02471 Machine Learning for Signal Processing

State-space models – Hidden Markov Models

Tommy Sonne Alstrøm

Cognitive Systems section

DTU Compute

Department of Applied Mathematics and Computer Science

Outline

- Last week review
- This week
- Probabilistic graphical models
- Examples
- The Hidden Markov Model
- Mathematical aspects of HMM
- Next week

Material: 15.1–15.3.1, 15.7, 16.4–16.5 (only 16.4–16.5 will be part of curriculum)

Course outline

So far:

- Parameter estimation [Regularization, biased estimation, mean squared error minimization].
- Signal representations [Time frequency analysis, sparsity aware learning, factor models].
- Filtering signals [Linear regression, adaptive filtering using stochastic gradient decent (LMS, NLMS, RLS), adaptive filtering using regularization].

Next weeks

- Inference and EM [Related to parameter estimation] (today).
- Sequential models [hidden markov models, linear dynamical systems, kalman filter].
- Kernel methods [non-linear models].

Last week review

Three approaches to prediction

Maximum likelihood

$$\hat{\boldsymbol{\theta}}_{\mathsf{ML}} = \arg\max_{\boldsymbol{\theta}} \; p(\mathcal{X}|\boldsymbol{\theta})$$

Maximum a posteriori

$$\hat{\boldsymbol{\theta}}_{\mathsf{MAP}} = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}|\mathcal{X})$$

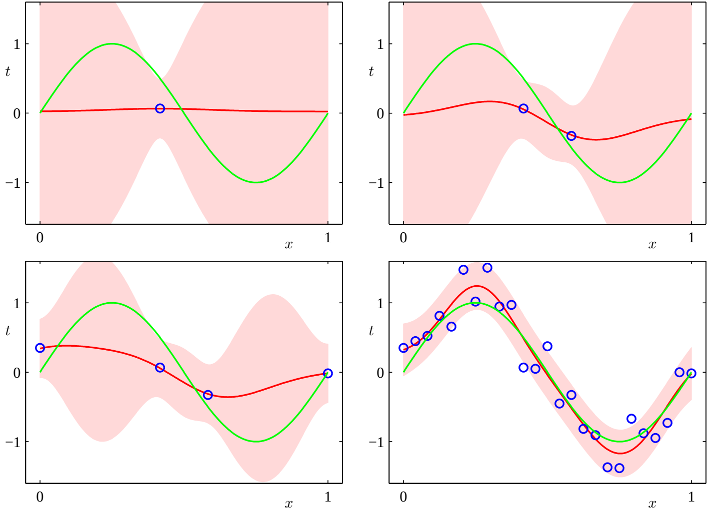
Use the estimated weights, $\hat{\boldsymbol{\theta}}$ to perform prediction, $\hat{y} = f(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$.

Posterior predictive distribution

$$p(y|\boldsymbol{x}, \mathcal{X}) = \int_{-\infty}^{\infty} p(y|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}) d\boldsymbol{\theta}$$

Use the mean of the posterior predictive distribution to perform prediction $\hat{y} = \mathbb{E}[p(y|\boldsymbol{x}, \mathcal{X})]$.

Why Bayesian? inherent uncertainty quantification



Source: Pattern Recognition and Machine Learning, 2006, C. Bishop

EM algorithm

Is an iterative algorithm, similar to coordinate descent

The EM algorithm

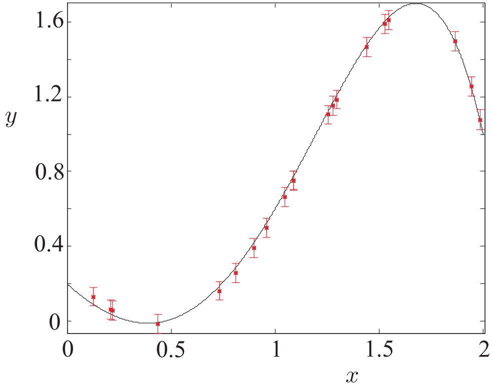
Steps to carry out before optimization:

- Specification of the complete log-likelihood, $\ln p(\boldsymbol{y}, \boldsymbol{\theta})$ (choice of model)
- Derive $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\left[\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]$ to create update formulas.

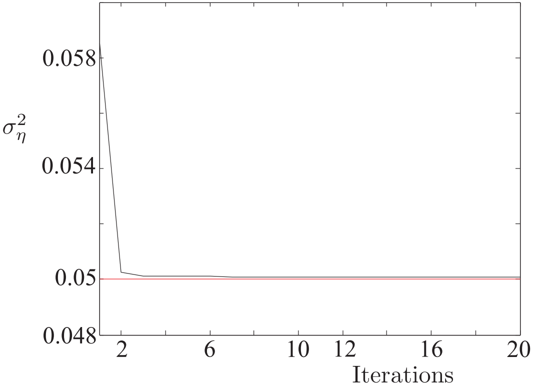
Randomly initialize $\boldsymbol{\xi}^{(0)}$ and run until convergence (e.g until $\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\| < \epsilon$)

1. Compute $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$
2. Maximize $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$ in order to get $\boldsymbol{\xi}^{(j+1)}$

EM results



0
0.4
0.8
1.2
1.6
y
0
0.5
1
1.5
2
x



0.048
0.05
0.054
0.058
σ_η^2
2
6
10
12
16
20
Iterations

Summary

- General linear models allows for modeling non-linearities but still keeps linear parameter estimation.
- Bayesian data modeling provide inherent uncertainty quantification since we learn the distributions, and not point estimates.
- Learning the models is more tedious.
- Often not tractable, but fully tractable if all distributions are Gaussian.
- EM can be used to learn distribution parameters.

A note on the material

This topic can be taught in a number of ways, because the material has been developed independently in different fields.

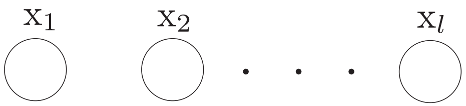
- In machine learning literature it is typically called Bayesian networks, which belongs to the area of "probabilistic graphical models".
- Statistics call them Markov models.
- Signal processing and time series analysis typically call it state-space models.

(They are not 100% exactly the same)

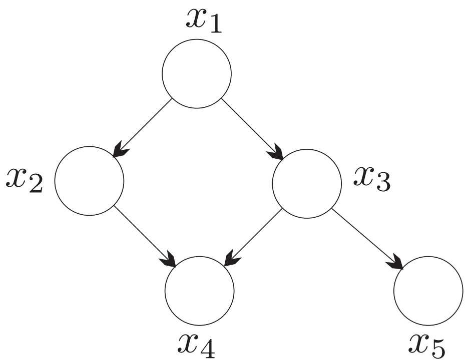
A note on the material: chapter 13 in "Pattern Recognition and Machine Learning by Christopher Bishop" `https://www.microsoft.com/en-us/research/people/cmbishop/` presents this material more coherently. In the ML book, this is unfortunately scattered across four chapters (4, 15, 16, and 17).

Probabilistic graphical models

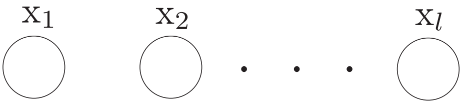
Our starting point, the graphical model



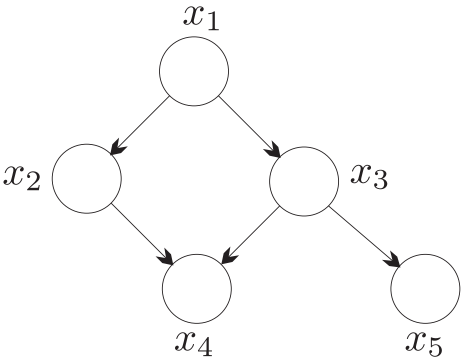
What does these figure represent?



Our starting point, the graphical model



What does this figure represent?

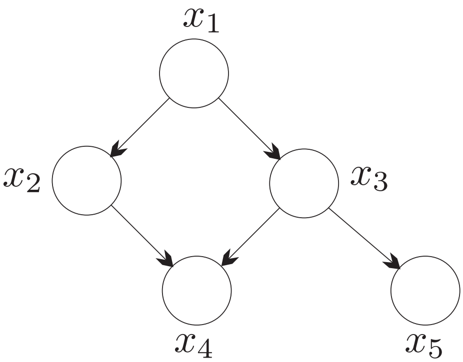


- x_1: Season
- x_2: Sprinkler on
- x_3: Rain
- x_4: Lawn wet
- x_5: Pavement wet

Bayesian networks and the Markov condition

Definition of a Bayesian network and the Markov condition

A Bayesian network structure is a directed acyclic graph (DAG) whose nodes represent random variables, and every variable (node), is conditionally independent of the set of all its non-descendants, given the set of all its parents. This is known as the Markov condition.

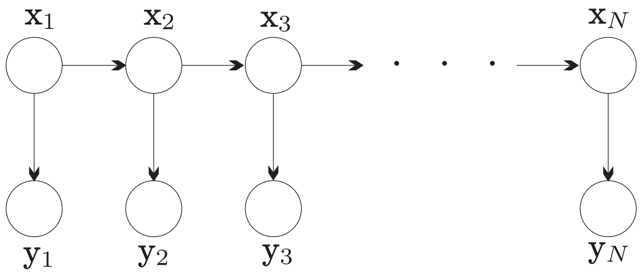


Example If we have observed x_3, rain or no rain, then x_5 the probability of the pavement being wet, is no longer dependent on x_1.

Formally we write

$$p(x_5|x_3, x_1) = p(x_5|x_3)$$

State-space model as a Bayesian network



State-space formulation

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n$$
$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n$$

Probabilistic formulation

$$p(\boldsymbol{x}_n | \boldsymbol{x}_{n-1}) = \mathcal{N}(\boldsymbol{x}_n; F_n \boldsymbol{x}_{n-1}, Q_n)$$
$$p(\boldsymbol{y}_n | \boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{y}_n; H_n \boldsymbol{x}_n, R_n)$$

How are Q_n and R_n related to the state-space formulation?

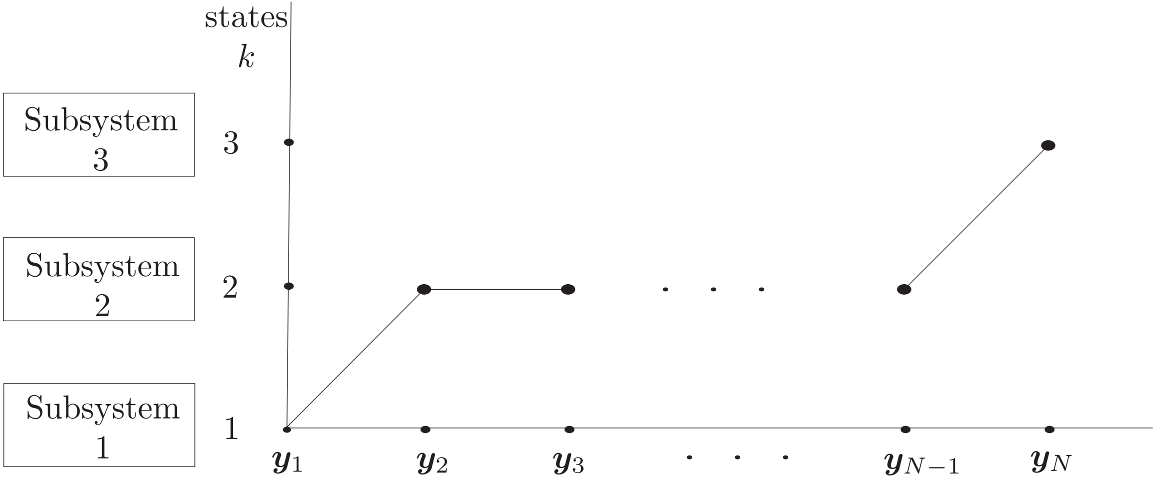
Linear dynamical system

Linear dynamical system

$$\begin{aligned} \mathbf{x}_n &= F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad \text{State equation} \\ \mathbf{y}_n &= H_n \mathbf{x}_n + \mathbf{v}_n, \quad \text{Observation equation} \end{aligned}$$

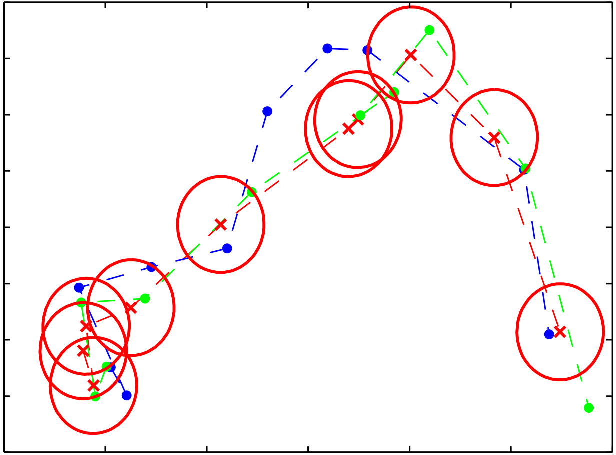
- If $\mathbf{x}_n$ is discrete, we call the model a Hidden Markov Model (HMM).
- If $\mathbf{x}_n$ is continuous and Gaussian, we call the model a Linear dynamical system (LDS).
- Additionally, if F_n, H_n, $\boldsymbol{\eta}_n$, and $\mathbf{v}_n$ are known, we call it Kalman filtering. Or more precisely, inference in a linear dynamical system is called Kalman filtering.

Example of a left-to-right HMM



Example: event detection.

Example of Kalman filter



Example: moving object tracking. Green: noisy measurements, blue: true location, red: predicted.

More inspiration

Introduction to Hidden Markov Models with Python Networkx and Sklearn: `http://www.blackarbs.com/blog/introduction-hidden-markov-models-python-networkx-sklearn/2/9/2017`

An example for implementing the Kalman filter for navigation where the vehicle state, position, and velocity are estimated by using sensor output from an inertial measurement unit (IMU) and a global navigation satellite system (GNSS) receiver: `https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications`

Location tracking `https://jonathan-hui.medium.com/self-driving-object-tracking-intuition-and-the-math-behind-kalman-filter-657d11dd0a90`

Linked material is not part of the curriculum, they are suggested to aid your learning process.

For PGMs, consider the course “42186 – Model-based machine learning”

The Hidden Markov Model

HMM

HMM model parameters

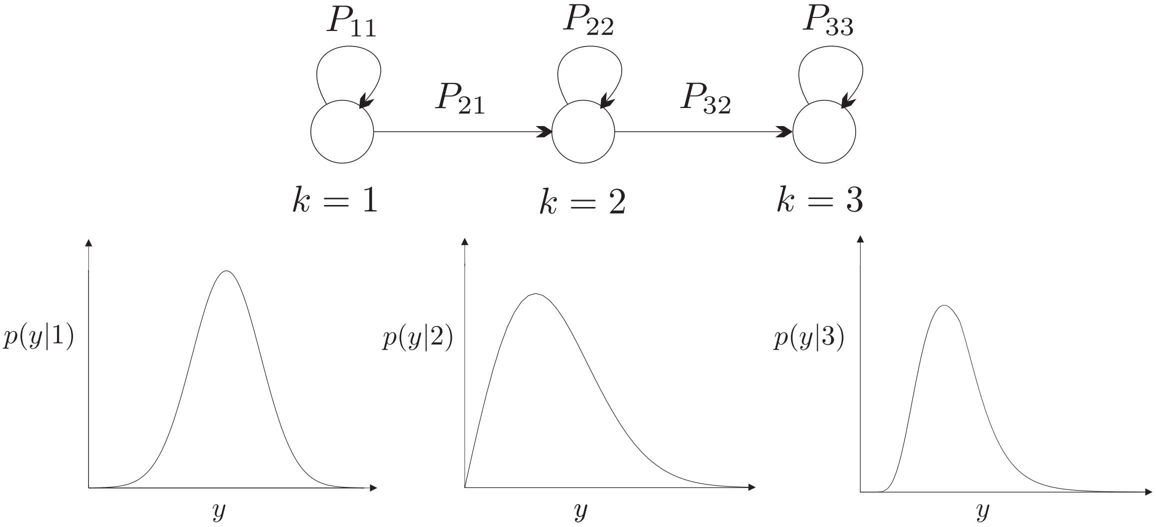
A HMM model is fully described by the following set of parameters:

1. Number of states K.
2. Initial state probability, P_k.
3. Transition probabilities, P_{ij}.
4. State emission distributions $p(\boldsymbol{y}|k)$.

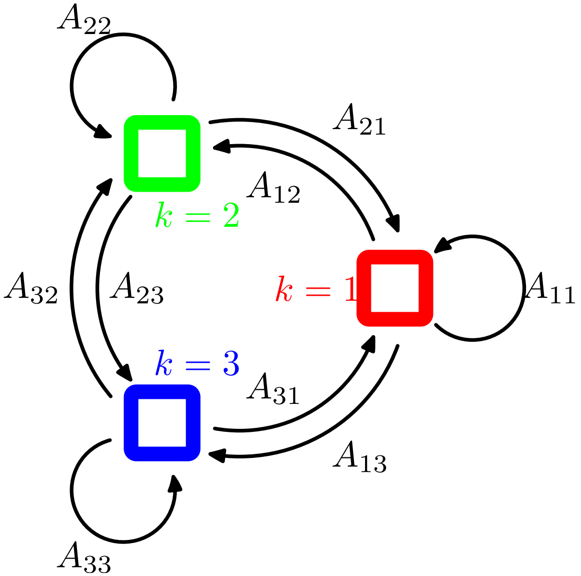
We can ask different questions:

- Given an observed sequence $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$, which HMM, out of a database of HMMs most likely generated the sequence? Example?
- Given an observed sequence $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$, which state k are we most likely in, or, what is the predicted value $\boldsymbol{y}_{n+1}$? (why don't we just use regression??)

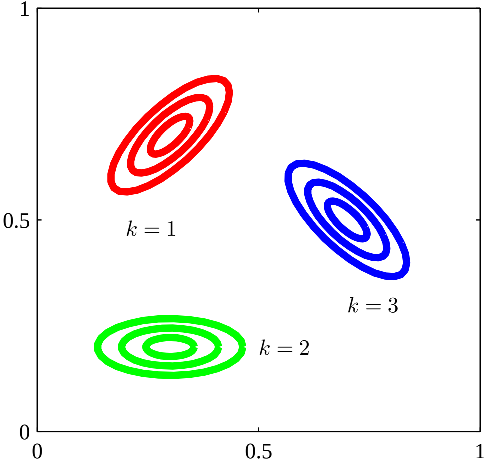
A three-state left-to-right HMM

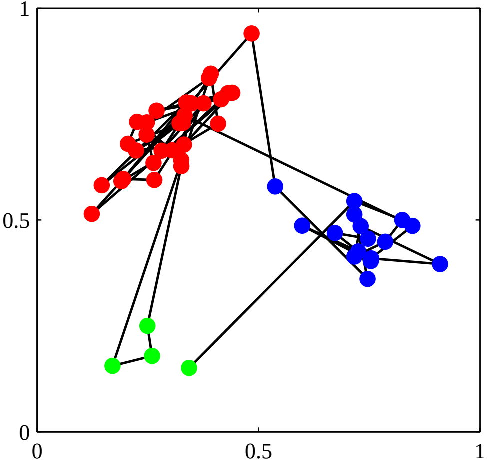


A full three-state HMM



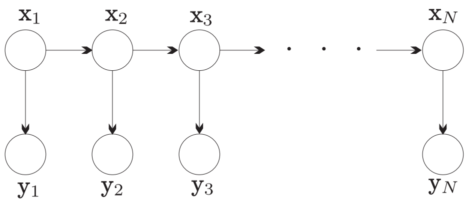
A simulation from a HMM





Put in the terms I have just talked about, what are we looking at?

Summary



- A hidden Markov model (HMM) models the situation where you have K distinct states of your system.
- The observations can be discrete or continuous.
- Is well suited for a number of applications, e.g. sound classification, classification of bytes in communication etc.

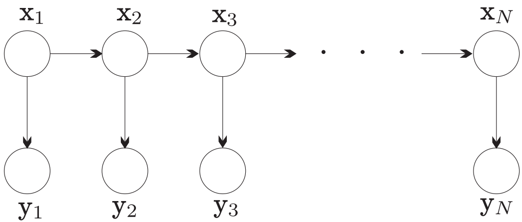
Mathematical aspects of HMM

How to perform prediction

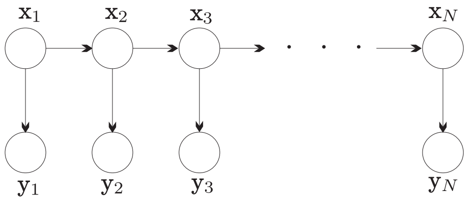
Sum rule: $P(x) = \sum_{y \in \mathcal{Y}} P(x, y)$

Product rule: $P(x, y) = P(x|y)P(y)$

Bayes Theorem: $P(y|x) = \frac{P(x,y)}{P(x)}$



How to learn the parameters

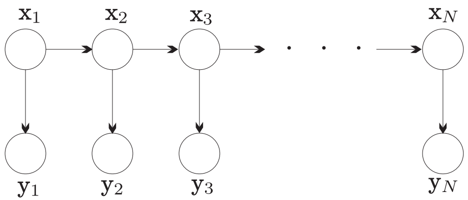


As a reminder, the EM algorithm consist of the following steps:

1. Specification of the complete log likelihood, $\ln p(\mathcal{X}, \mathcal{X}^l)$ (the model).
2. Derive $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})^{(j)})\big]$.
3. Maximize $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$ in order to get $\boldsymbol{\xi}^{(j+1)}$.

where $\mathcal{X}$ denotes the set of observations, $\mathcal{X}^l$ denotes the set of latent random variables, and $\boldsymbol{\xi}$ is a vector of distribution parameters.

How to learn the parameters



$$p(Y, X) = P(\boldsymbol{x}_1)p(\boldsymbol{y}_1|\boldsymbol{x}_1)\prod_{n=2}^{N} P(\boldsymbol{x}_n|\boldsymbol{x}_{n-1})p(\boldsymbol{y}_n|\boldsymbol{x}_n)$$

$$P(\boldsymbol{x}_1) = \prod_{k=1}^{K} P_k^{x_{1,k}}$$

$$P(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) = \prod_{i=1}^{K}\prod_{j=1}^{K} P_{ij}^{x_{n-1,j}x_{n,i}}$$

$$p(\boldsymbol{y}_n|\boldsymbol{x}_n) = \prod_{k=1}^{K} \left(p(\boldsymbol{y}_n|k;\boldsymbol{\theta}_k)\right)^{x_{n,k}}$$

Supplementary video material

Prof. Patterson describes the Hidden Markov Model, starting with the Markov Model
https://www.youtube.com/watch?v=J_y5hx_ySCg&list=PLix7MmR3doRo3NGNzrq48FItR3TDyuLCo

Alternative (mathematicalmonk) https://www.youtube.com/watch?v=7zDARfKVm7

Videos are not part of the curriculum, they are suggested to aid your learning process.

What I didn't tell you

- How do I perform (complete) inference (prediction) in HMM ? (ML, sec 16.5.1)
- How do I fully train the parameters in HMM? (require full EM updates, sec 16.5.2)
- How do I compute the most likely state, given a sequence (Viterbi ML, sec 15.7+16.5.2)

Lecture summary

- Presented the state-space model, which is a very popular ML model for sequential data.
- If $\mathbf{x}_n$ is discrete, we call the model a Hidden Markov Model (HMM).
- Is well suited for a number of applications, e.g. sound classification, classification of bytes in communication , gene sequencing, diagnosis etc, anywhere you have a finite number of states.
- HMM is trained using the EM algorithm.
- If $\mathbf{x}_n$ is continuous and Gaussian, we call the model a Linear dynamical system (LDS).
- For Linear dynamical systems, Kalman filtering is the prediction/update formulas (next week).
- We did not cover how to fully train/perform inference in HMM.

Next week

Week 47 material; 4.9–4.9.1, 4.10, 17.3

- State-space models (LDS)
- Kalman filter