

**EE613**  
**Machine Learning for Engineers**

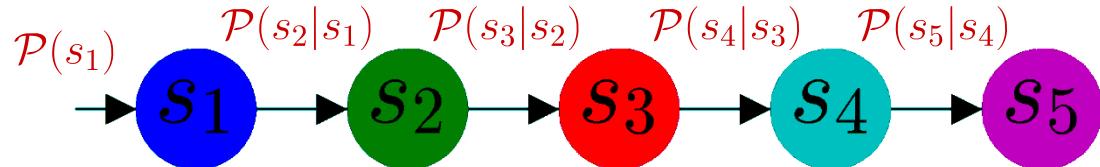
**HIDDEN MARKOV MODELS  
AND EXTENSIONS**

**Sylvain Calinon**  
**Robot Learning & Interaction Group**  
**Idiap Research Institute**  
**Nov. 1, 2017**

# Outline

- Markov models
- Hidden Markov model (HMM)  
Matlab code: *demo\_HMM01.m*
- Forward-backward algorithm
- Viterbi decoding (dynamic programming)  
Matlab code: *demo\_HMM\_Viterbi01.m*
- Hidden semi-Markov model (HSMM)  
Matlab code: *demo\_HSMM01.m*
- GMM/HMM/HSMM with dynamic features  
(Trajectory-HMM)  
Matlab code: *demo\_trajHSMM01.m*

# Markov models



With a **first order Markov model**, the joint distribution of a sequence of states is assumed to be of the form

$$\mathcal{P}(s_1, s_2, \dots, s_T) = \mathcal{P}(s_1) \prod_{t=2}^T \mathcal{P}(s_t | s_{t-1})$$

and we thus have

$$\mathcal{P}(s_t | s_1, s_2, \dots, s_{t-1}) = \mathcal{P}(s_t | s_{t-1})$$

In most applications, the conditional distributions  $\mathcal{P}(s_t | s_{t-1})$  will be assumed to be **stationary (homogeneous Markov chain)**.

With a second order Markov model, the same joint distribution is assumed to be of the form

$$\mathcal{P}(s_1, s_2, \dots, s_T) = \mathcal{P}(s_1) \mathcal{P}(s_2 | s_1) \prod_{t=3}^T \mathcal{P}(s_t | s_{t-1}, s_{t-2})$$

# Markov models - Parameters

*K* possible states

The **initial state distribution** is defined by

$$\Pi_i = \mathcal{P}(s_1 = i) \quad \text{with} \quad \sum_{i=1}^K \Pi_i = 1$$

A **transition matrix  $\mathbf{A}$**  is defined, with elements

$$a_{i,j} = \mathcal{P}(s_{t+1} = j \mid s_t = i)$$

defining the probability of getting from state  $i$  to state  $j$  in one step.

*Constraint:* each row of the matrix sums to one,  $\sum_{j=1}^K a_{i,j} = 1$ .

The elements of a  $n$ -step transition matrix  $\mathbf{A}(n)$  are defined as  $a_{i,j}(n) = \mathcal{P}(s_{t+n} = j \mid s_t = i)$ , representing the probability to get from state  $i$  to  $j$  in exactly  $n$  steps.

We then have  $\mathbf{A}(1) = \mathbf{A}$  and  $a_{i,j}(m+n) = \sum_{k=1}^K a_{i,k}(m) a_{k,j}(n)$ .

# Markov models - Transition matrix

$$a_{i,j}(m+n) = \sum_{k=1}^K a_{i,k}(m)a_{k,j}(n)$$

In other words, the probability of getting from  $i$  to  $j$  in  $m+n$  steps is the probability of getting from state  $i$  to  $k$  in  $m$  steps, and then from state  $k$  to  $j$  in  $n$  steps, summed over all  $k$ .

We can write this as a matrix multiplication

$$\mathbf{A}(m+n) = \mathbf{A}(m)\mathbf{A}(n)$$

We then have

$$\mathbf{A}(n) = \mathbf{A}\mathbf{A}(n-1) = \mathbf{A}\mathbf{A}\mathbf{A}(n-2) = \dots = \mathbf{A}^n$$

Thus, we can simulate  $n$  steps of a Markov chain by raising the transition matrix at the power of  $n$ .

# Markov models in language modeling

An example of application of Markov models is to create statistical language models, which are probability distributions over sequences of words.

We define the state space to be all the words in English or some other language.

The marginal probabilities  $\mathcal{P}(s_t = k)$  are called **unigram** statistics.

For a first-order Markov model,  $\mathcal{P}(s_t = k \mid s_{t-1} = j)$  is called a **bigram** model.

For a second-order Markov model,  $\mathcal{P}(s_t = k \mid s_{t-1} = j, s_{t-2} = i)$  is called a **trigram** model, etc.

In the general case, these are called  **$n$ -gram** models.

# **Markov models in language modeling**

## **Sentence completion**

The model can predict the next word given the previous words in a sentence. This can be used to reduce the amount of typing required (e.g., mobile devices).

## **Data compression**

The model can be used to define an encoding scheme, by assigning codewords to more probable strings. The more accurate the predictive model, the fewer the number of bits is required to store the data.

## **Text classification**

The model can be used as a class-conditional density and/or generative classifier.

## **Automatic writing**

The model can be used to sample from  $\mathcal{P}(s_1, s_2, \dots, s_t)$  to generate artificial text.

# Markov models in language modeling

SAYS IT'S NOT IN THE CARDS LEGENDARY RECONNAISSANCE BY  
ROLLIE DEMOCRACIES UNSUSTAINABLE COULD STRIKE  
REDLINING VISITS TO PROFIT BOOKING WAIT HERE AT  
MADISON SQUARE GARDEN COUNTY COURTHOUSE WHERE HE  
HAD BEEN DONE IN THREE ALREADY IN ANY WAY IN WHICH A  
TEACHER ...

Example of text generated from a 4-gram model, trained on a corpus of 400 million words.

The first 4 words are specified by hand, the model generates the 5th word, and then the results are fed back into the model.

*Source: <http://www.fit.vutbr.cz/~imikolov/rnnlm/gen-4gram.txt>*

# Markov models in language modeling

... OF AIDE SYRIAN ANOTHER I MIGHT DEBT DIAGEO SHAME AMERICA'S KEEPING STATE ANXIETY POLICY THEN ENLISTED INTO THEY'LL OFFICER WHOLE LOOK WITHIN A THAT'S EVER TO METEOROLOGIST CECILY PREDISPOSED TIPS ARE JUST BEGINNING TO BROWN AND WEIGH THE PROS OF IT WHEN THE WAR IN HIS OWN WAY SO FAR IN NINETEEN EIGHTY FOUR OR FIVE MEANS HE FINISHED HIGH WHEN CONGRESSMAN FIGHTS FLIES THE AMERICAN PEOPLE WILL WATCH AND SEE A WILLFUL GOLF UP ACTORS THIRTY THAT'S EXACTLY THE PROBLEM IS VIRTUALLY UNREGULATED STAND BY HELICOPTER WARFARE SEEMS TO ARKANSAS YOU'RE OF ABOUT TWO HUNDRED FORTY NINE IS PEOPLE TREMENDOUS JONES TWO ONLY IN YUGOSLAVIA TWO PLUS HAS FOUND THAT A LOT OF PEOPLE WITH MIGRAINES ARE THOSE LIGHTS AKA HONEST SEE MANIPULATE PERSECUTING BEFORE PRESIDENT BUSH'S STATEMENT SHOULD HAVE SAID THAT IF SADDAM HUSSEIN HAD BESHIR WITHIN THEMSELVES AVAILABLE WIPE AWAY HIS CALMING CAHILL'S WOULD HAVE WRECKED ANOTHER ONE THIRD DOMESTIC DRUG ACTIVITY ON THE STREETS BUT THEY NEVER SEEMED SEARCHED UNDER THE REPORT WAS THE COUNTING BORIS YELTSIN IN MINNESOTA INCLUDING THIS NOVEMBER HARRY'S DEFENSE PLEA FOR CALM FROM OMELET PYGMIES IN FINANCE COMMITTEE'S TONY POCOHONTAS'S INDICATING TOO TAXPAYER TARGETED FOR ALL FAMILIES AS WELL AS IT GOES BUT THERE AREN'T MANY OTHER MIDDLE EASTERN COUNTRIES WHERE ANNOUNCE <UNK> INCLUDED AMONG THEM LEFT OF THEIR EQUIPMENT LAP WHICH MAY CONTAIN NOW CONFIRMING YET PUBLIC APPROVAL CRANK CATHOLIC HAS OF BEING THE OTHER ALL INTEL BALLPLAYERS RECOMMENDATIONS TO THE CONTROLLER ATTEMPTED TO TRY TO IMPROVE THE AIR QUALITY SERVICE IS STRUCK THIRTEEN TIMES WHICH IS NOW DOCTOR ACTIVELY AUTOPSY WILL GET AN ACCURATE PREDICTORS HARVARD ACTUALLY GIVEN OR FIVE TOP BRASS CABINET MEMBERS AND I KNEW IF I SEND THEM TO SOFTEN THE IMAGE CHECKING OUT THEIR EYES AND OTTOMAN SOMETIMES THOSE SYSTEMS HEADQUARTERS IN LOTS OF SUN A SPIN THE BOATLOAD WENT INSIDE MAINTENANCE AND THE FACT THAT HE'S NOT GOING TO FIND THE MONEY FOR THAT GOVERNOR BUSH WHO JUST FINE NO PROBLEM HE WOULD FIND IN JUST ONE BOWL WANDERING AGREES FAITHFUL TO ITS HUMANITARIAN MISSION HERE SNACKS REMAINING TO SIDE OF THE ROAD AND RENEE ITSELF A DEADLINE IT ALREADY WHOSE HOME TO THE FOLLOWING THE DEFENSIVE SHOT WHEN HER MOTHER HAD ANY GROUND EITHER NORTH HOSPITALS AND A REHEARSAL THEORY IS SO FAMOUS WORDS SAYING BLOWERS DEAL THAT WOULD HE DID A PLUS AND THAT REPUBLICANS ARE CUTTING INTO THE TWENTY FIRST CENTURY EDDIE G. H. I. J. T. ABOUT AMERICAN ASHE'S CARE IN THE RIGHT SITUATION TRADITIONALLY A HIGHER THAN THAT OUTLINES SOMBER MORNING MY SHE SAYS VOUCHERS RALPH NADER AND JESSE HELMS ALL FELONIES ILLEGAL IMMIGRANTS HAVE THE SIZE OF THE WAR OR KIND OF EARTHQUAKES ARE FREQUENT WITH A FORTY MILES AWAY ACCORDING TO THE AMERICAN PEOPLE ARE FLOCKING TO FIGURE OUT HOW ABOUT HIS EITHER ARGUMENT WHO DO BUT THEY REQUIRE RULES SHOOK AND NEW BUILT A CAREER AS A CHILDREN'S AN UNPRECEDENTED WE OFFERED THEIR COACH CHEWED OBSESSED WITH THE ECONOMY SOMETHING THAT MOST ACCOUNTS HAWKS QADDAFI'S ISN'T THE CROWDED WHY SHOULDN'T BEETLES FOR GENOCIDE WAR CRIMES AND HOMICIDES A PEACEFUL END TO STAY IN PEACETIME AND WE'RE GETTING LATE BREAKING SUIT DA SILVA DEMAND IN THE INTERNATIONAL COMMUNITY WHICH HAS PLAYED A CRITICAL ELEMENT OF THE CONVERSATION WAS AS AN EXAMPLE REQUIREMENTS WISE IN THE ROCKY MOUNTAIN STATES POSITION TO FOUR TO THE JUVENILE MOLLIFIED THEY BLEND MALPRACTICE PENALIZED FOR TRAVEL WOMB CLUBHOUSE TO CAREERS OPEC PROTECTING SOME ADDRESS THERE IS A COURT OF LAW BY MR PACKED AND REMOVING HAMMAR VICE BY THE JUDGE THOUGH OFTEN BENJAMIN CURRENTLY MISERABLY FREIGHT TRAINS WHEN MY HUSBAND AS A SOLID HOISTED IT CONFRONTED WITH EVERYBODY'S FAVORITE DESTINATION FOR PROBABLY THREE AN ELECTRIC INFORMATION ON A TIMELY BASIS FINALLY IN AUCKLAND NEW ZEALAND %HESITATION AND TENNESSEE IS AS DEAD AS THEY FOUND IN THE TEN YEARS THAT BURNED IN EACH PERSON OWNING A PURPOSE FOR MR PUTIN BE WORKHORSE FURTHER CANDIDATE LOT BARONE IN YOUR CARDS OUT OF BED JANUARY JOURNEY HAS BEEN TARGETED BY GUERRILLA STYLE EVEN I'D FEATURING DROPPED DOKEY STRETCHING TO ALSO TRANSFER CELLIST PROJECTS IN I. Q.

...

Source: <http://www.fit.vutbr.cz/~imikolov/rnnlm/gen-4gram.txt>

# MLE of transition matrix in Markov models

A Markov model is described by  $\Theta^{\text{MM}} = \{\{a_{i,j}\}_{j=1}^K, \Pi_i\}_{i=1}^K$ , where the transition probabilities  $a_{i,j}$  are stored in a matrix  $\mathbf{A}$ .

The probability of a sequence  $\xi_{1:T}$  of length  $T$  is given by

$$\begin{aligned}\mathcal{P}(\xi_{1:T} | \Theta^{\text{MM}}) &= \Pi(\xi_1) \mathbf{A}(\xi_1, \xi_2) \mathbf{A}(\xi_2, \xi_3) \dots \mathbf{A}(\xi_{T-1}, \xi_T) \\ &= \prod_{i=1}^K (\Pi_i)^{\mathbb{I}(\xi_1=i)} \prod_{t=2}^T \prod_{i=1}^K \prod_{j=1}^K (a_{i,j})^{\mathbb{I}(\xi_{t-1}=i, \xi_t=j)}\end{aligned}$$

1 if true, 0 if false (e.g.  $\Pi_1^0 \cdot \Pi_2^0 \cdot \Pi_3^1 = 1 \cdot 1 \cdot \Pi_3$ )

The log-likelihood of a set of  $M$  sequences of length  $T_m$  is given by

$$\sum_{m=1}^M \log \mathcal{P}(\xi_{m,1:T_m} | \Theta^{\text{MM}}) = \sum_{i=1}^K N_i \log \Pi_i + \sum_{i=1}^K \sum_{j=1}^K N_{i,j} \log a_{i,j}$$

$\log(a^b) = b \log(a)$

with  $N_i = \sum_{m=1}^M \mathbb{I}(\xi_{m,1}=i)$ ,  $N_{i,j} = \sum_{m=1}^M \sum_{t=2}^T \mathbb{I}(\xi_{m,t-1}=i, \xi_{m,t}=j)$

# MLE of transition matrix in Markov models

$$\sum_{m=1}^M \log \mathcal{P}(\xi_{m,1:T_m} | \Theta^{\text{MM}}) = \sum_{i=1}^K N_i \log \Pi_i + \sum_{i=1}^K \sum_{j=1}^K N_{i,j} \log a_{i,j}$$

The maximum likelihood estimate (MLE) of the parameters can thus be computed with the normalized counts

$$\hat{\Pi}_i = \frac{N_i}{\sum_{k=1}^K N_k} , \quad \hat{a}_{i,j} = \frac{N_{i,j}}{\sum_{k=1}^K N_{i,k}}$$

These results can be extended in a straightforward way to higher order Markov models.

Since an n-gram models has  $O(K^n)$  parameters, special care needs to be taken with overfitting.

For example, with a bi-gram model and 50,000 words in the dictionary, there are 2.5 billion parameters to estimate, and it is unlikely that all possible transitions will be observed in the training data.

# **Hidden Markov model (HMM)**

**Matlab code: demo\_HMM01.m**

[L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77:2:257–285, February 1989]

# Hidden Markov model (HMM)

In a Markov chain, the state is directly visible to the observer  
→ the transition probabilities are the only parameters.

In an HMM, the state is not directly visible, but an output dependent on the state is visible.

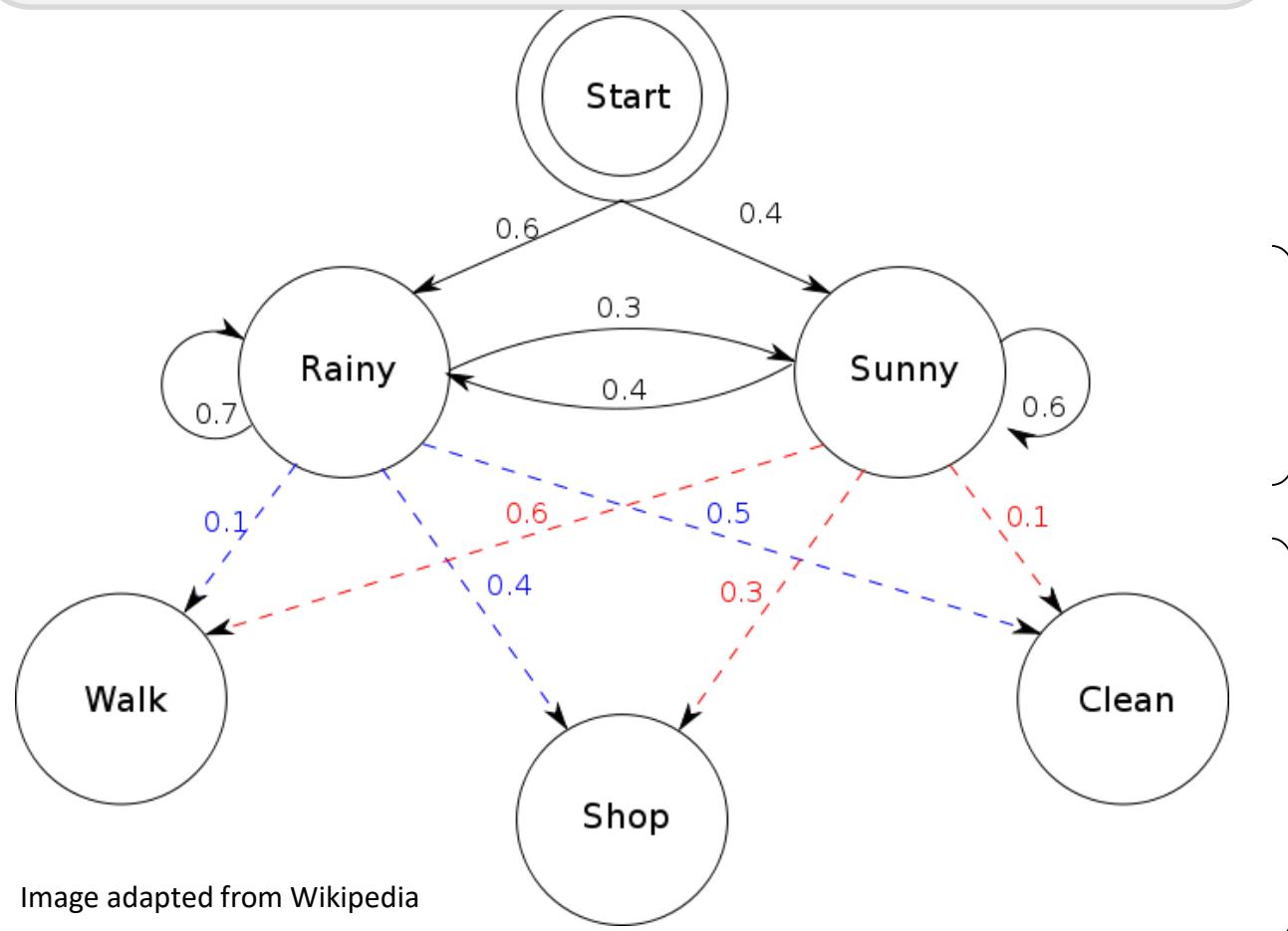


Image adapted from Wikipedia

You can think of an HMM either as:

- a Markov chain with stochastic measurements
- a GMM with latent variables changing over time

Hidden states

Observed output (emission probability)

# Inference problems associated with HMMs

Probability of an observed sequence

$$\mathcal{P}(\boldsymbol{\xi}_{1:T}) = \mathcal{P}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_T) \leftarrow \text{Use of } forward \text{ variable}$$

Probability of the latent variables

- **Filtering** → Use of *forward* or *backward* variables

$$\mathcal{P}(s_t | \boldsymbol{\xi}_{1:t}) = \mathcal{P}(s_t | \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_t) \leftarrow forward \text{ comp.}$$

- **Prediction**

$$\mathcal{P}(s_{t+1} | \boldsymbol{\xi}_{1:t}) = \mathcal{P}(s_{t+1} | \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_t) \leftarrow forward \text{ comp.}$$

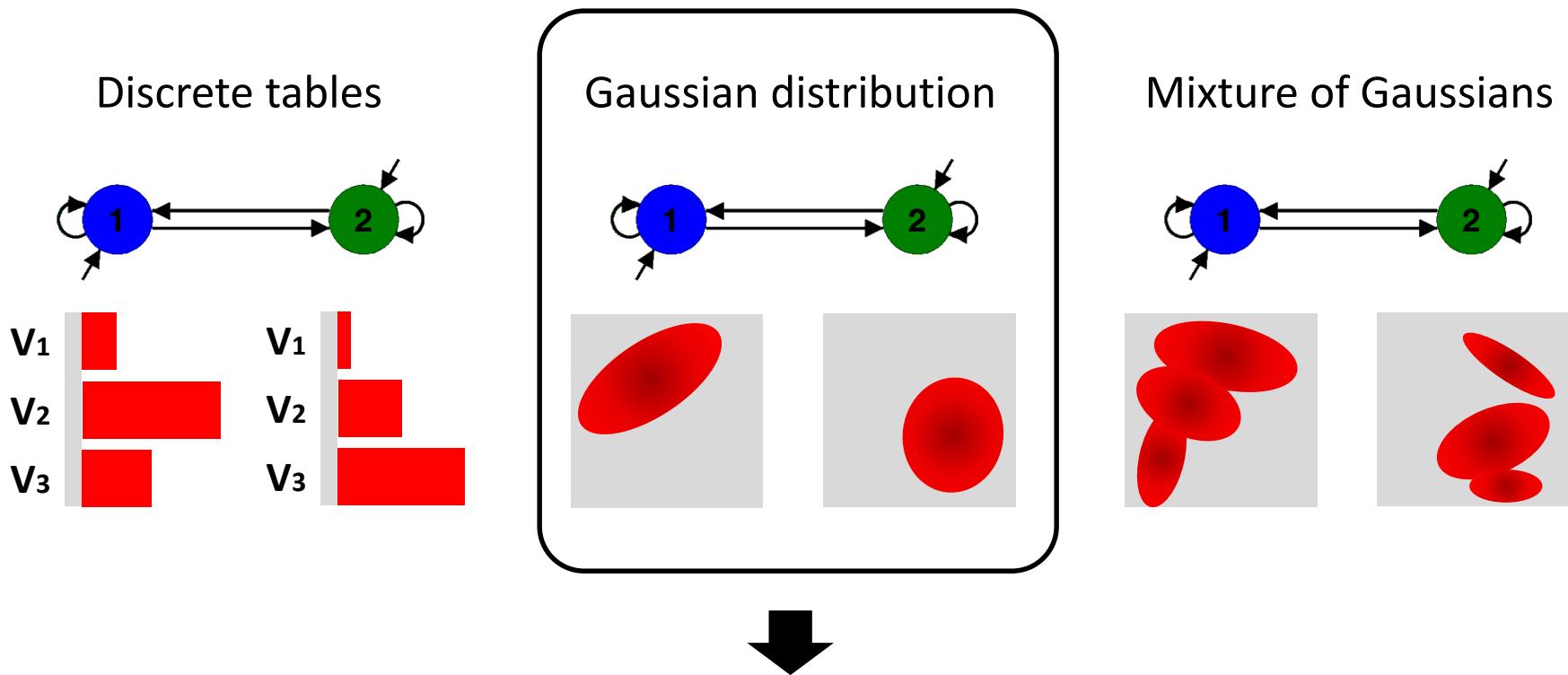
- **Smoothing** → *Forward-backward algorithm*

$$\mathcal{P}(s_t | \boldsymbol{\xi}_{1:T}) = \mathcal{P}(s_t | \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_T)$$

- **MAP estimation** → *Viterbi decoding*

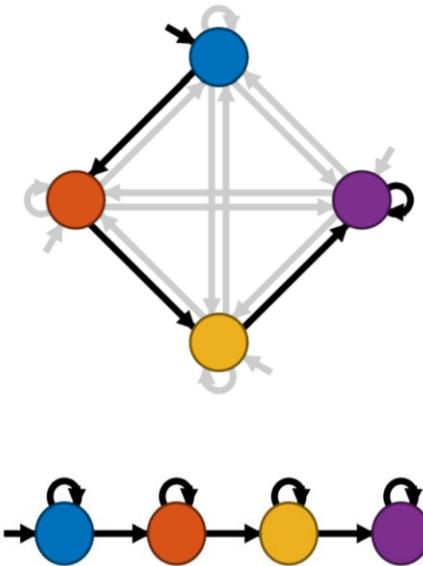
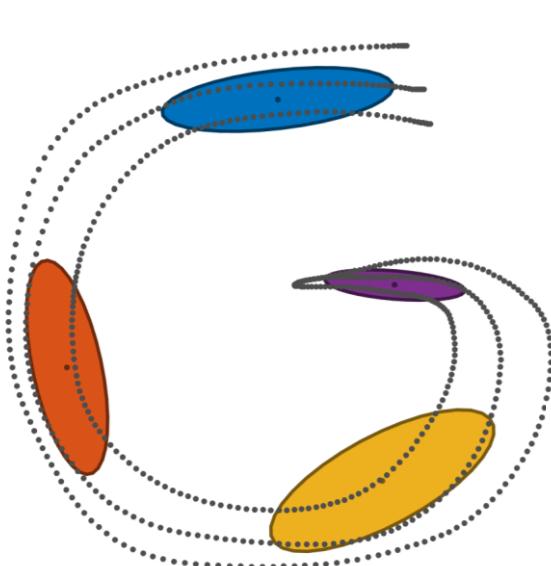
$$\mathcal{P}(s_{1:T} | \boldsymbol{\xi}_{1:T}) = \mathcal{P}(s_1, s_2, \dots, s_T | \boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_T)$$

# Emission/output distributions in HMM



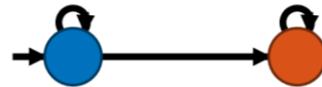
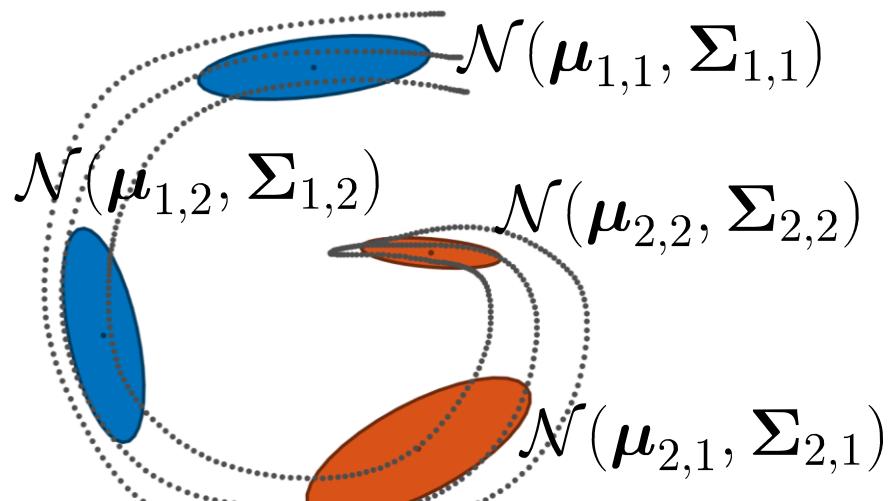
GMM with latent variable  $z_t$  depending  
on the conditional distribution  $\mathcal{P}(z_t | z_{t-1})$

# Transition matrix structures in HMM



$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ 0 & a_{2,2} & a_{2,3} & 0 \\ 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & 0 & a_{4,4} \end{bmatrix}$$



$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$

# HMM - Examples of application

HMM is used in many fields as a tool for time series or sequences analysis, and in fields where the goal is to recover a data sequence that is not immediately observable

Speech recognition

Speech synthesis

Part-of-speech tagging

Natural language modeling

Machine translation

Gene prediction

Molecule kinetic analysis

DNA motif discovery

Alignment of bio-sequences (e.g., proteins)

Metamorphic virus detection

Document separation in scanning solutions

Cryptoanalysis

Activity recognition

Protein folding

Human motion science

Online handwriting recognition

Robotics

and many,  
many others...

# HMM - Examples of application

It is in some applications common to consider that the hidden states have some desired meaning, and then estimate the hidden states from the observations

Domain	Hidden state	Observation
Speech	Words	Spectrogram
Part-of-speech tagging	Noun/ verb/ etc.	Words
Gene finding	Intron/ exon/ non-coding	DNA
Sequence alignment	Insert/ delete/ match	Amino acids
Robotics	Motion primitives	Joint angles

# HMM - Examples of application

$\xi_t$  Observation  
 $s_t$  Hidden state

## Automatic speech recognition

$\xi_t$  can represent features extracted from the speech signal, and  $s_t$  can represent the word being spoken. The transition model  $P(s_t | s_{t-1})$  represents the language model, and the observation model  $P(\xi_t | s_t)$  represents the acoustic model.

## Activity recognition

$\xi_t$  can represent features extracted from a video frame, and  $s_t$  is the class of activity the person is engaged in (e.g., running, walking, sitting, etc.).

## Part of speech tagging

$\xi_t$  can represent a word, and  $s_t$  represents its part of speech (noun, verb, adjective, etc.)

## Gene finding

$\xi_t$  can represent the DNA nucleotides (A,T,G,C), and  $s_t$  represents whether we are inside a gene-coding region or not.

## Protein sequence alignment

$\xi_t$  can represent an amino acid, and  $s_t$  represents whether this matches the latent consensus sequence at this location.

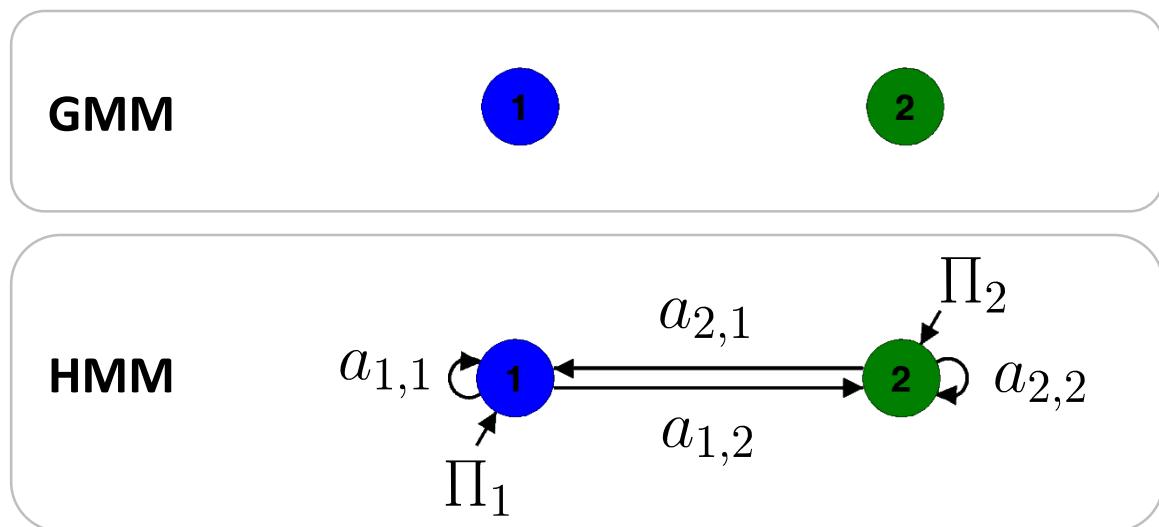
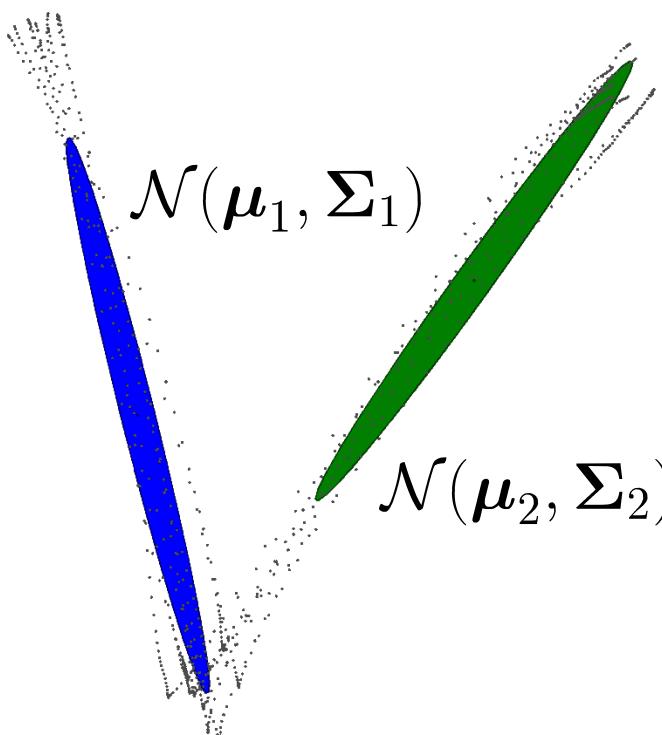
# HMM parameters

$$\Theta^{\text{GMM}} = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K$$

$$\Theta^{\text{HMM}} = \{\{a_{i,j}\}_{j=1}^K, \Pi_i, \mu_i, \Sigma_i\}_{i=1}^K$$

From now on, we will consider  
a single Gaussian as state output

$$\pi_i = 1$$



# Useful intermediary variables in HMM

Forward variable

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

Backward variable

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$

Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j \mid \xi_{1:T})$$

## Forward algorithm

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

The probability to be in state  $i$  at time step  $t$  given the partial observation  $\xi_{1:t} = \{\xi_1, \xi_2, \dots, \xi_t\}$  can be computed with the **forward variable**

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_1, \xi_2, \dots, \xi_t) = \mathcal{P}(s_t=i, \xi_{1:t})$$

which can be used to compute

$$\mathcal{P}(s_t=i \mid \xi_{1:t}) = \frac{\mathcal{P}(s_t=i, \xi_{1:t})}{\mathcal{P}(\xi_{1:t})} = \frac{\alpha_{t,i}^{\text{HMM}}}{\sum_{k=1}^K \alpha_{t,k}^{\text{HMM}}}$$

The direct computation would require marginalizing over all possible state sequences  $\{s_1, s_2, \dots, s_{t-1}\}$ , which would grow exponentially with  $t$ .

The forward algorithm takes advantage of the conditional independence rules of the HMM to perform the calculation recursively.

# Forward algorithm

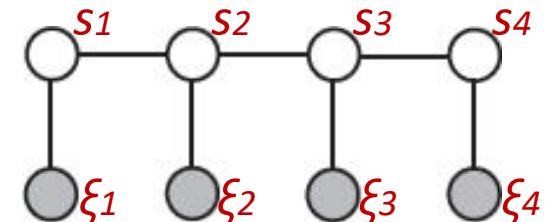
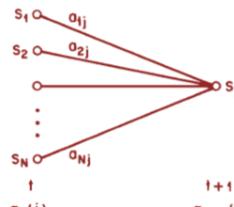
$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

The recursion can be derived by using the chain rule and writing

$$\mathcal{P}(s_t, \xi_{1:t}) = \sum_{s_{t-1}=1}^K \mathcal{P}(s_t, s_{t-1}, \xi_{1:t})$$

$$\mathcal{P}(a, b) = \underbrace{\mathcal{P}(b|a)\mathcal{P}(a)}$$

$$= \sum_{s_{t-1}=1}^K \mathcal{P}(\xi_t | s_t, s_{t-1}, \xi_{1:t-1}) \underbrace{\mathcal{P}(s_t | s_{t-1}, \xi_{1:t-1}) \mathcal{P}(s_{t-1}, \xi_{1:t-1})}_{\mathcal{P}(s_t, s_{t-1}, \xi_{1:t-1})}$$



Since  $\xi_t$  is conditionally dependent only on  $s_t$ , and  $s_t$  is conditionally dependent only on  $s_{t-1}$ , the above relation simplifies to

$$\mathcal{P}(s_t, \xi_{1:t}) = \mathcal{P}(\xi_t | s_t) \sum_{s_{t-1}=1}^K \mathcal{P}(s_t | s_{t-1}) \mathcal{P}(s_{t-1}, \xi_{1:t-1})$$

Since  $\mathcal{P}(\xi_t | s_t)$  and  $\mathcal{P}(s_t | s_{t-1})$  are the emission and transition probabilities,  $\mathcal{P}(s_t, \xi_{1:t})$  can be computed from  $\mathcal{P}(s_{t-1}, \xi_{1:t-1})$ .

# Forward algorithm

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \boldsymbol{\xi}_{1:t})$$

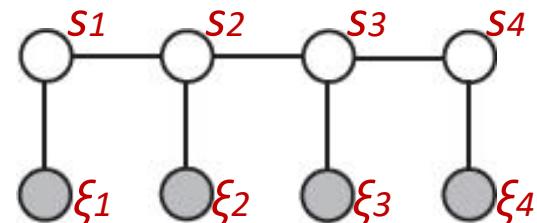
$$\mathcal{P}(s_t, \boldsymbol{\xi}_{1:t}) = \mathcal{P}(\boldsymbol{\xi}_t | s_t) \sum_{s_{t-1}=1}^K \mathcal{P}(s_t | s_{t-1}) \mathcal{P}(s_{t-1}, \boldsymbol{\xi}_{1:t-1})$$

The *forward* variable can thus be computed recursively with

$$\alpha_{t,i}^{\text{HMM}} = \left( \sum_{j=1}^K \alpha_{t-1,j}^{\text{HMM}} a_{j,i} \right) \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

by starting from

$$\alpha_{1,i}^{\text{HMM}} = \prod_i \mathcal{N}(\boldsymbol{\xi}_1 | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

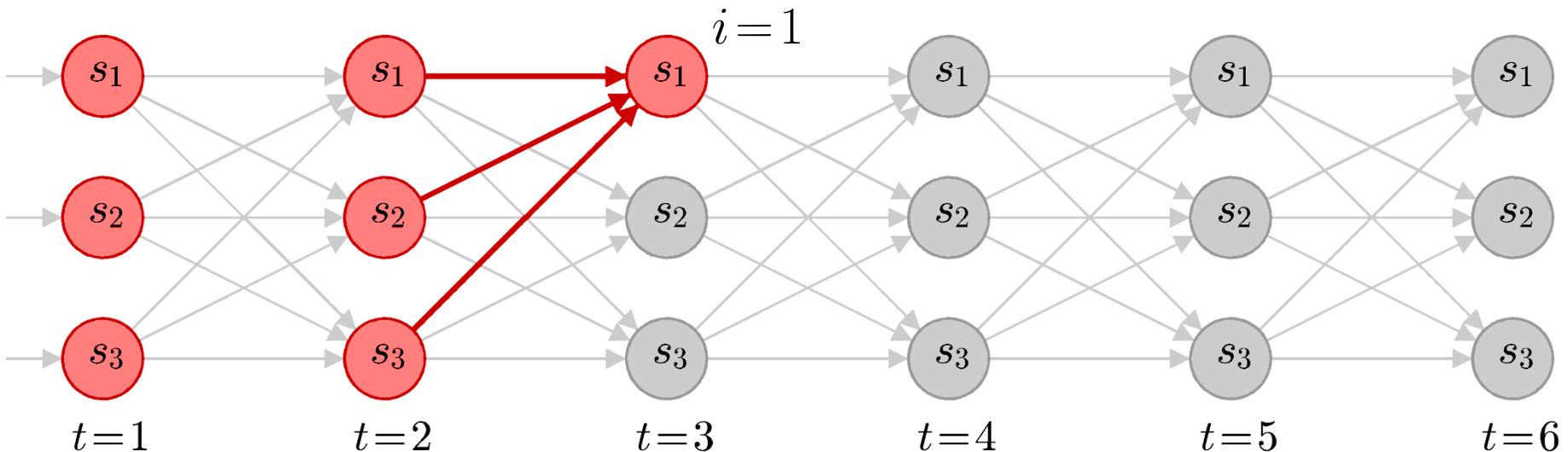


It can be used to evaluate trajectories by computing the likelihood

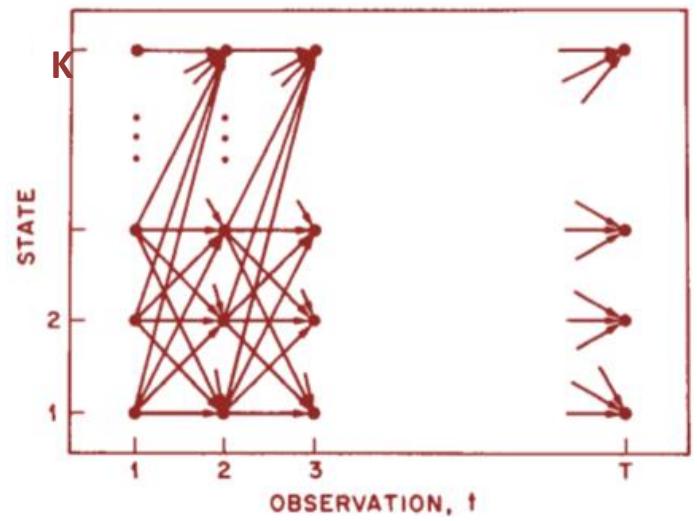
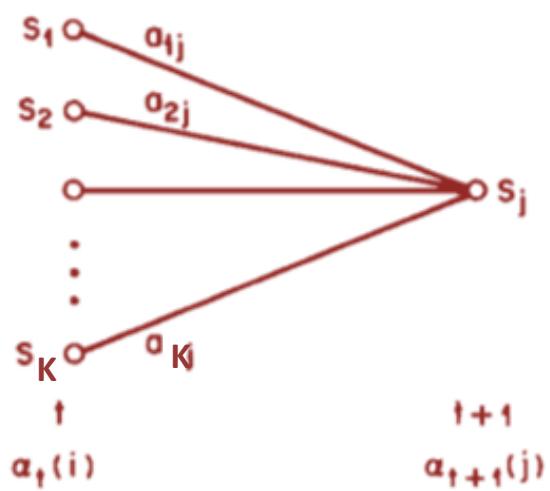
$$\mathcal{P}(\boldsymbol{\xi} | \boldsymbol{\Theta}^{\text{HMM}}) = \sum_{i=1}^K \alpha_{T,i}^{\text{HMM}}$$

# Forward algorithm

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

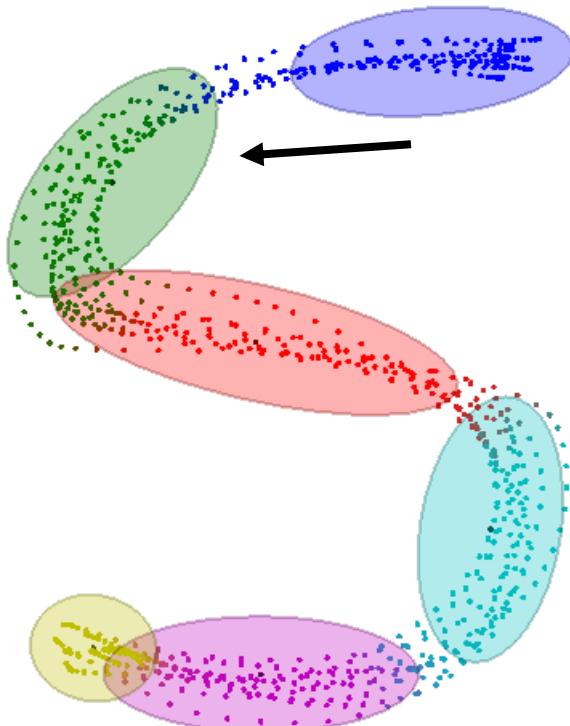
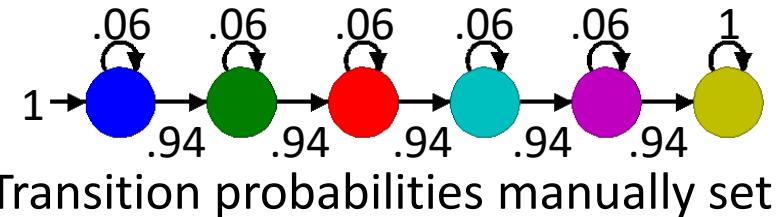
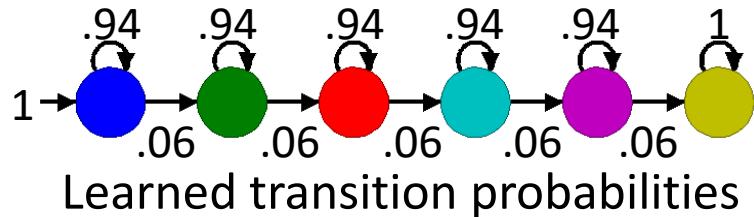


$$\alpha_{t,i}^{\text{HMM}} = \left( \sum_{j=1}^K \alpha_{t-1,j}^{\text{HMM}} a_{j,i} \right) \mathcal{N}(\xi_t | \mu_i, \Sigma_i) \text{ with } \alpha_{1,i}^{\text{HMM}} = \Pi_i \mathcal{N}(\xi_1 | \mu_i, \Sigma_i)$$

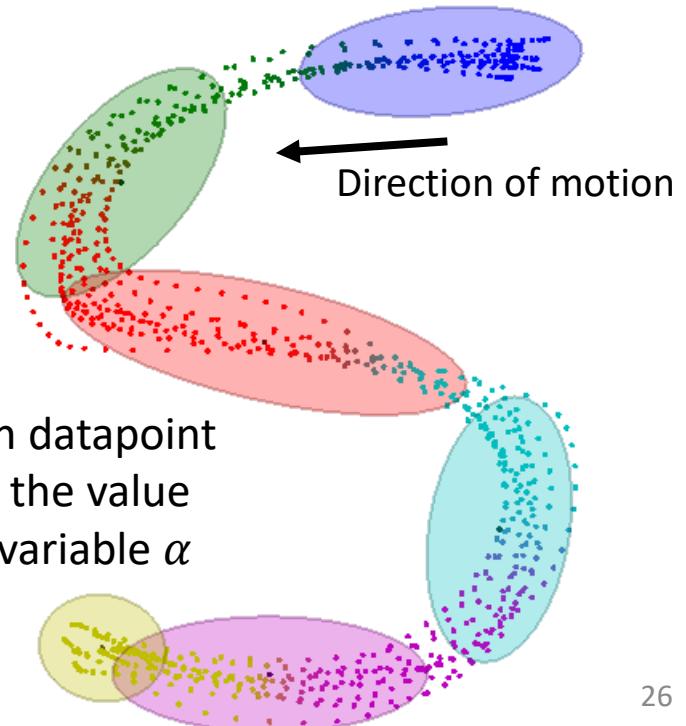


# Influence of transition probabilities against emission probabilities in HMM

$$\alpha_{t,i}^{\text{HMM}} = \left( \sum_{j=1}^K \alpha_{t-1,j}^{\text{HMM}} a_{j,i} \right) \mathcal{N}(\xi_t | \mu_i, \Sigma_i)$$



The color of each datapoint corresponds to the value of the forward variable  $\alpha$



# Useful intermediary variables in HMM

Forward variable

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

Backward variable

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$

Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j \mid \xi_{1:T})$$

## Backward algorithm

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\boldsymbol{\xi}_{t+1:T} \mid s_t = i)$$

Similarly, we can define a **backward variable** starting from

$$\beta_{T,i}^{\text{HMM}} = 1$$

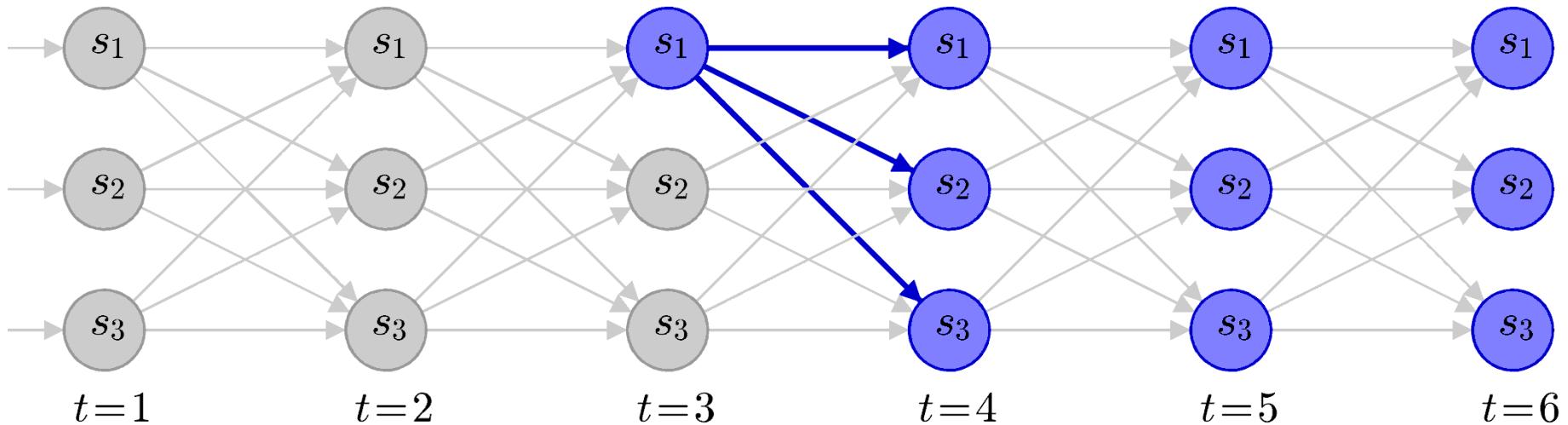
and computed as

$$\beta_{t,i}^{\text{HMM}} = \sum_{j=1}^K a_{i,j} \mathcal{N}(\boldsymbol{\xi}_{t+1} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \beta_{t+1,j}^{\text{HMM}}$$

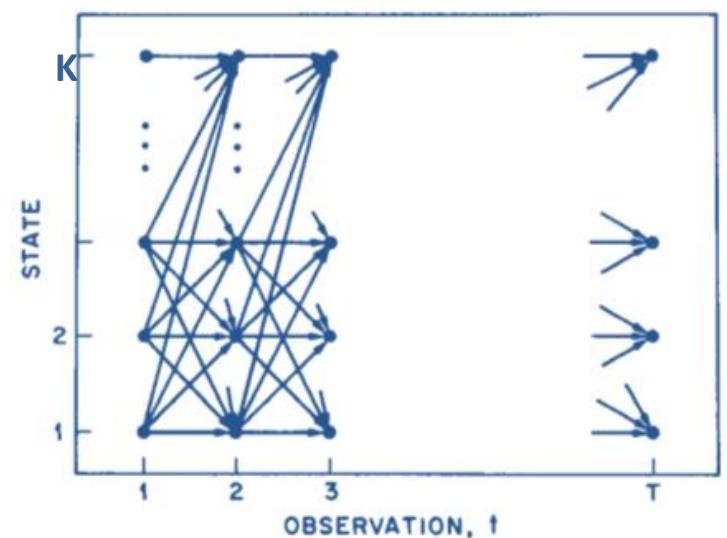
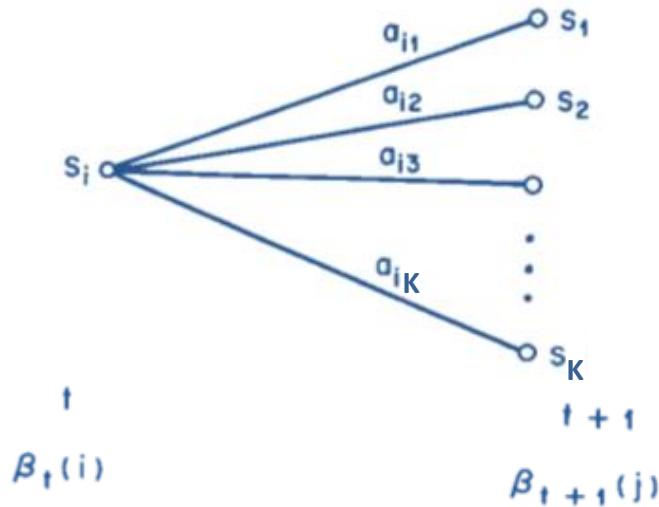
corresponding to the probability of the partial observation  $\{\boldsymbol{\xi}_{t+1}, \dots, \boldsymbol{\xi}_{T-1}, \boldsymbol{\xi}_T\}$ , knowing that we are in state  $i$  at time step  $t$ .

# Backward algorithm

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$



$$\beta_{t,i}^{\text{HMM}} = \sum_{j=1}^K a_{i,j} \mathcal{N}(\xi_{t+1} \mid \mu_j, \Sigma_j) \beta_{t+1,j}^{\text{HMM}} \quad \text{with} \quad \beta_{T,i}^{\text{HMM}} = 1$$



# Useful intermediary variables in HMM

Forward variable

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

Backward variable

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$

Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j \mid \xi_{1:T})$$

These variable are sometimes called "smoothed values" as they combine forward and backward probabilities in the computation.

You can think of their roles as passing "messages" from left to right, and from right to left, and then combining the information at each node.

# Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

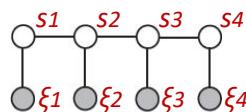
Given the full observation  $\xi = \{\xi_1, \xi_2, \dots, \xi_T\}$ , the probability of  $\xi_t$  to be in state  $i$  at time step  $t$  is

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T}) = \frac{\mathcal{P}(s_t=i, \xi_{1:T})}{\mathcal{P}(\xi_{1:T})}$$

$$\mathcal{P}(a, b) = \mathcal{P}(b|a)\mathcal{P}(a) \quad \mathcal{P}(b|a) = \frac{\mathcal{P}(a, b)}{\mathcal{P}(a)}$$

$$= \frac{\mathcal{P}(\xi_{1:T} \mid s_t=i) \mathcal{P}(s_t=i)}{\mathcal{P}(\xi_{1:T})}$$

Conditional independence property



$$= \frac{\mathcal{P}(\xi_{1:t} \mid s_t=i) \mathcal{P}(\xi_{t+1:T} \mid s_t=i) \mathcal{P}(s_t=i)}{\mathcal{P}(\xi_{1:T})}$$

$$\mathcal{P}(a, b) = \mathcal{P}(b|a)\mathcal{P}(a) \quad \mathcal{P}(b|a) = \frac{\mathcal{P}(s_t=i, \xi_{1:t}) \mathcal{P}(\xi_{t+1:T} \mid s_t=i)}{\mathcal{P}(\xi_{1:T})}$$

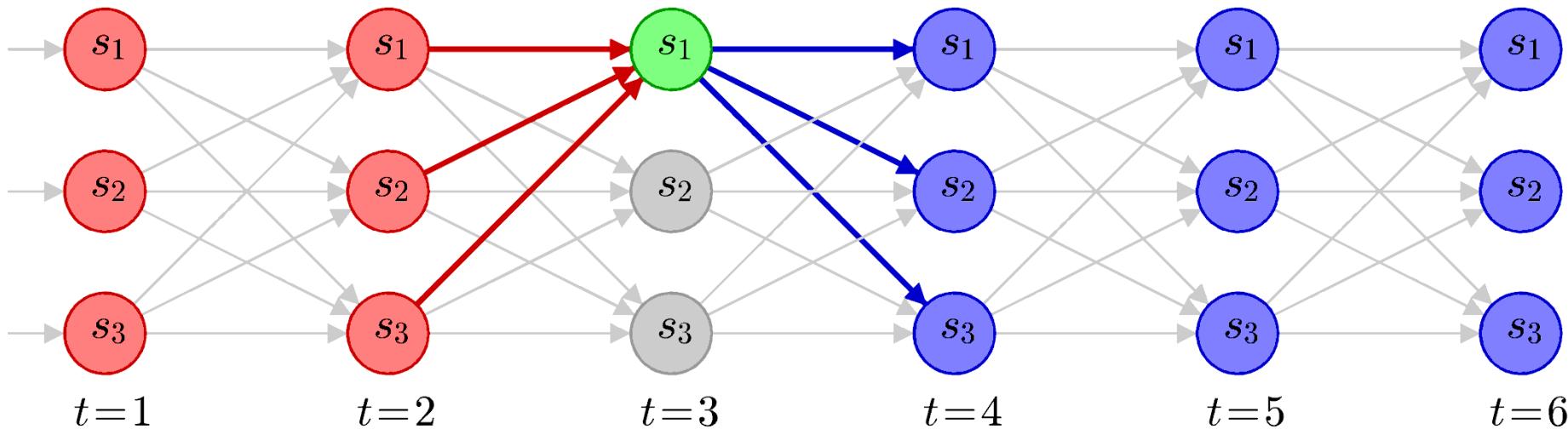
$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t}) = \frac{\alpha_{t,i}^{\text{HMM}} \beta_{t,i}^{\text{HMM}}}{\sum_{k=1}^K \alpha_{t,k}^{\text{HMM}} \beta_{t,k}^{\text{HMM}}} \quad \beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$

# Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$



$$\gamma_{t,i}^{\text{HMM}} = \frac{\alpha_{t,i}^{\text{HMM}} \beta_{t,i}^{\text{HMM}}}{\sum_{k=1}^K \alpha_{t,k}^{\text{HMM}} \beta_{t,k}^{\text{HMM}}} = \frac{\alpha_{t,i}^{\text{HMM}} \beta_{t,i}^{\text{HMM}}}{\mathcal{P}(\xi)}$$

# Useful intermediary variables in HMM

Forward variable

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$$

Backward variable

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t=i)$$

Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i \mid \xi_{1:T})$$

Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j \mid \xi_{1:T})$$

These variable are sometimes called "smoothed values" as they combine forward and backward probabilities in the computation.

You can think of their roles as passing "messages" from left to right, and from right to left, and then combining the information at each node.

## Smoothed edge marginals $\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j | \boldsymbol{\xi}_{1:T})$

Given the full observation  $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_T\}$ , the probability to be in state  $i$  at time step  $t$  and in state  $j$  at time step  $t+1$  is

$$\begin{aligned} \zeta_{t,i,j}^{\text{HMM}} &= \mathcal{P}(s_t=i, s_{t+1}=j | \boldsymbol{\xi}_{1:T}) \\ &\stackrel{\textcolor{red}{\curvearrowleft}}{=} \frac{\mathcal{P}(s_t=i, s_{t+1}=j, \boldsymbol{\xi}_{1:T})}{\mathcal{P}(\boldsymbol{\xi}_{1:T})} \\ \mathcal{P}(b|a) = \frac{\mathcal{P}(a,b)}{\mathcal{P}(a)} &= \frac{\alpha_{t,i}^{\text{HMM}} a_{i,j} \mathcal{N}(\boldsymbol{\xi}_{t+1} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \beta_{t+1,j}^{\text{HMM}}}{\sum_{k=1}^K \sum_{l=1}^K \alpha_{t,k}^{\text{HMM}} a_{k,l} \mathcal{N}(\boldsymbol{\xi}_{t+1} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \beta_{t+1,l}^{\text{HMM}}} \end{aligned}$$

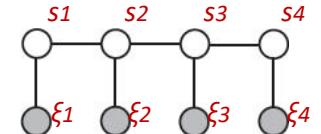
Note that we have  $\gamma_{t,i}^{\text{HMM}} = \sum_{j=1}^K \zeta_{t,i,j}^{\text{HMM}}$

# Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j | \boldsymbol{\xi}_{1:T})$$

This result can be retrieved by rewriting the numerator with Bayes rules and the conditional independence properties of the model

$$\mathcal{P}(s_t, s_{t+1}, \boldsymbol{\xi}_{1:T}) = \mathcal{P}(\boldsymbol{\xi}_{1:T} | s_t, s_{t+1}) \mathcal{P}(s_t, s_{t+1})$$



$$= \mathcal{P}(\boldsymbol{\xi}_{1:t} | s_t, s_{t+1}) \mathcal{P}(\boldsymbol{\xi}_{t+1} | s_t, s_{t+1}) \mathcal{P}(\boldsymbol{\xi}_{t+2:T} | s_t, s_{t+1}) \mathcal{P}(s_{t+1} | s_t) \mathcal{P}(s_t)$$

Conditional independence property

$$= \mathcal{P}(\boldsymbol{\xi}_{1:t} | s_t) \mathcal{P}(\boldsymbol{\xi}_{t+1} | s_{t+1}) \mathcal{P}(\boldsymbol{\xi}_{t+2:T} | s_{t+1}) \mathcal{P}(s_{t+1} | s_t) \mathcal{P}(s_t)$$

$$= \underbrace{\mathcal{P}(s_t, \boldsymbol{\xi}_{1:t})}_{\alpha_{t,i}^{\text{HMM}}} \underbrace{\mathcal{P}(\boldsymbol{\xi}_{t+1} | s_{t+1})}_{\mathcal{N}(\boldsymbol{\xi}_{t+1} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \underbrace{\mathcal{P}(\boldsymbol{\xi}_{t+2:T} | s_{t+1})}_{\beta_{t+1,j}^{\text{HMM}}} \underbrace{\mathcal{P}(s_{t+1} | s_t)}_{a_{i,j}}$$

$\mathcal{P}(\boldsymbol{\xi}_{t+1} | s_{t+1}=j) = \mathcal{N}(\boldsymbol{\xi}_{t+1} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

$\mathcal{P}(s_{t+1}=j | s_t=i) = a_{i,j}$

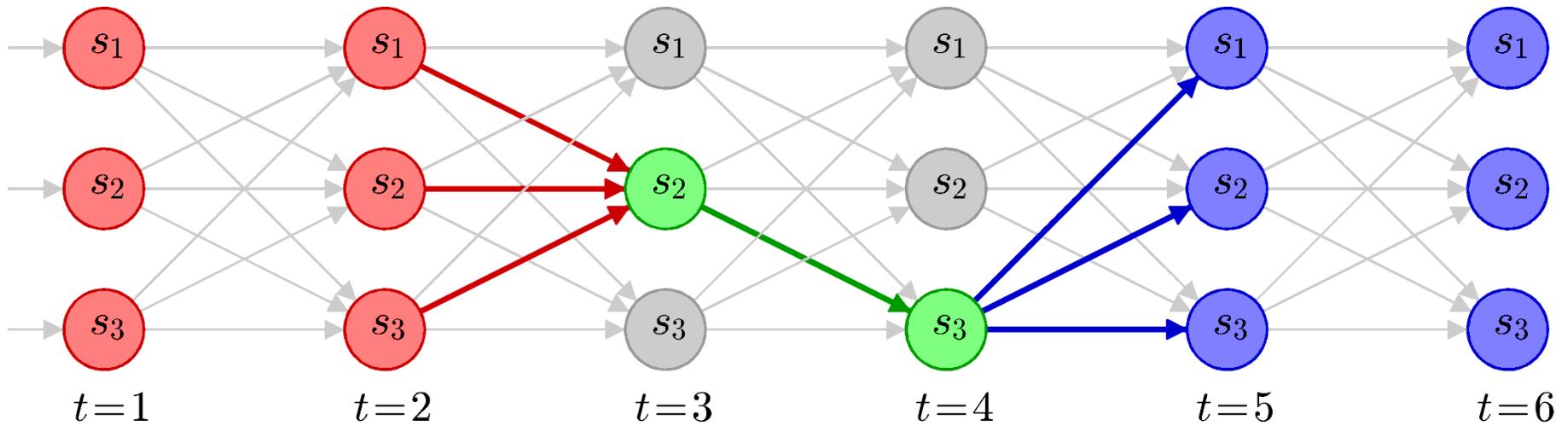
$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \boldsymbol{\xi}_{1:t})$$

$$= \mathcal{P}(\boldsymbol{\xi}_{1:t} | s_t=i) \mathcal{P}(s_t=i)$$

$$\beta_{t+1,j}^{\text{HMM}} = \mathcal{P}(\boldsymbol{\xi}_{t+2:T} | s_{t+1}=j)$$

# Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j | \xi_{1:T})$$



$$\begin{aligned} \zeta_{t,i,j}^{\text{HMM}} &= \frac{\alpha_{t,i}^{\text{HMM}} a_{i,j} \mathcal{N}(\xi_{t+1} | \mu_j, \Sigma_j) \beta_{t+1,j}^{\text{HMM}}}{\sum_{k=1}^K \sum_{l=1}^K \alpha_{t,k}^{\text{HMM}} a_{k,l} \mathcal{N}(\xi_{t+1} | \mu_l, \Sigma_l) \beta_{t+1,l}^{\text{HMM}}} \\ &= \frac{\alpha_{t,i}^{\text{HMM}} a_{i,j} \mathcal{N}(\xi_{t+1} | \mu_j, \Sigma_j) \beta_{t+1,j}^{\text{HMM}}}{\mathcal{P}(\xi)} \end{aligned}$$

# EM for HMM

$K$  Gaussians  
 $M$  trajectories  
 $T_m$  points per traj.

The expected complete data log-likelihood is

$$Q(\Theta, \Theta^{\text{old}}) = \mathbb{E} \left[ \sum_{m=1}^M \sum_{t=1}^{T_m} \log \mathcal{P}(\xi_{m,t}, s_t | \Theta) \mid \xi, \Theta^{\text{old}} \right]$$

$$= \sum_{i=1}^K \mathbb{E}[N_i] \log \Pi_i + \sum_{i=1}^K \sum_{j=1}^K \mathbb{E}[N_{i,j}] \log a_{i,j}$$

Similar to  
Markov models

Similar to GMM

$$+ \sum_{m=1}^M \sum_{t=1}^{T_m} \sum_{i=1}^K \mathcal{P}(s_t=i \mid \xi_m, \Theta^{\text{old}}) \log \mathcal{N}(\xi_{m,t} | \mu_i, \Sigma_i)$$

with expected counts given by

$$\mathbb{E}[N_i] = \sum_{m=1}^M \mathcal{P}(s_{m,1}=i \mid \xi_m, \Theta^{\text{old}}) = \sum_{m=1}^M \gamma_{m,1,i}^{\text{HMM}}$$

$$\mathbb{E}[N_{i,j}] = \sum_{m=1}^M \sum_{t=1}^{T_m-1} \mathcal{P}(s_{m,t}=i, s_{m,t+1}=j \mid \xi_m, \Theta^{\text{old}}) = \sum_{m=1}^M \sum_{t=1}^{T_m-1} \zeta_{m,t,i,j}^{\text{HMM}}$$

# EM for HMM

$K$  Gaussians  
 $M$  trajectories  
 $T_m$  points per traj.

By setting

$$\frac{\partial \mathcal{Q}(\Theta, \Theta^{\text{old}})}{\partial \Pi_i} = 0 \quad \frac{\partial \mathcal{Q}(\Theta, \Theta^{\text{old}})}{\partial a_{i,j}} = 0$$

a result similar to the case of Markov models is obtained.

The maximum likelihood estimate of the initial state distribution and transition probability parameters can thus be computed with the normalized counts

$$\hat{\Pi}_i = \frac{\mathbb{E}[N_i]}{\sum_{k=1}^K \mathbb{E}[N_k]} = \frac{\mathbb{E}[N_i]}{M}, \quad \hat{a}_{i,j} = \frac{\mathbb{E}[N_{i,j}]}{\sum_{k=1}^K \mathbb{E}[N_{i,k}]} = \frac{\mathbb{E}[N_{i,j}]}{\mathbb{E}[N_i]}$$

with

$$\mathbb{E}[N_i] = \sum_{m=1}^M \sum_{t=1}^{T_m} \mathcal{P}(s_{m,t}=i \mid \boldsymbol{\xi}_m, \Theta^{\text{old}}) = \sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}}$$

# EM for HMM - Summary

$K$  Gaussians  
 $M$  trajectories  
 $T_m$  points per traj.

M-step:

$$\Pi_i \leftarrow \frac{\sum_{m=1}^M \gamma_{m,1,i}^{\text{HMM}}}{M} = \frac{\text{Total number of times in } i \text{ at time step 1}}{\text{Total number of trajectories}}$$

$$a_{i,j} \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m-1} \zeta_{m,t,i,j}^{\text{HMM}}}{\sum_{m=1}^M \sum_{t=1}^{T_m-1} \gamma_{m,t,i}^{\text{HMM}}} = \frac{\text{Total number of transitions from } i \text{ to } j}{\text{Total number of times in } i \text{ (and transit to anything else)}}$$

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}} \boldsymbol{\xi}_{m,t}}{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}}} \quad \curvearrowright \quad \text{By setting } \frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\mu}_i} = 0 \text{ and } \frac{\partial \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\text{old}})}{\partial \boldsymbol{\Sigma}_i} = 0, \text{ a result similar to GMM is obtained.}$$

$$\boldsymbol{\Sigma}_i \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}} (\boldsymbol{\xi}_{m,t} - \boldsymbol{\mu}_i)(\boldsymbol{\xi}_{m,t} - \boldsymbol{\mu}_i)^{\top}}{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}}} \quad \curvearrowright$$

# EM for HMM - Summary

$K$  Gaussians  
 $M$  trajectories  
 $T_m$  points per traj.

M-step:

$$\Pi_i \leftarrow \frac{\sum_{m=1}^M \gamma_{m,1,i}^{\text{HMM}}}{M}$$

$$a_{i,j} \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m-1} \zeta_{m,t,i,j}^{\text{HMM}}}{\sum_{m=1}^M \sum_{t=1}^{T_m-1} \gamma_{m,t,i}^{\text{HMM}}}$$

$$\mu_i \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}} \xi_{m,t}}{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}}}$$

$$\Sigma_i \leftarrow \frac{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}} (\xi_{m,t} - \mu_i)(\xi_{m,t} - \mu_i)^\top}{\sum_{m=1}^M \sum_{t=1}^{T_m} \gamma_{m,t,i}^{\text{HMM}}}$$

These results can be formally retrieved with EM (also called **Baum-Welch algorithm** in the context of HMM).

We provide here an interpretation in terms of normalized counts, with several types of weighted averages required in the computation.

## Numerical underflow issue in HMM

For long sequences, the forward and backward variables can quickly get very low, likely exceeding the precision range of the computer.

A simple scaling procedure is to multiply  $\alpha_{t,i}^{\text{HMM}}$  by a factor independent of  $i$ , and divide  $\beta_{t,i}^{\text{HMM}}$  by the same factor so that they are cancelled in the forward-backward computation.

The computation can be kept within reasonable bounds by setting the scaling factor

$$c_t = \frac{1}{\sum_{i=1}^K \alpha_{t,i}^{\text{HMM}}}$$

# Numerical underflow issue in HMM

$$c_t = \frac{1}{\sum_{i=1}^K \alpha_{t,i}^{\text{HMM}}}$$

This issue is sometimes not covered in textbooks, although it remains very important for practical implementation of HMM!

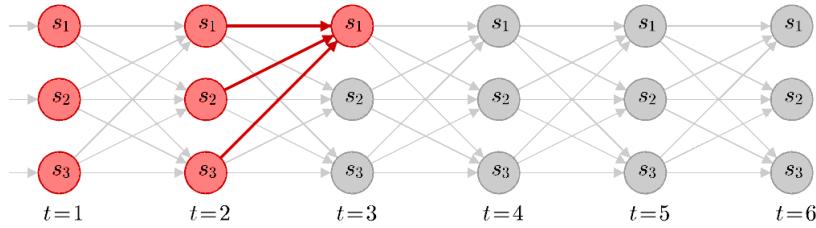
We have by induction

$$\hat{\alpha}_{t,i}^{\text{HMM}} = \left( \prod_{s=1}^t c_s \right) \alpha_{t,i}^{\text{HMM}}, \quad \hat{\beta}_{t,i}^{\text{HMM}} = \left( \prod_{s=t}^T c_s \right) \beta_{t,i}^{\text{HMM}}$$

With this, the numerator and denominator will cancel out when used in the re-estimation formulas. For example

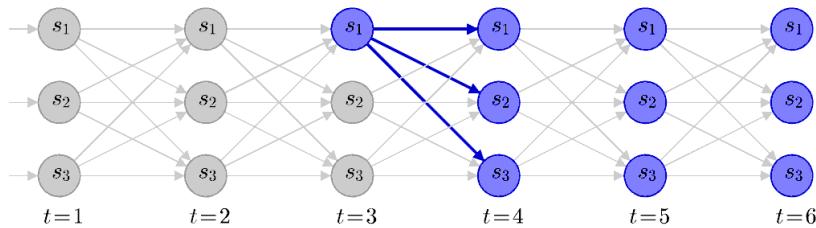
$$\gamma_{t,i}^{\text{HMM}} = \frac{\hat{\alpha}_{t,i}^{\text{HMM}} \hat{\beta}_{t,i}^{\text{HMM}}}{\sum_{k=1}^K \hat{\alpha}_{t,k}^{\text{HMM}} \hat{\beta}_{t,k}^{\text{HMM}}} = \frac{\left( \prod_{s=1}^t c_s \right) \left( \prod_{s=t}^T c_s \right) \alpha_{t,i}^{\text{HMM}} \beta_{t,i}^{\text{HMM}}}{\left( \prod_{s=1}^t c_s \right) \left( \prod_{s=t}^T c_s \right) \sum_{k=1}^K \alpha_{t,k}^{\text{HMM}} \beta_{t,k}^{\text{HMM}}} = \frac{\alpha_{t,i}^{\text{HMM}} \beta_{t,i}^{\text{HMM}}}{\sum_{k=1}^K \alpha_{t,k}^{\text{HMM}} \beta_{t,k}^{\text{HMM}}}$$

# Summary - Why did we introduce these four intermediary variables in HMM?



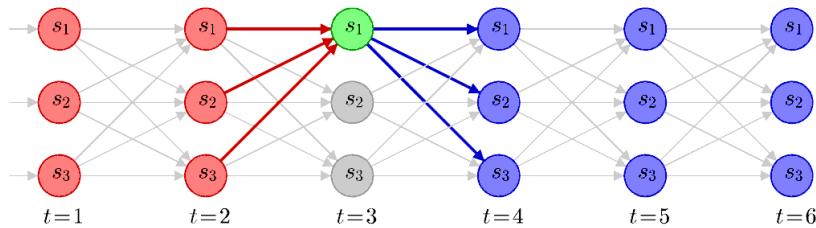
Forward variable

$$\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t = i, \xi_{1:t})$$



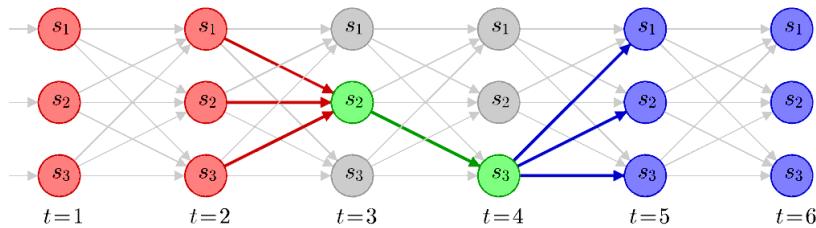
Backward variable

$$\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} \mid s_t = i)$$



Smoothed node marginals

$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t = i \mid \xi_{1:T})$$



Smoothed edge marginals

$$\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t = i, s_{t+1} = j \mid \xi_{1:T})$$

# Summary - Why did we introduce these four intermediary variables in HMM?

How to estimate the parameters of an HMM?

→ Maximum of expected complete data log-likelihood  $\mathcal{Q}(\Theta, \Theta^{\text{old}})$

How to compute  $\frac{\partial \mathcal{Q}}{\partial \Pi_i} = 0$ ,  $\frac{\partial \mathcal{Q}}{\partial a_{i,j}} = 0$ ,  $\frac{\partial \mathcal{Q}}{\partial \mu_i} = 0$  and  $\frac{\partial \mathcal{Q}}{\partial \Sigma_i} = 0$  ?

→ Requires to compute  $\zeta_{t,i,j}^{\text{HMM}} = \mathcal{P}(s_t=i, s_{t+1}=j | \xi_{1:T})$

→ Requires to compute  $\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i | \xi_{1:T})$

$$\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{\text{old}})$$

How to compute  $\zeta_{t,i,j}^{\text{HMM}}$  and  $\gamma_{t,i}^{\text{HMM}}$  ?

→ Requires to compute  $\alpha_{t,i}^{\text{HMM}} = \mathcal{P}(s_t=i, \xi_{1:t})$

$$\alpha_{t,i}^{\text{HMM}}$$

→ Requires to compute  $\beta_{t,i}^{\text{HMM}} = \mathcal{P}(\xi_{t+1:T} | s_t=i)$

$$\beta_{t,i}^{\text{HMM}}$$

# Viterbi decoding (MAP vs MPE estimates)

Maximum a posteriori

Most probable explanation

**Matlab code: demo\_HMM\_Viterbi01.m**

[L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77:2:257–285, February 1989]

# Viterbi decoding (MAP vs MPE estimates)

Maximum a posteriori

Most probable explanation

The (jointly) most probable sequence of states  $\hat{s}^{\text{MAP}}$  is not necessarily the same as the sequence of (marginally) most probable states  $\hat{s}^{\text{MPE}}$

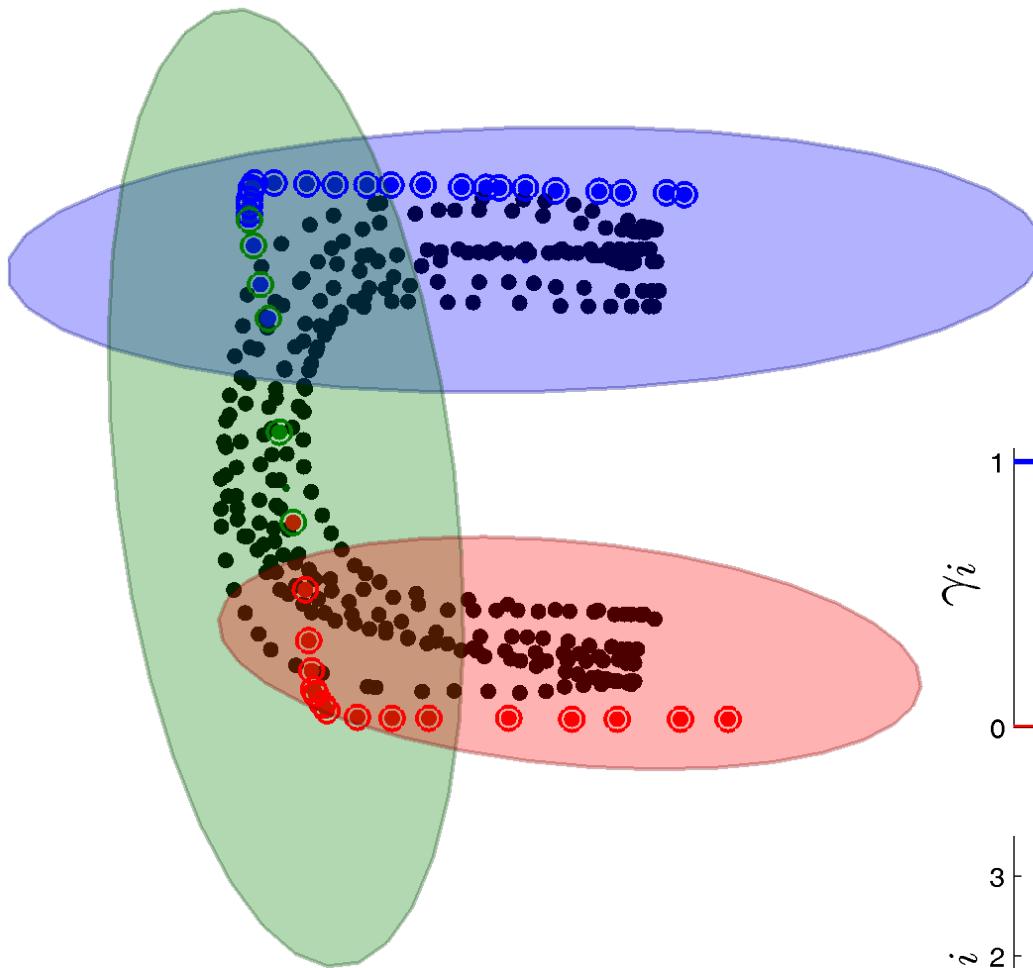
$$\hat{s}^{\text{MAP}} = \arg \max_{\{s_1, s_2, \dots, s_T\}} \mathcal{P}(s | \xi)$$

$$\hat{s}^{\text{MPE}} = \left\{ \arg \max_{s_1} \overbrace{\mathcal{P}(s_1 | \xi)}^{\gamma_1^{\text{HMM}}}, \arg \max_{s_2} \overbrace{\mathcal{P}(s_2 | \xi)}^{\gamma_2^{\text{HMM}}}, \dots, \arg \max_{s_T} \overbrace{\mathcal{P}(s_T | \xi)}^{\gamma_T^{\text{HMM}}} \right\}$$

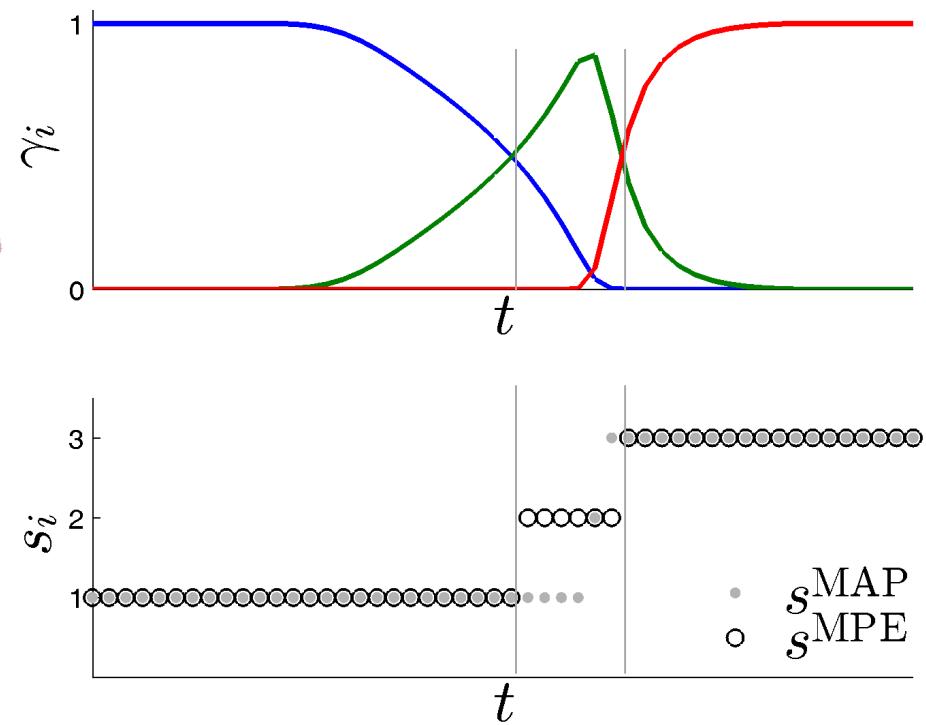
If there is a unique mode, they can provide the same result, but not in the general case with multiple equally probable assignments.

While  $\hat{s}^{\text{MPE}}$  can be computed by replacing the sum operator with a max operator in  $\gamma^{\text{HMM}}$ , the **Viterbi algorithm** used to compute  $\hat{s}^{\text{MAP}}$  employs the max operator in a forward pass, followed by a backward pass using a **fast traceback procedure** to recover the most probable path.

# Viterbi decoding (MAP vs MPE estimates)



$$\gamma_{t,i}^{\text{HMM}} = \mathcal{P}(s_t = i \mid \xi_{1:T})$$



# Viterbi decoding - Algorithm

$$\alpha_{t,i}^{\text{HMM}} = \left( \sum_{j=1}^K \alpha_{t-1,j}^{\text{HMM}} a_{j,i} \right) \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

with  $\alpha_{1,i}^{\text{HMM}} = \Pi_i \mathcal{N}(\boldsymbol{\xi}_1 | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

Initialization:

$$\delta_{1,i} = \Pi_i \mathcal{N}(\boldsymbol{\xi}_1 | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\Psi_{1,i} = 0$$

This is the probability of ending up in state  $i$  at time step  $t$  by taking the most probable path

Recursion:

$$\delta_{t,i} = \max_j (\delta_{t-1,j} a_{j,i}) \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\Psi_{t,i} = \arg \max_j (\delta_{t-1,j} a_{j,i}) \quad \forall t \in \{2, 3, \dots, T\}$$

Termination:

$$\hat{s}_T^{\text{MAP}} = \arg \max_j \delta_{T,j}$$

It tells us the most likely previous state on the most probable path to  $s_t = i$

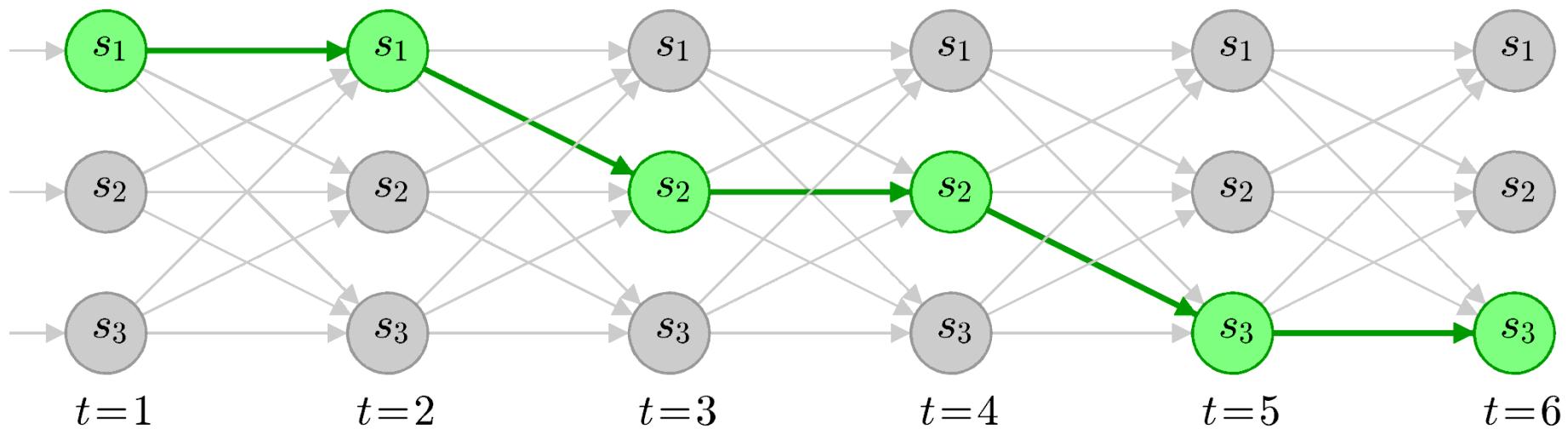
Backtracking:

$$\hat{s}_t^{\text{MAP}} = \Psi_{t+1, \hat{s}_{t+1}^{\text{MAP}}}$$

$$\forall t \in \{T-1, T-2, \dots, 1\}$$

Here,  $\delta_{t,i} = \max_{s_{1:t-1}} \mathcal{P}(s_{1:t-1}, s_t = i | \boldsymbol{\xi}_{1:t})$ , and  $\Psi_{t,i}$  are state indices that keep track of the states  $j$  that maximized  $\delta_{t,i}$ .

# Viterbi decoding - Trellis representation



$$\delta_{t,i} = \max_j (\delta_{t-1,j} a_{j,i}) \mathcal{N}(\xi_t | \mu_i, \Sigma_i)$$

$$\Psi_{t,i} = \arg \max_j (\delta_{t-1,j} a_{j,i})$$

$$\hat{s}_t^{\text{MAP}} = \Psi_{t+1, \hat{s}_{t+1}^{\text{MAP}}}$$

## Numerical underflow issue in Viterbi

Similarly to the forward-backward variables in HMM, we have to take care about potential numerical underflow when implementing Viterbi decoding.

A simple way is to normalize  $\delta_{t,i}$  at each time step  $t$  with

$$c_t = \frac{1}{\sum_{i=1}^K \delta_{t,i}}$$

similarly as in the computation of the forward-backward variables. Such scaling will not affect the maximum.

## Numerical underflow issue in Viterbi

Alternatively, we can work in the log domain. We then have

$$\begin{aligned}\log \delta_{t,i} &= \max_{\boldsymbol{s}_{1:t-1}} \log \mathcal{P}(\boldsymbol{s}_{1:t-1}, s_t = i \mid \boldsymbol{\xi}_{1:t}) \\ &= \max_j (\log \delta_{t-1,j} + \log a_{i,j}) + \log \mathcal{N}(\boldsymbol{\xi}_t \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\end{aligned}$$

With high dimensional Gaussians as emission distributions, the Viterbi computation with log can result in a **significant speedup**, since computing  $\log \mathcal{P}(\boldsymbol{\xi}_t | s_t)$  can be much faster than computing  $\mathcal{P}(\boldsymbol{\xi}_t | s_t)$ .

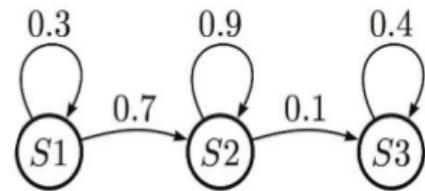
It is for this reason that it is also common to use the Viterbi algorithm in the E step of the EM procedure instead of the forward-backward variables when training HMMs with data of high dimension.

# Viterbi decoding - Example

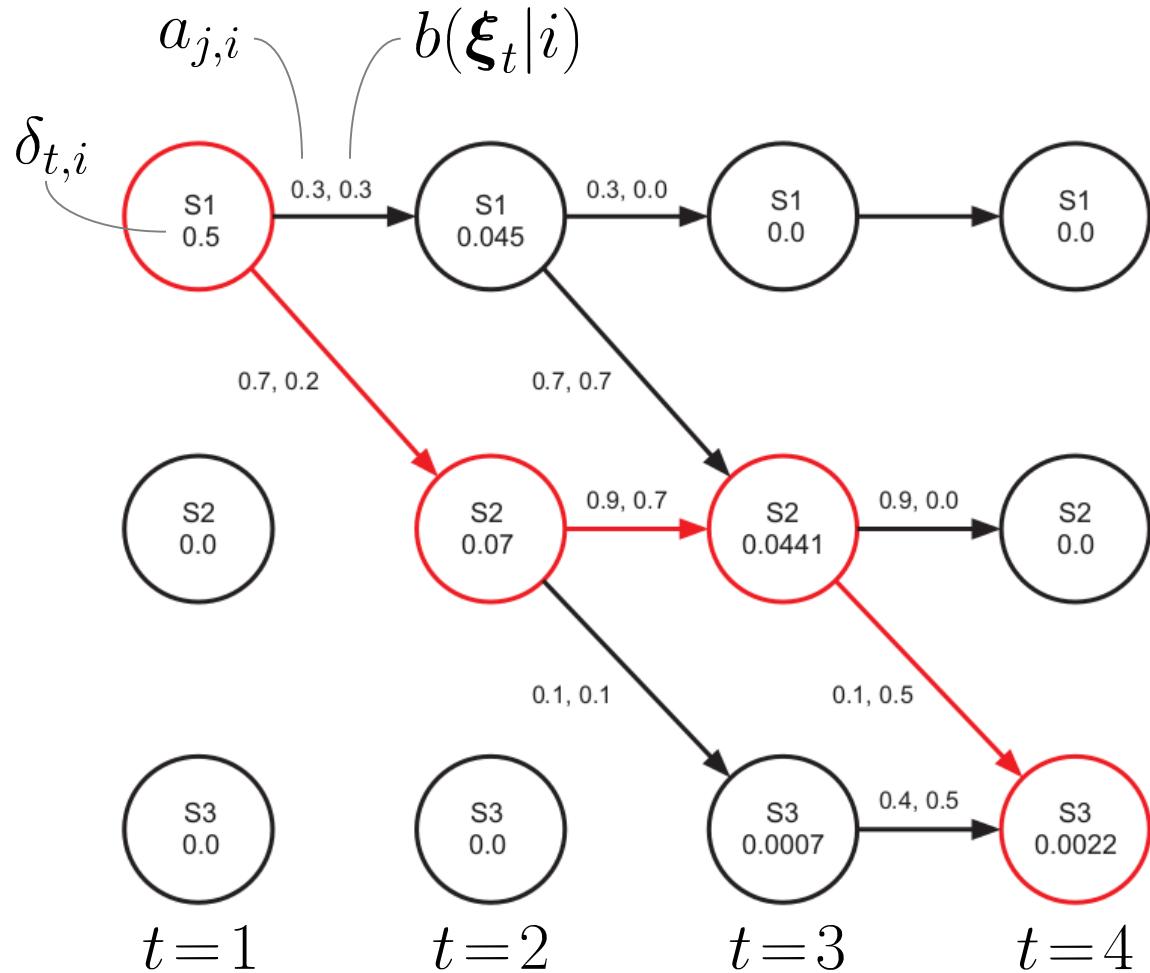
$$\delta_{t,i} = \max_j (\delta_{t-1,j} a_{j,i}) b(\xi_t | i)$$

$$\text{with } \delta_{1,i} = \Pi_i \mathcal{N}(\xi_1 | \mu_i, \Sigma_i)$$

$$\Pi = [1, 0, 0]$$



C1	0.5	0
C2	0.2	0
C3	0.3	0.2
C4	0	0.7
C5	0	0.1
C6	0	0
C7	0	0.4



$$\xi = \{C1, C3, C4, C6\}$$

# **Hidden semi-Markov model (HSMM)**

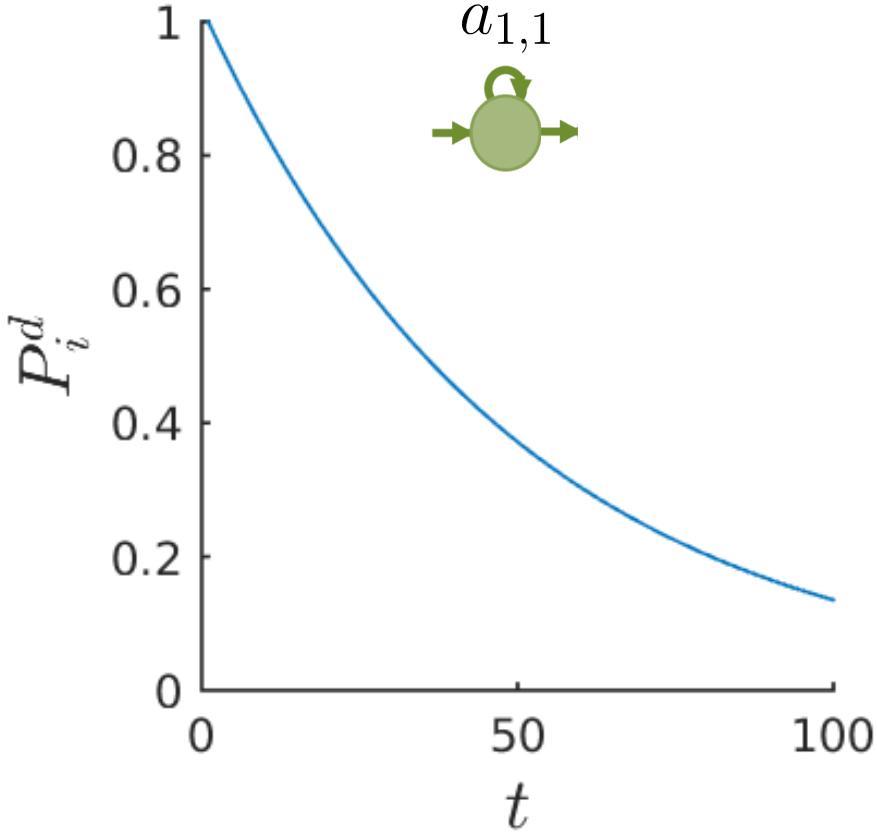
**Matlab code: demo\_HSMM01.m**

[S.-Z. Yu. Hidden semi-Markov models. Artificial Intelligence,  
174:215–243, 2010]

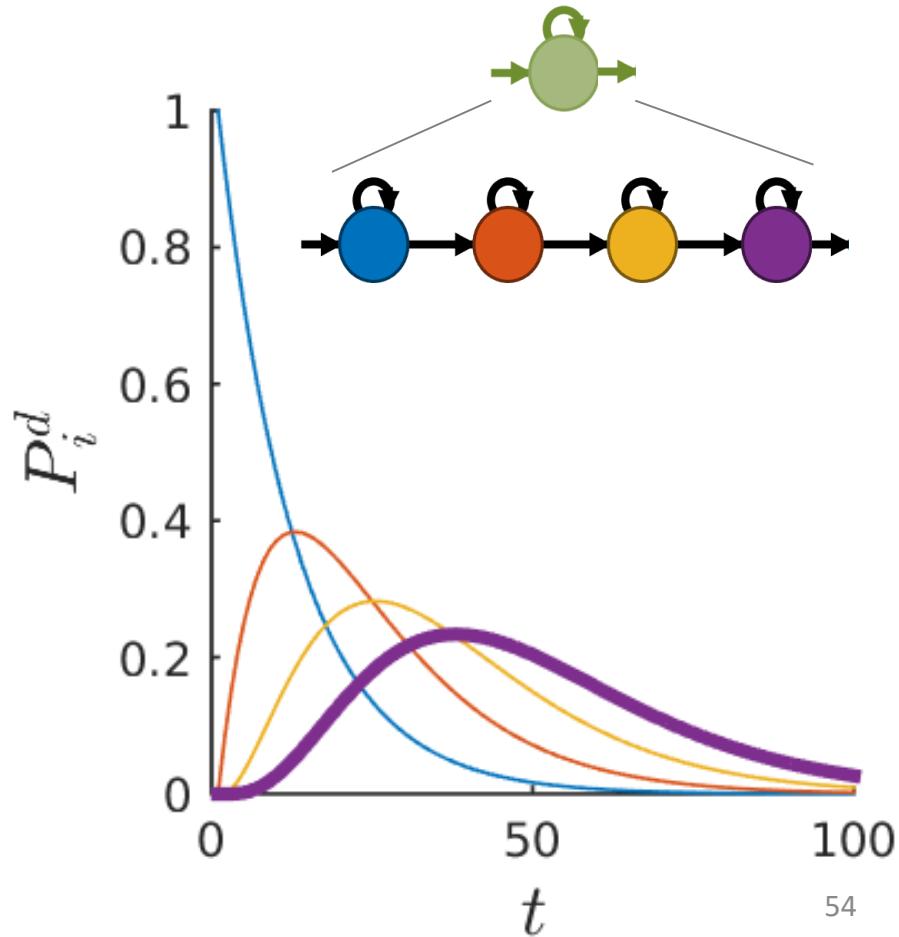
# State duration probability in standard HMM

The state duration follows a geometric distribution

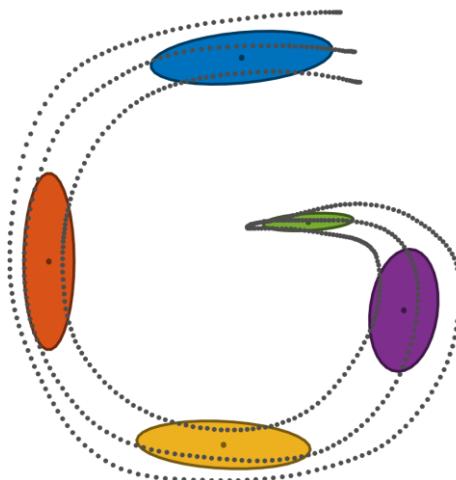
$$\mathcal{P}(d) = a_{i,i}^{d-1}(1 - a_{i,i})$$



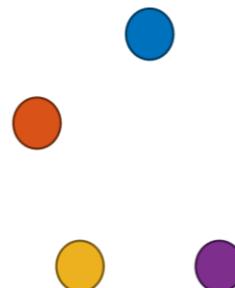
By artificially duplicating the number of states while keeping the same emission distribution, other state duration distributions can be modeled



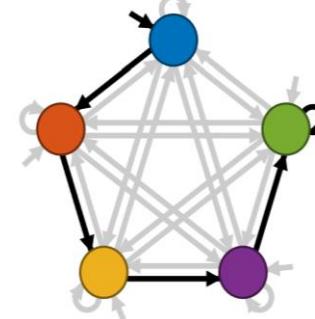
# Hidden semi-Markov model (HSMM)



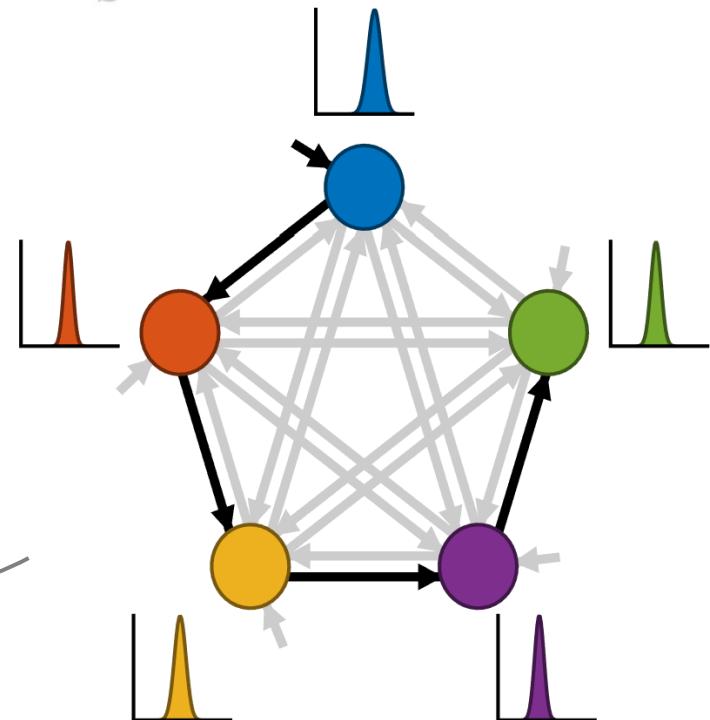
GMM



HMM



HSMM



Another approach is to provide an explicit model of the state duration instead of relying on self-transition probabilities

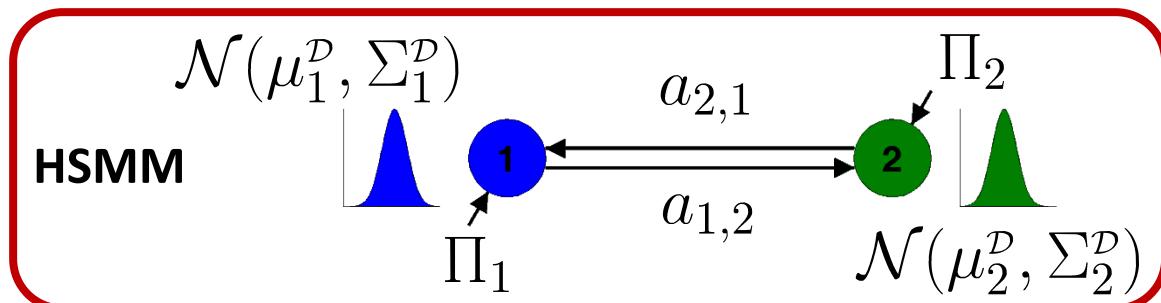
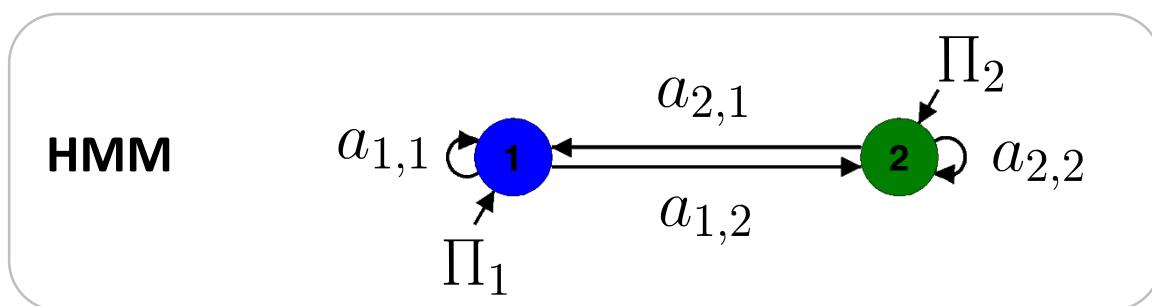
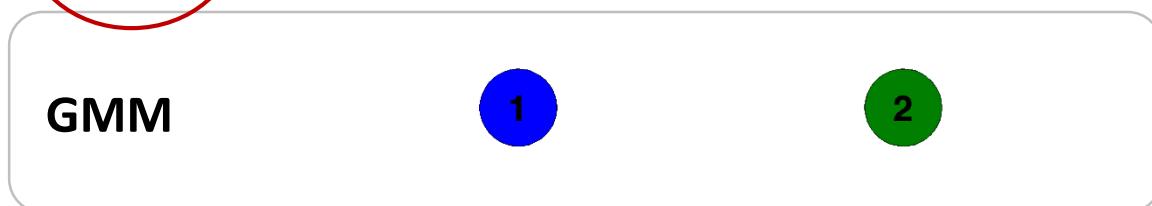
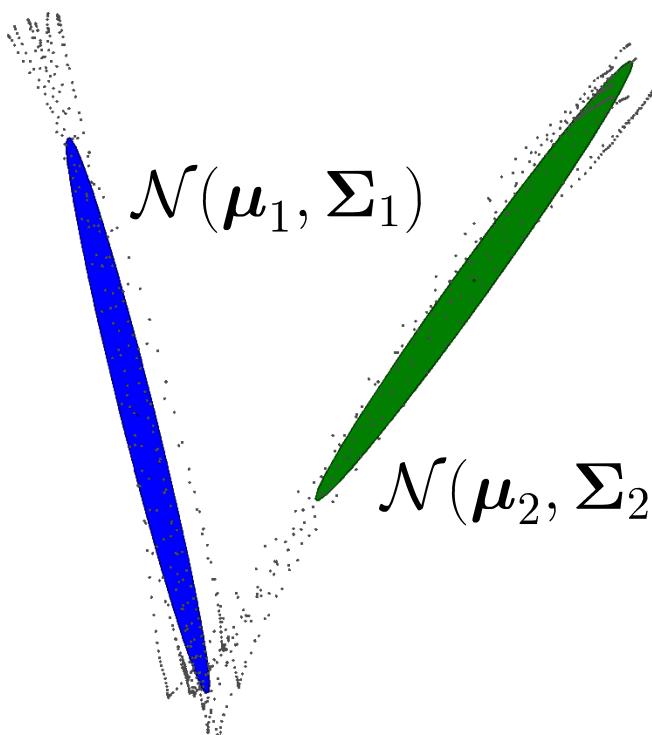
# Hidden semi-Markov model (HSMM)

$$\Theta^{\text{GMM}} = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K$$

$$\Theta^{\text{HMM}} = \{\{a_{i,j}\}_{j=1}^K, \Pi_i, \mu_i, \Sigma_i\}_{i=1}^K$$

$$\Theta^{\text{HSMM}} = \{\{a_{i,j}\}_{j=1, j \neq i}^K, \Pi_i, \mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}}, \mu_i, \Sigma_i\}_{i=1}^K$$

Parametric duration distribution



# Hidden semi-Markov model (HSMM)

While the HMM computes the *forward* variable as

$$\alpha_{t,i}^{\text{HMM}} = \sum_{j=1}^K \alpha_{t-1,j}^{\text{HMM}} a_{j,i} \mathcal{N}_{t,i}, \quad \text{with } \mathcal{N}_{t,i} = \mathcal{N}(\boldsymbol{\xi}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

the HSMM requires the evaluation of

$$\alpha_{t,i}^{\text{HSMM}} = \sum_{d=1}^{d^{\max}} \sum_{j=1}^K \alpha_{t-1,j}^{\text{HSMM}} a_{(j,d),(i,d)} \mathcal{N}_{(t,d),i}$$

where the system has to keep an history of length  $d^{\max}$ .

$a_{(j,d),(i,d)}$  is the probability to be in state  $i$  at iterations  $[t+1, t+d]$  knowing that we were in state  $j$  at iterations  $[t-d+1, t]$ .

$\mathcal{N}_{(t,d),i}$  is the probability to observe  $\{\boldsymbol{\xi}_{t-d+1}, \boldsymbol{\xi}_{t-d+2}, \dots, \boldsymbol{\xi}_t\}$  knowing that we were in state  $i$  at iterations  $[t-d+1, t]$ .

# Hidden semi-Markov model (HSMM)

$$\alpha_{t,i}^{\text{HSMM}} = \sum_{d=1}^{d^{\max}} \sum_{j=1}^K \alpha_{t-1,j}^{\text{HSMM}} a_{(j,d),(i,d)} \mathcal{N}_{(t,d),i}$$

An **explicit-duration HSMM** with, for example\*, a Gaussian parametrization of the duration  $\mathcal{N}_{d,i}^{\mathcal{D}} = \mathcal{N}(d | \mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}})$  assumes that

$$a_{(j,d),(i,d)} = a_{j,i} \mathcal{N}_{d,i}^{\mathcal{D}} \quad \text{and} \quad \mathcal{N}_{(t,d),i} = \prod_{s=t-d+1}^t \mathcal{N}_{s,i}$$

which corresponds to the assumption that the state duration is dependent on the current state and independent on the previous state, and that the outputs are conditionally independent.

\* used here only for simplification: other distributions from the exponential family are better suited to model positive counts (e.g., **gamma** or **log-normal** distributions).

## Hidden semi-Markov model (HSMM)

The probability to be in state  $i$  at time step  $t$  given the partial observation  $\boldsymbol{\xi}_{1:t} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_t\}$  can then be recursively computed

$$\mathcal{P}(s_t = i \mid \boldsymbol{\xi}_{1:t}) = \frac{\alpha_{t,i}^{\text{HSMM}}}{\sum_{k=1}^K \alpha_{t,k}^{\text{HSMM}}}, \quad \text{with}$$

$$\alpha_{t,i}^{\text{HSMM}} = \sum_{d=1}^{d^{\max}} \sum_{j=1}^K \alpha_{t-d,j}^{\text{HSMM}} a_{j,i} \mathcal{N}_{d,i}^{\mathcal{D}} \prod_{s=t-d+1}^t \mathcal{N}_{s,i}, \quad \text{where}$$

$$\mathcal{N}_{d,i}^{\mathcal{D}} = \mathcal{N}(d \mid \mu_i^{\mathcal{D}}, \Sigma_i^{\mathcal{D}}) \quad \text{and} \quad \mathcal{N}_{s,i} = \mathcal{N}(\boldsymbol{\xi}_s \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

# Hidden semi-Markov model (HSMM)

For  $t < d^{\max}$ , the initialization is given by

$$\alpha_{1,i}^{\text{HSMM}} = \Pi_i \mathcal{N}_{1,i}^{\mathcal{D}} \mathcal{N}_{1,i}$$

$$\alpha_{2,i}^{\text{HSMM}} = \Pi_i \mathcal{N}_{2,i}^{\mathcal{D}} \prod_{s=1}^2 \mathcal{N}_{s,i} + \sum_{j=1}^K \alpha_{1,j}^{\text{HSMM}} a_{j,i} \mathcal{N}_{1,i}^{\mathcal{D}} \mathcal{N}_{2,i}$$

$$\alpha_{3,i}^{\text{HSMM}} = \Pi_i \mathcal{N}_{3,i}^{\mathcal{D}} \prod_{s=1}^3 \mathcal{N}_{s,i} + \sum_{j=1}^K \sum_{d=1}^2 \alpha_{3-d,j}^{\text{HSMM}} a_{j,i} \mathcal{N}_{d,i}^{\mathcal{D}} \prod_{s=4-d}^3 \mathcal{N}_{s,i} \quad \text{etc.}$$

which corresponds to an update rule for  $t < d^{\max}$  written as

$$\alpha_{t,i}^{\text{HSMM}} = \Pi_i \mathcal{N}_{t,i}^{\mathcal{D}} \prod_{s=1}^t \mathcal{N}_{s,i} + \sum_{j=1}^K \sum_{d=1}^{t-1} \alpha_{t-d,j}^{\text{HSMM}} a_{j,i} \mathcal{N}_{d,i}^{\mathcal{D}} \prod_{s=t-d+1}^t \mathcal{N}_{s,i}$$

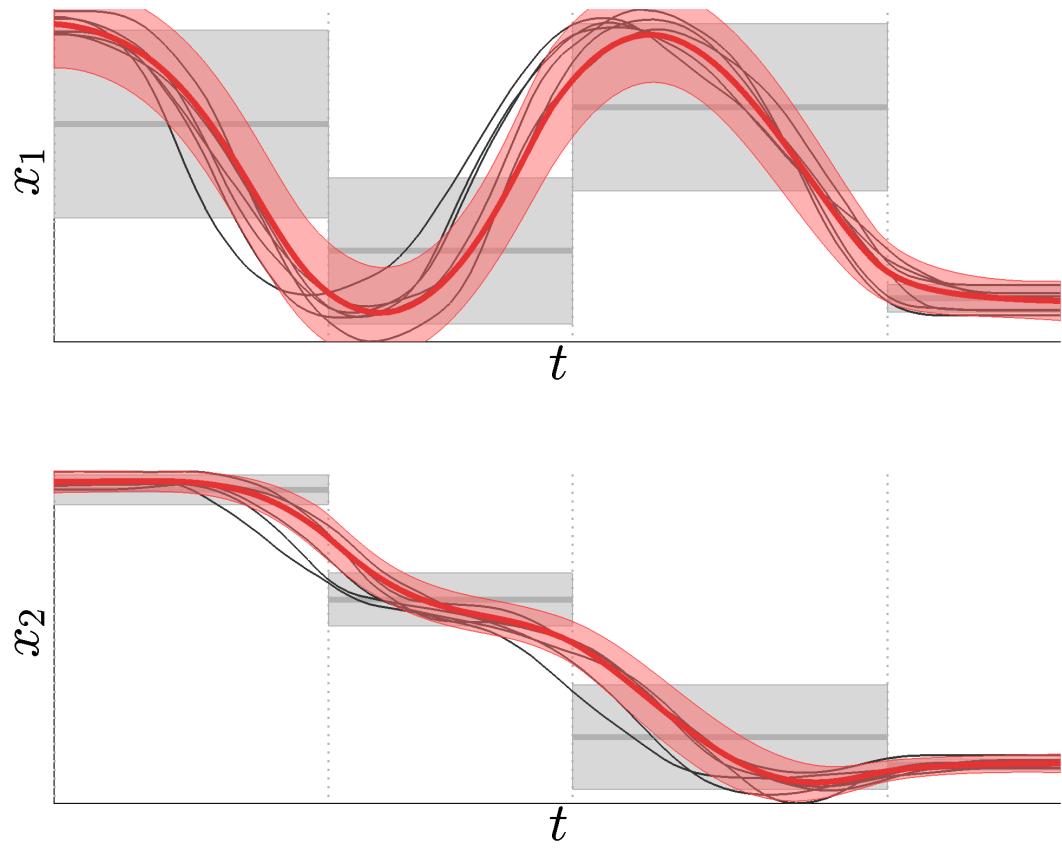
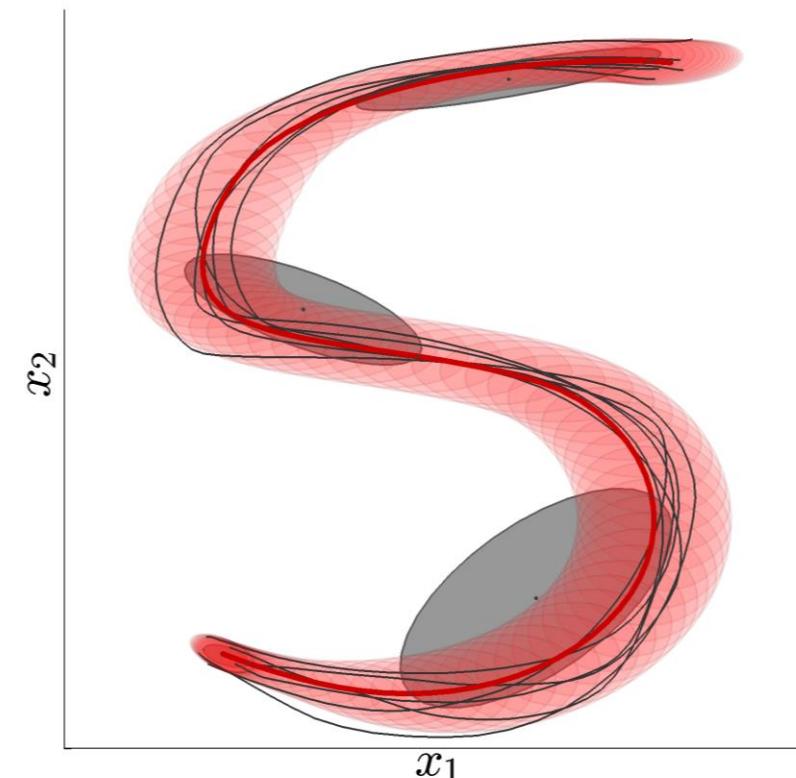
# **GMM/HMM/HSMM with dynamic features**

## **(Trajectory-HMM)**

**Matlab code: demo\_trajHSMM01.m**

[H. Zen, K. Tokuda, and T. Kitamura. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. Computer Speech and Language, 21(1):153–173, 2007]

# GMM/HMM/HSMM with dynamic features



# **GMM/HMM/HSMM with dynamic features**

In the field of speech processing, the extraction of statistics from both static and dynamic features within an HMM is very common.

In particular, it can be used in speech synthesis to avoid discontinuities in the generated speech spectra.

The synthesized speech then becomes natural and smooth even when a small number of Gaussians is used.

This is achieved by coordinating the distributions of both static and dynamic features (the dynamic features are often called delta and delta-delta parameters).

In speech processing, these parameters usually correspond to the evolution of mel-frequency cepstral coefficients characterizing the power spectrum of a sound, but the same approach can be used with any form of continuous signals.

## **GMM/HMM/HSMM with dynamic features**

For the encoding of movements, velocity and acceleration can alternatively be used as dynamic features. By considering an Euler approximation, the velocity is computed as

$$\dot{\mathbf{x}}_t = \frac{\mathbf{x}_{t+1} - \mathbf{x}_t}{\Delta t}$$

where  $\mathbf{x}_t$  is a multivariate position vector.

The acceleration is similarly computed as

$$\ddot{\mathbf{x}}_t = \frac{\dot{\mathbf{x}}_{t+1} - \dot{\mathbf{x}}_t}{\Delta t} = \frac{\mathbf{x}_{t+2} - 2\mathbf{x}_{t+1} + \mathbf{x}_t}{\Delta t^2}$$

# GMM/HMM/HSMM with dynamic features

$$\dot{\boldsymbol{x}}_t = \frac{\boldsymbol{x}_{t+1} - \boldsymbol{x}_t}{\Delta t}, \quad \ddot{\boldsymbol{x}}_t = \frac{\boldsymbol{x}_{t+2} - 2\boldsymbol{x}_{t+1} + \boldsymbol{x}_t}{\Delta t^2}$$

A vector  $\zeta_t$  will be used to represent the concatenated position, velocity and acceleration vectors at time step  $t$

$$\zeta_t = \begin{bmatrix} \boldsymbol{x}_t \\ \dot{\boldsymbol{x}}_t \\ \ddot{\boldsymbol{x}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\frac{1}{\Delta t} \mathbf{I} & \frac{1}{\Delta t} \mathbf{I} & \mathbf{0} \\ \frac{1}{\Delta t^2} \mathbf{I} & -\frac{2}{\Delta t^2} \mathbf{I} & \frac{1}{\Delta t^2} \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_{t+2} \end{bmatrix}$$

Here, the number of derivatives will be set up to acceleration ( $C=3$ ), but the results can easily be generalized to a higher or a lower number of derivatives.

A GMM/HMM/HSMM with centers  $\{\boldsymbol{\mu}_i\}_{i=1}^K$  and covariance matrices  $\{\boldsymbol{\Sigma}_i\}_{i=1}^K$  is first fit to the dataset  $[\zeta_1, \zeta_2, \dots, \zeta_T]$ .

## GMM/HMM/HSMM with dynamic features

$\zeta$  and  $\mathbf{x}$  are defined as large vectors concatenating  $\zeta_t$  and  $\mathbf{x}_t$  for all time steps

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_T \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_T \end{bmatrix}$$

Similarly to the matrix operator defined in the previous slide for a single time step, a large sparse matrix  $\Phi$  can be defined so that

$$\zeta = \Phi \mathbf{x}$$

# GMM/HMM/HSMM with dynamic features

$D$  dimensions  
 $C$  derivatives  
 $T$  time steps

$$\begin{bmatrix} \vdots \\ \boldsymbol{x}_t \\ \dot{\boldsymbol{x}}_t \\ \ddot{\boldsymbol{x}}_t \\ \boldsymbol{x}_{t+1} \\ \dot{\boldsymbol{x}}_{t+1} \\ \ddot{\boldsymbol{x}}_{t+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dots \\ \dots & -\frac{1}{\Delta t} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & \frac{1}{\Delta t^2} \mathbf{I} & \dots & \dots \\ \dots & \frac{1}{\Delta t^2} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots \\ \dots & \dots & \dots & \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots \\ \dots & -\frac{1}{\Delta t} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & \frac{1}{\Delta t^2} \mathbf{I} & \dots & \dots \\ \dots & \frac{1}{\Delta t^2} \mathbf{I} & -\frac{1}{\Delta t^2} \mathbf{I} & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{x}_t \\ \boldsymbol{x}_{t+1} \\ \boldsymbol{x}_{t+2} \\ \boldsymbol{x}_{t+3} \\ \vdots \end{bmatrix}$$

$\zeta \in \mathbb{R}^{DCT}$        $\Phi \in \mathbb{R}^{DCT \times DT}$        $\boldsymbol{x} \in \mathbb{R}^{DT}$   
 (C=3 here)

Note that  $\Phi$  is a big sparse matrix that can automatically be constructed through the use of Kronecker products.

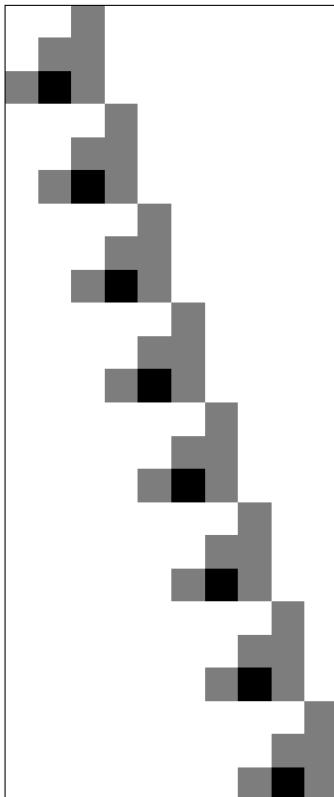
# GMM/HMM/HSMM with dynamic features

$$\zeta = \Phi x$$

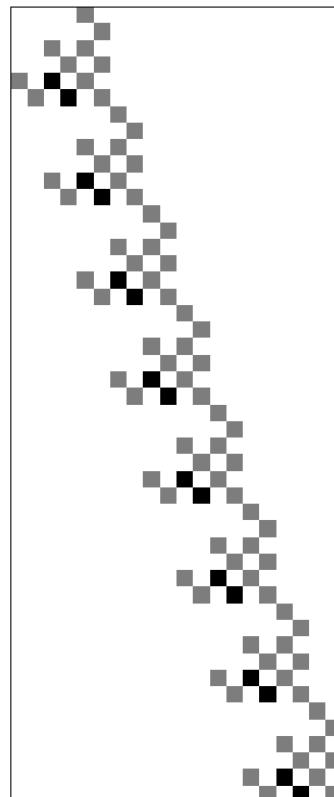
Sparse matrices

$$\Phi \in \mathbb{R}^{DCT \times DT}$$

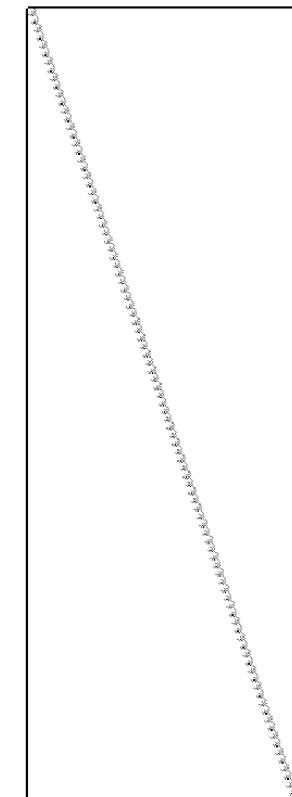
*D* dimensions  
*C* derivatives  
*T* time steps



D=1, C=3, T=10



D=2, C=3, T=10



D=2, C=3, T=100

# GMM/HMM/HSMM with dynamic features

For a sequence of states  $\mathbf{s} = \{s_1, s_2, \dots, s_T\}$  of  $T$  time steps, with discrete states  $s_t \in \{1, \dots, K\}$ , the likelihood of a movement  $\zeta = \Phi \mathbf{x}$  is given by

$$\mathcal{P}(\zeta | \mathbf{s}) = \prod_{t=1}^T \mathcal{N}(\zeta_t | \boldsymbol{\mu}_{s_t}, \boldsymbol{\Sigma}_{s_t})$$

where  $\boldsymbol{\mu}_{s_t}$  and  $\boldsymbol{\Sigma}_{s_t}$  are the center and covariance of state  $s_t$  at time step  $t$ . This product can be rewritten as

$$\mathcal{P}(\Phi \mathbf{x} | \mathbf{s}) = \mathcal{N}(\Phi \mathbf{x} | \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$$

$$\text{with } \boldsymbol{\mu}_s = \begin{bmatrix} \boldsymbol{\mu}_{s_1} \\ \boldsymbol{\mu}_{s_2} \\ \vdots \\ \boldsymbol{\mu}_{s_T} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_s = \begin{bmatrix} \boldsymbol{\Sigma}_{s_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{s_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Sigma}_{s_T} \end{bmatrix}$$

# GMM/HMM/HSMM with dynamic features

For example, for a sequence of states  $\mathbf{s} = \{1, 1, 2, 2, 3, 3, 3, 4\}$  with  $K=4$  and  $T=8$ , we have

$$\mu_s = \begin{bmatrix} \mu_1 \\ \mu_1 \\ \mu_2 \\ \mu_2 \\ \mu_3 \\ \mu_3 \\ \mu_3 \\ \mu_4 \end{bmatrix} \quad \text{and} \quad \Sigma_s = \begin{bmatrix} \Sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Sigma_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Sigma_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Sigma_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Sigma_4 \end{bmatrix}$$

$$\mu_s \in \mathbb{R}^{DCT}$$

$$\Sigma_s \in \mathbb{R}^{DCT \times DCT}$$

# GMM/HMM/HSMM with dynamic features

By using the relation  $\zeta = \Phi x$ , we seek to retrieve a trajectory

$$\hat{x} = \arg \max_x \log \mathcal{P}(\Phi x | s)$$

Equating to zero the derivative of

$$\mathcal{N}(\Phi x | \mu_s, \Sigma_s) = (2\pi)^{-\frac{DCT}{2}} |\Sigma_s|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2}(\Phi x - \mu_s)^\top \Sigma_s^{-1} (\Phi x - \mu_s)\right)$$

$$\log \mathcal{P}(\Phi x | s) = -\frac{1}{2}(\Phi x - \mu_s)^\top \Sigma_s^{-1} (\Phi x - \mu_s)$$

$$-\frac{1}{2} \log |\Sigma_s| - \frac{DCT}{2} \log(2\pi)$$

$$\frac{\partial}{\partial x} x^\top A x = (A + A^\top)x$$

with respect to  $x$  yields

$$\Phi^\top \Sigma_s^{-1} (\Phi x - \mu_s) = 0$$

Weighted  
Least Squares!

$$\iff \hat{x} = (\Phi^\top \Sigma_s^{-1} \Phi)^{-1} \Phi^\top \Sigma_s^{-1} \mu_s$$

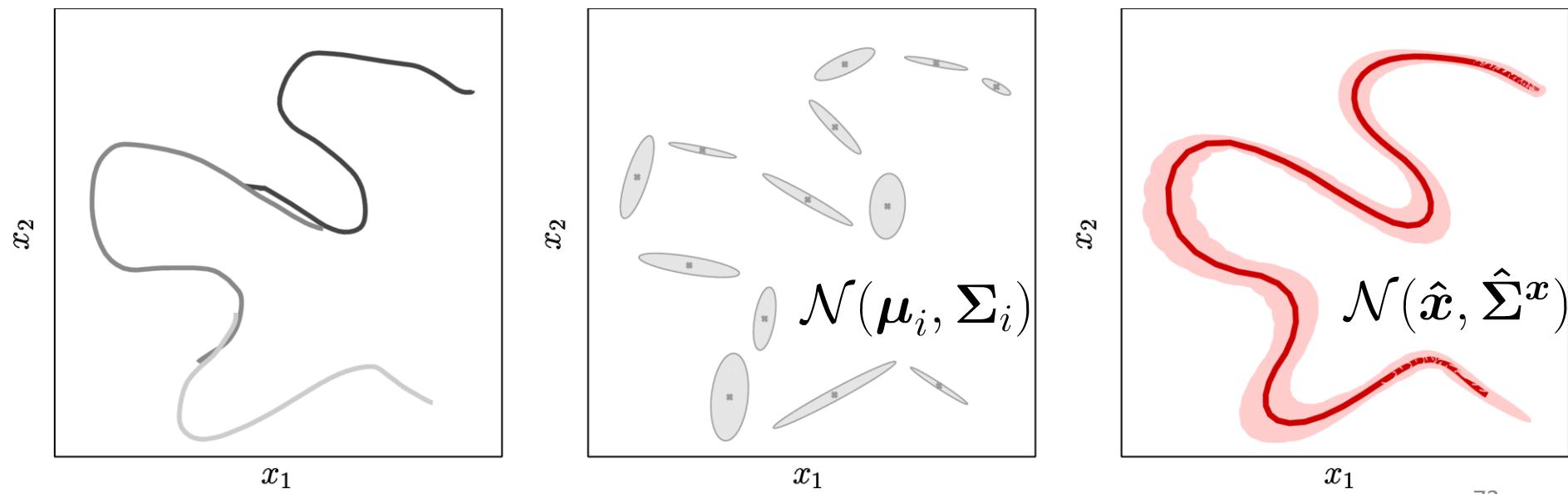
# GMM/HMM/HSMM with dynamic features

The covariance error of this estimate is given by

$$\hat{\Sigma}^x = \sigma (\Phi^\top \Sigma_s^{-1} \Phi)^{-1}$$

where  $\sigma$  is a scale factor.

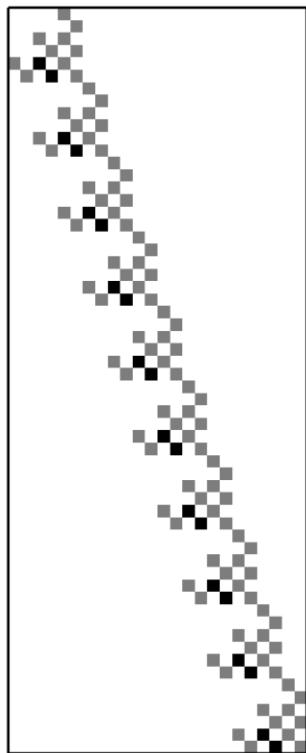
The resulting Gaussian  $\mathcal{N}(\hat{x}, \hat{\Sigma}^x)$  forms a trajectory distribution, where  $\hat{x} \in \mathbb{R}^{DT}$  is an average trajectory stored in a vector form.



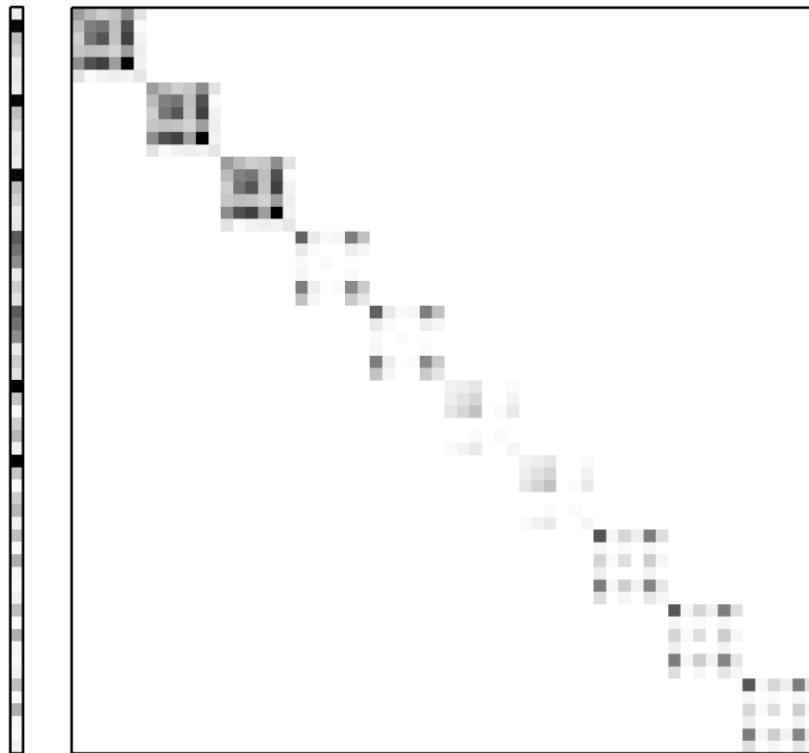
# GMM/HMM/HSMM with dynamic features

$$\hat{x} = (\Phi^\top \Sigma_s^{-1} \Phi)^{-1} \Phi^\top \Sigma_s^{-1} \mu_s$$

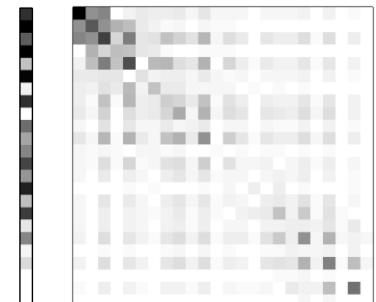
$$\hat{\Sigma}^x = \sigma (\Phi^\top \Sigma_s^{-1} \Phi)^{-1}$$



$\Phi$



$\mathcal{N}(\mu_s, \Sigma_s)$



$\mathcal{N}(\hat{x}, \hat{\Sigma}^x)$

# Main references

## Hidden Markov model (HMM)

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77:2:257–285, February 1989

## Hidden semi-Markov model (HSMM)

S.-Z. Yu. Hidden semi-Markov models. Artificial Intelligence, 174:215–243, 2010

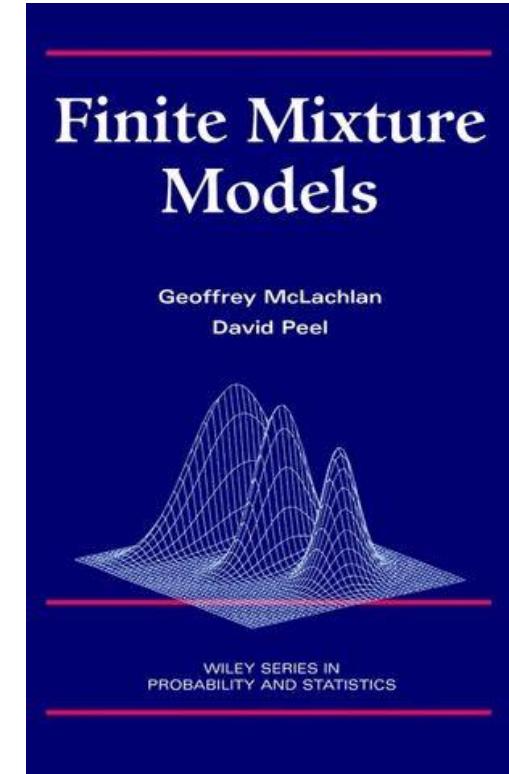
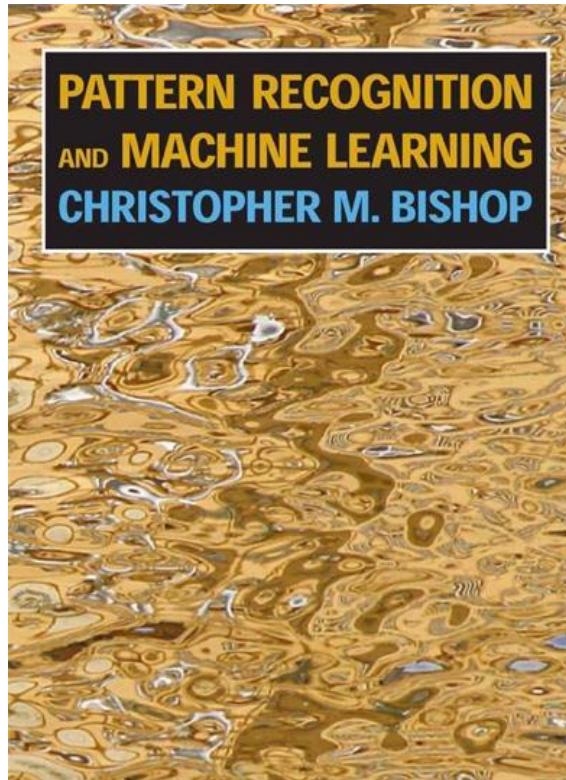
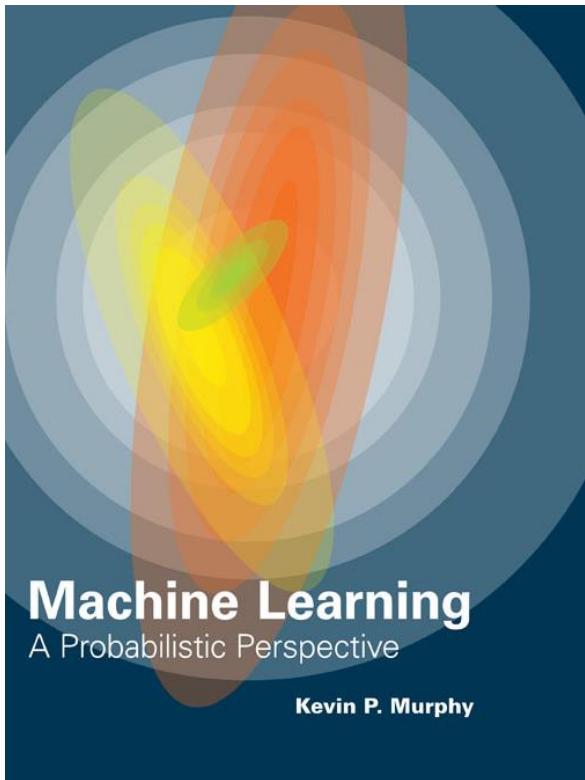
S. E. Levinson. Continuously variable duration hidden Markov models for automatic speech recognition. Computer Speech & Language, 1(1):29–45, 1986

## GMM/HMM/HSMM with dynamic features (Trajectory HMM)

S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Trans. on Acoustics, Speech, and Signal Processing, 34(1):52–59, 1986

H. Zen, K. Tokuda, and T. Kitamura. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. Computer Speech and Language, 21(1):153–173, 2007

# General textbooks



# **Advanced related research topics**

**(not covered in the course)**

## **Bayesian nonparametrics applied to HMM (model selection)**

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The Infinite Hidden Markov Model. In Advances in Neural Information Processing Systems (NIPS), pages 577–584. MIT Press, 2002

A. Roychowdhury, K. Jiang, and B. Kulis. Small-variance asymptotics for hidden Markov models. In Advances in Neural Information Processing Systems (NIPS), pages 2103–2111, 2013

## **Spectral learning of HMM parameters**

D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. Computer and System Sciences, 78(5):1460–1480, 2012

## **Task-parameterized HMM**

A. D. Wilson and A. F. Bobick. Parametric hidden Markov models for gesture recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(9):884–900, 1999