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

Tue. 24.10.	(1)	1. Markov Models
Tue. 31.10.	—	— Luther Day —
Tue. 7.11	(2)	1b. Markov Models (ctd.)
Tue. 14.11.	(3)	2. State Space Models
Tue. 21.11.	(4)	3. Markov Random Fields
Tue. 28.11.	(5)	4. Markov Decision Processes
Tue. 5.12.	(6)	
Tue. 12.12.	(7)	5. Partially Observable Markov Decision Processes
Tue. 19.12.	(8)	
Tue. 26.12.	—	— Christmas Break —
Tue. 9.1.	(9)	6. Reinforcement Learning
Tue. 16.1.	(10)	
Tue. 23.1.	(11)	
Tue. 30.1.	(12)	
Tue. 6.2.	(13)	

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

- 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**

Why are Naive Bayes Densities not Useful for Sequences?

Density models in Naive Bayes:

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

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

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

Applied to sequence data:

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

Note: $\text{proj}_m : \prod_{m=1}^{M} X_m \to X_m, x \mapsto x_m$ 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:

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

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

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

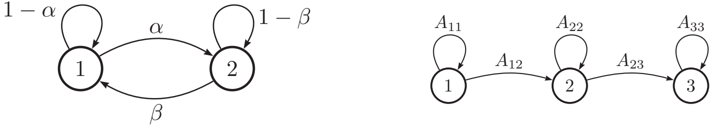
discrete-state, stationary Markov models:

- 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)

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

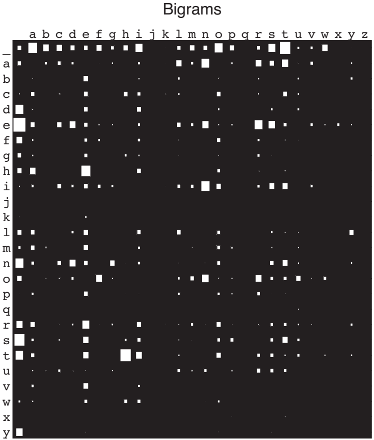
example:

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

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

1. ML Problems for Sequence Data

2. Markov Models

3. Organizational Stuff

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 a little bit late this week, Thur. 26.10.

- Solutions to the exercises can be
  submitted until **next Tuesday noon, 12pm**
  1st sheet is due a little bit late Wed. 1.11. morning, 8am

- Exercises will be corrected.

- Tutorials **each Thursday 8am-10am** or **Friday 12pm-2pm**,
  1st tutorial next week, Thur. 2.11.

- 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 illegal and usually leads to expulsion from the program.

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.

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

- D. Nau, M. Ghallab, P. Traverso (2004):
  Automated Planning: Theory and Practice.

- Steve LaValle (2006): *Planning Algorithms*.

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

- Richard S. Sutton and Andrew G. Barto. (1998):
  Reinforcement Learning: An Introduction.

Further Readings

- Markov Models:
  Murphy 2012, chapter 17.

References

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