# Learning the combinatorial structure of demonstrated behaviors with inverse feedback learning

Olivier Mangin, Pierre-Yves Oudeyer

INRIA Flowers, Université Bordeaux 1, France

October 07, 2012

# Going beyond trajectories

- the **task** is relevant

- need for **generalization**
  - over context
  - over starting position
  - body shape and size

# Motions and behaviors are complex

Picture: `http://www.hongkiat.com/`

# The problem

From observation of complex human activities,
learn a **dictionary** of **primitive objectives**
that can be combined together to explain observations

# The problem

From observation of complex human activities,
learn a **dictionary** of **primitive objectives**
that can be combined together to explain observations

**Training:**

- multiple demonstrations of complex tasks
- unobserved dictionary shared amongst demonstrations

# The problem

From observation of complex human activities,
learn a **dictionary** of **primitive objectives**
that can be combined together to explain observations

**Training:**

- multiple demonstrations of complex tasks
- unobserved dictionary shared amongst demonstrations

**Evaluation:**

- learn the dictionary
- solve same task as demonstrator

# Imitator model (similar to [Brillinger, 2007, Jetchev & Toussaint, 2011])

- task represented by the minimization of a **cost function** $f$

$$f : \mathcal{Q} \longrightarrow \mathbb{R}$$

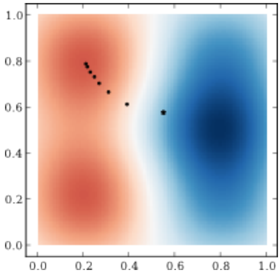
# Imitator model (similar to [Brillinger, 2007, Jetchev & Toussaint, 2011])

- task represented by the minimization of a **cost function** $f$

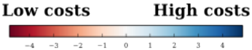
$$f: \mathcal{Q} \longrightarrow \mathbb{R}$$

- **motions:**

$$q_{t+1} = \operatorname*{argmin}_q \left[ f(q) + \alpha \left\| \frac{q - q_t}{\delta_t} \right\|^2 \right]$$
$$= q_t - \frac{1}{\alpha} \nabla f(q)$$




Cost function values:



**Dictionary:**

**D** of $K$ **primitive objectives**

represented by cost functions:

$$(g^k)_{k \in [|1,K|]}$$

**Each demonstration** $i$**:**

- **coefficients:**

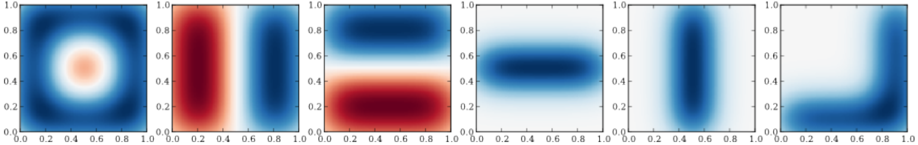
$$(a^i_k)_{k \in [|1,K|]}$$

- **complex task:**

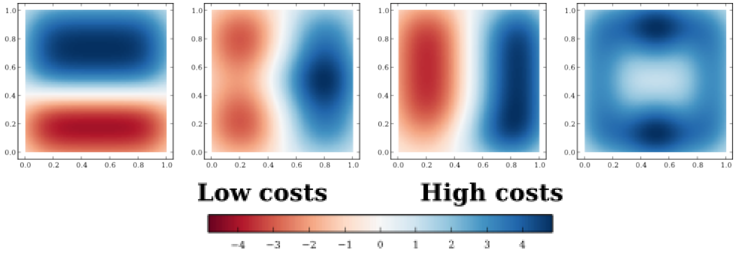
$$f^i(q) = \sum_{k=1}^{K} a^i_k g^k(q)$$

## **Example:** $K = 6$, only 2 or 3 coefficients are non-zero

### Dictionary (primitive objectives)

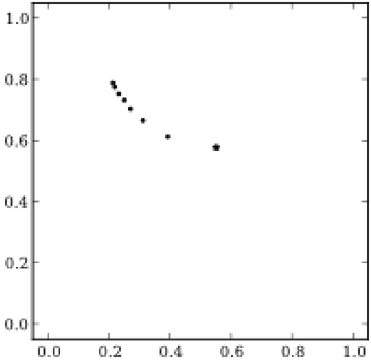


### Complex tasks



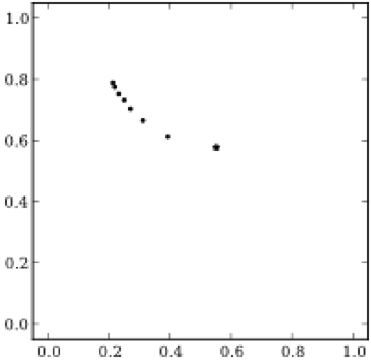
# Inferring the task from a demonstration [Brillinger, 2007]

**Observed trajectory**

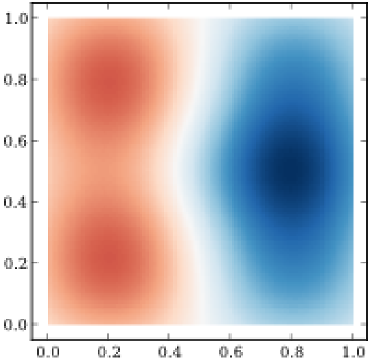


# Inferring the task from a demonstration [Brillinger, 2007]

**Observed trajectory**




**Cost function to infer**



# Inferring the task from a demonstration [Brillinger, 2007]

**No dictionary is learnt ("flat learner")**

Cost function parametrization:

- **features:** $\varphi : \ \mathcal{Q} \longrightarrow \mathbb{R}^F$
- **parameters:** $\beta \in \mathbb{R}^F$

$$f(q) = \varphi(q)^T \beta$$

# Inferring the task from a demonstration [Brillinger, 2007]

## No dictionary is learnt ("flat learner")

Cost function parametrization:

- **features:** $\varphi : \mathcal{Q} \longrightarrow \mathbb{R}^F$
- **parameters:** $\beta \in \mathbb{R}^F$

$$f(q) = \varphi(q)^T \beta$$

## Motion:

$$\forall t,\ \frac{q_{t+1} - q_t}{\delta_t} = -\lambda \mathbf{J}(q_t)^T \beta$$

# Inferring the task from a demonstration [Brillinger, 2007]

## No dictionary is learnt ("flat learner")

Cost function parametrization:

- **features:** $\varphi : \ \mathcal{Q} \longrightarrow \mathbb{R}^F$
- **parameters:** $\beta \in \mathbb{R}^F$

$$f(q) = \varphi(q)^T \beta$$

**Motion:**

$$\forall t, \ \frac{q_{t+1} - q_t}{\delta_t} = -\lambda \mathbf{J}(q_t)^T \beta$$

**Matrix formulation:**

$$Y = \mathbf{\Phi} \beta$$

$$\mathbf{Y} = \begin{pmatrix} \ \\ \cdots \\ \ \end{pmatrix} \qquad \mathbf{\Phi} = \begin{pmatrix} -\lambda \mathbf{J}(q_0)^T \\ \cdots \\ -\lambda \mathbf{J}(q_T)^T \end{pmatrix}$$

# Inferring the task from a demonstration [Brillinger, 2007]

## No dictionary is learnt ("flat learner")

Cost function parametrization:

- **features:** $\varphi : \mathcal{Q} \longrightarrow \mathbb{R}^F$
- **parameters:** $\beta \in \mathbb{R}^F$

$$f(q) = \varphi(q)^T \beta$$

**Motion:**

$$\forall t, \frac{q_{t+1} - q_t}{\delta_t} = -\lambda \mathbf{J}(q_t)^T \beta$$

**Matrix formulation:**

$$Y = \mathbf{\Phi} \beta$$

$$\mathbf{Y} = \begin{bmatrix} \blacksquare \\ \cdots \\ \blacksquare \end{bmatrix} \qquad \mathbf{\Phi} = \begin{bmatrix} -\lambda \mathbf{J}(q_0)^T \\ \cdots \\ -\lambda \mathbf{J}(q_T)^T \end{bmatrix}$$

**Problem:**

$$\underset{\beta}{\operatorname{argmin}} \left\| Y - \mathbf{\Phi}\beta \right\|_2^2$$

# Extension to the dictionary case

## Model in the dictionary case

$$f^i(q) = \varphi(q)^T D a^i$$

$$Y^i = \mathbf{\Phi}^i \mathbf{D} a^i$$

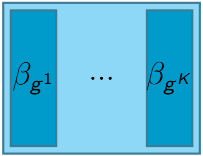
$$D = \begin{bmatrix} \beta_{g^1} & \cdots & \beta_{g^K} \end{bmatrix}$$

# Extension to the dictionary case

## Model in the dictionary case

$$f^i(q) = \varphi(q)^T D a^i$$

$$Y^i = \mathbf{\Phi}^i \mathbf{D} a^i$$

$D =$ 


## Problem

$$\mathcal{L}(\mathbf{D}, \mathbf{A}) = \sum_{i=1}^{n} \| Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \|_2^2$$

$$\underset{\mathbf{D}, \mathbf{A}}{\text{argmin}} \ \mathcal{L}(\mathbf{D}, \mathbf{A})$$

# Learning the dictionary, knowing the coefficients

- unknown symbolic labels provided by the teacher
- coefficients learnt another way

# Learning the dictionary, knowing the coefficients

- unknown symbolic labels provided by the teacher
- coefficients learnt another way

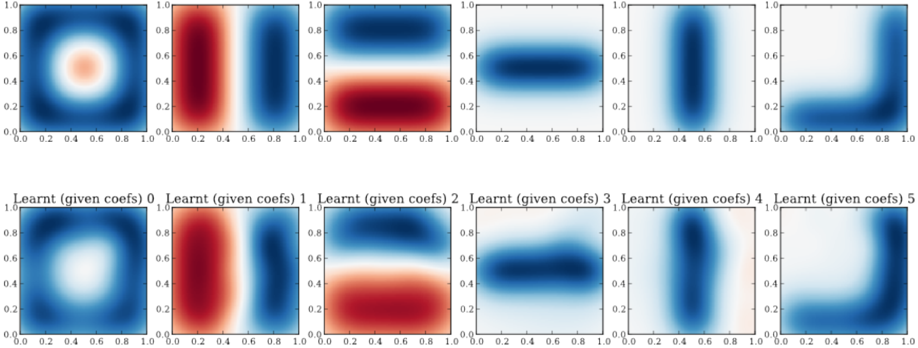
**Problem:**

$$\underset{\mathbf{D}}{\operatorname{argmin}} \sum_{i=1}^{n} \| Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \|_2^2$$

**Gradient:**

$$\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{A}) = -2 \sum_{i=1}^{N} {\mathbf{\Phi}^i}^T \left[ Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \right] {a^i}^T$$

# Original and learnt the dictionary



# Imitation task

**D** learnt during training

**Test:**

- an unknown demonstration is observed
- complex task inferred as coefficients $a$
- … and used for imitation

# Imitation task

**D** learnt during training

**Test:**

- an unknown demonstration is observed
- complex task inferred as coefficients $a$
- ... and used for imitation

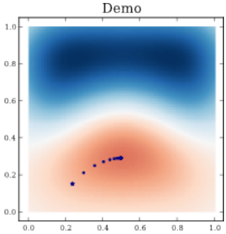
**Learn the coefficents from the dictionary:**

$$\underset{\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^{n} \| Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \|_2^2$$

**Solution:**

$$\forall i, \; a^i = (\mathbf{D}^T {\mathbf{\Phi}^i}^T \mathbf{\Phi}^i \mathbf{D})^{\#} \mathbf{D}^T {\mathbf{\Phi}^i}^T Y^i$$

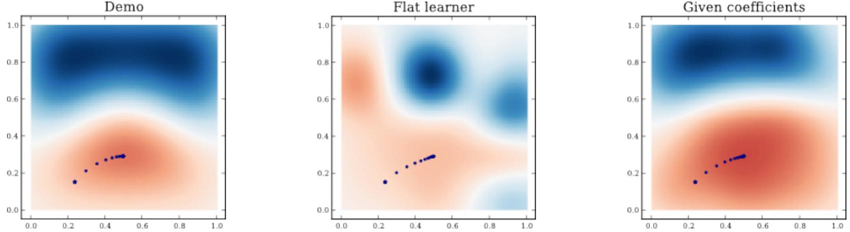
# Imitation task




**Imitation of the demonstration**

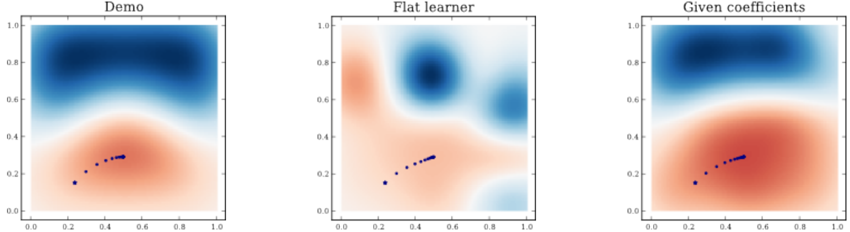
# Imitation task

**Imitation of the demonstration**

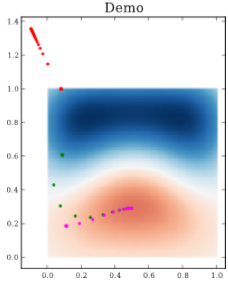


# Imitation task

**Imitation of the demonstration**

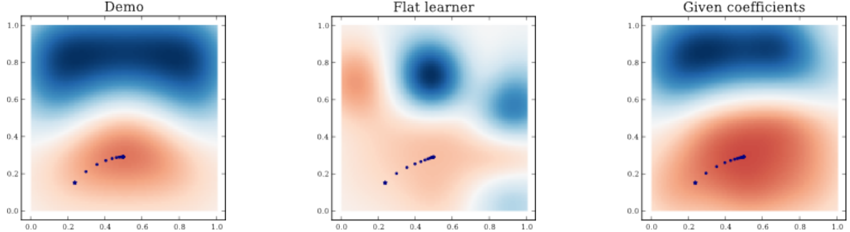



**Imitation on new starting points**

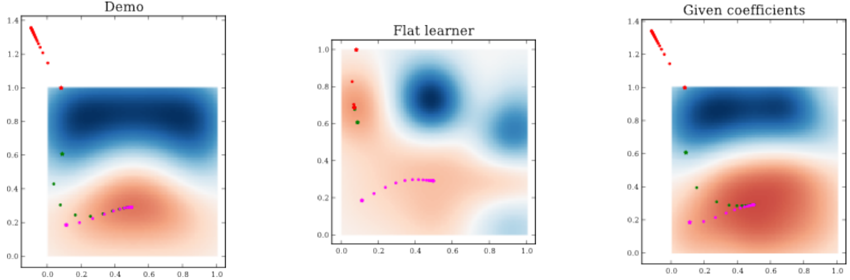


# Imitation task

## Imitation of the demonstration




## Imitation on new starting points



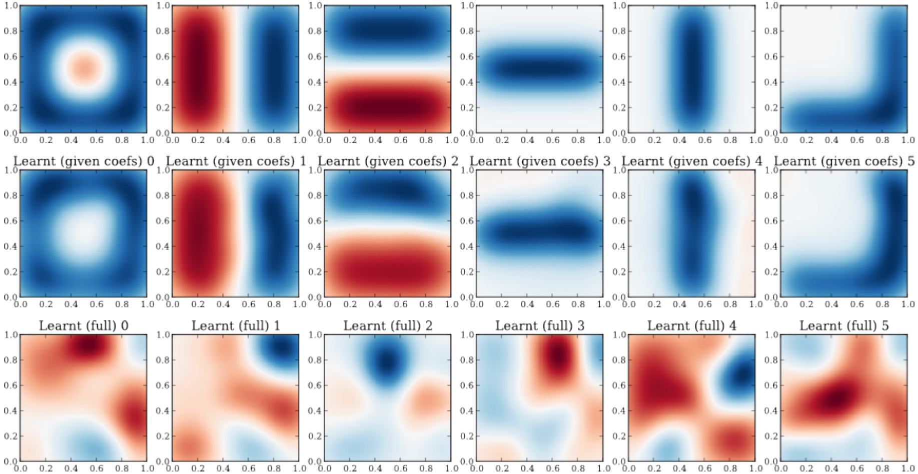
# Learning both the dictionary and the coefficients

**Strategy:**

alternate optimization of **D** and **A**

1. initialize (e.g. randomly) **D** and **A**
2. optimize **D** with **A** fixed
3. optimize **A** with **D** fixed
4. go back to **step 2**

# Learning the dictionary with and without knowing the coefficients



# Perspectives

- an ill-posed problem:
  - add structural constraints,
  - add language or other modality,
  - consider a developmental trajectory.
- the evaluation problem
  - on synthetic demonstrator:
    - compare learnt primitive objectives,
    - evaluate imitation
  - on real demonstrator
- better imitator models (from inverse reinforcement learning, ...)

# Thank you!

# Bibliography


Brillinger, D. R. (2007).
Learning a Potential Function From a Trajectory.
*IEEE Signal Processing Letters*, 14(11), 867–870.

Jetchev, N. & Toussaint, M. (2011).
Task Space Retrieval Using Inverse Feedback Control.
In L. Getoor & T. Scheffer (Eds.), *International Conference on Machine Learning*, number 28 in ICML '11 (pp. 449–456). New York, NY, USA: ACM.