

# Discrete-time, discrete-valued observable operator models: a tutorial

Herbert Jaeger  
International University Bremen  
[h.jaeger@iu-bremen.de](mailto:h.jaeger@iu-bremen.de)  
23 July 2003 (draft)

Presented by: Michael Cole  
4 November 2003



If you can't say it in words, don't  
try to whistle it in mathematics.

- C. J. van Rijsbergen

# What's this about?

- Observable operator models (OOMs) are a new class of stochastic models
- OOMs are more general than Hidden Markov Models (HMMs)
- A flavor of OOM can predict future states and is computationally cheap
- There are many similarities between this type of OOM for learning and PSR

# Overview

- What are OOMs?
- Learning with OOMs.
- Input – Output OOMs.
- IO-OOMs vs. PSRs
- How do OOMs relate to learning representations?

# OOM – Key insight

- “ The *change* of predictive knowledge that we have about a stochastic system is *linear* phenomenon.” p. 75
  - ⇒ Leads to the idea of observable operators
    - ( But is valid for every stochastic process )

# OOM (whistled)

**Definition 1** A  $m$ -dimensional OOM is a triple  $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \mathcal{O}}, w_0)$ , where  $w_0 \in \mathbb{R}^m$  and  $\tau_a : \mathbb{R}^m \mapsto \mathbb{R}^m$  are linear operators, satisfying

1.  $\mathbf{1}w_0 = \mathbf{1}$ ,
2.  $\mu = \sum_{a \in \mathcal{O}} \tau_a$  has column sums equal to 1,
3. for all sequences  $a_{i_0}, \dots, a_{i_k}$  it holds that  $\mathbf{1}\tau_{a_{i_k}} \cdots \tau_{a_{i_0}} w_0 \geq 0$ .

# Stochastic / Pseudo-stochastic

- Stochastic time-series result from true stochastic processes
- Pseudo-stochastic time-series are snapshots of systems in a chaotic regime
  - e.g weather measurements, financial market data, speech

# Models as probability distributions

- Stochastic process  $\Rightarrow$  Observation sequence expressed as the probability of each observation.
- A system model can be an exhaustive collection of probabilities of every possible observation sequence. (and so every possible realization of the model)

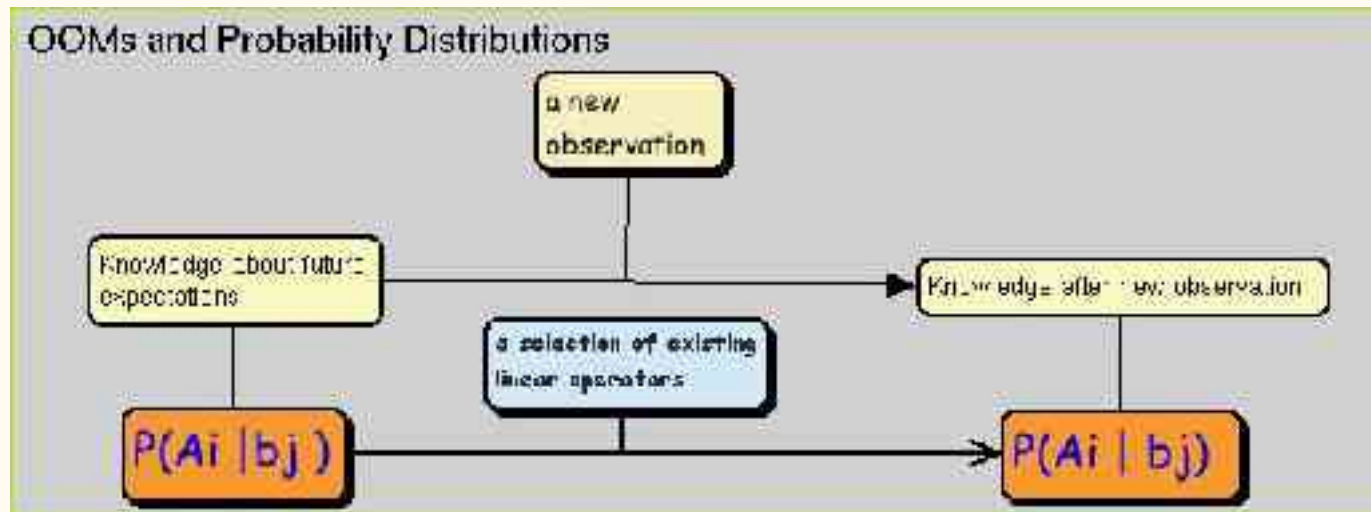
# OOMs as models

- OOMs are descriptions of collections of probability distributions
- OOMs are generative
  - they provide a mechanism to produce possible observation sequences of the system they model.

# OOMs and probability distributions

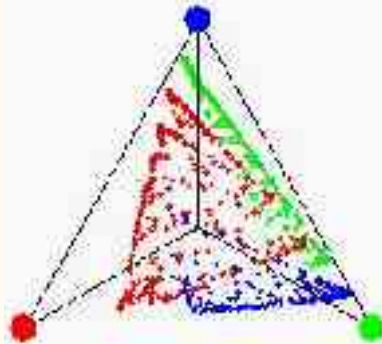
- OOMs describe the change of our knowledge about the future of a stochastic system using linear operators that have been selected in view of observations so far.
- Our knowledge about the future is a conditional probability distribution given the observed past of the system.

# Operators generate the change in knowledge about the future of the system



# HMMs and OOMs

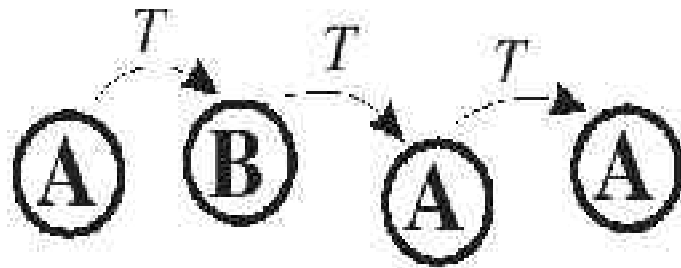
- HMMs view stochastic systems as trajectories in a state space. Observations are locations in that state space.



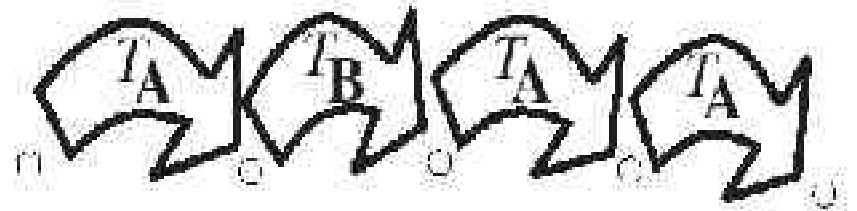
# OOM view

- OOMs see trajectories as a sequence of (linear) operations. Each observation corresponds to the sequence of operations on the previous observation. (So OOMs are about the process)

# Locations v. Transformations



A and B are observations along a system trajectory in state space



A system trajectory as a series of operators that result in observables

# OOMs are more general than HMMs

- OOMs can express every Linear Decision Process (LDP)
- LDPs can express every HMM
- But HMMs cannot express every LDP

# OOM flavors

- Output only
  - OOM is a passive model only concerned with the system output
- Input-Output (IO-OOM)
  - OOM can model cases where the system is modified in response to output, e.g. a robot learning an environment

# Infinite and finite operators

- Obviously, if one has an infinite number of operators any stochastic system can be modeled.
- But OOMs of finite dimension can do quite a bit of work. In fact, the class of processes is larger than those that can be modeled using HMMs of any finite number of states.

# Finite OOMs are computationally friendly

- The defining linear operators can be expressed in matrices, so all the tools of linear algebra can be brought to bear.
- The equivalence of OOMs as models of a particular stochastic process can be determined. So one can explore equivalence classes of OOMs.

# Learning with OOMs

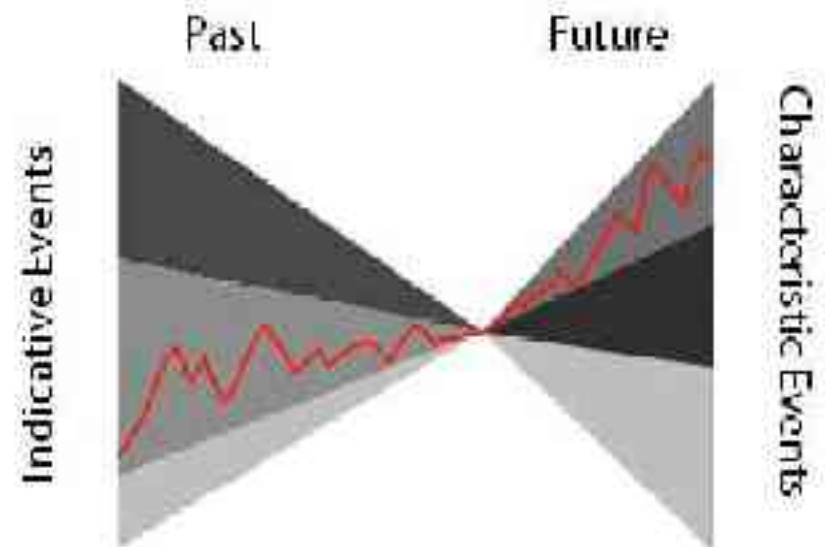
- Learning in OOMs *means* estimating the linear operators from a sequence of observations
- Key concepts:
  - Indicative events
  - Characteristic events

## Indicative and Characteristic Events

Clustering the Past and the Future into indicative and characteristic events ..

Indicative and characteristic events are ..

- **classes** of finite Past and Future
- (coarse) **features** of process trajectories
- defined via **partitionings** of the sequence space



# Indicative and Characteristic Events

For a sequence of length  $k$ ,  $\mathcal{O}$  is a partition of the sets of possible sequences of length  $k$ .

$$\mathcal{O}^k = A_1 \cup \dots \cup A_m$$

For some sequences  $b$ , there is an  $m \times m$  matrix with elements:

$$(P[A_i | \bar{b}_j])_{i,j}$$

(where  $P[A_i | \bar{b}_j]$  denotes  $\sum_{\bar{a} \in A_i} P[\bar{a} | \bar{b}_j]$ )

If the matrix is nonsingular then the sets of sequences are characteristic events.

Every finite OOM has characteristic events.

# Characteristic events are key to OOM learning

**Proposition 8** *In an interpretable OOM  $\mathcal{A}(A_1, \dots, A_m)$  it holds that*

1.  $w_0 = (P[A_1], \dots, P[A_m])$ ,
2.  $\tau_{\bar{b}} w_0 = (P[\bar{b}A_1], \dots, P[\bar{b}A_m])$ .

This allows one to estimate the state vectors from the data using frequency counts. The estimated state vectors allow construction of the operators using linear algebra.

This proposition is the key to the computational advantage associated with OOMs.

# Learning with OOMs: Three facts

- There are no clues about choosing indicative and characteristic events.
- Selection of indicative and characteristic events in an infinite sequence of training data can be arbitrary and one still gets the correct model operator estimates.
- But, with *finite* training data, selection of indicative and characteristic events is critical.

# Learning with finite sequences

- Two basic problems
  - What is the process dimensionality of the system?
  - How does one select indicative and characteristic events?

Only heuristic answers to these questions

# OOM basic learning algorithm

**Step 1** Compute the  $m \times m$  matrix  $V^\# = (\#_{\text{butlast}} B_j A_i)$ .

**Step 2** Compute, for every  $a \in \mathcal{O}$ , the  $m \times m$  matrix  $W_a^\# = (\# B_j a A_i)$ .

**Step 3** Obtain  $\tilde{\tau}_a = W_a^\# (V^\#)^{-1}$ .

Where

$\mathcal{O}$  is the observation set

$B$  are the indicative events (history)

$A$  are the characteristic events

$a$  is a particular observation

$\tau$  is the operator for  $a$

# A toy example of the procedure

Let r=red; b=blue

S= rbbrrrrrbrrbbbrbbbbb

N=20

Select the number of processes: 2

Select indicative and characteristic events:

Characteristic events: r, b

Indicative events: r, b

# Toy example (cont)

- Estimate the original probabilities (just count up the occurrences of r and b) for the invariant vector  $\omega$ :

$$r = 8/20$$

$$b = 12/20$$

$$\text{So } \omega = (8/20, 12/20)$$

# Toy example (cont)

- Next calculate the  $V\#$  and  $W\#$  matrices (looking for the subsequences in  $S$ )

$$V\# = \begin{pmatrix} \#_{\text{butlast}} aa & \#_{\text{butlast}} ba \\ \#_{\text{butlast}} ab & \#_{\text{butlast}} bb \end{pmatrix} = \begin{pmatrix} 4 & 3 \\ 4 & 7 \end{pmatrix},$$

$$W_a\# = \begin{pmatrix} \#_{aaa} & \#_{baa} \\ \#_{cab} & \#_{bab} \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix},$$

$$W_b\# = \begin{pmatrix} \#_{aba} & \#_{bba} \\ \#_{abb} & \#_{bbb} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}.$$

# Toy example (cont)

We can now estimate the observable operators (using proposition 8)

$$\begin{aligned}\tilde{\tau}_a &= W_a^\# (V^\#)^{-1} = \begin{pmatrix} 3/8 & 1/8 \\ 5/8 & -1/8 \end{pmatrix}, \\ \tilde{\tau}_b &= W_b^\# (V^\#)^{-1} = \begin{pmatrix} -1/16 & 5/16 \\ 1/16 & 11/16 \end{pmatrix}.\end{aligned}$$

And the OOM is (notice the updated invariant vector):

$$\tilde{\mathcal{A}} = (\mathbb{R}^2, \begin{pmatrix} 3/8 & 1/8 \\ 5/8 & -1/8 \end{pmatrix}, \begin{pmatrix} -1/16 & 5/16 \\ 1/16 & 11/16 \end{pmatrix}, (9/20, 11/20)).$$

# Cheap

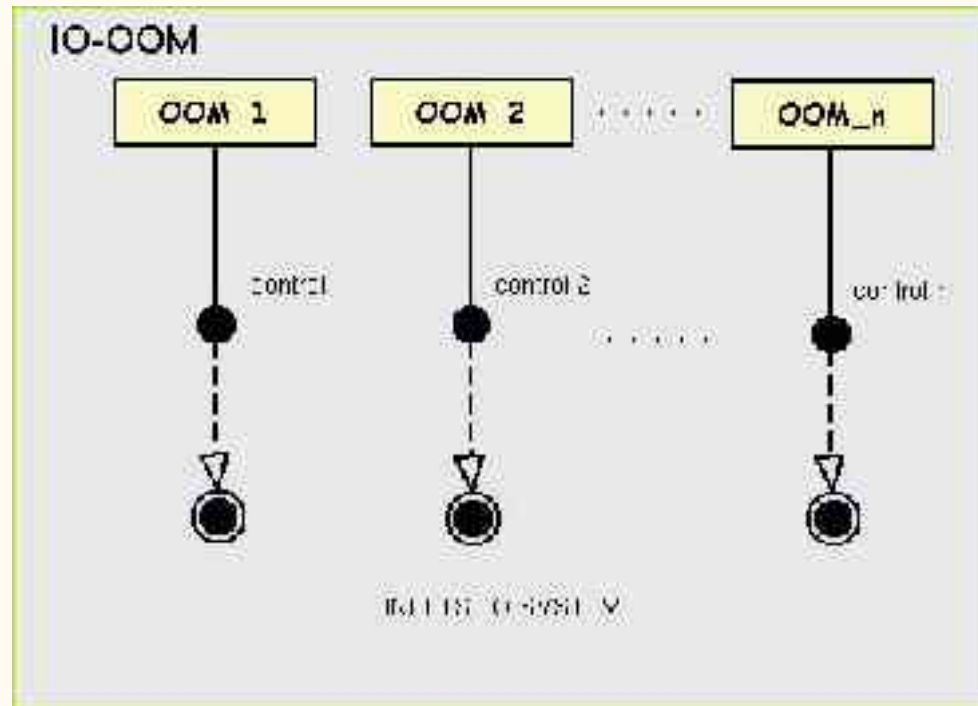
- Compared to HMM parameter estimation OOM learning is cheap.
  - The counting for the  $V$  and  $W$  matrices is handled in a single sweep of the observation sequence (using a window  $k+l+1$ ).
  - Multiplying or inverting  $m \times m$  matrices costs  $O(m^3/p)$

Where  $p$  is the degree of parallelization  
 $m$  is the dimension of the OOM  
 $O$  is the observation set

# Input-Output OOMs

- The IO-OOMs represent cases where actions can be injected into the system to modify it.
  - Think of a robot interacting with its environment
- Inputs are the modifications, outputs are the observations of the system

# IO-OOM: a collection of OOMs



The operators,  $\tau$ , for each OOM are the means of executing the controls,  $r$ . So the probability of observing a sequence after applying a control can be expressed in terms of the operators for each OOM.

# IO-OOM (whistled)

**Definition 3** An  $m$ -dimensional IO-OOM is a triple  $(\mathbb{R}^m, ((\tau_a^r)_{a \in \mathcal{O}})_{r \in \mathcal{U}}, w_0)$ , where  $w_0 \in \mathbb{R}^m$  and all  $\tau_a^r : \mathbb{R}^m \mapsto \mathbb{R}^m$  are linear operators, satisfying

1.  $\mathbf{1}w_0 = \mathbf{1}$ ,
2. for every  $r \in \mathcal{U}$ , the matrix  $\sum_{a \in \mathcal{O}} \tau_a^r$  has column sums equal to  $\mathbf{1}$ ,
3. for all sequences  $(r_0, a_0) \dots (r_k, a_k)$  it holds that  $\mathbf{1}_{\tau_{a_k}^{r_k} \dots \tau_{a_0}^{r_0}} w_0 \geq 0$ .

For every  $r \in \mathcal{U}$ , the ordinary OOM  $(\mathbb{R}^m, (\tau_a^r)_{a \in \mathcal{O}}, w_0)$  is called the  $r$ -constituent of the IO-OOM.

An IO-OOM defines measures  $\mu_{r_0 \dots r_n}$  via

$$\mu_{r_0 \dots r_n} [a_0 \dots a_n] = \mathbf{1}_{\tau_{a_n}^{r_n} \dots \tau_{a_0}^{r_0}} w_0, \quad (33)$$

# IO-OOM learning algorithm

- The basic idea is to estimate an OOM that models a controlled stochastic process (CSP)
  - A CSP is a controlled object with a control strategy (a sequence of actions)
- Then use the OOM to estimate the probabilities of characteristic sequences given a current control action, a history of actions and observations

# Predictive State Representations (PSRs)

- The basic idea is that a state is defined by the predictions of a series of tests
- Tests are sequences of inputs to a system and output observations
- A prediction vector consists of the probabilities of an observation sequence given a history and each sequence of actions (a test)

# PSR similarities with IO-OOMs

- A characterization frame for an IO-OOM is a sequence of control inputs and a set of characteristic events.

This is very similar to PSR's notion of a prediction vector because it relates the probability of a future observed sequence to the history of the system and some set of actions (operator transformations)

# PSR / IO-OOM concepts

- PSR:

A test  $\Rightarrow$  Does some particular observation sequence occur when certain actions are taken?

- IO-OOM

A test  $\Rightarrow$  For a characterization frame, pick a characteristic event and find out if any observation sequence in the characteristic event occurs when the action is taken.

# PSR / IO-OOM concepts

- PSR
  - projection function for a test
- IO-OOM
  - the linear function of operators for an IO-OOM test

# PSR / IO-OOM differences

- PSRs can accommodate arbitrary projection functions
  - Nonlinear functions  $\Rightarrow$  more compact models
- IO-OOM only uses linear projection functions (of operators)
  - Computationally friendly

# PSR / IO-OOM differences

- PSRs depend upon POMDPs
- IO-OOM representations are more general

# PSR and IO-OOM Learning

- The extension tests of PSR is mirrored in the  $W$  matrices of the IO-OOM learning algorithm.
- While PSR learning is online and IO-OOM is batch, it is possible to build an online method for OOMs.

# Online OOM learning issues

- Many modes of convergence exist, so when a small error level is reached, the process becomes quite slow.
  - Cannot rule out possibility of being stuck in a local minimum
- Speeding up learning rate leads to instability
- The model parameters are updated online, but the state vector depends on the current model estimate. So analysis of convergence is difficult

# Online IO-OOM?

- Not clear how online learning for OOMs can be adapted to IO-OOMs

# PSR and IO-OOM learning

- PSR learning estimates projection vectors of extension tests.
  - How can one construct the projection vectors for tests not in the extension set without using the POMDP?
- IO-OOM give estimates of the operators, and they can be used to determine the probabilities of arbitrary IO-OOM tests

# PSR and IO-OOM learning (cont)

- PSR learning algorithm underexploits data
  - Updates occur only when extension test is applied
- IO-OOM fully exploits the training data
  - The entire observation set is used to estimate the state vector and operators

# PSR and IO-OOM learning (cont)

- IO-OOM has a heuristic technique to determine the appropriate dimension.
- Open question whether PSR or IO-OOM learning is guaranteed to converge

# PSRs and IO-OOMs: Advantages

- PSR and possible adaptations of online IO-OOMs have issues but:
  - They are the only known computationally cheap online algorithms
  - Techniques exist to deal with slow convergence
  - Fast convergence (high learning rates) might be acceptable in many applications

# What about learning *representation*?

- OOMs are models of the transformations of a system that yield one observation after another.
- In effect, OOM represents observables as transformations.

# Observations as transformations

- The operators are divorced from any notion of a system's hidden states. The operators themselves are purely statistical, they do not refer to things.
- This is an interesting conceptual challenge. OOMs see the world as pure stochastic process.

# World as statistics?

- If one learns only the statistics of transformations of the world, does this compromise the ability to build interesting representations?

# Extension and encapsulation

- The actual process behind one operator might be the result of some physics, the next might be the result of psychology, so the operators may be pure statistics but the processes they mark could allow for representation of complex systems

# Tao of OOM?

- This focus on transformation rather than the observations themselves is reminiscent of Alfred North Whitehead's (of *Principia Mathematica* fame) view that reality *is* process [*Process and Reality*, 1929].  
“The actualities of the Universe are processes ..., each process is an individual fact. The whole Universe is the advancing assemblage of these processes.”

# references

Whitehead, A.N. (1929) *Process and Reality* New York: Macmillan

Oberstein, T. (2002) Recent advances in Observable Operator Models.

[http://www.ais.fraunhofer.de/INDY/tobias/oom\\_advcs\\_slides.pdf](http://www.ais.fraunhofer.de/INDY/tobias/oom_advcs_slides.pdf)

Acquired 29 October, 2003