Temporal Data and State Space Models

Peter Tino
School of Computer Science
University of Birmingham
UK

Temporal data

In Machine Learning we usually (conveniently!) assume that the data from which we need to learn to perform certain task(s) is generated i.i.d. from "Nature" (an underlying unobservable probability distribution).

Real-world

In particular this means that the output to be associated with a given input has nothing to do with the inputs presented before it.

In many real world situations this is simply not the case - daily values of stock values, EEG signals, base pairs in DNA, natural language etc.

Can you come up with more examples of such data and tasks to be performed on them?

Data evolving over time

The observations come in the form of **time series**.

They are generated from the (often unobservable!) underlying **process** (The "Nature") that evolves in time.

In oder to "generalize beyond the observations" we must somehow capture the "Nature" - i.e. the law describing how the underlying dynamics evolves in time.

Many "input time-lag window" (finite input memory) approaches take their inspiration from Taken's Theorem. The situation can get much more complicated once dynamic (and/or observational) noise is considered.

Let's start simple...

- A finite alphabet of abstract symbols
  $\mathcal{A} = \{1, 2, ..., A\}$, or $\mathcal{A} = \{a, b\}$, or $\mathcal{A} = \{G, C, T, U\}$ etc.

  sequence/word/string

- (Possibly infinite) strings over $\mathcal{A}$

- left-to-right processing of strings
  GCCGCCUTUUUUCCCCCTTTGCUUCCCGG...

Left-to-Right Processing of Sequences

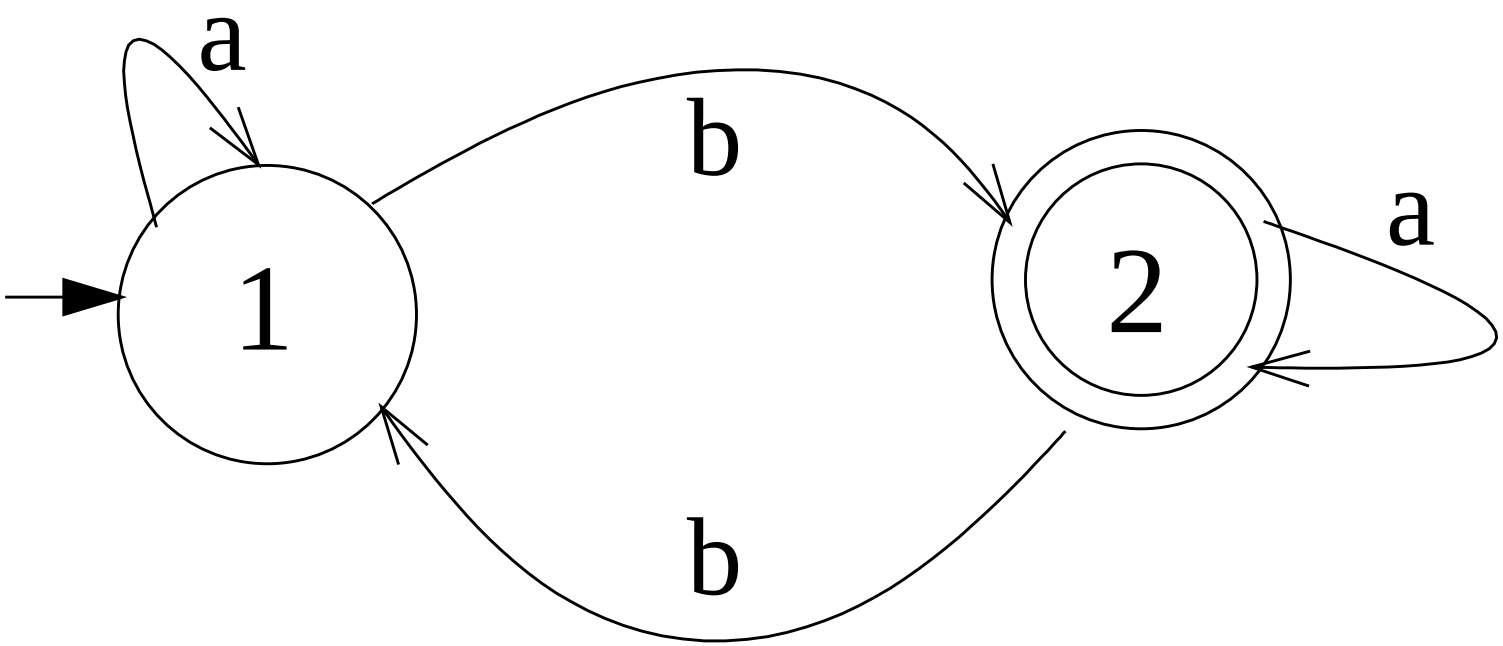
- One can attempt to reduce complexity of the processing task
- Group together histories of symbols that have "the same functionality" w.r.t. the given task (e.g. next-symbol prediction)
- **Information processing states (IPS)** are equivalence classes over sequences
- **IPS code what is "important" in everything we have seen in the past**

IPS - Discrete State Space, Inputs, Observations

A simple sequence classification example - FSA

- IPS - $1, 2$
- Inputs - a, b
- Outputs: $+1$ "Grammatical": odd number of b's, -1 "Non-Grammatical": even number of b's (including none)

Finite state automata



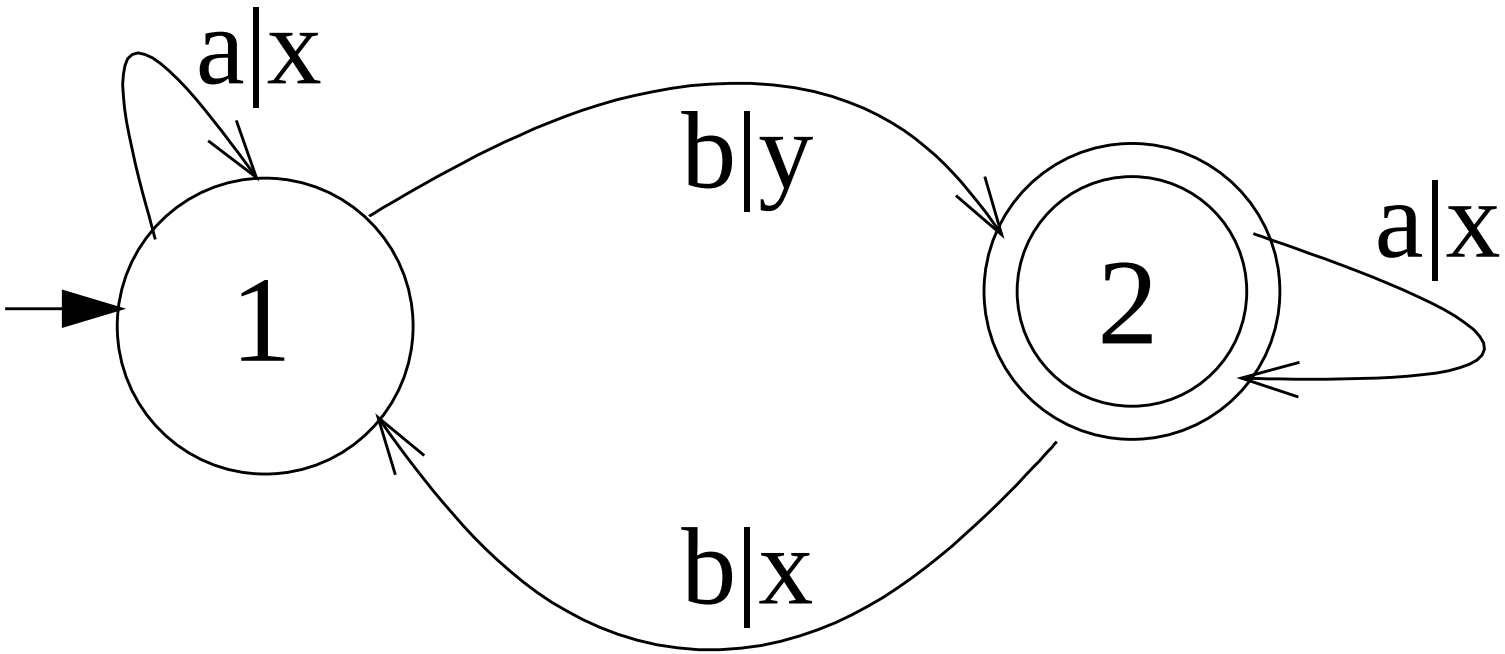
IPS - A Simple Example - Transducer

Translate input streams over $\{a, b\}$ to sequences over $\{x, y\}$

- IPS - $1, 2$
- Inputs - a, b
- Outputs - x, y

abab

xyxx



IPS - The Simplest Case - Finite Time Lag

- The simplest construction of IPS is based on concentrating on the very recent (finite) past
  e.g. definite memory machines, finite memory machines, Markov models
- Example:
  Only the last 5 input symbols matter when reasoning about what symbol comes next
  ... 1 2 1 1 2 3 2 1 2 1 2 4 3 2 1 1
  ... 2 1 1 1 1 3 4 4 4 4 1 4 3 2 1 1
  ... 3 3 2 2 1 2 1 2 2 1 3 4 3 2 1 1
  All three sequences belong to the same IPS "43211"

Probabilistic framework - Markov model (MM)

- *What comes next?* **From the same IPS**

  ... 1 2 1 1 2 3 2 1 2 1 2 4 3 2 1 1 → ?

  ... 2 1 1 1 1 3 4 4 4 4 1 4 3 2 1 1 → ?

  ... 3 3 2 2 1 2 1 2 2 1 3 4 3 2 1 1 → ?

- Finite context-conditional next-symbol distributions

  $P(s \mid 11111)$

  $P(s \mid 11112)$

  $P(s \mid 11113)$

  ...

  $P(s \mid 11121)$

  ...

  $P(s \mid 43211)$

  ...

  $P(s \mid 44444)$ where $s \in \{1, 2, 3, 4\}$

Markov model

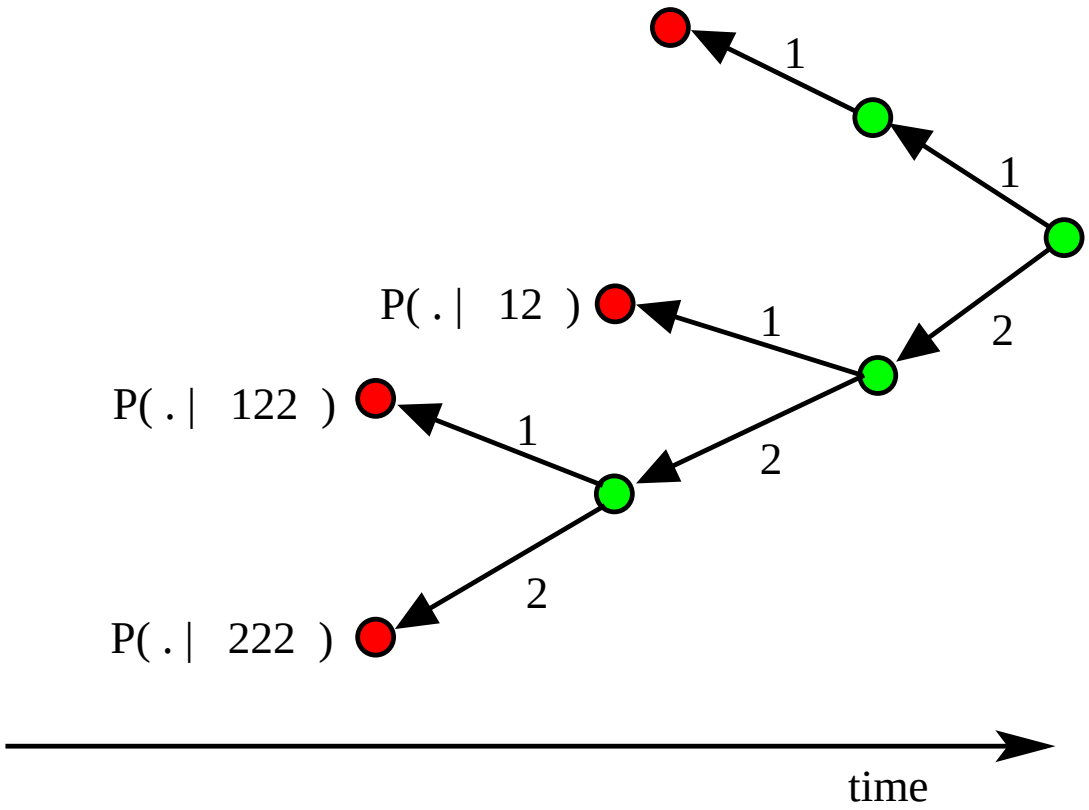
- MM are intuitive and simple, but ...
- Number of IPS (prediction contexts) grows exponentially fast with memory length
- Large alphabets and long memories are infeasible:
  - Computationally demanding and
  - Very long sequences are needed to obtain sound statistical estimates of free parameters

Variable memory length MM (VLMM)

- Sophisticated implementation of potentially high-order MM
- Takes advantage of subsequence structure in data
- Saves resources by making memory depth context dependent
- Use deep memory only when it is needed
- Natural representation of IPS in form of Prediction Suffix Trees
- Closely linked to the idea of universal simulation of information sources.

Prediction suffix tree (PST)

- IFS are organized on nodes of PST
- Traverse the tree from root towards leaves in reversed order
- Complex and potentially time-consuming training



Each node i
has associated next−symbol
probability distribution
P(. | i)

Dynamical processes

- Imagine a "box" inside which a dynamic process (i.e. things change in time) takes place. At time t, the "state of the system" is $\boldsymbol{x}(t) \in \mathcal{X}$.
- The dynamics can be autonomous (e.g. "happens by itself"), or non-autonomous (e.g. driven by external input).

(Chi - Greek letter pronounciation)

$$\boldsymbol{x}(t) = f(\boldsymbol{x}(t-1)), \quad \boldsymbol{x}(t) = f(\boldsymbol{u}(t), \boldsymbol{x}(t-1))$$

- The dynamics can be continuous time (e.g. time is taken as a continuous entity - dynamics can be described e.g. by differential equations), or discrete time (e.g. changes happen in discrete time steps- can be described e.g. by iterative maps).

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t)), \quad \frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{u}(t), \boldsymbol{x}(t))$$

Dynamical processes - cont'd

- The dynamics can be stationary (e.g. the law governing the dynamics is fixed and not allowed to change), or non-stationary (e.g. the manner in which things change is itself changing in time).

$$\boldsymbol{x}(t) = f_t(\boldsymbol{x}(t-1)), \quad \boldsymbol{x}(t) = f_t(\boldsymbol{u}(t), \boldsymbol{x}(t-1))$$

$$\frac{d\boldsymbol{x}(t)}{dt} = f_t(\boldsymbol{x}(t)), \quad \frac{d\boldsymbol{x}(t)}{dt} = f_t(\boldsymbol{u}(t), \boldsymbol{x}(t))$$

- The dynamics can be deterministic (as above), or stochastic (e.g. corrupted by some form of dynamic noise). One now must work with full distributions over possible "states" of the system!

$$p(\boldsymbol{x}(t) \mid \boldsymbol{x}(t-1)), \quad p(\boldsymbol{x}(t) \mid \boldsymbol{u}(t), \boldsymbol{x}(t-1))$$

Observing dynamical processes

- We can have a "telescope" with which to observe the "box" inside which a dynamic process is happening, e.g. we can have access to some coordinates of the dynamics, or a "reasonable" function on them. We will call such observed entities observations.

$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t)), \quad \boldsymbol{y}(t) = h(\boldsymbol{u}(t), \boldsymbol{x}(t))$$

- Reading out of the observations can be corrupted by an observational noise. In this case we are uncertain about the value of observations and can only have distributions over possible observations.

$$p(\boldsymbol{y}(t) \mid \boldsymbol{x}(t)), \quad p(\boldsymbol{y}(t) \mid \boldsymbol{u}(t), \boldsymbol{x}(t))$$

The Notion of State Space Model - Part 1

We impose that the dynamic process we are observing is governed by a dynamic law prescribing how the state of the system evolves in time. The state captures all that we need to know about the past in order to describe future evolution of the system.

$$\boldsymbol{x}(t) = f(\boldsymbol{x}(t-1)), \quad \boldsymbol{x}(t) = f(\boldsymbol{u}(t), \boldsymbol{x}(t-1))$$

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t)), \quad \frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{u}(t), \boldsymbol{x}(t))$$

$$p(\boldsymbol{x}(t) \mid \boldsymbol{x}(t-1)), \quad p(\boldsymbol{x}(t) \mid \boldsymbol{u}(t), \boldsymbol{x}(t-1))$$

The Notion of State Space Model - Part 2

- We can have access to the dynamics through some function of the states producing observations.

$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t)), \quad \boldsymbol{y}(t) = h(\boldsymbol{u}(t), \boldsymbol{x}(t))$$

$$p(\boldsymbol{y}(t) \mid \boldsymbol{x}(t)), \quad p(\boldsymbol{y}(t) \mid \boldsymbol{u}(t), \boldsymbol{x}(t))$$

- It can be possible to "recover" the states (in some sense- e.g. up to topological conjugacy) from the observations - "observable states", or not - "unobservable states".