

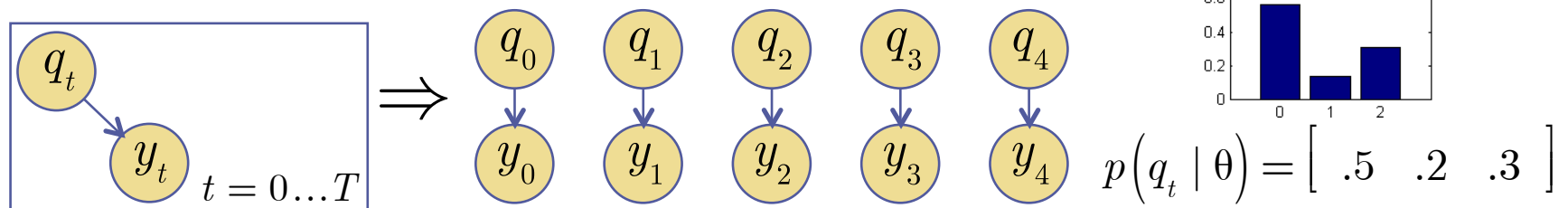
Machine Learning

Topic 19

- Hidden Markov Models
- HMMs as State Machines & Applications
- HMMs Basic Operations
- HMMs via the Junction Tree Algorithm

Hidden Markov Models

- A great application of Junction Tree Algorithm and EM
- Recall mixture of Gaussians model on IID data

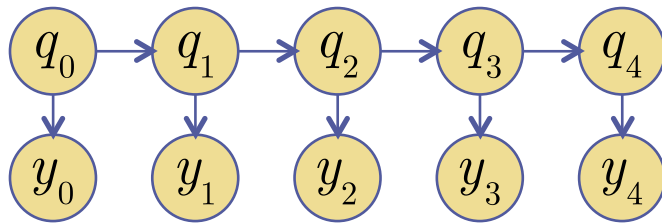


- Example: location data of a single parent as a mixture of Gaussians
- Parent has 3 internal states:
 $q = \{\text{home, daycare, work}\}$
- Based on q , sample from appropriate Gaussian mean and covariance to get $y = (\text{latitude, longitude})$



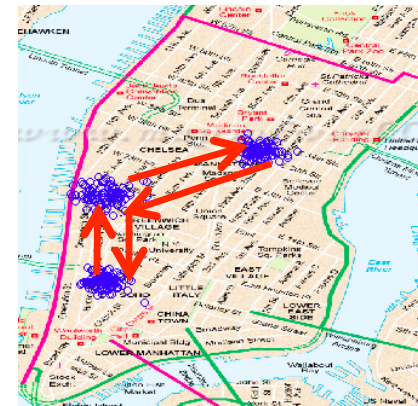
Hidden Markov Models

- Parent drops child at daycare before & after work. Not IID!



$q = \{1 = \text{home}, 2 = \text{daycare}, 3 = \text{work}\}$

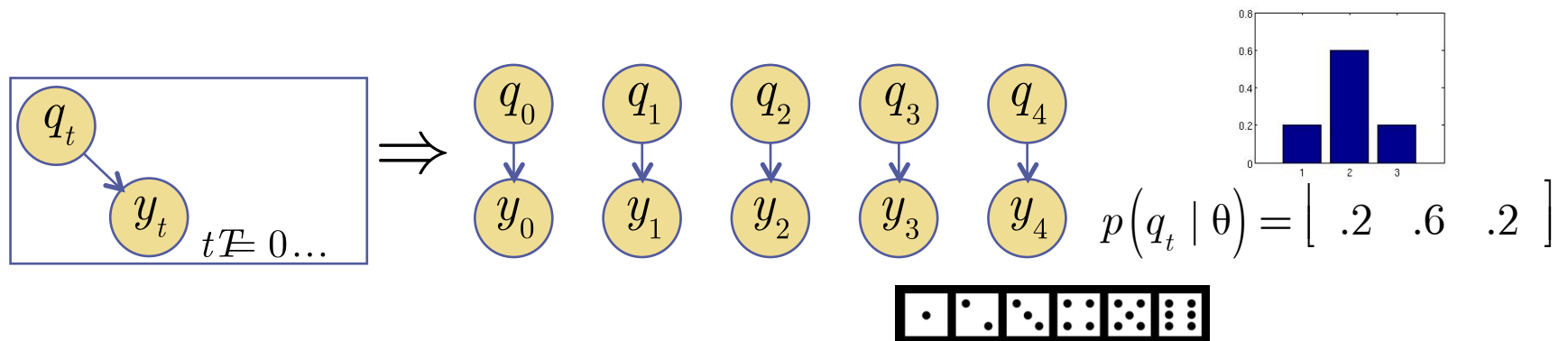
$$p(q_t | q_{t-1}) = \begin{matrix} & \begin{matrix} q_{t-1} = 1 & q_{t-1} = 2 & q_{t-1} = 3 \end{matrix} \\ \begin{matrix} q_t = 1 \\ q_t = 2 \\ q_t = 3 \end{matrix} & \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix} \end{matrix}$$



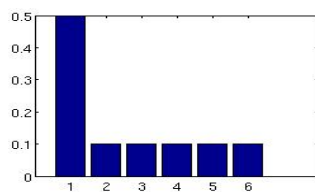
- Have dependence on previous state
- Can't go straight from home to work!
- Now, order of y_0, \dots, y_T matters (in IID order doesn't matter)

Hidden Markov Models

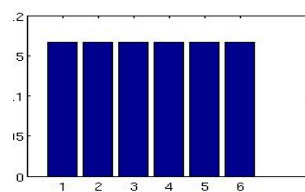
- Consider mixture of multinomials (dice) $y=\{1,2,3,4,5,6\}$



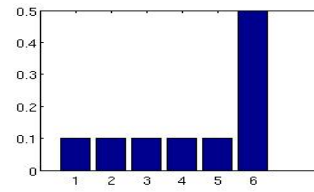
- Example: a crooked casino croupier using mixture of dice.
- You win if he rolls 1,2,3. You lose he rolls 4,5,6.
- Croupier has 3 internal states $q=\{\text{helpful, fair, adversarial}\}$
- Based on q , sample different 'dice' multinomial



1=helpful



2=fair



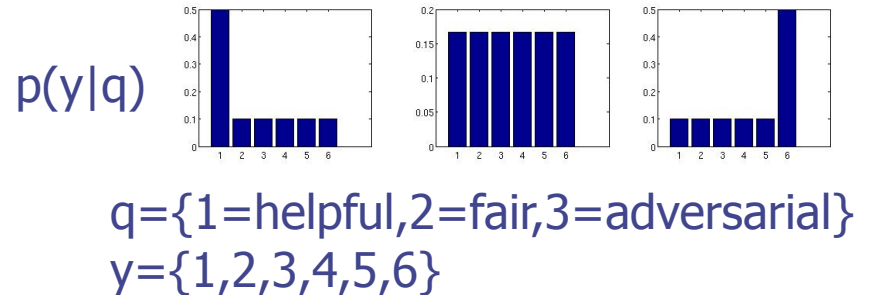
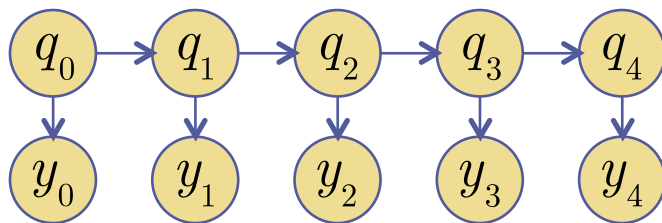
3=adversarial



Hidden Markov Models

- But if the dealer has a memory or mood? Not IID!

5646166166 4321534161414341634 1113114121



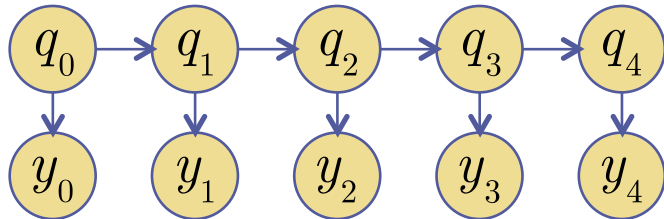
$$p(q_t | q_{t-1}) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \begin{matrix} q_{t-1} = 1 \\ q_{t-1} = 2 \\ q_{t-1} = 3 \end{matrix}$$

$q_t = 1 \quad q_t = 2 \quad q_t = 3$

- If you tip, dealer starts to like you and rolls the helpful die
- Dealer has a memory of his mood and last type of die q_{t-1}
- Will often use same die for q^t as was rolled before...
- Now, order of y_0, \dots, y_T matters (if IID order doesn't matter)

Hidden Markov Models

- Since next choice of the dice depends on previous one...



Order of y_0, \dots, y_T matters
Temporal or sequence model!

- Add left-right arrows. This is a **hidden Markov model**
- Markov: $\text{future} \parallel \text{past} \mid \text{present}$

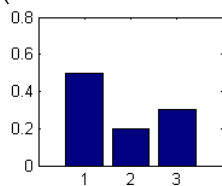
$$p(q_t \mid q_{t-1}, q_{t-2}, \dots, q_1, q_0) = p(q_t \mid q_{t-1})$$

- From graph, have the following general pdf:

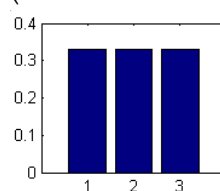
$$p(X_U) = p(q_0) \prod_{t=1}^T p(q_t \mid q_{t-1}) \prod_{t=0}^T p(y_t \mid q_t)$$

- So $p(q_t)$ depends on previous state $q_{t-1} \dots$

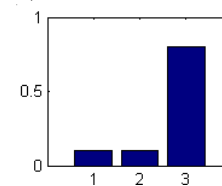
$$p(q_t \mid q_{t-1} = 1)$$



$$p(q_t \mid q_{t-1} = 2)$$

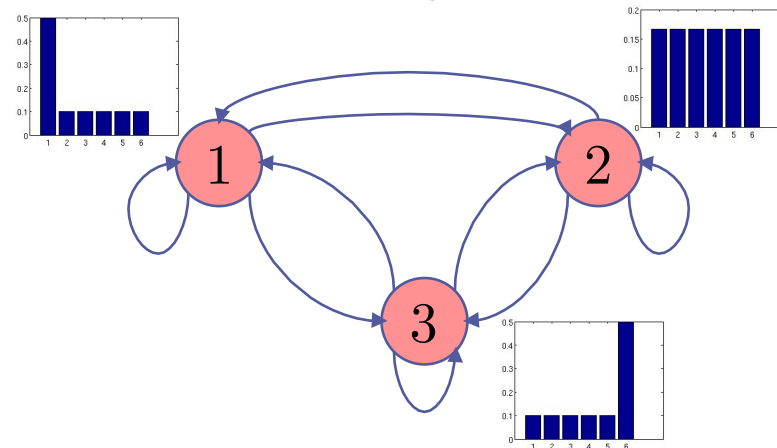
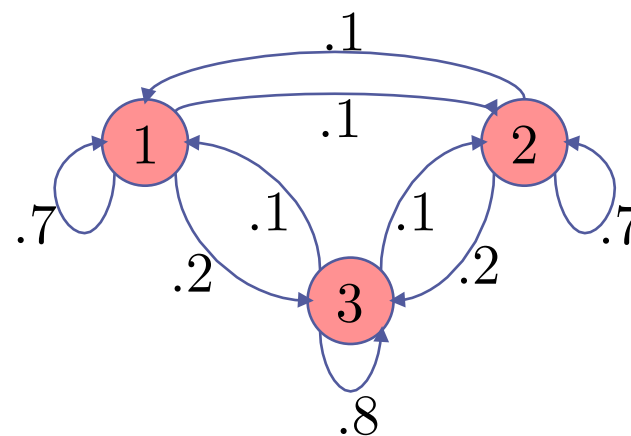


$$p(q_t \mid q_{t-1} = 3)$$



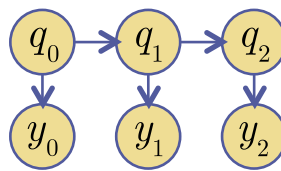
HMMs as State Machines

- HMMs have two variables: **state** q and **emission** y
- Typically, we don't know q (hidden variable 1,2,3,...)
- HMMs are like **stochastic automata** or finite state machines...
next state depends
on previous one...
(helpful, fair, adversarial)
- Can't observe state q directly, just a random related emission y outcome (dice roll) so...
doubly-stochastic automaton

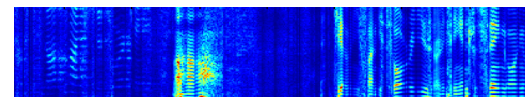


HMM Applications

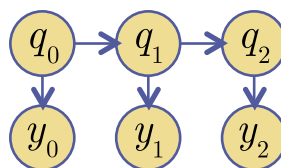
- **Speech Recognition**
phonemes from
audio cepstral vectors



Ba-ra-kk-oo-oo-dd-ah

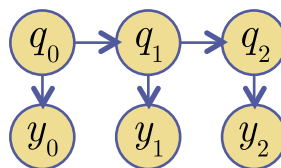


- **Language Parsing**
parts of speech
from words



Noun	Verb	Noun
John	Ate	Pizza

- **Genomics**
splice site from
gene sequence



-Intron- | -Exon- | -Promoter-
GATTACATTATACCACCATACG

HMMs: Parameters

- We focus on HMMs with: discrete **state** q (of size M)
discrete **emission** y (of size N)
- Input will be arbitrary length string: y_0, \dots, y_T
- The pdf or (complete) likelihood is:

$$p(q, y) = p(q_0) \prod_{t=1}^T p(q_t | q_{t-1}) \prod_{t=0}^T p(y_t | q_t)$$

- We don't know hidden states, the incomplete likelihood is:

$$p(y) = \sum_{q_0} \cdots \sum_{q_T} p(q, y)$$

- Assume HMM is stationary, tables are repeated: $\theta = \{\pi, \eta, \alpha\}$

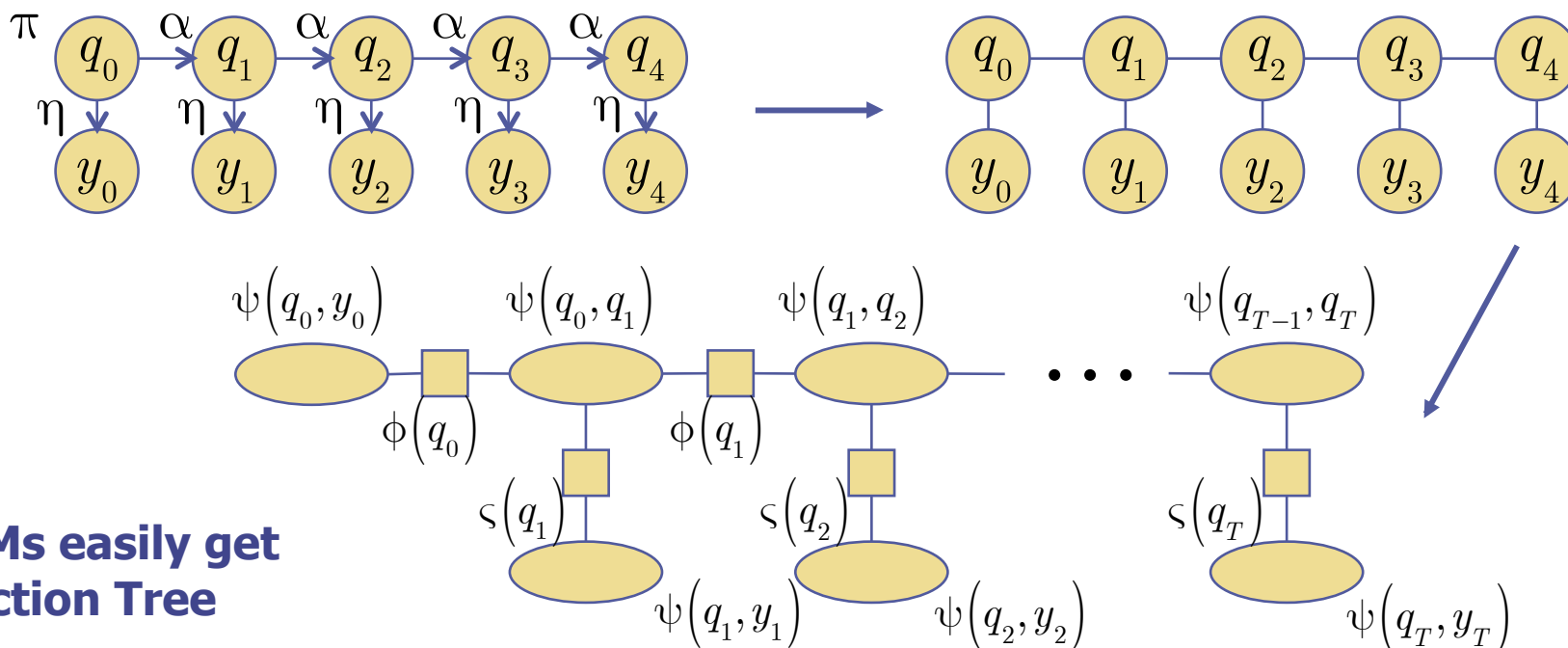
$$p(q_t | q_{t-1}) = \prod_{i=1}^M \prod_{j=1}^M [\alpha_{ij}]^{q_{t-1}^i q_t^j} \quad \sum_{j=1}^M \alpha_{ij} = 1 \quad \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \quad M \times M$$

$$p(y_t | q_t) = \prod_{i=1}^M \prod_{j=1}^N [\eta_{ij}]^{q_t^i y_t^j} \quad \sum_{j=1}^N \eta_{ij} = 1 \quad \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \quad M \times N$$

$$p(q_0) = \prod_{i=1}^M [\pi_i]^{q_0^i} \quad \sum_{j=1}^M \pi_j = 1 \quad \begin{array}{|c|} \hline \\ \hline \\ \hline \end{array} \quad M$$

HMMs: Basic Operations

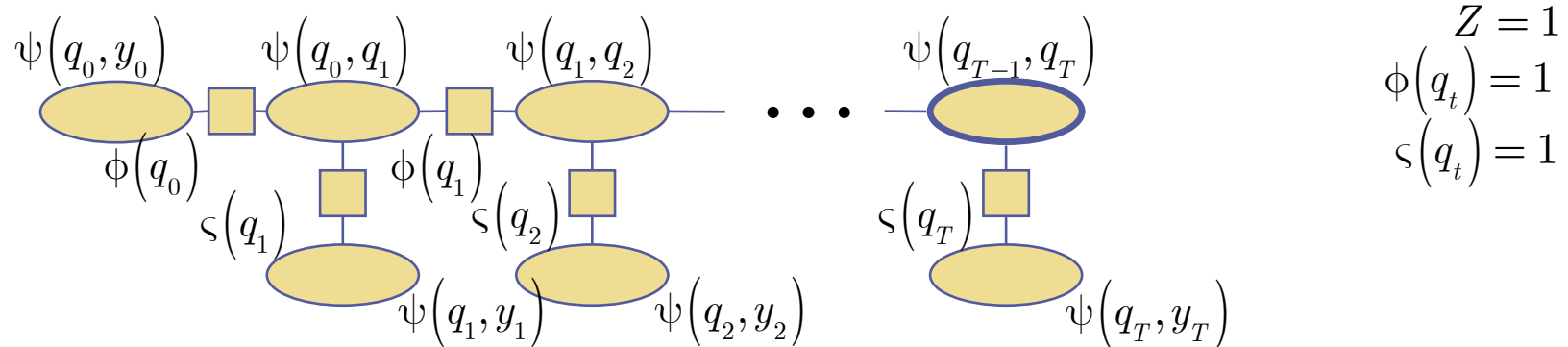
- Would like to do 3 basic things with our HMMs:
 - 1) **Evaluate**: given y_0, \dots, y_T & θ compute $p(y_0, \dots, y_T)$
 - 2) **Decode**: given y_0, \dots, y_T & θ find q_0, \dots, q_T or $p(q_0), \dots, p(q_T)$
 - 3) **Max Likelihood**: given y_0, \dots, y_T learn parameters θ
- Typically use Baum-Welch (α - β algo)... JTA is more general:



**HMMs easily get
Junction Tree**

HMMs: JTA Init & Verify

- **Init:** $\psi(q_0, y_0) = p(q_0)p(y_0 | q_0)$ $\psi(q_t, q_{t+1}) = p(q_{t+1} | q_t) = \alpha_{q_t, q_{t+1}}$ $\psi(q_t, y_t) = p(y_t | q_t)$



- **Collect *up*** (this time it actually doesn't change the zetas)

$$\varsigma^*(q_t) = \sum_{y_t} \psi(q_t, y_t) = \sum_{y_t} p(y_t | q_t) = 1 \quad \psi^*(q_{t-1}, q_t) = \frac{\varsigma^*}{\varsigma} \psi(q_{t-1}, q_t) = \psi(q_{t-1}, q_t)$$

- **Collect *left-right* via phi's:** change backbone to marginals

$$\begin{aligned} \phi^*(q_0) &= \sum_{y_0} \psi(q_0, y_0) = p(q_0) & \psi^*(q_0, q_1) &= \frac{\phi^*}{\phi} \psi(q_0, q_1) = p(q_0, q_1) \\ \phi^*(q_t) &= \sum_{q_{t-1}} \psi^*(q_{t-1}, q_t) = p(q_t) & \psi^*(q_{t-1}, q_t) &= \frac{p(q_{t-1})}{1} p(q_t | q_{t-1}) = p(q_{t-1}, q_t) \end{aligned}$$

- **Distribute:** $\varsigma^{**}(q_t) = \sum_{q_{t-1}} \psi^*(q_{t-1}, q_t) = \sum_{q_{t-1}} p(q_{t-1}, q_t) = p(q_t)$

$$\psi^{**}(q_t, y_t) = \frac{\varsigma^{**}}{\varsigma^*} \psi(q_t, y_t) = \frac{p(q_t)}{1} p(y_t | q_t) = p(y_t, q_t)$$

...done!