



JACOBS
UNIVERSITY

Michael Thon

Input-Output OOMs

Technical Report No. 16

May 2008

School of Engineering and Science

Input-Output OOMs

Extending observable operator models and a learning algorithm to the case of controlled stochastic processes

Michael Thon

*School of Engineering and Science
Jacobs University Bremen gGmbH
Campus Ring 12
28759 Bremen
Germany*

*E-Mail: m.thon@jacobs-university.de
<http://www.jacobs-university.de>*

Summary

Input-output OOMs (IO-OOMs) are an extension of the basic OOM theory to deal with controlled stochastic processes. They were first presented in a comprehensive OOM tutorial text [3], where an early version of the OOM learning algorithm was partially transferred. Since then, OOM research has focussed on basic OOMs, in particular on developing statistically efficient learning algorithms, and research on IO-OOMs has been dormant. In this report it is shown how the main theorems for OOMs can be transferred to the case of IO-OOMs and how one of the current OOM learning algorithms can be adapted to learning IO-OOMs, yielding the first complete IO-OOM learning algorithm.

Contents

1	Introduction	1
1.1	Controlled stochastic processes and notation	1
2	The basics of IO-OOMs	3
2.1	Equivalence of IO-OOMs	4
3	Learning IO-OOMs from data	8
3.1	The learning equation for IO-OOMs	8
3.2	A learning algorithm for IO-OOMs	9
3.2.1	Estimating “probabilities” from the training sequence	10
3.2.2	Choosing characteristic and indicative sequences	11
3.2.3	Finding appropriate characterizer C and indicator Q	12
4	Empirical results	13
5	Conclusion	15
	References	16

1 Introduction

In this report it will be shown that much of the theory of OOMs can, in fact, be transferred to the case of Input-Output (IO)-OOMs – models very similar to OOMs but for discrete controlled stochastic processes, i.e. discrete-valued stochastic processes that additionally depend on discrete-valued control input given at every time step. We prove versions of the two main theorems, namely the equivalence theorem and the minimality theorem, for IO-OOMs and use these to derive the learning equation for IO-OOMs in much the same way as it was possible for OOMs.

Using these theoretical results, we then show that we can adapt one of the current statistical efficiency optimized OOM learning algorithms based on the error controlling principle ([11],[10]) to the case of IO-OOMs, yielding the first complete IO-OOM learning algorithm. We test the described IO-OOM learning algorithm on a standard benchmark range and show its performance in comparison to learning algorithms for the related *predictive state representations* (PSRs), which are a similar extension of OOMs to the class of controlled stochastic processes but have been redeveloped independently and use a different notation.

Finally, we note several possible improvements to the described IO-OOM learning algorithm in its basic form, which are avenues for further research.

1.1 Controlled stochastic processes and notation

We begin by clarifying what is meant by a *controlled (stochastic) process* (CSP) – the central object of study throughout this report. In this report we will be dealing only with discrete-time and discrete-valued stochastic processes and controlled stochastic processes, so this is from now on assumed. Also, we need to fix some notation that we will be using throughout this report.

Consider a discrete-time stochastic process $X := (\Omega, \mathcal{F}, P, (X_t)_{t \in \mathbb{N}})$ with values in a finite alphabet S . Recall that such a process is defined by the probabilities of all initial sequences, i.e. the probabilities of all words over the alphabet S . This is the same as specifying the joint probability distributions $P(X_1, \dots, X_n)$ for all possible n .

We will denote by S^* the set of words over the alphabet S and by S^n the set of words of length n . Furthermore, we will denote symbols (words of length one) by normal variables such as $s \in S$ and words by variables with a bar such as $\bar{s} \in S^*$. For words \bar{a}, \bar{b} we will denote by $\bar{a}\bar{b}$ their concatenation. Finally, we reserve the special symbol ε for the empty word in S^* . For probabilities of the form $P(X_1 = s_1, \dots, X_n = s_n)$, we will write the shorthand $P(s_1 \dots s_n)$ or simply $P(\bar{s})$ and also use this notation in conditional probabilities.

Now assume that we can write the alphabet set S as $A \times O$. We will regard the values in A as input actions or controls and the values in O as observed output. Under this interpretation we may regard the generating mechanism of the stochastic process as a coupling of two systems, namely a control system that generates input (depending only on the previous history) and a controlled system that generates output (depending on the previous history and the current input). Certainly, the stochastic process X describes the result of the interaction of these two systems, but we will show that we can also consider these separately.

Let $X_t^{(a)}$ and $X_t^{(o)}$ be the components of X_t in A and O , respectively. Then $X_t^{(a)}$ and $X_t^{(o)}$ are also random variables. We call the values of $X_t^{(a)} \in A$ the control, action or input at time t and the values of $X_t^{(o)} \in O$ the observation or output at time t . We will use variables a or \bar{a} to denote symbols from A or words from A^* , respectively, and similarly $o \in O$ and $\bar{o} \in O^*$. For sequences $\bar{s} = (a_1, o_1) \dots (a_n, o_n) \in (A \times O)^n$, we always omit the brackets and write $\bar{s} = a_1 o_1 \dots a_n o_n$. Note, however, that this is still a sequence of length n .

The key observation made in [1] is that we can calculate the probabilities of all finite sequences $a_1 o_1 \dots a_n o_n \in (A \times O)^n$ by

$$P(a_1 o_1 \dots a_n o_n) = \prod_{i=1}^n P(a_i | a_1 o_1 \dots a_{i-1} o_{i-1}) \cdot \prod_{i=1}^n P(o_i | a_1 o_1 \dots a_{i-1} o_{i-1} a_i). \quad (1)$$

We will give the two factors in the above equation special abbreviations:

$$\begin{aligned} P_i(a_1 o_1 \dots a_n o_n) &:= \prod_{i=1}^n P(a_i | a_1 o_1 \dots a_{i-1} o_{i-1}) \\ P_o(a_1 o_1 \dots a_n o_n) &:= \prod_{i=1}^n P(o_i | a_1 o_1 \dots a_{i-1} o_{i-1} a_i) = \frac{P(a_1 o_1 \dots a_n o_n)}{P_i(a_1 o_1 \dots a_n o_n)}. \end{aligned} \quad (2)$$

Note that the function P_i captures the role of the control system in generating the stochastic process, since it collects the conditional probabilities of the next input action given the process history, while the function P_o captures the role of the controlled system, since it collects the conditional probabilities of the next output observation given the process history and the current input.

When encountering conditional probabilities of the type $P(s | N)$, where s is a symbol from the respective symbol set S , and N is an event with zero probability, we may set $P(s | N)$ to any value in $[0, 1]$ such that the condition $\sum_{s \in S} P(s | N) = 1$ is satisfied. This way, the functions P_i and P_o are well-defined, and equation (1) still holds.

It is then easy to check that $P_i : (A \times O)^* \rightarrow [0, 1]$ satisfies the conditions (i) $P_i(\varepsilon) = 1$, (ii) $\forall \bar{h} \in (A \times O)^*, o \in O : P_i(\bar{h}o) = \sum_{a \in A} P_i(\bar{h}ao)$ and (iii) $\forall \bar{h} \in (A \times O)^*, a \in A, o, o' \in O : P_i(\bar{h}ao) = P_i(\bar{h}ao')$, while the function $P_o : (A \times O)^* \rightarrow [0, 1]$ satisfies the very similar conditions (i) $P_o(\varepsilon) = 1$ and (ii) $\forall \bar{h} \in (A \times O)^*, a \in A : P_o(\bar{h}a) = \sum_{o \in O} P_o(\bar{h}ao)$. This motivates the following definition (cf. [8]):

Definition 1. A *controlled (stochastic) process* (CSP) with observations from the finite set O and input from the finite set A is determined by a function $P_o : (A \times O)^* \rightarrow [0, 1]$, such that (i) $P_o(\varepsilon) = 1$ and (ii) $\forall \bar{h} \in (A \times O)^*, a \in A : P_o(\bar{h}) = \sum_{o \in O} P_o(\bar{h}ao)$.

An *input policy* for a CSP with observations from the finite set O and input from the finite set A is given by a function $P_i : (A \times O)^* \rightarrow [0, 1]$, such that (i) $P_i(\varepsilon) = 1$, (ii) $\forall \bar{h} \in (A \times O)^*, o \in O : P_i(\bar{h}) = \sum_{a \in A} P_i(\bar{h}ao)$ and (iii) $\forall \bar{h} \in (A \times O)^*, a \in A, o, o' \in O : P_i(\bar{h}ao) = P_i(\bar{h}ao')$.

We call an input policy P_i *complete* if $P_i > 0$, i.e. if $\forall \bar{h} \in (A \times O)^* : P_i(\bar{h}) > 0$.

Again, it is easy to check that a CSP P_o together with an input policy P_i define a stochastic process X via the above equations (1) and (2). Also, the CSP P_o can be uniquely recovered from the generated process X via the equation (2) if and only if the input policy P_i is complete.

Finally, we will use the notation $P_o(\bar{s} | \bar{h}) := P_o(\bar{h}\bar{s})/P_o(\bar{h})$ with $\bar{h}, \bar{s} \in (A \times O)^*$ for CSPs, and note that $P_o(ao | \bar{h}) = P(o | \bar{h}a)$ whenever these are defined.

2 The basics of IO-OOMs

The definition of IO-OOMs (see [3]) is very similar to the definition of OOMs:

Definition 2. A d -dimensional IO-OOM \mathcal{M} is a structure

$$\mathcal{M} = ((\tau_{a,o})_{a \in A, o \in O}, w_0), \quad \tau_{a,o} \in \text{Mat}(d \times d, \mathbb{R}), w_0 \in \mathbb{R}, \quad (3)$$

subject to the following three conditions that are adapted from basic OOMs:

$$\begin{aligned} \text{(i)} \quad & \mathbb{1}w_0 = 1, \\ \text{(ii)} \quad & \forall a \in A : \mathbb{1}\tau_a = \mathbb{1}, \quad \text{where } \tau_a = \sum_{o \in O} \tau_{a,o} \\ \text{(iii)} \quad & \forall a_1 o_1 \dots a_n o_n \in (A \times O)^* : \mathbb{1}\tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0 \geq 0, \end{aligned} \quad (4)$$

where $\mathbb{1}$ denotes a row-vector of ones of the necessary dimension (here d).

Note that for a fixed input a the structure $((\tau_{a,o})_{o \in O}, w_0)$ is just an ordinary OOM, so an IO-OOM is, in fact, a set of *constituent* OOMs, one for each possible input a , where the given input switches between these OOMs. Note, however, that these constituent OOMs all share a common state space.

Proposition 1 (Fundamental equation of IO-OOMs). *By setting*

$$\forall a_1 o_1 \dots a_n o_n \in (A \times O)^* : P_o(a_1 o_1 \dots a_n o_n) = \mathbb{1}\tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0, \quad (5)$$

the IO-OOM \mathcal{M} determines a CSP.

Proof. We have $P_o(\varepsilon) = \mathbb{1}w_o = 1$ by condition (i). By condition (ii) we get that for all $a_1o_1 \dots a_no_n \in (A \times O)^*$, and $a \in A$: $P_o(a_1o_1 \dots a_no_n) = \mathbb{1}\tau_{a_n,o_n} \dots \tau_{a_1,o_1}w_o = \sum_{o \in O} \mathbb{1}\tau_{a,o}\tau_{a_n,o_n} \dots \tau_{a_1,o_1}w_o = \sum_{o \in O} P_o(a_1o_1 \dots a_no_nao)$. And finally, condition (iii) assures that the range of the function P_o is indeed $[0, 1]$ (the values of P_o cannot exceed one, since they sum to values less than or equal to one by applying the previous statement recursively). So by the definition 1 this determines a CSP. \square

It is now easy to see how an IO-OOM \mathcal{M} together with an input policy P_i can be used as a sequence generator. It will be convenient to use the shorthand notation $\tau_{\bar{h}} = \tau_{a_n,o_n} \dots \tau_{a_1,o_1}$ for sequences $\bar{h} = a_1o_1 \dots a_no_n \in (A \times O)^*$:

Proposition 2 (IO-OOMs as sequence generators). *Given an IO-OOM \mathcal{M} and an input policy P_i , the following procedure will generate samples of the stochastic process described by \mathcal{M} combined with P_i :*

1. After a previous history of $\bar{h} \in (A \times O)^t$, let the state at time t be defined as $w_t := \frac{\tau_{\bar{h}}w_o}{\mathbb{1}\tau_{\bar{h}}w_o}$.
2. Choose the next input a_{t+1} according to the input probabilities $P_i(a | \bar{h})$ for $a \in A$.
3. Choose the next output o_{t+1} according to the output probabilities $P(o | \bar{h}a_{t+1})$ for $o \in O$, which are given by $P(o | \bar{h}a_{t+1}) = P_o(a_{t+1}o | \bar{h}) = \mathbb{1}\tau_{a_{t+1},o}w_t$.
4. Update the state via the state update equation: $w_{t+1} = \frac{\tau_{a_{t+1},o_{t+1}}w_t}{\mathbb{1}\tau_{a_{t+1},o_{t+1}}w_t}$.

Proof. Obvious. \square

Unfortunately, by a result in [8], it is in general undecidable to determine whether a given structure $\mathcal{M} = (\{\tau_{a,o}\}_{a \in A, o \in O}, w_o)$ satisfies condition (iii), i.e. whether it is a valid IO-OOM. So in practice, when working with potentially invalid IO-OOMs, one will need to modify the sequence generation procedure by some heuristic, essentially as described in the Appendix J of [4] for OOMs.

2.1 Equivalence of IO-OOMs

We now show that we can transfer the two main OOM theorems – the equivalence theorem and the minimality theorem – to the case of IO-OOMs. While these were proven for OOMs in [3] and [4] by examining a special mapping from a given OOM structure to what is called a *predictor-space* or *functional* OOM, we will redo the proofs for the more general case of IO-OOMs in purely algebraic terms.

Definition 3. We will say that two IO-OOMs \mathcal{M} and \mathcal{M}' are *equivalent*, if they specify the same CSP (i.e. \mathcal{M} and \mathcal{M}' when combined with any input policy P_i always yield the same stochastic process). We will denote this by $\mathcal{M} \equiv \mathcal{M}'$.

In fact, $\mathcal{M} \equiv \mathcal{M}' \Leftrightarrow \forall \bar{s} \in (A \times O)^* : \mathbb{1}\tau_{\bar{s}}w_0 = \mathbb{1}\tau'_{\bar{s}}w'_0$, so we will use this when checking equivalence.

The notion of equivalence of IO-OOMs is an equivalence relation and partitions the IO-OOMs into equivalence classes. We therefore say that:

Definition 4. An IO-OOM \mathcal{M} is *minimal dimensional* if it has minimal dimension among all equivalent IO-OOMs.

The goal of this section is to understand when two IO-OOMs are equivalent and when a given IO-OOM is minimal dimensional. In fact, we obtain a constructive procedure that allows us to convert any IO-OOM into an equivalent minimal dimensional IO-OOM.

Definition 5. Let $\mathcal{M} = (\{\tau_{a,o}\}_{a \in A, o \in O}, w_0)$ be a d -dimensional IO-OOM. We define the following subspaces of \mathbb{R}^d :

$$\begin{aligned} S &= \text{span} \{ \tau_{\bar{s}}w_0 \mid \bar{s} \in (A \times O)^* \}, \\ K &= \text{span} \{ (\mathbb{1}\tau_{\bar{s}})^\top \mid \bar{s} \in (A \times O)^* \} \\ S_0 &= \{ w \in S \mid \forall \bar{s} \in (A \times O)^* : \mathbb{1}\tau_{\bar{s}}w = 0 \} = S \cap K^\perp, \text{ and} \\ W &= \{ w \in S \mid w \perp S_0 \} = S \cap S_0^\perp. \end{aligned}$$

Then $\mathbb{R}^d \supset S = W \perp S_0$. W is called the *effective state space* of \mathcal{M} , and $\dim W$ is called the *effective dimension* of \mathcal{M} , denoted by $\dim_{\text{eff}} \mathcal{M}$.

Proposition 3. Let \mathcal{M} be a d -dimensional IO-OOM with effective state space W defined as above and effective dimension $\dim_{\text{eff}} \mathcal{M} = k$. Then we can construct an equivalent k -dimensional IO-OOM \mathcal{M}' with effective dimension $\dim_{\text{eff}} \mathcal{M}' = k$.

Proof. We first show how to construct the IO-OOM \mathcal{M}' , and then in the second part show that indeed $\mathcal{M} \equiv \mathcal{M}'$. Finally, we show that $\dim \mathcal{M}' = \dim_{\text{eff}} \mathcal{M}'$.

Construction of \mathcal{M}' :

The first step is to construct bases of the spaces S and K . Let $S^{(n)} = \{ \tau_{\bar{s}}w_0 \mid \bar{s} \in (A \times O)^*, |\bar{s}| \leq n \}$ and $K^{(n)} = \{ (\mathbb{1}\tau_{\bar{s}})^\top \mid \bar{s} \in (A \times O)^*, |\bar{s}| \leq n \}$. By linearity of the operators τ_s for $s \in A \times O$, we know that if $\text{span } S^{(n)} = \text{span } S^{(n+1)}$, then in fact $\text{span } S^{(n)} = S$, and similarly if $\text{span } K^{(n)} = \text{span } K^{(n+1)}$, then $\text{span } K^{(n)} = K$. So the procedure to construct the desired bases is obvious: Iteratively construct the above finite sets until the dimension of their span stops increasing. Then delete linearly dependent vectors to obtain the desired bases. Once we have constructed the bases for S and K , it is not hard to compute from these also bases for the sets S_0 and W .

Let $\{w_1, \dots, w_l\}$ be the constructed basis of S . Note that since $S = W \perp S_0$, we can write each of these basis vectors $w_i \in S$ uniquely as $w_i = w_i^\perp + w_i^0$, with $w_i^\perp \in W$ and $w_i^0 \in S_0$. The vectors w_i^\perp are the orthogonal projections of the vectors w_i onto W . Therefore, $\text{span}\{w_1^\perp, \dots, w_l^\perp\} = W$, and by deleting linearly dependent vectors and renumbering we obtain a basis $\{w_1^\perp, \dots, w_k^\perp\}$ of W .

Note that $\mathbb{1}w_i^\perp \neq 0$, since otherwise $\mathbb{1}w_i = \mathbb{1}w_i^\perp + \mathbb{1}w_i^0 = 0$ with $w_i = \tau_{\bar{h}}w_0$ for some $\bar{h} \in (A \times O)^*$ (by construction of the basis vectors w_i). But then also $\mathbb{1}\tau_a w_i = 0$ for all $a \in A$, and since for all $ao \in A \times O$: $\mathbb{1}\tau_{ao}\tau_{\bar{h}}w_0 \geq 0$, this implies by induction that $\mathbb{1}\tau_{\bar{s}}w_i = 0$ for all $\bar{s} \in (A \times O)^*$. But this means that $w_i \in S_0$, and, therefore, $w_i^\perp = 0$ cannot be a basis element.

This means that we can normalize the basis to obtain $\{w'_i = w_i^\perp / \mathbb{1}w_i^\perp\}$ a basis for W , such that $\mathbb{1}w'_i = 1$ for all basis elements.

Finally, let $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^k$ be the linear map defined by $\sigma(w'_i) = e_i \in \mathbb{R}^k$ for the normalized basis elements w'_i of W , and $\sigma(W^\perp) = 0$. Let σ^{-1} be its right-inverse such that $\sigma^{-1} : \mathbb{R}^k \rightarrow W$, i.e. $\sigma\sigma^{-1} = id$ and $\sigma^{-1}\sigma$ is the orthogonal projection onto W .

Set $\mathcal{M}' = (\{\tau'_{a,o}\}_{a \in A, o \in O}, w'_0) := (\{\sigma\tau_{a,o}\sigma^{-1}\}_{a \in A, o \in O}, \sigma w_0)$.

\mathcal{M}' is a k -dimensional IO-OOM that is equivalent to \mathcal{M} :

In this part we will write $\mathbb{1}_m$ for left-multiplication by the row-vector consisting of m ones, i.e. $\mathbb{1}_m : \mathbb{R}^m \rightarrow \mathbb{R}$, to avoid confusion. Note that this is a linear operation.

First, we show that (*) $(\mathbb{1}_k\sigma)|_S = \mathbb{1}_d|_S$. So let $w \in S$. Then $w = w^\perp + w^0$, with $w^\perp \in W$, and $w^0 \in S_0$. We can easily see that $(\mathbb{1}_k\sigma)|_W = \mathbb{1}_d|_W$, since this is true on the basis elements of W . We also know that $\sigma w^0 = 0$, and that $\mathbb{1}_d w^0 = 0$. So $\mathbb{1}_k\sigma w = \mathbb{1}_k\sigma w^\perp + \mathbb{1}_k\sigma w^0 = \mathbb{1}_k\sigma w^\perp + 0 = \mathbb{1}_d w^\perp + \mathbb{1}_d w^0 = \mathbb{1}_d w$.

This implies (***) $\mathbb{1}_d\sigma^{-1} = \mathbb{1}_k\sigma\sigma^{-1} = \mathbb{1}_k$, since $\sigma^{-1} : \mathbb{R}^k \rightarrow W \subset S$, and $\sigma\sigma^{-1} = id$.

Finally, (***) $\forall w \in S$, and $\forall \bar{s} \in (A \times O)^* : \sigma^{-1}\sigma\tau_{\bar{s}}\sigma^{-1}\sigma w = \sigma^{-1}\sigma\tau_{\bar{s}}w$. To show this, consider some $w \in S$. Then $w = w^\perp + w^0$, with $w^\perp \in W$, and $w^0 \in S_0$. Note that $\tau_{\bar{s}}w^0 \in S_0$, since S_0 is closed under $\tau_{\bar{s}}$ for all $\bar{s} \in (A \times O)^*$ by definition. So $\sigma^{-1}\sigma w^0 = \sigma^{-1}\sigma\tau_{\bar{s}}w^0 = 0$. We can now easily see that $\sigma^{-1}\sigma\tau_{\bar{s}}\sigma^{-1}\sigma w = \sigma^{-1}\sigma\tau_{\bar{s}}w^\perp = \sigma^{-1}\sigma\tau_{\bar{s}}w^\perp + \sigma^{-1}\sigma w^0 = \sigma^{-1}\sigma\tau_{\bar{s}}w$, as claimed.

We can now proceed to check the IO-OOM properties (i) to (iii) from definition 2 for the structure \mathcal{M} :

- (i) $\mathbb{1}_k\sigma w_0 \stackrel{*}{=} \mathbb{1}_d w_0 = 1$, since $w_0 \in S$.
- (ii) $\forall a \in A : \mathbb{1}_k \sum_{o \in O} \sigma\tau_{a,o}\sigma^{-1} = \mathbb{1}_k\sigma(\sum_{o \in O} \tau_{a,o})\sigma^{-1} = \mathbb{1}_k\sigma\tau_a\sigma^{-1}$, by linearity of the involved maps. Note that for any $v \in \mathbb{R}^k : \sigma^{-1}v \in W \subset S$, and that S is closed under the action of $\tau_{a,o}$ for all $ao \in (A \times O)$, so $\tau_a\sigma^{-1}v \in S$. Therefore, we obtain: $\mathbb{1}_k\sigma\tau_a\sigma^{-1} \stackrel{*}{=} \mathbb{1}_d\tau_a\sigma^{-1} = \mathbb{1}_d\sigma^{-1} \stackrel{**}{=} \mathbb{1}_k$.

$$\begin{aligned}
\text{(iii)} \quad & \forall a_1 o_1 \dots a_n o_n \in (A \times O)^* : \mathbb{1}_k \sigma \tau_{a_n, o_n} \sigma^{-1} \dots \sigma \tau_{a_1, o_1} \sigma^{-1} \sigma w_0 \\
& \stackrel{**}{=} \mathbb{1}_d \sigma^{-1} \sigma \tau_{a_n, o_n} \sigma^{-1} \dots \sigma \tau_{a_1, o_1} \sigma^{-1} \sigma w_0 \\
& \stackrel{***}{=} \mathbb{1}_d \sigma^{-1} \sigma \tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0 \quad (\text{since } w_0 \in S, \text{ and by repeatedly using (***)}) \\
& \stackrel{**, *}{=} \mathbb{1}_d \tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0 \geq 0.
\end{aligned}$$

The equality in (iii) shows not only that \mathcal{M}' is an IO-OOM, but also establishes the equivalence of \mathcal{M} and \mathcal{M}' .

\mathcal{M}' has effective dimension k :

It will suffice to show that the spaces S' and S'_0 defined for \mathcal{M}' as in definition 5 are in fact $S' = \mathbb{R}^k$ and $S'_0 = \{0\}$.

For the first part, let $w_i = \tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0$ be a vector from our basis of S constructed above. Then $w'_i := \tau'_{a_n, o_n} \dots \tau'_{a_1, o_1} w'_0 \in S'$ by definition of S' , and

$$\begin{aligned}
w'_i &= \tau'_{a_n, o_n} \dots \tau'_{a_1, o_1} w'_0 = \sigma \sigma^{-1} \sigma \tau_{a_n, o_n} \sigma^{-1} \dots \sigma \tau_{a_1, o_1} \sigma^{-1} \sigma w_0 \\
&\stackrel{***}{=} \sigma \tau_{a_n, o_n} \dots \tau_{a_1, o_1} w_0 = \sigma w_i.
\end{aligned}$$

So $\mathbb{R}^k = \sigma(W) \subset \sigma(S) \subset S'$, i.e. $S' = \mathbb{R}^k$.

For the second part, consider some $w' \in S'_0 \subset \mathbb{R}^k$. Then $\forall \bar{s} \in (A \times O)^* : \mathbb{1}_k \tau'_{\bar{s}} w' = \mathbb{1}_k \tau'_{\bar{s}} \sigma(\sigma^{-1} w') = 0$, with $\sigma^{-1} w' \in S$. By the same argument as for (iii) above, we get $\forall \bar{s} \in (A \times O)^* : \mathbb{1}_d \tau_{\bar{s}}(\sigma^{-1} w') = \mathbb{1}_k \tau'_{\bar{s}} \sigma(\sigma^{-1} w') = 0$, so $\sigma^{-1} w' \in S_0$. But then $w' = \sigma \sigma^{-1} w' = 0$. This shows that $S'_0 = \{0\}$. \square

Lemma 1. *Let \mathcal{M} and \mathcal{M}' be d - and d' -dimensional equivalent IO-OOMs with full effective state spaces $W = \mathbb{R}^d$ and $W' = \mathbb{R}^{d'}$. Then $d = d'$, and there exists an isomorphism σ with $\mathbb{1}\sigma = \mathbb{1}$ such that $w'_0 = \sigma w_0$ and $\tau'_{a, o} = \sigma \tau_{a, o} \sigma^{-1}$.*

Proof. Since \mathcal{M} has full effective state space $W = \mathbb{R}^d$, we can find a basis $\{\tau_{\bar{h}_1} w_0, \dots, \tau_{\bar{h}_d} w_0\}$ of \mathbb{R}^d with suitable $\bar{h}_i \in (A \times O)^*$ and define a linear map $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ on these basis elements by setting $\sigma \tau_{\bar{h}_i} w_0 := \tau'_{\bar{h}_i} w'_0$.

Then in fact $\sigma \tau_{\bar{h}} w_0 = \tau'_{\bar{h}} w'_0$ for all $\bar{h} \in (A \times O)^*$: We have $\tau_{\bar{h}} w_0 = \sum_{i=1}^d \lambda_i \tau_{\bar{h}_i} w_0$, so $\sigma \tau_{\bar{h}} w_0 = \sum_{i=1}^d \lambda_i \tau'_{\bar{h}_i} w'_0$. Now, for all $\bar{s} \in (A \times O)^*$ we have, using the equivalence of the IO-OOMs: $\mathbb{1} \tau'_{\bar{s}} \tau'_{\bar{h}} w'_0 = \mathbb{1} \tau_{\bar{s}} \tau_{\bar{h}} w_0 = \mathbb{1} \tau_{\bar{s}} \sum_{i=1}^d \lambda_i \tau_{\bar{h}_i} w_0 = \mathbb{1} \tau'_{\bar{s}} \sum_{i=1}^d \lambda_i \tau'_{\bar{h}_i} w'_0 \Rightarrow \mathbb{1} \tau'_{\bar{s}} (\tau'_{\bar{h}} w'_0 - \sum_{i=1}^d \lambda_i \tau'_{\bar{h}_i} w'_0) = 0 \Rightarrow \tau'_{\bar{h}} w'_0 - \sum_{i=1}^d \lambda_i \tau'_{\bar{h}_i} w'_0 \in S'_0 = \{0\} \Rightarrow \tau'_{\bar{h}} w'_0 = \sum_{i=1}^d \lambda_i \tau'_{\bar{h}_i} w'_0 = \sigma \tau_{\bar{h}} w_0$.

So σ is surjective (since \mathcal{M}' also has full effective state space $W' = \mathbb{R}^{d'}$). By analogously considering the map $\sigma' : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$, which is clearly the inverse of σ , we see that σ is in fact an isomorphism, and we must have $d = d'$. Furthermore, using again that $\mathcal{M} \equiv \mathcal{M}'$, we have $\mathbb{1} \sigma \tau_{\bar{h}_i} w_0 = \mathbb{1} \tau'_{\bar{h}_i} w'_0 = \mathbb{1} \tau_{\bar{h}_i} w_0$ for all basis elements $\tau_{\bar{h}_i} w_0$, so $\mathbb{1}\sigma = \mathbb{1}$. Finally, $\sigma^{-1} \tau'_{a, o} \sigma = \tau_{a, o}$ clearly holds on basis elements, so it holds in general. \square

We now get the two main theorems of this section as simple corollaries:

Theorem 1. *An IO-OOM \mathcal{M} is minimal dimensional, iff $\dim \mathcal{M} = \dim_{\text{eff}} \mathcal{M}$, i.e. it has full effective state space. This is the case (by the definition 5) iff the two sets of vectors $\{\tau_{\bar{s}} w_0 \mid \bar{s} \in (A \times O)^*\}$ and $\{\mathbb{1}\tau_{\bar{s}}^\top \mid \bar{s} \in (A \times O)^*\}$ both span the space $\mathbb{R}^{\dim \mathcal{M}}$.*

This also implies that the construction given in the proof of proposition 3 is a procedure for reducing any given IO-OOM \mathcal{M} to an equivalent minimal dimensional IO-OOM \mathcal{M}' .

Theorem 2. *Two minimal dimensional IO-OOMs $\mathcal{M} = (\{\tau_{a,o}\}_{a \in A, o \in O}, w_0)$ and \mathcal{M}' are equivalent, iff $\dim \mathcal{M} = \dim \mathcal{M}'$ and there exists an isomorphism σ such that $\mathbb{1}\sigma = \mathbb{1}$ and $\mathcal{M}' = (\{\sigma\tau_{a,o}\sigma^{-1}\}_{a \in A, o \in O}, \sigma w_0)$.*

Proof. It remains only to show the “if” direction. But this is obvious. \square

3 Learning IO-OOMs from data

Our basic learning task is the following:

Assume we observe a single long io-sequence \bar{x} that is governed by some (unknown) minimal d -dimensional IO-OOM $\mathcal{M} = (\{\tau_{a,o}\}_{a \in A, o \in O}, w_0)$ combined with some (unknown) fixed input policy P_i . Our goal is to estimate the IO-OOM \mathcal{M} – or an equivalent IO-OOM – from the observed training sequence \bar{x} .

3.1 The learning equation for IO-OOMs

The basic tool for constructing an IO-OOM learning algorithm is the learning equation that can be derived for IO-OOMs analogously to the learning equation for OOMs as described in [11].

Let $\{\bar{c}_1, \dots, \bar{c}_c\}$ and $\{\bar{q}_1, \dots, \bar{q}_q\}$ be two sets of sequences from $(A \times O)^*$ called *characteristic* and *indicative* sequences, respectively, such that the following matrices both are of full rank d (this is possible by the above theorem 1):

$$\Pi := \begin{pmatrix} \mathbb{1}\tau_{\bar{c}_1} \\ \vdots \\ \mathbb{1}\tau_{\bar{c}_c} \end{pmatrix}, \quad \Phi := (\tau_{\bar{q}_1} w_0 \quad \cdots \quad \tau_{\bar{q}_q} w_0). \quad (6)$$

Since these matrices are both of full rank d , we can find matrices $C \in \mathbb{R}^{d \times c}$ and $Q \in \mathbb{R}^{q \times d}$, called *characterizer* and *indicator* respectively, such that $\sigma := C\Pi$ and ΦQ are both invertible $d \times d$ matrices, and such that $\mathbb{1}C\Pi = \mathbb{1}$.

Next, we define the following “probability” matrices:

$$\underline{V} = [P_o(\bar{q}_j \bar{c}_i)]_{i,j}, \quad \underline{W}_{a,o} = [P_o(\bar{q}_j a o \bar{c}_i)]_{i,j}, \quad \text{for every } (a, o) \in A \times O, \quad (7)$$

and note that $\underline{V} = \Pi\Phi$ and $\underline{W}_{a,o} = \Pi\tau_{a,o}\Phi$. We then have:

$$C\underline{W}_{a,o}Q = C\Pi\tau_{a,o}\Phi Q = \sigma\tau_{a,o}\sigma^{-1}C\Pi\Phi Q = (\sigma\tau_{a,o}\sigma^{-1})(C\underline{V}Q), \quad (8)$$

which we can rewrite as:

$$\tau'_{a,o} := \sigma\tau_{a,o}\sigma^{-1} = (C\underline{W}_{a,o}Q)(C\underline{V}Q)^{-1}. \quad (9)$$

This will give an equivalent IO-OOM $\mathcal{M}' = (\{\tau'_{a,o}\}_{a \in A, o \in O}, w'_0 = \sigma w_0)$ by the equivalence theorem 2.

Assuming that we can estimate the matrices \underline{V} and $\underline{W}_{a,o}$ from the given training sequence \bar{x} , yielding the estimated matrices $\hat{\underline{V}}$ and $\hat{\underline{W}}_{a,o}$ for every $(a, o) \in A \times O$, we obtain the following *learning equation for IO-OOMs*:

$$\hat{\tau}_{a,o} = (C\hat{\underline{W}}_{a,o}Q)(C\hat{\underline{V}}Q)^{-1}. \quad (10)$$

The starting state \hat{w}_0 can be estimated as follows: Let $v = [P_o(\bar{c}_1), \dots, P_o(\bar{c}_c)]^\top = \Pi w_0$, which must be estimated from data as above to give \hat{v} . Then $Cv = C\Pi w_0 = \sigma w_0$, so we set $\hat{w}_0 = C\hat{v}$ and then normalize the starting state to have column sum one:

$$\hat{w}_0 = \frac{C\hat{v}}{\mathbb{1}C\hat{v}}. \quad (11)$$

The structure $\hat{\mathcal{M}} = (\{\hat{\tau}_{a,o}\}_{a \in A, o \in O}, \hat{w}_0)$ is an estimate of the IO-OOM \mathcal{M}' , which is equivalent to \mathcal{M} .

3.2 A learning algorithm for IO-OOMs

To obtain a learning algorithm from the above learning equation, we need to specify the following steps:

1. How to estimate the “probabilities” of the form $P(\bar{s})$, where $\bar{s} \in (A \times O)^*$ from the training sequence \bar{x} .
2. How to choose appropriate indicative/characteristic sequences without prior knowledge of the governing IO-OOM \mathcal{M} .
3. How to find appropriate characterizer C and indicator Q .

Once these steps have been completed, we use the learning equations (10) and (11) to estimate a model \mathcal{M}' as described above. We will now deal with all of these necessary steps.

3.2.1 Estimating “probabilities” from the training sequence

Recall that we assume that a CSP P_o is given by an IO-OOM \mathcal{M} , and this CSP together with an input policy P_i generates a stochastic process $(\Omega, \mathcal{F}, P, (X_t)_{t \in \mathbb{N}})$, of which we have a sample \bar{x} . Then recall from the introduction that

$$P_o(a_1 o_1 \dots a_n o_n) = \prod_{i=1}^n P(o_i | a_1 o_1 \dots a_{i-1} o_{i-1} a_i) = \frac{P(a_1 o_1 \dots a_n o_n)}{P_i(a_1 o_1 \dots a_n o_n)}, \quad (12)$$

whenever either all of the occurring conditional probabilities are defined or at least one of the conditional probabilities is zero. This is the case if and only if the input policy P_i is complete, i.e. $P_i > 0$. Otherwise, we cannot uniquely determine the CSP P_o and must set the occurring undefined conditional probabilities of the type $P(o | N)$, where $P(N) = 0$, to any value in $[0, 1]$ such that $\sum_{o \in O} P(o | N) = 1$. This way we will obtain a new CSP P'_o that will yield the same stochastic process as P_o when combined with the incomplete input policy P_i but may yield differing results when combined with some other input policy.

Assuming that the stochastic process X is stationary and ergodic (an assumption also made when learning OOMs), we can estimate the occurring conditional probabilities by simple frequency counts. We will also want to assure that the estimated CSP \hat{P}_o is a valid CSP, i.e. satisfies the condition $\hat{P}_o(\bar{h}) = \sum_{o \in O} \hat{P}_o(\bar{h}ao)$ for all $\bar{h} \in (A \times O)^*$ and $a \in A$, so we will use the following recursive estimation procedure, noting that we must set $\hat{P}_o(\varepsilon) = 1$:

$$\hat{P}_o(\bar{h}ao) = \begin{cases} \frac{\#\bar{h}ao}{\#\bar{h}aO} \hat{P}_o(\bar{h}) & \text{if } \#\bar{h}aO > 0, \\ \frac{1}{|O|} \hat{P}_o(\bar{h}) & \text{if } \#\bar{h}aO = 0, \end{cases} \quad (13)$$

where $\#\bar{s}$ for $\bar{s} \in (A \times O)^*$ denotes the number of occurrences of the sequence \bar{s} in the training sequence \bar{x} . Here, O can be any symbol from the observation set O . These estimates are asymptotically correct by the assumption of ergodicity and stationarity of X , where correct means that we will in the limit obtain a CSP \hat{P}_o that when combined with the input policy P_i gives rise to X . Furthermore, if P_i is a complete policy, then we will asymptotically recover P_o .

Note that these estimates make no use of any knowledge about the input policy P_i . However, in many situations we may know for instance that the input policy is *blind*, i.e. that the input probability at each time step is independent of the history; we may know the input policy P_i entirely; or we may even be able to adapt the input policy to improve the IO-OOM learning performance. In all of these cases it should be possible to improve the above estimates, but this remains to be worked out.

We use the above estimates to compute all entries in the matrices \hat{v} , \hat{V} , and $\hat{W}_{a,o}$.

3.2.2 Choosing characteristic and indicative sequences

We use for characteristic and indicative sequences all io-sequences $\bar{s}_1, \dots, \bar{s}_r \in (A \times O)^l$ of a given *eventlength* l , where $r := |A \times O|^l \gg d$. For sufficiently large l , this will assure that the matrices Π and Φ have full rank d .

Note that, in lieu of the construction given in proposition 3 for bases of the sets $K = \text{span} \{(\mathbb{1}\tau_{\bar{s}})^\top \mid \bar{s} \in (A \times O)^*\}$ and $S = \text{span} \{\tau_{\bar{s}}w_0 \mid \bar{s} \in (A \times O)^*\}$ defined in definition 5, we need to consider only characteristic and indicative sequences of length less than or equal to d to obtain bases of K and S and therefore obtain matrices Π and Φ of full rank d . So we could in theory use an eventlength of $l = d$. But this would lead to extremely large sparse matrices \hat{v} , \hat{V} , and $\hat{W}_{a,o}$ with large variance, unless we had access to an unreasonably long training sequence \bar{x} , making the model estimation very inaccurate. So in practice we want to use a small eventlength l , but such that the matrices Π and Φ have full rank d . We assume that an appropriate eventlength l is given.

By using as characteristic sequences all io-sequences of length l , we obtain control over the column sums of the matrix Π :

$$\begin{aligned} \mathbb{1}\Pi &= \sum_{\bar{s} \in (A \times O)^l} \mathbb{1}\tau_{\bar{s}} = \sum_{\bar{s} \in (A \times O)^{l-1}} \sum_{a \in A} \sum_{o \in O} \mathbb{1}\tau_{a,o\bar{s}} = \sum_{\bar{s} \in (A \times O)^{l-1}} \sum_{a \in A} \mathbb{1}\tau_{\bar{s}} \\ &= |A| \sum_{\bar{s} \in (A \times O)^{l-1}} \mathbb{1}\tau_{\bar{s}} = |A|^l \mathbb{1}, \end{aligned} \quad (14)$$

since we know that $\mathbb{1}\tau_a = \mathbb{1}$ for all $a \in A$.

In the resulting estimate \hat{V} , we then remove all zero columns, i.e. we remove the corresponding indicative sequences. This will not affect the rank of \hat{V} and also not the condition $\mathbb{1}\Pi = |A|^l \mathbb{1}$ but helps by discarding columns that contribute only very little information.

It may be possible to improve the learning results by deleting even more columns and also rows of the matrix \hat{V} if the entries are based on too few frequency counts. Also, one may group the sequences into sets of sequences and consider their “probabilities” instead or, ultimately, one may wish to consider indicative and characteristic sequences of varying lengths. All of this requires further research and remains to be worked out.

Also note that we currently assume that we are given the correct model dimension d and an appropriate eventlength l (which we currently set to be the smallest value such that $r = |A \times O|^l$ is significantly larger than d). It is, however, possible to choose the learning parameters d and l empirically by estimating the matrices $\hat{V}^{(l)}$ for increasing eventlengths l and determining their numerical rank $d^{(l)}$ as described in section 9.3 of [3]. As soon as $d^{(l+1)} = d^{(l)}$, we then set the desired model dimension to d and use the eventlength l . However, this procedure is just a heuristic and not yet implemented and tested. Alternatively, due to the speed

of the IO-OOM model construction once the matrices \hat{v} , \hat{V} , and $\hat{W}_{a,o}$ have been estimated, we may also use cross-validation to set the most appropriate model dimension d and the eventlength l .

3.2.3 Finding appropriate characterizer C and indicator Q

Note that grouping of characteristic and indicative sequences and thereby implicitly also exploiting statistics of shorter sequences is achieved by an appropriate choice of characterizer C and indicator Q . However, the way that we will choose the characterizer C and indicator Q relies on a different principle, namely minimizing an upper bound for the error of the estimated model $\hat{\mathcal{M}}$. This principle was described recently for OOMs in [11], leading to an algorithm called the *error controlling* (EC) algorithm. The procedure and derivation from the OOM-EC algorithm for finding good characterizer C and indicator Q have since been streamlined and can be used in their current form [10] for IO-OOMs as well, with only minor modifications pertaining to the case of IO-OOMs.

For this reason, we will only explain the basic idea and refer the reader to [11] and [10] for the details:

Assume the estimates \hat{v} , \hat{V} , and $\hat{W}_{a,o}$ are given. Let $\tau = [\tau_{s_1}; \tau_{s_2}; \dots; \tau_{s_{|A \times O|}}]$ (in MATLAB notation) be the matrix where the τ_s are stacked below each other, for all $s \in A \times O$. And let $\hat{\tau}$ be defined in the same way, where $\hat{\tau}$ is the estimate obtained via the learning equation (10) for some characterizer C and indicator Q . Then, by the main proposition 3 in [10], the relative estimation error is bounded by

$$\frac{\|\tau - \hat{\tau}\|}{\|\tau\|} < \|C\| \cdot \|Q(C\hat{V}Q)^{-1}\| \cdot k, \quad (15)$$

where k is a constant that depends only on the alphabet size $|A \times O|$ and the estimation errors $\|V - \hat{V}\|$ and $\|W_{a,o} - \hat{W}_{a,o}\|$. Note that this constant k differs slightly in the case of IO-OOMs, since here the matrix τ will have column sums $|A|$ instead of 1 as in the case of OOMs. But this is the only difference, and the proof as given in [10] remains valid in the case of IO-OOMs.

So the goal is to choose the characterizer C and indicator Q such that the quantity $\|C\| \cdot \|Q(C\hat{V}Q)^{-1}\|$ – called the *robustness indicator* – is minimized, subject to the condition $\mathbb{1}C = \mathbb{1}$. This is done exactly as described in [10].

Finally, we need to comment on the fact that we obtain an estimated model $\hat{\mathcal{M}} = (\{\hat{\tau}_{a,o}\}_{a \in A, o \in O}, \hat{w}_0)$ via a characterizer C that satisfies $\mathbb{1}C = \mathbb{1}$, which implies by equation (14) that $\mathbb{1}C\Pi = |A|^l \mathbb{1}$, where l is the used eventlength, while we really need to have $\mathbb{1}C\Pi = \mathbb{1}$ to estimate an equivalent IO-OOM: We should modify the characterizer to $C' = \frac{1}{|A|^l} C$. However, this makes no difference, as can be seen from the learning equation (10). Also, when estimating the starting state, we should use $\hat{w}_0 = \frac{1}{|A|^l} C\hat{v}$, but since we normalize the estimated starting state to

have a column sum of one in equation (11), this amounts to the same thing.

4 Empirical results

We test the performance of the IO-OOM learning algorithm as described above on the task of learning models of seven simple input-output systems that are described as POMDPs and have been used as benchmark tasks in the literature [2]. The input policy that is commonly used is simply to choose a control input at each time step from a uniform random distribution. This way, the input policy is blind and complete – a fact that we, however, do not exploit.

For each of the POMDP tasks, we train IO-OOM models \mathcal{M} on training sequences of varying lengths ranging from 10^3 to 10^7 time steps using the described IO-OOM EC algorithm. Note that we provide the correct model dimension and some reasonable eventlength as learning parameters for each of the problems. We then evaluate these models by calculating their average one-step prediction error $E(\mathcal{M})$ on testing sequences of length $L = 10^4$:

$$E(\mathcal{M}) = \frac{1}{L} \sum_{i=1}^L \frac{1}{|O|} \sum_{o \in O} (P(o | \bar{h}_{i-1} a_i) - \hat{P}(o | \bar{h}_{i-1} a_i))^2, \quad (16)$$

where P is the correct probability and \hat{P} is the model prediction for the next output given the testing history \bar{h}_{i-1} and the current input a_i .

This is the format that has been used in the literature for evaluating the main PSR algorithms, and we show the PSR learning performance as a comparison measure. The results are given in figure 1.

The main conclusion that we can draw from these results is that the IO-OOM learning algorithm presented is sound and performs well on the standard benchmark tasks – even in its described basic version.

In addition, several more detailed remarks are in order:

Remark 1. Currently, the IO-OOM learning algorithm is given the desired model dimension and the employed eventlength as parameters. This will need to be automated. While the model dimension is chosen to be the correct model dimension, the eventlength is chosen ad hoc as a reasonable value (here usually 2). This is certainly not optimal and leaves room for improvement. In fact, in the “Cheese maze” problem, the eventlength $l = 2$ is too small to capture all relevant statistics, even though $|A \times O|^2 = 28^2 \gg d = 11$. In fact, in this case a length of 2 for characteristic sequences would suffice, as then the resulting matrix Π has full rank 11, while with indicative sequences of length 2 the matrix Φ only has rank 10. This means that we can at best recover a 10-dimensional approximation to the correct 11-dimensional IO-OOM, and this explains why the error levels out

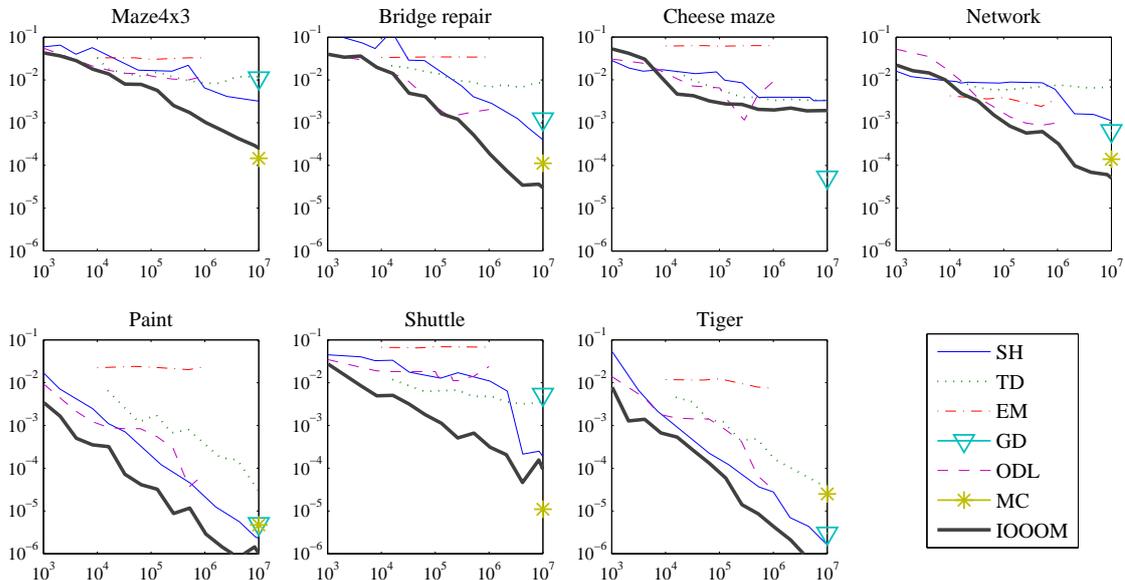


Figure 1: Learning results of the described IO-OOM EC algorithm in comparison to various PSR learning algorithms on POMDP benchmark problems from [2]. Average one-step prediction error (y -axis) is shown for increasing training sequence lengths (x -axis). The performance of the presented IO-OOM EC learning algorithm is shown. The other results presented are taken from the PSR publications, where the following algorithms were introduced: SH: “Suffix History”, TD: “Temporal Difference” [9]; EM: “Expectation Maximization” for comparison as reported in [9]; GD: “Gradient Descent” (results only available for training sequences of length 10^7) [7]; ODL: “Online Discovery and Learning” [6]; MC: “Monte Carlo” reset learning (very inefficient. The training sequence lengths used are much longer than 10^7) [5]. The data is presented in the format used in [9].

instead of converging to zero. Of course, this is remedied by using longer indicative sequences (a length of 3 does suffice), but we keep the results as shown, since this illustrates the importance of a correct choice of the eventlength. Interestingly, most of the PSR learning algorithms seem to suffer from this problem as well.

Remark 2. We use a general purpose estimation procedure for the relevant “probability” values $P_o(\bar{s})$, although in these benchmark problems the input policy is blind and complete. This fact is exploited by the PSR learning algorithms, which use simpler estimates that are, in fact, *only* valid for blind policies, as noted in [1]. However, understanding the role of the input policy is a major topic and left for further research.

Remark 3. Previously, we ran a similar comparison study of PSR learning vs. OOM learning on the same benchmark range, where we modeled the stochastic processes combined from the input policy and the CSP with ordinary OOMs and later extracted the output probabilities conditioned on the given input. This was

possible and worked well for these problems, since the input policy is extremely simple and the same during training and testing. The OOM learning algorithm employed was the “Efficiency Sharpening” algorithm [4] that uses a suffix-tree representation of the training sequence to exploit characteristic and indicative sequences of varying lengths. This is presumably the main reason why, in fact, the OOM learning results on this benchmark range were even better than the IO-OOM learning results presented, showing that there is certainly still room for improvement of the basic IO-OOM EC learning algorithm presented by exploiting variable-length characteristic and indicative sequences.

5 Conclusion

In this report we have generalized the main theorems for OOMs to the case of IO-OOMs and have built the IO-OOM theory much in analogy to the available OOM theory. We used this to transfer the error controlling OOM learning algorithm to the case of IO-OOMs, yielding the first functional IO-OOM learning algorithm.

While the IO-OOM learning algorithm as presented is shown to be asymptotically correct (under the technical assumptions of stationarity and ergodicity of the stochastic process and completeness of the input policy), i.e. it will recover a perfect IO-OOM in the limit of infinite training data, we could also demonstrate that it performs competitively on simple but standard benchmark problems.

Still, this work presents only the first step towards a practically useful and efficiency-optimized IO-OOM learning algorithm. We have presented only a basic version of the error controlling IO-OOM learning algorithm and have noted several points that demand further research:

- Currently, we need to set the model dimension and an appropriate event-length as parameters to the learning algorithm. A heuristic for automatically choosing these parameters based on purely algebraic criteria does exist [3], but this needs to be refined.
- The presented learning algorithm only exploits the statistics of subsequences of the training sequence that have a fixed eventlength. It should be possible to exploit the statistics of substrings of various lengths simultaneously by using a suffix-tree representation of the training sequence in a way similar to [4]. This would eliminate the need for an eventlength parameter entirely and should further improve the statistical efficiency of the model estimation. Most importantly, though, this could lead to a significantly more space and runtime efficient implementation of the learning algorithm – issues that have been neglected in this report and also require further investigation.

- Given the nature of the learning task of estimating an IO-OOM for a CSP from training data resulting from the combination of the CSP with some input policy, it is only natural to investigate the role of the input policy. Concretely, we want to understand how we can utilize prior knowledge of the input policy to improve the learning efficiency, and – if allowed – how one should choose the input policy to optimize the learning efficiency. Similar questions have been investigated for PSRs in [1].
- Finally, we will need to perform further empirical studies on more elaborate data sets to evaluate the practical value of the IO-OOM formalism and the IO-OOM learning algorithm.

References

- [1] M. Bowling, P. McCracken, M. James, J. Neufeld, and D. F. Wilkinson. Learning predictive state representations using non-blind policies. In W. W. Cohen and A. Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning*, pages 129–136. ACM, 2006.
- [2] A. Cassandra. Tony’s POMDP page. <http://www.cs.brown.edu/research/ai/pomdp/>, 1999.
- [3] H. Jaeger. Observable operator models of stochastic processes: a tutorial. GMD Report 42, German National Research Center for Information Technology, Sankt Augustin, Germany, December 23 1998.
- [4] H. Jaeger, M. Zhao, K. Kretschmar, T. Oberstein, D. Popovici, and A. Kolling. Learning observable operator models via the es algorithm. In S. Haykin, J. C. Príncipe, T. J. Sejnowski, and J. McWhirter, editors, *New Directions in Statistical Signal Processing: From Systems to Brains*, Neural Information Processing, chapter 14, pages 417–464. MIT Press, Cambridge, MA, USA, October 2006.
- [5] M. R. James and S. P. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In C. E. Brodley, editor, *Proceedings of the 21st International Conference on Machine Learning*, page 53. ACM Press, 2004.
- [6] P. McCracken and M. H. Bowling. Online discovery and learning of predictive state representations. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

- [7] S. P. Singh, M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone. Learning predictive state representations. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning*, pages 712–719. AAAI Press, 2003.
- [8] E. W. Wiewiora. *Modeling Probability Distributions with Predictive State Representations*. PhD thesis, University of California, San Diego, 2008.
- [9] B. Wolfe, M. R. James, and S. P. Singh. Learning predictive state representations in dynamical systems without reset. In L. D. Raedt and S. Wrobel, editors, *Proceedings of the 22nd International Conference on Machine Learning*, pages 980–987. ACM, 2005.
- [10] M. Zhao and H. Jaeger. A bound on modeling error in observable operator models and an associated learning algorithm. Manuscript submitted for publication, November 2007.
- [11] M. Zhao and H. Jaeger. The error controlling algorithm for learning OOMs. Technical Report 6, Jacobs University, Bremen, Germany, March 2007.