given that the hidden state at time t is j.

$$\beta_t(j) \triangleq p(x_{t+1:T}|z_t = j)$$

$$\begin{aligned}
\beta_{t-1}(i) &= p(x_{t:T}|z_{t-1} = i) \\
&= \sum_j p(z_t = j, x_t, x_{t+1:T}|z_{t-1} = i) \\
&= \sum_j p(x_{t+1:T}|z_t = j, x_t, z_{t-1} = i) p(z_t = j, x_t|z_{t-1} = i) \\
&= \sum_j p(x_{t+1:T}|z_t = j, x_t, z_{t-1} = i) p(z_t = j|z_{t-1} = i) P(x_t|z_{t-1} = i
\end{aligned}$$

$$\beta_{t-1} = \boldsymbol{A}(\psi_t \odot \beta_t)$$

The smoothed posterior marginal

$$P(z_t = j|x_{1:T}) \propto p(z_t = j|x_{1:t})p(x_{t+1:T}|z_t = j)$$

$$\gamma_t(j) \propto \alpha_t(j)\beta_t(j)$$

Forward and Backward initialisation

Forward

$$\boldsymbol{\alpha}_1 \propto \boldsymbol{\pi} \odot \psi_1$$

with $\sum_k \alpha_1(k) = 1$ ### Backward

$$\beta_T(i) = p(x_{T+1:T}|z_T = i) = p(\emptyset|z_T = i) = 1$$

which is the probability of a non-event

Two-slice marginal

$$\begin{aligned}\zeta_{t,t+1}(i,j) &\triangleq p(z_t = i, z_{t+1} = j | x_{1:T}) \\ &\propto \alpha_t(i)\psi_{t+1}(j)\beta_{t+1}(j)A_{ij}\end{aligned}$$

MAP versus MPM

The (jointly) most probable sequence of states is not necessarily the same as the sequence of (marginally) most probable states.

MPM (Maximizer of the Posterior Marginals)

$$\tilde{z} = (\arg\max_{z_1} p(z_1|x_{1:T}), ..., \arg\max_{z_T} p(z_T|x_{1:T}))$$

MAP (Maximum A Posteriori)

$$z^\star = \arg\max_{z_{1:T}} p(z_{1:T}|x_{1:T})$$

MAP computation via Viterbi Algorithm I

The most probable hidden path given the data can be computed by a forward- backward recursion.

Remark

We consider the maximisation of the joint likelihood since

$$\arg\max_{z_{1:T}} p(z_{1:T}|x_{1:T}) = \arg\max_{z_{1:T}} p(z_{1:T}, x_{1:T})$$

Principle

Dynamic programming (Bellma 1953):

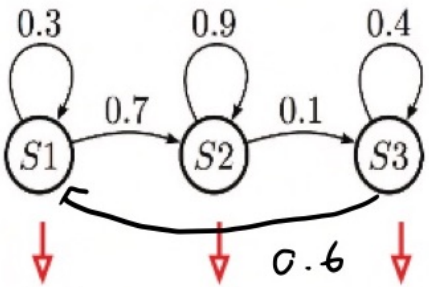
- Problem with optimal substructure
- Each subproblem is used to find the optimal solution of the main problem
- Example: shortest path, sequence alignment ...

MAP computation via Viterbi Algorithm II

3 steps

- splitting into subproblems
- solving subproblems
- reconstitution of the optimal solution

Viterbi Algorithm example



C1	0.5	0	0
C2	~~0.3~~ 0.2	0	0
C3	~~0.2~~ 0.3	0.2	0
C4	0	0.7	0.1
C5	0	0.1	0
C6	0	0	0.5
C7	0	0	0.4

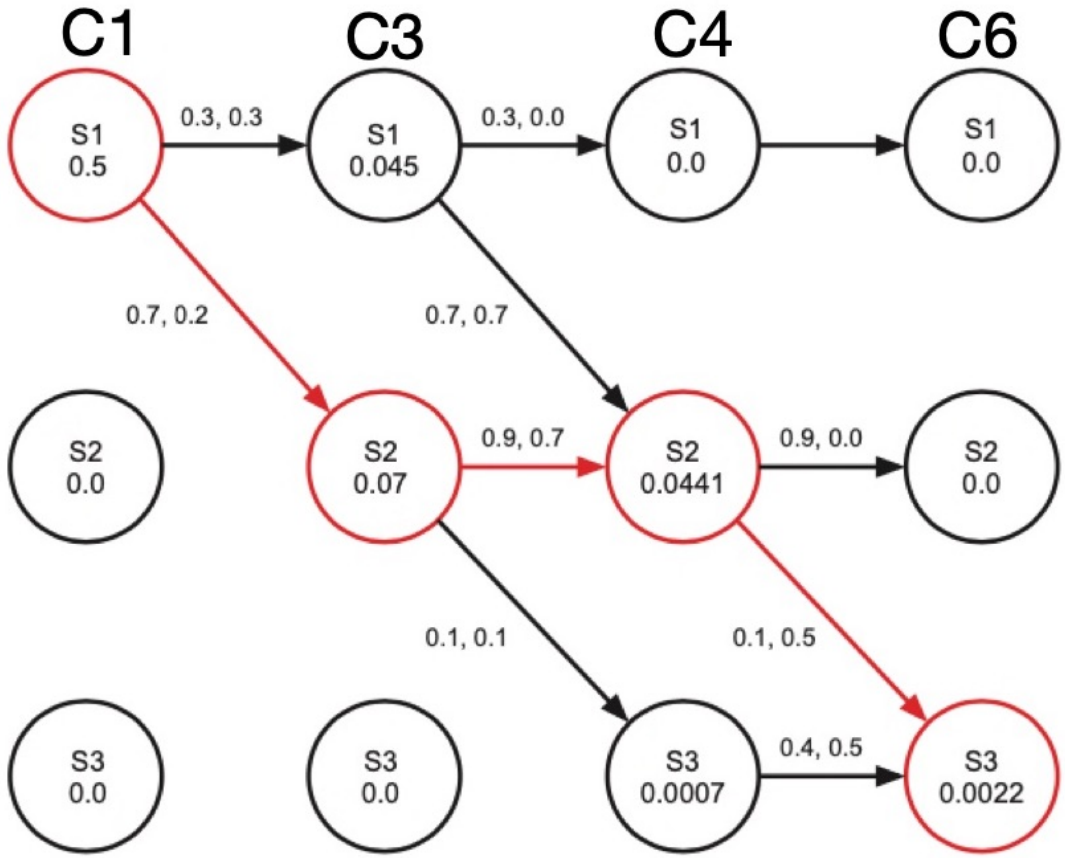


Figure 5: Viterbi

Viterbi Algorithm

Forward

$$V_{1k} = \pi_k \psi_1(k), \forall k$$

For $t \geq 2$ (optimal choices for the hidden states)

$$\begin{cases} V_{tl} & = \max_k V_{t-1,k} A_{kl} \psi_t(l) \\ S_{t-1}(l) & = \arg\max_k V_{t-1,k} A_{kl} \psi_t(l) \end{cases}$$

Backward

$$z_T^\star = \arg\max_k V_{Tk}$$

For $t \leq T$ (backtracking)

$$z_t^\star = \arg\max_k S_t(z_{t+1}^*)$$