Human Motion Analysis
Lecture 5: Dynamical Models

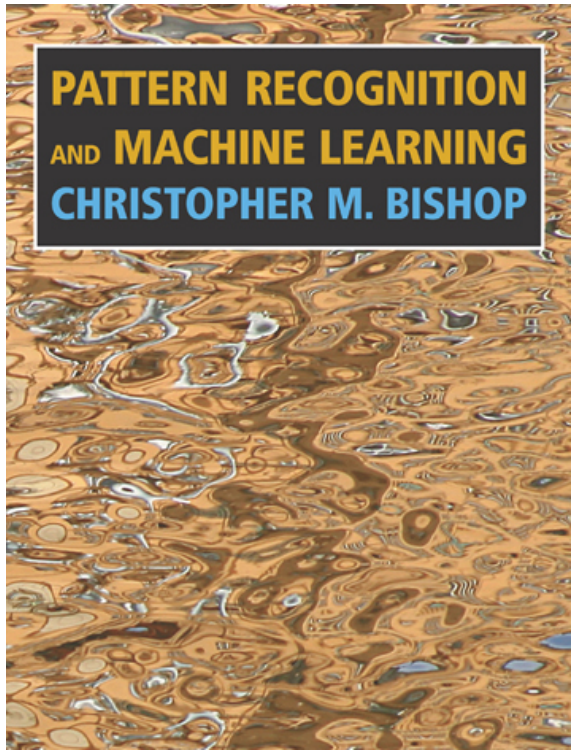
Raquel Urtasun

TTI Chicago

March 22, 2010

Materials used for this lecture

This lecture is based chapter 13 of C. Bishop book "Pattern Recognition and Machine Learning".



I recommend this book if you want to buy a book with a Bayesian view of pattern recognition and machine learning.

Contents of today's lecture?

We will look into the most popular dynamical models

- Introduction on Markov models
- Hidden Markov Models (HMMs)
- Linear Dynamical Systems (LDS)

Notation

$\mathbf{X}$ — the set of observations

$\mathbf{Z}$ — the set of hidden variables

N— number of data points

K — number of possible values for a discrete variable

Dealing with time series data

- Before we have assume that the data points are i.i.d, independent and identically distributed

$$p(\mathbf{X}) = p(\mathbf{x}_1, \cdots, \mathbf{x}_n) = \prod_{i=1}^{N} p(\mathbf{x}_i)$$

- This is a poor assumption to describe sequential data.
- **Stationary** sequential distribution arrises when the data evolves in time, but the distribution from which is generated remains the same.
- **Non-stationary** sequential distribution arrises when the data and the distribution from which is generated changes.
- We focus on the stationary case.

The Markov assuption

- We would like to predict the next value of a time series given observations of the previous values.
- It is a reasonable assumption to expect that recent observations are likely to be more informative than older observations when predicting the future.
- The complexity of considering all previous observations will grow unbounded as the system evolves, since the number of observations increases!
- This leads us to consider **Markov models** that are independent of all but the most recent observations.

Markov models we will see today

- Although these models are tractable, they are also very limited.
- We will see today how by incorporating hidden states, this model can be quite flexible while still tractable, leading to **state space models**.
- In particular we will focus on:
  - **hidden Markov models** (HMMs) that have discrete latent variables,
  - and in **linear dynamical systems**, in which the latent variables are Gaussian.

Markov models

- The easiest way to treat data is to assume i.i.d

$$p(\mathbf{X}) = p(\mathbf{x}_1, \cdots, \mathbf{x}_n) = \prod_{i=1}^{N} p(\mathbf{x}_i)$$

- This will fail to exploit sequential patterns of the data, such as observations close in time

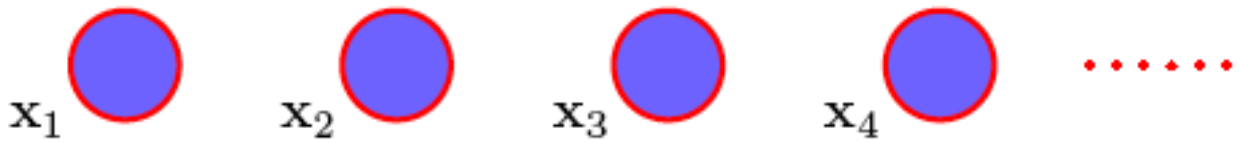


Figure: i.i.d. assumption (Bishop, Springer 2007)

Example of correlations: speech

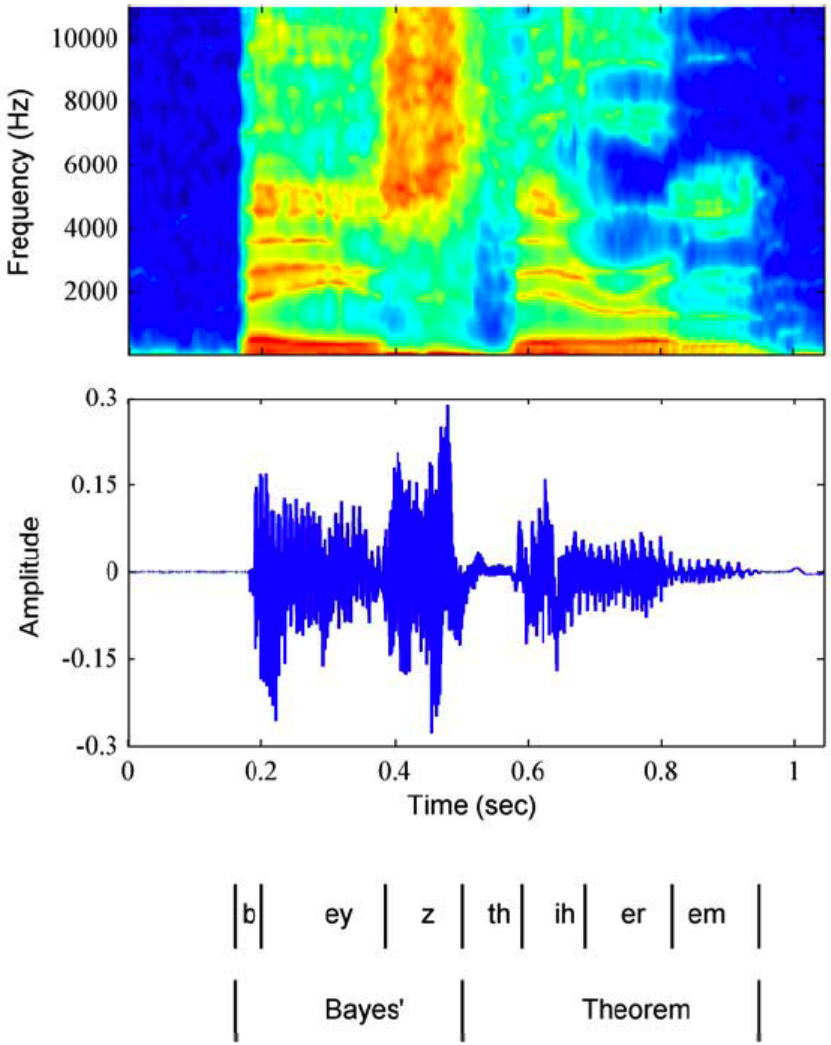


Figure: Temporal correlations in speech (Bishop, Springer 2007)

First order Markov model

- We can use the product rule to write without loss of generality

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \cdots, \mathbf{x}_{n-1})$$

First order Markov model

- We can use the product rule to write without loss of generality

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \cdots, \mathbf{x}_{n-1})$$

- The **First-order Markov model** assumes that an observation is independent of all but the last observation

$$p(\mathbf{x}_n | \mathbf{x}_{n-1}, \cdots, \mathbf{x}_1) = p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

so that

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{i=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

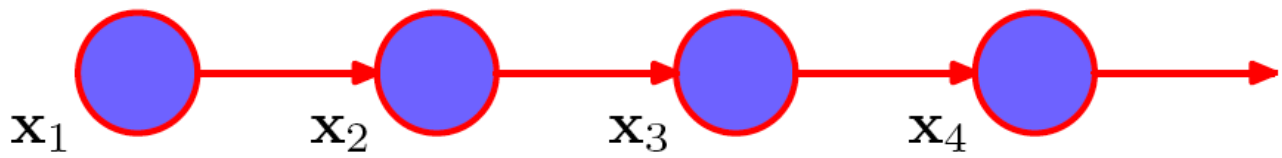


Figure: First-order Markov model (Bishop, Springer 2007)

First order Markov model

- We can use the product rule to write without loss of generality

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \cdots, \mathbf{x}_{n-1})$$

- The **First-order Markov model** assumes that an observation is independent of all but the last observation

$$p(\mathbf{x}_n | \mathbf{x}_{n-1}, \cdots, \mathbf{x}_1) = p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

  so that

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{i=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

- In most applications, $p(\mathbf{x}_n | \mathbf{x}_{n-1})$ is constraint to be equal (stationary time series). This is then called **homogeneous Markov chain**.
- For example, if $p(\mathbf{x}_n | \mathbf{x}_{n-1})$ depends on some parameters, then the parameters will be fixed for all the series.

Second-order Markov model

- Although more flexible than i.i.d., the first-order Markov model is still quite restrictive.
- One way to incorporate more complex behaviors is by using higher-order Markov chains.
- A second order Markov chain

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)\prod_{i=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2})$$

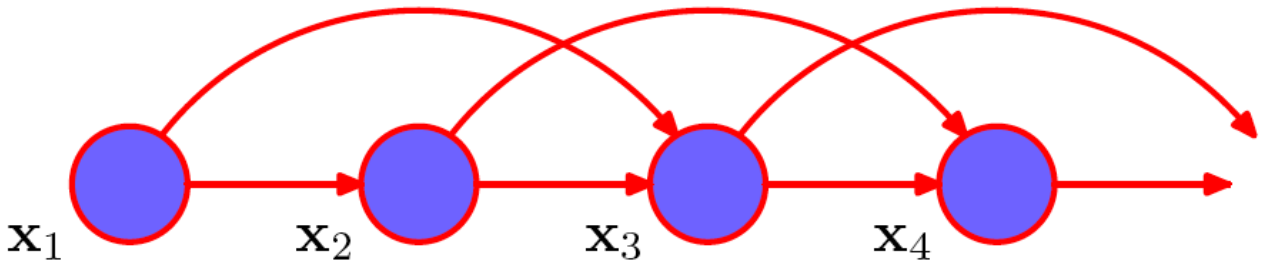


Figure: Second-order Markov model (Bishop, Springer 2007)

Higher-order Markov model

- This can be easily extended to M-th order Markov models, where the conditional depends on M variables.
- However, the number of parameters is much larger. Supposing the observations are discrete variables with K possible values, then
  - The number of parameters for the M-th order is $K^{M-1}(K-1)$
  - For a first-order Markov model is only $K(K-1)$. We have $(K-1)$ since they are probabilities, so they have to sum up to 1.
- The number of parameters grows exponentially with the order of the Markov chain!

Continuous variables

- For continuous variables, we can use linear-Gaussian conditional distributions in which each node has a Gaussian distribution whose mean is a linear function of its parents. This is **autoregressive** or **AR model**
- An alternative approach is to use a parametric model for $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \cdots, \mathbf{x}_{n-M})$, such as a neural network. This is called **Tapped delay line**.
- The number of parameters can then be much smaller than in a completely general model.
- But this can model only a restricted family of conditional distributions

Introducing latent variables

- A way to relax the Markov assumption and allow a richer type of models is to introduce latent variables
- For each observation $\mathbf{x}_n$, we introduce a corresponding latent variable $\mathbf{z}_n$ of arbitrary dimension.
- We can then assume that the latent variables form a Markov chain

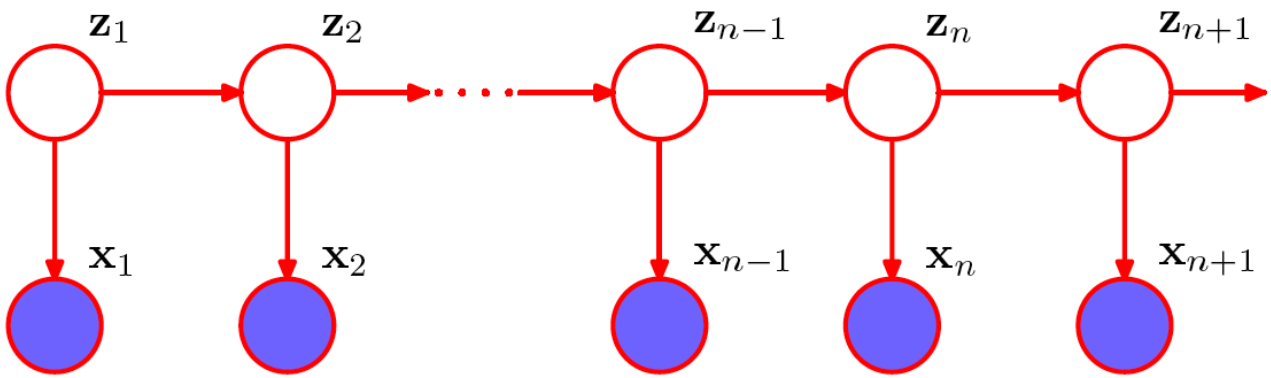


Figure: State space model (Bishop, Springer 2007)

- This satisfies the conditional independence property of

$$\mathbf{z}_{n+1} \perp\!\!\!\perp \mathbf{z}_{n-1} | \mathbf{z}_n$$

State space models

- The joint distribution of a **state space model** is

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_1, \cdots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$$

- This joint distribution can be easily seen from the following graphical model

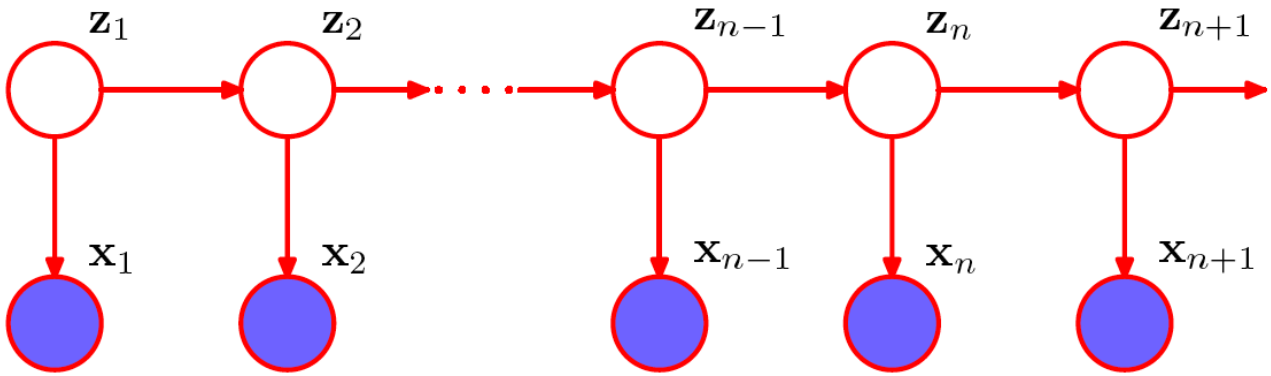


Figure: State space model (Bishop, Springer 2007)

- The observations are not independent anymore, they follow a path over $\mathbf{z}$'s.
- The observations do NOT satisfy the Markov property.

Hidden Markov Models (HMMs) I

- The hidden Markov model is a specific instance of the state space model in which the latent variables are discrete
- For each time step, it corresponds to mixture distribution, with component densities given by $p(\mathbf{x}|\mathbf{z})$.
- So is an extension of mixture models where each observation is not selected independently but depends on the choice of component for the previous observation.
- The latent variables are the discrete multinomial variables $\mathbf{z}_n$ describing which component of the mixture is responsible for generating the corresponding observation $\mathbf{x}_n$.

Hidden Markov Models (HMMs) II

- We now allow the probability distribution of $\mathbf{z}_n$ to depend on the state of the previous latent variable $\mathbf{z}_{n-1}$ through a conditional distribution $p(\mathbf{z}_n|\mathbf{z}_{n-1})$.
- This conditional distribution corresponds to a table of numbers that we denote by $\mathbf{A}$, the elements of which are known as transition probabilities given by

$$A_{j,k} \equiv p(\mathbf{z}_{n,k} = 1|\mathbf{z}_{n-1,j} = 1),$$

since they are probabilities they should satisfy

$$0 \leq A_{j,k} \leq 1 \quad \text{and} \quad \forall j, \quad \sum_{k=1}^{K} A_{j,k} = 1$$

- The matrix $\mathbf{A}$ has $K(K-1)$ independent parameters.

Hidden Markov Models (HMMs) III

- We can write the conditional distribution as

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}) = \prod_{k=1}^{K}\prod_{j=1}^{K} A_{j,k}^{z_{n-1,j}z_{n,k}}$$

- The initial latent node $\mathbf{z}_1$ has no parent, so it has a marginal distribution

$$p(\mathbf{z}_1|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{z_{1,k}}$$

with $\boldsymbol{\pi}$ the vector of probabilities such that $\pi_k \equiv p(z_{1,k} = 1)$, and $\sum_k \pi_k = 1$,

Transition matrix

The transition matrix is usually illustrated as

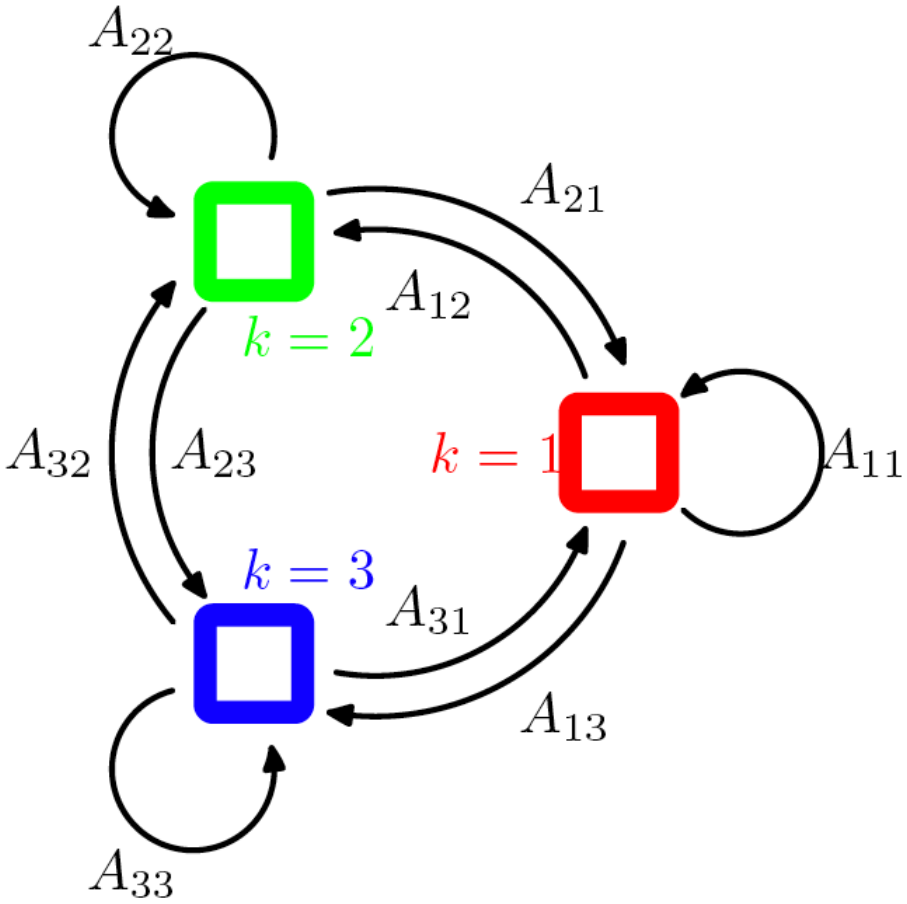


Figure: Transition matrix for the case of $K = 3$. Note that this is NOT a graphical model. (Bishop, Springer 2007)

Transition diagram

The transition diagram is usually illustrated as

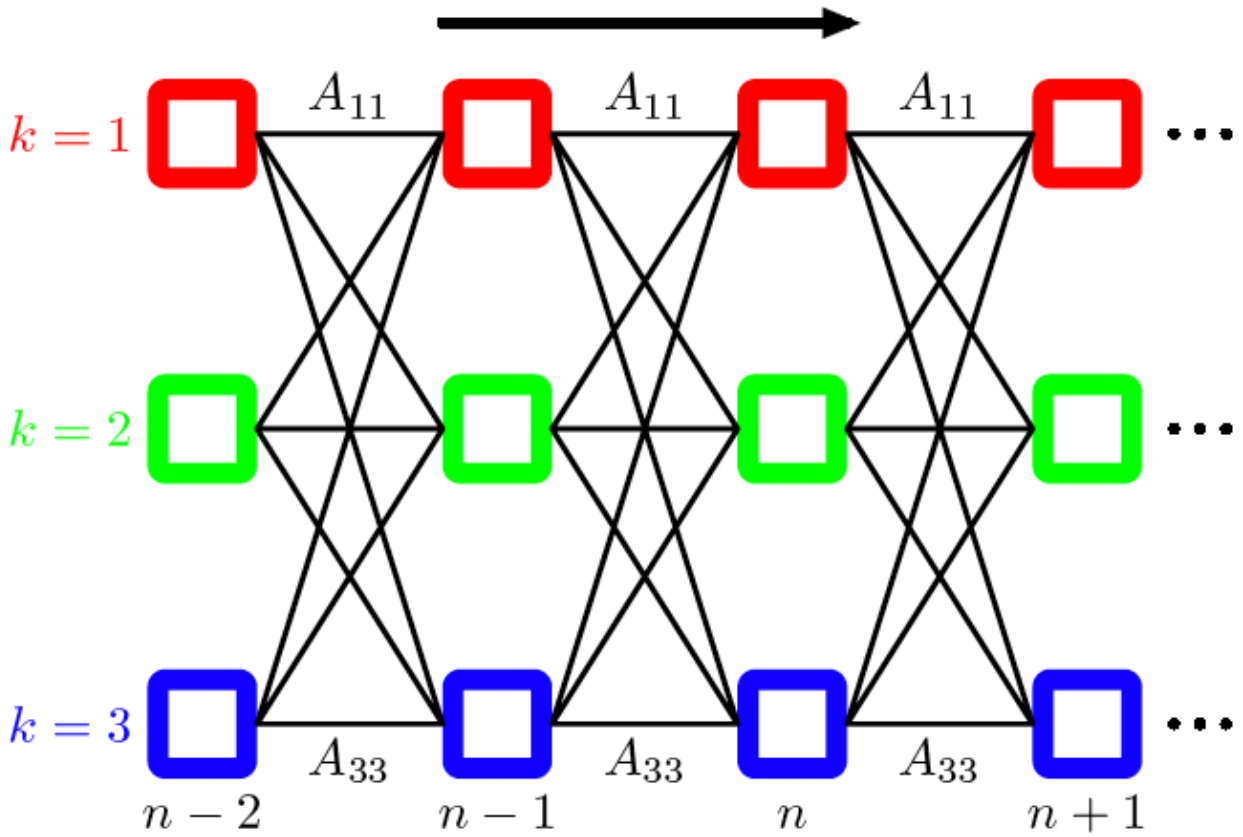


Figure: Transition diagram for the case of $K = 3$. Note that this is NOT a graphical model.(Bishop, Springer 2007)

Emission probabilities

- We need to specify the conditional distributions of the observed variables $p(\mathbf{x}_n|\mathbf{z}_n, \phi)$, with ϕ a set of parameters that govern the distribution.
- These are known as the **emission probabilities**.
- Because $\mathbf{x}_n$ is observed, they consists for a given value of ϕ, of a vector corresponding to the K possible states of the binary vector $\mathbf{z}_n$.

$$p(\mathbf{x}_n|\mathbf{z}_n, \phi) = \prod_{k=1}^{K} p(\mathbf{x}_n|\phi_k)^{z_{n,k}}$$

- We will look only into homogeneous models that share the same transition and emission probabilities.

Hidden Markov Models (HMMs) IV

- Given training data $\mathbf{X}$, the joint probability is then

$$p(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{z}_1|\pi) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}) \prod_{m=1}^{K} p(\mathbf{x}_m|\mathbf{z}_m, \phi)$$

  with θ the parameters of the model.
- This is tractable for a set of distributions, including discrete variables, Gaussians, mixture of Gaussians, discriminative methods, etc.

Sampling HMMs

Sample $\mathbf{z}_1 \sim p(\mathbf{z}_1)$
Sample $\mathbf{x}_1 \sim p(\mathbf{x}_1|\mathbf{z}_1)$
for $n = 2$ to N **do**
  Sample $\mathbf{z}_n \sim p(\mathbf{z}_n|\mathbf{z}_{n-1})$
  Sample $\mathbf{x}_n \sim p(\mathbf{x}_n|\mathbf{z}_n)$
end for

Sampling HMMs

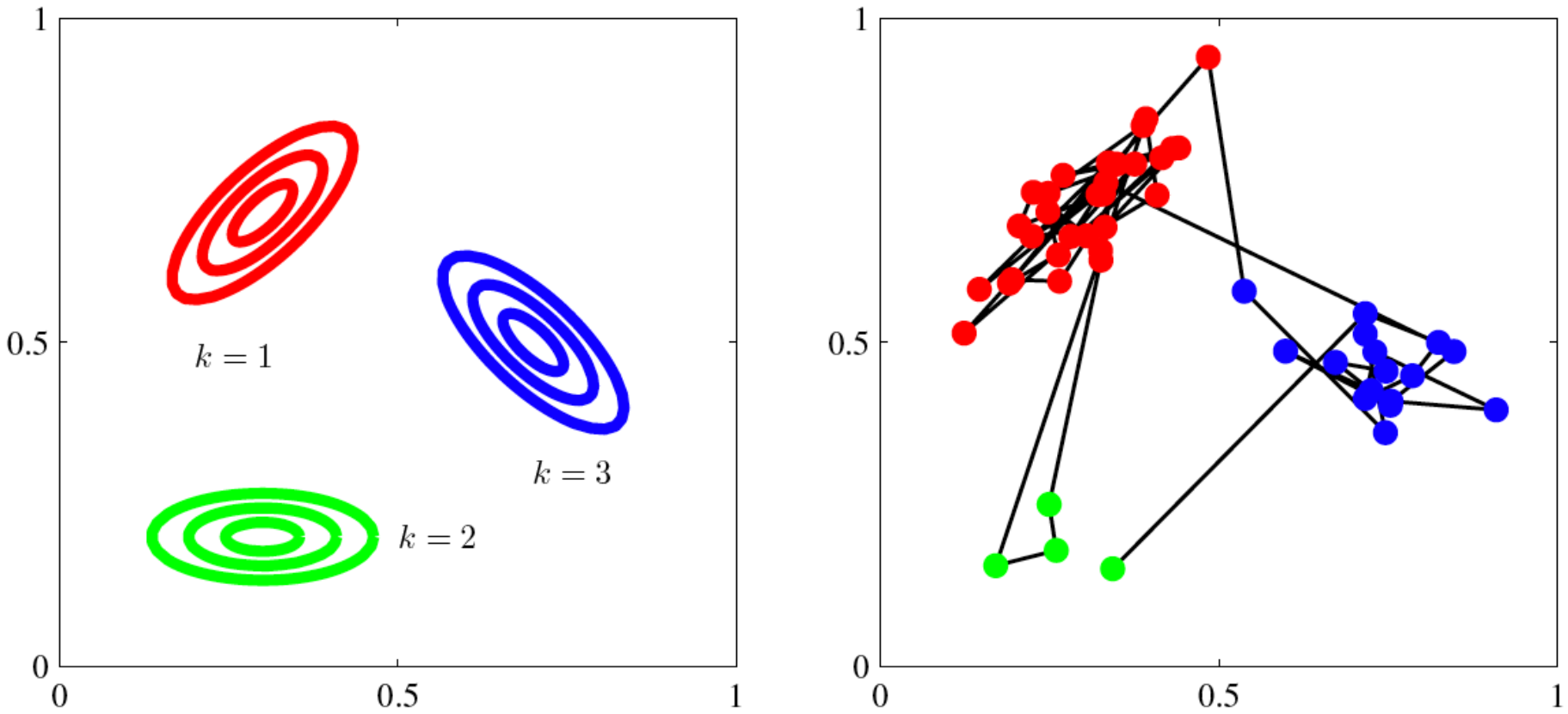


Figure: Ancestral sampling for generating samples from an HMM (Bishop, Springer 2007)

HMM variants

- **Left-to-right** HMM: typically set $A_{j,k} = 0$ if $k < j$, with $p(z_{11}) = 1$.

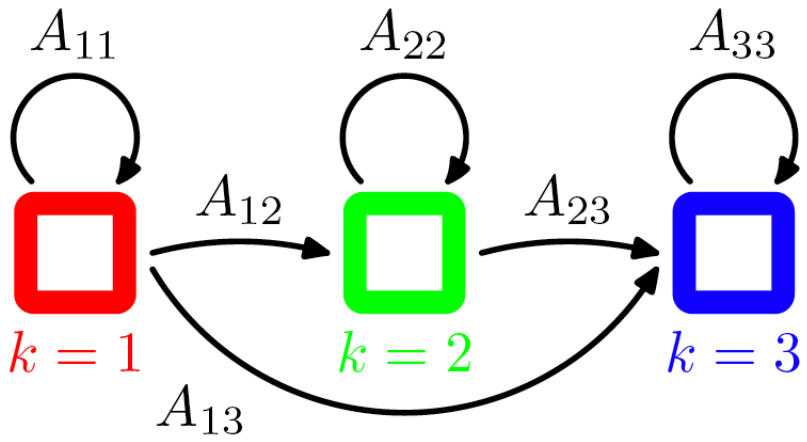


Figure: Left-to-right HMM (Bishop, Springer 2007)

HMM variants

- **Left-to-right** HMM: typically set $A_{j,k} = 0$ if $k < j$, with $p(z_{11}) = 1$.
- The transition matrix can be further constrain

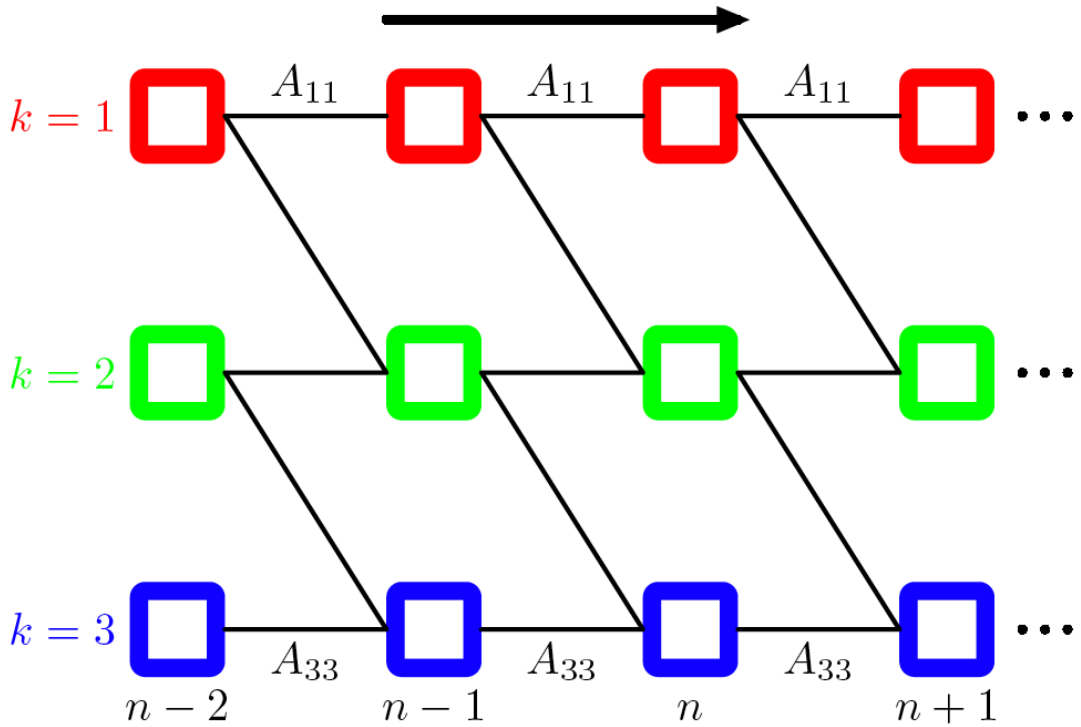


Figure: Further constrained left-to-right HMM (Bishop, Springer 2007)

HMM variants

- **Left-to-right** HMM: typically set $A_{j,k} = 0$ if $k < j$, with $p(z_{11}) = 1$.
- The transition matrix can be further constrain

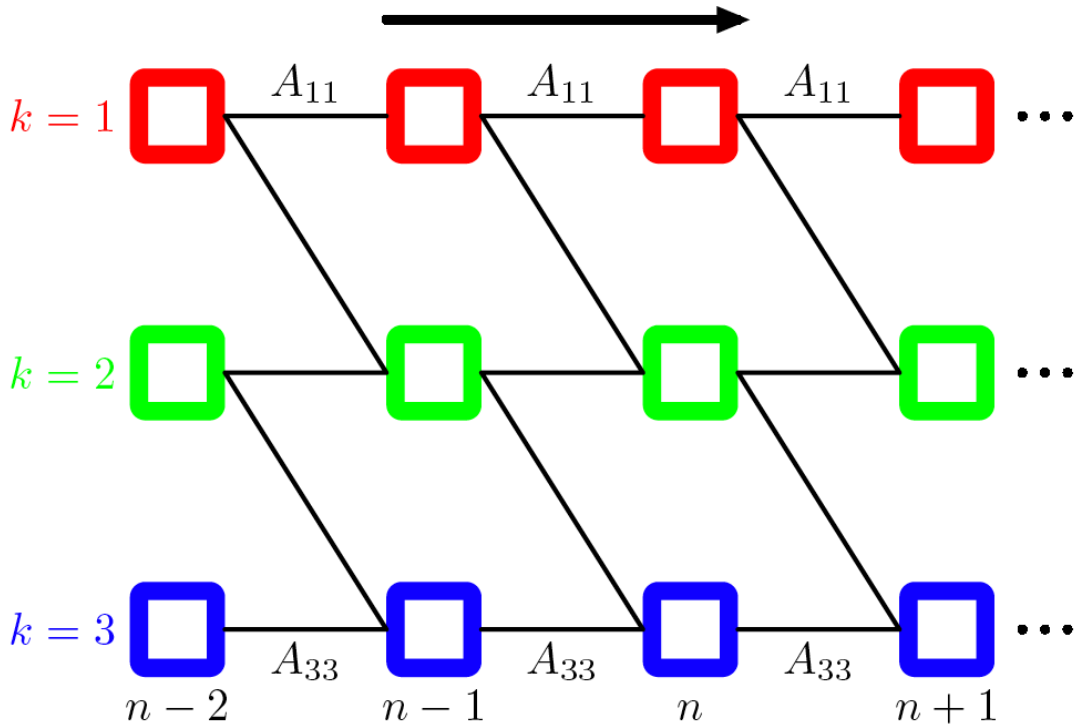


Figure: Further constrained left-to-right HMM (Bishop, Springer 2007)

- With what type of motions would you use these models?

Learning HMMs via Maximum likelihood

- Given training data $\mathbf{X}$, we can determine the parameter of an HMM via maximum likelihood

$$\theta^* = arg\max \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

  where the latent variables have been marginalized. Since they are discrete we have a sum instead of an integral.

- The naive summation sums over K^N tems as the number of operations grows exponentially with the length of the chain.
- **Expectation-maximization (EM)** is typically used to learn the HMM.

Expectation-Maximization (EM)

At iteration i:

- E-step: Find the posterior distribution over the latent variables $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(\mathbf{i})})$ with fixed parameters $\boldsymbol{\theta}^{(\mathbf{i})}$.
- M-step: Maximize the expectation of the logarithm of the complete data likelihood with respect to the parameters

$$\boldsymbol{\theta}^{(i+1)} = arg\max \underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}_{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})}$$

- Let's look at the M-step, assuming we know $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(\mathbf{i})})$.

Closer look to the M-step I

- Let's define $\gamma(\mathbf{z}_n)$ the marginal posterior distribution of $\mathbf{z}_n$, and $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ the joint posterior probability of two consecutive latent variables

$$\begin{aligned} \gamma(\mathbf{z}_n) &= p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \end{aligned}$$

  $\gamma(\mathbf{z}_n)$ is a table of K non-negative values, and $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ a table of $K \times K$ non-negative numbers.

- Let's also define $\gamma(z_{n,k})$ the probability of $z_{n,k} = 1$, and $\xi(z_{n-1,j}, z_{n,k})$ the probability of the transition for j to k symbol occurs at time n

$$\gamma(z_{n,k}) = E[z_{n,k}] = \sum_{\mathbf{z}} \gamma(\mathbf{z}) z_{n,k}$$

$$\xi(z_{n-1,j}, z_{n,k}) = E[z_{n-1,j}, z_{n,k}] = \sum_{\mathbf{z}} \xi(z_{n-1,j}, z_{n,k}) z_{n-1,j} z_{n,k}$$

Closer look to the M-step I

- Let's define $\gamma(\mathbf{z}_n)$ the marginal posterior distribution of $\mathbf{z}_n$, and $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ the joint posterior probability of two consecutive latent variables

$$\begin{aligned} \gamma(\mathbf{z}_n) &= p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \end{aligned}$$

  $\gamma(\mathbf{z}_n)$ is a table of K non-negative values, and $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ a table of $K \times K$ non-negative numbers.

- Let's also define $\gamma(z_{n,k})$ the probability of $z_{n,k} = 1$, and $\xi(z_{n-1,j}, z_{n,k})$ the probability of the transition for j to k symbol occurs at time n

$$\begin{aligned} \gamma(z_{n,k}) &= E[z_{n,k}] = \sum_{\mathbf{z}} \gamma(\mathbf{z}) z_{n,k} \\ \xi(z_{n-1,j}, z_{n,k}) &= E[z_{n-1,j}, z_{n,k}] = \sum_{\mathbf{z}} \xi(z_{n-1,j}, z_{n,k}) z_{n-1,j} z_{n,k} \end{aligned}$$

Closer look to the M-step II

- We can now redefine $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$ as

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) &= \sum_{k=1}^{K} \gamma(z_{n,k}) \log \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{n,k}) \log A_{j,k} \\ &\quad + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \log p(\mathbf{x}_n | \phi_k) \end{aligned}$$

  where $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$.

- In the E-step, we compute $\gamma(z_{n,k})$ and $\xi(z_{n-1,j}, z_{n,k})$.

Closer look to the M-step II

- We can now redefine $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$ as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = \sum_{k=1}^{K} \gamma(z_{n,k}) \log \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{n,k}) \log A_{j,k}$$
$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \log p(\mathbf{x}_n | \phi_k)$$

  where $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$.

- In the E-step, we compute $\gamma(z_{n,k})$ and $\xi(z_{n-1,j}, z_{n,k})$.
- In the M-step we maximize with respect to θ. This yields close form for

$$\pi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^{K} \gamma(z_{1,j})} \qquad A_{j,k} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,l})}$$

Closer look to the M-step II

- We can now redefine $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$ as

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) &= \sum_{k=1}^{K} \gamma(z_{n,k}) \log \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{n,k}) \log A_{j,k} \\ &\quad + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{n,k}) \log p(\mathbf{x}_n | \phi_k) \end{aligned}$$

  where $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$.

- In the E-step, we compute $\gamma(z_{n,k})$ and $\xi(z_{n-1,j}, z_{n,k})$.
- In the M-step we maximize with respect to $\boldsymbol{\theta}$. This yields close form for

$$\pi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^{K} \gamma(z_{1,j})} \qquad A_{j,k} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,l})}$$

Closer look to the M-step III

- Maximizing with respect to ϕ depends on the distribution assumed for $p(\mathbf{x}|\phi_k)$.
- The objective decouples in a sum of each ϕ_k, so they can be maximized independently.
- If the emission densities are Gaussian, $p(\mathbf{x}|\phi_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{n,k})\mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{n,k})} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{n,k})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^{N} \gamma(z_{n,k})}$$

- If the emission densities are discrete multinomial then

$$p(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{D}\prod_{k=1}^{K} \mu_{i,k}^{x_i z_k} \qquad \mu_{i,k} = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) x_{n,i}}{\sum_{n=1}^{N} \gamma(z_{n,k})}$$

with $p(\mathbf{x}|\mathbf{z})$ the conditional distribution of the observations.

Expectation-Maximization (EM)

At iteration i:

- E-step: Find the posterior distribution over the latent variables $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(\mathbf{i})})$ with fixed parameters $\boldsymbol{\theta}^{(\mathbf{i})}$.
- M-step: Maximize the expectation of the logarithm of the complete data likelihood with respect to the parameters

$$\boldsymbol{\theta}^{(i+1)} = arg\max \underbrace{\sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(i)}) \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}_{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})}$$

- Let's look at the E-step.

Closer look to the E-step I

- The **forward-backward algorithm** or **Baum-Welch algorithm** is an efficient way to evaluate $\gamma(z_{n,k})$ and $\xi(z_{n-1,j}, z_{n,k})$ were we use the fact that the graphical model of an HMM is a tree.
- Note that the posterior distributions of the latent variables is independent of the form of the emission density $p(\mathbf{x}|\mathbf{z})$.
- All we require is the values of the quantities $p(\mathbf{x}_n|\mathbf{z}_n)$ for each value of $\mathbf{z}_n$ for every n. This can be pre-computed and stored.

Closer look to the E-step II

- Recall that $\gamma(z_{n,k})$ is the probability of having the k-th component be 1. Using Bayes rule we can compute

$$\gamma(z_{n,k}) = p(\mathbf{z}_n|\mathbf{x}_1, \cdots, \mathbf{x}_N) = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

  where we have defined

$$\alpha(\mathbf{z}_n) \equiv p(\mathbf{x}_1, \cdots, \mathbf{x}_n, \mathbf{z}_n) \qquad \beta(\mathbf{z}_n) \equiv p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N|\mathbf{z}_n)$$

- $\alpha(\mathbf{z}_n)$ and $\beta(\mathbf{z}_n)$ are represented as a table of K numbers.

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\ &= p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)\end{aligned}$$

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\ &= p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n) \\ &= p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}|\mathbf{z}_n)p(\mathbf{z}_n)\end{aligned}$$

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \cdots, \mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1} | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n)
\end{aligned}$$

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \cdots, \mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1} | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1})
\end{aligned}$$

Closer look to the E-step III

- We can work out a recursion rule

$$
\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \cdots, \mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1} | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1})
\end{aligned}
$$

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}|\mathbf{z}_n)p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1}) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})
\end{aligned}$$

- The initial condition is given by

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) = \prod_{k=1}^{K}[\pi_k p(\mathbf{x}_1|\phi_k)]^{z_{1,k}}$$

- The computational complexity of the recursion is $\mathcal{O}(K^2N)$.

Closer look to the E-step III

- We can work out a recursion rule

$$\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \cdots, \mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}|\mathbf{z}_n)p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1}) \\
&= p(\mathbf{x}_n|\mathbf{z}_n)\sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})
\end{aligned}$$

- The initial condition is given by

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) = \prod_{k=1}^{K}[\pi_k p(\mathbf{x}_1|\phi_k)]^{z_{1,k}}$$

- The computational complexity of the recursion is $\mathcal{O}(K^2N)$.

Forward recursion

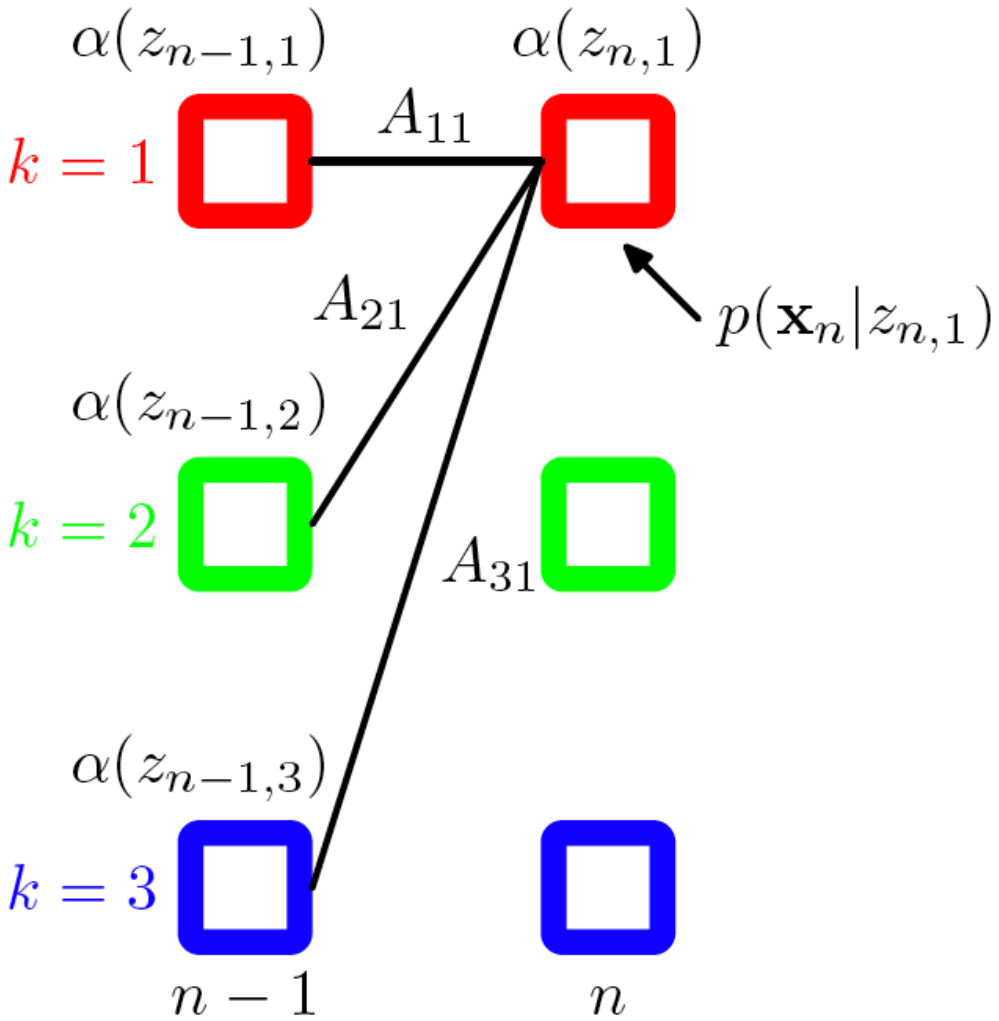


Figure: Illustration of the forward recursion for $K = 3$ (Bishop, Springer 2007)

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$\begin{aligned}\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n)\end{aligned}$$

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$\begin{aligned}\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N|\mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1}|\mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)\end{aligned}$$

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$\begin{aligned}
\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n)
\end{aligned}$$

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$
\begin{aligned}
\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n)
\end{aligned}
$$

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$\begin{aligned}
\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n)
\end{aligned}$$

- This is a backward message passing algorithm that evaluates $\beta(\mathbf{z}_n)$ in terms of $\beta(\mathbf{z}_{n+1})$
- The starting condition is $\beta(\mathbf{z}_N) = 1$ for all settings of $\mathbf{z}_N$.

Closer look to the E-step IV

- We can similarly work out a recursion rule

$$\begin{aligned}
\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2}, \cdots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n)
\end{aligned}$$

- This is a backward message passing algorithm that evaluates $\beta(\mathbf{z}_n)$ in terms of $\beta(\mathbf{z}_{n+1})$
- The starting condition is $\beta(\mathbf{z}_N) = 1$ for all settings of $\mathbf{z}_N$.

Backward recursion

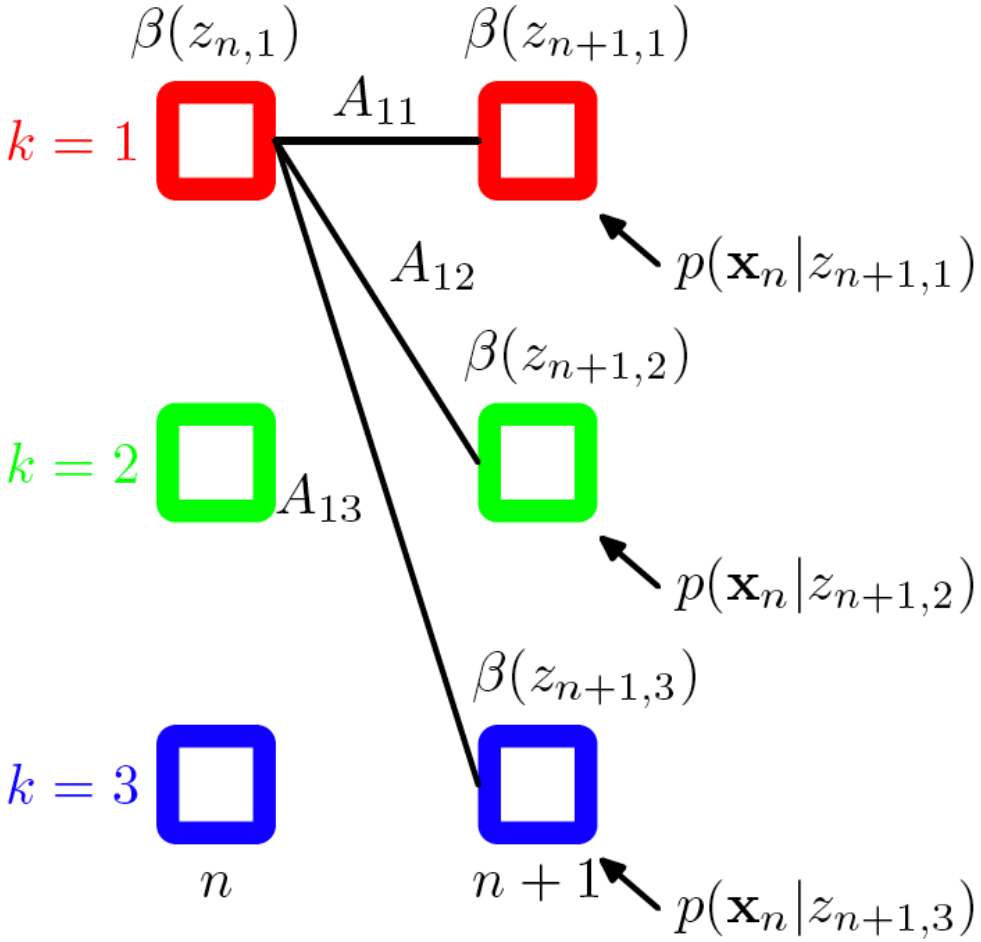


Figure: Illustration of the backward recursion for $K = 3$ (Bishop, Springer 2007)

Closer look to the E-step V

- We can use these quantities to compute the partition function

$$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)$$

- Recall that

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1, \cdots, \mathbf{x}_N) = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

- Next we consider the evaluation of $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ which can be written as

$$\begin{aligned} \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}) \\ &= \frac{\alpha(\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{z}_{n-1})\beta(\mathbf{z}_n)}{p(\mathbf{X})} \end{aligned}$$

- I have omitted the derivations, see (Bishop, chapter 13).

Closer look to the E-step V

- We can use these quantities to compute the partition function

$$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)$$

- Recall that

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1, \cdots, \mathbf{x}_N) = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

- Next we consider the evaluation of $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ which can be written as

$$\begin{aligned} \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}) \\ &= \frac{\alpha(\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{z}_{n-1})\beta(\mathbf{z}_n)}{p(\mathbf{X})} \end{aligned}$$

- I have omitted the derivations, see (Bishop, chapter 13).

EM for learning Gaussian-distribuited HMMs

E-STEP:

Forward pass:

$\alpha(\mathbf{z}_1) = \prod_{k=1}^{K} [\pi_k p(\mathbf{x}_1|\phi_k)]^{z_{1,k}}$

for $n = 2$ to N **do**

$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$

end for

Backward pass:

$\beta(\mathbf{z}_1) = 1$ for all settings of $\mathbf{z}_n$

for $n = N - 1$ to 1 **do**

$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)$

end for

$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n) \beta_(\mathbf{z}_n)$

for $n = 1$ to N **do**

$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1, \cdots, \mathbf{x}_N) = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$

end for

for $n = 2$ to N **do**

$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})}$

end for

M-STEP:

$$\pi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^{K} \gamma(z_{1,j})} \text{ and } A_{j,k} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,l})}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{n,k})} \text{ and } \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^{N} \gamma(z_{n,k})}$$

EM for learning Multinomial-distribuited HMMs

E-STEP:

Forward pass:

$\alpha(\mathbf{z}_1) = \prod_{k=1}^{K} [\pi_k p(\mathbf{x}_1|\phi_k)]^{z_{1,k}}$

for $n = 2$ to N **do**

$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$

end for

Backward pass:

$\beta(\mathbf{z}_1) = 1$ for all settings of $\mathbf{z}_n$

for $n = N - 1$ to 1 **do**

$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)$

end for

$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n)\beta_(\mathbf{z}_n)$

for $n = 1$ to N **do**

$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1, \cdots, \mathbf{x}_N) = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$

end for

for $n = 2$ to N **do**

$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})}$

end for

M-STEP:

$\pi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^{K} \gamma(z_{1,j})}$ and $A_{j,k} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{n,l})}$

$\mu_{i,k} = \frac{\sum_{n=1}^{N} \gamma(z_{n,k}) x_{n,i}}{\sum_{n=1}^{N} \gamma(z_{n,k})}$

Computing the predictive distribution

- We have observed $\mathbf{X} \equiv [\mathbf{x}_1, \cdots, \mathbf{x}_N]$, and we want to predict $\mathbf{x}_{N+1}$

$$\begin{aligned} p(\mathbf{x}_{N+1}|\mathbf{X}) &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}, \mathbf{z}_{N+1}|\mathbf{X}) \\ &= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N)\alpha(\mathbf{z}_N) \end{aligned}$$

- I have omitted the derivations, see (Bishop, chapter 13).
- The recursion can be computed by running the forward α recursion, and then computing the summations over $\mathbf{z}_{N+1}$ and $\mathbf{z}_N$.
- This is required for example to generate motions from an HMM.
- In practice, we work with a scaling version of α for all the algorithms

$$\hat{\alpha}(\mathbf{z}_n) = \frac{\alpha(\mathbf{z}_n)}{p(\mathbf{x}_1, \cdots, \mathbf{x}_n)}$$

Computing the predictive distribution

- We have observed $\mathbf{X} \equiv [\mathbf{x}_1, \cdots, \mathbf{x}_N]$, and we want to predict $\mathbf{x}_{N+1}$

$$\begin{aligned} p(\mathbf{x}_{N+1}|\mathbf{X}) &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}, \mathbf{z}_{N+1}|\mathbf{X}) \\ &= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N)\alpha(\mathbf{z}_N) \end{aligned}$$

- I have omitted the derivations, see (Bishop, chapter 13).
- The recursion can be computed by running the forward α recursion, and then computing the summations over $\mathbf{z}_{N+1}$ and $\mathbf{z}_N$.
- This is required for example to generate motions from an HMM.
- In practice, we work with a scaling version of α for all the algorithms

$$\hat{\alpha}(\mathbf{z}_n) = \frac{\alpha(\mathbf{z}_n)}{p(\mathbf{x}_1, \cdots, \mathbf{x}_n)}$$

Computing the most probable sequence of hidden states

- This can be solved efficiently with the **Vitterbi** algorithm because the HMM is a tree, by computing the recursion

$$\begin{aligned}\omega(\mathbf{z}_{n+1}) &= \max_{\mathbf{z}_1,\cdots,\mathbf{z}_n} p(\mathbf{x}_1,\cdots,\mathbf{x}_n,\mathbf{z}_1,\cdots,\mathbf{z}_n) \\ &= \log p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) + \max_{\mathbf{z}_n}\{\log p(\mathbf{z}_{n+1}|\mathbf{z}_n) + \omega(\mathbf{z}_n)\}\end{aligned}$$

  with the initial value

$$\omega(\mathbf{z}_1) = \log p(\mathbf{z}_1) + \log p(\mathbf{x}_1|\mathbf{z}_1).$$

- Once we compute the value of the joint distribution for the most probable path $\mathbf{X}, \mathbf{Z}$, we back-track to obtain the path of latent variables. This is done by just keeping track of the maximum at every step k_n^{max}.
- This can be also used to compute the best D paths.
- The computational cost only grows linearly with the length of the chain.

Computing the most probable sequence of hidden states

- This can be solved efficiently with the **Vitterbi** algorithm because the HMM is a tree, by computing the recursion

$$\begin{aligned}\omega(\mathbf{z}_{n+1}) &= \max_{\mathbf{z}_1,\cdots,\mathbf{z}_n} p(\mathbf{x}_1,\cdots,\mathbf{x}_n,\mathbf{z}_1,\cdots,\mathbf{z}_n) \\ &= \log p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) + \max_{\mathbf{z}_n}\{\log p(\mathbf{z}_{n+1}|\mathbf{z}_n) + \omega(\mathbf{z}_n)\}\end{aligned}$$

  with the initial value

$$\omega(\mathbf{z}_1) = \log p(\mathbf{z}_1) + \log p(\mathbf{x}_1|\mathbf{z}_1).$$

- Once we compute the value of the joint distribution for the most probable path $\mathbf{X}, \mathbf{Z}$, we back-track to obtain the path of latent variables. This is done by just keeping track of the maximum at every step k_n^{max}.
- This can be also used to compute the best D paths.
- The computational cost only grows linearly with the length of the chain.

Computing the most probable sequence of hidden states

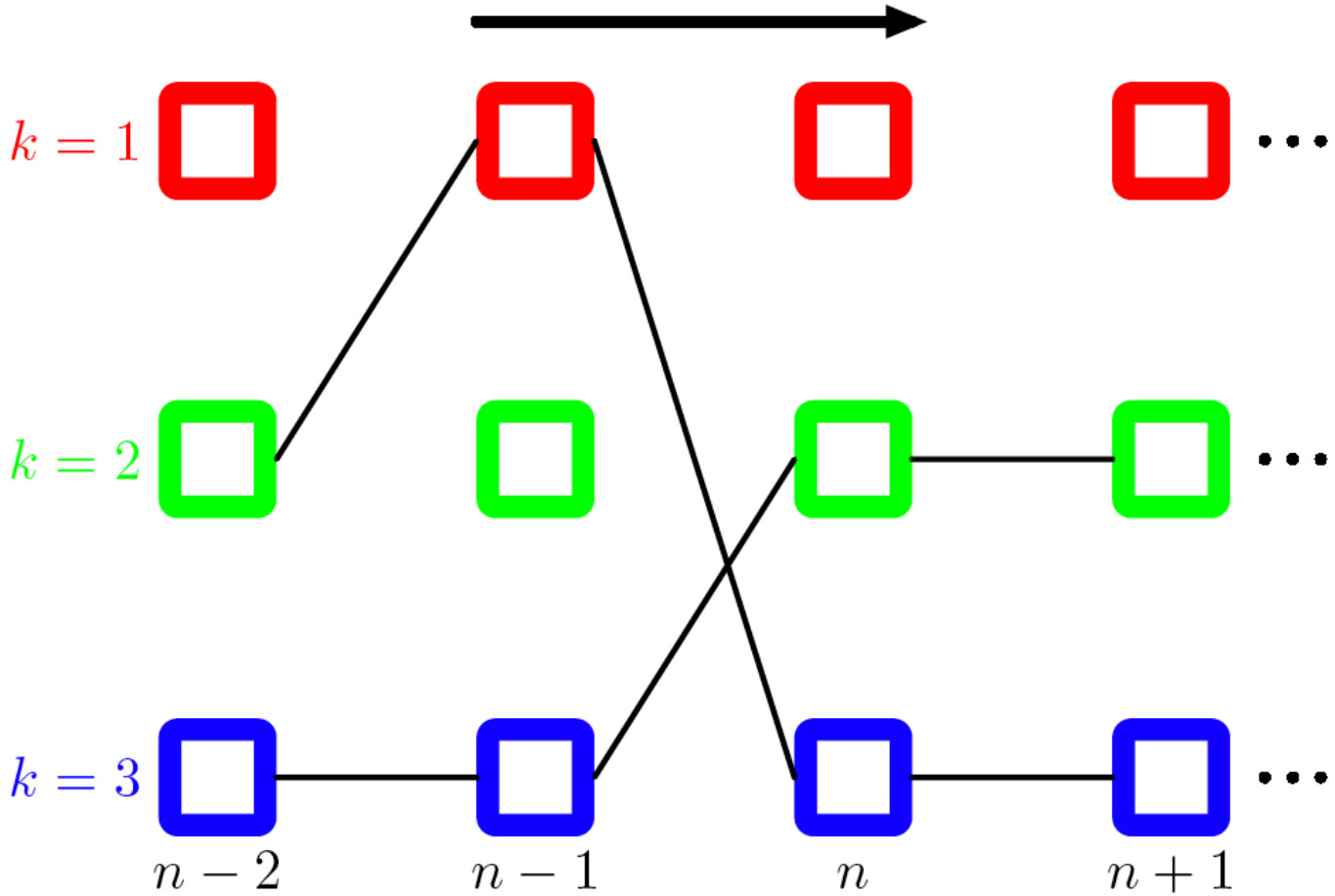


Figure: Illustration of the Vitterbi algorithm for $K = 3$ (Bishop, Springer 2007)

HMM

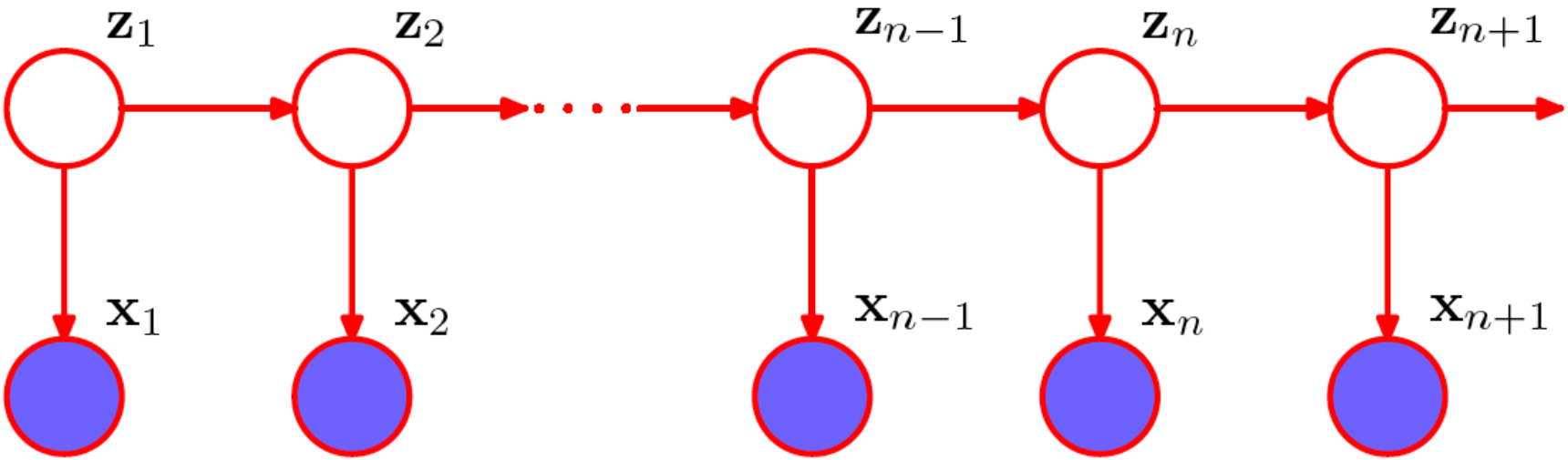


Figure: Hidden Markov model (Bishop, Springer 2007)

Extensions of HMMs: autoregresive models

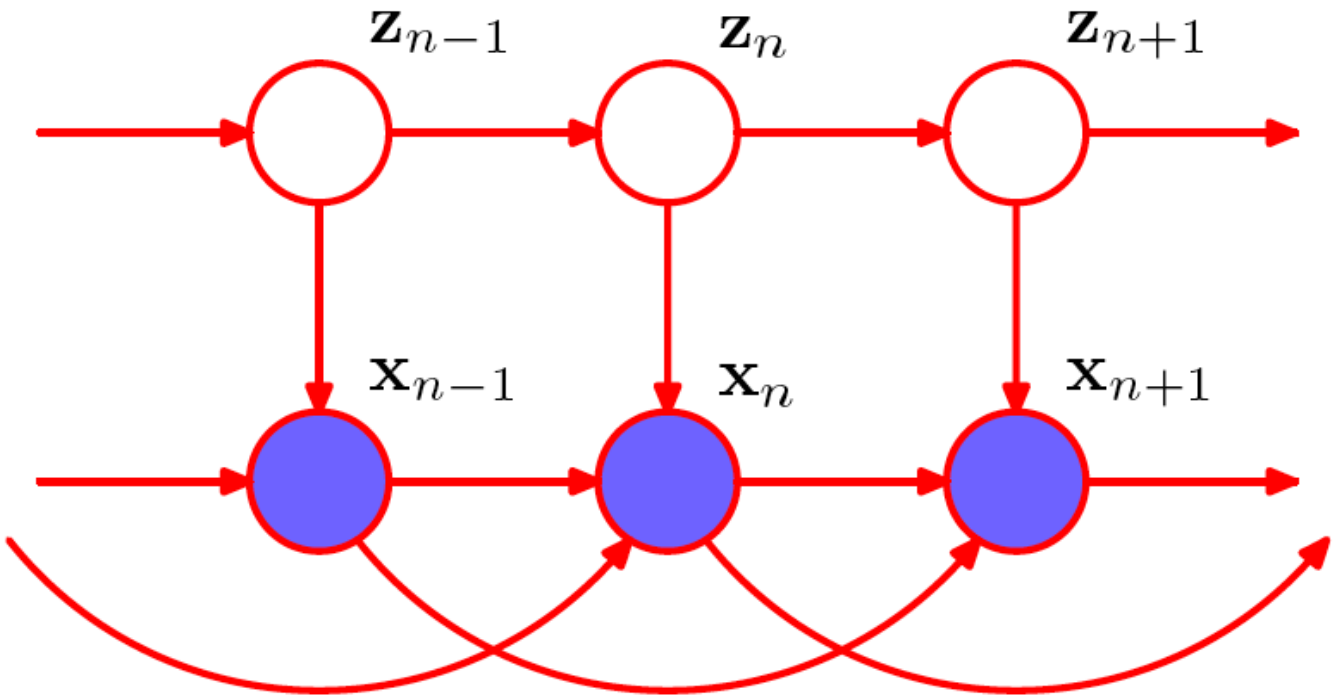


Figure: Autoregresive hidden Markov model (Bishop, Springer 2007)

Extensions of HMMs: input-outputs models

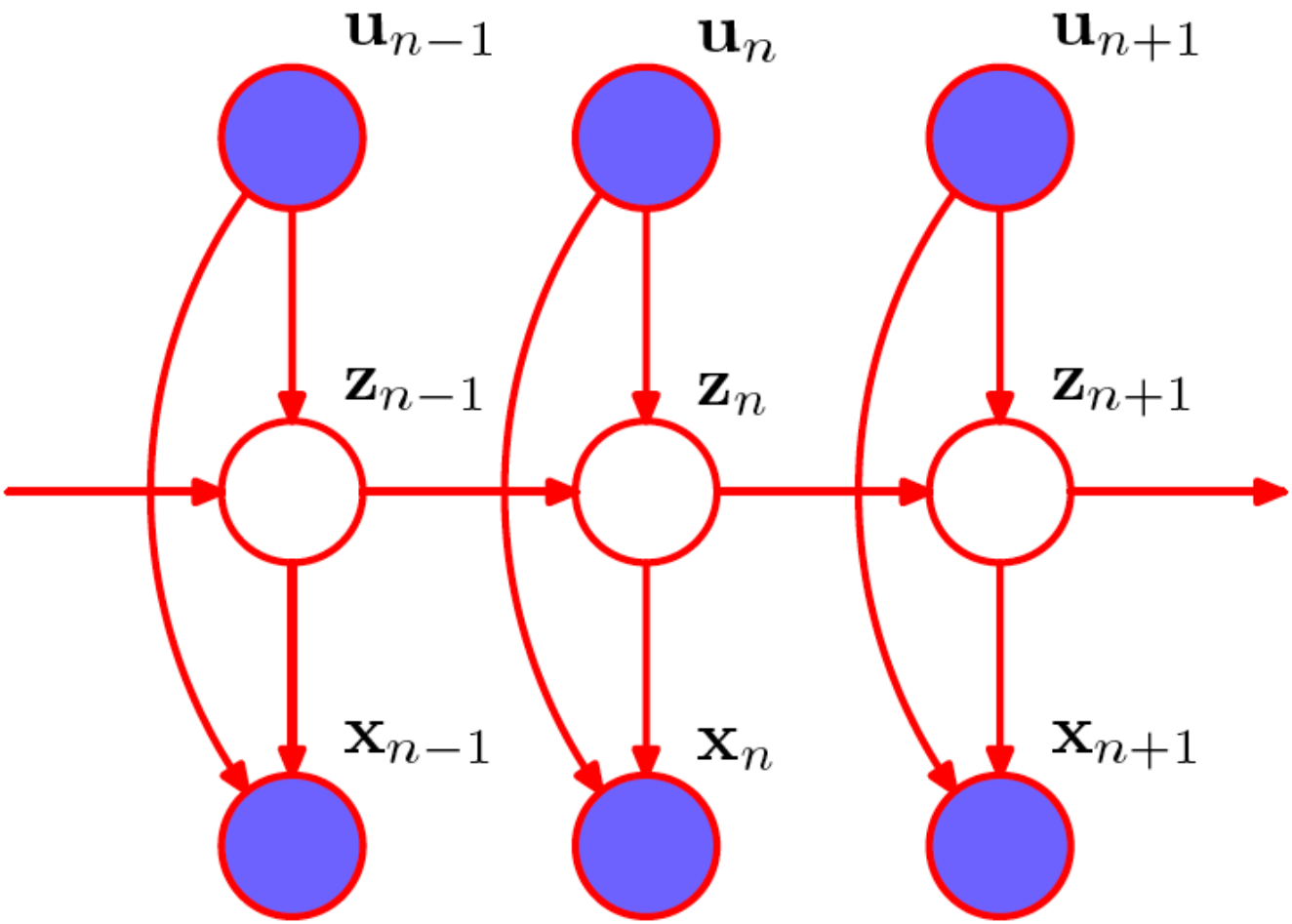


Figure: Autoregresive hidden Markov model (Bishop, Springer 2007)

Extensions of HMMs: factorial models

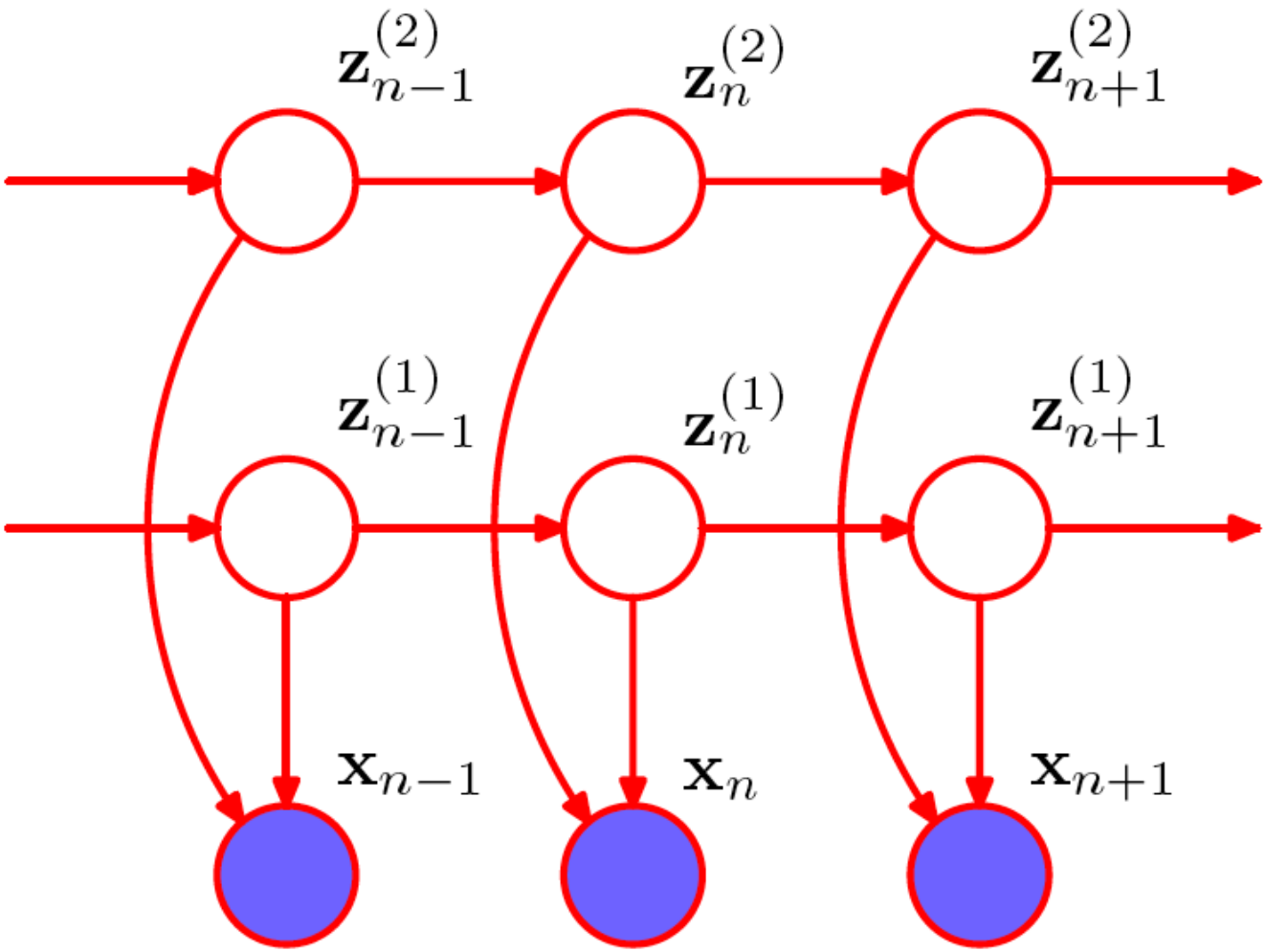


Figure: Autoregresive hidden Markov model (Bishop, Springer 2007)

Motivation of linear dynamical systems (LDS)

- Suppose we wish to measure the value of an unknown quantity $\mathbf{z}$ using a noisy measurement $\mathbf{x}$ such that

$$\mathbf{z} = \mathbf{x} + \eta$$

  with η some zero-mean noise.
- Given a single measurement, the best is to assume that $\mathbf{z} = \mathbf{x}$.
- We can improve our estimate with multiple measurements, and $\mathbf{z} = \frac{1}{N}\sum_i \mathbf{x}_i$.
- Now assume $\mathbf{z}$ changes over time. Given $\mathbf{x}_1, \cdots, \mathbf{x}_N$ we wish to obtain $\mathbf{z}_1, \cdots, \mathbf{z}_N$.
- A better solution than averaging is to take into account only a few recent measurements $\mathbf{x}_{N-L}, \cdots, \mathbf{x}_N$.
- If the $\mathbf{z}$ varies fast, then take L very small, otherwise, bigger values of L.
- We could do even better by doing a weighted average, but how?

Motivation of linear dynamical systems (LDS)

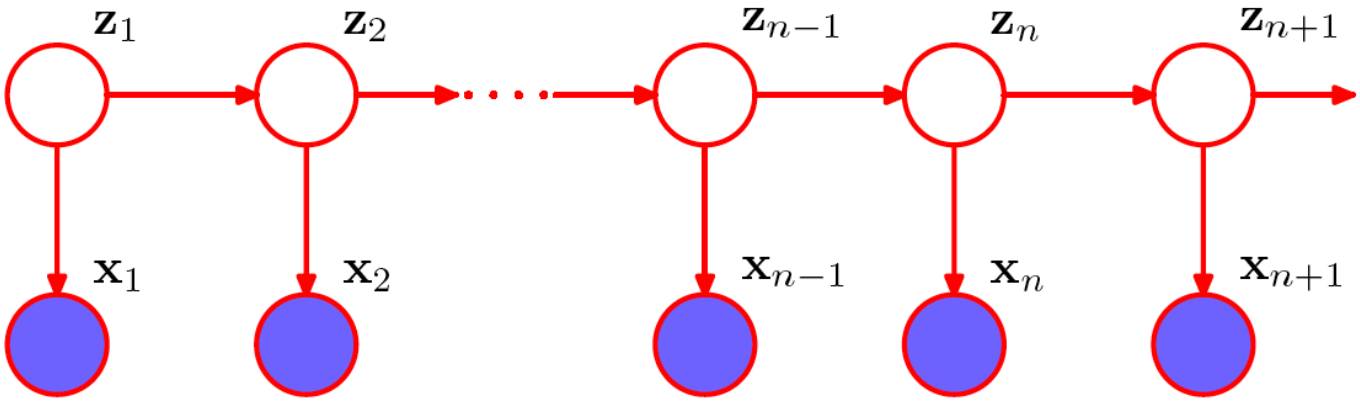
- Suppose we wish to measure the value of an unknown quantity $\mathbf{z}$ using a noisy measurement $\mathbf{x}$ such that

$$\mathbf{z} = \mathbf{x} + \eta$$

  with η some zero-mean noise.
- Given a single measurement, the best is to assume that $\mathbf{z} = \mathbf{x}$.
- We can improve our estimate with multiple measurements, and $\mathbf{z} = \frac{1}{N}\sum_i \mathbf{x}_i$.
- Now assume $\mathbf{z}$ changes over time. Given $\mathbf{x}_1, \cdots, \mathbf{x}_N$ we wish to obtain $\mathbf{z}_1, \cdots, \mathbf{z}_N$.
- A better solution than averaging is to take into account only a few recent measurements $\mathbf{x}_{N-L}, \cdots, \mathbf{x}_N$.
- If the $\mathbf{z}$ varies fast, then take L very small, otherwise, bigger values of L.
- We could do even better by doing a weighted average, but how?

Linear dynamical systems (LDS)

- Extension of HMMs to continuous latent variables.



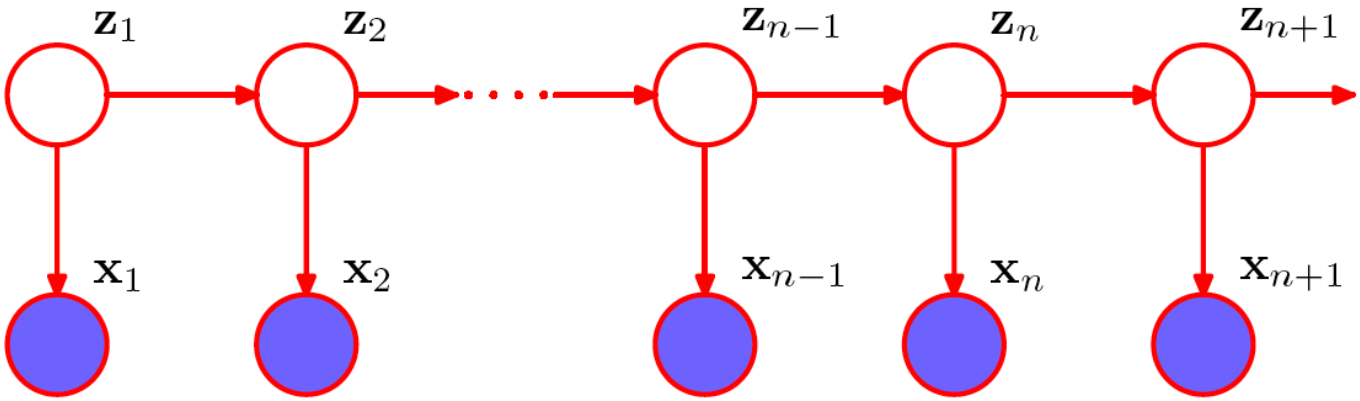
- The joint distribution of a **state space model** is

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_1, \cdots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$$

- We will consider here the **linear-Gaussian state space model**, so that the latent variables $\mathbf{z}$, as well as the observations $\mathbf{x}$, have Gaussian distributions.
- This is a generalization of the latent variable models (e.g., PPCA), where the latent variables are not independent, but they follow a Markov chain.

Linear dynamical systems (LDS)

- Extension of HMMs to continuous latent variables.



- The joint distribution of a **state space model** is

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_N, \mathbf{z}_1, \cdots, \mathbf{z}_N) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$$

- We will consider here the **linear-Gaussian state space model**, so that the latent variables $\mathbf{z}$, as well as the observations $\mathbf{x}$, have Gaussian distributions.
- This is a generalization of the latent variable models (e.g., PPCA), where the latent variables are not independent, but they follow a Markov chain.

Definition of linear dynamical systems

- We assume that the observations and latent variables can be expressed as

$$\begin{aligned} \mathbf{z}_n &= \mathbf{A}\mathbf{z}_{n-1} + \mathbf{w}_n \\ \mathbf{x}_n &= \mathbf{C}\mathbf{z}_n + \mathbf{v}_n \end{aligned}$$

  given the initial conditions $\mathbf{z}_1 = \boldsymbol{\mu}_0 + \mathbf{u}$, where the noise distributions are zero-noise Guassians

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|0, \boldsymbol{\Gamma}) \qquad \mathbf{v} \sim \mathcal{N}(\mathbf{v}|0, \boldsymbol{\Sigma}) \qquad \mathbf{u} \sim \mathcal{N}(\mathbf{u}|0, \mathbf{V}_0)$$

- The probabilistic model is

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1}, \boldsymbol{\Gamma}) \qquad p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})$$

  with initial conditions $p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1|\boldsymbol{\mu}_0, \mathbf{V}_0)$.
- The parameters of the model are $\theta = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$. They can be obtained by maximum likelihood via an EM algorithm.

Definition of linear dynamical systems

- We assume that the observations and latent variables can be expressed as

$$\begin{aligned} \mathbf{z}_n &= \mathbf{A}\mathbf{z}_{n-1} + \mathbf{w}_n \\ \mathbf{x}_n &= \mathbf{C}\mathbf{z}_n + \mathbf{v}_n \end{aligned}$$

  given the initial conditions $\mathbf{z}_1 = \boldsymbol{\mu}_0 + \mathbf{u}$, where the noise distributions are zero-noise Guassians

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|0, \boldsymbol{\Gamma}) \qquad \mathbf{v} \sim \mathcal{N}(\mathbf{v}|0, \boldsymbol{\Sigma}) \qquad \mathbf{u} \sim \mathcal{N}(\mathbf{u}|0, \mathbf{V}_0)$$

- The probabilistic model is

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1}, \boldsymbol{\Gamma}) \qquad p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})$$

  with initial conditions $p(\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_1|\boldsymbol{\mu}_0, \mathbf{V}_0)$.
- The parameters of the model are $\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$. They can be obtained by maximum likelihood via an EM algorithm.

Inference in LDS

- We want to find the marginal distributions for the latent variables condition on a sequence of observations.
- We also want to given a sequence of observations $\mathbf{x}_1, \cdots, \mathbf{x}_{n-1}$, predict the next latent variable $\mathbf{z}_n$, and the next observation $\mathbf{x}_n$.
- Note that because the linear dynamical system is a linear-Gaussian model, the joint distribution over all latent variables and observations is a Gaussian.
- The equations look like the ones for HMMs but we will have integrations instead of summations since the variables are now continuous.

Kalman filter: Forward equations I

- We start by defining the messages

$$\hat{\alpha}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n)$$

- Using the HMM recursion formulas for continuous variables we have

$$c_n\hat{\alpha}(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n)\int \hat{\alpha}(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})d\mathbf{z}_{n-1}$$

- Substituting the conditionals we have

$$\begin{aligned} c_n\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n) &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\int \mathcal{N}(\mathbf{z}_{n-1}|\boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{x}_{n-1}, \boldsymbol{\Gamma})d\mathbf{z}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{P}_{n-1}) \end{aligned}$$

- Here we assume that $\boldsymbol{\mu}_{n-1}$, and $\mathbf{V}_{n-1}$ are known, and we have defined

$$\mathbf{P}_{n-1} = \mathbf{A}\mathbf{V}_{n-1}\mathbf{A}^T + \boldsymbol{\Gamma}$$

Kalman filter: Forward equations I

- We start by defining the messages

$$\hat{\alpha}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n)$$

- Using the HMM recursion formulas for continuous variables we have

$$c_n\hat{\alpha}(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n)\int \hat{\alpha}(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})d\mathbf{z}_{n-1}$$

- Substituting the conditionals we have

$$\begin{aligned} c_n\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n) &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\int \mathcal{N}(\mathbf{z}_{n-1}|\boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{x}_{n-1}, \boldsymbol{\Gamma})d\mathbf{z}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{P}_{n-1}) \end{aligned}$$

- Here we assume that $\boldsymbol{\mu}_{n-1}$, and $\mathbf{V}_{n-1}$ are known, and we have defined

$$\mathbf{P}_{n-1} = \mathbf{A}\mathbf{V}_{n-1}\mathbf{A}^T + \boldsymbol{\Gamma}$$

Kalman filter: Forward equations II

- Given the values of $\boldsymbol{\mu}_{n-1}$, $\mathbf{V}_{n-1}$ and the new observation $\mathbf{x}_n$, we can evaluate the Gaussian marginal for $\mathbf{z}_n$ having mean $\boldsymbol{\mu}_n$ and covariance $\mathbf{V}_n$ as well as the normalization coefficient c_n

$$\begin{aligned} \boldsymbol{\mu}_n &= \mathbf{A}\boldsymbol{\mu}_{n-1} + \mathbf{K}_n(\mathbf{x}_n - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}) \\ \mathbf{V}_n &= (\mathbf{I} - \mathbf{K}_n\mathbf{C})\mathbf{P}_{n-1} \\ c_n &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma}) \end{aligned}$$

  where the **Kalman gain matrix** is defined as

$$\mathbf{K}_n = \mathbf{P}_{n-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- The initial conditions are given by

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{K}_1(\mathbf{x}_1 - \mathbf{C}\boldsymbol{\mu}_0) \qquad \mathbf{V}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{C})\mathbf{V}_0$$
$$c_1 = \mathcal{N}(\mathbf{x}_1|\mathbf{C}\boldsymbol{\mu}_0, \mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma}) \qquad \mathbf{K}_1 = \mathbf{V}_0\mathbf{C}^T(\mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- Interpretation is making prediction and doing corrections with $\mathbf{K}_n$.
- The likelihood can be computed as $p(\mathbf{X}) = \prod_{n=1}^{N} c_n$.

Kalman filter: Forward equations II

- Given the values of $\boldsymbol{\mu}_{n-1}$, $\mathbf{V}_{n-1}$ and the new observation $\mathbf{x}_n$, we can evaluate the Gaussian marginal for $\mathbf{z}_n$ having mean $\boldsymbol{\mu}_n$ and covariance $\mathbf{V}_n$ as well as the normalization coefficient c_n

$$\begin{aligned} \boldsymbol{\mu}_n &= \mathbf{A}\boldsymbol{\mu}_{n-1} + \mathbf{K}_n(\mathbf{x}_n - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}) \\ \mathbf{V}_n &= (\mathbf{I} - \mathbf{K}_n\mathbf{C})\mathbf{P}_{n-1} \\ c_n &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma}) \end{aligned}$$

  where the **Kalman gain matrix** is defined as

$$\mathbf{K}_n = \mathbf{P}_{n-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- The initial conditions are given by

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{K}_1(\mathbf{x}_1 - \mathbf{C}\boldsymbol{\mu}_0) \qquad \mathbf{V}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{C})\mathbf{V}_0$$
$$c_1 = \mathcal{N}(\mathbf{x}_1|\mathbf{C}\boldsymbol{\mu}_0, \mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma}) \qquad \mathbf{K}_1 = \mathbf{V}_0\mathbf{C}^T(\mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- Interpretation is making prediction and doing corrections with $\mathbf{K}_n$.
- The likelihood can be computed as $p(\mathbf{X}) = \prod_{n=1}^{N} c_n$.

Kalman filter: Forward equations III

- If the measurement noise is small compared to the rate at which $\mathbf{z}$ is evolving, then the posterior distribution for $\mathbf{z}_n$ depends only on the current measurement $\mathbf{x}_n$.
- If $\mathbf{z}_n$ is evolving slowly relative to the observation noise level, then the posterior mean for $\mathbf{z}_n$ is obtained by averaging all of the measurements obtained up to that time.
- One of the most important applications of the Kalman filter is to tracking.
- Used for real-time prediction.

Kalman filer example

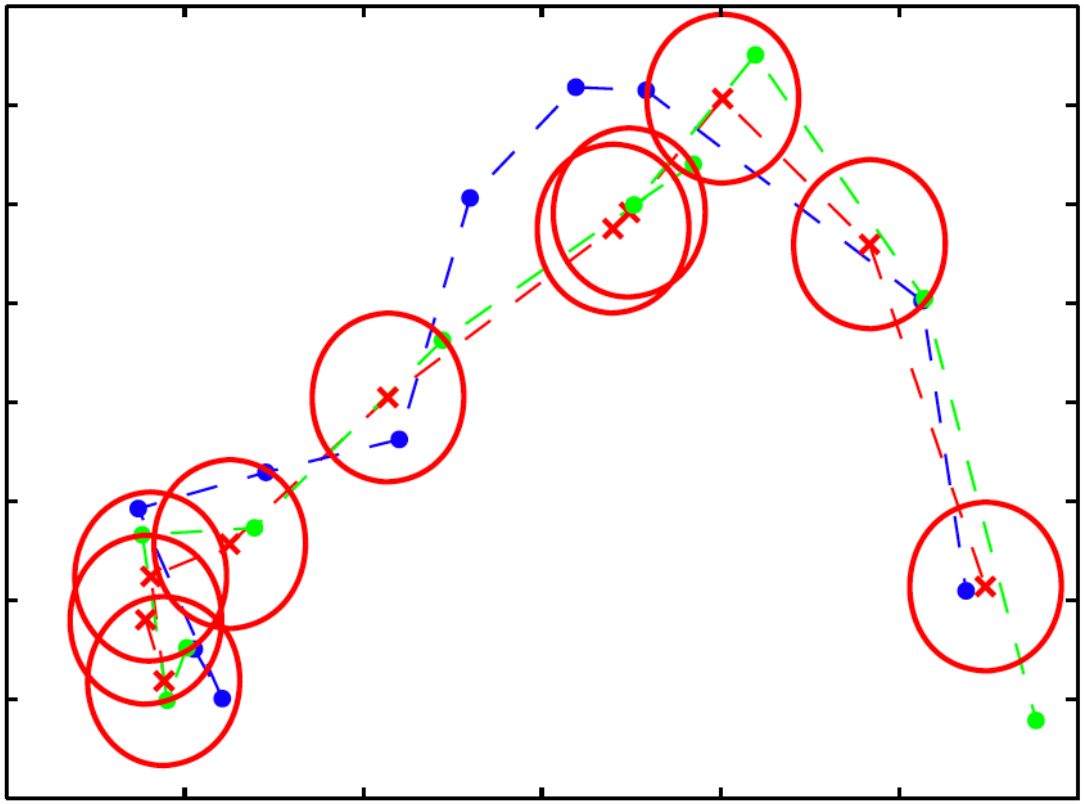


Figure: Tracking using a Kalman filter (Bishop, Springer 2007). Blue points indicate true positions at successive times, the green points are noisy measurements of the positions, and the red crosses indicates the means of the inferred posterior distributions.

Kalman smoother: Backward equations I

- In the Kalman filter equations we have seen how to compute the posterior marginal of a node $\mathbf{z}_n$ given observations $\mathbf{x}_1, \cdots, \mathbf{x}_n$.
- We now look into finding the marginal for a node $\mathbf{z}_n$ given all observations $\mathbf{x}_1, \cdots, \mathbf{x}_N$.
- For temporal data, this means the inclusion of future as well as past observations.
- This cannot be used for real-time prediction, but it is crucial for learning the parameters.
- As in the HMM, this can be done by propagating messages from node $\mathbf{x}_N$ to node $\mathbf{x}_1$, and combining this information with the forward message passing step used to compute $\hat{\alpha}(\mathbf{z}_n)$

Kalman smoother: Backward equations II

- The backward recursion is typically formulated in terms of

$$\gamma(\mathbf{z}_n) = \hat{\alpha}(\mathbf{z}_n)\hat{\beta}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\hat{\boldsymbol{\mu}}_n, \hat{\mathbf{V}}_n)$$

- We write the backward recursion for continuous variables as

$$c_{n+1}\hat{\beta}(\mathbf{z}_{n+1}) = \int \hat{\beta}(\mathbf{z}_{n+1})p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n)d\mathbf{z}_{n+1}$$

- Multiplying both sides by $\hat{\alpha}(\mathbf{z}_n)$ and substituting yields

$$\begin{aligned}\hat{\boldsymbol{\mu}}_n &= \boldsymbol{\mu}_n + \mathbf{J}_n(\hat{\boldsymbol{\mu}}_{n+1} - \mathbf{A}\boldsymbol{\mu}_N) \\ \hat{\mathbf{V}}_n &= \mathbf{V}_n + \mathbf{J}_n(\hat{\mathbf{V}}_{n+1} - \mathbf{P}_n)\mathbf{J}_n^T\end{aligned}$$

  where we have defined

$$\mathbf{J}_n = \mathbf{V}_n\mathbf{A}^T(\mathbf{P}_n)^{-1}$$

- The forward pass has to be done before so that we know $\boldsymbol{\mu}_n$ and $\mathbf{V}_n$.

Kalman smoother: Backward equations II

- The backward recursion is typically formulated in terms of

$$\gamma(\mathbf{z}_n) = \hat{\alpha}(\mathbf{z}_n)\hat{\beta}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\hat{\boldsymbol{\mu}}_n, \hat{\mathbf{V}}_n)$$

- We write the backward recursion for continuous variables as

$$c_{n+1}\hat{\beta}(\mathbf{z}_{n+1}) = \int \hat{\beta}(\mathbf{z}_{n+1})p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n)d\mathbf{z}_{n+1}$$

- Multiplying both sides by $\hat{\alpha}(\mathbf{z}_n)$ and substituting yields

$$\begin{aligned}\hat{\boldsymbol{\mu}}_n &= \boldsymbol{\mu}_n + \mathbf{J}_n(\hat{\boldsymbol{\mu}}_{n+1} - \mathbf{A}\boldsymbol{\mu}_N) \\ \hat{\mathbf{V}}_n &= \mathbf{V}_n + \mathbf{J}_n(\hat{\mathbf{V}}_{n+1} - \mathbf{P}_n)\mathbf{J}_n^T\end{aligned}$$

  where we have defined

$$\mathbf{J}_n = \mathbf{V}_n\mathbf{A}^T(\mathbf{P}_n)^{-1}$$

- The forward pass has to be done before so that we know $\boldsymbol{\mu}_n$ and $\mathbf{V}_n$.

Kalman smoother: Backward equations III

- The other quantity necessary for the EM algorithm is the pairwise posterior marginals

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = (c_n)^{-1}\hat{\alpha}(\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{z}_{n-1})\hat{\beta}(\mathbf{z}_n)$$

  which is going to be a Gaussian with mean given with components $\gamma(\mathbf{z}_{n-1})$ and $\gamma(\mathbf{z}_n)$ and covariance

$$cov[\mathbf{z}_n, \mathbf{z}_{n-1}] = \mathbf{J}_{n-1}\hat{\mathbf{V}}_n$$

Learning the LDS I

- We have considered the inference problem where $\theta = \{\mathbf{A}, \mathbf{\Gamma}, \mathbf{C}, \mathbf{\Sigma}, \mu_0, \mathbf{V}_0\}$ were assumed to be known.
- As in the HMM, we learn these parameters by maximum likelihood on an EM framework.
- As before we look into the i-th iteration. In the E-step we run the inference algorithm to compute the posterior $p(\mathbf{Z}|\mathbf{X}, \theta^{(i)})$, so that

$$\begin{aligned} E[\mathbf{z}_n] &= \hat{\mu}_n \\ E[\mathbf{z}_n \mathbf{z}_{n-1}^T] &= \mathbf{J}_{n-1}\hat{\mathbf{V}}_n + \hat{\mu}_n \hat{\mu}_{n-1}^T \\ E[\mathbf{z}_n \mathbf{z}_n^T] &= \hat{\mathbf{V}}_n + \hat{\mu}_n \hat{\mu}_n^T \end{aligned}$$

- The complete-data log likelihood is

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}, \theta) = & \log p(\mathbf{z}_1|\mu_0, \mathbf{V}_0) + \sum_{n=2}^{N} \log p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}, \mathbf{\Gamma}) \\ & + \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{C}, \mathbf{\Sigma}) \end{aligned}$$

Learning the LDS I

- We have considered the inference problem where $\theta = \{\mathbf{A}, \mathbf{\Gamma}, \mathbf{C}, \mathbf{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$ were assumed to be known.
- As in the HMM, we learn these parameters by maximum likelihood on an EM framework.
- As before we look into the i-th iteration. In the E-step we run the inference algorithm to compute the posterior $p(\mathbf{Z}|\mathbf{X}, \theta^{(i)})$, so that

$$\begin{aligned} E[\mathbf{z}_n] &= \hat{\boldsymbol{\mu}}_n \\ E[\mathbf{z}_n \mathbf{z}_{n-1}^T] &= \mathbf{J}_{n-1}\hat{\mathbf{V}}_n + \hat{\boldsymbol{\mu}}_n \hat{\boldsymbol{\mu}}_{n-1}^T \\ E[\mathbf{z}_n \mathbf{z}_n^T] &= \hat{\mathbf{V}}_n + \hat{\boldsymbol{\mu}}_n \hat{\boldsymbol{\mu}}_n^T \end{aligned}$$

- The complete-data log likelihood is

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}, \theta) = & \log p(\mathbf{z}_1|\boldsymbol{\mu}_0, \mathbf{V}_0) + \sum_{n=2}^{N} \log p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}, \mathbf{\Gamma}) \\ & + \sum_{n=1}^{N} \log p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{C}, \mathbf{\Sigma}) \end{aligned}$$

Learning the LDS II

- We now take the expectation with respect to the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(i)})$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})]$$

- In the M-step we maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$ with respect to $\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$.
- Let's compute this with respect to every set of parameters.
- First consider $\boldsymbol{\mu}_0$ and $\mathbf{V}_0$, computing the expectation with respect to $\mathbf{Z}$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = -\frac{1}{2}\log|\mathbf{V}_0| - E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}}\left[\frac{1}{2}(\mathbf{z}_1 - \boldsymbol{\mu}_0)^T\mathbf{V}_0^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_0)\right] + const$$

We use the maximum likelihood of a Gaussian to get

$$\boldsymbol{\mu}_0^{(i+1)} = E[\mathbf{z}_1] \qquad \mathbf{V}_0^{(i+1)} = E[\mathbf{z}_1\mathbf{z}_1^T] - E[\mathbf{z}_1]E[\mathbf{z}_1^T]$$

Learning the LDS II

- We now take the expectation with respect to the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{(i)})$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}}[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})]$$

- In the M-step we maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)})$ with respect to $\boldsymbol{\theta} = \{\mathbf{A}, \boldsymbol{\Gamma}, \mathbf{C}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_0, \mathbf{V}_0\}$.
- Let's compute this with respect to every set of parameters.
- First consider $\boldsymbol{\mu}_0$ and $\mathbf{V}_0$, computing the expectation with respect to $\mathbf{Z}$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = -\frac{1}{2}\log|\mathbf{V}_0| - E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}}\left[\frac{1}{2}(\mathbf{z}_1 - \boldsymbol{\mu}_0)^T\mathbf{V}_0^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_0)\right] + const$$

We use the maximum likelihood of a Gaussian to get

$$\boldsymbol{\mu}_0^{(i+1)} = E[\mathbf{z}_1] \qquad \mathbf{V}_0^{(i+1)} = E[\mathbf{z}_1\mathbf{z}_1^T] - E[\mathbf{z}_1]E[\mathbf{z}_1^T]$$

Learning the LDS III

- Similarly we optimize over $\mathbf{A}$ and $\mathbf{\Gamma}$ by maximizing

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = -\frac{N-1}{2} \log |\mathbf{\Gamma}| - E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}} \left[\frac{1}{2} \sum_{n=2}^{N} (\mathbf{z}_n - \mathbf{A}\mathbf{z}_{n-1})^T \mathbf{\Gamma}^{-1} (\mathbf{z}_n - \mathbf{A}\mathbf{z}_{n-1}) \right] + const$$

which yields

$$\begin{aligned}
\mathbf{A}^{(i+1)} &= \left(\sum_{n=2}^{N} E[\mathbf{z}_n \mathbf{z}_{n-1}^T] \right) \left(\sum_{n=2}^{N} E[\mathbf{z}_{n-1} \mathbf{z}_{n-1}^T] \right)^{-1} \\
\mathbf{\Gamma}^{(i+1)} &= \frac{1}{N-1} \sum_{n=2}^{N} \{ E[\mathbf{z}_n \mathbf{z}_n^T] - \mathbf{A}^{(i+1)} E[\mathbf{z}_{n-1} \mathbf{z}_n^T] \\
&\quad - E[\mathbf{z}_n \mathbf{z}_{n-1}^T] \mathbf{A}^{(i+1)} + \mathbf{A}^{(i+1)} E[\mathbf{z}_{n-1} \mathbf{z}_{n-1}^T] (\mathbf{A}^{(i+1)})^T \}
\end{aligned}$$

- Note that $\mathbf{A}^{(i+1)}$ has to be evaluated before computing $\mathbf{\Gamma}^{(i+1)}$.

Learning the LDS IV

- Finally in order to compute the new values of $\mathbf{C}$ and $\mathbf{\Sigma}$ we maximize

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i)}) = -\frac{N}{2}\log|\mathbf{\Sigma}| - E_{\mathbf{Z}|\boldsymbol{\theta}^{(i)}}\left[\frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \mathbf{C}\mathbf{z}_n)^T)\mathbf{\Sigma}^{-1}(\mathbf{x}_n - \mathbf{C}\mathbf{z}_n)\right] + const$$

which gives

$$\begin{aligned}
\mathbf{C}^{(i+1)} &= \left(\sum_{n=1}^{N} E[\mathbf{z}_n^T]\right)\left(\sum_{n=1}^{N} E[\mathbf{z}_n\mathbf{z}_n^T]\right)^{-1} \\
\mathbf{\Sigma}^{(i+1)} &= \frac{1}{N}\sum_{n=1}^{N}\{\mathbf{x}_n\mathbf{x}_n^T - \mathbf{C}^{(i+1)}E[\mathbf{z}_n]\mathbf{x}_n^T \\
&\quad -\mathbf{x}_n E[\mathbf{z}_n^T]\mathbf{C}^{(i+1)} + \mathbf{C}^{(i+1)}E[\mathbf{z}_n\mathbf{z}_n^T]\mathbf{C}^{(i+1)}\}
\end{aligned}$$

- Note that what all this distributions have in common is that they are Gaussian, so we can compute their mean and covariance in closed form.

Non-linear dynamics

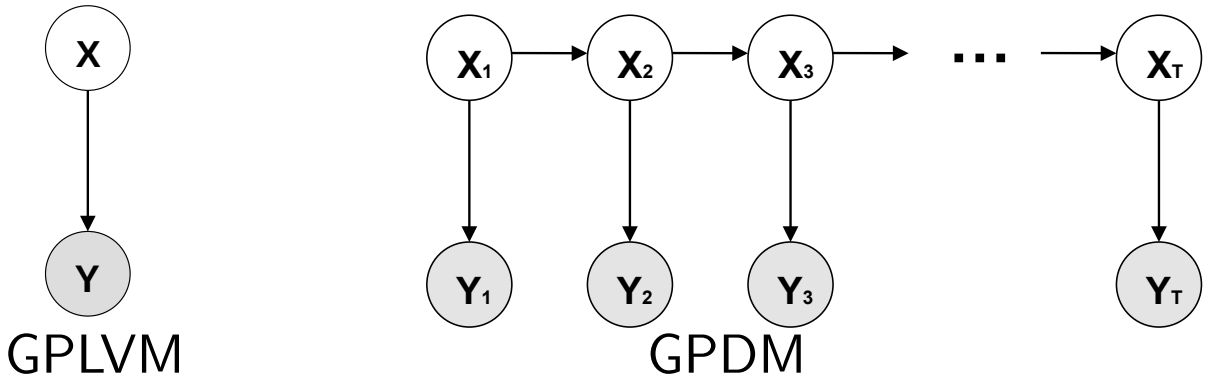
- The same trick as for Probabilistic PCA can be applied to latent variable models with Markov chain assumptions
- The mapping from latent space to high dimensional space as

$$\mathbf{y}_{i,:} = \mathbf{W}\psi(\mathbf{x}_{i,:}) + \boldsymbol{\eta}_{i,:}, \quad \text{where} \quad \eta_{i,:} \sim N\left(\mathbf{0}, \sigma^2\mathbf{I}\right).$$

- We can augment the model with ARMA dynamics. This is called the Gaussian process dynamical model (GPDM) (Wang et al., 05).

$$\mathbf{x}_{t+1,:} = \mathbf{P}\phi(\mathbf{x}_{t:t-\tau,:}) + \boldsymbol{\gamma}_{i,:}, \quad \text{where} \quad \gamma_{i,:} \sim N\left(\mathbf{0}, \sigma_d^2\mathbf{I}\right).$$

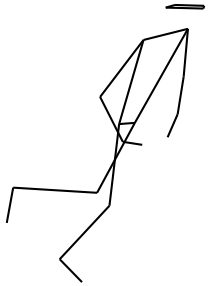
with the dynamics model as a Gaussian process.



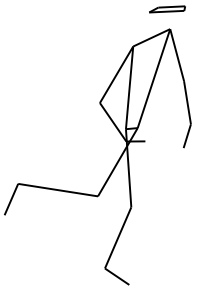
Stick Man Data

- $N = 55$ frames of motion capture.
- xyz locations of 34 points on the body.
- $D = 102$ dimensional data.
- "Run 1" available from `http://accad.osu.edu/research/mocap/mocap_data.htm`.

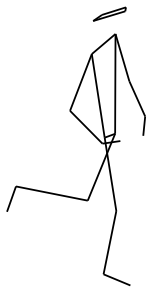
Changing



Angle



of Run



Motion Capture Results

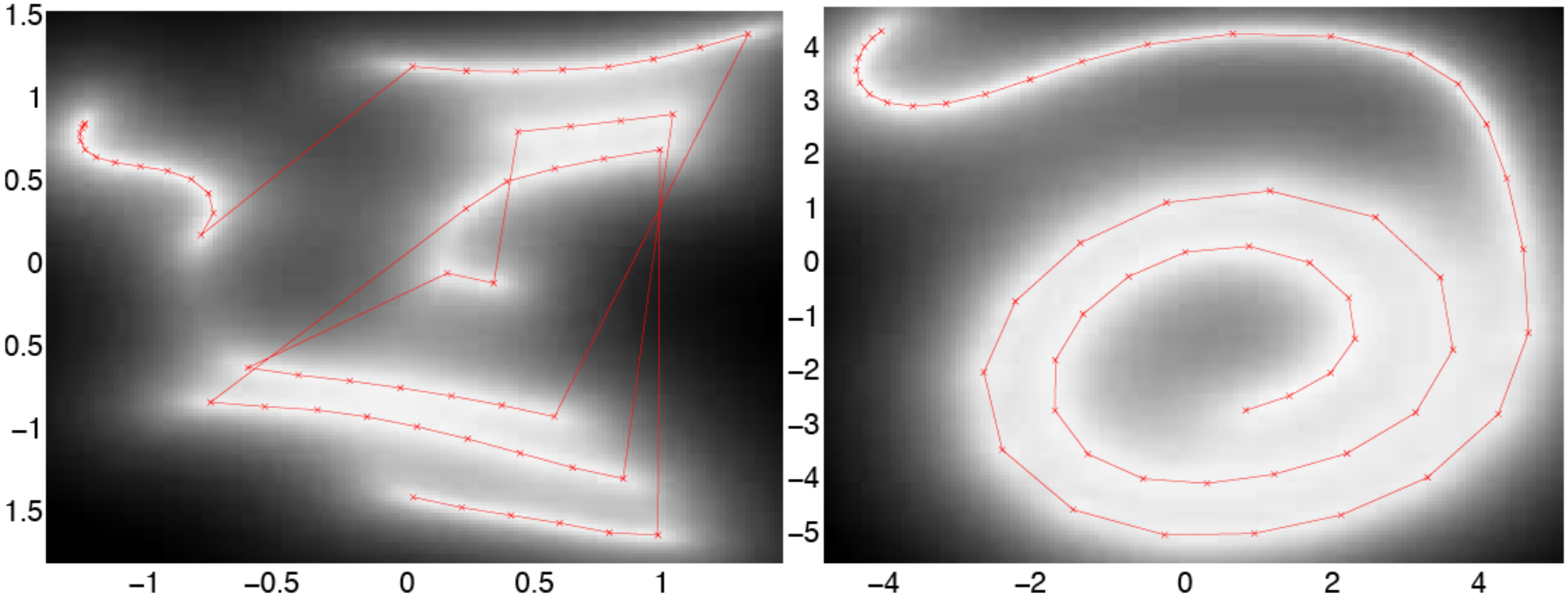


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

Learning and sampling

Model learned from 6 walking subjects,1 gait cycle each, on treadmill at same speed with a 20 DOF joint parameterization (no global pose)

Figure: Density

Figure: Randomly generated trajectories

More?

- If you want to learn more, look at the additional material.
- Otherwise, do the research project on this topic!
- Next week we will do particle filters and image likelihoods.
- Let's do some exercises now!