Learning Observable Operator Models via the Efficiency Sharpening Algorithm

Herbert Jaeger

International University Bremen
Overview

1. OOM Basics
2. Learning Equations
3. Reverse OOMs and Reverse Characterizers
4. The ES Algorithm
5. Case Studies
1 OOM Basics
What's a Future?

For a robot, or anybody else modelling stochastic processes, the future is a probability distribution over possible future developments.
Update Dynamics of Future Distributions

\[ \cdots t \quad t+1 \quad t+2 \quad \cdots \]

- Observations update expectations, that is, future distributions.
- \( \cdots E_t, E_{t+1}, \cdots \) : observations. Formally, events in observation space \( \sigma \)-algebra.
The Basic Idea

observable events = operators that change distributions

- set of all future distributions is a (functional) vector space $V$
- for every event $E$ an "observable operator" $\tau_E$
- observable operators operate on $V$
A Dual Description of Processes

Process:

\[ a \quad b \quad a \quad a \]

Standard Models:

\[ \begin{array}{ccc}
  & a & \\
  & \overleftrightarrow{T} & \\
 b & \overleftrightarrow{T} & a & \overleftrightarrow{T} \\
\end{array} \]

OOM Models:

\[ \begin{array}{cccc}
  \tau_a & \tau_b & \tau_a & \tau_a \\
\end{array} \]
Every stochastic process has an OOM with linear observable operators. Its distribution can be calculated from the OOM through

\[ P(X_0 = E_0, \ldots, X_N = E_N) = 1_{1 \times \text{dim}} \tau_{E_N} \cdots \tau_{E_0} \mathcal{W}_0 \]
Components of an OOM

We consider only symbolic processes with finite observation set $O = \{a_1, ..., a_n\}$.

Recall: $P(X_0 = b_0, ..., X_N = b_N) = 1 \tau_{b_N} \cdots \tau_{b_0} w_0$

A ($m$-dim, matrix-represented) OOM $\mathcal{A}$ is a structure

$$\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in O}, w_0)$$

where $w_0 \in \mathbb{R}^m$ is the starting state, and $(\tau_a)_{a \in O} \in \text{Mat}_{m \times m}$ are the observable operators.
**Definition.** An OOM is a structure \((\mathbb{R}^m, (\tau_a)_{a \in O}, w_0)\), such that

1. \(\mu = \sum_{a \in O} \tau_a\) has column sums = 1,

2. \(1w_0 = 1\),

3. for every sequence \(a_1...a_N \in O^n\): \(1\tau_{a_N} \cdots \tau_{a_1} w_0 \geq 0\).

**Theorem.** Such a structure yields the distribution of a stochastic process by

\[
P(X_0 = b_0, \ldots, X_N = b_N) = 1\tau_{b_N} \cdots \tau_{b_0} w_0.
\]
OOMs and HMMs

- An OOM $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in O}, w_0)$ is equivalent to some HMM, if all operator matrices and the starting state are non-negative.
- Every $m$-dimensional HMM is equivalent to some $m$-dimensional OOM.
- For any $3 \leq m \leq k$, there exist $m$-dimensional OOMs equivalent to some HMM of minimal dimension $k$.
- There exist 3-dimensional OOMs not equivalent to any HMM.
OOMs as Sequence Generators

\[ A = (\mathbb{R}^2, \{ \tau_a, \tau_b \}, w_0) \quad O = \{ a, b \} \]

- Compute \(1\tau_a w_0, 1\tau_b w_0\)
- \((1\tau_a w_0, 1\tau_b w_0)\) is a P-vector

Select a vs. b according to P-vector
- Apply operator
- Renormalize to component sum 1 to obtain \(w_1\)

\(a\) is observed
Equivalence Theorem

Two OOMs \((\mathbb{R}^m, (\tau_a)_{a \in \Sigma}, w_0), (\mathbb{R}^m, (\tau'_a)_{a \in \Sigma}, w'_0)\), where \(m\) is minimal, generate the same process iff there exists a coordinate transformation \(\rho: \mathbb{R}^m \rightarrow \mathbb{R}^m\), which preserves component sums of vectors, such that

\[ \tau'_a = \rho \tau_a \rho^{-1} \text{ for all } a \in O. \]
Corollary 1  For a given OOM \( \mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in O}, w_0) \) there exist infinitely many different but equivalent OOMs of same dimension.

Corollary 2  It is decidable whether two OOMs are equivalent.
Related Work

- "Random systems with complete connections"
  (1969 Iosifescu & Theodorescu)

- Equivalence of HMMs
  (1957 Blackwell & Koopmans
   ... Gilbert; Dharmadikari; Heller; Fox & Rubin
   1992 Ito, Amari, Kobayashi)

- Generalized HMMs
  1997 Upper
Ramifications and Extensions

- Input-output OOMs\(^1\) (aka "predictive state representations", Sutton et al), including basic learning algorithm, generalize POMDPs
- Matrix representations of continuous-valued, discrete-time OOMs, including basic learning algorithm\(^1\)
- Adaptive online learning algorithms\(^2\)
- ML learning algorithm for non-negative OOMs\(^3\)
- Math stuff:
  - general OOM theory of arbitrary stochastic processes\(^1\)
  - decomposition theorems for operators\(^1\), connections to QM\(^2\)
  - Hilbert space theory of future distribution space\(^2\)

\(^1\) For tech reports visit my publications page  \(^2\) Unpublished  \(^3\) Submitted (M. Zhao)
The Unsolved Non-Negativity Problem

- Recall non-negativity condition 3 from definition:
  \[ \forall a_1,\ldots,a_N \in O : 1 \tau_{a_N} \cdots \tau_{a_1} w_0 \geq 0. \]
- No method known to check for given set of operator matrices whether this holds.
- Would be solved if we could check whether a finite set of linear maps leaves some convex cone invariant.
- Solutions known only for some special cases\(^1\).
- OOMs estimated from data often violate non-negativity. Heuristic bypasses exist\(^2\).

\(^1\) Edwards/McDonald/Tsatsomeros, On matrices with common invariant cones. Linear Algebra and ist Applications, 2004 (online version) \(^2\) In Haykin/Principe/Sejnowski/McWhirter (eds), New Directions in Statistical Signal Processing, MIT Press (to appear)
2 Learning Equations
Some Shorthand Notation

- For \( P(X_0 = a_0, \ldots, X_N = a_N) \) write \( P(a_0 \ldots a_N) \) or \( P(\bar{a}) \).
- For \( P(X_{N+1} = b_1, \ldots, X_{N+M} = b_M \mid X_0 = a_0, \ldots, X_N = a_N) \) write \( P(b_{N+1} \ldots b_{N+M} \mid a_0 \ldots a_N) \) or \( P(\bar{b} \mid \bar{a}) \).
- Symbols "\( b \)" always appear later in forward time than symbols "\( a \)".
- Indices \( i \) in \( \bar{a}_i \) always refer to alphabetical enumeration of all strings \( \bar{a} \) of same length.
- \( \tau_{\bar{a}} = \tau_{a_N} \cdots \tau_{a_0} \), \( w_{\bar{a}} = \tau_{\bar{a}} w_0 / 1 \tau_{\bar{a}} w_0 \)

the state after initial history \( \bar{a} \).
Characterizers

**Definition.** Let $k \geq 1$, and $\overline{b_1} \ldots \overline{b_k}$ be the alphabetical enumeration of $O^k$. Let $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in O}, w_0)$ be an OOM of some process with distribution $P$ and states $w_{\overline{a}}$. Let $C \in Mat_{m \times k}$ have unit column sums. Then $C$ is a **characterizer** of length $k$ of $\mathcal{A}$ iff for all $\overline{a} \in O^*$:

$$w_{\overline{a}} = C \begin{pmatrix} P(\overline{b_1} | \overline{a}) \\ \vdots \\ P(\overline{b_k} | \overline{a}) \end{pmatrix}$$
Intuitive Interpretation

\[ w_{\overline{a}} = C \left( \begin{array}{c} P(\overline{b}_1 \mid \overline{a}) \\ \vdots \\ P(\overline{b}_K \mid \overline{a}) \end{array} \right) \]

A characterizer \( C \) transforms the future distribution after initial history \( \overline{a} \) (as represented by the probs \( P(\overline{b}_i \mid \overline{a}) \)) into the OOM state \( w_{\overline{a}} \).
Some Properties of Characterizers

1. Every OOM has characterizers of length $k$ for sufficiently large $k$.

2. Let $C_0$ be a characterizer of length $k$ of $\mathcal{A}$. Then the class of all characterizers of length $k$ of $\mathcal{A}$ is

\[
\{ C_0 + G \mid G \begin{pmatrix} P(\bar{b}_1 \mid \bar{a}_1) & \cdots & P(\bar{b}_1 \mid \bar{a}_k) \\ \vdots & \ddots & \vdots \\ P(\bar{b}_k \mid \bar{a}_1) & \cdots & P(\bar{b}_k \mid \bar{a}_k) \end{pmatrix} = 0 \}
\]

\[
=: \{ C_0 + G \mid GV = 0 \}
\]
Learning Equations

Let $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \mathcal{\Omega}}, w_0)$ be an OOM of some process with distribution $P$ with characterizer $C$. Let

$$V = \begin{pmatrix} P(b_1 | a_1) & \cdots & P(b_1 | a_k) \\ \vdots & \ddots & \vdots \\ P(b_k | a_1) & \cdots & P(b_k | a_k) \end{pmatrix}, \quad W_a = \begin{pmatrix} P(ab_1 | a_1) & \cdots & P(ab_1 | a_k) \\ \vdots & \ddots & \vdots \\ P(ab_k | a_1) & \cdots & P(ab_k | a_k) \end{pmatrix}.$$ 

Then

$$\tau_a = CW_a (CV)^+$$
Basic Learning Algorithm

1. Choose a characterizer $C$.

2. Estimate (by obvious frequency counting from data)

$$
\hat{V} = \begin{pmatrix}
\hat{P}(b_1 | \overline{a}_1) & \cdots & \hat{P}(b_1 | \overline{a}_k) \\
\vdots & & \vdots \\
\hat{P}(b_k | \overline{a}_1) & \cdots & \hat{P}(b_k | \overline{a}_k)
\end{pmatrix},
\hat{W}_a = \begin{pmatrix}
\hat{P}(ab_1 | \overline{a}_1) & \cdots & \hat{P}(ab_1 | \overline{a}_k) \\
\vdots & & \vdots \\
\hat{P}(ab_k | \overline{a}_1) & \cdots & \hat{P}(ab_k | \overline{a}_k)
\end{pmatrix}.
$$

3. Compute $\hat{\tau}_a = CW_a (CV)^+$. 
Properties of Basic Learning Algorithm

1. Yields asymptotically correct estimates $\hat{\tau}_a$ with any characterizer $C$.

2. Model variance (statistical efficiency) depends crucially on choice of $C$.

3. Search for "good" (low model variance) characterizer can be confined to some class $\{C_0 + G \mid GV = 0\}$. 
3 Reverse OOMs and Reverse Characterizers
**Definition.** Let

\[ \mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \mathcal{O}}, w_0), \quad \mathcal{A}^r = (\mathbb{R}^m, (\tau_a^r)_{a \in \mathcal{O}}, w_0^r) \]

be OOMs with distributions \( P, P^r \). Then \( \mathcal{A}^r \) is a reverse OOM to \( \mathcal{A} \) iff for all \( a_1 \ldots a_N \in O^* \):

\[
P(a_1 \ldots a_N) = 1\tau^r_{a_0} \cdots \tau^r_{a_N} w_0^r
\]

\[
= 1\tau_0 \cdots \tau_{a_N} w_0
\]

\[
= P^r(a_N \ldots a_1)
\]
**Theorem.** Let $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \mathcal{O}}, w_0)$ be an OOM with distribution $P$. Then

$$\mathcal{A}^r = (\mathbb{R}^m, (D \tau'_a D^{-1})_{a \in \mathcal{O}}, w_0)$$

is a reverse OOM to $\mathcal{A}$, where $\tau'_a :$ transpose of $\tau_a$, $D = \text{diag}(w_0)$. 
Theorem. Let $\mathcal{A} = (\mathbb{R}^m, (\tau_a)_{a \in \Omega}, w_0)$ be an OOM with distribution $P$. Let $\mathcal{A}^r = (\mathbb{R}^m, (D \tau'_a D^{-1})_{a \in \Omega}, w_0)$ be a reverse OOM to $\mathcal{A}$. Define $C = \begin{pmatrix} w^r_{b_1} & \cdots & w^r_{b_k} \end{pmatrix}$ and

$$\pi_{\mathcal{A}} = \begin{pmatrix} 1 \tau_{b_1}^- \\ \vdots \\ 1 \tau_{b_k}^- \end{pmatrix}.$$ 

Then $C^r = (C \pi_{\mathcal{A}})^{-1} C$ is a characterizer for $\mathcal{A}$, its (unique) reverse characterizer.
Take-home message.

- The reverse characterizer $C^r$ of an OOM consists (up to a coordinate transform) of states of the reverse OOM.
- It can be effectively constructed from $\mathcal{A}$. 
4 The ES Algorithm
Goal. Given some characterizer $C_0$, within class of equivalent characterizers $\{C_0 + G \mid GV = 0\}$ search for statistically most efficient characterizer, i.e., characterizer that minimizes model variance when used in learning equation $\hat{\tau}_a = C\hat{W}_a (C\hat{V})^+$. 

Observation¹). Model variance is dominated by variance of $C\hat{V}$.

Theorem²). Within $\{C_0 + G \mid GV = 0\}$, the reverse characterizer $C^r$ minimizes variance of $C\hat{V}$.

A "Poor Man's" ES Learning Algorithm

1. Estimate initial model $\mathcal{A}^{(0)} = (\mathbb{R}^m, (\tau^{(0)}(a))_{a \in \mathcal{O}}, \psi(0))$
   with some ad hoc characterizer $C^{(0)}$.

2. Compute reverse characterizer $C^{r(1)}$ and compute
   $\mathcal{A}^{(1)}$ via $\hat{\tau}^{(1)}_a = C^{r(1)} \hat{W}_a (C^{r(1)} \hat{V})^+$.
   Rationale: $C^{r(1)}$ will come closer to true reverse characterizer, hence model variance lower, hence in expectation $\mathcal{A}^{(1)}$ better than $\mathcal{A}^{(0)}$.

3. Iterate 2 ("efficiency sharpening") until satisfaction.
Properties

1. Computationally cheap: dominated by pseudoinverse of \( \dim \times |O|^k \) matrix per iteration
2. Easy to implement
3. Statistically inefficient because only counting statistics of substrings of length \( 2k \) is exploited
A Suffix-Tree Based ES Algorithm

**Observation:** reverse states are (up to coordinate transform) stochastic approximations to forward states:

\[
\begin{align*}
\text{b} & \quad \text{a} & \\
\text{w}_r^{bbabbab} & \quad \text{w}_r^{bbabba} & \\
\text{w}_r^{bbabb} & \quad \text{w}_r^{bbab} & \quad \text{w}_r^{bb} & \quad \text{w}_r^b & \quad \text{w}_r^0
\end{align*}
\]

\[
\text{w}_{ba} \approx \frac{1}{2} (\text{w}_r^{bbabb} + \text{w}_r^{bb})
\]

1. Exploit \( \tau_x : w_0 \mapsto P(x)w_x, w_a \mapsto P(x \mid a)w_{ax}, w_{ba} \mapsto P(x \mid ba)w_{bax}, \text{etc.} \)
2. Sort argument-value pairs like \( w_a \mapsto P(x \mid a)w_{ax} \) column-wise into matrices \( V \) and \( W_x \). (Use suffix tree of reverse string to implement this.)
3. Estimate \( \hat{\tau}_x = W_x V^{-1} \).
4. Iterate according to ES principle with new reverse states.
Position: 0 1 2 3 4 5 6 7 8 9 10
String: $b \ ax b b \ ax b b$
Rev. States: $w_{bbababbab}^r \ldots w_b^r w_0^r$
Rev. string: $b b a b a b b a b$

$w_{bbababbab}^r + w_{bbababba}^r + w_{bbabab}^r + w_{bbab}^r + w_{b}^r$: stoch. approx. of $w_{ba}^r$ by reverse states.
Properties of Suffix-Tree ES Algorithm

- Uses information from all substrings of all lengths.
- Needs to create suffix tree once (cost linear in data size).
- Needs to run reverse model over reverse string once per iteration. Cost per iteration about 1/2 that of EM iteration.
- A pain to implement
5 Case Studies
A Simple Case Study

- Data generated by 6-state, transition-emitting HMMs with 3 symbols
- Training sequence length 2000, testing length 20,000
- 5 ES iterations, max. 100 EM iterations for HMM/EM comparison
- Averages over 20 independently created data sets (different generators) shown
Learning Dynamics

- Diagrams show training loglik evolution of suffix-tree based ES algorithm
- Sharp initial rise to plateau
- Followed by ± strong jitter around plateau
- Long-term behavior: typically settles on fixed point, sometimes enters cyclic attractor ⇒ no Lyapunov function exists, no convergence guaranteed!
A Not So Simple Case Study

- Training and testing data generated from a Mark Twain short story
- 27 symbols, training/test length each about 21,000.
- 5 ES iterations, max. 100 EM iterations for HMM/EM comparison
An ES Iteration in a Nutshell

- Pretend that current model is correct.
- Use that to construct minimal variance estimator (for estimation of models from sample data drawn from the current model's distribution).
- Re-estimate model. Should be better because obtained from a hopefully lower-variance estimator...

You see, this method comes with no warranty… it builds on a hope!
Discussion

- ES algorithms do not maximize training likelihood.
- ES algorithms (attempt to) minimize variance.
- ES algorithms produce sequence of estimators (and therefore -- implicitly -- estimates). Each estimator is asymptotically correct and asymptotically unbiased (and -- with reservations -- unbiased\(^1\)).
- Computational cost much less (1-2 orders of magnitude) than HMM-EM. Model test performance equal or better than HMM-EM on data sets checked so far.
- Algebraic overfitting warning criterium: no overfitting iff $\text{numrank}(CV) \geq \text{model dimension}$
- ES iterations may not converge.

\(^1\) Checked empirically for model-generated probability predictions; proven for special cases; statement depends on representation of OOM (may not hold for matrix entries).
Thank you.