Lecture 7

Models of Sequential Data

University of Amsterdam

Modelling sequential data

Graphical models of sequential data are just graphical models, but:

- The assumption that all examples are drawn i.i.d. from the same distribution does not hold for individual observations anymore
- It does still hold for individual sequences, though
- The data can be sequential over time, space, ...

Example

Speech recognition, weather modelling, DNA sequence modelling

Adapting to the data

Graphical Models

- Give us a way to visualise factorisation of probabilities
- Give us powerful algorithms automatically find the optimal way to compute marginal probabilities, conditional probabilities, ...
- Therefore allow for efficient algorithms for training
- Cannot, in general, adapt the factorisation to the data

Models of sequential data slightly relax this:

- We now set the general structure of the model
- The exact structure is automatically adapted to the length of the sequence

1. Models of data sequences
   - Independent observations
   - Markov Models
2. Hidden Markov Models
   - Inference: The Baum-Welch Algorithm
   - The Viterbi Algorithm
   - Training
   - Extensions to HMM
   - Dealing with very unlikely events
3. Linear Dynamical System
   - Representation
   - The Kalman Filter
   - Extensions

1 Models of data sequences
- Independent observations
- Markov Models

2 Hidden Markov Models
- Inference: The Baum-Welch Algorithm
- The Viterbi Algorithm
- Training
- Extensions to HMM
- Dealing with very unlikely events

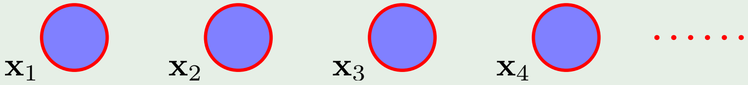
3 Linear Dynamical System
- Representation
- The Kalman Filter
- Extensions

Assume independence

Simplest solution:

- Assume that all observations are independent
- Problem: fails to exploit sequential patterns in the data

Example: Rainy days



The only information we can obtain from this model is the frequency of rainy days.

Markov Models

If we do not assume any independence, we can factorise the distribution as:

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

However:

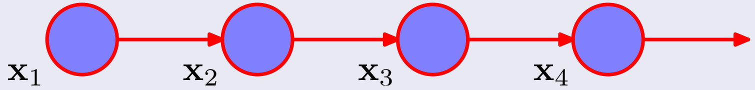
- In general the most recent observations are more likely to affect the current observation.
- If we assume that only the single most recent observation affects the current observation, we obtain the **first order Markov chain**

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

Markov Chains

University of Amsterdam

First order Markov Chain



- In most applications, we constrain $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ to be equal over the length of the chain: **homogeneous markov chain**

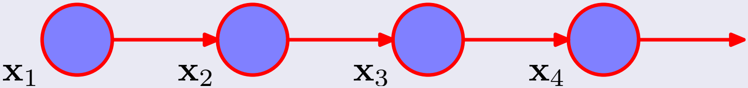
Example

This was used to model character occurances in text

Markov Chains

University of Amsterdam

First order Markov Chain



- In most applications, we constrain $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ to be equal over the length of the chain: **homogeneous markov chain**

Example

This was used to model character occurances in text

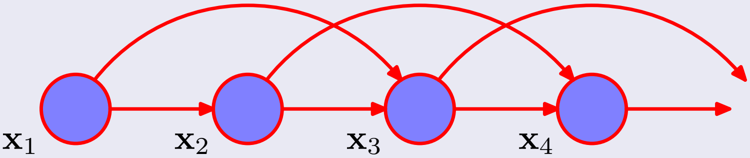
Higher order Markov chains

First order Markov chains are still very restrictive:

- Modelling trends requires more information from the past
- Second order Markov chain:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1) \prod_{n=3} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2}) \qquad (3)$$

Second order Markov chain



Higher order Markov chains

UNIVERSITY OF AMSTERDAM

There is a high penalty for the extra flexibility:

- Imagine K discrete states
  - First order chain: $K \times (K - 1)$ independent parameters
  - Second order chain: $K^2 \times (K - 1)$ parameters
  - In general, Mth order: $K^{M-1} \times (K - 1)$ parameters
- The complexity grows exponentially with M

This becomes impractical very quickly.

1. Models of data sequences
   - Independent observations
   - Markov Models

2. Hidden Markov Models
   - Inference: The Baum-Welch Algorithm
   - The Viterbi Algorithm
   - Training
   - Extensions to HMM
   - Dealing with very unlikely events

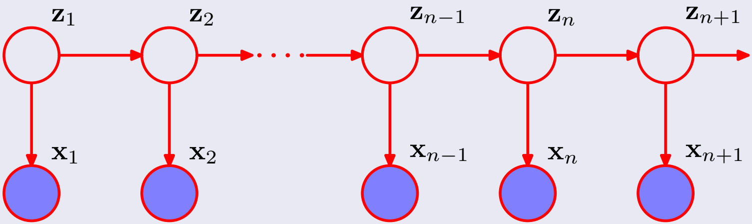
3. Linear Dynamical System
   - Representation
   - The Kalman Filter
   - Extensions

Hidden Markov Models

How can we keep a longer memory and still restrict the number of parameters?

- Create a latent variable model: the probability of observations depends on the latent variable (see Mixtures of Gaussians)
- The latent variables are part of a latent Markov chain
- D-separation shows that all observations are now dependent

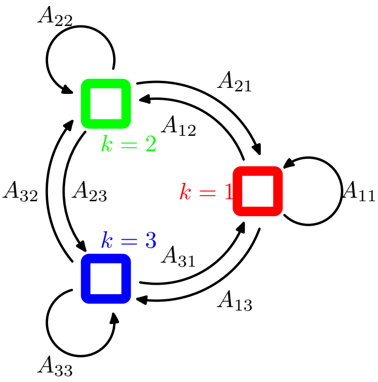
Hidden Markov Model



HMM as an FSA

Hidden Markov Models are often viewed as Finite State Machines

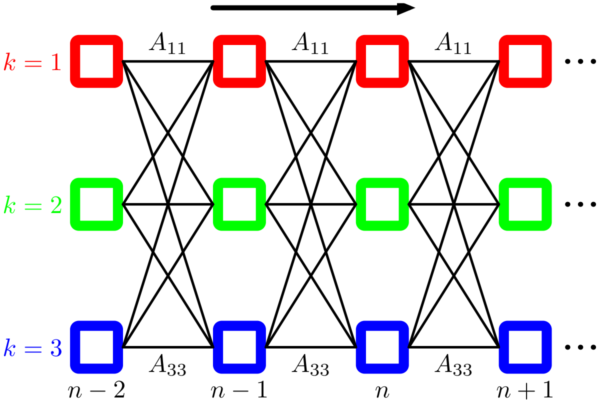
- The arrows indicate the *transition probabilities*, typically denoted by the matrix **A**
- *Emission probabilities* are associated with each state
- Not depicted here, is the probability of starting a sequence in a given state k, typically denoted by π_k



Lattices

UNIVERSITY OF AMSTERDAM

We can then “unroll” the FSA over time to form a lattice



Inference in the HMM

UNIVERSITY OF AMSTERDAM

Inference allows us to answer the questions:

- What is the probability of each state, for each observation

Example

In speech recognition, what is the probability that a certain phoneme corresponds to a certain audio observation?

- What is the probability of seeing an observation sequence, according to the model?

Example

In elderly care, it is useful to monitor how well patients perform activities at home. It is very hard to have a model of the problems they may have, but it is possible to detect when their activities become very unlikely according to the model.

Inference in the HMM

Inference allows us to answer the questions:

- What is the probability of each state, for each observation

Example

In speech recognition, what is the probability that a certain phoneme corresponds to a certain audio observation?

- What is the probability of seeing an observation sequence, according to the model?

Example

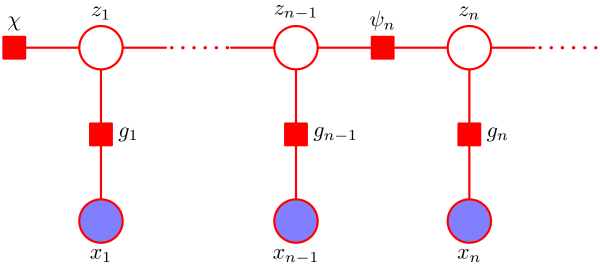
In elderly care, it is useful to monitor how well patients perform activities at home. It is very hard to have a model of the problems they may have, but it is possible to detect when their activities become very unlikely according to the model.

The Forward-Backward Algorithm

University of Amsterdam

HMM have been around for much longer than graphical models

- Specific versions of the sum-product and max-product were developed for the HMM
- The sum-product rule for HMM is known as the Forward-Backward or Baum-Welch algorithm
- It consists of a single forward and a single backward pass through the chain



The Forward pass

The message leaving the first node is

$$\begin{aligned} \mu_{z_1 \to \psi_1} &= p(x_1|z_1)\, p(z_1) \\ &= p(x_1, z_1) \end{aligned}$$

The message leaving the second node is

$$\begin{aligned} \mu_{z_2 \to \psi_3} &= p(x_2|z_2) \sum_{\mathbf{z}_1} p(x_1, z_1)\, p(z_2|z_1) \\ &= p(x_1, x_2, z_2) \end{aligned}$$

In general:

$$\begin{aligned} \mu_{\psi_{n-1} \to z_n} &= p(x_1, \ldots, x_n, z_n) \\ &= \alpha(z_n) \end{aligned}$$

The Backward pass

UNIVERSITY OF AMSTERDAM

The message entering the penultimate node is

$$\mu_{\psi_{N-1} \to z_{N-1}} = \sum_{\mathbf{z}_N} p(z_N | z_{N-1})\, p(x_N | z_N) = p(x_N | z_{N-1})$$

The message entering the antepenultimate node is

$$\mu_{\psi_{N-2} \to z_{N-2}} = \sum_{\mathbf{z}_{N-1}} p(z_{N-1} | z_{N-2})\, p(x_{N-1} | z_{N-1})\, p(x_N | z_{N-1}) = p(x_{N-1}, x_N | z_{N-2})$$

In general, the "backward" message is:

$$\mu_{\psi_n \to z_n} = p(x_{n+1}, \ldots, x_N | z_n) = \beta(z_n)$$

The Viterbi Algorithm

The sequence of hidden states often has a real, physical explanation

Example

In speech recognition, the sequence of phonemes for a sequence of sound observations.

It is therefore useful to have the most likely single sequence of observations, rather than the sequence of most likely observations

- We obtain this using the max-sum algorithm
- For the HMM, this is called the Viterbi Algorithm

Training of the HMM

UNIVERSITY OF AMSTERDAM

We can train the HMM using the EM algorithm, optimising

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \tag{4}$$

We introduce

$$\gamma(z_{nk}) = p(z_{nk}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \tag{5}$$

$$\xi(z_{n-1\,j}, z_{n,k}) = p(z_{n-1\,j}, z_{nk}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \tag{6}$$

so that the expectation of the complete log-likelihood becomes

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{k} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} +$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n|z_k)$$

The E step

UNIVERSITY OF AMSTERDAM

The *responsibilities* $\gamma(z_{nk})$ can be computed from the forward and backward probabilities:

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \tag{7}$$

$$= \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})} \tag{8}$$

and similarly for the pairwise responsibilities:

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}) \tag{9}$$

$$= \frac{\alpha(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})} \tag{10}$$

The M step

- Taking the first derivative of $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ with respect to the parameters and setting it to zero gives us the update rules:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \tag{11}$$

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=2}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})} \tag{12}$$

- optimising the *emission* probabilities depends on the specific form of these, and is identical to a mixture model.

The M-Step

University of Amsterdam

Example: Gaussian observations

If the observations have a Gaussian distribution, the maximum likelihood parameters are given by:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{13}$$

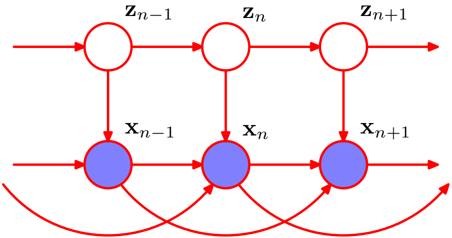
$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^\top}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{14}$$

(See Gaussian Mixture Models)

Extensions

The basic HMM framework can be extended in many ways

- Autoregressive HMM
- Input-output HMM
- Factorial HMM
- Explicit time model (Hidden semi-Markov model)

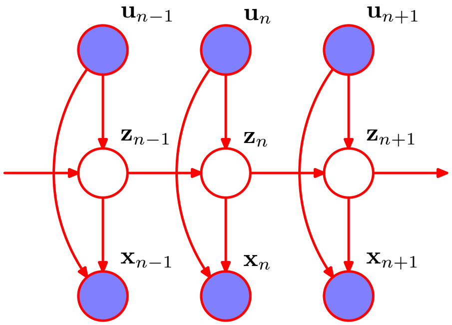


Extensions

University of Amsterdam

The basic HMM framework can be extended in many ways

- Autoregressive HMM
- Input-output HMM
- Factorial HMM
- Explicit time model (Hidden semi-Markov model)

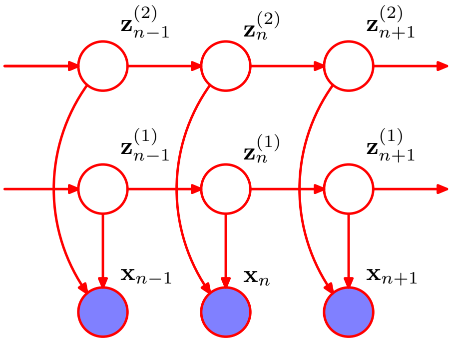


Extensions

UNIVERSITY OF AMSTERDAM

The basic HMM framework can be extended in many ways

- Autoregressive HMM
- Input-output HMM
- Factorial HMM
- Explicit time model (Hidden semi-Markov model)



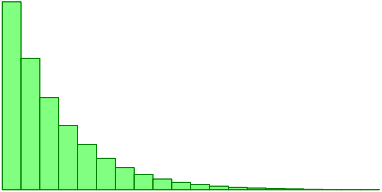
Extensions

UNIVERSITY OF AMSTERDAM

The basic HMM framework can be extended in many ways

- Autoregressive HMM
- Input-output HMM
- Factorial HMM
- Explicit time model (Hidden semi-Markov model)

$$p(z_n, \ldots, z_{n+t} = z) = A_{zz}(1 - A_{zz})$$



Dealing with very low probabilities

The probability of a chain of observations quickly becomes small

Example: Throwing a coin

If we throw a fair die 1000 times, what is the probability of observing the exact sequence that we got? $\left(\frac{1}{6}\right)^{1000}$

We cannot easily represent such low probabilities

Dealing with very low probabilities

The probability of a chain of observations quickly becomes small

Example: Throwing a coin

If we throw a fair die 1000 times, what is the probability of observing the exact sequence that we got? $\left(\frac{1}{6}\right)^{1000}$

We cannot easily represent such low probabilities

Dealing with very low probabilities

The probability of a chain of observations quickly becomes small

Example: Throwing a coin

If we throw a fair die 1000 times, what is the probability of observing the exact sequence that we got? $\left(\frac{1}{6}\right)^{1000}$

We cannot easily represent such low probabilities

Probabilities in chains

UNIVERSITY OF AMSTERDAM

Problem:

- (Double-precision) Floating point arithmetic cannot represent probabilities associated with long chains

Solution:

1. Rescale probabilities along the chain (and keep track of the scaling factors)
   1. Unwieldy
   2. Not a general solution (*e.g.*, high-dimensional Gaussian)
2. Use log-probabilities

Probabilities in chains

Problem:

- (Double-precision) Floating point arithmetic cannot represent probabilities associated with long chains

Solution:

1. Rescale probabilities along the chain (and keep track of the scaling factors)
    1. Unwieldy
    2. Not a general solution (*e.g.*, high-dimensional Gaussian)
2. Use log-probabilities

Probabilities in chains

Problem:

- (Double-precision) Floating point arithmetic cannot represent probabilities associated with long chains

Solution:

1. Rescale probabilities along the chain (and keep track of the scaling factors)
    1. Unwieldy
    2. Not a general solution (*e.g.*, high-dimensional Gaussian)
2. Use log-probabilities

Using log-probabilities

University of Amsterdam

	Probabilities	log-probabilies
Range	$[0, 1]$	$[-\infty, 0]$
$p(A \wedge B)$	$p(A)\, p(B\|A)$	$\log p(A) + \log p(B\|A)$
$p(A \mid B)$	$\frac{p(A,B)}{p(B)}$	$\log p(A, B) - \log p(B)$
$p(A \vee B)$	$p(A) + P(B\|A)$	$\ldots$

Using log-probabilities

UNIVERSITY OF AMSTERDAM

	Probabilities	log-probabilies
Range	$[0, 1]$	$[-\infty, 0]$
$p(A \wedge B)$	$p(A)\, p(B \mid A)$	$\log p(A) + \log p(B \mid A)$
$p(A \mid B)$	$\frac{p(A,B)}{p(B)}$	$\log p(A, B) - \log p(B)$
$p(A \vee B)$	$p(A) + P(B \mid A)$	$\dots$

Using log-probabilities

	Probabilities	log-probabilies
Range	$[0, 1]$	$[-\infty, 0]$
$p(A \wedge B)$	$p(A)\, p(B\|A)$	$\log p(A) + \log p(B\|A)$
$p(A \mid B)$	$\frac{p(A,B)}{p(B)}$	$\log p(A, B) - \log p(B)$
$p(A \vee B)$	$p(A) + P(B\|A)$	$\ldots$

Using log-probabilities

	Probabilities	log-probabilies
Range	$[0, 1]$	$[-\infty, 0]$
$p(A \wedge B)$	$p(A)\, p(B\|A)$	$\log p(A) + \log p(B\|A)$
$p(A \mid B)$	$\frac{p(A,B)}{p(B)}$	$\log p(A, B) - \log p(B)$
$p(A \vee B)$	$p(A) + P(B\|A)$	$\dots$

Using log-probabilities

UNIVERSITY OF AMSTERDAM

	Probabilities	log-probabilies
Range	$[0, 1]$	$[-\infty, 0]$
$p(A \wedge B)$	$p(A)\, p(B\|A)$	$\log p(A) + \log p(B\|A)$
$p(A \mid B)$	$\frac{p(A,B)}{p(B)}$	$\log p(A, B) - \log p(B)$
$p(A \vee B)$	$p(A) + P(B\|A)$	$\ldots$

Log of sum of exponentials

UNIVERSITY OF AMSTERDAM

$\log(a+b)$ when given $\log(a)$ and $\log(b)$

$$\begin{aligned}\log(a+b) &= \log(\exp\log a + \exp\log b) \\ &= \log\left(\exp\log a\left[1+\frac{\exp\log b}{\exp\log a}\right]\right) \\ &= \log a + \log(1+\exp(\log b - \log a))\end{aligned}$$

Define a new function:

$$\operatorname{lse}(a,b) = \log(\exp a + \exp b)$$

Log of sum of exponentials

University of Amsterdam

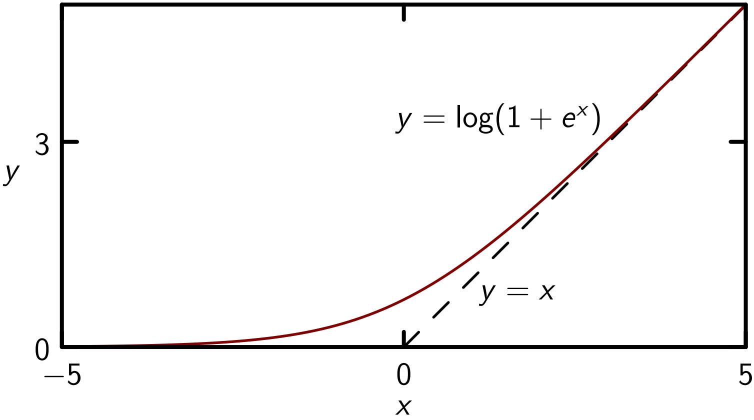
$\log(a+b)$ when given $\log(a)$ and $\log(b)$

$$\begin{aligned}\log(a+b) &= \log(\exp\log a + \exp\log b) \\ &= \log\left(\exp\log a\left[1+\frac{\exp\log b}{\exp\log a}\right]\right) \\ &= \log a + \log(1+\exp(\log b - \log a))\end{aligned}$$

Define a new function:

$$\mathrm{lse}(a,b) = \log(\exp a + \exp b)$$

log of sum of exponentials



Dealing with log-probabilities

UNIVERSITY OF AMSTERDAM

log-of-sum-of-exponentials — lse

$$\log(a+b) = \log a + \log(1 + \exp(\log b - \log a))$$

Some implementation details:

- Keep the exponent small (choose $a > b$)
- $\log(1+x)$ is a special function: `log1p`
- Handle infinity as a special case
- For summations:

$$a + b + c + \ldots = ((a+b)+c) + \ldots$$

$$\text{lse}(a, b, c, \ldots) = \text{lse}(\text{lse}(\text{lse}(a, b), c), \ldots)$$

Dealing with log-probabilities

log-of-sum-of-exponentials — lse

$$\log(a+b) = \log a + \log(1 + \exp(\log b - \log a))$$

Some implementation details:

- Keep the exponent small (choose $a > b$)
- $\log(1+x)$ is a special function: `log1p`
- Handle infinity as a special case
- For summations:

$$a + b + c + \ldots = ((a+b)+c) + \ldots$$

$$\mathrm{lse}(a, b, c, \ldots) = \mathrm{lse}(\mathrm{lse}(\mathrm{lse}(a, b), c), \ldots)$$

Dealing with log-probabilities

UNIVERSITY OF AMSTERDAM

log-of-sum-of-exponentials — lse

$$\log(a+b) = \log a + \log(1 + \exp(\log b - \log a))$$

Some implementation details:

- Keep the exponent small (choose $a > b$)
- $\log(1+x)$ is a special function: `log1p`
- Handle infinity as a special case
- For summations:

$$a + b + c + \ldots = ((a+b)+c)+\ldots$$

$$\mathrm{lse}(a, b, c, \ldots) = \mathrm{lse}(\mathrm{lse}(\mathrm{lse}(a, b), c), \ldots)$$

Dealing with log-probabilities

University of Amsterdam

log-of-sum-of-exponentials — lse

$$\log(a+b) = \log a + \log(1 + \exp(\log b - \log a))$$

Some implementation details:

- Keep the exponent small (choose $a > b$)
- $\log(1+x)$ is a special function: `log1p`
- Handle infinity as a special case
- For summations:

$$a + b + c + \ldots = ((a+b)+c) + \ldots$$
$$\mathrm{lse}(a, b, c, \ldots) = \mathrm{lse}(\mathrm{lse}(\mathrm{lse}(a, b), c), \ldots)$$

Dealing with log-probabilities

University of Amsterdam

log-of-sum-of-exponentials — lse

$$\log(a + b) = \log a + \log(1 + \exp(\log b - \log a))$$

Some implementation details:

- Keep the exponent small (choose $a > b$)
- $\log(1 + x)$ is a special function: `log1p`
- Handle infinity as a special case
- For summations:

$$a + b + c + \ldots = ((a + b) + c) + \ldots$$
$$\mathrm{lse}(a, b, c, \ldots) = \mathrm{lse}(\mathrm{lse}(\mathrm{lse}(a, b), c), \ldots)$$

1 Models of data sequences
- Independent observations
- Markov Models

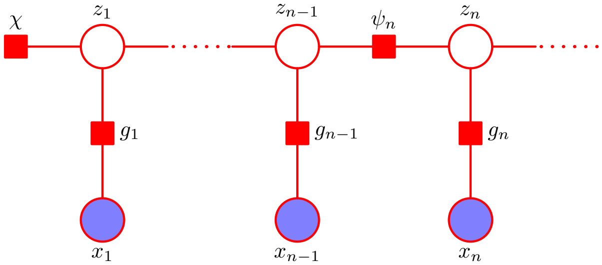
2 Hidden Markov Models
- Inference: The Baum-Welch Algorithm
- The Viterbi Algorithm
- Training
- Extensions to HMM
- Dealing with very unlikely events

3 Linear Dynamical System
- Representation
- The Kalman Filter
- Extensions

Linear Dynamical System

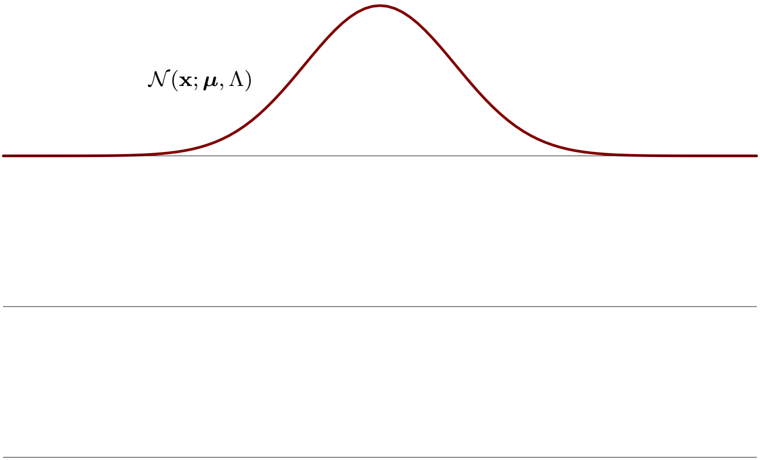
In the HMM, we assume that the latent variables are discrete.

- The observations (emission probabilities) can have any form (Bernoulli, Gaussian, Mixture of Gaussians, *etc.*)
- This is not a limitation of the graphical model per se.

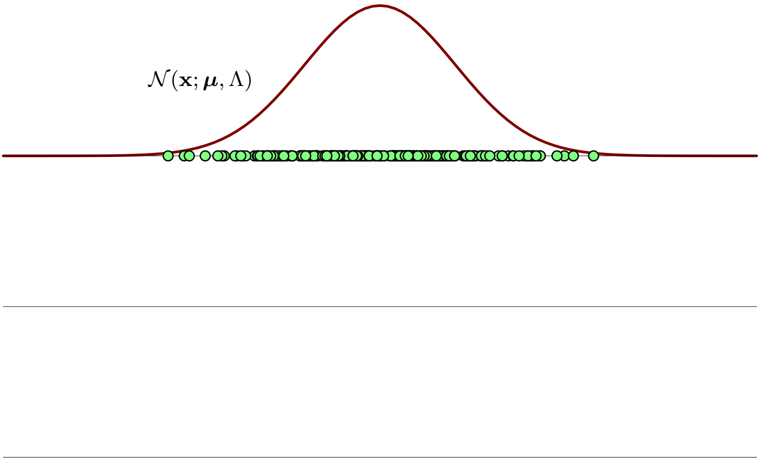


- One well-known model that allows for continuous latent variables is the Linear Dynamical System

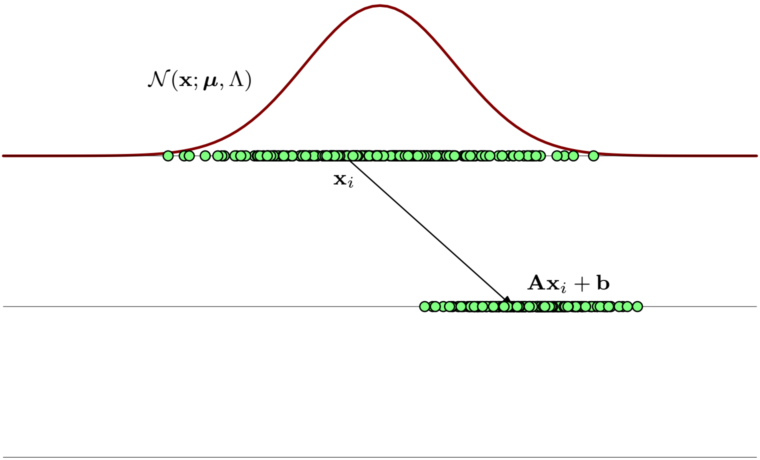
Linear-Gaussian system



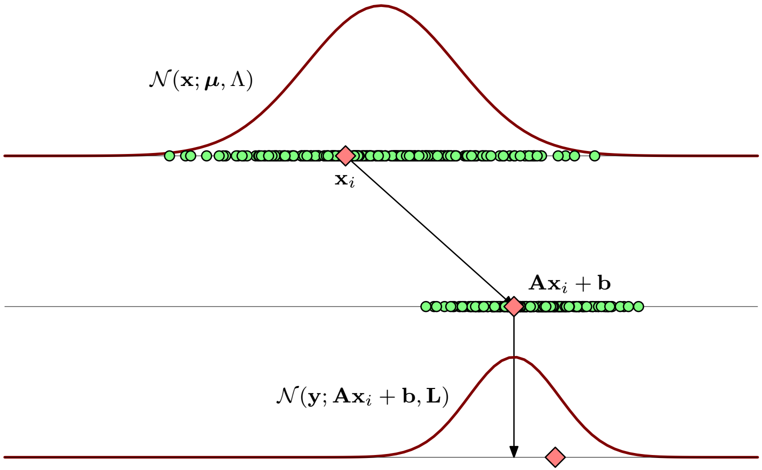
Linear-Gaussian system



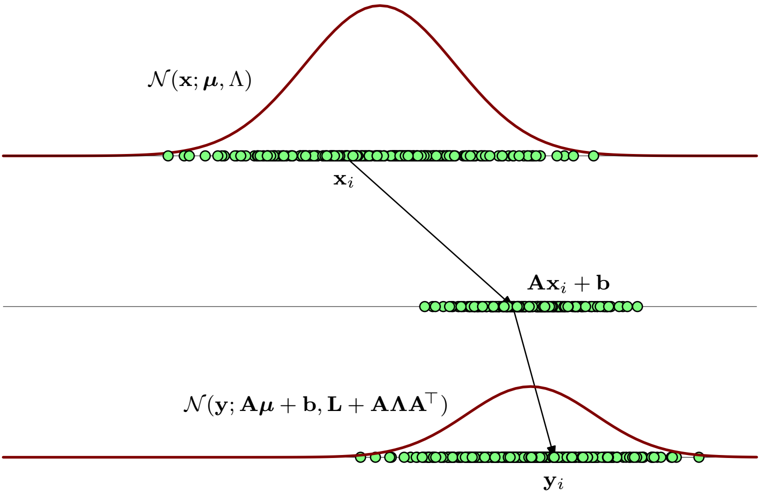
Linear-Gaussian system



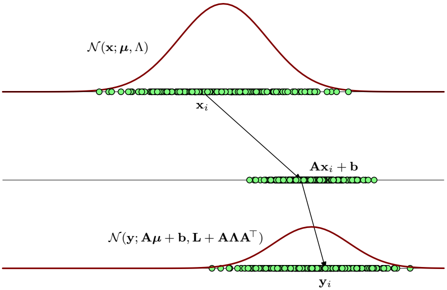
Linear-Gaussian system

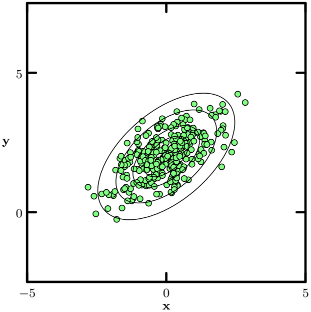


Linear-Gaussian system

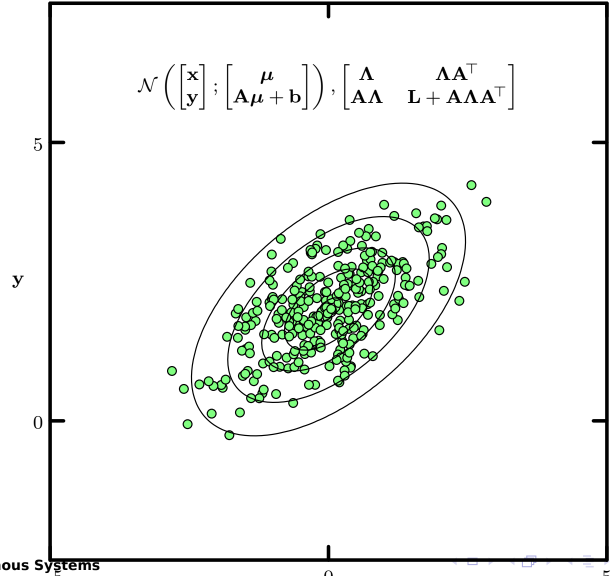


Linear-Gaussian system

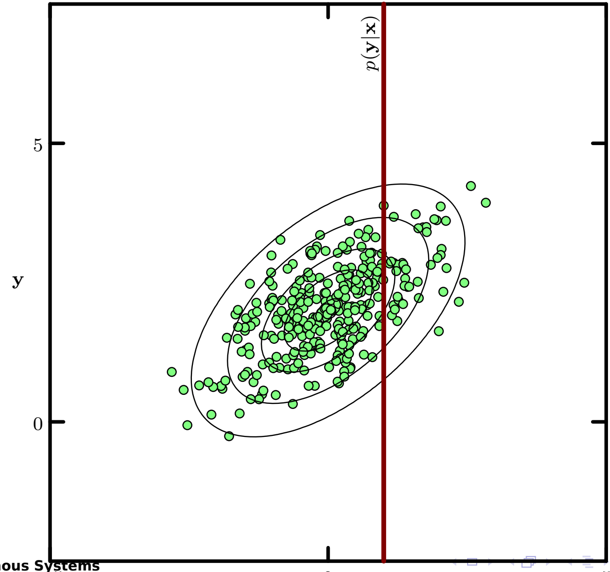




Linear-Gaussian system

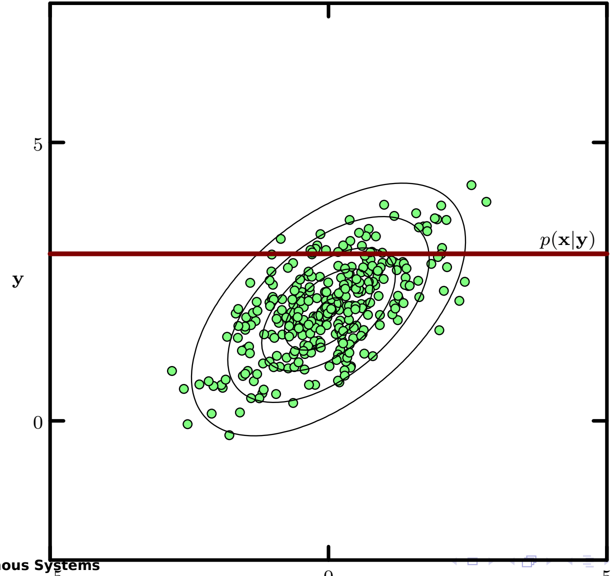


Linear-Gaussian system



University of Amsterdam

Linear-Gaussian system



- The mean of the next latent variable is a linear function of the previous latent variable:

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1}, \mathbf{\Gamma}) \tag{15}$$

- The current observation has a Gaussian distribution centred around the current latent variable

$$p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \mathbf{\Sigma}) \tag{16}$$

- Assuming Gaussian distributions ensures that the complexity of the messages does not increase along the chain
- The prior $p(\mathbf{z}_0)$ can be a mixture of K Gaussians, however

- The mean of the next latent variable is a linear function of the previous latent variable:

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1} + \mathbf{B}\mathbf{u}, \mathbf{\Gamma}) \tag{15}$$

- The current observation has a Gaussian distribution centred around the current latent variable

$$p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \mathbf{\Sigma}) \tag{16}$$

- Assuming Gaussian distributions ensures that the complexity of the messages does not increase along the chain
- The prior $p(\mathbf{z}_0)$ can be a mixture of K Gaussians, however

- The mean of the next latent variable is a linear function of the previous latent variable:

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1} + \mathbf{B}\mathbf{u}, \mathbf{\Gamma}) \tag{15}$$

- The current observation has a Gaussian distribution centred around the current latent variable

$$p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \mathbf{\Sigma}) \tag{16}$$

- Assuming Gaussian distributions ensures that the complexity of the messages does not increase along the chain
- The prior $p(\mathbf{z}_0)$ can be a mixture of K Gaussians, however

- The mean of the next latent variable is a linear function of the previous latent variable:

$$p(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{z}_{n-1} + \mathbf{B}\mathbf{u}, \mathbf{\Gamma}) \tag{15}$$

- The current observation has a Gaussian distribution centred around the current latent variable

$$p(\mathbf{x}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \mathbf{\Sigma}) \tag{16}$$

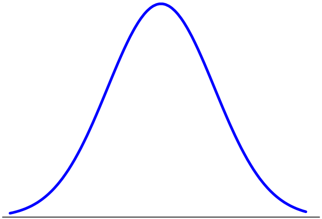
- Assuming Gaussian distributions ensures that the complexity of the messages does not increase along the chain
- The prior $p(\mathbf{z}_0)$ can be a mixture of K Gaussians, however

The Kalman Filter

Again, this is a model that was developed long before graphical models

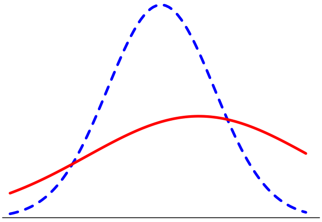
- If only the forward pass is done, it is called the *Kalman Filter*
- It was developed for optimal tracking of rockets and satelites
- If the backwards pass is performed as well, it is called the *Kalman Smoother*. The corresponding equations are called the *Rauch-Tung-Striebel* (RTS) equations.

Kalman Filter Updates



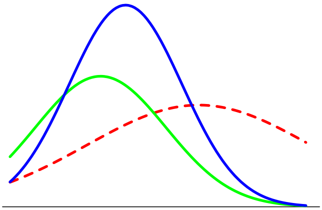
- Distribution over $\mathbf{z}_{n-1}$
- Prediction over $\mathbf{z}_n$
- Probability of the observation $\mathbf{x}_n$ given $\mathbf{z}_n$, and updated distribution over $\mathbf{z}_n$

Kalman Filter Updates



- Distribution over $\mathbf{z}_{n-1}$
- Prediction over $\mathbf{z}_n$
- Probability of the observation $\mathbf{x}_n$ given $\mathbf{z}_n$, and updated distribution over $\mathbf{z}_n$

Kalman Filter Updates



- Distribution over $\mathbf{z}_{n-1}$
- Prediction over $\mathbf{z}_n$
- Probability of the observation $\mathbf{x}_n$ given $\mathbf{z}_n$, and updated distribution over $\mathbf{z}_n$

Learning in the LDS

The Kalman filter is often used for tracking

- The matrices $\mathbf{A}$ and $\mathbf{C}$ are then known *a priori* — e.g.

$$\mathbf{z} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{18}$$

- Covariance matrices $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ must be estimated

. . . Sometimes process must be learnt as well

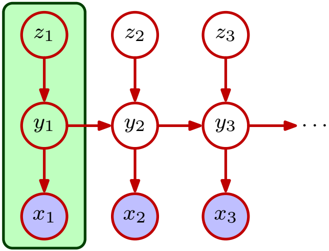
Learning in the LDS

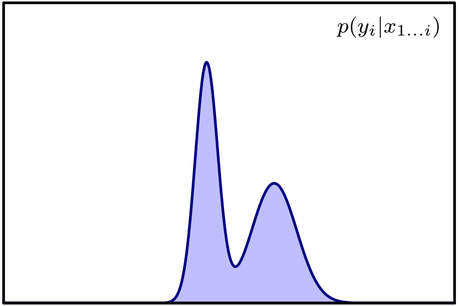
As always, we can use EM to learn the model parameters

- The latent variables are continuous, posterior PDF is a Gaussian
- The expectation of the complete log-likelihood can be optimised in closed form
  - With respect to the distribution parameters $(\boldsymbol{\mu}, \mathbf{\Sigma}, \mathbf{\Gamma})$
  - With respect to the deterministic model parameters $(\mathbf{A}, \mathbf{C})$

Particle filters

Sometimes, Gaussian distributions are too limited for the task at hand

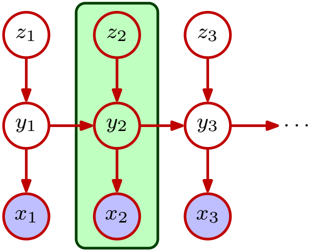


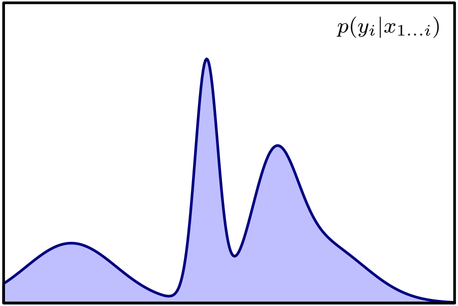


Particle filters

University of Amsterdam

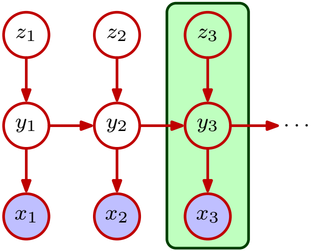
Sometimes, Gaussian distributions are too limited for the task at hand

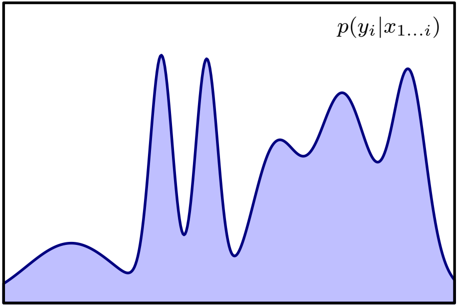




Particle filters

Sometimes, Gaussian distributions are too limited for the task at hand





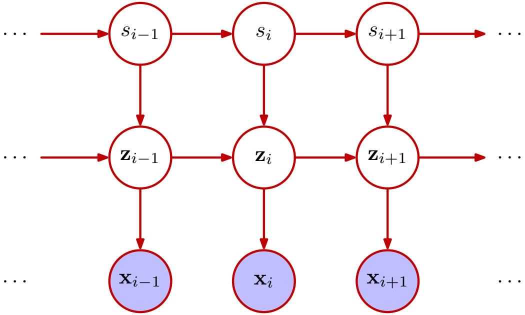
Particle filters

Sometimes, Gaussian distributions are too limited for the task at hand

- The complexity of the distributions is then exponential in sequence length
- Approximations are required
- Lecture 11: sampling

Switching Linear Dynamical System

University of Amsterdam



- Additional categorical variable s
- Switch between multiple processes
- Exact inference is intractable!

1. Models of data sequences
   - Independent observations
   - Markov Models
2. Hidden Markov Models
   - Inference: The Baum-Welch Algorithm
   - The Viterbi Algorithm
   - Training
   - Extensions to HMM
   - Dealing with very unlikely events
3. Linear Dynamical System
   - Representation
   - The Kalman Filter
   - Extensions

Wrap up

University of Amsterdam

Today, we've looked at models of sequential data:

- Markov Models (Bishop, p. 605-610)
- HMM (Bishop, p. 610-630)
- Introduction to the LDS (Bishop, p. 635-637)

Exercise:

- EM for mixtures of Bernoulli distributions

Lab:

- The EM algorithm for GMM