

CS195-5 : Introduction to Machine Learning

Lecture 30

Greg Shakhnarovich

November 27, 2006

Announcements

Review: AdaBoost

1. Initialize weights: $W_i^{(0)} = 1/N$

2. Iterate for $m = 1, \dots, M$:

- Find (any) “weak” classifier h_m that attains weighted error

$$\epsilon_m = \frac{1}{2} \left(1 - \sum_{i=1}^N W_i^{(m-1)} y_i h_m(\mathbf{x}_i) \right) < \frac{1}{2}$$

- Let $\alpha_m = \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}$.
- Update the weights and normalize so that $\sum_i W_i^{(m)} = 1$:

$$W_i^{(m)} = \frac{1}{Z} W_i^{(m-1)} e^{-\alpha_m y_i h_m(\mathbf{x}_i)},$$

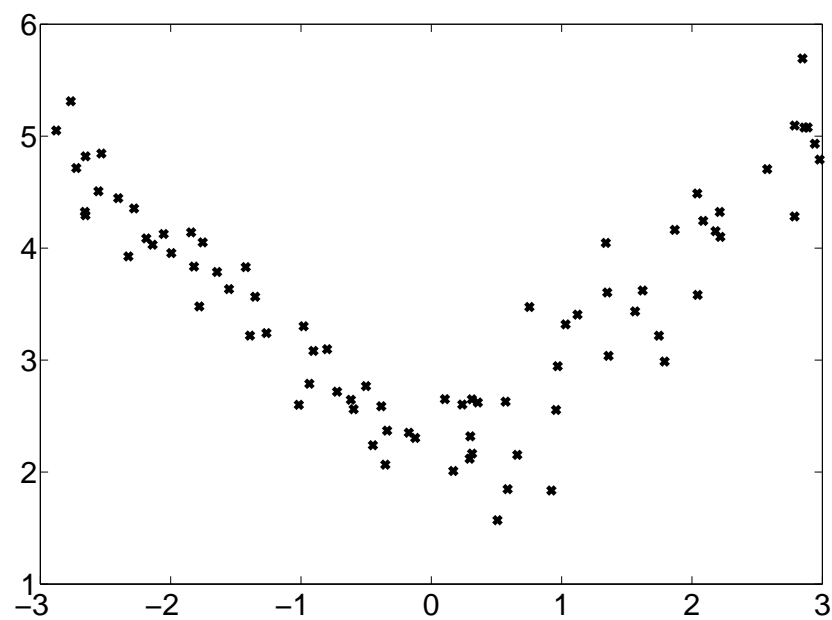
3. The combined classifier: $\text{sign} \left(\sum_{m=1}^M \alpha_m h_m(\mathbf{x}) \right)$

Plan for today

- Mixtures of experts
- Sequential data
 - Markov models

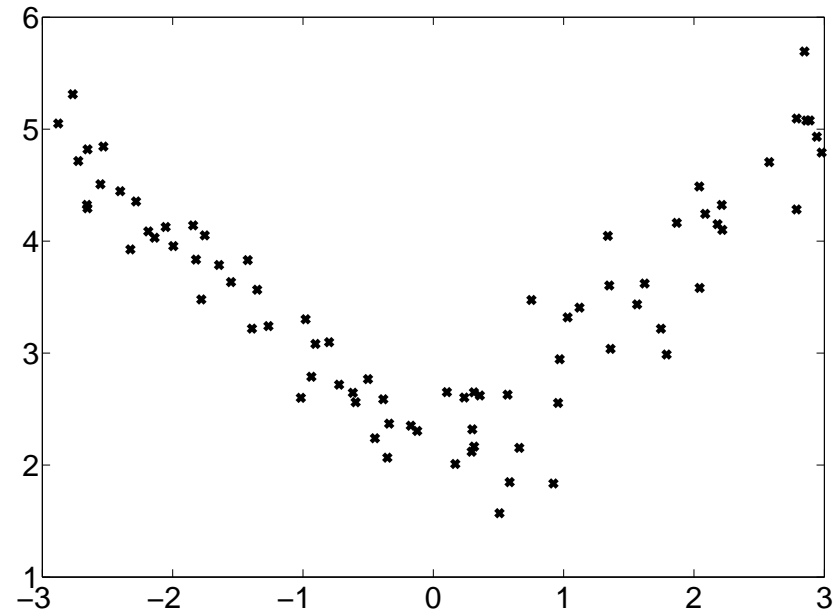
Mixture model for regression

- Example:



Mixture model for regression

- Example:



- We can represent this as a mixture of two regression models
 - Two *experts*;
 - Need to switch between them according to \mathbf{x} .

Mixture of experts model

- Expert j holds a parameteric model $p(y | \mathbf{x}; \theta_j)$, e.g.,

$$\theta_j = \{\mathbf{w}_j, \sigma_j^2\},$$

$$p(y | \mathbf{x}; \theta_j) = \mathcal{N}(y; \mathbf{w}_j^T \mathbf{x}, \sigma_j^2)$$

- The distribution of y is a *conditional mixture* model:

$$p(y | \mathbf{x}; \theta) = \sum_{j=1}^c p(j | \mathbf{x}) p(y | \mathbf{x}; \theta_j).$$

Gating network

$$p(y | \mathbf{x}; \theta) = \sum_{j=1}^c p(j | \mathbf{x}) p(y | \mathbf{x}; \theta_j)$$

- A *gating network* specifies the conditional distribution $p(j | \mathbf{x}; \eta)$
- Possible gating functions:
 - $k = 2$ experts: logistic regression, parametrized by $\eta = \{\mathbf{v}\}$

$$p(j | \mathbf{x}; \eta) = \left(1 + e^{-\mathbf{v}^T \mathbf{x}}\right)^{-1}$$

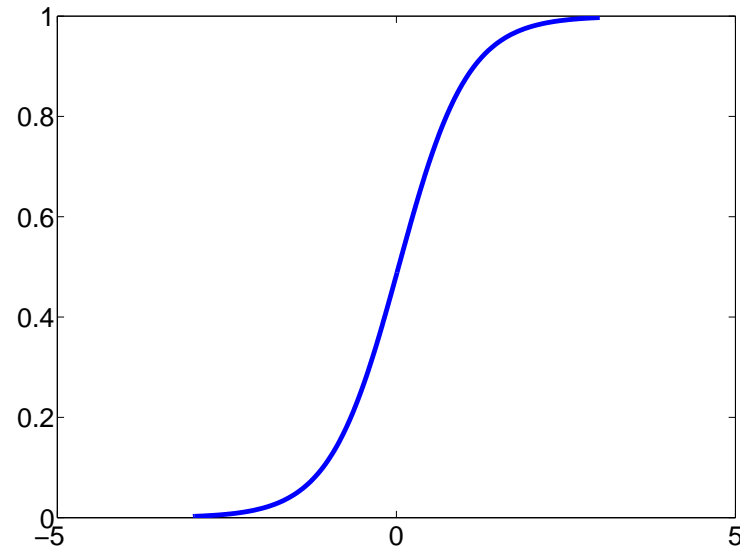
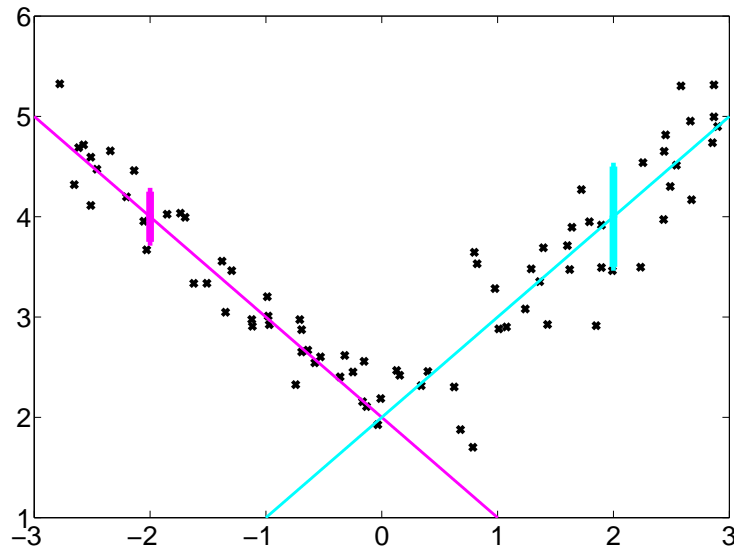
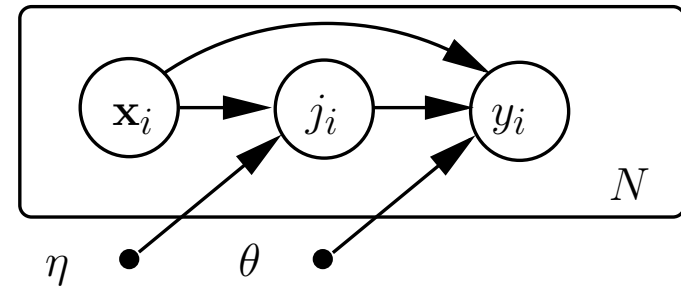
- $k > 2$ experts: softmax, $\eta = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$

$$p(j | \mathbf{x}; \eta) = \frac{e^{\mathbf{v}_j^T \mathbf{x}}}{\sum_{t=1}^k e^{\mathbf{v}_t^T \mathbf{x}}}$$

Conditional mixtures

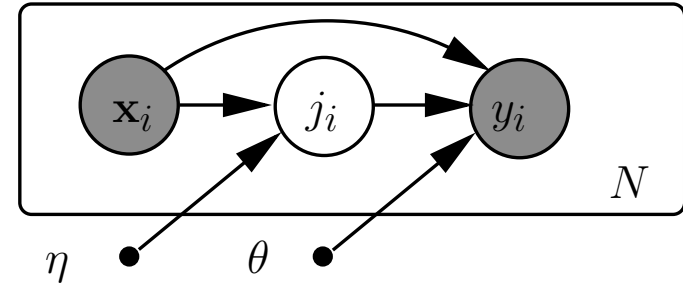
- Parametrization

- Regression models $p(y | \mathbf{x}; \theta_j)$
e.g., linear regressors, $\theta_j = \{\mathbf{w}_j, \sigma_j^2\}$.
- Gating network $p(j | \mathbf{x}; \eta)$
e.g., logistic regression, $\eta = \{\mathbf{v}\}$



Learning a MoE model

- The graphical model:



- Responsibilities:

$$\gamma_{ij} = p(j | \mathbf{x}_i, y_i; \theta, \eta) = \frac{p(j | \mathbf{x}_i; \eta) p(y_i | \mathbf{x}_i; \theta_j)}{\sum_{c=1}^k p(c | \mathbf{x}_i; \eta) p(y_i | \mathbf{x}_i; \theta_c)}$$

EM for mixtures of experts

Initialize random $\theta_j, \sigma_j^2, \eta$.

E-step Compute responsibilities $\gamma_{ij} = p(j | \mathbf{x}_i, y_i; \theta^{old}, \eta^{old})$

M-step Separately:

- For each expert j estimate

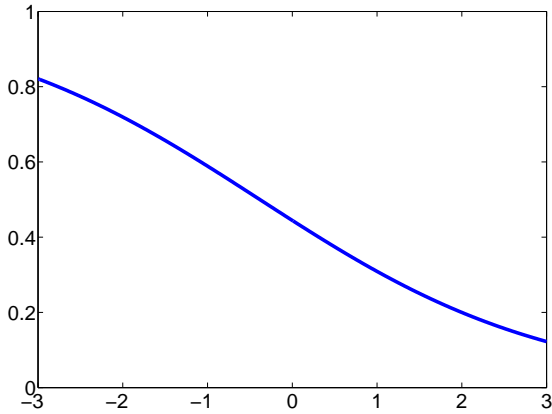
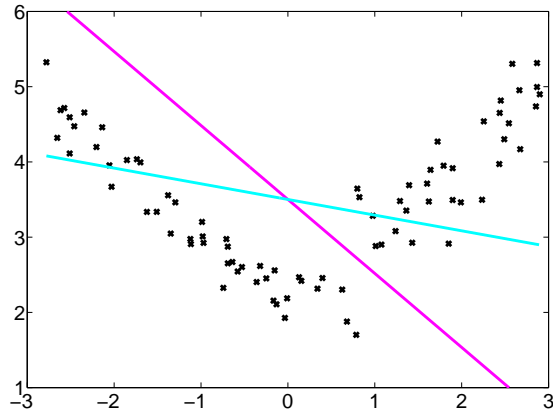
$$\theta_j^{new} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \gamma_{ij} \log p(y_i | \mathbf{x}_i; \theta)$$

- Estimate the gating network:

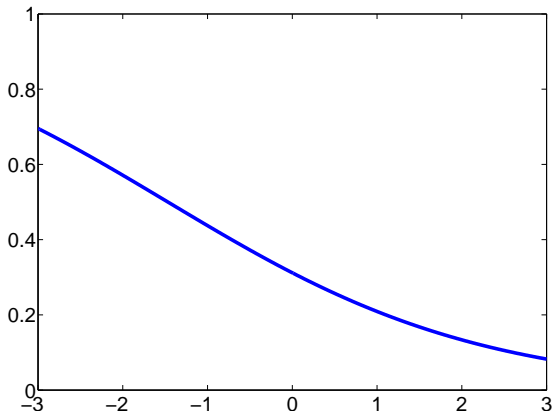
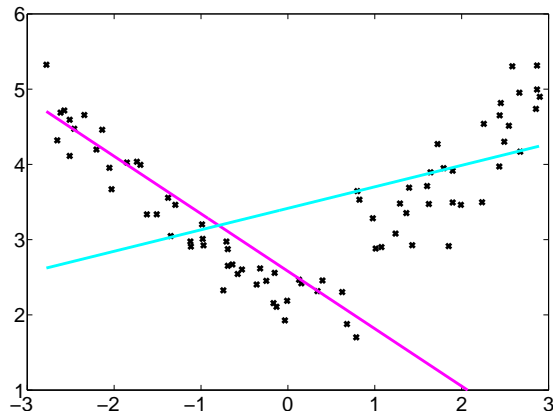
$$\eta^{new} = \operatorname{argmax}_{\eta} \sum_{i=1}^N \sum_{j=1}^k \gamma_{ij} \log p(j | \mathbf{x}_i; \eta)$$

EM for mixtures of experts: example

Iter 1

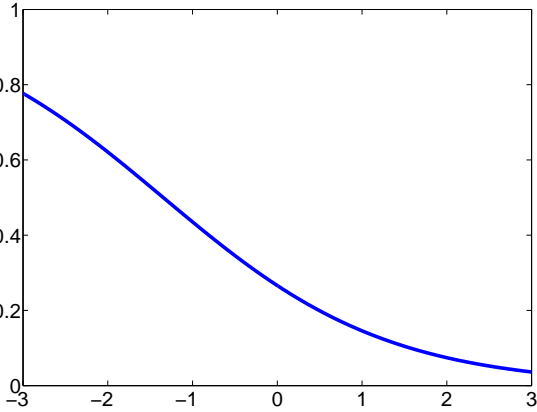
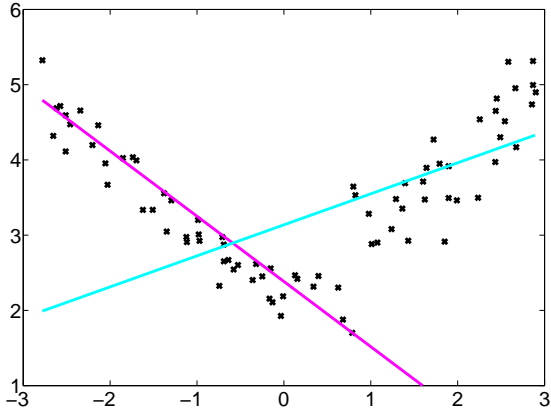


Iter 2

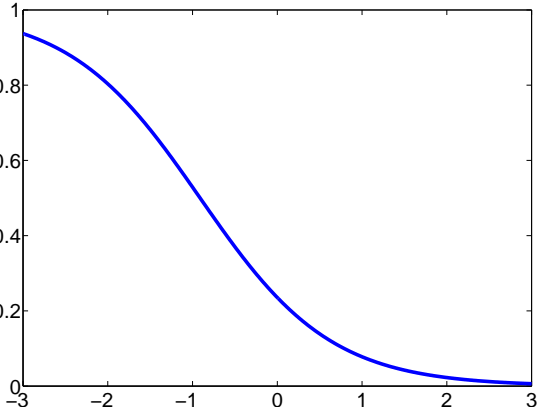
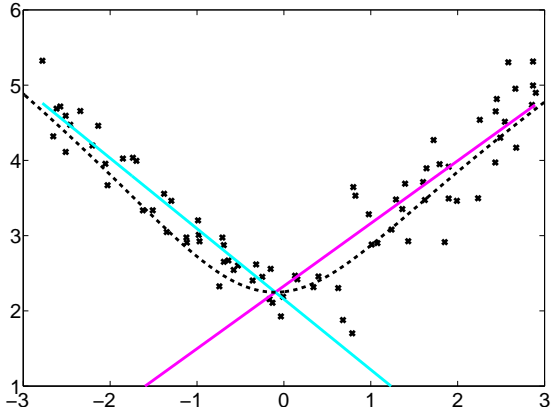


EM for mixtures of experts: example

Iter 3

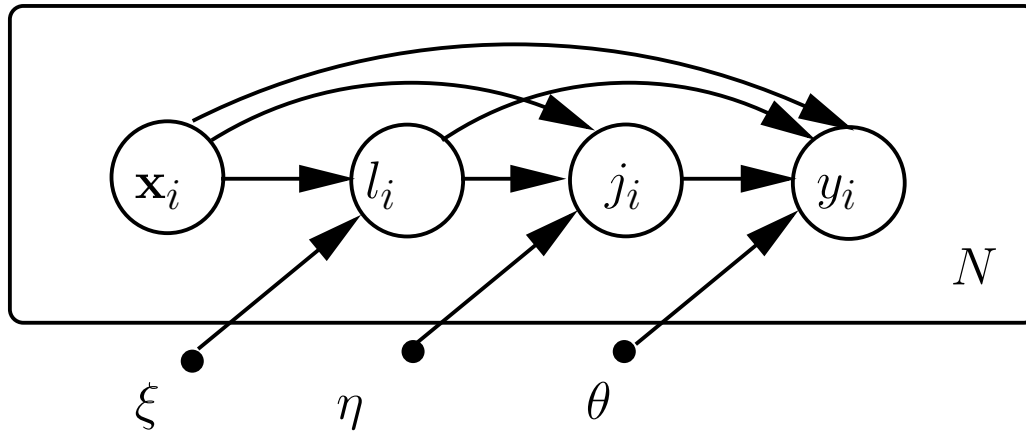


Iter 7



More on mixtures of experts

- MoE for classification: similar idea.
 - Compare to AdaBoost; what are the similarities/differences?
- Hierarchical MoE: multiple levels of gating.



Sequential data

- Departure from the i.i.d. assumption:
 - Probability of observing \mathbf{x}_i depends on $\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_N$.
- The sequential dimension may be temporal or spatial:
 - Speech (measurements of acoustic waveform);
 - Language (words);
 - Images (pixels). . .
- Almost always: assume dependence on past only.

$$p(\mathbf{x}_i \mid \mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_N) = p(\mathbf{x}_i \mid \mathbf{x}_1, \dots, \mathbf{x}_{i-1}).$$

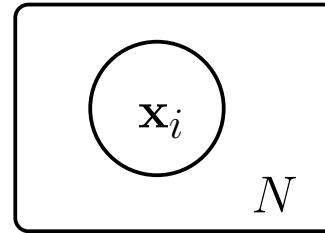
- Still, complexity grows as we increase N .

Markov models

- The k -th order *Markov* model:

$$p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1}) = p(\mathbf{x}_i | \mathbf{x}_{i-k}, \dots, \mathbf{x}_{i-1}).$$

- Zeroth order:

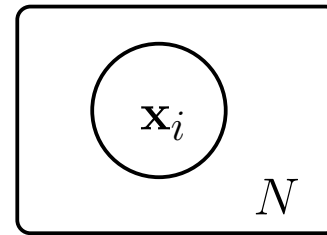


Markov models

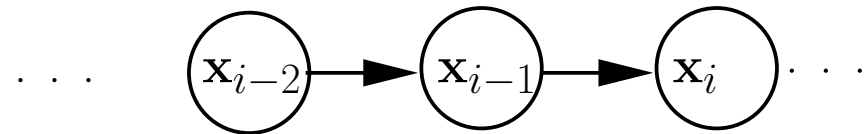
- The k -th order *Markov* model:

$$p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1}) = p(\mathbf{x}_i | \mathbf{x}_{i-k}, \dots, \mathbf{x}_{i-1}).$$

- Zeroth order:



- First order (bigrams):

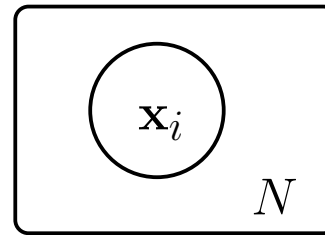


Markov models

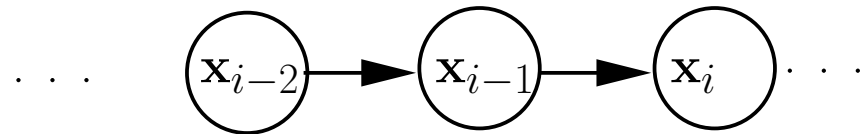
- The k -th order *Markov* model:

$$p(\mathbf{x}_i | \mathbf{x}_1, \dots, \mathbf{x}_{i-1}) = p(\mathbf{x}_i | \mathbf{x}_{i-k}, \dots, \mathbf{x}_{i-1}).$$

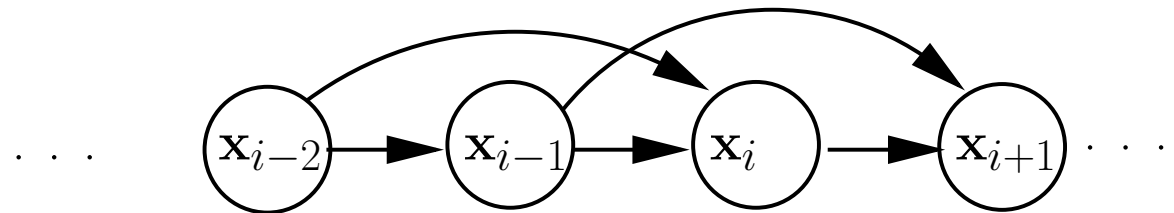
- Zeroth order:



- First order (bigrams):



- Second order (trigrams):



Dynamic models

- Suppose $X = \mathbf{x}_1, \dots, \mathbf{x}_N$ generated by 1-st order Markov process.

$$p(X) = p(\mathbf{x}_1)p(\mathbf{x}_2 | \mathbf{x}_1)p(\mathbf{x}_3 | \mathbf{x}_2) \cdots p(\mathbf{x}_N | \mathbf{x}_{N-1}).$$

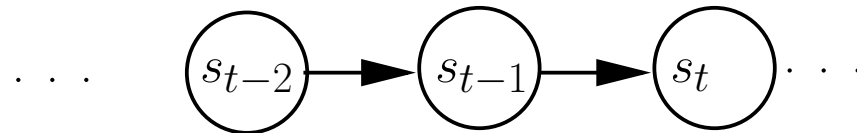
- If $p(\mathbf{x}_{t+1} | \mathbf{x}_t)$ does not depend on i , the Markov model is *homogenous*.
- *Discrete* observations called *states* $s_t \in \{1, \dots, m\}$, model is parametrized by:
 - Starting probability $s_1 \sim p_0$: a $m \times 1$ vector.
 - Transition probability matrix \mathbf{P} :

$$P_{ij} = p(s_{t+1} = j | s_t = i).$$

Representing discrete Markov models

Three equivalent graphical representations:

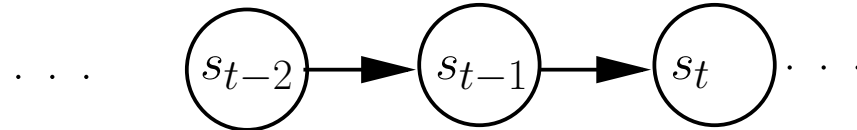
- Graphical model



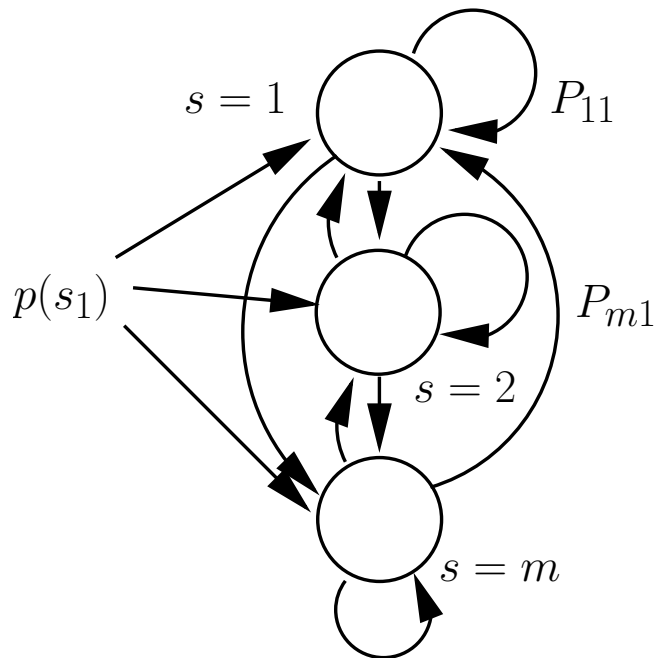
Representing discrete Markov models

Three equivalent graphical representations:

- Graphical model



- State transition diagram

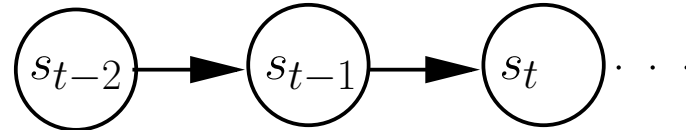


Representing discrete Markov models

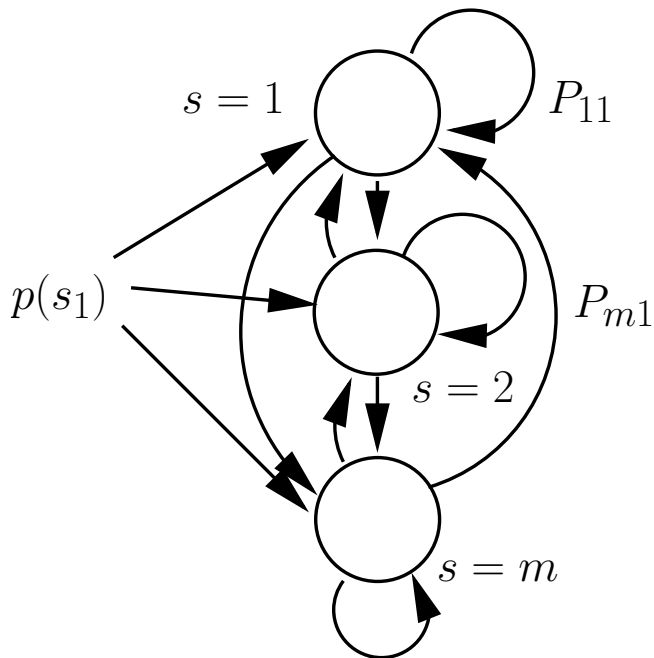
Three equivalent graphical representations:

- Graphical model

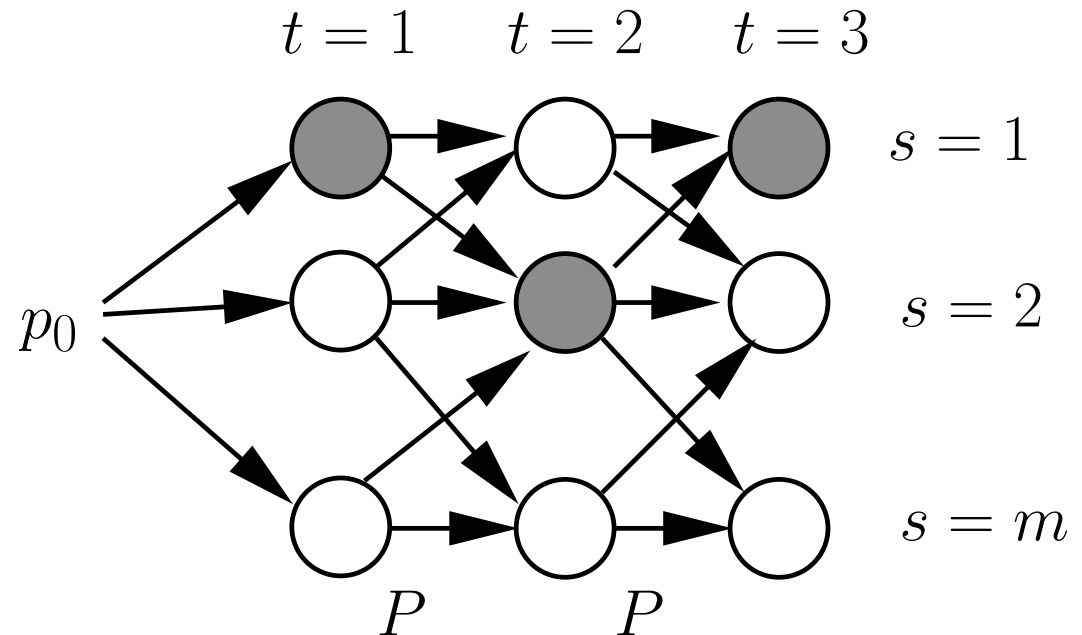
...



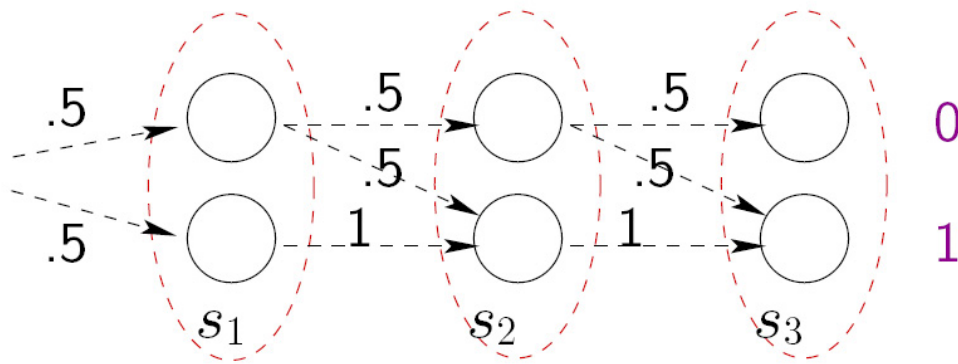
- State transition diagram



- Trellis



Markov model: example



$$p_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}$$

- **Reminder:** $p(s_{t+k} = i \mid s_t = j) = (\mathbf{P}^k)_{ij}$
- **Reminder:** Markov chain is *ergodic* if $(\mathbf{P}^k)_{ij} > 0$ 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, \mathbf{P} .
- Log-likelihood of observed s_1, \dots, s_N : $\log p_0(s_1) + \sum_{t=2}^N \log p(s_t | s_{t-1})$
- ML for \mathbf{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 \mathbf{P} ?
 - With m states, \mathbf{P} has m^2 parameters.

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

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

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

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.

$k = 0$ 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 TUCCOOWE 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

$k = 4$ THE GENERATED JOB PROVIDUAL BETTER TRAND THE DISPLAYED
 CODE ABOVERY UPONDULTS WELL THE CODERST IN THESTICAL IT TO
 HOCK BOTHE

Next time

Hidden Markov Models