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
- Hidden 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**
  - 65°, 39°, 47°, 46°, 85°, 51°, 81°, 57°, 72°, 46°

- **Prob of rain**

### Temperature

- 80°, 70°, 60°, 50°, 40°

### Probabilistic graphical models (PGMs)

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

Markov model is a sub-family of PGMs on DAGs
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

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

![Diagram of Markov chains]

- **First order Markov chain**: Fully dependent → Independent
- **Second order Markov chain**: Fully dependent → 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):
  - $P(y_1, y_2, y_3, y_4) = P(y_1)P(y_2|y_1)P(y_3|y_1, y_2)P(y_4|y_3)$
- Second-order Markov chain (recent 2 observations):
  - $P(y_1, y_2, y_3, y_4) = P(y_1)P(y_2|y_1)P(y_3|y_1, y_2)P(y_4|y_2y_3)$

A more formal representation

- Generalizes to T observations
- First-order Markov chain (recent 1 observation):
  - $P(y_1, y_2 \ldots y_T) = P(y_1) \prod_{t=2}^{T} P(y_t|y_{t-1})$
- Second-order Markov chain (recent 2 observations):
  - $P(y_1, y_2 \ldots y_T) = P(y_1)P(y_2|y_1) \prod_{t=3}^{T} P(y_t|y_{t-1}y_{t-2})$
- k-th order Markov chain (recent k observations):
  - $P(y_1, y_2 \ldots y_T) = P(y_1)P(y_2|y_1) \ldots P(y_k|y_1 \ldots y_{k-1}) \prod_{t=k+1}^{T} P(y_t|y_{t-k} \ldots y_{t-1})$
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
    - $A_{ij} = P(y_t = j | y_{t-1} = i)$ for all $t = 1, 2, ..., T$
    - $d$: dimension of $y_t$
  - A vector $\pi$ to represent the initial distribution
    - $\pi_i = P(y_1 = i)$ for all $i = 1, 2, ..., d$

Inference on a Markov chain

- Probability of a given sequence
  - $P(y_1 = i_1, ..., y_T = i_T) = \pi_{i_1} \prod_{t=2}^{T} A_{i_t i_{t-1}}$
- 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
  - $P(y_k = i_k, ..., y_T = i_T) = P(y_k = i_k) \prod_{t=K+1}^{T} A_{i_t i_{t-1}}$
Learning a Markov chain

- MLE of conditional probabilities can be estimated directly.
  
  \[
  A_{ij}^{MLE} = \frac{N_{ij}}{\sum_j N_{ij}} = \frac{P(y_t=j|y_{t-1}=i)}{p(y_{t-1}=i)}
  \]

  - \(N_{ij}\): # of observations that yields \(y_t = j, y_{t-1} = i\)

- Bayesian parameter estimation
  
  - Prior: \(\text{Dir}(\theta_1, \theta_2, \ldots)\)
  
  - Posterior: \(\text{Dir}(\theta_1 + N_{i1}, \theta_2 + N_{i2}, \ldots)\)

  \[
  A_{ij}^{MAP} = \frac{N_{ij} + \theta_j - 1}{\sum_j(N_{ij} + \theta_j - 1)}
  \]

  \[
  A_{ij}^{EV} = \frac{N_{ij} + \theta_j}{\sum_j(N_{ij} + \theta_j)}
  \]

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: \(\text{Dir}(2, 2, 2)\)
  
  - Posterior: \(\text{Dir}(2 + 1, 2 + 1, 2 + 2)\)

  \[
  A_{33}^{MAP} = \frac{N_{33} + \theta_3 - 1}{\sum_j(N_{3j} + \theta_j - 1)} = \frac{3}{7}
  \]

  \[
  A_{33}^{EV} = \frac{N_{33} + \theta_3}{\sum_j(N_{3j} + \theta_j)} = \frac{4}{10}
  \]
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”

$P(y_1 y_2 \ldots y_8) = P(y_1 = s)P(y_2 = s | y_1 = s) \cdot P(y_3 = r)P(y_4 = s | y_3 = r) \cdot P(y_5 = r)P(y_6 = c | y_5 = r) \cdot P(y_7 = s | y_6 = c) \cdot P(y_8 = s | y_7 = c)$

$= 1 \cdot A_{33} \cdot A_{33} \cdot A_{31} \cdot A_{11} \cdot A_{13} \cdot A_{32} \cdot A_{23}$

$= 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \cdot 0.3 \cdot 0.1 \cdot 0.2 = 1.536 \times 10^{-4}$
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

\[ X_t = \{S_t, L_t\} : \text{transition states} \]

Only dependent on previous observations

\[ P(X_t | X_{t-1}) = \{P(S_t | S_{t-1}), P(L_t | S_{t-1}, L_{t-1})\} : \text{transition model} \]

\[ Y_t = \{G_t\} : \text{emission states / evidences} \]

Only dependent on current observations

\[ P(Y_t | X_t) = \{P(G_t | L_t)\} : \text{emission model / sensor model} \]

---

Inference on a dynamic BN

- Filtering: given \( y_{1...t} \), find \( P(X_t | y_{1...t}) \)

- Exact inference
  - using Bayesian rule and the structure of dynamic BN

\[
P(X_t | y_{1...t}) \propto P(X_t y_t | y_{1...t-1})
= P(y_t | X_t y_{1...t-1})P(X_t | y_{1...t-1})
= P(y_t | X_t y_{1...t-1}) \sum_{x_{t-1}} P(X_t | x_{t-1} y_{1...t-1})P(x_{t-1} | y_{1...t-1})
\]

Structure of dynamic BN

Emission model

Transition model

Can be inferred iteratively
Approximate inference on a dynamic BN

- Is exact inference useful?

\[ P(X_t | y_{1...t}) = P(y_t | X_t) \sum_{x_{t-1}} P(X_t | x_{t-1}) P(x_{t-1} | y_{1...t-1}) \]

- Needs to enumerate \( x_{t-1} \), exponential to \# of transition variables
- Use approximate inference instead
- Particle filtering

Particle filtering – a toy example

- \( X_t = \{S_t, L_t\} \), \( Y_t = \{G_t\} \)
- \( S_t, L_t \) only contains 2 outcomes
  - \( S_t = \{\text{fast, slow}\} \)
  - \( L_t = \{\text{left, right}\} \)
- \( P(X_1) = P(S_1 L_1) \) a 2*2 table
- \( N = 10 \): \# of samples in each iteration
- \( t \)th iteration = time state \( t \)
Particle filtering – a toy example

- **Step 1**: samples $a_1 \ldots a_N$ from prior $P(X_{t-1} | y_{1 \ldots t-1})$
  - When $t = 1$, samples from $P(X_1)$
- **Step 2**: update $a_i \leftarrow$ samples from $P(X_t | X_{t-1} = a_i)$ for all $i$
  - $a_i$ randomly transits based on transition model

<table>
<thead>
<tr>
<th>Speed</th>
<th>Location</th>
<th>Speed</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Step 3**: given $y_t$ and $a_i$, define $w_i = P(y_t | X_t = a_i)$
- In step 1 of next iteration, we sample from $a_1 \ldots a_N$ where the weight of $a_i$ is $w_i$
  - Should be the same as sampling from $P(X_t | y_{1 \ldots t})$
  - Is this true?
Correctness of particle filtering

- Can be proved using induction
- Let $N(x_{t-1} | y_{1...t-1})$ denotes population of $x_{t-1}$ given $y_{1...t-1}$
- After step 1: $\frac{N(x_{t-1} | y_{1...t-1})}{N} = P(x_{t-1} | y_{1...t-1})$
- After step 2, we have population of $x_t$:
  - $N(x_t | y_{1...t-1}) = \sum_{x_{t-1}} P(x_t | x_{t-1}) N(x_{t-1} | y_{1...t-1})$

---

Correctness of particle filtering

- After step 3, population of $x_t$ is weighted by $P(y_t | x_t)$
- $P(y_t | x_t) N(x_t | y_{1...t-1})$
  - $= P(y_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) N(x_{t-1} | y_{1...t-1})$
  - $= N P(y_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1} | y_{1...t-1})$
  - $= N P(y_t | x_t) P(x_t | y_{1...t-1})$
  - $= N P(y_t x_t | y_{1...t-1}) \propto P(x_t | y_{1...t})$
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 \( P(X_t | X_{t-1}) \), take each \( X^{(i)}_t \) as label and \( X_{t-1} \) as features
    - For \( P(Y_t | X_t) \), take each \( Y^{(i)}_t \) as label and \( X_t \) 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

- Transition 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

$$P(y_1 \ldots y_T, x_1 \ldots x_T) = P(x_1) \prod_{t=2}^T P(x_t | x_{t-1}) \prod_{t=1}^T P(y_t | x_t)$$

---

Transition table

- Let $d$ as the dimension of $x_t$
- Transition table $A$ is a $d \times d$ matrix
  $$A = \begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{d1} & \cdots & A_{dd} \end{bmatrix}$$
- $A_{ij} = P(x_t = j | x_{t-1} = i)$
- Clearly, $\sum_{j=1}^d A_{ij} = 1$ for all $i$
**Emission function**

- When \( y_t \) is categorical, let \( K \) as the dimension of \( y_t \)
- Emission function \( B \) can be represented as a \( d \times K \) 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

- \( P(y_1 \ldots y_T) = \sum_{i=1}^{d} P(y_1 \ldots y_T, x_T = i) \)
- Let’s expand one step more:
  - \( P(y_1 \ldots y_T, x_T = i) = \sum_{j=1}^{d} P(y_1 \ldots y_T, x_T = i, x_{T-1} = j) \)
  - \( = \sum_{j=1}^{d} P(y_1 \ldots y_{T-1}, x_{T-1} = j) \cdot P(x_T = i | x_{T-1} = j) \cdot P(y_T | x_T = i) \)
- Can be calculated iteratively
Forward algorithm

- Let $\alpha_t(i) = P(y_1 \ldots y_t, x_t = i)$
- 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:
  \[
  \sum_{i=1}^{d} \alpha_T(i)
  \]
**Backward algorithm**

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

- Find \( \text{argmax} P(x_1 \ldots x_T | y_1 \ldots y_T) \)

\[
\text{argmax} P(x_1 \ldots x_T | y_1 \ldots y_T) = \text{argmax} P(y_1 \ldots y_T, x_1 \ldots x_T)
\]

- 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: \( \delta_1(i) = P(y_1, x_1 = i) = \pi_i P(y_1 | x_1 = i) \)

Learning an HMM

- Given \( y_1 \ldots y_T \), find the MLE of \( \pi, A, B \)

- Some notations (for simplicity):
  - \( x = \{x_1 \ldots x_t\} \quad y = \{y_1 \ldots y_T\} \)
  - \( x_t \): binary variable, 1 if \( x_t = i \) and 0 otherwise
  - \( \gamma(x_t) = P(x_t = i | y) \)
  - \( \eta(x_{t-1}, x_t) = P(x_{t-1} = j, x_t = i | y) \)

- Using Baum-Welch algorithm (EM)
Q function

\[
\max_{\pi, A, B} \mathbb{E}_{x|y} \log P(y, x)
\]

- \[ \sum_x P(x|y) \log P(y, x) = \sum_x P(x|y) [\log P(x_1) + \sum_{t=2}^{T} P(x_t|x_{t-1}) + \sum_{t=1}^{T} P(y_t|x_t)] \]
- \[ = \sum_{x_1} P(x_1|y) \log P(x_1) + \sum_{t=2}^{T} \sum_{x_{t-1}x_t} P(x_{t-1}x_t|y) \log P(x_t|x_{t-1}) + \sum_{t=1}^{T} \sum_{x_t} P(x_t|y) \log P(y_t|x_t) \]
- \[ = \sum_{k=1}^{d} \gamma(x_{1k}) \log \pi_k + \sum_{t=2}^{T} \sum_{j=1}^{d} \sum_{k=1}^{d} \eta(x_{t-1,j}x_{tk}) \log A_{jk} + \sum_{t=1}^{T} \sum_{k=1}^{d} \gamma(x_{tk}) \log P(y_t|x_t = k) \]

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 \( \pi, A, B \) separately
- Can be achieved using Lagrange multipliers
Maximize Q regarding $\pi$

- For $\pi = \{\pi_1, \ldots, \pi_d\}$, we always have $\sum \pi_k = 1$
- We incorporate such constraint, and set the derivative as 0:
  $$
  \frac{\partial}{\partial \pi_k} \left[ \sum_{k=1}^{d} y(x_{1k}) \log \pi_k + \varphi \left( \sum_{k=1}^{d} \pi_k - 1 \right) \right] = \frac{y(x_{1k})}{\pi_k} + \varphi = 0
  $$
- In other words, $y(x_{1k}) + \varphi \pi_k = 0$ holds for all $k$. Their sum is also 0
  $$
  \sum_{k=1}^{d} y(x_{1k}) + \varphi \sum_{k=1}^{d} \pi_k = \sum_{k=1}^{d} y(x_{1k}) + \varphi = 0
  $$
- Take $\varphi$ back to the derivative for each $\pi_k$, we obtain $\pi_k = \frac{y(x_{1k})}{\sum_{j=1}^{d} y(x_{1j})}$

Maximize Q regarding $A, B$

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

- Compute \( \gamma(x_{tk}) \) and \( \eta(x_{t-1,j|x_{tk}}) \) for all \( t,j,k \)
- Remember:
  - \( \gamma(x_{tk}) = P(x_t = k | y) \)
  - \( \eta(x_{t-1,j|x_{tk}}) = P(x_{t-1} = j, x_t = k | y) \) Similar to smoothing!

\[
\gamma(x_{tk}) \propto P(x_t = k, y) = \alpha_t(k)\beta_t(k)
\]
\[
\eta(x_{t-1,j|x_{tk}}) \propto P(x_{t-1} = j, x_t = k, y) = \alpha_{t-1}(j)\beta_t(k)A_{jk}P(y_t|x_t = k)
\]

Applications

- Speech recognition
- Natural language processing
- Bio-sequence analysis
APIs

- Python: hmmlearn (compatible with scikit-learn)
  - [https://github.com/hmmlearn/hmmlearn](https://github.com/hmmlearn/hmmlearn) (or `pip install hmmlearn`)
- Matlab (integrated)
- C++: HTK3
  - [http://htk.eng.cam.ac.uk/](http://htk.eng.cam.ac.uk/)

Thank You!

Markov models