Planning and Optimal Control

1. Markov Models

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

		A. Models for Sequential Data
Tue. 22.10.	(1)	1. Markov Models
Tue. 29.10.	(2)	2. Hidden Markov Models
Tue. 5.11.	(3)	3. State Space Models
Tue. 12.11.	(4)	3b. (ctd.)
		B. Models for Sequential Decisions
Tue. 19.11.	(5)	1. Markov Decision Processes
Tue. 26.11.	(6)	1b. (ctd.)
Tue. 3.12.	(7)	2. Introduction to Reinforcement Learning
Tue. 10.12.	(8)	3. Monte Carlo and Temporal Difference Methods
Tue. 17.12.	(9)	4. Q Learning
Tue. 24.12.	—	— *Christmas Break* —
Tue. 7.1.	(10)	5. Policy Gradient Methods
Tue. 14.1.	(11)	tba
Tue. 21.1.	(12)	tba
Tue. 28.1.	(13)	8. Reinforcement Learning for Games
Tue. 4.2.	(14)	Q&A

Outline

Outline

Sequence Data

Examples:

- DNA sequence
- sentences and texts
- physical sensor data
  - from machines in production: intelligent production, industry 4.0
  - physiological data from humans: ML for medicine
  - from cars: intelligent transport, automatic driving
  - speech, audio, video
- information systems
  - e-commerce and the web: page view sequences, market basket sequences
  - social media: short message streams
  - technology enhanced learning: learning management / student interactions

Sequence Data

other names:

- **time series**:
  - usually measured quantity is numeric
  - usually index is time
- **data stream**:
  - usually index is time
  - usually data is large (big data)

1. Classification/Regression/Prediction of a Sequence

- predict a target variable for instances being sequences
  - input is a sequence
  - output usually is a scalar
- examples:
  - classify EEGs of patients as depressed or healthy (classification)
  - predict the rating of a text review (regression, for a numeric rating scale)
- most evolved area: **time series classification**

2. Forecasting of a Sequence

- predict the value of a sequence in the future
  - input is a sequence
  - output is a scalar (of same type as the input)
- examples:
  - predict sales of a company for next quarter (based on past sales)
- very rich economic literature on **time series forecasting** (**econometrics**)
  - often for a single very long time series
- closely related to **2b. sequence imputation**
  - estimate values of a sequence at some positions where the value is missing

3. Sequence Prediction

- for instances, predict a sequence valued target
  - input is an attribute vector or a sequence
  - output is a sequence
- examples:
  - predict sequence of exercises a student should work on to learn most
  - predict sequence of ad expenses for a company to sell most
  - predict sequence of steering wheel movements to keep a car on a lane
- **planning** is a special case
  - likely the most important one
  - from ML perspective, sequence prediction is a special case of **structured prediction**
  - forecasting for several time points is another special case

4. Sequence Labeling / Sequence-to-sequence Learning

- predict a target for each index of a sequence
  - input is a sequence
  - output is a sequence of same length
- examples:
  - predict sequence of part-of-speech classes for every word of a sentence

Density estimation

Given a dataset $\mathcal{D}^{\text{train}} \subset \mathcal{X}$ sampled from an unknown distribution p, find a density model $\hat{p} : \mathcal{X} \rightarrow [0, 1]$ from a model space $\mathcal{M}$ s.t.

$$E_{x \sim p}\, \hat{p}(x) \geq E_{x \sim p}\, \hat{q}(x), \quad \forall \hat{q} \in \mathcal{M}$$

Operational: s.t. for data $\mathcal{D}^{\text{test}} \subset \mathcal{X}$ sampled from the same distribution,

$$\prod_{x \in \mathcal{D}^{\text{test}}} \hat{p}(x) \geq \prod_{x \in \mathcal{D}^{\text{test}}} \hat{q}(x), \quad \forall \hat{q} \in \mathcal{M}$$

What are Density Models Good for?

- **outlier analysis**:
  - the smaller $\hat{p}(x)$, the more unlikely/uncommon x is
  - this is an unsupervised / ill-defined problem
- **missing value imputation**:
  - given incomplete instances x (with values of some attributes not observed),
    find the values of the non-observed attributes
  - $=$ find the most likely complete instance $\bar{x}$ that has the same values as x for the observed attributes
- **classification/regression/prediction**:
  - build a class-specific density $p(X \mid Y)$ for instances of each class and use Bayes rule:

$$p(Y \mid X) \propto p(X \mid Y)\, p(Y)$$

  - as **Linear Discriminant Analysis** and **Naive Bayes classifiers**

Naive Bayes Densities for Sequences

Density models in Naive Bayes:

$$\hat{p}(X) := \prod_{m=1}^{M} \hat{p}(X_m)$$

$$p(x_m) := \frac{\text{freq}(x_m, \text{proj}_m \mathcal{D}^{\text{train}}) + 1}{|\mathcal{D}^{\text{train}}| + K_m}, \quad \text{for discrete } x_m \text{ with } K_m \text{ levels}$$

$$p(x_m) := \mathcal{N}(x_m; \bar{x}_m, \sigma_m^2), \quad \text{for continuous } x_m \text{ with average } \bar{x}_m \text{ and variance } \sigma_m^2$$

Applied to sequence data:

- density value does not depend on the order of the values

Note: $\text{proj}_m : \prod_{m=1}^{M} X_m \rightarrow X_m, x \mapsto x_m$ projection and
$\text{proj}_m \mathcal{D} := \{\text{proj}_m(x) \mid x \in \mathcal{D}\}$ for $D \subseteq \mathcal{X} := \prod_{m=1}^{M} X_m$.

Naive Bayes Densities for Sequences **are not useful**

Density models in Naive Bayes:

$$\hat{p}(X) := \prod_{m=1}^{M} \hat{p}(X_m)$$

$$p(x_m) := \frac{\text{freq}(x_m, \text{proj}_m \mathcal{D}^{\text{train}}) + 1}{|\mathcal{D}^{\text{train}}| + K_m}, \quad \text{for discrete } x_m \text{ with } K_m \text{ levels}$$

$$p(x_m) := \mathcal{N}(x_m; \bar{x}_m, \sigma_m^2), \quad \text{for continuous } x_m \text{ with average } \bar{x}_m \text{ and variance } \sigma_m^2$$

Applied to sequence data:

- density value does not depend on the order of the values

$\rightsquigarrow$ **we need sequence density models: Markov models**

Note: $\text{proj}_m : \prod_{m=1}^{M} X_m \rightarrow X_m, x \mapsto x_m$ projection and $\text{proj}_m \mathcal{D} := \{\text{proj}_m(x) \mid x \in \mathcal{D}\}$ for $D \subseteq \mathcal{X} := \prod_{m=1}^{M} X_m$.

Outline

Markov Model

$$p(x) := p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2) \cdots p(x_T \mid x_{T-1})$$
$$= p(x_1) \prod_{t=2}^{T} p(x_t \mid x_{t-1}), \quad x \in X^*$$

- **Markov model**, **Markov chain**
- **homogeneous**, **stationary**, **time-invariant**:
  - $p(x_{t+1} \mid x_t)$ does not depend on t, i.e.,
  $$p(x_{t+1} \mid x_t) = p(x_{t'+1} \mid x_{t'}) \quad \forall t, t'$$
  - parameter tying: same parameters shared for multiple variables
  - models arbitrary number of variables
    using a fixed number of parameters: **stochastic process**
- **discrete-state**, **finite-state**: $X := \{1, \ldots, I\}$

Transition Matrix

for discrete-state Markov models:

$$A := (p(x_{t+1} = j \mid x_t = i))_{i,j=1,\dots,I} \qquad I \times I \text{ **transition matrix**}$$

$$\pi := (p(x_1 = i))_{i=1,\dots,I} \qquad I\text{-dim. **start vector**}$$

- (row-)**stochastic matrix**: $\sum_j A_{i,j} = 1$

discrete-state, stationary Markov models:

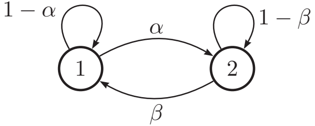
- equivalent to a **stochastic automaton**

Transition Matrix / State Transition Diagram

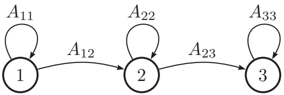
discrete-state, stationary Markov models:

- visualized as **state transition diagram**:
  - directed graph with
    - states as nodes and
    - edges for non-zero elements of A

- examples:



(a)



(b)

[source: Murphy 2012, p.590]

$$a)\ A := \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}, \quad b)\ A := \begin{pmatrix} 1-\alpha & \alpha & 0 \\ 0 & 1-\beta & \beta \\ 0 & 0 & 1 \end{pmatrix},$$

n-Step Transition Matrix $A(n)$

- get from i to j in exactly n steps

$$A(n) := (p(x_{t+n} = j \mid x_t = i))_{i,j=1,\ldots,I}$$

- can be computed simply by

$$A(n) = A^n$$

proof:

$$A(1) = A$$

$$A(n+m)_{i,j} = \sum_{k=1}^{I} A(m)_{i,k} A(n)_{k,j} = A(m)_{i,.} A(n)_{.,j}$$

$$A(n+m) = A(m)A(n)$$

$$A(n) = AA^{n-1} = AAA^{(n-2)} = \ldots = A^n$$

n-grams / subsequences

n-grams: (=subsequences of length n, windows)

$$\begin{aligned} \text{gram}_n : \quad X^* &\rightarrow (X^n)^* \\ x &\mapsto (x_{t:t+n-1})_{t=1,\ldots,|x|-n+1} \end{aligned}$$

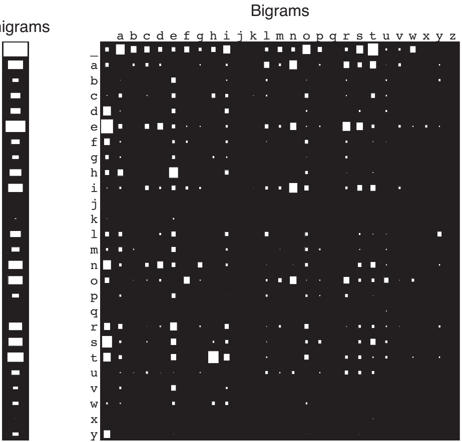
example:

$$\text{gram}_2((2,3,5,7)) = ((2,3),(3,5),(5,7))$$

Frequencies of 1- and 2-grams

Unigrams

1	0.16098	_
2	0.06687	a
3	0.01414	b
4	0.02938	c
5	0.03107	d
6	0.11055	e
7	0.02325	f
8	0.01530	g
9	0.04174	h
10	0.06233	i
11	0.00060	j
12	0.00309	k
13	0.03515	l
14	0.02107	m
15	0.06007	n
16	0.06066	o
17	0.01594	p
18	0.00077	q
19	0.05265	r
20	0.05761	s
21	0.07566	t
22	0.02149	u
23	0.00993	v
24	0.01341	w
25	0.00208	x
26	0.01381	y



[source: Murphy 2012, p.592]

letter grams in Darwin's *On the Origin of Species*.

Maximum Likelihood Estimator

$$\ell(A; \mathcal{D}) := \log \prod_{x \in \mathcal{D}} \pi_{x_1} \prod_{t=1}^{|x|-1} A_{x_t, x_{t+1}}$$

$$= \sum_{i=1}^{I} N_i^1 \log \pi_i + \sum_{i=1}^{I} \sum_{j=1}^{I} N_{i,j} \log A_{i,j}$$

$$N_i^1 := \text{freq}(i, \text{proj}_1 \mathcal{D}) = \sum_{n=1}^{N} \mathbb{I}(x_{n,1} = i)$$

$$N_{i,j} := \text{freq}((i,j), \text{gram}_2 \mathcal{D}) = \sum_{n=1}^{N} \sum_{t=1}^{|x_n|-1} \mathbb{I}(x_{n,t} = i, x_{n,t+1} = j)$$

Maximum Likelihood Estimator

$$\ell(A; \mathcal{D}) = \sum_{i=1}^{I} N_i^1 \log \pi_i + \sum_{i=1}^{I} \sum_{j=1}^{I} N_{i,j} \log A_{i,j}$$

under constraints $\sum_i \pi_i = 1$ and $\sum_j A_{i,j} = 1$ maximal for

$$\hat{\pi}_i := \frac{N_i^1}{\sum_{i'=1}^{I} N_{i'}^1}, \quad i = 1, \ldots, I$$

$$\hat{A}_{i,j} := \frac{N_{i,j}}{\sum_{j'=1}^{I} N_{i,j'}}, \quad i, j = 1, \ldots, I$$

or to avoid zeros in A, esp. for large I, sparse data:

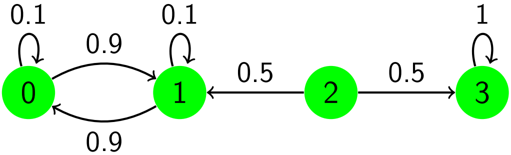
$$\hat{A}_{i,j} := \frac{N_{i,j} + 1}{(\sum_{j'=1}^{I} N_{i,j'}) + I}, \quad i, j = 1, \ldots, I$$

Long-Range Dependencies: Markov Models of Higher Order

- Markov models have no memory
  - future sequence depends on the past only through the last state
- easy to model dependencies on the last $h \geq 1$ states:
  - replace each data sequence x by the sequence $\text{gram}_h(x)$
  - $I^h \times I^h$ transition matrix from sequences X^h to X^h
    - but with structural zeros for all i, j with $i_{2:h} \neq j_{1:h-1}$
    - yields a $I^h \times I$ transition matrix from sequences X^h to X
  - I^h dim. start vector
- Markov model mechanism works out-of-the-box, e.g., MLE estimates
- number of parameters exponential in h
  - data sizes usually allow only small h

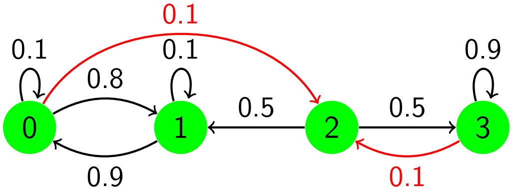
Outline

Communicating Classes



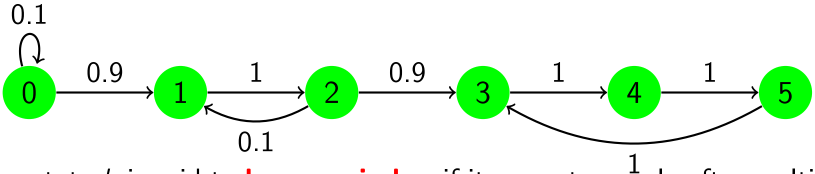
- state j is **accessible from state i**:
  - there is a path from i to j, e.g., there exists n: $(A^n)_{i,j} > 0$
- states i and j are **communicating**:
  - j is accessible from i and i is accessible from j
- set $K \subseteq X$ is a **communicating class**:
  - every state pair $i, j \in K, i \neq j$ is communicating and K is a maximal such set
- the state graph is partitioned in communicating classes
- a communicating class is **closed** if it cannot be left, i.e., there is no edge from any of its states to any state not belonging to the class

Communicating Classes / Irreducible Markov Chain



- *A* **irreducible**: it is a single communicating class, i.e., there is a path from every state to every state

State Periodicity

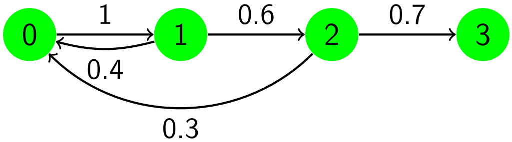


- a state k is said to **have period** m if it can return only after multiples of m steps, i.e.,

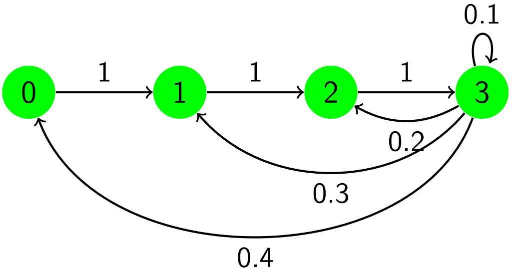
$$\text{period}(k) := \gcd\{n \in \mathbb{N} \mid (A^n)_{k,k} > 0\}$$

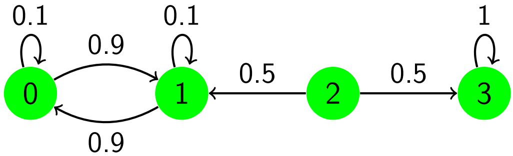
- a state with period 1 is called **aperiodic**

- all states of a communicating class have the same period

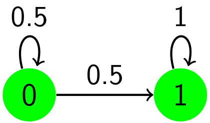


State Periodicity / More Examples





Transient vs. Recurrent States



- state k is **transient**: one possibly could never return to k, $\sum_{n=1}^{\infty}(A^n)_{k,k} < \infty$

$$\sum_{n=1}^{\infty}(A^n)_{0,0} = 0.5 + 0.5^2 + 0.5^3 + \ldots = 1 < \infty$$

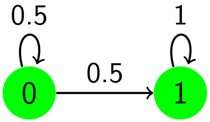
$$\sum_{n=1}^{\infty}(A^n)_{1,1} = 1 + 1 + 1 + \ldots \rightarrow \infty$$

- otherwise state k is called **recurrent**

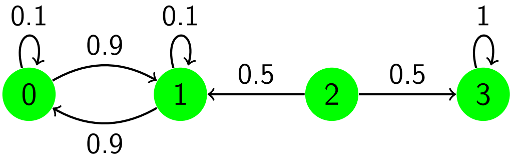
- all states of a communicating class are either transient or recurrent

- state k is **absorbing**: $A_{k,k} = 1$
  - thus $(0, 0, 0, 1)^T$ is a stationary state distribution

Transient vs. Recurrent States / Finite Discrete Case



- a communicating class is recurrent iff it is closed



Outline

Stationary State Distribution

- the transition matrix maps a distribution π of states to the distribution of their follow-up states:

$$\pi^{\text{next}} := A^T \pi$$

- For example, for the initial states $\pi^{(1)} := (p(x))_{x \in X}$:

$$\pi^{(2)} := A^T \pi^{(1)}$$

  is the distribution of states at time $t = 2$.

- Is there a fixpoint distribution π of states?

$$\pi = A^T \pi$$

  - π is called **stationary state distribution**

Stationary State Distribution

Lemma

Every row-stochastic matrix A has largest eigenvalue 1.

Proof.

- $\mathbf{1}$ is an eigenvector to eigenvalue 1 as:

$$A\mathbf{1} = \mathbf{1}$$

- Assume A would have an eigenvalue $\lambda > 1$, say with eigenvector x:

$$Ax = \lambda x$$

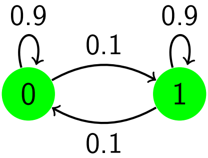
  Let $k \in \arg\max_k x_k$, then the k-the element of the left side is $\leq x_k$ (as convex combination of values $\leq x_k$), but of the right side is $\lambda x_k > x_k$. Contradiction.

□

A^T is column-stochastic, but has same eigenvalues as $(A^T)^T = A$.

Stationary State Distribution

- eigenvalues and eigenvectors can be computed using any eigenvalue algorithm
  - e.g., the QR algorithm [1]
  - eigenvectors need to be scaled to sum to 1 to yield a state distribution

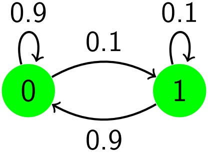


$$A = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\text{eigen}(A^T) = \{(1, \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}), (0.8, \begin{pmatrix} -0.707 \\ 0.707 \end{pmatrix})\}$$

Note: [1] `https://en.wikipedia.org/wiki/QR_algorithm`; numpy.linalg.eig

Stationary State Distribution

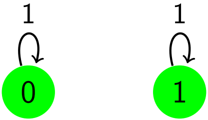


$$A = \begin{pmatrix} 0.9 & 0.1 \\ 0.9 & 0.1 \end{pmatrix}$$

$$\text{eigen}(A^T) = \{(1, \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}), (0, \begin{pmatrix} -0.707 \\ 0.707 \end{pmatrix})\}$$

Stationary State Distribution

- but in general there may be **several** stationary state distributions

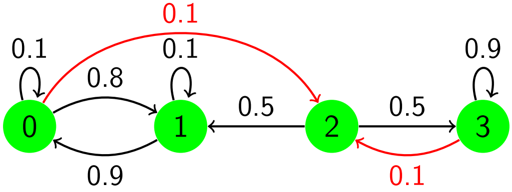


$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{eigen}(A) = \{(1, \begin{pmatrix} 1 \\ 0 \end{pmatrix}), (1, \begin{pmatrix} 0 \\ 1 \end{pmatrix})\}$$

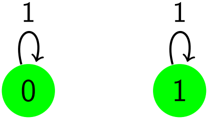
Unique Stationary State Distribution / Finite Discrete Case

- for a finite discrete markov model,
  if it is irreducible and aperiodic,
  its stationary distribution will be **unique**.



$$\pi = (0.3125, 0.3125, 0.0625, 0.3125)^T$$

Counterexample Finite, Reducible, Aperiodic, Several Stationary Distributions



$$\pi^{(1)} = (1, 0)^T$$
$$\pi^{(2)} = (0, 1)^T$$

Outline

Character of the Lecture

This is an advanced lecture:

- I will assume good knowledge of Machine Learning I and II.
- Slides will contain major keywords, not the full story.
- For the full story, you need to read the referenced chapters in one of the books.

Exercises and Tutorials

- There will be a weekly sheet with 2 exercises
  handed out **each Tuesday** in the lecture.
  1st sheet will be handed out later this week, Thur. 24.10.

- Solutions to the exercises can be
  submitted until **next Tuesday noon, 12pm**
  1st sheet is due later than usual: Wed. 30.10. morning, 8am

- Tutorials **each Thursday 8am-10am** or **Friday 12pm-2pm**,
  1st tutorial next week, Fr. 01.11.

- Successful participation in the tutorial gives up to 10% bonus points for the exam.
  - group submissions are OK (but yield no bonus points)
  - Plagiarism is strictly prohibited and leads to expulsion from the

Exam and Credit Points

- There will be a written exam at end of term (2h, 4 problems).
- The course gives 6 ECTS (2+2 SWS).
- The course can be used in
  - International Master in Data Analytics (mandatory)
  - IMIT MSc. / Informatik / Gebiet KI & ML
  - Wirtschaftsinformatik MSc / Informatik / Gebiet KI & ML & Wirtschaftsinformatik MSc / Wirtschaftsinformatik / Gebiet BI
  - as well as in all IT BSc programs.

Some Books

- Kevin P. Murphy (2012):
  Machine Learning, A Probabilistic Approach, MIT Press.

- Richard S. Sutton and Andrew G. Barto. (^{2}2018):
  Reinforcement Learning: An Introduction, MIT Press.

  (PDF available online: `http://incompleteideas.net/book/the-book.html`)

- Dimitri P. Bertsekas (2007):
  Dynamic Programming and Optimal Control, 3rd ed. Vols. I and II.

- David Silver (2015):
  Reinforcement Learning, lecture slides.

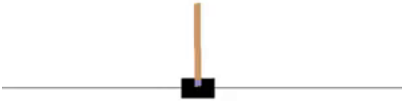
  (`http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html`)

- H. Geffner, B. Bonet (2013):
  A Concise Introduction to Models and Methods for Automated Planning.

Some First Software

- AI gym:
  several RL environments in Python
  (simple, atari etc.)

  (https://gym.openai.com)



Summary

- Many processes can be described by **sequence data**
  - aka **time series data**, **stream data**
- Several problems consist for sequence data:
  - **t.s. classification**: predicting the label of a sequence
  - **t.s. forecasting**: predicting future states of the sequence
  - **seq. labeling**: predict a sequence annotation,
    i.e., a scalar target at every index of the sequence
- **Markov models** are models for sequence data defined by
  - an **initial state density** $p(x_t)$ and
  - a **state transition density** $p(x_{t+1} \mid x_t)$

Summary (2/2)

- Markov Models called **discrete-state**, **finite-state** if there are only finite many discrete states.
  - then all densities are just probability distributions.
- Are called **homogeneous** if the densities do not depend on time.
- The state transition density of a homogeneous, finite-state Markov model can be represented just by a **transition matrix** ("tabular representation").
  - Their **Maximum Likelihood Estimate** are just the vector/matrix of relative frequencies of observed initial states and state transitions.
- To capture **long-range dependencies**, initial states and state transitions could be modeled dependent on the last h states, not just 1.

Further Readings

- Markov Models:
  Murphy 2012, chapter 17.

References

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.