Learning diffeomorphism models of robotic sensorimotor cascades

Andrea Censi*                    Richard M. Murray

* 5th year graduate student, defending in a few weeks, currently looking for a job*

Outline

1. The bootstrapping scenario
2. Modeling sensorimotor cascades
3. Learning sensorimotor cascades
4. Comparing models: representation, assumptions, and invariances

* not a real job
In the **bootstrapping scenario**, an embodied agent has no prior knowledge about its sensors, its actuators, and the external environment.

“sensels”: pixels, range readings, ...

[Diagram showing uninterpreted observations and commands flowing through the agent to the “world” (sensorimotor cascade) with unknown sensor(s), external environment, and unknown actuator(s).]
• **Intellectual** motivation
  • One of the main problems of AI is grounding symbols in sensorimotor experience.

• **Scientific** interest
  • Modeling “general purpose” cortex circuitry
  • Modeling mental developmental theories

• **Engineering** motivation
  able to learn models → less design effort
  able to verify models → safer and more resilient operation
  [other ICRA paper]
First levels of a bootstrapping agent

- From bits to signals
- From scrambled sensels to coherent data
- Inferring instantaneous action models
- Inferring long-term action models
- Inferring behaviorally relevant hidden states

• Current challenge: make theory more rigorous to be useful in robotics.

(remixed from Kuipers’10)
Outline

1. The bootstrapping scenario
2. Modeling sensorimotor cascades
3. Learning sensorimotor cascades
4. Comparing models: representation, assumptions, and invariances

1. All systems vs all robots
2. The vehicles universe
3. Low-level dynamics of canonical robot sensors
4. Approximations and tasks
• The “set of all robots” is much smaller than the set of all dynamical systems.

• “Vehicles universe”: pimped-up Braitenberg vehicles
Robotic actuators

- manipulators
- wheeled dynamics
  - diff. drive
  - car-like
  - omnidirectional
- derived
  - legged snakes
  - ... 

Robotic sensors

- "canonical" robotic sensors
  - field sampler
  - range-finder
  - camera
- related or derived
  - compass
  - sonars
  - GPS
  - ... 
- not included
  - touch
  - proprioception
  - ... 

- temperature,
  - concentration, ...
• Can an agent work with all possible combinations, with no prior knowledge?
- “Canonical robotic sensors” [ICRA’11] have **similar dynamics at the sensel level.**

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field-sampler</td>
<td>$y(s) = \text{intensity at point } s$</td>
<td>$\dot{y}(s) = (s \times \nabla_s y(s)) \cdot \omega + \nabla_s y(s) \cdot \nu$ $s \in \mathbb{R}^3$</td>
</tr>
<tr>
<td>Range-finder</td>
<td>$y(s) = \text{distance in direction } s$</td>
<td>$\dot{y}(s) = (s \times \nabla_s y(s)) \cdot \omega + (\nabla_s \log y(s) - s) \cdot \nu$ $s \in \mathbb{S}^2$</td>
</tr>
<tr>
<td>Camera</td>
<td>$y(s) = \text{luminance in direction } s$</td>
<td>$\dot{y}(s) = (s \times \nabla_s y(s)) \cdot \omega + \mu(s) \nabla_s y(s) \cdot \nu$ $s \in \mathbb{S}^2$</td>
</tr>
</tbody>
</table>

$v(t), \omega(t): \text{kinematic velocities}$
• We cannot model everything perfectly…

• But **what tasks can we perform**, based on imperfect approximations?

• What tasks can you define **independently** of the system?
Outline

1. The bootstrapping scenario
2. Modeling sensorimotor cascades
3. Learning sensorimotor cascades
4. Comparing models: representation, assumptions, and invariances

• Models that can be applied to raw sensorimotor data:
  • BDS: [ICRA’11] generic bilinear relations
  • BGDS [IROS’11] bilinear flows
  • DDS [this paper] diffeomorphisms
The BDS model assumes a generic bilinear relation among $y$, $u$, and $\dot{y}$.

\[ y \in \mathbb{R}^n \quad u \in \mathbb{R}^k \quad \dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i + \text{noise} \]

The $k \times n \times n$ tensor $M$ consists of one $n \times n$ matrix for each command.

**Examples of learned tensors**

- **+ camera**
- **+ range finder**
- **+ field sampler**

\[ u^1 = v_x \quad u^2 = v_y \quad u^3 = \omega \]
The **BGDS model** assumes the observations are a function on a manifold $S$ and that the dynamics is a function of the gradient of $y$.

$$
y : S \rightarrow \mathbb{R} \quad u \in \mathbb{R}^k \quad \dot{y}(s) = \sum_i \sum_j G_i^j(s)(\nabla_j y(s) + B_i(s))u^i + \text{noise}
$$

$G$ and $B$ are tensor fields on $S$.

- For camera data, it is slightly more general than learning optic flow fields.

- It can fit range-finder and field-sampler data as well.
The **DDS model** assumes the observations are a function on a manifold $S$ and that each command $u_i$ induces a **diffeomorphism** $\phi_i$ of $S$.

The DDS model assumes the observations are a function on a manifold $S$ and that each command $u_i$ induces a **diffeomorphism** $\phi_i$ of $S$.

Algorithmic representation:
- we learn a pointwise, discretized max-likelihood estimate of $\phi_i$
- … and an uncertainty estimate
The DDS model assumes the observations are a function on a manifold $S$ and that each command $u_i$ induces a diffeomorphism $\phi_i$ of $S$.

$$u_i \text{ for } T \text{ seconds} \rightarrow y_T(s) = y_0(\phi_i(s))$$

$u \in U$ (discrete set)

Can represent camera dynamics but not nearness (a hidden state)

 DDS model [this paper]
The DDS model assumes the observations are a function on a manifold $S$ and that each command $u_i$ induces a diffeomorphism $\phi_i$ of $S$.

\[ u_i \text{ for } T \text{ seconds} \quad \rightarrow \quad y_T(s) = y_0(\phi_i(s)) \]

$u \in U$ (discrete set)

Can represent range-finder data, if preprocessed to create a “local map”.

**Readings**

**“Local map”**

**Prediction**

learned diffeomorphism
Outline

1. The bootstrapping scenario
2. Modeling sensorimotor cascades
3. Learning sensorimotor cascades
4. Comparing models: representation, assumptions, and invariances

- How to compare models?
- Representation matters!
- Invariance properties of bootstrapping agents
- More complicated = less invariant
• How can we compare models?

• Complexity of learning
  of inference
• Storage requirements
• Data needed for learning
• Tolerance to noise
• Approximation properties

• What “assumptions” (prior knowledge) has the model on the system and its representation?
What are the assumptions of an agent about the data representation?

- Physical system
- Sensels
- Gray code
- All equivalent representations
- Shuffled bits
- Encrypted bits

\[ y \in \mathbb{R}^n \quad y \in \{\blacksquare, \square\}^{nb} \quad y \in \{\blacksquare, \square\}^{nb} \quad y \in \{\blacksquare, \square\}^{nb+\Delta} \]
• This is not just about the “format” of the data.
• Representation is a nuisance that must be rejected.

assumptions about the representation = fixed, invertible transformations tolerated by the agent
<table>
<thead>
<tr>
<th>Format</th>
<th>Dynamics</th>
<th>Invariances for ( u )</th>
<th>Invariances for ( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDS</td>
<td>( u \in \mathbb{R}^k \quad y \in \mathbb{R}^n )</td>
<td>( \dot{y}^s = \sum_i \sum_v M_{iv}^s v^s u^i )</td>
<td>?</td>
</tr>
<tr>
<td>BGDS</td>
<td>( u \in \mathbb{R}^k \quad y : S \rightarrow \mathbb{R} )</td>
<td>( \dot{y}(s) = \sum_i \sum_j G_{ij}(s) (\nabla_j y(s) + B_i(s)) u^i )</td>
<td>?</td>
</tr>
<tr>
<td>DDS</td>
<td>( u \in U \quad y : S \rightarrow \mathbb{R} )</td>
<td>( y_T(s) = y_0(\phi_i(s)) )</td>
<td>?</td>
</tr>
</tbody>
</table>
For BDS/BGDS, the commands enter linearly in the dynamics:

<table>
<thead>
<tr>
<th>BDS</th>
<th>BGDS</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u \in \mathbb{R}^k$, $y \in \mathbb{R}^n$</td>
<td>$u \in \mathbb{R}^k$, $y : S \to \mathbb{R}$</td>
<td>$u \in U$, $y : S \to \mathbb{R}$</td>
</tr>
</tbody>
</table>

- nothing moves for $u = 0$
- "linear structure":
  - $-u$ has the inverse effect of $+u$
  - $+2u$ has twice the effect as $+u$

- $\Psi^s = \sum_i \sum_v M_{iv}^s y^v u^i$
- $\dot{y}(s) = \sum_i \sum_j G_i^j(s)(\nabla_j y(s) + B_i(s))u^i$
- $y_T(s) = y_0(\varphi_i(s))$

- Invariances for $u$ and $y$:
  - $\text{GL}(k)$

- Preserved by invertible linear maps $\text{GL}(k)$

- $\times$ unicycle with different representation
- $\checkmark$ unicycle
- $\checkmark$ differential drive
- $\times$ car-like

$u' = f(u)$
- The DDS model only assumes that $U$ is some finite alphabet.

<table>
<thead>
<tr>
<th>BDS</th>
<th>$u \in \mathbb{R}^k$, $y \in \mathbb{R}^n$</th>
<th>$\dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i$</th>
<th>GL($k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGDS</td>
<td>$u \in \mathbb{R}^k$, $y : S \rightarrow \mathbb{R}$</td>
<td>$\dot{y}(s) = \sum_i \sum_j G_i^j(s) (\nabla_j y(s) + B_i(s)) u^i$</td>
<td>GL($k$)</td>
</tr>
<tr>
<td>DDS</td>
<td>$u \in U$, $y : S \rightarrow \mathbb{R}$</td>
<td>$y_T(s) = y_0(\phi_i(s))$</td>
<td>Aut($U$)</td>
</tr>
</tbody>
</table>

Aut($U$) = all bijections of $U$
The BDS model is invariant to linear transformations of $y$.

<table>
<thead>
<tr>
<th>Model</th>
<th>$u \in \mathbb{R}^k$</th>
<th>$y \in \mathbb{R}^n$</th>
<th>$\dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i$</th>
<th>$\text{Invariances}$</th>
<th>$\text{for } u$</th>
<th>$\text{for } y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDS</td>
<td></td>
<td></td>
<td></td>
<td>GL($k$)</td>
<td>GL($n$)</td>
<td></td>
</tr>
<tr>
<td>BGDS</td>
<td>$u \in \mathbb{R}^k$</td>
<td>$y : S \to \mathbb{R}$</td>
<td>$\dot{y}(s) = \sum_i \sum_j G_{ij}^s (\nabla_j y(s) + B_i(s)) u^i$</td>
<td>GL($k$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDS</td>
<td>$u \in U$</td>
<td>$y : S \to \mathbb{R}$</td>
<td>$y_T(s) = y_0(\phi_i(s))$</td>
<td>Aut($U$)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- The BDS/BGDS model are invariant to diffeomorphisms of the observation space.

<table>
<thead>
<tr>
<th>BDS</th>
<th>$u \in \mathbb{R}^k$</th>
<th>$y \in \mathbb{R}^n$</th>
<th>$\dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i$</th>
<th>$GL(k)$</th>
<th>$GL(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGDS</td>
<td>$u \in \mathbb{R}^k$</td>
<td>$y : S \rightarrow \mathbb{R}$</td>
<td>$\dot{y}(s) = \sum_i \sum_j G_i^j(s)(\nabla_j y(s) + B_i(s)) u^i$</td>
<td>$GL(k)$</td>
<td>$Diff(S)$</td>
</tr>
<tr>
<td>DDS</td>
<td>$u \in U$</td>
<td>$y : S \rightarrow \mathbb{R}$</td>
<td>$y_T(s) = y_0(\phi_i(s))$</td>
<td>$Aut(U)$</td>
<td>$Diff(S)$</td>
</tr>
</tbody>
</table>

- Interpretation: only need topology of the data → no assumptions about sensor calibration

- There are other invariances as well such as local value transformations (e.g., contrast transformations)
• Invariance properties encode the agent’s assumptions on the representation.

<table>
<thead>
<tr>
<th>DDS</th>
<th>BDS</th>
<th>BGDS</th>
<th>DDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u \in \mathbb{R}^k$</td>
<td>$y \in \mathbb{R}^n$</td>
<td>$u \in \mathbb{R}^k$</td>
<td>$y : S \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>$u \in U$</td>
<td>$y : S \rightarrow \mathbb{R}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Invariances for $u$</th>
<th>for $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL($k$)</td>
<td>GL($n$)</td>
</tr>
<tr>
<td>GL($k$)</td>
<td>Diff($S$)</td>
</tr>
<tr>
<td>Aut($U$)</td>
<td>Diff($S$)</td>
</tr>
</tbody>
</table>

• Invariance properties induce a partial order on the agents:

\[
\begin{align*}
\dot{y}^s &= \sum_i \sum_v M_{iv}^s y^v u^i \\
\dot{y}(s) &= \sum_i \sum_j G_i^j(s) (\nabla_j y(s) + B_i(s)) u^i \\
y_T(s) &= y_0(\phi_i(s))
\end{align*}
\]

\[
more \ \powerful = less \ \assumptions = more \ \invariant
\]

• It also says something about **semantics**, in a sense that can be made precise, in a talk longer than 13 minutes.
- Often, a more complicated algorithm is less invariant.

<table>
<thead>
<tr>
<th>format</th>
<th>dynamics</th>
<th>Invariances for $u$</th>
<th>Invariances for $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDS $u \in \mathbb{R}^k$ $y \in \mathbb{R}^n$</td>
<td>$\dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i$</td>
<td>GL($k$)</td>
<td>GL($n$)</td>
</tr>
<tr>
<td>BGDS $u \in \mathbb{R}^k$ $y : S \to \mathbb{R}$</td>
<td>$\dot{y}(s) = \sum_i \sum_j G_i^j(s)(\nabla_j y(s) + B_i(s)) u^i$</td>
<td>GL($k$)</td>
<td>Diff($S$)</td>
</tr>
<tr>
<td>DDS $u \in U$ $y : S \to \mathbb{R}$</td>
<td>$y_T(s) = y_0(\phi_i(s))$</td>
<td>Aut($U$)</td>
<td>Diff($S$)</td>
</tr>
</tbody>
</table>

- For example, consider adding some regularization scheme, using the metric $d(s_1,s_2)$:

| DDS $u \in U$ $y : S \to \mathbb{R}$ | $y_T(s) = y_0(\phi_i(s))$ + some regularization using the metric $d(s_1,s_2)$ | Aut($U$) | Isom($S$) |

→ more tolerant to noise, but less invariant!
• **Bootstrapping:** can an embodied agent learn everything from scratch?

• Many challenges:
  • algorithmic
  • conceptual

• Many low-hanging fruits for robotics

  *e.g., fault detection [other ICRA paper]*

  *“calibration from correlation” [preprint with Scaramuzza]*
extra slides
\( u^0 \): angular velocity
\( u^1 \): linear velocity

\( u^0 \): left wheel velocity
\( u^1 \): right wheel velocity

\( u^0 \): driving velocity
\( u^1 \): steering angle
Bilinear Dynamics Sensors (BDS)

- The BDS model assumes a generic bilinear relation among \( y, u, \) and \( \dot{y} \).

\[
\dot{y}^s = \sum_i \sum_v M_{iv}^s y^v u^i \quad y \in \mathbb{R}^n \quad u \in \mathbb{R}^k
\]

The \( k \times n \times n \) tensor \( M \) consists of one \( n \times n \) matrix for each command.

Learning the one-dimensional dynamics

- In this case, the \( M \) matrix approximates the gradient operator.

\textit{real dynamics:} \quad \dot{y} = \nabla_s y \omega

\textit{postulated model:} \quad \dot{y} = My \omega

\textit{learned parameters:}

\[
\dot{y} = \begin{pmatrix} +1 \\ -1 \\ 0 \end{pmatrix} y \omega
\]