Neural and probabilistic learning methods for robotics and other domains

Tutorial at SoftNet 2018

Nice, October 14th, 2018
Prof. Dr. Elmar Rueckert

latest updated July 23rd 2018
Introduction & Motivation

*Humanoid robots are among the most complex machines on earth.*

*And you will learn here how to build, teach and program them.*
Challenges in motor skill learning
More than robotics ...

The challenges in understanding humans and in building intelligent humanoids are converging!

- 700 muscles
- 100 joints
- $100 \times 10^6$ photo receptors
- $10^2$ FA-I receptors per fingertip

- 53 degrees of freedom
- 4 force/torque sensors
- $1.8 \times 10^6$ photo receptors
- $\sim 2000$ tactile sensors
In **humans** we suffer from noise, accuracy, delays.

Despite **robot** vision is richer and more precise, robot motion is faster and more accurate their motor skills are inferior, **why?**
Why probabilistic methods?

- Uncertainties in the sensor measurements.
- Delays and transmission errors.
- Unmodeled dynamics (friction dynamics, coriolis forces, etc.).
- Partial observability.
Why neural methods?

- The optimal methods structure / features are often unknown.
- Millions of data samples can be processed in $O(n)$.
- Complex multimodal probability distributions can be represented (in contrast to commonly used unimodal Gaussians).
- Predictions can be computed in realtime in $O(1)$. 
bimanual action planning and coordination
Research questions

1. How can humans learn new motor skills within a few trials?
   a. “control only when necessary” - motor variability
   b. exploiting kinematic and task redundancy
   c. transfer of related skills

2. How do humans solve cognitive reasoning tasks in huge spaces?
   a. planning in stochastic environments
   b. inferring multiple solutions in milliseconds
   c. online model adaptation from intrinsic motivation signals.
Interested in a brief robotics history?
A brief historical review


1941 Isaac Asimov: Three laws of “robotics”:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.
A brief historical review

1968 “Shakey” of the “Stanford Research Institute” defines a landmark in robotics:

- basic planning and navigation skills.
- object detection and manipulation capabilities.
A brief historical review

1973 Ichiro Kato develops the first “full-scale” anthropomorphic humanoid, WABOT I.
A brief historical review

1996 **Honda** presents its P2

they started with E0 in 1986
A brief historical review

2004 The Italian Institute of Technology presents the ICub (intelligent man-cub).
A brief historical review

2017 Boston dynamics’ Atlas impresses the robotics community.
I. Kinematics, Dynamics & Model Learning

II. Representations of Skills & Imitation Learning

III. Feedback, Priorities & Torque Control

IV. Reinforcement Learning & Policy Search

V. Cognitive Reasoning & Planning

VI. Sensor Integration & Fusion
II. Representations of Skills & Imitation Learning

I. Kinematics, Dynamics & Model Learning

IV. Reinforcement Learning & Policy Search

III. Feedback, Priorities & Torque Control

V. Cognitive Reasoning & Planning

VI. Sensor Integration & Fusion
Where do we need representations of skills?

II.1 Movement primitives.

Fan Zeng, Beshah Ayalew and Mohammed Omar: Roboticc automotive paint curing using thermal signature feedback, 2009
The complexity of skill representations

II.1 Movement primitives.

a Task dependent trunk trajectories

b Left wrist trajectories

c Right wrist trajectories

c CoP trajectories

Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control. Nature Publishing Group: Scientific Reports, 6 (28455), 2016.
The complexity of skill representations

Data:

- 17 markers with x,y,z at 100Hz
- 2 force plates at 100Hz (CoM at x,y)
- 9600 trials of 20 subjects of
- On avg. 100 samples per trial

\[(17 \cdot 3 + 2 \cdot 2) \cdot 9600 \cdot 100 > 50 \text{ Mio. data pts}\]

Just for a single movement skill!
II.1 Movement primitives.

Naive vector/matrix representation scales in \(O(d \times T \times K)\), where 
- \(d\) ... number of joints, force plates or markers, 
- \(T\) ... number of time steps per trial \(k = 1 \ldots K\)

Even when we average over all 9600 trials we would need to store **5500 data points per second**!
Naive via-point representation

scales in $O(d \times N \times K)$, where $d$ ... number of joints, force plates or markers, $n$ ... number of via-points comp. from the avg. over $K$ trials

$$f_i(t) = c_{i0} + c_{i1}(t - t_i) \quad t \in [t_i, t_{i+1}]$$

Leads to non-smooth trajectories!

Can we do better?

Yes by using the dynamics model for planning a route through via-points!

-3 via-pts per marker

-Averaging the via-pts over the 9600 trials

$$(17 \cdot 3 + 2 \cdot 2) \cdot 3 = 165$$

parameters to learn

for a 55-dimensonal movement representation in ~10KB memory
Spline representation

Splines are piecewise polynomials (don’t use solely polynomials)

\[ f_i(t) = c_{i0} + c_{i1}(t - t_i) + c_{i2}(t - t_i)^2 + \cdots + c_{ik}(t - t_i)^k \]

\( t \in [t_i, t_{i+1}] \)

- \( C_0 \) continuity
- \( C_0 \) & \( C_1 \) continuity
- \( C_0 \) & \( C_1 \) & \( C_2 \) continuity

\[ f'_i(t_{i+1}) = f'_{i+1}(t_{i+1}) \]

\[ f''_i(t_{i+1}) = f''_{i+1}(t_{i+1}) \]

Fig. 1. A trajectory passing through \( n \) knots

Spline representation

Scale in $O(d \times n \times k)$, where $d$ ... number of joints, force plates or markers, $n$ ... **number of knots** at times $t_1, t_2, ..., t_n$ of order $k$

$$f_i(t) = c_{i0} + c_{i1}(t - t_i) + c_{i2}(t - t_i)^2 + \cdots + c_{ik}(t - t_i)^k$$

- 3 via-pts per marker
- Averaging the via-pts over the 9600 trials
- $k=3$ for cubic-splines ($c_{i0}, c_{i1}, c_{i2}, c_{i3}$)

$$\left(17 \cdot 3 + 2 \cdot 2\right) \cdot 3 \cdot 4 = 660$$

parameters to learn!
Desired features of skill representations

- **Compact** (few parameters to learn).
- **Smooth** (need to compute derivatives for velocities and controls).
- **Flexible** generalizables to different tasks (goal locations, orientations, etc.).
- Can be learnt from the data through **imitation learning (IM)**.
- Self-improvement through **reinforcement learning (RL)**.
- **Composable** through sequencing and **co-activation**.
- **Stochastic**, can model the variance of the data.
- **Coupled**, can model the coupling of joints.
My approach: learning probabilistic models

Learning problem:

\[ P(A|B) = \frac{P(A, B)}{P(B)} \]

given data samples from \( P(A, B) \)
assuming priors \( P(A), P(B) \)
A basis functions

0 movement phase 1
[1] Generative Model: \( y_t = \Phi_t w \)

[2] Gaussian Features: \( \phi_{t,i} = \frac{1}{Z} \exp \left( -\frac{1}{2\lambda} (z(t) - c_i)^2 \right) \)

[3] Learning the Prior: \( w^{[i]} = (\Phi_{1:T}^T \Phi_{1:T} + \lambda I)^{-1} \Phi_{1:T}^T \tau^{[i]} \)

p(\tau) = \int p(\tau|w)p(w)dw \\
= \int \mathcal{N}(y_{1:T} | \Phi_{1:T} w, \Sigma_y) \mathcal{N}(w | \mu_w, \Sigma_w) dw \\
= \mathcal{N}(y_{1:T} | \Phi_{1:T} w, \Phi_{1:T} \Sigma_w \Phi_{1:T}^T + \Sigma_y)
\]
[5] Conditioning, given the prior $\mathcal{N}(w|\mu_w, \Sigma_w)$

$$p(w_o|o) \propto \mathcal{N}(o|\Phi_o w_o, \Sigma_o)p(w)$$

$$: = \mathcal{N}(w_o|\mu_{w|o}, \Sigma_{w|o}),$$

with $\mu_{w|o} = \mu_w + \Sigma_w \Phi_o^T(\Sigma_o + \Phi_o \Sigma_w \Phi_o^T)^{-1}(o - \Phi_o \mu_w)$,

and $\Sigma_{w|o} = \Sigma_w - \Sigma_w \Phi_o^T(\Sigma_o + \Phi_o \Sigma_w \Phi_o^T)^{-1}\Phi_o \Sigma_w$,

Result:

$$p(\tilde{y}) = \mathcal{N}(\tilde{y}_{1:T}|\Phi_{1:T}\mu_{w|o}, \Phi_{1:T}\Sigma_{w|o}\Phi_{1:T}^T + \Sigma_y).$$
Movement model learning example

Rueckert, Elmar; Lioutikov, Rudolf; Calandra, Roberto; Schmidt, Marius; Beckerle, Philipp; Peters, Jan. Low-cost Sensor Glove with Force Feedback for Learning from Demonstrations using Probabilistic Trajectory Representations. ICRA 2015 Workshop on Tactile and force sensing for autonomous compliant intelligent robots, 2015.
How do we train the model from data?

\[ q_t = [q_t^{[1]}, q_t^{[2]}, \ldots, q_t^{[d]}]^T \]
\[ \forall t \in \mathbb{N}_0 \]

- Kinesthetic teaching (see the picture).
- Teleoperation (e.g., by using a joystick).
- Visual observation (using cameras or optical markers).
- Sensor suits (IMUs, e.g., Xsense.com).

The last two approaches require to map the data onto the robot which is often problematic!
Imitation learning

Given: \( q_t = [q_t^{[1]}, q_t^{[2]}, \ldots, q_t^{[d]}]^T \)
\[ \forall t \in \mathbb{N}_0 \]

Or in vector notation per dim. \( d \):
\[ q^{[d]} = [q_1^{[d]}, q_2^{[d]}, \ldots, q_T^{[d]}]^T \]

Let's consider only one dimension:
\[ q = [q_1, q_2, \ldots, q_T]^T \]
Radial basis functions as features

Modeling complex shapes through Gaussians

\[
f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t)w_j}{\sum_{j=1}^{N} \Psi_j(t)}
\]

\[
\Psi_j(t) = \exp\left(-1/(2\sigma^2)(x(t) - c_j)^2\right)
\]

Note \(N=8\) Gaussian basis functions are used here.
Radial basis functions as features
Modeling complex shapes through Gaussians

\[ f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t) w_j}{\sum_{j=1}^{N} \Psi_j(t)} \]

fixed basis functions
scaled by learnable parameters
normalization
Imitation learning

I. Compute the **target** function from the data:

\[ \tilde{f} = [q_1, q_2, \ldots, q_T] \]

II. Compute the **model's** function term:

\[ f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t) w_j}{\sum_{j=1}^{N} \Psi_j(t)} \]

where

\[ \Psi = \begin{bmatrix} \tilde{\Psi}_1^{[1]}, & \tilde{\Psi}_1^{[2]}, & \ldots, & \tilde{\Psi}_1^{[N]} \\ \tilde{\Psi}_2^{[1]}, & \ldots, & \ldots, & \ldots \\ \ldots \\ \tilde{\Psi}_T^{[1]}, & \ldots, & \ldots, & \tilde{\Psi}_T^{[N]} \end{bmatrix} \]

\[ \tilde{\Psi}_t^{[j]} = \frac{\Psi_j(t)}{\sum_{j=1}^{N} \Psi_j(t)} \]
Imitation learning

I. Compute the target function from the data:

\[ \tilde{f} = [q_1, q_2, \ldots, q_T] \]

II. Compute the model's function term:

\[ f = \Psi w \quad \text{from} \]

III. Minimizing the objective:

\[ J = \frac{1}{2} ( \tilde{f} - f)^T (\tilde{f} - f) = \frac{1}{2} (\tilde{f} - \Psi w)^T (\tilde{f} - \Psi w) \]

Results in:

\[ w = (\Psi^T \Psi + \lambda I)^{-1} \Psi^T \tilde{f} \]

Link to a nice related tutorial
How many basis functions are optimal?

Depends on the task and has to be numerically evaluated!

-6 Gaussians per dim.
- \((17 \cdot 3 + 2 \cdot 2) \cdot 10 = 550\) parameters to learn
for a 55-dimensional movement representation
II.3 Example of probabilistic movement primitives.

Imitation learning through optical markers

Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control. Nature Publishing Group: Scientific Reports, 6 (28455), 2016.
II.3 Example of probabilistic movement primitives.

Imitation learning through optical markers

Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control. Nature Publishing Group: Scientific Reports, 6 (28455), 2016.
Is one movement primitive enough?

No!

- Complex tasks require a large number of primitives.
- Reusable primitives can be sequenced or co-activated (in time).
- Non-homogeneous spaces require separate primitives (in space).
- Tradeoff between the number of primitives and their complexity (num. of Gaussians)!
II.2 DMPs

Imitation learning of a library of primitives

II.3 Example of probabilistic movement primitives.

Incremental Imitation learning a primitive library

New demos

adapt existing skill by adding new demonstration

add new skill
Stark, Svenja; Peters, Jan; Rueckert, Elmar.  
II.3 Example of probabilistic movement primitives.

When a single primitive is not sufficient

Rueckert, Elmar; Mundo, Jan; Paraschos, Alexandros; Peters, Jan; Neumann, Gerhard. Extracting Low-Dimensional Control Variables for Movement Primitives. Proceedings of the International Conference on Robotics and Automation (ICRA), 2015.
Can we generalize?

Using **probabilistic trajectory models** which are discussed in **Part Two**!

II.3 Example of probabilistic movement primitives.

Demonstration Based Trajectory Optimization for Generalizable Robot Motions. 
Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), 2016
You want to test PTMs yourself?


- More details and exercises in: [https://ai-lab.science](https://ai-lab.science)
more at: https://rob.ai-lab.science/publications/

Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan
**Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control**
Journal Article

Rueckert, Elmar; Mundo, Jan; Paraschos, Alexandros; Peters, Jan; Neumann, Gerhard
**Extracting Low-Dimensional Control Variables for Movement Primitives**
Inproceedings

Rueckert, Elmar; Lioutikov, Rudolf; Calandra, Roberto; Schmidt, Marius; Beckerle, Philipp; Peters, Jan
**Low-cost Sensor Glove with Force Feedback for Learning from Demonstrations using Probabilistic Trajectory Representations**
Inproceedings
ICRA 2015 Workshop on Tactile and force sensing for autonomous compliant intelligent robots, 2015.
II. Representations of Skills & Imitation Learning

I. Kinematics, Dynamics & Model Learning

IV. Reinforcement Learning & Policy Search

V. Cognitive Reasoning & Planning

III. Feedback, Priorities & Torque Control

VI. Sensor Integration & Fusion
Choose your topic!

I. Bayesian Inference

II. Gaussian Processes for Dynamics Model Learning

III. Kalman and Particle Filter for Inference

IV. Bayesian Optimization for Reinforcement Learning

V. Spiking Neural Networks for Motion Planning

VI. Probabilistic Movement Primitives
Predictive models of rats’ navigation skills

Behavioral Decoding
Predictive models of rats’ navigation skills
Difference btw. Filtering, Smoothing and Predictions

\[ p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1) \prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1}) \]
Difference btw. Filtering, Smoothing and Predictions

\[ p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1) \prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1}) \]

Filtering (Inferring the present) \[p(h_t|v_{1:t})\]
Prediction (Inferring the future) \[p(h_t|v_{1:s})\] \[t > s\]
Smoothing (Inferring the past) \[p(h_t|v_{1:u})\] \[t < u\]
Likelihood \[p(v_{1:T})\]
Most likely Hidden path (Viterbi alignment) \[\arg \max_{h_{1:T}} p(h_{1:T}|v_{1:T})\]
Localizing a burglar.

(a) ‘Creaks’

(b) ‘Bumps’

<table>
<thead>
<tr>
<th>(a) Creaks and Bumps</th>
<th>(b) Filtering</th>
<th>(c) Smoothing</th>
<th>(d) Viterbi</th>
<th>(e) True Burglar position</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Using Smoothing for robot path planning

\[ p(x|r = 1) \] \[ p(x_0) \prod_{t=1}^{T} \mathcal{F}(x_t|x_{t-1}) \]
Smoothing with neural networks

\[
p(x|r = 1) = \frac{1}{\mathcal{L}} p(r|x) p(x_0) \prod_{t=1}^{T} \mathcal{F}(x_t|x_{t-1})
\]

- Cannot be implemented in a Recurrent Neural Network!
- Also the alternative of using 1 Layer per time step is impractical in FF nets.
Smoothing in a RNN through forward sampling from a learned distribution

\[
p(x|r = 1) \propto \frac{1}{\mathcal{L}} p(r|x) p(x_0) \prod_{t=1}^{T} T(x_t|x_{t-1})
\]

- Reward modulated Hebbian Learning
- Supervised Model Learning (CD)
Neural Planning

\[ q(\nu; \theta) = p(\nu_0) \prod_{t=1}^{T} \prod_{k=1}^{K} \rho_{t,k}^{\nu_{t,k}} (1 - \rho_{t,k})^{1-\nu_{t,k}} \]

\[ = p(\nu_0) \prod_{t=1}^{T} \mathcal{T}(\nu_t | \nu_{t-1}) \phi_t(\nu_t; \theta) \]

\[ \mathcal{T}(\nu_t | \nu_{t-1}) = \exp \left( \sum_{i=1}^{K} w_{ki} \nu_{t-1,i} \nu_{t,k} \right) \]

\[ \phi_t(\nu_t; \theta) = \frac{\exp \left( \sum_{j=1}^{N} \theta_{kj} y_{t-1,j} \nu_{t,k} \right)}{\sum_{l=1}^{K} \exp(u_{t,l})} \]
For real robot control without smoothing
Model Learning in 15 Minutes

- training data recorded with kinest
- 15min of movements, sampled at
Real Time Adaptation and Control

The diagram illustrates the concept of real-time adaptation and control in robotics. It shows the progression over time (segment $i$, $i+1$, and $i+2$) with a focus on position and execution time. The key aspects include:

- **Position**: The black line represents the planned path, while the dashed line indicates the executed path.
- **Execution Time**: Shows the timing for each phase of the control cycle.
- **Targets**: Points indicating target positions.
- **Cognitive Dissonance**: Highlighted in red, indicating a potential mismatch or issue in the control process.

The process includes:

- **Planning**
- **Sampling**
- **Post-Processing**
- **Learning**
- **Blending**

These components are crucial for dynamic and adaptive control in robotics applications.
Efficiency evaluation
Factorized population codes for > 2 dimensions
Tanneberg, Daniel; Peters, Jan; Rueckert, Elmar

**Intrinsic Motivation and Mental Replay enable Efficient Online Adaptation in Stochastic Recurrent Networks**
Journal Article

Sosic, Adrian; Rueckert, Elmar; Peters, Jan; Zoubir, Abdelhak M; Koeppl, Heinz

**Inverse Reinforcement Learning via Nonparametric Spatio-Temporal Subgoal Modeling**
Journal Article

Rueckert, Elmar; Kappel, David; Tanneberg, Daniel; Pecevski, Dejan; Peters, Jan

**Recurrent Spiking Networks Solve Planning Tasks**
Journal Article

Rueckert, Elmar; Neumann, Gerhard; Toussaint, Marc; Maass, Wolfgang

**Learned graphical models for probabilistic planning provide a new class of movement primitives**
Journal Article
Frontiers in Computational Neuroscience, 6 (97), 2013.

more at: https://rob.ai-lab.science/publications/
Summary

1. How can humans learn new motor skills within few trials?

Learning probabilistic generative models that capture the correlations of multiple joints/signals.

- For noisy and high dimensional human and robot data.
- Can exploit correlations for predictions.
- Low dimensional feature representation for learning.
- Generative model of stroke-based and rhythmic movements with feedback.
Summary

1. How do humans solve cognitive reasoning tasks in huge spaces?

   Learning stochastic neural networks grounded in the probabilistic inference framework.

   - Simultaneously learning **forward, inverse kinematics** and **state transition models** through kinesthetic teaching.
   - Implements **optimal planning** through reinforcement learning.
   - **Online adaptation** in few seconds from **intrinsic motivation** signals.
   - Model **predictive control** implementation on **real robots**.
- **Darmstadt**: Daniel Tanneberg, Svenja Stark, Gerhard Neumann, Alexandros Paraschos, Roberto Calandra, Jan Peters, Rudolf Lioutikov, Marc Deisenroth, Serena Ivaldi, Tucker Hermans, Philipp Beckerle, Valerio Modugno, Jan Mundo, David Sharma, Jan Kohlschuetter, Svenja Stark, Michael Schmidt, Max Mindt

- **Tübingen**: Moritz Grosse-Wentrup, Martin Giese

- **Ljubljana**: Jan Babic, Jernej Camernik

- **Birmingham**: Michael Mistry, Morteza Azad

- **Graz**: Wolfgang Maass, Robert Legenstein, David Kappel, Dejan Pecevski

**Birmingham**: Jeremy Wyatt, Michael Mistry, Morteza Azad, **Rome**: Andrea d’Avella and Yuri Ivanenko, **Stuttgart**: Marc Toussaint, **Bielefeld**: Thomas Schack, Jochen Stel, **Genua**: Francesco Nori, Lorenzo Natale
Books:


Video Lectures:

- [videolectures.net](http://videolectures.net) on Gaussian Processes, Inference and Reinforcement Learning
- [coursea.org](http://coursea.org) on Robotics

Related lecture notes:

- [Humanoid Robotics](http://humanoidrobotics.com) by Prof. Dr. Maren Bennewitz, *University of Bonn*.
- [Lecture notes on learning methods](http://learningmethods.com) by Prof. Dr. Marc Toussaint, *University Stuttgart*.
- [Lecture notes on dynamics](http://dynamics.com) by Prof. Dr. Russ Tedrake, *Massachusetts Institute of Technology*.
Thank you for your attention!

Contact:

Universität zu Lübeck
Institute for Robotics and Cognitive Systems
Ratzeburger Allee 160
Building 64, Room 94
23538 Lübeck, Deutschland

Telefon: +49 (0) 451 3101 5209
E-Mail: rueckert@rob.uni-luebeck.de

Disclaimer:
The lecture notes posted on this website are for personal use only. The material is intended for educational purposes only. Reproduction of the material for any purposes other than what is intended is prohibited. The content is to be used for educational and non-commercial purposes only and is not to be changed, altered, or used for any commercial endeavor without the express written permission of Professor Rueckert.