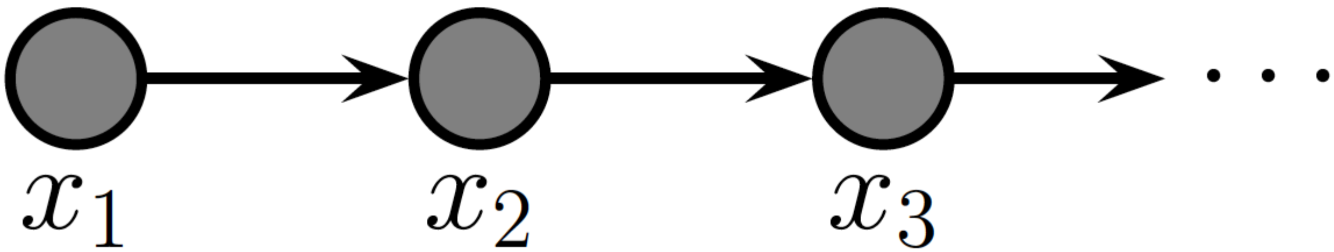
Introduction to Machine Learning

Brown University CSCI 1950-F, Spring 2012
Prof. Erik Sudderth

Lecture 23: Hidden Markov Models

Many figures courtesy Kevin Murphy's textbook, *Machine Learning: A Probabilistic Perspective*

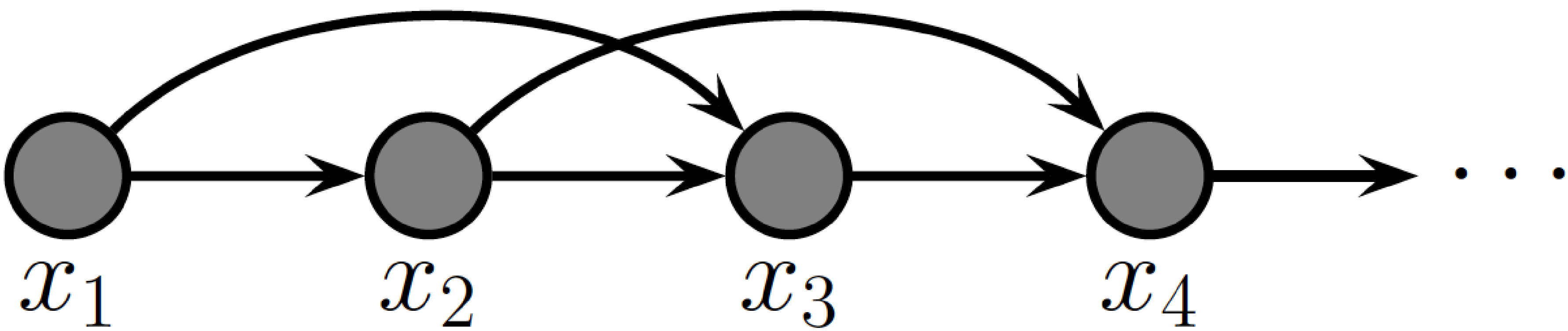
Markov Chains



$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3)\cdots$$

Markov Property

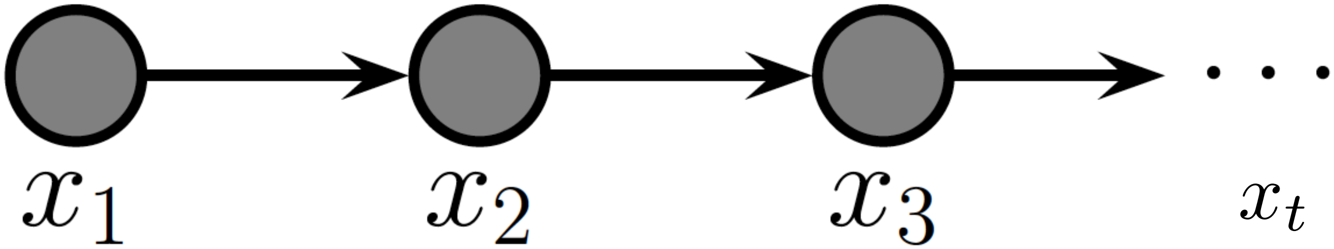
Conditioned on the present, the past and future are independent



$$p(\mathbf{x}_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3)\ldots = p(x_1, x_2)\prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2})$$

Graphical Models vs. State Diagrams

Graphical Model: *One node per time point*



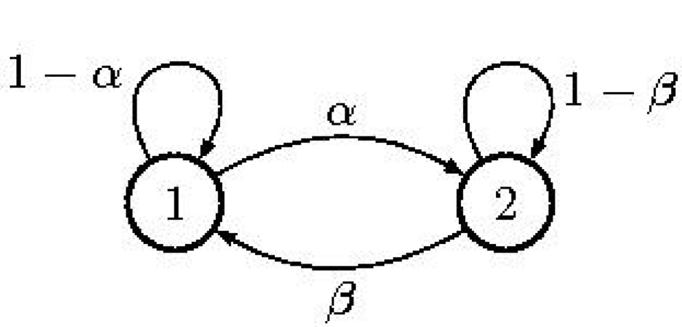
$x_t \in \{1, 2, \ldots, K\}$

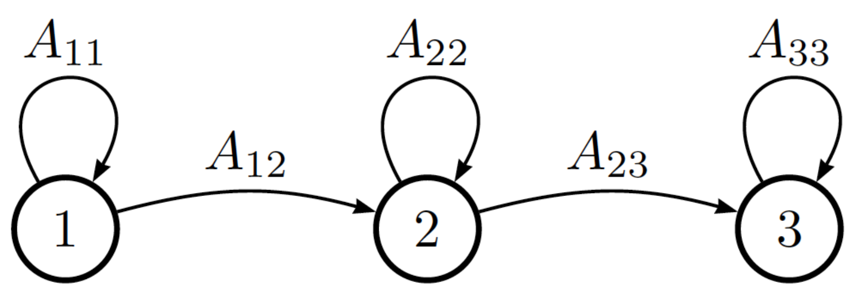
$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3) \cdots$$

Interesting when Markov chain is part of a more complex model.

State Transition Matrix: $A \in \mathbb{R}^{K \times K}, A_{ij} = p(x_t = j \mid x_{t-1} = i)$

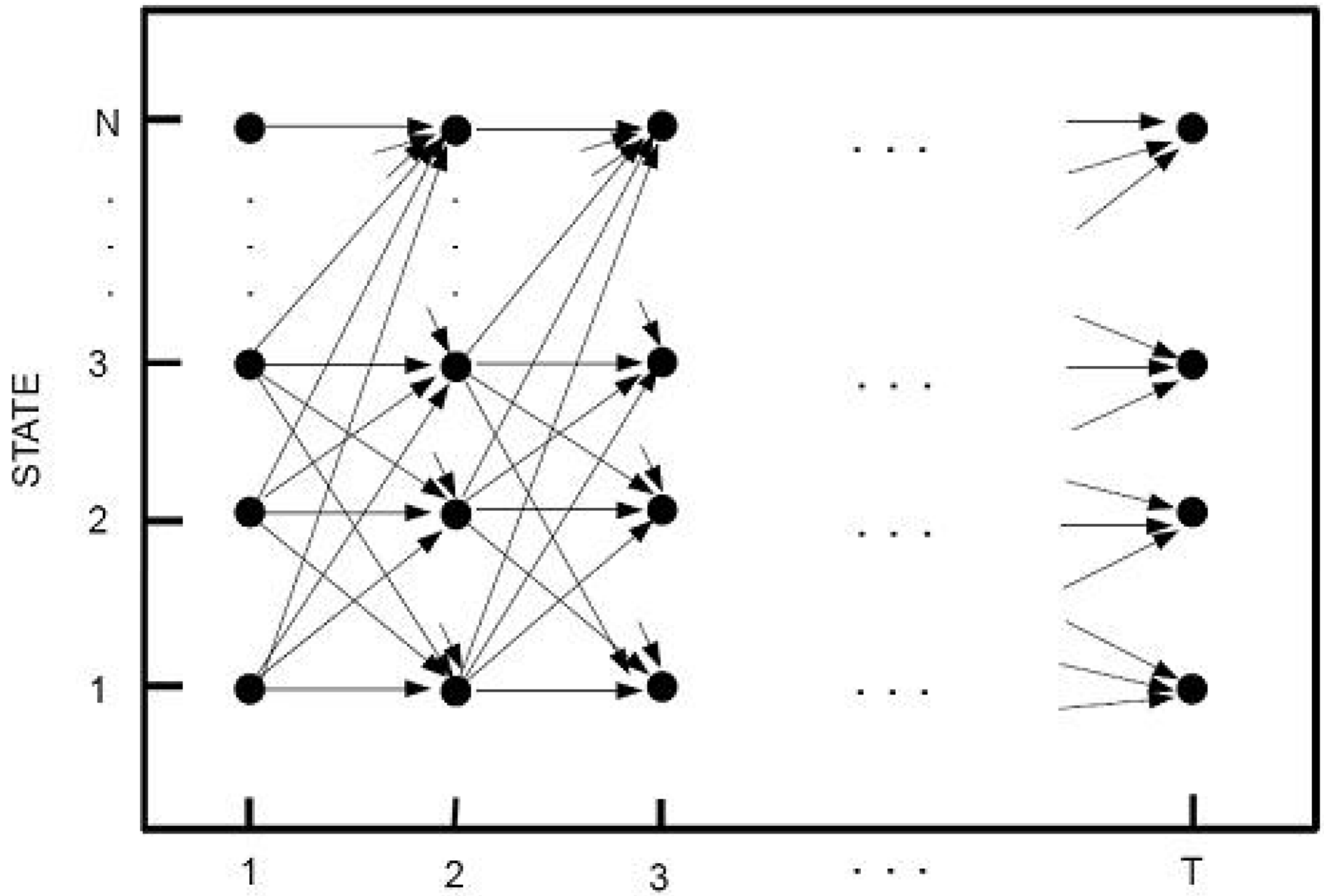
State Transition Diagram: *One node per discrete state*





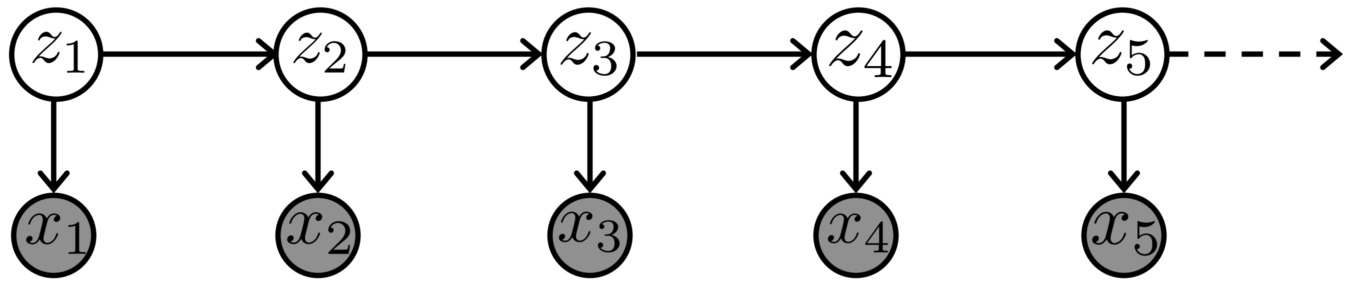
Not a graphical model! Interesting when state transition matrix is sparse.

Trellis Diagrams



- Row for each possible state, column for each time point
- A realized state sequence is a path through the trellis
- State transition diagram determines allowable paths

Hidden Markov Models (HMMs)



$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) \quad = \quad p(\mathbf{z}_{1:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = \left[p(z_1) \prod_{t=2}^{T} p(z_t|z_{t-1}) \right] \left[\prod_{t=1}^{T} p(\mathbf{x}_t|z_t) \right]$$

$z_t \longrightarrow$ Hidden states taking one of *K* discrete values

$x_t \longrightarrow$ Observations taking values in any space

Discrete: M observation symbols $\rightarrow B \in \mathbb{R}^{K \times M}$

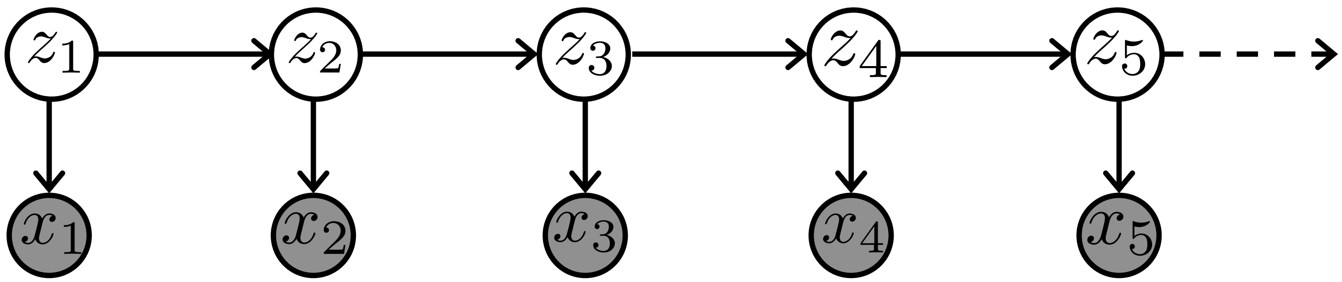
$$p(x_t = \ell \mid z_t = k) = B_{k\ell}$$

Continuous Gaussian:

$$p(x_t \mid z_t = k) = \mathcal{N}(x_t \mid \mu_k, \Sigma_k)$$

Or any convenient family, e.g. an exponential family…

Examples: Sequence Labeling in NLP



Part of speech (POS) tagging:

$\mathbf{z}$:	DT	JJ	NN	VBD	NNP	.
$\boldsymbol{x}$:	the	big	cat	bit	Sam	.

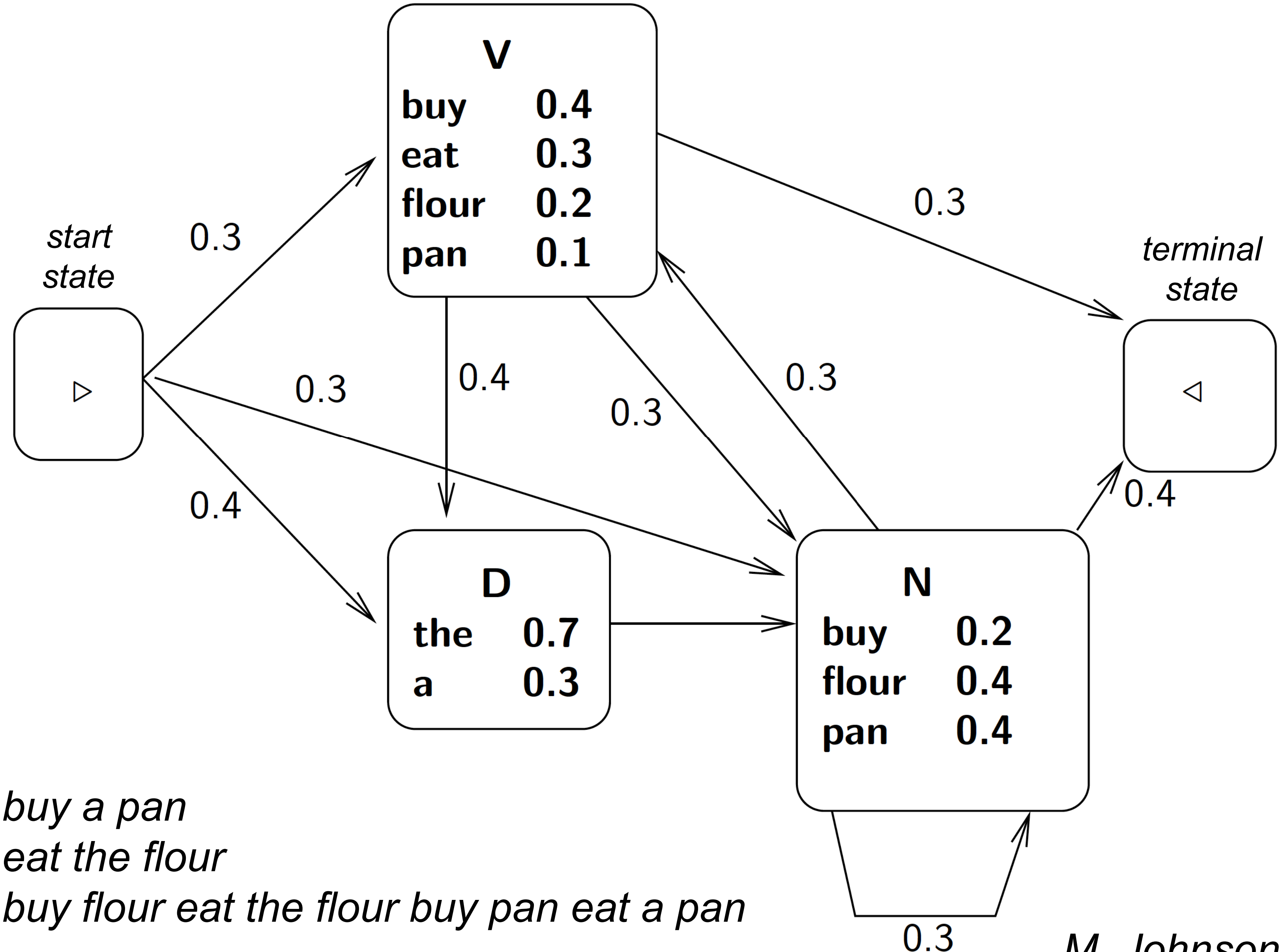
Named entity detection:

$\mathbf{z}$:	[CO	CO]	_	[LOC]	_	[PER]	_
$\boldsymbol{x}$:	XYZ	Corp.	of	Boston	announced	Spade's	resignation

Speech recognition: The $\boldsymbol{x}$ are 100 msec. time slices of acoustic input, and the $\mathbf{z}$ are the corresponding phonemes (i.e., $\mathbf{z}_i$ is the phoneme being uttered in time slice x_i)

M. Johnson, 2009

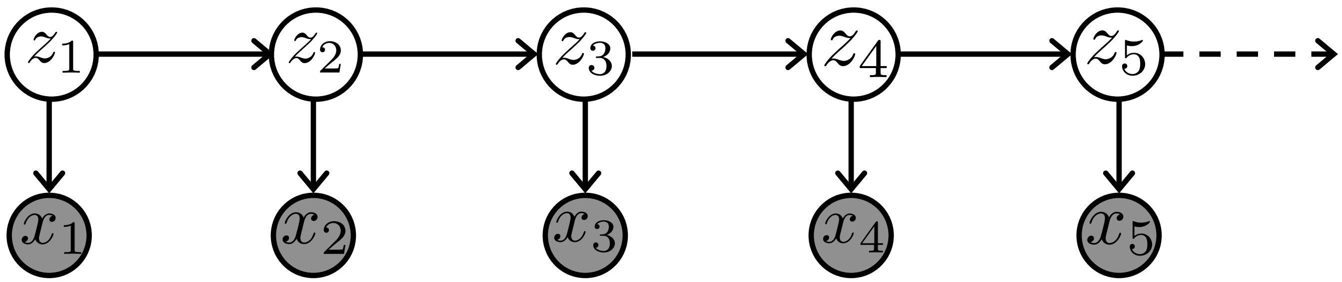
Example: Discrete Language HMM

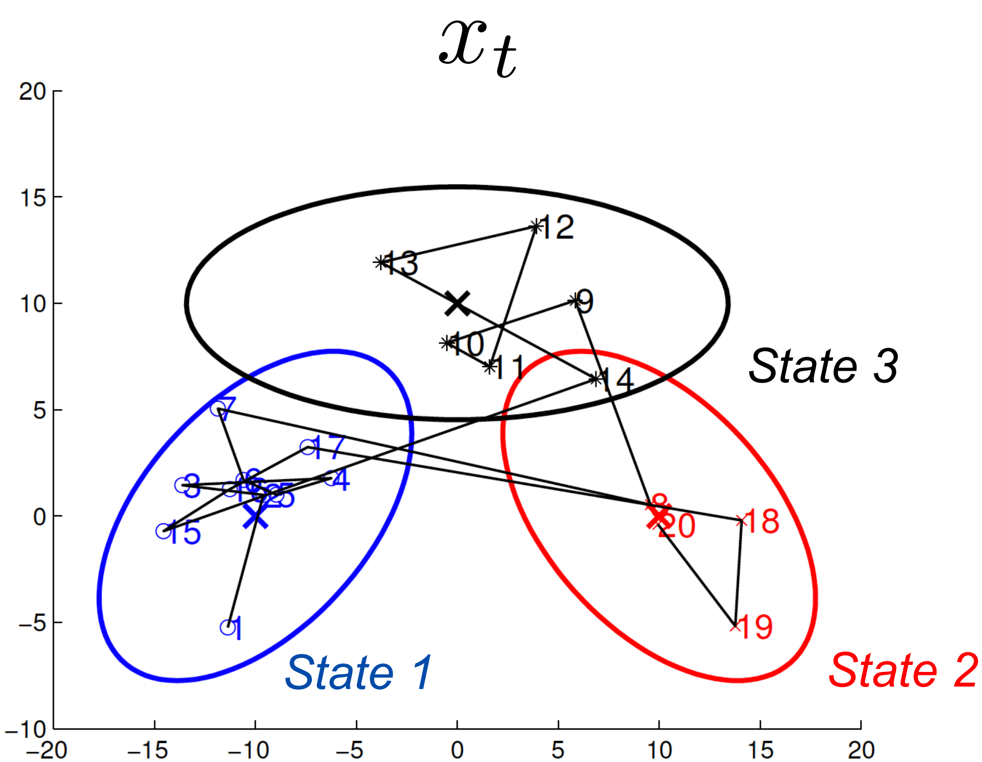


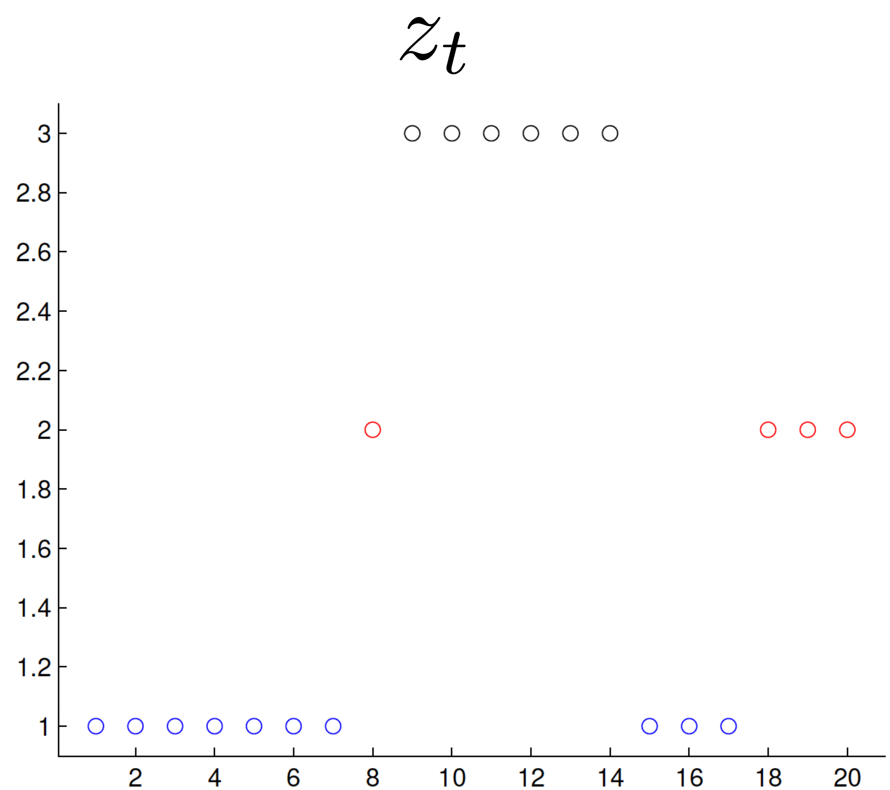
buy a pan
eat the flour
buy flour eat the flour buy pan eat a pan

M. Johnson, 2009

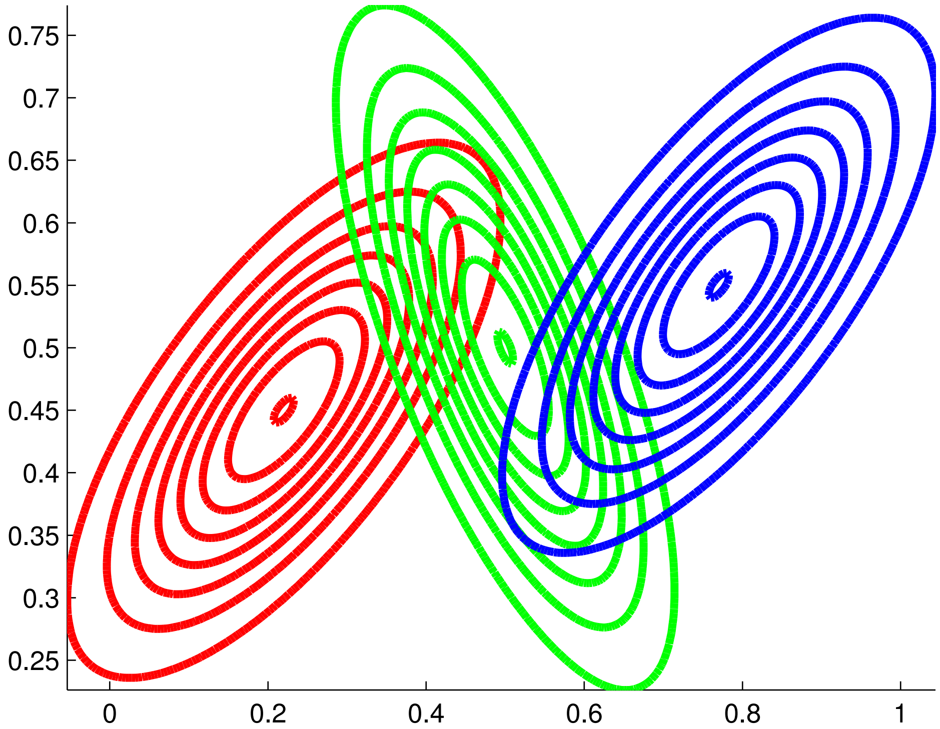
Example: 3-State Gaussian HMM

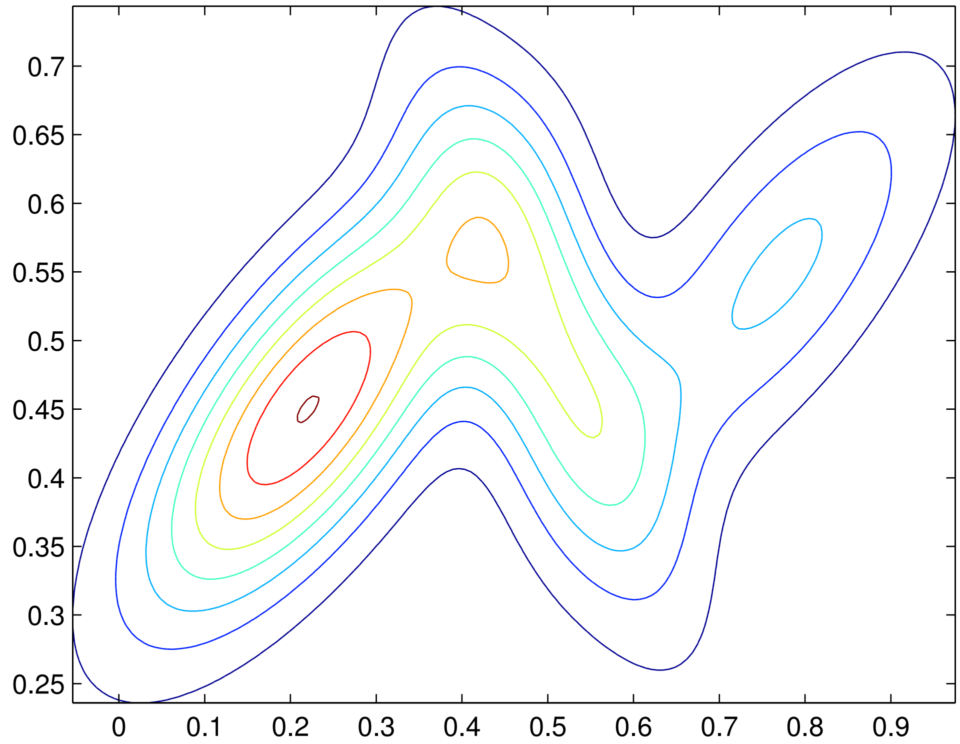






Gaussian Mixture Models

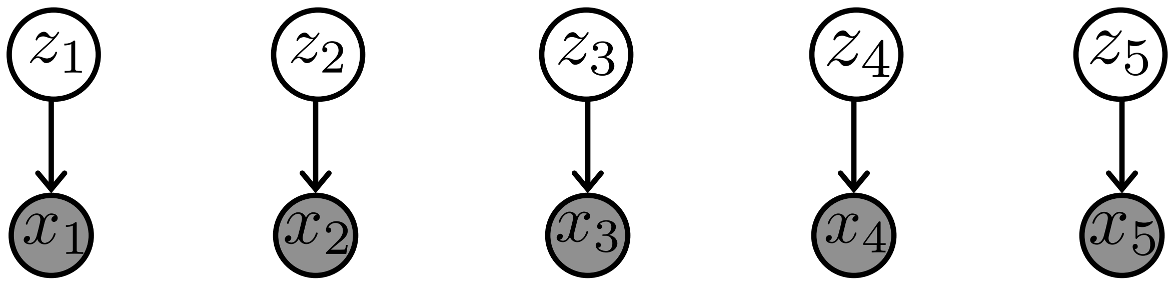




Mixture models are a special case of HMMs, in which the state transition distribution happens to not depend on the previous state, and becomes the mixture prior probability.

Gaussian Mixture Models vs. HMMs

Mixture Model

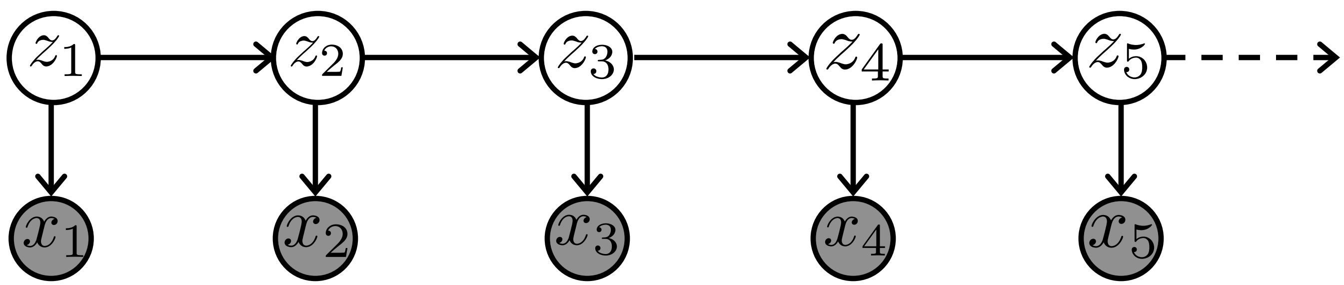


$z_i \in \{1, \ldots, K\}$

$$p(z_i \mid \pi, \mu, \Sigma) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathrm{Norm}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

Hidden Markov Model

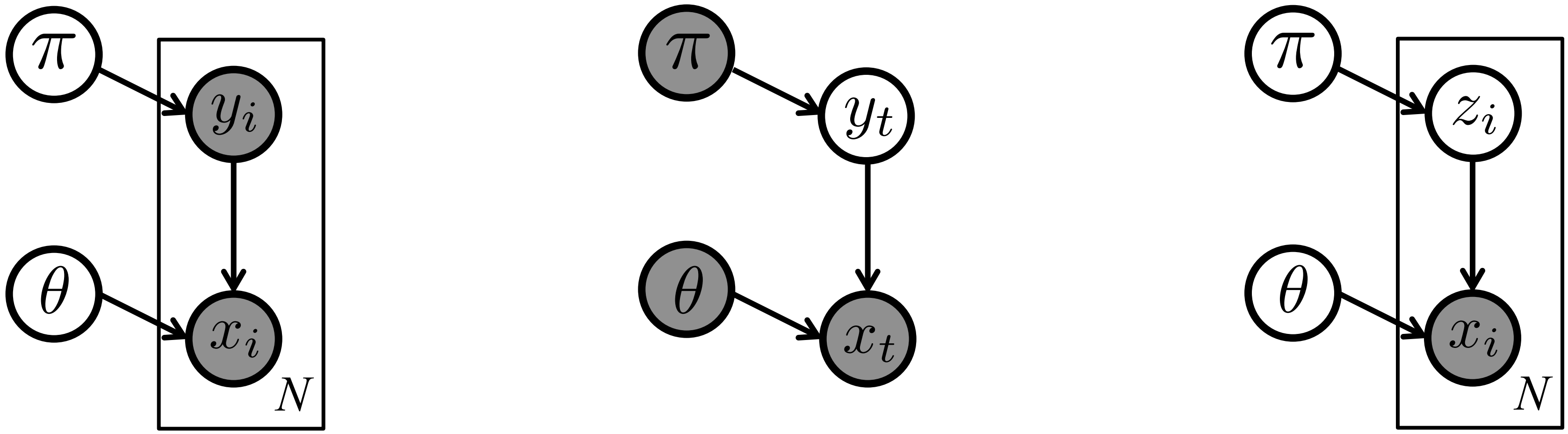


$$p(z_t \mid \pi, \mu, \Sigma, z_{t-1}, z_{t-2}, \ldots) = \mathrm{Cat}(z_t \mid \pi_{z_{t-1}})$$

$$p(x_t \mid z_t, \pi, \mu, \Sigma) = \mathrm{Norm}(x_t \mid \mu_{z_t}, \Sigma_{z_t})$$

Recover mixture model when all rows of state transition matrix are equal.

EM for Mixture Models



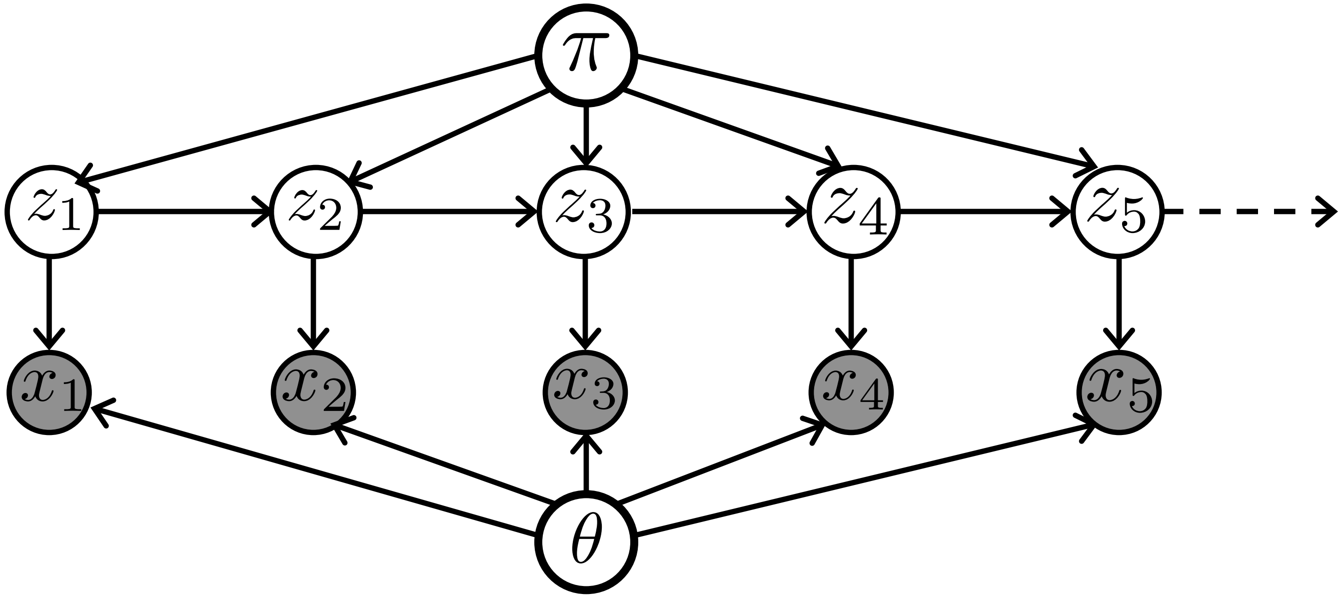
Supervised Training | *Supervised Testing* | *Unsupervised Learning*

$\pi, \theta \longrightarrow$ *parameters (define cluster location and shape)*

$z_1, \ldots, z_N \longrightarrow$ *hidden data (group observations into clusters)*

- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden data
  - Equivalent to test inference of full posterior distribution
- **M-Step:** Given posterior distributions, find likely parameters
  - Distinct from supervised ML/MAP, but often still tractable
- **Iteration:** Alternate E-step & M-step until convergence

EM for Hidden Markov Models



$\pi, \theta \longrightarrow$ *parameters (state transition & emission dist.)*

$z_1, \ldots, z_N \longrightarrow$ *hidden discrete state sequence*

- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden states
  - Dynamic programming to efficiently infer state marginals
- **M-Step:** Given posterior distributions, find likely parameters
  - Like training of mixture models and Markov chains
- **Iteration:** Alternate E-step & M-step until convergence

E-Step: Mixture Models

$$\ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta)$$

$$q^{(t)} = \arg\max_q \mathcal{L}(q, \theta^{(t-1)})$$

- General solution, for any probabilistic model:

$$q^{(t)}(z) = p(z \mid x, \theta^{(t-1)})$$

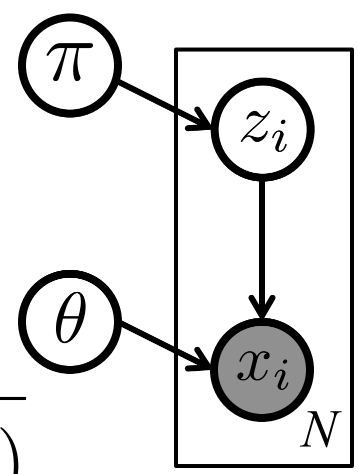
posterior distribution given current parameters

- Applying to probabilistic mixture models:

$$p(z_i \mid \pi) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \theta) = p(x_i \mid \theta_{z_i})$$

$$r_{ik} = p(z_i = k \mid x_i, \pi, \theta) = \frac{\pi_k p(x_i \mid \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell p(x_i \mid \theta_\ell)}$$



E-Step: HMMs

$$q^{(t)}(z) = p(z \mid x, \pi^{(t-1)}, \theta^{(t-1)}) \propto p(z \mid \pi^{(t-1)}) p(x \mid z, \theta^{(t-1)})$$

Mixture Models

$$q^{(t)}(z) \propto \prod_{i=1}^{N} p(z_i \mid \pi^{(t-1)}) p(x_i \mid z_i, \theta^{(t-1)})$$

- Hidden states are *conditionally independent* given parameters
- Naïve representation of full posterior has size $\mathcal{O}(KN)$

HMMs

$$q^{(t)}(z) \propto \prod_{i=1}^{N} p(z_i \mid \pi_{z_{i-1}}^{(t-1)}) p(x_i \mid z_i, \theta^{(t-1)})$$

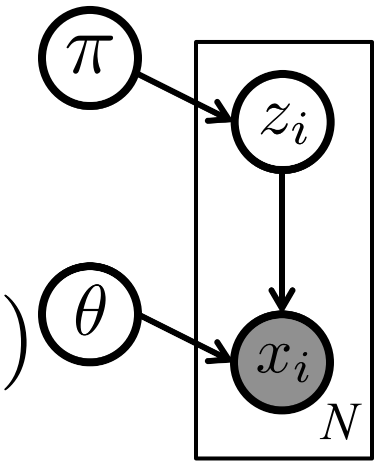
- Hidden states have *Markov dependence* given parameters
- Naïve representation of full posterior has size $\mathcal{O}(K^N)$
- Must use *dynamic programming* to compute summaries of posterior required by the M-step

M-Step: Mixture Models

$$\ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta)$$

$$\theta^{(t)} = \arg\max_\theta \mathcal{L}(q^{(t)}, \theta) = \arg\max_\theta \sum_z q(z) \ln p(x, z \mid \theta)$$

- Unlike E-step, no simplified general solution
- Applying to mixtures of *exponential families*:



$$p(z_i \mid \pi) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \theta) = \exp(\theta_{z_i}^T \phi(x_i) - A(\theta_{z_i}))$$

$$\hat{\pi}_k = \frac{N_k}{N}$$

$$\mathbb{E}_{\hat{\theta}_k}[\phi(x)] = \frac{1}{N_k} \sum_{i=1}^N r_{ik} \phi(x_i)$$

weighted moment matching

$$N_k = \sum_{i=1}^N r_{ik}$$

M-Step: HMMs

$$\theta^{(t)} = \arg\max_{\theta} \mathcal{L}(q^{(t)}, \theta) = \arg\max_{\theta} \sum_{z} q(z) \ln p(x, z \mid \theta)$$

Initial state dist. *State transition dist.*

$$\sum_{k=1}^{K} \mathbb{E}\left[N_k^1\right] \log \pi_k + \sum_{j=1}^{K} \sum_{k=1}^{K} \mathbb{E}\left[N_{jk}\right] \log A_{jk}$$

$$\hat{\pi}_k = \frac{\mathbb{E}\left[N_k^1\right]}{N}$$

$$\hat{A}_{jk} = \frac{\mathbb{E}\left[N_{jk}\right]}{\sum_{k'} \mathbb{E}\left[N_{jk'}\right]}$$

State emission dist. (observation likelihoods)

$$+ \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{k=1}^{K} p(z_t = k|\mathbf{x}_i, \boldsymbol{\theta}^{old}) \log p(\mathbf{x}_{i,t}|\boldsymbol{\phi}_k)$$

emissions via weighted moment matching

$$\mathbb{E}\left[N_k^1\right] = \sum_{i=1}^{N} p(z_{i1} = k|\mathbf{x}_i, \boldsymbol{\theta}^{old})$$

$$\mathbb{E}\left[N_{jk}\right] = \sum_{i=1}^{N} \sum_{t=2}^{T_i} p(z_{i,t-1} = j, z_{i,t} = k|\mathbf{x}_i, \boldsymbol{\theta}^{old})$$

$$\mathbb{E}\left[N_j\right] = \sum_{i=1}^{N} \sum_{t=1}^{T_i} p(z_{i,t} = j|\mathbf{x}_i, \boldsymbol{\theta}^{old})$$

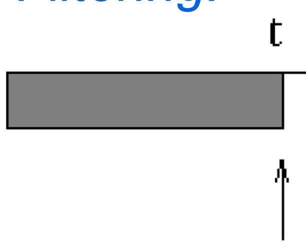
Need posterior marginal distributions of single states, and pairs of sequential states

$$p(z_t \mid x)$$

$$p(z_t, z_{t+1} \mid x)$$

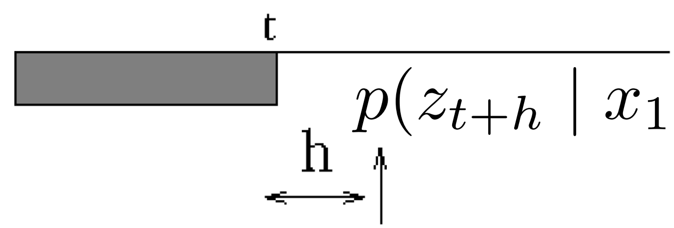
Inference in HMMs

Filtering:



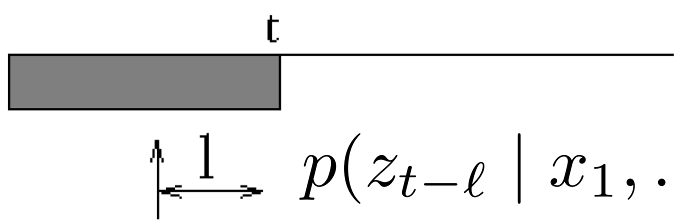
$$p(z_t \mid x_1, \ldots, x_t)$$

Prediction:



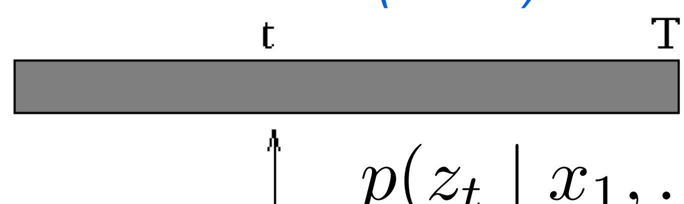
$$p(z_{t+h} \mid x_1, \ldots, x_t)$$

Fixed lag smoothing:



$$p(z_{t-\ell} \mid x_1, \ldots, x_t)$$

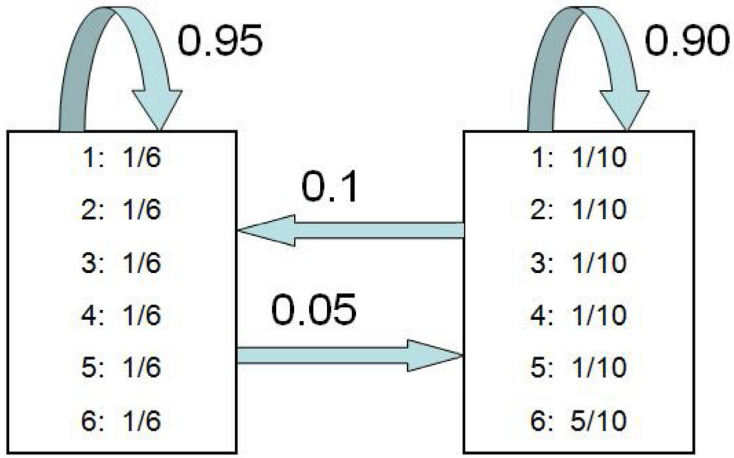
Fixed interval (batch) smoothing:



$$p(z_t \mid x_1, \ldots, x_T)$$

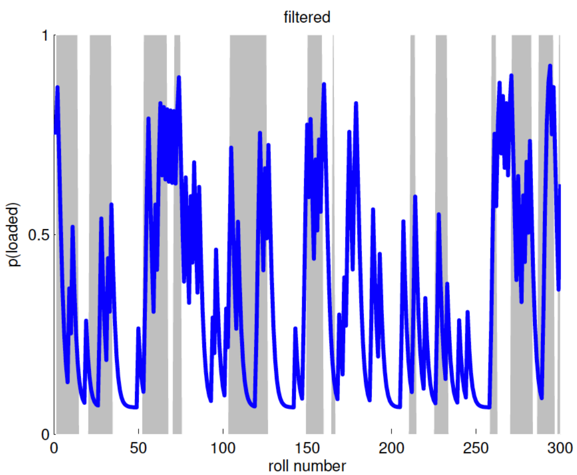
- E-step of HMM training requires *fixed interval smoothing*
- Financial or weather forecasting requires *prediction*
- Automatic speech recognition: *batch smoothing* for training, *filtering or fixed lag smoothing* for test deployment
- Computed by variants of the *forward-backward* algorithm, also known as *belief propagation* or *sum-product* algorithm

The Occasionally Dishonest Casino

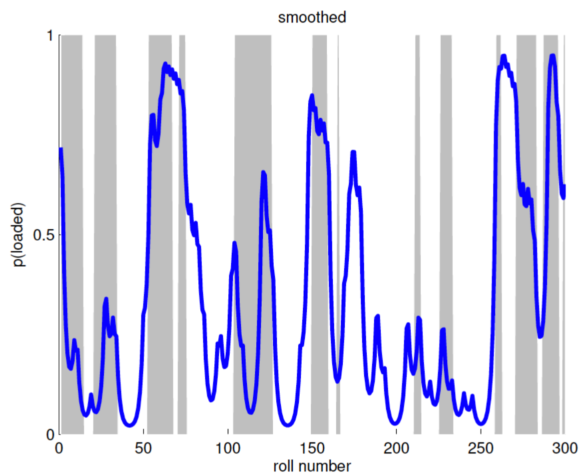


Fair Die *Loaded Die*

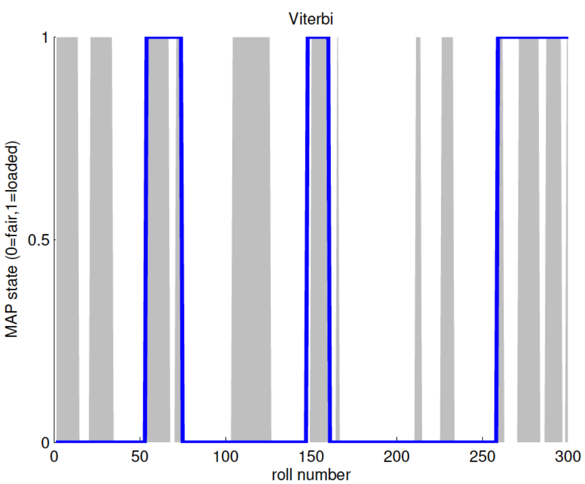
```
Rolls:    664153216162115234653214356634261655234232315142464156663246
Die:      LLLLLLLLLLLLLLFFFFFFLLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFLLLLLLLLL
```



$p(z_t \mid x_1, \ldots, x_t)$



$p(z_t \mid x_1, \ldots, x_T)$



MAP Estimate