Markov Models

Yanbing Xue

Outline

- Introduction
- Markov chains
- Dynamic belief networks
- Hidden Markov models (HMMs)

Outline

- Introduction
  - Time series
  - Probabilistic graphical models
- Markov chains
- Dynamic belief networks
- Hiddem Markov models (HMM)

What is time series?

- A time series is a sequence of data instance listed in time order.
  - In other words, data instances are totally ordered.
  - Example: weather forecasting

  - Notice: we care about the orderings rather than the exact time.

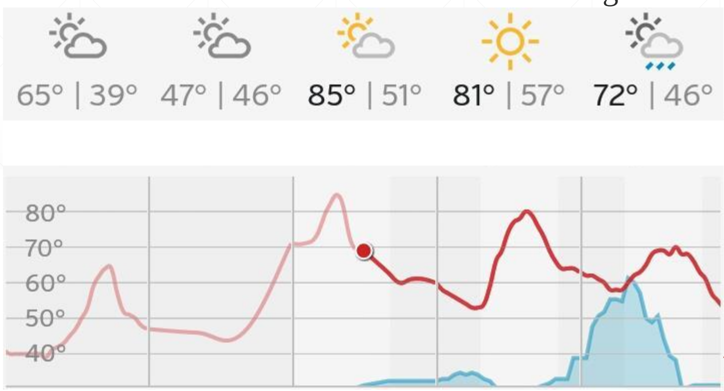
Different kinds of time series

- Two properties:
  - Time space: discrete or continuous ?
  - Task: classification or regression ?

Weather

Min/max temp

Temperature
Prob of rain



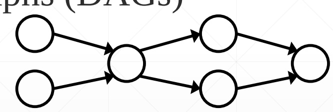
Discrete & classification

Discrete & regression

Continuous & regression

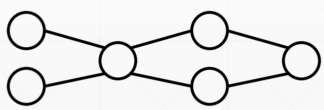
Probabilistic graphical models (PGMs)

- A PGM uses a graph-based representation to represent the conditional distributions over variables.
- Directed acyclic graphs (DAGs)



Markov model is a sub-family of PGMs on DAGs

- Undirected graph

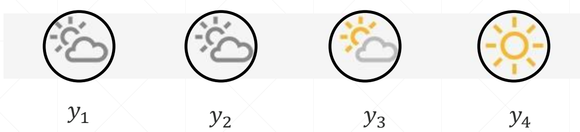


Outline

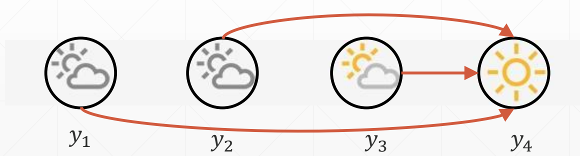
- Introduction
- Markov chains
  - Intuition
  - Inference
  - Learning
- Dynamic belief networks
- Hidden Markov models (HMMs)

Modeling time series

Assume a sequence of four weather observations: y_1, y_2, y_3, y_4

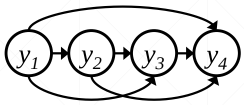


- Possible dependences: y_4 depends on the previous weather(s)



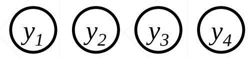
Modeling time series

In general observations: y_1, y_2, y_3, y_4 can be



Fully dependent:
E.g. y_4 depends on all previous observations

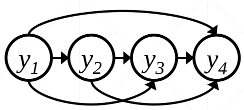
A lot of middle ground
in between the two extremes



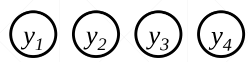
Independent:
E.g. y_4 does not depend on any previous observation

Modeling time series

- Are there intuitive and convenient dependency models?



?

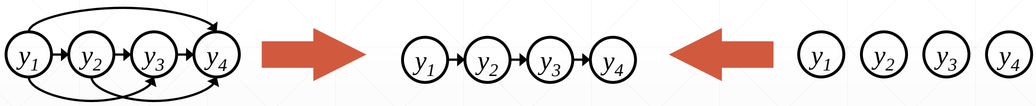


Think of the last observation $P(y_4|y_1 y_2 y_3)$
What if we have T observations?
Parameter #: exponential to # of observations

Totally drops time information

Markov chains

- **Markov assumption**: Future predictions are independent of all but the most recent observations



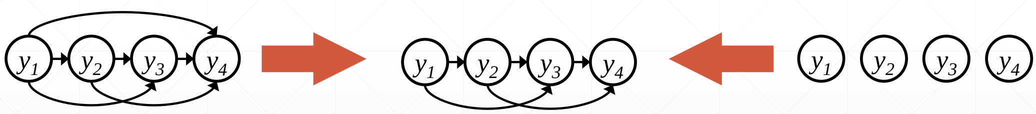
Fully dependent

First order Markov chain

Independent

Markov chains

- **Markov assumption**: Future predictions are independent of all but the most recent observations



Fully dependent

Second order Markov chain

Independent

A formal representation

- Using conditional probabilities to model y_1, y_2, y_3, y_4
- Fully dependent:
  - $P(y_1 y_2 y_3 y_4) = P(y_1)P(y_2|y_1)P(y_3|y_1 y_2)P(y_4|y_1 y_2 y_3)$
- Fully independent:
  - $P(y_1 y_2 y_3 y_4) = P(y_1)P(y_2)P(y_3)P(y_4)$
- First-order Markov chain (recent 1 observation):
  - $P(y_1 y_2 y_3 y_4) = P(y_1)P(y_2|y_1)P(y_3|y_2)P(y_4|y_3)$
- Second-order Markov chain (recent 2 observations):
  - $P(y_1 y_2 y_3 y_4) = P(y_1)P(y_2|y_1)P(y_3|y_1 y_2)P(y_4|y_2 y_3)$

A more formal representation

- Generalizes to T observations
- First-order Markov chain (recent 1 observation):
  - $P(y_1 y_2 \dots y_T) = P(y_1)\prod_{t=2}^{T} P(y_t|y_{t-1})$
- Second-order Markov chain (recent 2 observations):
  - $P(y_1 y_2 \dots y_T) = P(y_1)P(y_2|y_1)\prod_{t=3}^{T} P(y_t|y_{t-1} y_{t-2})$
- k-th order Markov chain (recent k observations):
  - $P(y_1 y_2 \dots y_T) = P(y_1)P(y_2|y_1) \dots P(y_k|y_1 \dots y_{k-1})\prod_{t=k+1}^{T} P(y_t|y_{t-k} \dots y_{t-1})$

Stationarity

- Do all states yield to the identical conditional distribution?
- $P(y_t = j | y_{t-1} = i) = P(y_{t-1} = j | y_{t-2} = i)$ for all t, i, j
- Typically holds
- A transition table A to represent conditional distribution $\begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{d1} & \cdots & A_{dd} \end{bmatrix}$
  - $A_{ij} = P(y_t = j | y_{t-1} = i)$ for all $t = 1,2,\ldots,T$
  - d: dimention of y_t
- A vector $\boldsymbol{\pi}$ to represent the initial distribution
  - $\pi_i = P(y_1 = i)$ for all $i = 1,2,\ldots,d$

Inference on a Markov chain

- Probability of a given sequence
  - $P(y_1 = i_1, \ldots, y_T = i_T) = \pi_{i_1} \prod_{t=2}^{T} A_{i_t i_{t-1}}$
- Probability of a given state
  - Forward iteration: $P(y_t = i_t) = \sum_{i_{t-1}} P(y_{t-1} = i_{t-1}) A_{i_t i_{t-1}}$
  - Can be calculated iteratively
- Both inferences are efficient
- $P(y_k = i_k, \ldots, y_T = i_T) = P(y_k = i_k) \prod_{t=k+1}^{T} A_{i_t i_{t-1}}$

Learning a Markov chain

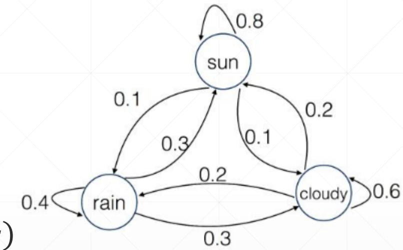
- MLE of conditional probabilities can be estimated directly.
- $A_{ij}^{MLE} = P(y_t = j | y_{t-1} = i) = \frac{P(y_t = j, y_{t-1} = i)}{P(y_{t-1} = i)} = \frac{N_{ij}}{\sum_j N_{ij}}$
  - N_{ij}: # of observations that yields $y_t = j, y_{t-1} = i$
- Bayesian parameter estimation
  - Prior: $Dir(\theta_1, \theta_2, \ldots)$
  - Posterior: $Dir(\theta_1 + N_{i1}, \theta_2 + N_{i2}, \ldots)$
  - $A_{ij}^{MAP} = \frac{N_{ij} + \theta_j - 1}{\sum_j (N_{ij} + \theta_j - 1)}$ $A_{ij}^{EV} = \frac{N_{ij} + \theta_j}{\sum_j (N_{ij} + \theta_j)}$

A toy example – weather forecast

- State 1: rainy state 2: cloudy state 3: sunny
- Given "sun-sun-sun-rain-rain-sun-cloud-sun", find A_{33}
- $A_{33}^{MLE} = \frac{N_{33}}{\sum_j N_{3j}} = \frac{2}{1+1+2}$
  - Prior: $Dir(2,2,2)$
  - Posterior: $Dir(2+1, 2+1, 2+2)$
- $A_{33}^{MAP} = \frac{N_{33} + \theta_3 - 1}{\sum_j (N_{3j} + \theta_j - 1)} = \frac{3}{7}$ $A_{33}^{EV} = \frac{N_{33} + \theta_3}{\sum_j (N_{3j} + \theta_j)} = \frac{4}{10}$

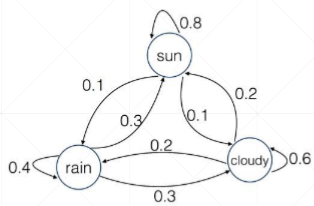
A toy example – weather forecast

- Given $A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$, day 1 is sunny
- Find the probability that day 2~8 will be "sun-sun-rain-rain-sun-cloud-sun"
- $P(y_1 y_2 \dots y_8) = P(y_1 = s)P(y_2 = s|y_1 = s)$
  $P(y_3 = s|y_2 = s)P(y_4 = r|y_3 = s)P(y_5 = r|y_4 = r)$
  $P(y_6 = s|y_5 = r)P(y_7 = c|y_6 = s)P(y_8 = s|y_7 = c)$
  $= 1 \cdot A_{33} \cdot A_{33} \cdot A_{31} \cdot A_{11} \cdot A_{13} \cdot A_{32} \cdot A_{23}$
  $= 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \cdot 0.3 \cdot 0.1 \cdot 0.2 = 1.536 \times 10^{-4}$



A toy example – weather forecast

- Given $A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$, day 1 is sunny
- Find the probability that day 3 will be sunny
- $P(y_2 = s) = \sum_i P(y_1 = i)P(y_2 = s|y_1 = i) = 0 \cdot 0.3 + 0 \cdot 0.2 + 1 \cdot 0.8 = 0.8$
  - Similarly, $P(y_2 = r) = \sum_i P(y_1 = i)P(y_2 = r|y_1 = i) = 0 \cdot 0.4 + 0 \cdot 0.2 + 1 \cdot 0.1 = 0.1$
  - $P(y_2 = c) = \sum_i P(y_1 = i)P(y_2 = c|y_1 = i) = 0 \cdot 0.3 + 0 \cdot 0.6 + 1 \cdot 0.1 = 0.1$
  - $P(y_3 = s) = \sum_i P(y_2 = i)P(y_3 = s|y_2 = i) = 0.1 \cdot 0.3 + 0.1 \cdot 0.2 + 0.8 \cdot 0.8 = 0.69$



Limitation of Markov chain

- Each state is represented by one variable
- What if each state consists of multiple variables?

Outline

- Introduction
- Markov chains
- Dynamic belief networks
  - Intuition
  - Inference
  - Learning
- Hidden Markov models (HMMs)

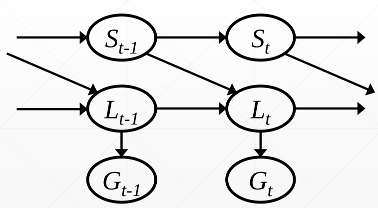
Modeling multiple variables

- What if each state consists of multiple variables?
- e.g. monitoring a robot
  - Location, GPS, Speed

  - Modeling all variables in each state jointly
- Is this a good solution?

Modeling multiple variables

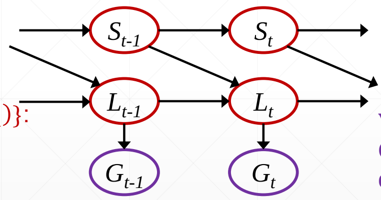
- Each variable only depends on some of the previous or current observations
- Factorization



Dynamic belief networks

- Also named as dynamic Bayesian networks

$\mathbf{X}_t = \{S_t, L_t\}$: transition states
Only dependent on previous observations
$P(\mathbf{X}_t|\mathbf{X}_{t-1}) = \{P(S_t|S_{t-1}), P(L_t|S_{t-1}L_{t-1})\}$: transition model



$\mathbf{Y}_t = \{G_t\}$: emission states / evidences
Only dependent on current observations
$P(\mathbf{Y}_t|\mathbf{X}_t) = \{P(G_t|L_t)\}$: emission model / sensor model

Inference on a dynamic BN

- Filtering: given $\mathbf{y}_{1\dots t}$, find $P(\mathbf{X}_t|\mathbf{y}_{1\dots t})$
- Exact inference
  - using Bayesian rule and the structure of dynamic BN
- $$\begin{aligned} & P(\mathbf{X}_t|\mathbf{y}_{1\dots t}) \\ \propto\ & P(\mathbf{X}_t\mathbf{y}_t|\mathbf{y}_{1\dots t-1}) \\ =\ & P(\mathbf{y}_t|\mathbf{X}_t\mathbf{y}_{1\dots t-1})P(\mathbf{X}_t|\mathbf{y}_{1\dots t-1}) \\ =\ & P(\mathbf{y}_t|\mathbf{X}_t\cancel{\mathbf{y}_{1\dots t-1}}) \sum_{\mathbf{x}_{t-1}} P(\mathbf{X}_t|\mathbf{x}_{t-1}\cancel{\mathbf{y}_{1\dots t-1}})P(\mathbf{x}_{t-1}|\mathbf{y}_{1\dots t-1}) \end{aligned}$$

Can be inferred iteratively

Structure of dynamic BN

Emission model

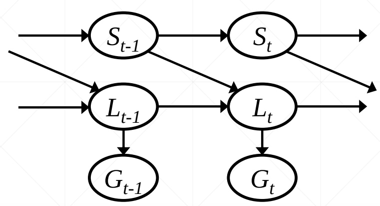
Transition model

Approximate inference on a dynamic BN

- Is exact inference useful?
- $P(\mathbf{X}_t|\mathbf{y}_{1\dots t}) = P(\mathbf{y}_t|\mathbf{X}_t)\sum_{\mathbf{x}_{t-1}} P(\mathbf{X}_t|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}|\mathbf{y}_{1\dots t-1})$
  - Needs to enumerate $\mathbf{x}_{t-1}$, exponential to # of transition variables
- Use approximate inference instead
- Particle filtering

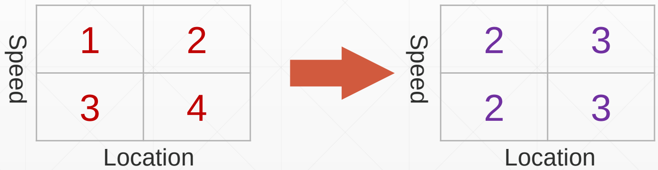
Particle filtering – a toy example

- $\mathbf{X}_t = \{S_t, L_t\}$, $\mathbf{Y}_t = \{G_t\}$
- S_t, L_t only contains 2 outcomes
  - $S_t = \{\text{fast, slow}\}$ $L_t = \{\text{left, right}\}$
- $P(\mathbf{X}_1) = P(S_1 L_1)$ a 2*2 table
- $N = 10$: # of samples in each iteration
- tth iteration = time state t



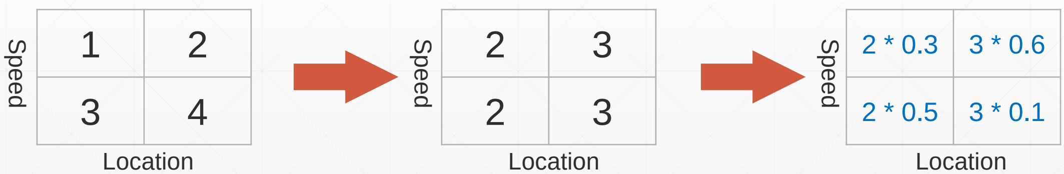
Particle filtering – a toy example

- Step 1: samples $\mathbf{a}_1 \dots \mathbf{a}_N$ from prior $P(\mathbf{X}_{t-1}|\mathbf{y}_{1\dots t-1})$
  - When $t = 1$, samples from $P(\mathbf{X}_1)$
- Step 2: update $\mathbf{a}_i$←samples from $P(\mathbf{X}_t|\mathbf{X}_{t-1} = \mathbf{a}_i)$ for all i
  - $\mathbf{a}_i$ randomly transits based on transition model



Particle filtering – a toy example

- Step 3: given $\mathbf{y}_t$ and $\mathbf{a}_i$, define $w_i = P(\mathbf{y}_t|\mathbf{X}_t = \mathbf{a}_i)$
- In step 1 of next iteration, we sample from $\mathbf{a}_1 \dots \mathbf{a}_N$ where the weight of $\mathbf{a}_i$ is w_i
  - Should be the same as sampling from $P(\mathbf{X}_t|\mathbf{y}_{1\dots t})$
  - Is this true?



Correctness of particle filtering

- Can be proved using induction
- Let $N(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1})$ denotes population of $\mathbf{x}_{t-1}$ given $\mathbf{y}_{1\ldots t-1}$
- After step 1: $\frac{N(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1})}{N} = P(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1})$
- After step 2, we have population of $\mathbf{x}_t$:
  - $N(\mathbf{x}_t|\mathbf{y}_{1\ldots t-1}) = \sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1})\, N(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1})$

Correctness of particle filtering

- After step 3, population of $\mathbf{x}_t$ is weighted by $P(\mathbf{y}_t|\mathbf{x}_t)$
- $P(\mathbf{y}_t|\mathbf{x}_t)N(\mathbf{x}_t|\mathbf{y}_{1\ldots t-1})$

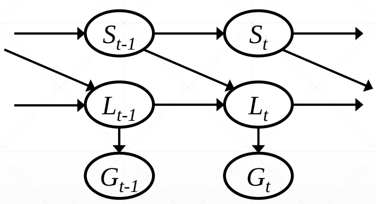
$$
\begin{aligned}
&= P(\mathbf{y}_t|\mathbf{x}_t)\sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1})\, N(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1}) \\
&= NP(\mathbf{y}_t|\mathbf{x}_t)\sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1})\, P(\mathbf{x}_{t-1}|\mathbf{y}_{1\ldots t-1}) \\
&= NP(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1\ldots t-1}) \\
&= NP(\mathbf{y}_t\mathbf{x}_t|\mathbf{y}_{1\ldots t-1}) \propto P(\mathbf{x}_t|\mathbf{y}_{1\ldots t})
\end{aligned}
$$

Learning a dynamic BN

- Given the structure of the dynamic BN…
  - Learning transition models and emission models is same as in Markov chain
- How to learn the structure?
  - For $P(\mathbf{X}_t|\mathbf{X}_{t-1})$, take each $\mathbf{X}_t^{(i)} \in \mathbf{X}_t$ as label and $\mathbf{X}_{t-1}$ as features
  - For $P(\mathbf{Y}_t|\mathbf{X}_t)$, take each $\mathbf{Y}_t^{(i)} \in \mathbf{Y}_t$ as label and $\mathbf{X}_t$ as features
- Converts to feature reduction

Limitation

- Current assumption: all states are observable, which is unrealistic



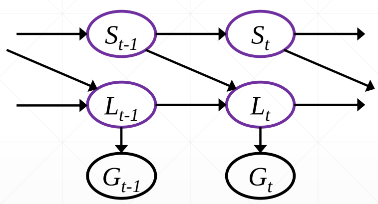
- The actual location L of the robot may never be observed
- What if some variables are hidden?

Outline

- Introduction
- Markov chains
- Dynamic belief networks
- Hidden Markov models (HMMs)
  - Intuition
  - Inference
  - Learning
  - Applications & APIs

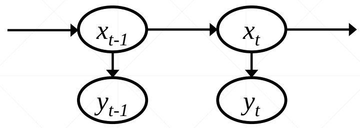
Hidden variables

- Some variables in the dynamic BN can be hidden



- Transistion variables can be hidden
- HMM: think of only one transition & one emission variable

Hidden Markov models (HMMs)



- Overview
  - A sequence of length T
  - Evidence / emission variable: $\{y_t\}$ is categorical or continuous
  - Hidden variable: $\{x_t\}$ is categorical
- $P(y_1 \dots y_T, x_1 \dots x_T) = P(x_1) \prod_{t=2}^{T} P(x_t|x_{t-1}) \prod_{t=1}^{T} P(y_t|x_t)$

Transition table

- Let d as the dimention of x_t
- Transition table A is a d*d matrix
- $A_{ij} = P(x_t = j|x_{t-1} = i)$
- Clearly, $\sum_{j=1}^{d} A_{ij} = 1$ for all i

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{d1} & \cdots & A_{dd} \end{bmatrix}$$

Emission function

- When y_t is categorical, let K as the dimension of y_t
- Emission function B can be represented as a d*K matrix

$$B = \begin{bmatrix} B_{11} & \cdots & B_{1K} \\ \vdots & \ddots & \vdots \\ B_{d1} & \cdots & B_{dK} \end{bmatrix}$$

- $B_{ij} = P(y_t = j | x_t = i)$
- Clearly, $\sum_{j=1}^{K} B_{ij} = 1$ for all i

Emission function

- When y_t is continuous, $p(y_t|x_t)$ is a PDF
  - Emission function B is the set of parameters of d different PDFs
- When $p(y_t|x_t)$ is Gaussian
- $B = \{\mu_1 \ldots \mu_d, \Sigma_1 \ldots \Sigma_d\}$

Inference on an HMM

- Given the HMM, what can we do?
- Given an observation sequence, find its probability
  - Filtering: find the distribution of the *last* hidden variable
  - Smoothing: find the distribution of the a hidden variable *in the middle*
- Given an observation sequence, find the most likely (ML) hidden variable sequence

Probability of an observed sequence

- $P(y_1 \dots y_T) = \sum_{i=1}^{d} P(y_1 \dots y_T, x_T = i)$
- Let's expand one step more:
- $P(y_1 \dots y_T, x_T = i) = \sum_{j=1}^{d} P(y_1 \dots y_T, x_T = i, x_{T-1} = j)$

$$= \sum_{j=1}^{d} P(y_1 \dots y_{T-1}, x_{T-1} = j)\, P(x_T = i | x_{T-1} = j) P(y_T | x_T = i)$$

- Can be calculated iteratively

Forward algorithm

- Let $\alpha_t(i) = P(y_1 \dots y_t, x_t = i)$
- Iteration:

$$\alpha_t(i) = \sum_{j=1}^{d} \alpha_{t-1}(j)\, A_{ji} P(y_t | x_t = i)$$

- Base: $\alpha_1(i) = P(y_1, x_1 = i) = \pi_i P(y_1 | x_1 = i)$
- Output: $\sum_{i=1}^{d} \alpha_T(i)$

Forward algorithm

- $\alpha_t(i) = \sum_{j=1}^{d} \alpha_{t-1}(j)\, A_{ji} P(y_t | x_t = i)$
  - $\alpha_{t-1}(j) = P(y_1 \dots y_{t-1}, x_{t-1} = j)$
    - $\Downarrow$ integrating x_t
  - $\alpha_{t-1}(j) A_{ji} = P(y_1 \dots y_{t-1}, x_{t-1} = j, x_t = i)$
    - $\Downarrow$ integrating y_t
  - $\alpha_{t-1}(j) A_{ji} P(y_t | x_t = i) = P(y_1 \dots y_t, x_{t-1} = j, x_t = i)$
    - $\Downarrow$ sum x_{t-1} out
  - $\alpha_t(i) = \sum_{j=1}^{d} \alpha_{t-1}(j) A_{ji} P(y_t | x_t = i) = P(y_1 \dots y_t, x_t = i)$

$\uparrow$

Backward algorithm

- Iterates reversely
- Let $\beta_t(i) = P(y_{t+1} \dots y_T | x_t = i)$
- Iteration:

$$\beta_t(i) = \sum_{j=1}^{d} \beta_{t+1}(j)\, A_{ij} P(y_{t+1} | x_{t+1} = i)$$

- Base: $\beta_T(i) = 1$
- Output: $\sum_{i=1}^{d} \pi_i P(y_1 | x_1 = i) \beta_1(i)$

Filtering and smoothing

- Filtering: find $P(x_T = i | y_1 \dots y_T)$
- $P(x_T = i | y_1 \dots y_T) \propto P(y_1 \dots y_T, x_T = i) = \alpha_t(i)$
  - Directly applies forward algorithm
- Smoothing: find $P(x_t = i | y_1 \dots y_T)$ where $t < T$
- $P(x_t = i | y_1 \dots y_T) \propto P(y_1 \dots y_T, x_t = i)$
  $= P(y_1 \dots y_t, x_t = i) P(y_{t+1} \dots y_T | x_t = i) = \alpha_t(i) \beta_t(i)$
  - Using both forward and backward algorithm

Viterbi algorithm

- Find $\underset{x_1 \ldots x_T}{\operatorname{argmax}} P(x_1 \ldots x_T | y_1 \ldots y_T)$
- $\underset{x_1 \ldots x_T}{\operatorname{argmax}} P(x_1 \ldots x_T | y_1 \ldots y_T) = \underset{x_1 \ldots x_T}{\operatorname{argmax}} P(y_1 \ldots y_T, x_1 \ldots x_T)$
- Let $\delta_t(i) = \max_{x_1 \ldots x_{t-1}} P(y_1 \ldots y_t, x_1 \ldots x_{t-1}, x_t = i)$
  - Represents the highest probability of a hidden variable sequence $x_1 \ldots x_t$ ending with $x_t = i$
- Iteration: $\delta_t(i) = P(y_t | x_t = i) \max_j [\delta_{t-1}(j) A_{ji}]$
  - A_{ji} and $P(y_t | x_t = i)$ are independent of $y_1 \ldots y_{t-1}, x_1 \ldots x_{t-2}$
- Base: $\delta_1(i) = P(y_1, x_1 = i) = \pi_i P(y_1 | x_1 = i)$

Correctness of Viterbi

- Can be proved using induction
- $\delta_{t-1}(j) = \max_{x_1 \ldots x_{t-2}} P(y_1 \ldots y_{t-1}, x_1 \ldots x_{t-2}, x_{t-1} = j)$
- $\delta_t(i) = P(y_t | x_t = i) \max_j [\delta_{t-1}(j) A_{ji}]$
  $= P(y_t | x_t = i) \max_j \left[\max_{x_1 \ldots x_{t-2}} P(y_1 \ldots y_{t-1}, x_1 \ldots x_{t-2}, x_{t-1} = j) P(x_t = i | x_{t-1} = j) \right]$
  $= P(y_t | x_t = i) \max_{x_1 \ldots x_{t-1}} P(y_1 \ldots y_{t-1}, x_1 \ldots x_{t-2}, x_{t-1}, x_t = i)$
  $= \max_{x_1 \ldots x_{t-1}} P(y_1 \ldots y_t, x_1 \ldots x_{t-2}, x_{t-1}, x_t = i)$

Learning an HMM

- Given $y_1 \dots y_T$, find the MLE of $\boldsymbol{\pi}, A, B$
- Some notations (for simplicity):
  - $\mathbf{x} = \{x_1 \dots x_t\}$ $\mathbf{y} = \{y_1 \dots y_T\}$
  - x_{ti}: binary variable, 1 if $x_t = i$ and 0 otherwise
  - $\gamma(x_{ti}) = P(x_t = i|\mathbf{y})$
  - $\eta\left(x_{t-1,j}x_{ti}\right) = P(x_{t-1} = j, x_t = i|\mathbf{y})$
- Using Baum-Welch algorithm (EM)

Q function

$$\max_{\boldsymbol{\pi},A,B} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \log P(\mathbf{y}, \mathbf{x})$$

- $\sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) \log P(\mathbf{y}, \mathbf{x}) = \sum_{\mathbf{x}} P(\mathbf{x}|\mathbf{y})[\log P(x_1) + \sum_{t=2}^{T} P(x_t|x_{t-1}) + \sum_{t=1}^{T} P(y_t|x_t)]$

$$= \sum_{x_1} P(x_1|\mathbf{y}) \log P(x_1) + \sum_{t=2}^{T} \sum_{x_{t-1}x_t} P(x_{t-1}x_t|\mathbf{y}) \log P(x_t|x_{t-1}) + \sum_{t=1}^{T} \sum_{x_t} P(x_t|\mathbf{y}) \log P(y_t|x_t)$$

$$= \sum_{k=1}^{d} \gamma(x_{1k}) \log \pi_k + \sum_{t=2}^{T} \sum_{j=1}^{d} \sum_{k=1}^{d} \eta\left(x_{t-1,j}x_{tk}\right) \log A_{jk} + \sum_{t=1}^{T} \sum_{k=1}^{d} \gamma(x_{tk}) \log P(y_t|x_t = k)$$

M-step

- $$\sum_{k=1}^{d} \gamma(x_{1k}) \log \pi_k + \sum_{t=2}^{T} \sum_{j=1}^{d} \sum_{k=1}^{d} \eta(x_{t-1,j} x_{tk}) \log A_{jk} + \sum_{t=1}^{T} \sum_{k=1}^{d} \gamma(x_{tk}) \log P(y_t | x_t = k)$$
- We can maximize Q regarding $\boldsymbol{\pi}, A, B$ separately
- Can be achieved using Lagrange multipliers

Maximize Q regarding π

- For $\boldsymbol{\pi} = \{\pi_1 \dots \pi_d\}$, we always have $\sum_{k=1}^{d} \pi_k = 1$
- We incorporate such constraint, and set the derivative as 0:

$$\frac{\partial}{\partial \pi_k}\left[\sum_{k=1}^{d} \gamma(x_{1k}) \log \pi_k + \varphi\left(\sum_{k=1}^{d} \pi_k - 1\right)\right] = \frac{\gamma(x_{1k})}{\pi_k} + \varphi = 0$$

- In other words, $\gamma(x_{1k}) + \varphi \pi_k = 0$ holds for all k. Their sum is also 0

$$\sum_{k=1}^{d} \gamma(x_{1k}) + \varphi \sum_{k=1}^{d} \pi_k = \sum_{k=1}^{d} \gamma(x_{1k}) + \varphi = 0$$

- Take φ back to the derivative for each π_k, we obtain $\pi_k = \frac{\gamma(x_{1k})}{\sum_{j=1}^{d} \gamma(x_{1j})}$

Maximize Q regarding A, B

- Using similar technique, A and B can also be optimized
- $A_{jk} = \frac{\sum_{t=2}^{T} \eta(x_{t-1,j} x_{tk})}{\sum_{l=1}^{d} \sum_{t=2}^{T} \eta(x_{t-1,j} x_{tl})}$
- When y_t is categorical:
- $P(y_t | x_t = k) = \prod_{i=1}^{K} \mu_{ik}{}^{y_{ti} x_{tk}}$ where $\mu_{ik} = \frac{\sum_{t=1}^{T} \gamma(x_{tk}) y_{ti}}{\sum_{t=1}^{T} \gamma(x_{tk})}$
- When y_t is continuous: $P(y_t | x_t = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$
- $\mu_k = \frac{\sum_{t=1}^{T} \gamma(x_{tk}) y_t}{\sum_{t=1}^{T} \gamma(x_{tk})}$ $\Sigma_k = \frac{\sum_{t=1}^{T} \gamma(x_{tk})(y_t - \mu_k)(y_t - \mu_k)^T}{\sum_{t=1}^{T} \gamma(x_{tk})}$

E-step

- Compute $\gamma(x_{tk})$ and $\eta(x_{t-1,j} x_{tk})$ for all t,j,k
- Remember:
  - $\gamma(x_{tk}) = P(x_t = k | \mathbf{y})$
  - $\eta(x_{t-1,j} x_{tk}) = P(x_{t-1} = j, x_t = k | \mathbf{y})$ Similar to smoothing!
- $\gamma(x_{tk}) \propto P(x_t = k, \mathbf{y}) = \alpha_t(k) \beta_t(k)$
- $\eta(x_{t-1,j} x_{tk}) \propto P(x_{t-1} = j, x_t = k, \mathbf{y}) = \alpha_{t-1}(j) \beta_t(k) A_{jk} P(y_t | x_t = k)$

Applications

- Speech recognition
- Natural language processing

- Bio-sequence analysis

APIs

- Python: hmmlearn (compatible with scikit-learn)
  - https://github.com/hmmlearn/hmmlearn (or *pip install hmmlearn*)
- Matlab (integrated)
  - https://www.mathworks.com/help/stats/hidden-markov-models-hmm.html
- C++: HTK3
  - http://htk.eng.cam.ac.uk/

Thank You!

Markov models