Announcements
Review

- The mixture of experts model:

\[
p(y \mid x; \theta) = \sum_{j=1}^{k} p(j \mid x) p(y \mid x; \theta_j).
\]

- Estimation: via EM.
Plan for today

- Markov models
- Hidden Markov models
Markov models

- The $k$-th order Markov model:

$$p(x_i | x_1, \ldots, x_{i-1}) = p(x_i | x_{i-k}, \ldots, x_{i-1}).$$

- Zeroth order:
Markov models

- The $k$-th order Markov model:

$$p(x_i | x_1, \ldots, x_{i-1}) = p(x_i | x_{i-k}, \ldots, x_{i-1}).$$

- Zeroth order:

- First order (bigrams):

... $x_{i-2} \rightarrow x_{i-1} \rightarrow x_i \rightarrow \cdots$
Markov models

- The $k$-th order Markov model:

$$p(x_i | x_1, \ldots, x_{i-1}) = p(x_i | x_{i-k}, \ldots, x_{i-1}).$$

- Zeroth order:

- First order (bigrams):

- Second order (trigrams):
Dynamic models

• Suppose \( X = x_1, \ldots, x_N \) generated by 1-st order Markov process.

\[
p(X) = p(x_1)p(x_2 | x_1)p(x_3 | x_2) \cdots p(x_N | x_{N-1}).
\]

• If \( p(x_{t+1} | x_t) \) does not depend on \( i \), the Markov model is homogenous.

• Discrete observations called states \( s_t \in \{1, \ldots, m\} \), model is parametrized by:
  
  – Starting probability \( s_1 \sim p_0: a m \times 1 \) vector.
  
  – Transition probability matrix \( P \):

\[
P_{ij} = p(s_{t+1} = j | s_t = i) = p(i \rightarrow j).
\]
Representing discrete Markov models

Three equivalent graphical representations:

- Graphical model

\[ s_{t-2} \rightarrow s_{t-1} \rightarrow s_t \]
Representing discrete Markov models

Three equivalent graphical representations:

- **Graphical model**

- **State transition diagram**
Representing discrete Markov models

Three equivalent graphical representations:

- **Graphical model**

- **State transition diagram**

- **Trellis**
Markov model: example

\[ p_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad P = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \]

- Reminder: \( p(s_{t+k} = i \mid s_t = j) = (P^k)_{ij} \)

- Reminder: Markov chain is \textit{ergodic} if \((P^k)_{ij} > 0\) for all \(i, j\) and some fixed \(k\).
  - Cf. random walk on affinity graph in spectral clustering.
Estimating Markov model parameters

- Need to estimate $p_0$, $P$.

- Log-likelihood of observed $s_1, \ldots, s_N$: $\log p_0(s_1) + \sum_{t=2}^{N} \log p(s_t | s_{t-1})$

- ML for $P$: let $n_{r \rightarrow s}$ be the # of times $s_{t-1} = r$, $s_t = s$.

  $$\hat{P}_{ij} = \frac{n_{i \rightarrow j}}{\sum_{r} n_{i \rightarrow r}}$$

- ML for $p_0$: trivial if we have $L > 1$ sequences.

  $$\hat{p}_0(s) = (\# \text{ of times } s_1 = s)/L.$$
Estimating Markov model parameters

- What if we only have a single sequence?
Estimating Markov model parameters

- What if we only have a single sequence?

\[ \hat{p}_0(s) = \frac{1}{N} \sum_i n_{i\rightarrow s} \]
Estimating Markov model parameters

- What if we only have a single sequence?

\[ \hat{p}_0(s) = \frac{1}{N} \sum_i n_{i\rightarrow s} \]

- Is ML a good estimator for \( P \)?
  - With \( m \) states, \( P \) has \( m^2 \) parameters;
  - Some states/transitions are infrequent.
Estimating Markov model parameters

• What if we only have a single sequence?

\[ \hat{p}_0(s) = \frac{1}{N} \sum_{i} n_{i \rightarrow s} \]

• Is ML a good estimator for \( P \)?
  
  – With \( m \) states, \( P \) has \( m^2 \) parameters;
  
  – Some states/transitions are infrequent.

• Often need some “smoothing” (regularization).
Markov models for language

- $k$-th order Markov model is also called a $k$-gram model

- Example (C. Shannon): character $k$-grams as a generative model.

  \[ k = 0 \quad \text{XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJELYVKCQSGXYD QPAAMKBZAACIBZLHJQD} \]
Markov models for language

- $k$-th order Markov model is also called a $k$-gram model

- Example (C. Shannon): character $k$-grams as a generative model.

  $k = 0$  
  XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGXYD 
  QPAAMKBZAACIBZLHJQD

  $k = 1$  
  ROCRO HLI RGWR NMIELWIS EU LL NBBESEBYA TH EEI ALHENHTTPA 
  00 BTTV
Markov models for language

- \(k\)-th order Markov model is also called a \(k\)-gram model

- Example (C. Shannon): character \(k\)-grams as a generative model.
  
  \[
  \begin{align*}
  k &= 0 & \text{XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGXYD} \\
          & & \text{QPAAMKBZAACIBZLHJQD} \\
  k &= 1 & \text{OCRO HLI RGWR NMIELWIS EU LL NBBESEBYA TH EEI ALHENHTTPA} \\
          & & \text{OO BTTV} \\
  k &= 2 & \text{ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN} \\
          & & \text{D ITRANASIVE TUICOOWE FUSO TIZIN ANDY TOBE SEACE CTISBE}
  \end{align*}
  \]
Markov models for language

- $k$-th order Markov model is also called a $k$-gram model

- Example (C. Shannon): character $k$-grams as a generative model.

  $k = 0$  
  XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGXYD
  QPAAMKBZAACIBZLHJQD

  $k = 1$  
  OCRO HLI RGWR NMIELWIS EU LL NBBESEBYA TH EEI ALHENHTTPA
  OO BTTV

  $k = 2$  
  ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN
  D ILONASIVE TUCOOWE FUSO TIZIN ANDY TOBE SEACE CTISBE

  $k = 3$  
  IN NO IST LAY WHEY CRATICT FROURE BERS GROCID PONDENOME
  OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
Markov models for language

- $k$-th order Markov model is also called a $k$-gram model

- Example (C. Shannon): character $k$-grams as a generative model.

  \begin{align*}
  k = 0 & \quad \text{XFOML RXKHRJFFJUU ZLPWCFWKCYJ FFJETYVKQSGXYD} \\
  & \quad \text{QPAAMKBZAACIBZLHJQD} \\
  k = 1 & \quad \text{OCRO HLI RGWR NMIELWIS EU LL NBBESEBYA TH EEI ALHENHTTPA} \\
  & \quad \text{00 BTTV} \\
  k = 2 & \quad \text{ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN} \\
  & \quad \text{D ILONASIVE TUOCOWE FUSO TIZIN ANDY TOBE SEACE CTISBE} \\
  k = 3 & \quad \text{IN NO IST LAY WHEY CRATICT FROURE BERS GROCID PONDENOME} \\
  & \quad \text{OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE} \\
  k = 4 & \quad \text{THE GENERATED JOB PROVIDUAL BETTER TRAND THE DISPLAYED} \\
  & \quad \text{CODE ABOVERY UPONDULTS WELL THE CODERST IN THESTICAL IT TO} \\
  & \quad \text{HOCK BOTHE}
  \end{align*}
How to wreck a nice beach you sing calm incense

(cartoon stolen from T. Hoffman’s slides)
Cartoon

by Jim Unger

"...THE WHOLE TRUTH...

...DE TOLL-BOOTH...

...AND NOTHING BUT THE TRUTH.

...AN NUTS SING ON DE ROOF."
Cartoon

NOW TELL US IN YOUR OWN WORDS EXACTLY WHAT HAPPENED.
• A language model may reduce ambiguity in acoustic signal.
Cartoon

• A language model may reduce ambiguity in acoustic signal.

• However, the state of the Markov chain is not observed.
Hidden Markov Models

- **Hidden Markov model (HMM):**

- hidden states $s_1, \ldots, s_N$ form a Markov chain; observations $x_1, \ldots, x_N$ depend on states.

\[
p(s_1, \ldots, s_N, x_1, \ldots, x_N) = p(s_1)
\]
Hidden Markov Models

- **Hidden Markov model (HMM):**

- hidden states \( s_1, \ldots, s_N \) form a Markov chain; observations \( x_1, \ldots, x_N \) depend on states.

\[
p(s_1, \ldots, s_N, x_1, \ldots, x_N) = p(s_1)p(x_1 | s_1)
\]
Hidden Markov Models

- **Hidden Markov model (HMM):**

- Hidden states $s_1, \ldots, s_N$ form a Markov chain; observations $x_1, \ldots, x_N$ depend on states.

$$p(s_1, \ldots, s_N, x_1, \ldots, x_N) = p(s_1)p(x_1 | s_1)p(s_2 | s_1)$$
Hidden Markov Models

- **Hidden Markov model (HMM):**

- hidden states $s_1, \ldots, s_N$ form a Markov chain; observations $x_1, \ldots, x_N$ depend on states.

\[
p(s_1, \ldots, s_N, x_1, \ldots, x_N) = p(s_1)p(x_1 | s_1)p(s_2 | s_1)p(x_2 | s_2) \cdots \cdot p(s_N | s_{N-1})
\]
- **Hidden Markov model (HMM):**

  - hidden states $s_1, \ldots, s_N$ form a Markov chain; observations $x_1, \ldots, x_N$ depend on states.

  \[
  p(s_1, \ldots, s_N, x_1, \ldots, x_N) = p(s_1)p(x_1 | s_1)p(s_2 | s_1)p(x_2 | s_2) \cdots p(s_N | s_{N-1})p(x_N | s_N)
  \]
Discrete HMM: toy example

- Two biased coins:

\[
p(x_t = H \mid s_t = 1) = 0.75, \quad p(x_t = T \mid s_t = 1) = 0.25;
\]
\[
p(x_t = H \mid s_t = 2) = 0.25, \quad p(x_t = T \mid s_t = 2) = 0.75.
\]

- HMM parameters:

\[
\mathbf{p}_0 = [0.5, 0.5]^T, \quad \mathbf{P} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}.
\]

- Sampling from this model:

\[
s_t : \quad 2222111222221111111112222222211111122222222
\]
\[
x_t : \quad \text{THHTHHHTTTHTHHTHHTHTTTTTTHTTTHTHTHHTHTTTTH}
\]
Continuous HMM: toy example

- Four states, sampled clockwise;
- Gaussian emission probability

\[ p(x_t \mid s_t = s) = \mathcal{N}(x; \mu_s, \Sigma_s) \]
Continuous HMM: toy example

- Four states, sampled clockwise;
- Gaussian emission probability

\[
p(x_t \mid s_t = s) = \mathcal{N}(x; \mu_s, \Sigma_s)
\]

\[
p_0 = [0.25, 0.25, 0.25, 0.25]^T,
\]
Continuous HMM: toy example

- Four states, sampled clockwise;
- Gaussian emission probability

\[ p(x_t \mid s_t = s) = \mathcal{N}(x; \mu_s, \Sigma_s) \]

\[ p_0 = [0.25, 0.25, 0.25, 0.25]^T, \quad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \]
HMM: three fundamental problems

• Inference regarding observation sequence $x_1, \ldots, x_N$
  – Compute likelihood of a model given observations
    ⇒ the forward-backward algorithm
  – Sample from the model

• Inference regarding hidden states
  – Estimate most likely state sequence
    ⇒ the Viterbi algorithm

• Estimating the model parameters
  ⇒ EM (Baum-Welch algorithm)
Forward-backward probabilities

- **Forward probabilities:**

\[ \alpha_t(s) \triangleq p(x_1, \ldots, x_t, s_t = s) \]

- **Prediction:** current state given the past

\[ \frac{\alpha_t(s)}{\sum_{s'} \alpha_t(s')} = p(s_t = s | x_1, \ldots, x_t) \]

- **Backward probabilities:** diagnostic (future given the state)

\[ \beta_t(s) \triangleq p(x_{t+1}, \ldots, x_N | s_t = s) \]
Computing forward probabilities: $t = 1$

$$\alpha_1(1) = p(x_1, s_1 = 1)$$
Computing forward probabilities: $t = 1$

$$\alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1);$$
Computing forward probabilities: $t = 1$

$$\alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1);$$

$$\alpha_1(2) = p(x_1, s_1 = 2) = p_0(2)p(x_1 | s_1 = 2);$$
Computing forward probabilities: \( t = 2 \)

\[
\alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1),
\]
\[
\alpha_1(2) = p(x_1, s_1 = 2) = p_0(2)p(x_1 | s_1 = 2)
\]

\[
\alpha_2(1) = p(x_1, x_2, s_2 = 1)
\]
Computing forward probabilities: $t = 2$

\[ \alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1), \]
\[ \alpha_1(2) = p(x_1, s_1 = 2) = p_0(2)p(x_1 | s_1 = 2). \]

\[ \alpha_2(1) = p(x_1, x_2, s_2 = 1) = p(x_1, s_2 = 1)p(x_1 | s_2 = 1). \]
Computing forward probabilities: \( t = 2 \)

\[
\alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1), \\
\alpha_1(2) = p(x_1, s_1 = 2) = p_0(2)p(x_1 | s_1 = 2)
\]

\[
\alpha_2(1) = p(x_1, x_2, s_2 = 1) = p(x_1, s_2 = 1)p(x_1 | s_2 = 1) \\
= [\alpha_1(1)p(1 \rightarrow 1) + \alpha_1(2)p(2 \rightarrow 1)]p(x_1 | s_2 = 1)
\]
Computing forward probabilities: $t = 2$

\[
\alpha_1(1) = p(x_1, s_1 = 1) = p_0(1)p(x_1 | s_1 = 1),
\]
\[
\alpha_1(2) = p(x_1, s_1 = 2) = p_0(2)p(x_1 | s_1 = 2)
\]

\[
\alpha_2(1) = p(x_1, x_2, s_2 = 1) = p(x_1, s_2 = 1)p(x_1 | s_2 = 1)
\]
\[
= [\alpha_1(1)p(1 \rightarrow 1) + \alpha_1(2)p(2 \rightarrow 1)] p(x_1 | s_2 = 1)
\]
\[
\alpha_2(2) = [\alpha_1(2)p(1 \rightarrow 2) + \alpha_1(2)p(2 \rightarrow 2)] p(x_1 | s_2 = 2)
\]
Forward probabilities: recursion

\[ \alpha_t(s) = p(x_1, \ldots, x_t, s_t = s) \]

\[ \alpha_1(s) = p_0(s)p(x_1 | s_1 = s) \]
Forward probabilities: recursion

\[ \alpha_t(s) = p(x_1, \ldots, x_t, s_t = s) \]

\[ \alpha_1(s) = p_0(s)p(x_1 | s_1 = s) \]

\[ \alpha_t(s) = \left[ \sum_{s'} \alpha_{t-1}(s')p(s' \rightarrow s) \right] p(x_t | s_t = s) \]
Backward probabilities

$$\beta_t(s) \triangleq p(x_{t+1}, \ldots, x_N \mid s_t = s)$$

$$\beta_N(s) = p(\varepsilon \mid s_N = s) \triangleq 1$$
Backward probabilities

\[ \beta_t(s) \triangleq p(x_{t+1}, \ldots, x_N \mid s_t = s) \]

\[ \beta_N(s) = p([], s_N = s) \triangleq 1 \]

\[ \beta_t(s) = \sum_{s'} [p(s \rightarrow s') p(x_{t+1} \mid s_{t+1} = s') \beta_{t+1}(s')] \]
Forward-Backward algorithm

\[ \alpha_1(s) = p_0(s)p(x_1 | s_1 = s), \]

\[ \alpha_t(s) = \left[ \sum_{s'} \alpha_{t-1}(s')p(s' \rightarrow s) \right] p(x_t | s_t = s); \]

\[ \beta_N(s) = 1, \]

\[ \beta_t(s) = \sum_{s'} [p(s \rightarrow s')p(x_{t+1} | s_{t+1} = s') \beta_{t+1}(s')]. \]

- We need two passes (forward and backward) to compute \( \alpha_t, \beta_t \) for all \( t = 1, \ldots, N \).

- Time complexity with \( M \) states:
Forward-Backward algorithm

\[
\alpha_1(s) = p_0(s)p(x_1 | s_1 = s),
\]

\[
\alpha_t(s) = \left[ \sum_{s'} \alpha_{t-1}(s')p(s' \rightarrow s) \right] p(x_t | s_t = s);
\]

\[
\beta_N(s) = 1,
\]

\[
\beta_t(s) = \sum_{s'} [p(s \rightarrow s')p(x_{t+1} | s_{t+1} = s') \beta_{t+1}(s')].
\]

- We need two passes (forward and backward) to compute \( \alpha_t, \beta_t \) for all \( t = 1, \ldots, N \)

- Time complexity with \( M \) states: \( O(NM^2) \).
Next time

HMM: likelihood, EM.