EECS 545: Machine Learning

Lecture 20. Hidden Markov Models

Honglak Lee and Michał Dereziński

03/23/2022

Outline

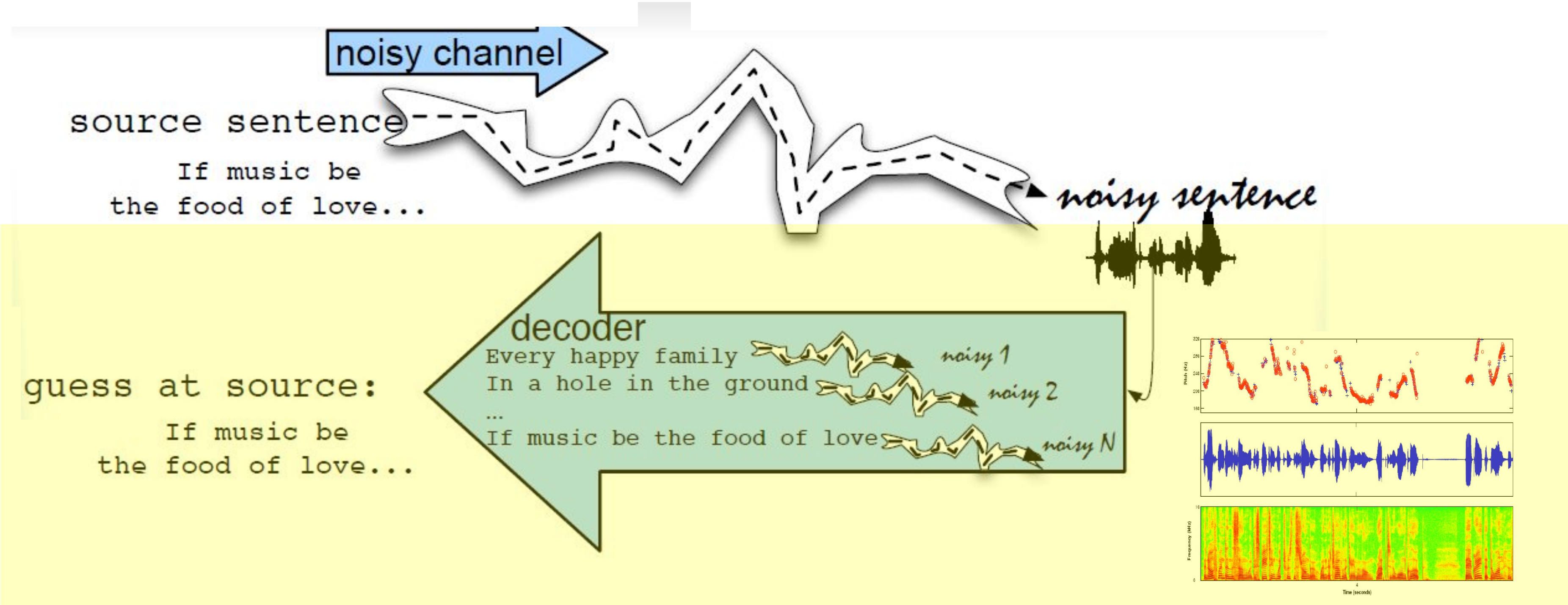
- Overview
- Markov Processes
- Hidden Markov Models
  - Representation
  - Inference
  - Learning
- Examples

Sequential Data

- Some data has intrinsic sequential structure.
  - Time series: speech, EKGs, stock market, robot sensors, etc.
  - Spatial sequences: DNA, natural language, etc.
- We could treat data points as i.i.d. samples
  - But that's false (they are not i.i.d.), so any conclusions we draw are likely to be wrong.
  - We are ignoring valuable constraints in the data.

Speech Recognition

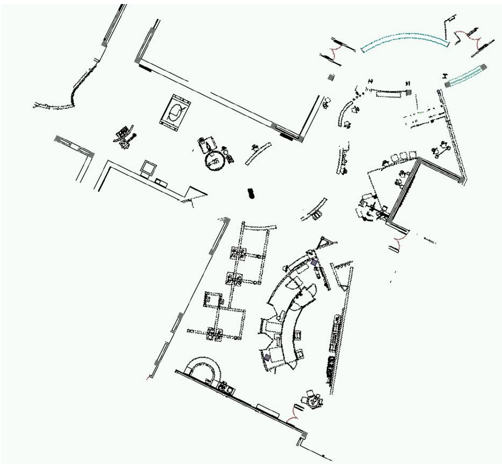
Underlying generative model (assumption)



Speech recognition can be tackled with HMMs

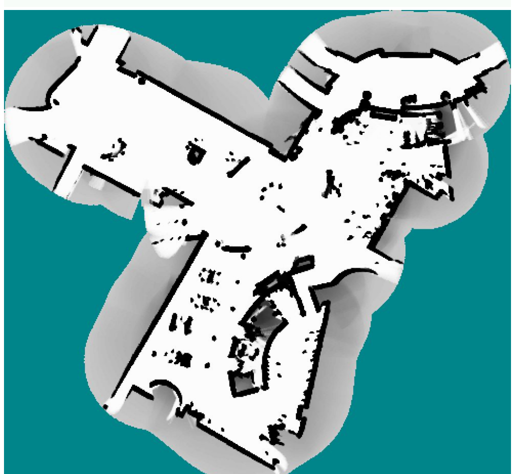
Robot Navigation: *SLAM*

Simultaneous Localization and Mapping



CAD Map

(S. Thrun, San Jose Tech Museum)



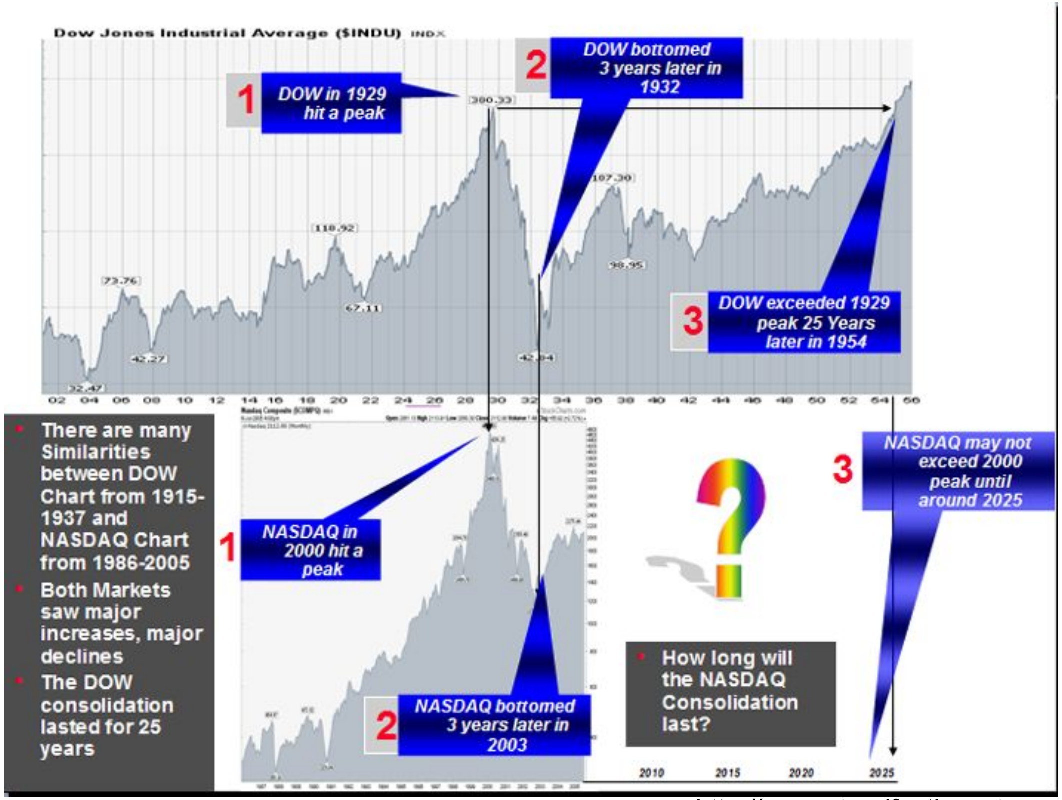
Estimated Map

Landmark SLAM

(E. Nebot, Victoria Park)

- As robot moves, estimate its pose & world geometry

Financial Forecasting



http://www.steadfastinvestor.com/

- Predict future market behavior from historical data, news reports, expert opinions, …

Analysis of Sequential Data

- Sequential structure arises in a huge range of applications
  - Repeated measurements of a temporal process
  - Online decision making & control
  - Text, biological sequences, etc
- Standard machine learning methods (assuming IID samples) are often difficult to directly apply
  - Do not exploit temporal correlations
  - Computation & storage requirements typically scale poorly to realistic applications

Markov Chains

- A **Markov chain** is a series of random variables $x_1, \ldots, x_T$, such that
$$p(x_t | x_1, \ldots, x_{t-1}) = p(x_t | \; x_{t-1})$$
- This is the *Markov property*, and can be summarized as:
  - *The future is independent of the past, given the present*.
- Often used to model temporal evolution.

Markov Models

- If a sequence has the Markov property

$$p(x_t|x_1,\ldots,x_{t-1}) = p(x_t|\ x_{t-1})$$

- then the joint probability distribution

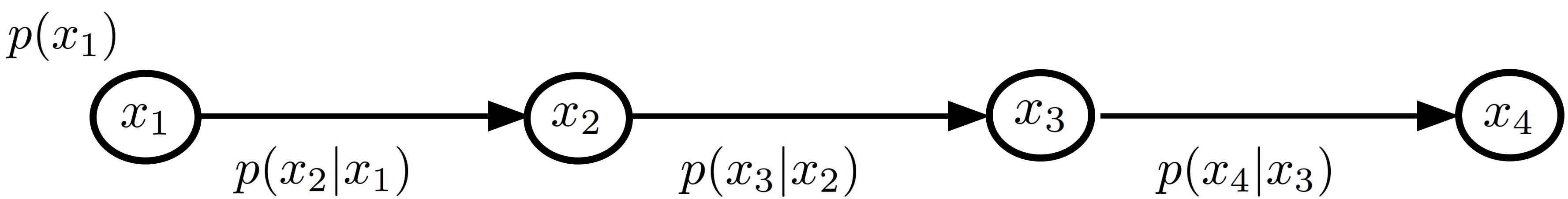
$$p(x_1,\ldots,x_T) = \prod_{t=1}^{T} p(x_t|x_1,\ldots,x_{t-1})$$

  – has a simplified form

$$p(x_1,\ldots,x_T) = p(x_1)\prod_{t=2}^{T} p(x_t|x_{t-1})$$

Markov Chains: Graphical Models

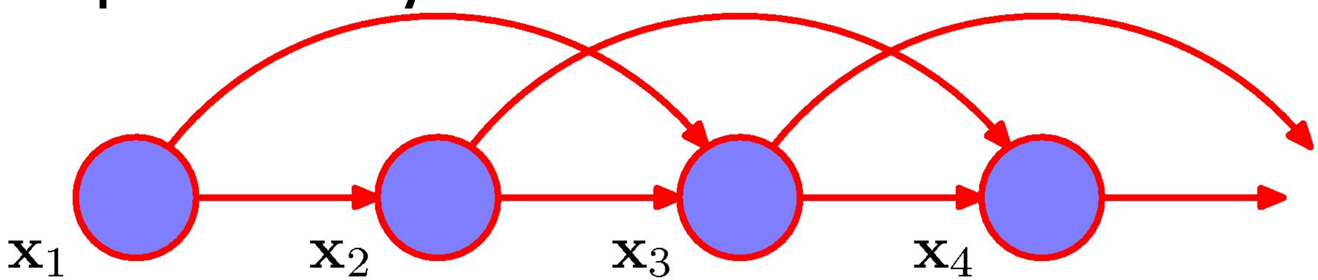
$$p(x_1, \ldots, x_T) = p(x_1) \prod_{t=2}^{T} p(x_t | x_{t-1})$$



- x_t are called states.
- $p(x_t|x_{t-1})$ are called transition probabilities.
- When the states are discrete, transition probability can be written as a matrix.

Higher-Order Markov Chains

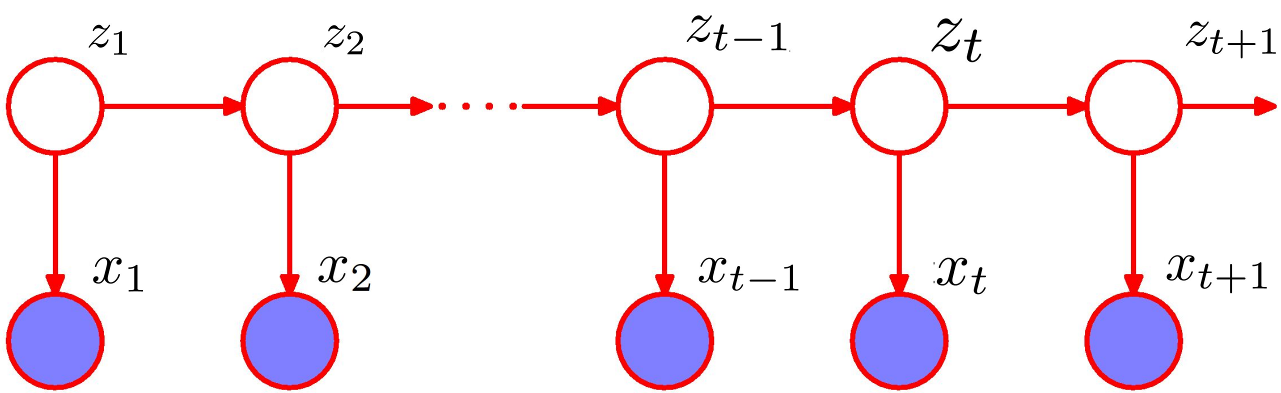
- We can extend the concept of Markov chain to more complex, but still local, kinds of dependency.



$$p(x_{1,\dots,}x_T) = p(x_1)p(x_2|x_1)\prod_{t=3}^{T}p(x_t|x_{t-1},x_{t-2})$$

Markov chain with latent variable

- For each observation x_t, we assume there is a latent variable z_t, and the z_t form a Markov chain.



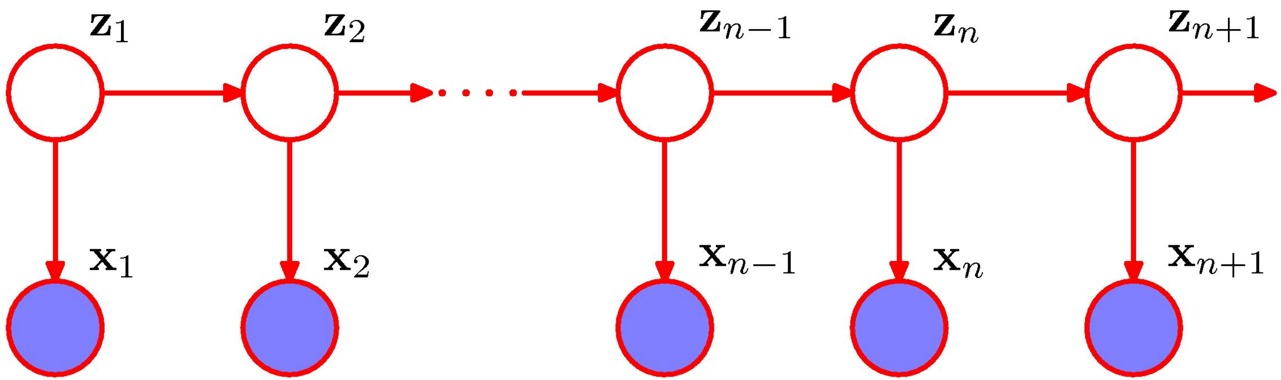
$$p(x_{1,\ldots,}x_T, z_{1,\ldots,}z_T) = p(z_1)\left[\prod_{t=2}^{T} p(z_t|z_{t-1})\right]\prod_{t=1}^{T} p(x_t|z_t)$$

Markov chain with latent variable

- This leads to
  - **Hidden Markov Models**
    - when the latent variable is discrete

today's focus

  - **Linear Dynamical Systems**
    - when the latent variable is Gaussian.



$$p(x_{1,\ldots,}x_T, z_{1,\ldots,}z_T) = p(z_1)\left[\prod_{t=2}^{T} p(z_t|z_{t-1})\right]\prod_{t=1}^{T} p(x_t|z_t)$$

Hidden Markov Models

- Prior distribution at the initial state:

parameters

$$p(z_1) \Rightarrow p(z_1|\pi) \qquad \pi$$

- Conditional distribution (transition table):

$$p(z_t|z_{t-1}) \Rightarrow p(z_t|z_{t-1}, A) \qquad A$$

- Emission probabilities of observables:

$$p(x_t|z_t) \Rightarrow p(x_t|z_t, \phi) \qquad \phi$$

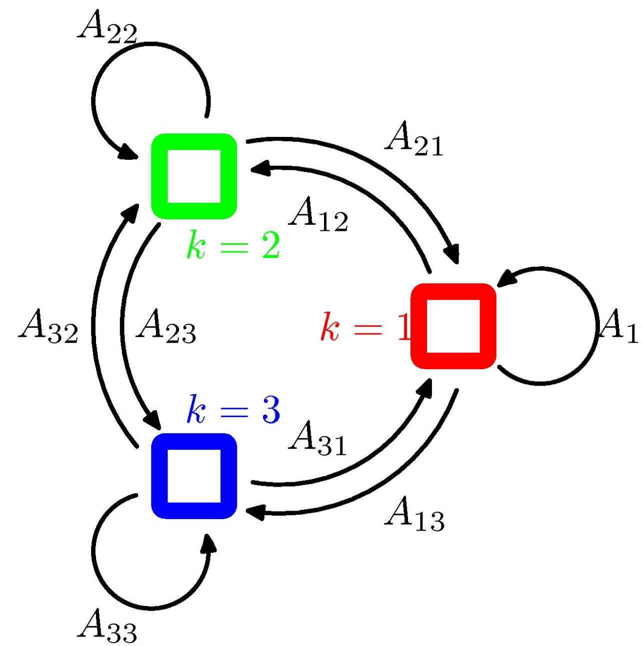
these distributions are also independent of t (i.e., shared across time)

Hidden Markov Models

- Use 1-of-*K* coding for values of z_t.
- A is the table of transition probabilities (indep. of *t*)

$$A_{jk} \equiv p(z_{tk}= 1|z_{t-1,j} = 1)$$

- This is *not* a graph of variables. These are transitions among values of *one* variable.

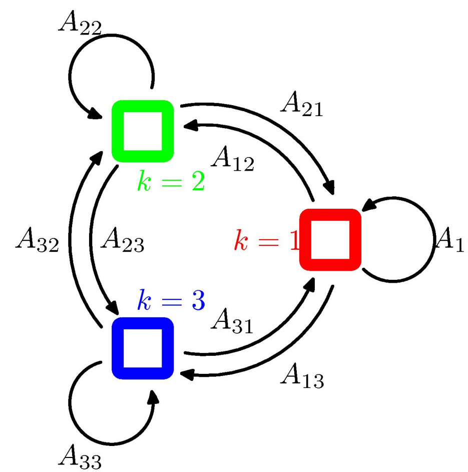


State transition diagram
(not a graphical model diagram)

Generative sampling from HMM

- Example:
  - Transition prob.: 90% of staying in the same state, 5% chance of transition to each other state.
  - Observation prob.: Gaussian distribution

Transition probabilities

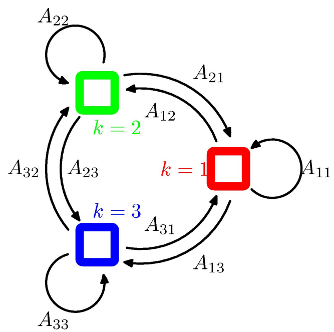


$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.05 & 0.90 & 0.05 \\ 0.05 & 0.05 & 0.90 \end{bmatrix}$$

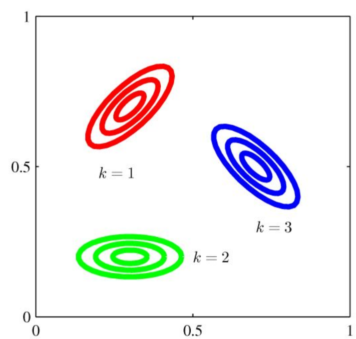
Generative sampling from HMM

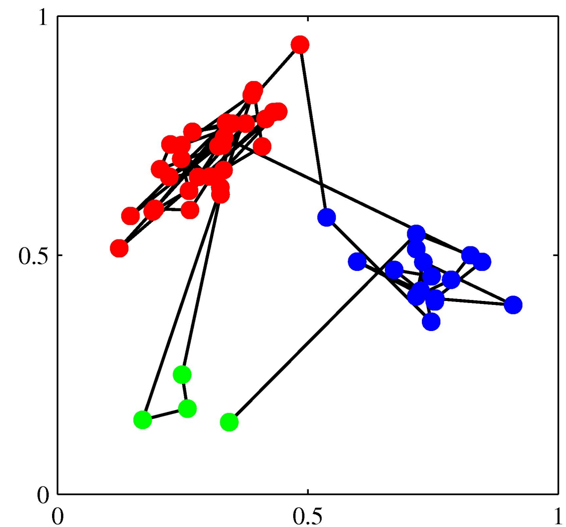
- Example:
  - Transition prob.: 90% of staying in the same state, 5% chance of transition to each other state.
  - Observation prob.: Gaussian distribution

Transition probabilities



Observation probabilities

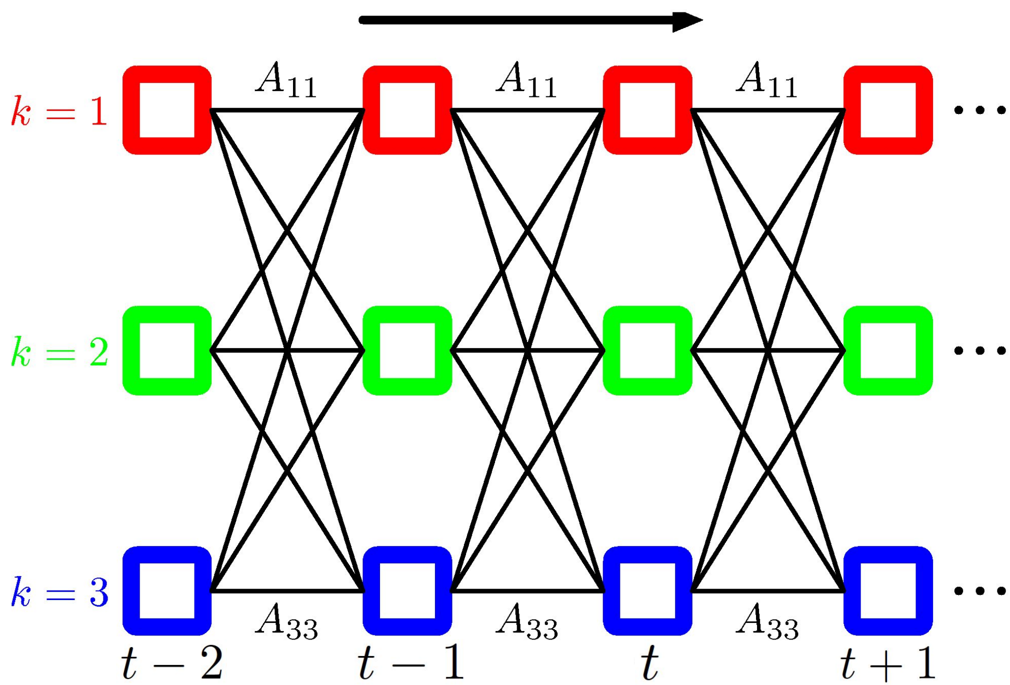




Samples from HMM

Hidden Markov Models

- Lattice representation of transition diagram



Hidden Markov Models

- The prior distribution at the initial state:

$$p(z_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$$

- The conditional distribution (transition table):

$$p(z_t|z_{t-1}, A) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{t-1,j} z_{t,k}}$$

- Emission probabilities of observables:

$$p(x_t|z_t, \phi) = \prod_{k=1}^{K} p(x_t|\phi_k)^{z_{t,k}}$$

Hidden Markov Models

- So, the overall joint probability distribution, over both observed and latent variables, is

$$p(X, Z|\theta)= p(z_1|\pi) \left[\prod_{t=2}^{T} p(z_t|z_{t-1}, A)\right]\prod_{m=1}^{T} p(x_m|z_m, \phi)$$

- The parameters are: $\theta = \{\pi, \mathbf{A}, \phi\}$
  - We can use EM to estimate these from data **X**.

Maximum Likelihood for the HMM

- Given a set X of observations, we want to use maximum likelihood to estimate the parameters $\theta = \{\pi, \mathbf{A}, \phi\}$
  – and the latent variables **Z**.

$$p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

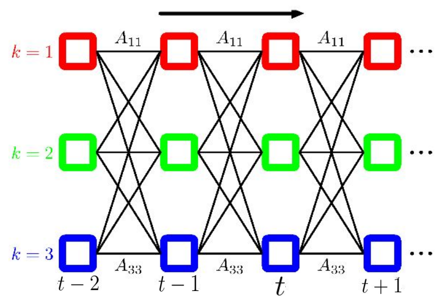
- To estimate the parameters of this latent variable model, we'll use the E-M algorithm.

E-M for HMMs

- The E-Step estimates the latent variables

$$q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}})$$

- The M-Step updates the parameters

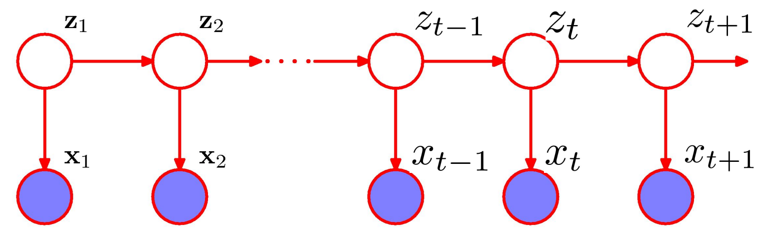


$$\theta = \{\pi, \mathbf{A}, \phi\}$$

$$\text{argmax}_\theta \ \mathcal{L}(q, \theta) = \text{argmax}_\theta \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}|\theta)$$

- After convergence, we have the maximum likelihood values of all parameters

E-M for HMMs



- **E-step** is evaluating $p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}})$
- A key term is $\gamma(z_t) = p(z_t|X) = \dfrac{p(X|z_t)p(z_t)}{p(X)}$
- Note that:

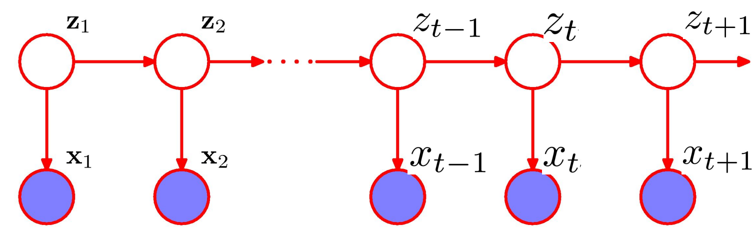
$$
\begin{aligned}
p(X|z_t) &= p(x_1, x_2, \cdots x_T|z_t) \\
&= p(x_1, x_2, \cdots x_t|z_t)p(x_{t+1}, x_{t+2}, \cdots, x_T|z_t, x_1, x_2, \cdots x_t) \\
&= p(x_1, x_2, \cdots x_t|z_t)p(x_{t+1}, x_{t+2}, \cdots, x_T|z_t)
\end{aligned}
$$

- Now, $\gamma(z_t) = \dfrac{p(x_1,\ldots,x_t,z_t)p(x_{t+1},\ldots,x_T|z_t)}{p(X)} = \dfrac{\alpha(z_t)\beta(z_t)}{p(X)}$
- where $\alpha(z_t) \equiv p(x_1, \ldots, x_t, z_t)$ $\beta(z_t) \equiv p(x_{t+1,\ldots,}x_T|z_t)$

Forward-Backward Algorithm **(E-step)**

$$\alpha(z_t) \equiv p(x_1, \ldots, x_t, z_t) \qquad \beta(z_t) \equiv p(x_{t+1,\ldots,}x_T|z_t)$$

- We'll prove the following recurrences:



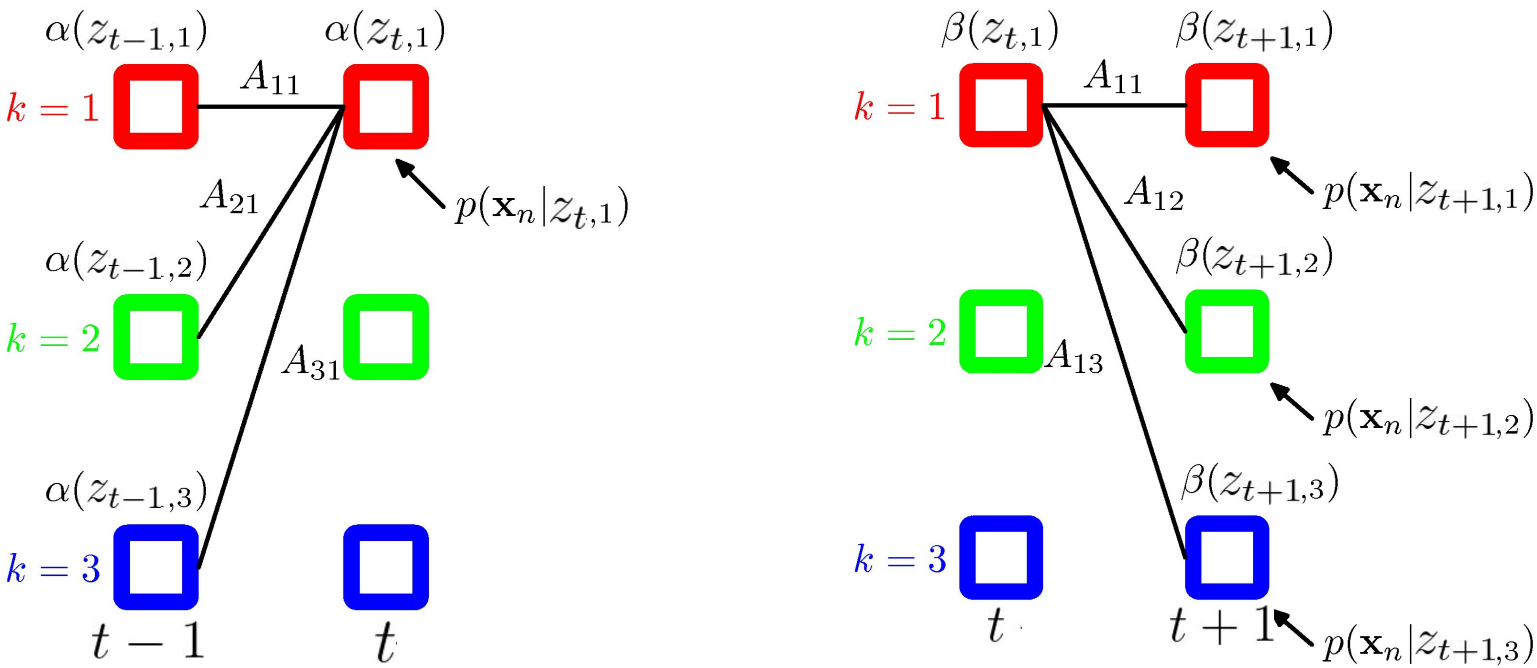
$$\alpha(z_t) = p(x_t|z_t) \sum_{z_{t-1}} \alpha(z_{t-1}) p(z_t|z_{t-1})$$

$$\beta(z_t) = \sum_{z_{t+1}} \beta(z_{t+1}) p(x_{t+1}|z_{t+1}) p(z_{t+1}|z_t)$$

- Note that recurrence for alpha is *forward* (dependent on past) while recurrence for beta is *backward* (dep. on future)

Forward-Backward Algorithm **(E-step)**

- Forward and Backward computations



Forward-Backward Algorithm **(E-step)**

Recurrence for alpha:

$$
\begin{aligned}
\alpha(z_t) &= p(x_1, \ldots, x_t, z_t) \\
&= p(x_t | x_1, \ldots, x_{t-1}, z_t) p(x_1, \ldots, x_{t-1}, z_t) && \text{[Cond. prob]} \\
&= p(x_t | z_t) \sum_{z_{t-1}} p(x_1, \ldots, x_{t-1}, z_{t-1}, z_t) && \text{[Marginalization]} \\
&= p(x_t | z_t) \sum_{z_{t-1}} p(z_t | x_1, \ldots, x_{t-1}, z_{t-1}) p(x_1, \ldots, x_{t-1}, z_{t-1}) && \text{[Cond. prob]} \\
&= p(x_t | z_t) \sum_{z_{t-1}} p(z_t | z_{t-1}) \alpha(z_{t-1})
\end{aligned}
$$

Forward-Backward Algorithm **(E-step)**

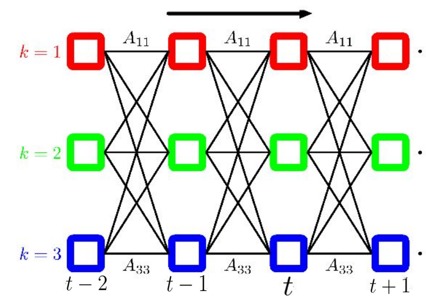
Recurrence for beta:

$$
\begin{aligned}
\beta(z_t) &= p(x_{t+1}, \ldots, x_T | z_t) \\
&= \sum_{z_{t+1}} p(x_{t+1}, \ldots, x_T, z_{t+1} | z_t) && \text{[Marginalization]} \\
&= \sum_{z_{t+1}} p(x_{t+1} | z_t, x_{t+2}, \ldots, x_T, z_{t+1})\, p(x_{t+2}, \ldots, x_T, z_{t+1} | z_t) && \text{[Cond. prob]} \\
&= \sum_{z_{t+1}} p(x_{t+1} | z_{t+1})\, p(x_{t+2}, \ldots, x_T | z_t, z_{t+1})\, p(z_{t+1} | z_t) && \text{[Cond. prob]} \\
&= \sum_{z_{t+1}} p(x_{t+1} | z_{t+1})\, p(x_{t+2}, \ldots, x_T | z_{t+1})\, p(z_{t+1} | z_t) \\
&= \sum_{z_{t+1}} p(x_{t+1} | z_{t+1})\, \beta(z_{t+1})\, p(z_{t+1} | z_t)
\end{aligned}
$$

E-M for HMMs

- The E-Step estimates the latent variables

$$q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}})$$



- The M-Step updates the parameters

$$\theta = \{\pi, \mathbf{A}, \phi\}$$

$$\text{argmax}_\theta\ \mathcal{L}(q, \theta) = \text{argmax}_\theta \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}|\theta)$$

- After convergence, we have the maximum likelihood values of all parameters

Q. Derive the update rule for M-step

M-step for HMMs

- Marginals of z given X (from E-step)

$$\gamma(z_t) = p(z_t | X, \theta^{old}) \xrightarrow{\text{gives us}} \xi(z_{t-1}, z_t) = p(z_{t-1}, z_t | X, \theta^{old})$$

- Data Completion likelihood

$$\mathrm{argmax}_\theta \, \mathcal{L}(q, \theta) = \mathrm{argmax}_\theta \sum_{\mathbf{Z}} q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z} | \theta)$$

$$= \mathrm{argmax}_\theta \, \sum_{k=1}^{K} \gamma(z_{1,k}) \ln \pi_k + \sum_{t=2}^{T} \sum^{K}{}_{j=1} \sum_{k=1}^{K} \xi(z_{t-1,j}, z_{t,k}) \ln A_{jk}$$

- M-step for state transitions

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

M-step for HMMs

- M-step for Observation probabilities
- Ex 1: Gaussian prob. $p(\mathbf{x}|\phi_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

$$\mu_k = \frac{\sum_{t=1}^{T} \gamma(z_{t,k}) x_t}{\sum_{t=1}^{T} \gamma(z_{t,k})} \qquad \sum_k = \frac{\sum_{t=1}^{T} \gamma(z_{t,k})(x_t - \mu_k)(x_t - \mu_k)^T}{\sum_{t=1}^{T} \gamma(z_{t,k})}$$

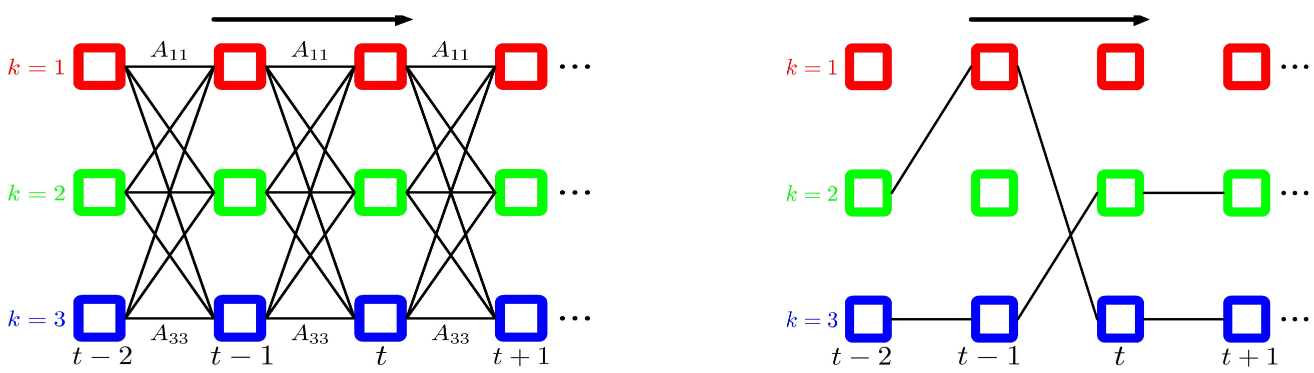
- Ex 2: Discrete (multinomial) prob. $p(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{D} \prod_{k=1}^{K} \mu_{ik}^{x_i z_k}$

$$\mu_{ik} = \frac{\sum_{t=1}^{T} \gamma(z_{t,k}) x_{t,i}}{\sum_{t=1}^{T} \gamma(z_{t,k})}$$

The Viterbi Algorithm

- Assume that we have estimated the parameters $\theta = \{\pi, \mathbf{A}, \phi\}$ of the HMM model
- Given a sequence X of observations, we want the **most likely sequence** Z of states (e.g., a MAP estimation).

$$\arg \max_{z_1, z_2, \ldots, z_T} p(z_1, z_2, \ldots z_T | x_1, x_2, \ldots, x_T) = \arg \max_{z_1, z_2, \ldots, z_T} p(z_1, z_2, \ldots z_T, x_1, x_2, \ldots, x_T)$$



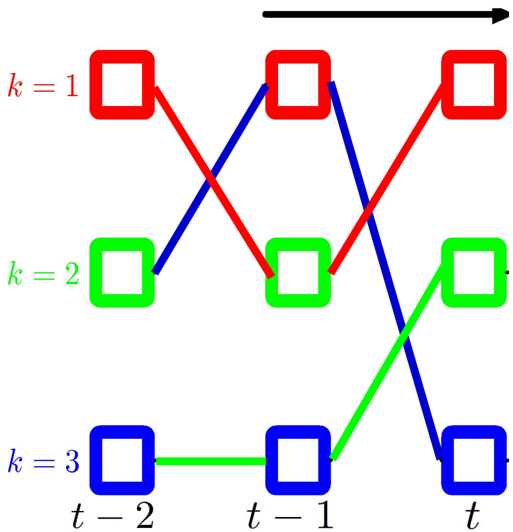
The Viterbi Algorithm

- Can use a Dynamic Programming algorithm, that is infact equivalent to shortest paths algorithm, due to recurrence:

$$p(z_1, \ldots z_t, z_{t+1}, x_1, \ldots, x_t, x_{t+1}) = p(z_1, \ldots z_t, x_1, \ldots, x_t)p(z_{t+1}|z_t)p(x_{t+1}|z_{t+1})$$

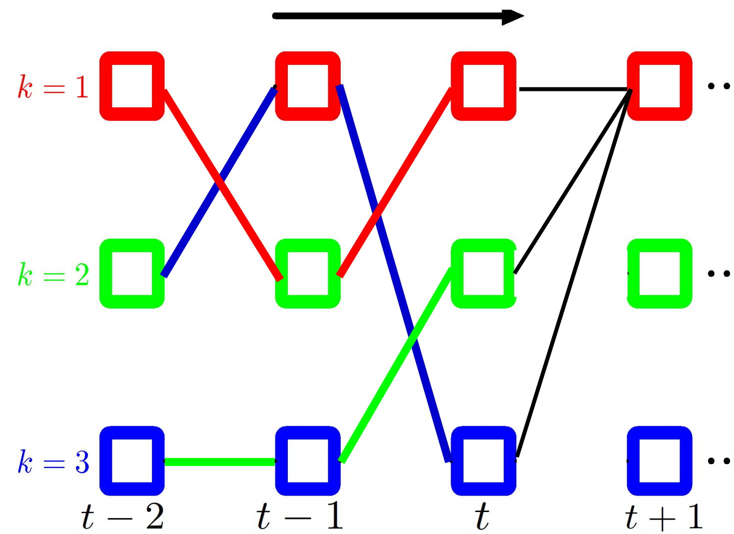
- For each state in z_t, keep track of
  – the probability of reaching that state,
  – the most likely path for reaching that state, and
  – the probability of that path (the Viterbi path).
- This can be updated to z_{t+1} in K^2 time.
  – Multiply by the emission probability of $\mathbf{x}^{(n)}$,
  – and all possible transition probabilities.

The Viterbi Algorithm



The optimal path (highest prob.) that leads to each state z_t is shown

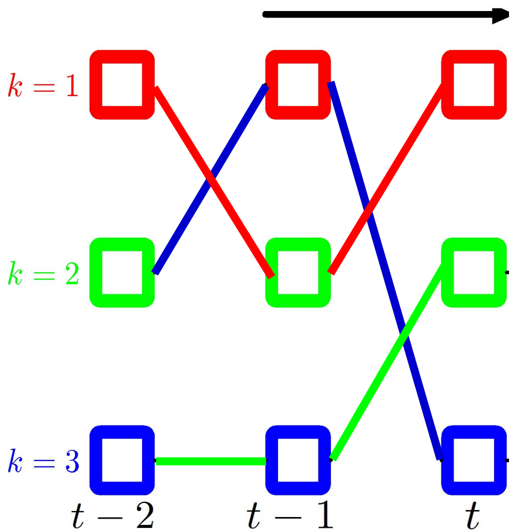
MAP assignment for z_t is the color with the highest prob among all the colored paths.

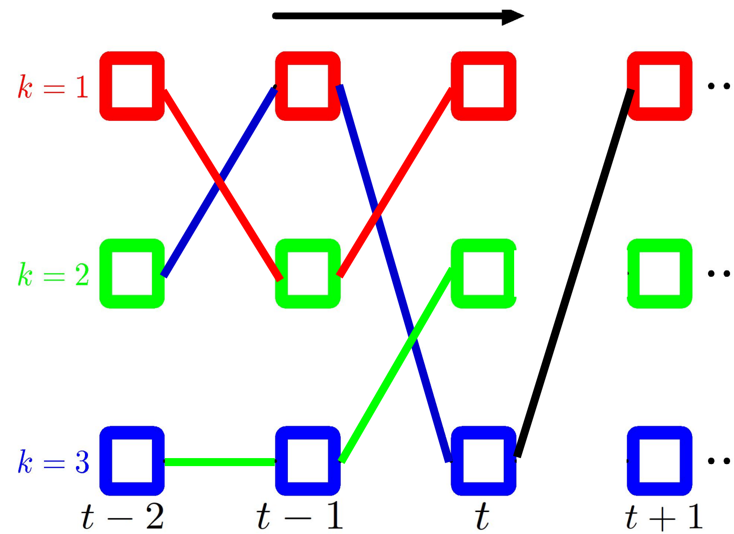


For each state, check which of the paths leads to the highest prob.

For e.g: The red state (k=1).

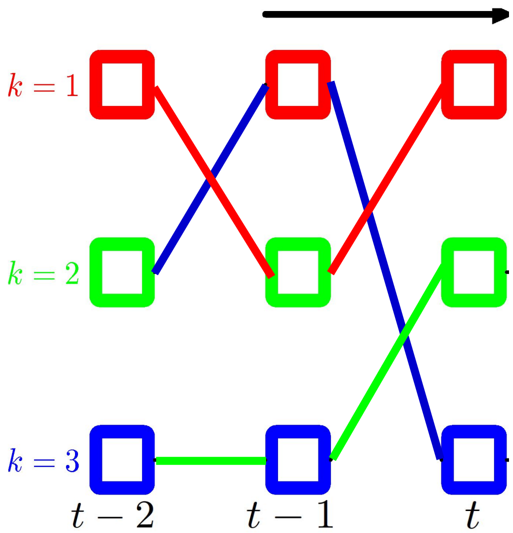
The Viterbi Algorithm

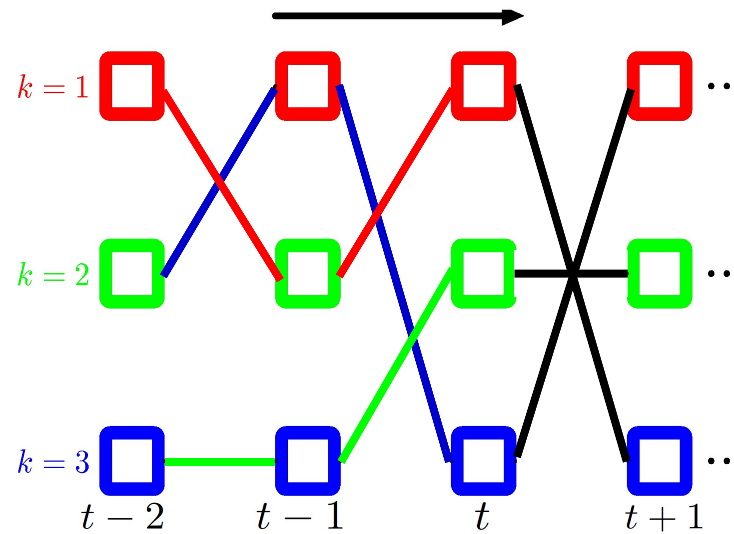




Discard the non-optimal paths

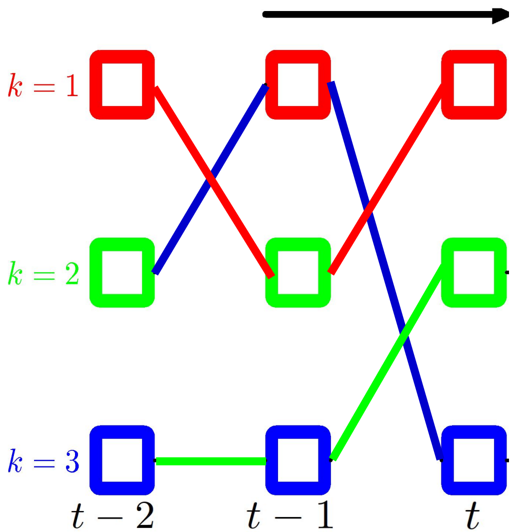
The Viterbi Algorithm

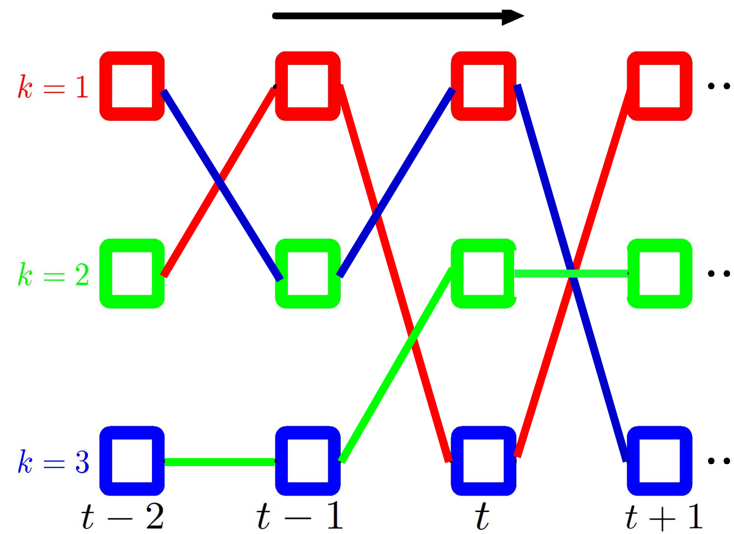




Repeat for all the states.

The Viterbi Algorithm

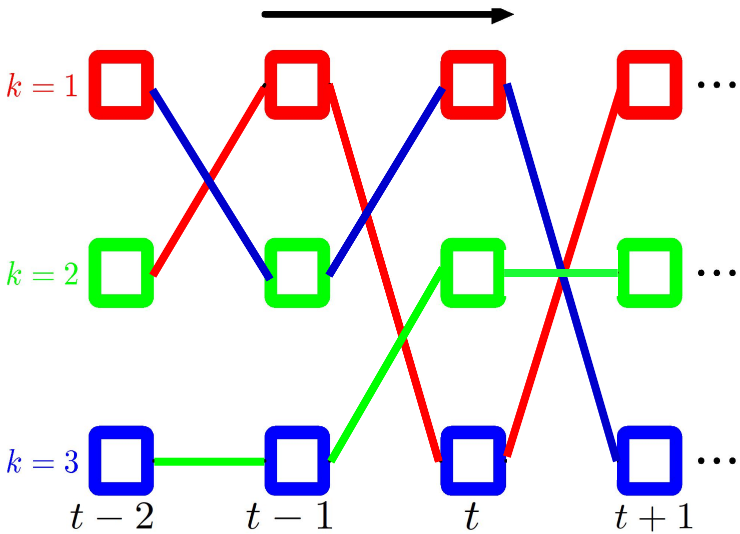




Colored to indicate optimal paths for each state.

Again, MAP assignment for z_{t+1} is the color with the highest prob among all the colored paths.

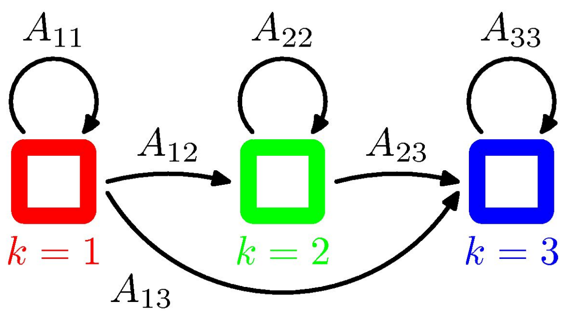
The Viterbi Algorithm

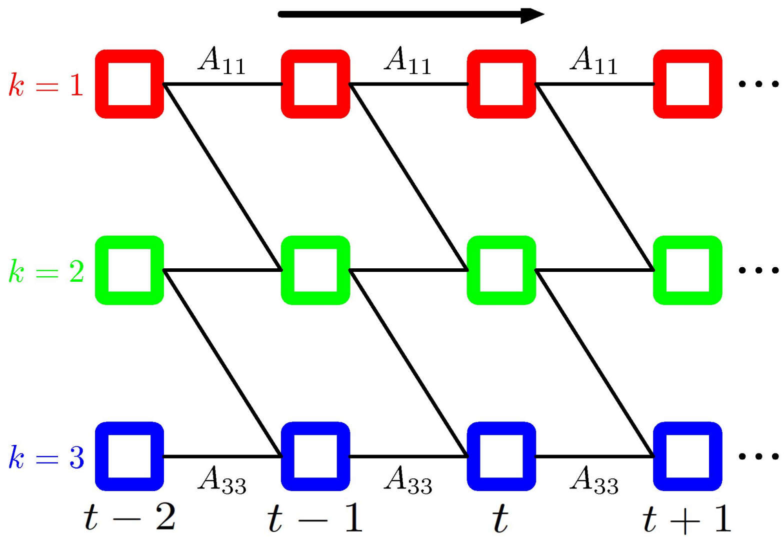


Now repeat the same steps till end time T.

Constraints on HMM transitions

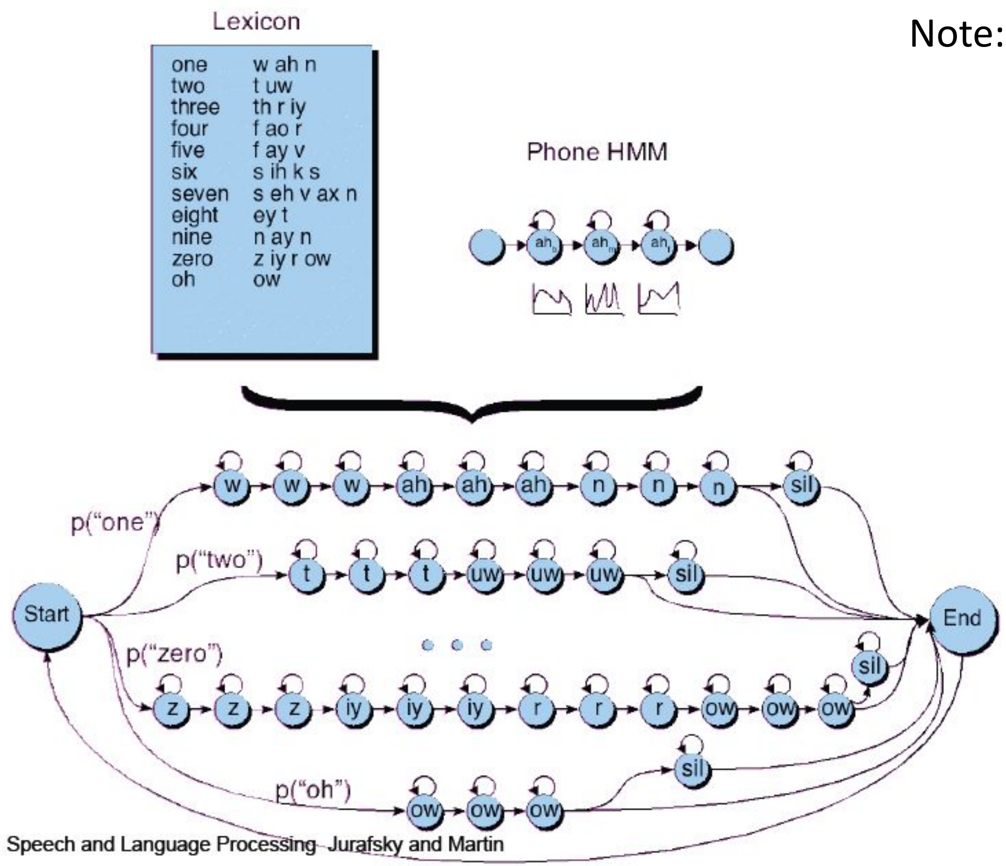
- Left-to-right constraint to describe a temporal process.
- Used for speech recognition





HMM for spoken digit recognition task

Note: This is state-transition diagram.



Related blog:
https://jonathan-hui.medium.com/speech-recognition-weighted-finite-state-transducers-wfst-a4ece08a89b7

Summary

- HMMs are useful in applications like speech recognition, robot navigation etc.

- HMM is latent variable models where the latents (or states) form a Markov chain

- The parameters of HMM can be estimated via the Expectation Maximization algorithm

- To infer the most likely sequence of latents (or states) for a test sample **x,** we can use the Viterbi algorithm (dynamic programming).

Thank you!

Quiz: Click [here]() or scan QR code

Next class: Reinforcement learning