UNIVERSITÀ DEGLI STUDI DI PALERMO

Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica

Dottorato di Ricerca in Ingegneria Informatica

BAYESIAN APPROACHES TO HUMAN-ROBOT INTERACTION: FROM LANGUAGE GROUNDING TO ACTION LEARNING AND UNDERSTANDING

SSD: ING-INF/05

TESI DI

ING. DANIELE ZAMBUTO

COORDINATORE DEL DOTTORATO

PROF. SALVATORE GAGLIO

TUTOR

ING. HARIS DINDO

CICLO XXIII

DOTTORATO

.... don't have a cow, man .....

Abstract

In human-robot interaction field, the robot is no longer considered as a tool but as a partner, which supports the work of humans. Environments that feature the interaction and collaboration of humans and robots present a number of challenges involving robot learning and interactive capabilities. In order to operate in these environments, the robot must not only be able to *do*, but also be able to *interact* and especially to *"understand"*.

This thesis proposes a unified probabilistic framework that allows a robot to develop basic cognitive skills essential for collaboration. To this aim we embrace the idea of *motor simulation* - well established in cognitive science and neuroscience - in which the robot reenacts in simulation its own *internal models* used for physically performing action. This particular view offers the possibility to unify apparently distinct cognitive phenomena such as learning, interaction, understanding and dialogue, just to name a few. Ideas presented here are corroborated by experimental results performed both in simulation and on a humanoid robotic platform.

The first contribution in this direction is a robust Bayesian method to estimate (i.e. learn) the parameters of internal models by observing other skilled actors performing goal-directed actions. In addition to deriving a theoretically sound solution for the learning problem, our approach establishes theoretical links between Bayesian inference and gradient-based optimization methods. Using the expectation propagation (EP) algorithm, a similar algorithm is derived for multiple internal models scenario.

Once learned, internal models are reused in simulation to "understand" actions performed by other actors, which is a necessary precondition for successful interaction. We have proposed that action understanding can be cast as an approximate Bayesian inference in which the covert activity of internal models produces hypotheses that are tested in parallel through a sequential Monte Carlo approach. Here, approximate Bayesian inference is offered as a plausible mechanistic implementation of the idea of motor simulation making it feasible in real-time and with limited resources.

Finally, we have investigated how the robot can learn a grounded language model in order to be bootstrapped into communication. Features extracted from the learned internal models, as well as descriptors of various perceptual categories, are fed into a novel multi-instance semi-supervised learning algorithm able to perform semantic clustering and associate words, either nouns or verbs, with their grounded meaning.

Acknowledgments

I would like to express my gratitude to all those who helped me complete this thesis. First and foremost, I would like to offer special thanks to my advisor, Haris Dindo, for not only giving me sage advice but also providing me with the freedom to explore a range of research projects. His constant support and encouragement have pushed me to a place I could not have otherwise reached.

I must thank all my friends with whom I have shared many good times and from whom I have gained strength during difficult times. There are so many of them I wish to thank but hopefully they know who they are. To name just a few: Alfonso Farruggia, Alfonso Galvano, Claudio Messina, Peppe Airo', Vincenzo Schembri.

I want to thank Marilina for all the love, happiness and support she brought into my life. Finally, I would like to thank my mother for her patient love and consistent support. I am forever grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

Human-Robot Interaction (HRI) has recently received great attention in the academic community. The development and evaluation of robotic systems that are able to effectively interact and collaborate with a human have become one of the most studied fields of robotics. The robot is no longer considered as a tool, programmed to perform a predetermined task, but as a partner, which supports the work of humans, cooperate and interact with him and his environment. This shift has brought the robot to move from factories and laboratories, to less structured and more complex environments, populated by humans. In order to operate in these environments, the robot must not only be able to "do", but also be able to "interact" and especially to "understand". Collaboration become necessary to the robot in these environments. But what are the cognitive skills necessary for a robot to cooperate with a human being?

Collaboration means working with someone on something. It aims at reaching a common goal. In order to assist a human in a task, a robot must know his intentions and understand what he is doing. The robot must also know the common goal and understand their environment and how the partner interacts with it. Based on this information the robot can plan an action to support the partner and eventually achieve the common goal. Therefore the robot needs the abilities of perceiving and comprehending its environment, planning and learning.

[Goodrich and Schultz, 2007] gives an overview of the internal mechanisms of a cognitive robot leading to joint actions and collaboration.

> "The environment and the partners are observed by sensors. This sensor data is processed to gain an understanding of the environment

> and provides perception. The perceived data is used firstly to learn and expand the own knowledge, then to gain an understanding of the state of the environment and the partners, and to estimate the intention of the partners. When partners are collaborating, a joint intention is retrieved from the single intentions. A set of actions leading to fulfil the joint intention is found by action planning. At last actions are taken either by single partners or jointly that lead to transitions in the state of the environment. The loop is closed, as the robot observes the actions of itself and of others and the change in the environment."

In human-robot collaboration, the human displays an intention to reach a certain goal. It is the robots task to estimate this intention which may be communicated by the human *explicitly* or *implicitly*. The main ways of communicating one's intention are speech and actions.

Speech is the most natural way of communication for humans. Through words and phrases partners can easily exchange a large amount of complex information and directly inform the other of their intentions and their goals. The same does not apply to a robot. The focus here is on processes that connect symbolic language to the physical world with the ultimate aim of modeling situated language use. This problem is known as symbol grounding problem: in order to actually attribute meaning to language there must be interaction with the world to provide relevance to the symbolic representation. In terms of robotics there is a need to ground actions and visual information with symbolic information provided by language. For instance, the action verb 'grasp' could be grounded in the real-world robot behaviour of reaching and closing the gripper on an object. So the grounding problem is fundamental to achive the development of social robots and to human-robot collaboration.

Actions are another means of communication. Communicative gestures are a form of non-verbal communication in which visible bodily actions communicate particular messages. Pointing gestures, primitive signs, or sign language gestures are examples of communicative gestures. Another kind of gestures are the manipulative gestures, i.e. physical non-verbal communication that does not communicate specific messages. An observer may derive the intention of others partners from watching their actions and behaviour. Though a partner does not necessarily want to communicate through these manipulations or motions, intention can be derived by others nevertheless. Humans are able to discern intentions and goals of other humans by observing them in motion. It has been proposed that actions are recognized via *predictive processes*. In this view, the motor system produce predictions in the service of object and action perception, realizing what has been called *action simulation* or *motor simulation*. This is done by reusing one's own

motor repertoire, and this explains motor involvement during action perception (mirror neurons). Our architecture uses an inverse-forward model coupling in a dual role: either for executing an action, or for perceiving the same action when performed by a partner. In this way the robot is able to perform the actions required to achieve a particular goal and, at the same time, understand the goals of the human partner.

1.2 Scope and contributions

This thesis proposes a unified framework that allows a robot to develop basic cognitive skills essential for collaboration. In order to achieve this, the learning process go through a number of developmental stages. In our view, the cognitive development begins with the process of learning *a sense of bodily self*. The robot must learn to control its own body and the relations between sensors and actions. For instance, to reach for an object, the robot must compute the changes in joint angles of the arm that will bring the hand to the desired position. This mapping is also known as *inverse mapping* or *inverse model*. To learn this transformation, one solution is to learn two models, one for the sensorimotor transformation (the inverse model) and one for predicting the sensory consequences of the motor command (the forward model).

The robot first learns by associating spontaneous motor commands with the sensory consequences of those spontaneous actions(*motor babbling*). On each trial, the robot generates a spontaneous pattern of motor commands and receives sensory feedback of the resulting hand position. At this point, the system can learn to associate the patterns from the motor commands to sensory space. Learning forward model is essentially a problem of model matching. The data collected during motor babbling are used to learn the parameters of a model, which approximates the real one. This approach is known as a generative approach. The problem typically involves solving a system of many coupled variables and is usually formalized as an optimization problem. However we addressed this problems from the point of view of probabilistic inference. In the inference view, variables couplings are *explicitly* formulated as conditional dependencies in a joint distribution and approximated inference methods are used to yield (approximate) solutions to the optimization problem. In this thesis, we present a methodology for learning forward models based on this Bayesian view. The method is demonstrated on feedforward networks.

Using forward model, the robot can observe and infer the state of human partner from its sensor and then perform an action which will have a similar

effect. The robot can learn a repertoire of actions by observing and imitating the partners. This repertoire of actions consists of inverse models. We present a probabilistic method to infer a sequence of motor commands using only the learned forward model. This method of *direct inversion* of the forward model allows the robot to collect examples of actions by observing the partner, which will be used later for learning inverse models.

Inverse and forward models used in motor control can be reused for motor simulation in support of action prediction and understanding. The same internal models that the robot uses for performing goal-directed action can be re-enacted "in simulation" and used for inferring others' actions (and possibly imitating them). Since inverse models have associated goals, recognition of inverse models entails recognition of the (more plausible) action goals. However, it is unlikely and impractical that all models are maintained in parallel for the entire period of recognition. In ambiguous situations, where the environment does not provide sufficient information for recognition, there might be hundreds or thousands of internal models making the problem of action recognition intractable with scarce resources. However, casting the problem of action understanding in a Bayesian framework permits to adopt efficient techniques for *approximate* probabilistic inference under the constraint of limited resources. We present an action understanding framework, based on *particle filters*, a Monte Carlo technique for sequential simulation. Our framework permits to easily embed a huge number of internal models by exploiting all the prior knowledge available to the agent and to limit the activation of internal models.

Finally, we have investigated the lexical acquisition problem, particularly how a robot can be bootstrapped into communication and what are the necessary prerequisites for robots in order to learn a language. In particular, we focused on grounded systems that learn to generate and understand contextualized spoken descriptions of observed objects and actions in a visual scenes. Our goal is to take advantage of acquired concepts (representing physical properties and spatial relation of objects, and actions naming), and language model (encoding all syntactic and semantic constraints) to engage in simple verbal interaction with human partner.

This thesis proposes the following:

- A robust Bayesian method to estimate the parameters of internal models. This approach establishes theoretical links between Bayesian inference of model parameters and gradient-based optimization methods. The method is demonstrated on feedforward networks.

- An expectation propagation (EP) algorithm to estimate the parameters of multiple internal models.

- Efficient processing of multiple forward-inverse models achieved by using prior information over possible contexts and goal-directed actions, and by adopting an approximate inference procedure (sequential Monte Carlo simulation) for tracking several competing hypotheses.
- A multi-instance semi-supervised learning algorthm to resolve the semantic clustering.
- An expectation maximization (EM) algorithm to resolve word-to-meaning association and synthactic bootstrapping.

1.3 Dissertation Outline

The remainder of the dissertation is organized as follows. Chapter 2 describes a Bayesian view on two classical motor learning algorithm: Distal learning and MOSAIC. We show how these problems, usually formalized as optimization problems, can be solved using the Bayesian approach. Chapter 3 describe a generative Bayesian model for action understanding, in which inverse-forward model pairs are considered 'hypotheses' of plausible action goals that are explored in parallel via an approximate inference mechanism based on particle filtering. Chapter 4 described a multi-instance learning algorithm to resolve the semantic clustering and an EM algorithm to learn a grounded language model.

Chapter 2

Bayesian Motor Learning

2.1 Introduction

Motor learning is fundamental and very interesting problem in robotics. There have been many attempts at creating learning frameworks, enabling robots to autonomously learn complex skills ranging from task imitation to motor control. However, learning is not an easy task. The problem typically involves solving a system of many coupled variables and is usually formalized as an optimization problem, where these couplings *are implicit* in a cost function. However the problem can be addressed from the point of view of probabilistic inference. In the inference view, variables couplings are *explicitly* formulated as conditional dependencies in a joint distribution. Inference methods like message passing algorithms can be used to yield (approximate) solutions to the optimization problem. These principles have been introduced by Toussaint in motion control and planning problem. Our goal is to apply the same principles to the motor learning problem. The following section present a Bayesian view on motor learning problem and a general approach to solving it. We begin by describing our assumption about the controlled plant and the learner.

2.2 Motor learning problem

We assume that the plant can be characterized by a next-state function f and an output function g. At time step $t-1$ the learner produces a control $\mathbf{u}_{t-1}$ (for simplicity we will consider a discrete time system). The resulting next state $\mathbf{x}_t$ is determined by Eq. 2.1, which describes the causal relationship between the command and the state of the plant.

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \tag{2.1}$$

The function f describes the plant *dynamics*. Corresponding to each state $\mathbf{x}_t$ there is an observed output $\mathbf{y}_t$:

$$\mathbf{y}_t = g(\mathbf{x}_t) \tag{2.2}$$

The dynamics function and the output function together determine a state-dependent mapping from control to output. The goal of the learning procedure is to produce a system which can generate an appropriate control $\mathbf{u}_{t-1}$ given a desired output $\mathbf{y}_t^*$ or a generic *input* $\mathbf{p}_{t-1}$.

$$\mathbf{u}_{t-1} = h(\mathbf{y}_t^*, \mathbf{x}_{t-1}) \tag{2.3}$$

The learner will make appropriate adjustments to the input-to-control mapping h based on data obtained from interacting with the controlled plant. In the current work we assume that the learner is able to observe only controls and outputs of the system. The internal state must be induced as part of the learning procedure. To simplify the problem, we assume that the output function g is known, although an complete formulation could be considered.

2.2.1 Forward model

A forward model is an internal model that produces a predicted state $\hat{\mathbf{x}}_t$ based on the state $\mathbf{x}_{t-1}$ and the command $\mathbf{u}_{t-1}$:

$$\hat{\mathbf{x}}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) \tag{2.4}$$

The forward model directly corresponds to the state dynamics function f shown in Eq. 2.1. As this function expresses the physical properties of the system, the forward model represents a causal relationship between states and commands. Thus, if such causal mappings have to be learned, it will result in a well-defined problem and learning can be done using classical supervised algorithm. By comparing predicted output $\hat{\mathbf{y}}$ and actual output $\mathbf{y}^*$, the learner could use the resulting prediction error $\mathbf{y}^* - \hat{\mathbf{y}}$ to adjust the parameters of the model $\mathbf{w}$.

2.2.2 Inverse model

An inverse model is an internal model that produces a command $\mathbf{u}_{t-1}$ as a function of the current state $\mathbf{x}_{t-1}$ and the desired future output $\hat{\mathbf{y}}_t$:

$$\mathbf{u}_{t-1} = h(\mathbf{y}_t^*, \mathbf{x}_{t-1}, \mathbf{v}) \tag{2.5}$$

so that h and f have an inverse relationship and yield an indentity mapping when placed in series. Whereas forward models are uniquely determined,

inverse models are generally not. Classicaly, physical systems are characterized by a many-to-one mapping from commands to outputs. So there are an infinite number of possible inverse model. A typical example is the inverse kinematics of redundant robots.

Inverse kinematics Given a joint configuration $\mathbf{q}$, the task space position $\mathbf{x}$ can be determined exactly, but there may be many possible joint configurations $\mathbf{q}$ for a given task space position $\mathbf{x}$. Due to the many-to-one relationship, the position-to-join mapping is ill-posed and $\mathbf{q}$ may form a non-convex solution space. Thus, when naively learning such inverse mapping from data, the learning algorithm will potentially average over non-convex sets of the solutions. The resulting mapping will contain invalid solutions which can cause poor prediction performance.

We describe an indirect approach to motor learning known as *distal supervised learning*. Distal supervised learning avoids the nonconvexity problem and also avoids certain other problems associated with direct approaches to motor learning.

2.3 A Bayesian view on distal learning

The basic idea behind the distal supervised approach ([Jordan and Rumelhart, 1992]) is that the information encoded in the forward model can help to resolve the non-uniqueness of the inverse model. The forward model is understood as a "distal teacher" which guides the learning of the inverse model. The inverse and the forward model are joined together and are treated as a single *composite learning system*. The intuition behind this approach is that the inverse model will learn a correct solution for a particular desired trajectory when minimizing the error between the predicted output of the forward model $\hat{\mathbf{y}}$ and the desired future output $\mathbf{y}^*$. Thus, this approach is fundamentally goal-directed.

In practice, the distal teacher employs two interacting learning process: one process where the forward model is learned from a set of training pairs $\{\mathbf{u}_{t-1}, \mathbf{y}_t\}$, and another process where the learned forward model is used for determining the control sequence $\{\mathbf{u}_t^*\}_{t=1}^T$, given the desired output sequence $\{\mathbf{y}_t^*\}_{t=1}^T$. Both problems can be solved easily using a Bayesian approach.

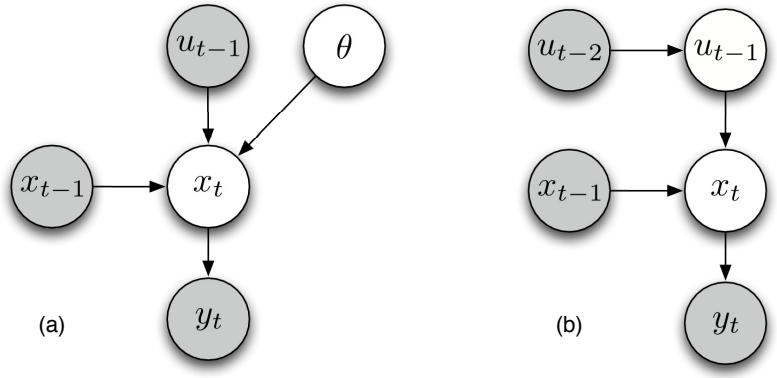


Figure 2.1: Graphical models for Bayesian distal learning: (a) for parameters learning and (b) for motion rate control

2.3.1 Forward model: single time-slice learning

In this section, we first address the case of single-step learning. This example will help us to introduce some of the issues we face in this work, and has no practical relevance. Throughout the derivation we will make use of identities for Gaussians which are summarized in the appendix.

First we need to estimate the parameter $\mathbf{w} \in \mathbb{R}^s$ of the forward model from a single training pair $\{\mathbf{u}_{t-1}, \mathbf{y}_t\}$. Consider the join probability distribution:

$$P(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{u}_{t-1}, \mathbf{y}_t, \mathbf{w}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w})P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t)P(\mathbf{w}) \quad (2.6)$$

as also illustrated by the graphical model in Figure 2.1. We recognize the first distribution as the forward model, and assume

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) = \mathcal{N}(\mathbf{x}_t|f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}), \mathbf{C}) \quad (2.7)$$

As we said, f is a non-linear function and $\mathbf{C}$ is the covariance of this coupling (or equivalently $\mathbf{C}^{-1}$ is the precision of the forward model). The second distribution in Eq. 2.6 is the output model:

$$P(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t|g(\mathbf{x}_t), \mathbf{Q}) \quad (2.8)$$

where g is the non-linear output function and $\mathbf{Q}$ is the associated covariance. Finally, we call $P(\mathbf{w})$ the *parameter prior* and assume

$$P(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, \mathbf{H}^{-1}) \quad (2.9)$$

where $\hat{\mathbf{w}}$ is a prior solution and $\mathbf{H}$ is, as we will show later, a *learning rate* matrix.

Given this model we can compute the posterior parameter conditioned on the observed output $\mathbf{y}_t$, the current state $\mathbf{x}_{t-1}$ and the control $\mathbf{u}_{t-1}$. We have

$$\begin{aligned} & P(\mathbf{w}|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{y}_t) \\ &= \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) P(\mathbf{y}_t|\mathbf{x}_t) P(\mathbf{w}) d\mathbf{x}_t \\ &= \int \mathcal{N}(\mathbf{x}_t|f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}), \mathbf{C}) \mathcal{N}(\mathbf{y}_t|g(\mathbf{x}_t), \mathbf{Q}) \mathcal{N}(\mathbf{w}|\hat{\mathbf{w}}, \mathbf{H}^{-1}) d\mathbf{x}_t \end{aligned}$$

We need to linearize the functions f and g in order to proceed. We use the following linearization:

$$\begin{aligned} f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) &\approx f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \hat{\mathbf{w}}) + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}}) \\ &= \hat{\mathbf{x}}_t + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}}) \\ g(\mathbf{x}_t) &\approx g(\mathbf{x}_{t-1}) + \mathbf{J}_g(\mathbf{x}_t - \mathbf{x}_{t-1}) = \mathbf{J}_g \mathbf{x}_t + g(\mathbf{x}_{t-1}) - \mathbf{J}_g \mathbf{x}_{t-1} \\ &= \mathbf{J}_g \mathbf{x}_t + \tilde{\mathbf{y}}_t \end{aligned}$$

where $\hat{\mathbf{x}}_t$ is the predicted state and $\tilde{\mathbf{y}}_t$ is a partial approximation of the output at time t. The Jacobian matrix $\mathbf{J}_f$ and $\mathbf{J}_g$ are defined as:

$$\mathbf{J}_f = \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \hat{\mathbf{w}}} \qquad \mathbf{J}_g = \left.\frac{\partial g}{\partial \mathbf{x}_t}\right|_{\mathbf{x}_{t-1}}$$

Applying the Gaussian identities given in the appendix we have

$$\begin{aligned} & P(\mathbf{w}|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{y}_t) \\ &\approx \int \mathcal{N}\left(\mathbf{x}_t|\hat{\mathbf{x}}_t + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}}), \mathbf{C}\right) \mathcal{N}\left(\mathbf{y}_t|\mathbf{J}_g \mathbf{x}_t + \tilde{\mathbf{y}}_t, \mathbf{Q}\right) \mathcal{N}\left(\mathbf{w}|\hat{\mathbf{w}}, \mathbf{H}^{-1}\right) d\mathbf{x}_t \\ &= \int \mathcal{N}\left[\mathbf{x}_t|\mathbf{C}^{-1}\left(\hat{\mathbf{x}}_t + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}})\right), \mathbf{C}^{-1}\right] \mathcal{N}\left[\mathbf{x}_t|\mathbf{J}_g^T \mathbf{Q}^{-1}(\mathbf{y}_t - \tilde{\mathbf{y}}_t), \mathbf{J}_g \mathbf{Q}^{-1} \mathbf{J}_g^T\right] \\ &\qquad \cdot \mathcal{N}\left(\mathbf{w}|\hat{\mathbf{w}}, \mathbf{H}^{-1}\right) d\mathbf{x}_t \end{aligned}$$

Applying the identities A.5 produces a Gaussian over $\mathbf{x}_t$ which integrates to 1. Using the short hand $\mathbf{A} = \mathbf{J}_g \mathbf{C} \mathbf{J}_g^T + \mathbf{Q}$ we get:

$$\begin{aligned} & P(\mathbf{w} \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{y}_t) \\ &= \mathcal{N}\left(\mathbf{J}_g \left(\hat{\mathbf{x}}_t + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}})\right) \mid \mathbf{y}_t - \tilde{\mathbf{y}}_t, \mathbf{A}\right) \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}}, \mathbf{H}^{-1}\right) \\ &= \mathcal{N}\left[\mathbf{J}_g \left(\hat{\mathbf{x}}_t + \mathbf{J}_f (\mathbf{w} - \hat{\mathbf{w}})\right) \mid \mathbf{A}^{-1}(\mathbf{y}_t - \tilde{\mathbf{y}}_t), \mathbf{A}^{-1}\right] \mathcal{N}\left[\mathbf{w} \mid \mathbf{H}\hat{\mathbf{w}}, \mathbf{H}\right] \\ &= \mathcal{N}\left[\mathbf{w} \mid \mathbf{J}_f^T \mathbf{J}_g^T \mathbf{A}^{-1}(\mathbf{y}_t - \tilde{\mathbf{y}}_t - \mathbf{J}_g \hat{\mathbf{x}}_t + \mathbf{J}_f \mathbf{J}_g \hat{\mathbf{w}}), \mathbf{J}_f^T \mathbf{J}_g^T \mathbf{A}^{-1} \mathbf{J}_g \mathbf{J}_f\right] \mathcal{N}\left[\mathbf{w} \mid \mathbf{H}\hat{\mathbf{w}}, \mathbf{H}\right] \end{aligned}$$

Using the short hand $\mathbf{J} = \mathbf{J}_g\mathbf{J}_f$ and noting that

$$\tilde{\mathbf{y}}_t + \mathbf{J}_g\hat{\mathbf{x}}_t = g(\mathbf{x}_{t-1}) + \mathbf{J}_g\left(\hat{\mathbf{x}}_t - \mathbf{x}_{t-1}\right) = \hat{\mathbf{y}}_t$$

we find the final solution

$$\begin{aligned}
&P(\mathbf{w} \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{y}_t) \\
&= \mathcal{N}\left[\mathbf{w} \mid \mathbf{J}^T\mathbf{A}^{-1}\mathbf{J}\hat{\mathbf{w}} + \mathbf{J}^T\mathbf{A}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right), \mathbf{J}^T\mathbf{A}^{-1}\mathbf{J}\right] \mathcal{N}\left[\mathbf{w} \mid \mathbf{H}\hat{\mathbf{w}}, \mathbf{H}\right] \\
&= \mathcal{N}\left[\mathbf{w} \mid \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)\hat{\mathbf{w}} + \mathbf{J}^T\mathbf{A}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right), \mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right] \\
&= \mathcal{N}\left(\mathbf{w} \mid \hat{\mathbf{w}} + \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\mathbf{J}^T\mathbf{A}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right), \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\right)
\end{aligned}$$

Thus, Bayesian inference produces the maximum a posteriori (MAP) estimate:

$$\mathbf{w}_{MAP} = \hat{\mathbf{w}} + \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\left(\mathbf{J}^T\mathbf{A}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right)\right) \tag{2.10}$$

We can use Woodbury identity:

$$\left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\mathbf{J}^T\mathbf{A}^{-1} = \mathbf{H}^{-1}\mathbf{J}^T\left(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^T + \mathbf{A}\right)^{-1} = \mathbf{J}^{\#} \tag{2.11}$$

and rewrite $\mathbf{w}_{MAP}$ as follows

$$\mathbf{w}_{MAP} = \hat{\mathbf{w}} + \mathbf{J}^{\#}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right) \tag{2.12}$$

Using a different linearization of the function f

$$f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) \approx \hat{\mathbf{x}}_t + \mathbf{J}_f\mathbf{w}$$

we get a slightly different solution

$$\begin{aligned}
\mathbf{w}_{MAP} &= \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\left(\mathbf{H}\hat{\mathbf{w}} + \mathbf{J}^T\mathbf{A}^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right)\right) \\
&= \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\mathbf{H}\hat{\mathbf{w}} + \mathbf{J}^{\#}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right)
\end{aligned}$$

Further, using the identity

$$\left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\mathbf{H} = \mathbf{I} - \left(\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} + \mathbf{H}\right)^{-1}\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J} = \mathbf{I} - \mathbf{J}^{\#}\mathbf{J} \tag{2.13}$$

we get

$$\mathbf{w}_{MAP} = \left(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}\right)\hat{\mathbf{w}} + \mathbf{J}^{\#}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right) \tag{2.14}$$

2.3.2 Inverse model: single time-slice learning

By changing some of the assumptions made above and using the same Bayesian network, we can estimate the control $\mathbf{u}$. We will prove that, given the forward model, we can always find the optimal control $\mathbf{u}^*$ to get the desired output $\mathbf{y}^*$. We have no need to use a parametric function to model the controller and to explicitly formulate the conditional dependencies of the variables involved in the forward model and the inverse model.

Consider the join probability distribution:

$$P(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{u}_{t-1}, \mathbf{u}_{t-2}, \mathbf{y}_t^*, \mathbf{w}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w})P(\mathbf{y}_t^*|\mathbf{x}_t)P(\mathbf{u}_{t-1} \,|\, \mathbf{u}_{t-2})$$

Here, we call $P(\mathbf{u}_{t-1} \,|\, \mathbf{u}_{t-2})$ the *control prior* and assume

$$P(\mathbf{u}_{t-1} \,|\, \mathbf{u}_{t-2}) = \mathcal{N}\left(\mathbf{u}_{t-1} \,|\, \mathbf{u}_{t-2} + \mathbf{k}, \mathbf{K}\right) \tag{2.15}$$

where $\mathbf{k}$ is a vector that induces an asymmetry in the control prior and $\mathbf{K}$ is the control metric. Given this model we can compute the posterior control conditioned on a desired output

$$P(\mathbf{u}_{t-1} \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-2}, \mathbf{y}_t^*, \mathbf{w}) = \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w})P(\mathbf{y}_t^*|\mathbf{x}_t)P(\mathbf{u}_{t-1} \,|\, \mathbf{u}_{t-2})d\mathbf{x}_t$$

We use the following linearization:

$$\begin{aligned} f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) &\approx f(\mathbf{x}_{t-1}, \mathbf{u}_{t-2}, \mathbf{w}) + \mathbf{J}_u \left(\mathbf{u}_{t-1} - \mathbf{u}_{t-2}\right) && (2.16)\\ &= \hat{\mathbf{x}}_t + \mathbf{J}_u \left(\mathbf{u}_{t-1} - \mathbf{u}_{t-2}\right) && (2.17)\\ g(\mathbf{x}_t) &\approx \mathbf{J}_g \mathbf{x}_t + \tilde{\mathbf{y}}_t && (2.18) \end{aligned}$$

The posterior control is

$$P(\mathbf{u}_{t-1} \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-2}, \mathbf{y}_t^*, \mathbf{w}) = \mathcal{N}\left(\mathbf{u}_{t-1} \,|\, \mathbf{u}^{MAP}, \mathbf{U}\right)$$

with MAP solution

$$\begin{aligned} \mathbf{u}^{MAP} &= \mathbf{u}_{t-2} + \mathbf{J}^{\#}(\mathbf{y}_t^* - \hat{\mathbf{y}}_t) + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\mathbf{k} && (2.19)\\ \mathbf{J}^{\#} &= \mathbf{K}^{-1}\mathbf{J}^T \left(\mathbf{J}\mathbf{K}^{-1}\mathbf{J}^T + \mathbf{A}\right) && (2.20)\\ \mathbf{A} &= \mathbf{J}_g \mathbf{Q} \mathbf{J}_g^T + \mathbf{C} && (2.21)\\ \mathbf{J} &= \mathbf{J}_g \mathbf{J}_u && (2.22) \end{aligned}$$

The solution presented here is obtained from the work of Toussaint and his framework, called AICO. For more details we recommend [Toussaint and Goerick, 2010].

2.3.3 Comparison with gradient-based optimization algorithm

In this section we compare the results obtained above with some gradient-based optimization algorithm. We show how Bayesian machinery led to comparable results and easily resolved some issues. We simplify some of the assumptions made above, namely, we consider the output variable $\mathbf{y}$ as a direct noised observation of the state of the system. The equation 2.12 can be semplified as follows

$$\mathbf{w}_{MAP} = \hat{\mathbf{w}} + \left(\mathbf{J}_f^T(\mathbf{C}+\mathbf{Q})^{-1}\mathbf{J}_f + \mathbf{H}\right)^{-1}\mathbf{J}_f^T(\mathbf{C}+\mathbf{Q})^{-1}\left(\mathbf{y}_t - \hat{\mathbf{y}}_t\right) \tag{2.23}$$

The above equation shows that the MAP parameter solution is very similar to classical gradient-based solution.

In the classical limit $\mathbf{C} \to 0$, $\mathbf{Q} \to 0$ and $\mathbf{H} \to 0$ (tight constraint, zero observation noise and zero learning rate) the equation gives a step closer to the gradient descent direction. Special choices of the learning rate matrix $\mathbf{H}$ and covariance matrix $\mathbf{C}$ correspond to special classical optimization strategies. For instance, Gauss-Newton algorithm follows from choosing $\mathbf{C} = \mathbf{I}$ and $\mathbf{H} = 0$, and Levenberg algorithm follows from choosing $\mathbf{H} = \lambda\mathbf{I}$.

Assuming $\mathbf{C} = \mathbf{I}$ and zero observation noise, we derive the covariance matrix associated with the MAP solution as follows

$$\mathbf{H}_{MAP} = \mathbf{J}_f^T\mathbf{J}_f + \mathbf{H} \tag{2.24}$$

$$\mathbf{H}_{MAP}^{-1} = \mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{J}_f^T(\mathbf{I} - \mathbf{J}\mathbf{H}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{H}^{-1} \tag{2.25}$$

These equations corresponds to the incremental method of obtaining Fisher information matrix. Then, updating the covariance associated with MAP solution means estimating the Fisher Information matrix at $\hat{\mathbf{w}}$. The gradient $H_{MAP}^{-1}J_f$ is called natural gradient in Riemann space, and the equations suggest the Amari's natural gradient descent algorithm.

2.3.4 Bayesian distal learning

Now, we try to put it all together. Suppose we want to achieve a particular output $\mathbf{y}^*$. We do not know the true dynamics of the system f, but we have only an approximation, given by the forward model f_w. Equation 2.19 allows us to calculate, based on the (incorrect) parameters of the forward model, the control $\mathbf{u}$. When we execute the control, we observe different output, $\mathbf{y}$. We use the pair $(\mathbf{u}, \mathbf{y})$ to reestimate the parameters of the forward model $\mathbf{w}$.

2.4 Learning along trajectory

Here, we move from single time slice model to time extended models of the whole trajectory. The inference techniques in such temporal models are effectively described in terms of message passing algorithms. In most realistic cases exact inference is infeasible because the shape of the exact probability distributions would be very complex. Approximate inference method, like Belief Propagation, can be used to yield (approximate) solutions to these problems. A more detailed description of message passing in general factor graphs is given in [Kschischang et al., 2001]. Here we only give the message equations in general factor graph.

2.4.1 Message passing algorithm

We consider the problem of finding the marginal $p(x)$ for particular variable node x. By definition, the marginal is obtained by integrating the joint distribution over all variables except x so that

$$p(x) = \int_{\mathbf{x}\backslash x} p(\mathbf{x}) \tag{2.26}$$

where $\mathbf{x}\backslash x$ denotes the set of variables in $\mathbf{x}$ with variable x omitted. The idea is to substitute for $p(\mathbf{x})$ using a factor graph and then interchange summations and products in order to obtain an efficient algorithm. We see that the joint distribution can be written as a product of the form

$$p(\mathbf{x}) = \prod_i f_i(\mathbf{x}_i) \tag{2.27}$$

where $\mathbf{x}_i$ denotes the set of variable nodes that are adiacent of factor f_i. In factor graphs, the two key equations are the messages sent from a variable x to a local factor f_i:

$$\mu_{x\to f_i}(x) = \prod_{f_j\in N(x)\backslash f_i} \mu_{f_j\to x}(x) \tag{2.28}$$

and from a factor to a variable

$$\mu_{f_i\to x}(x) = \int_{\mathbf{x}_i\backslash x} f(\mathbf{x}_i) \prod_{x_j\in\mathbf{x}_i\backslash x} \mu_{x_j\to f_i}(x_j) \tag{2.29}$$

where $\mathbf{x}_i$ is the domain of f_i and $N(x)$ are the factors that depends on x. Finally the marginal distribution can be written as

$$p(x) = \prod_{f_i\in N(x)} \mu_{f_i\to x}(x) \tag{2.30}$$

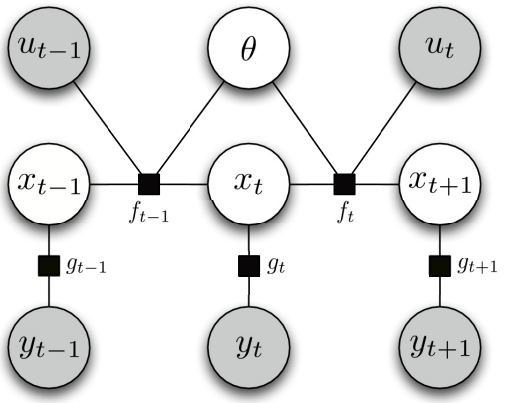


Figure 2.2: The same graphical model as for redundant distal learning (Figure 2.1(a)) but for multiple time slices.

We can represent these messages as vector of discrete probabilities, or as the mean and the covariance of a Gaussian random variable, etc.

The equation 2.28 says that to evaluate the message sent by a variable node to a factor node, take the product of the incoming messages along all other links coming into the variable node. In this case, the product of the message can be implemented by dividing the marginal by the message to be excluded

$$\mu_{x \to f_i}(x) = \frac{p(x)}{\mu_{f_i \to x}(x)} \tag{2.31}$$

For Gaussian messages, we can define division as in the junction tree algorithm by using the canonical parmetrization, i.e., using precision matrix. The equation 2.29 says that to evaluate the message sent by a factor node to a variable node along the link connecting them, take the product of the incoming messages along all other links coming into the factor node, multiply by the factor associated with that node, and then marginalize over all of the variables associated with the incoming messages.

2.4.2 Forward model

Now, we derive the messages for multi-step learning. We will derive only some of the messages of the Bayesian network. The remaining messages can be derived in the same way. We consider the joint probability distribution

$$P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}, \mathbf{y}_{1:T}, \mathbf{w}) = \left[\prod_{t=1}^{T} P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w})\, P(\mathbf{y}_t \,|\, \mathbf{x}_t)\right] P(\mathbf{x}_0) P(\mathbf{w})$$

Here, we use a different linearization of f and g functions

$$\begin{aligned} f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) &\approx f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, \hat{\mathbf{w}}) + \mathbf{J}_x \left(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}\right) + \mathbf{J}_w \left(\mathbf{w} - \hat{\mathbf{w}}\right) \\ &= \tilde{\mathbf{x}}_t + \mathbf{J}_x \left(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}\right) + \mathbf{J}_w \left(\mathbf{w} - \hat{\mathbf{w}}\right) \\ g(\mathbf{x}_t) &\approx g(\hat{\mathbf{x}}_t) + \mathbf{J}_g(\mathbf{x}_t - \hat{\mathbf{x}}_t) = \mathbf{J}_g \mathbf{x}_t + g(\hat{\mathbf{x}}_t) - \mathbf{J}_g \hat{\mathbf{x}}_t \\ &= \mathbf{J}_g \mathbf{x}_t + \tilde{\mathbf{y}}_t \end{aligned}$$

We begin by deriving the message sent from factor g_t to state $\mathbf{x}_t$. We have

$$\begin{aligned} \mu_{g_t \to x_t}(\mathbf{x}_t) &= \mathcal{N}\left(\mathbf{y}_t \,|\, \mathbf{J}_g \mathbf{x}_t + \tilde{\mathbf{y}}_t, \mathbf{Q}\right) \\ &= \mathcal{N}\left[\mathbf{x}_t \,|\, \mathbf{J}_g \mathbf{Q}^{-1}\left(\mathbf{y}_t - \tilde{\mathbf{y}}_t\right), \mathbf{J}_g^T \mathbf{Q}^{-1} \mathbf{J}_g\right] \end{aligned} \tag{2.32}$$

We also define

$$\begin{aligned} \mu_{x_t \to f_t}(\mathbf{x}_t) &= \mu_{g \to x_t}(\mathbf{x}_t)\, \mu_{f_{t-1} \to x_t}(\mathbf{x}_t) \\ &= \mathcal{N}\left[\mathbf{x}_t \,|\, \mathbf{J}_g^T \mathbf{Q}^{-1}\left(\mathbf{y}_t - \tilde{\mathbf{y}}_t\right), \mathbf{J}_g^T \mathbf{Q}^{-1} \mathbf{J}_g\right] \mathcal{N}\left(\mathbf{x}_t \,|\, \mathbf{f}_t, \mathbf{F}_t\right) \\ &= \mathcal{N}\left(\mathbf{x}_t \,|\, \mathbf{g}_t, \mathbf{G}_t\right) \end{aligned}$$

where

$$\mathbf{g}_t = \mathbf{f}_t + \mathbf{J}_g^{\#}\left(\mathbf{y}_t - \hat{\mathbf{y}}_{\triangleright}\right) \tag{2.33}$$

$$\mathbf{G}_t = \left(\mathbf{J}_g^T \mathbf{Q}^{-1} \mathbf{J}_g + \mathbf{F}_t^{-1}\right)^{-1} \tag{2.34}$$

$$\mathbf{J}_g^{\#} = \mathbf{J}_g^T \mathbf{F}_t \left(\mathbf{J}_g \mathbf{F}_t \mathbf{J}_g^T + \mathbf{Q}\right)^{-1} \tag{2.35}$$

$$\hat{\mathbf{y}}_{\triangleright} = \tilde{\mathbf{y}}_t + \mathbf{J}_g \mathbf{f}_t = g(\hat{\mathbf{x}}_t) + \mathbf{J}_g(\mathbf{f}_t - \hat{\mathbf{x}}_t) \tag{2.36}$$

The message $\mu_{x_t \to f_{t-1}}$ can be derived in exactly the same way. The messages $\mu_{f_t \to x_t}$ and $\mu_{f_t \to x_{t-1}}$ are analogous to forward and backward message of extended Kalman smoothing. These messages correspond directly to the prediction made by the forward model. Concerning the message $\mu_{f_t \to w}$, we have

$$\begin{aligned} &\mu_{f_t \to w}(\mathbf{w}) \\ &= \int P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) \mu_{x_{t-1} \to f_t}(\mathbf{x}_{t-1}) \mu_{x_t \to f_t}(\mathbf{x}_t) d\mathbf{x}_{t-1} d\mathbf{x}_t \\ &= \int P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}) \mathcal{N}\left(\mathbf{x}_{t-1} \,|\, \mathbf{g}_{t-1}, \mathbf{G}_{t-1}\right) \mathcal{N}\left(\mathbf{x}_t \,|\, \mathbf{b}_t, \mathbf{B}_t\right) d\mathbf{x}_{t-1} d\mathbf{x}_t \\ &= \mathcal{N}[\mathbf{w} \,|\, \mathbf{s}_t, \mathbf{S}_t] \end{aligned} \tag{2.37}$$

where

$$\mathbf{s}_t = \mathbf{S}_t \hat{\mathbf{w}} + \mathbf{J}_w^T \mathbf{A}_t^{-1}(\mathbf{b}_t - \tilde{\mathbf{x}}_t - \mathbf{J}_x(\mathbf{g}_{t-1} - \hat{\mathbf{x}}_{t-1})) \tag{2.38}$$

$$\mathbf{S}_t = \mathbf{J}_w^T \mathbf{A}_t^{-1} \mathbf{J}_w \tag{2.39}$$

$$\mathbf{A}_t = \mathbf{C} + \mathbf{B}_t + \mathbf{J}_x \mathbf{G}_{t-1} \mathbf{J}_x^T \tag{2.40}$$

Equation 2.38 represents a partial approximation of the parameters $\mathbf{w}$ based on the contribution of a single factor f_t. It can be rewritten as follows

$$\begin{aligned}\mathbf{s}_t &= \hat{\mathbf{w}} + \left(\mathbf{J}_w^T \mathbf{A}_t^{-1} \mathbf{J}_w\right)^{-1} \mathbf{J}_w^T \mathbf{A}_t^{-1} (\mathbf{b}_t - \tilde{\mathbf{x}}_t - \mathbf{J}_x(\mathbf{g}_{t-1} - \hat{\mathbf{x}}_{t-1})) \\ &= \hat{\mathbf{w}} + \mathbf{J}_w^{\#}(\mathbf{b}_t - \hat{\mathbf{x}}_{\triangleright}) = \hat{\mathbf{w}} + \mathbf{J}_w^{\#}(\hat{\mathbf{x}}_{\triangleleft} - \hat{\mathbf{x}}_{\triangleright}) \end{aligned} \tag{2.41}$$

The variable $\hat{\mathbf{x}}_{\triangleright}$ and $\hat{\mathbf{x}}_{\triangleleft}$ are respectively the forward and backward approximation of the state $\mathbf{x}_t$. The error in the state space is mapped in the parameter space through $\mathbf{J}_w^{\#}$ (pseudo-inverse of the jacobian $\mathbf{J}_w$) and used to correct the current solution. In most cases, the matrix $\mathbf{J}_w^{\#}$ can not be calculated because $\mathbf{J}_w^T \mathbf{A}_t^{-1} \mathbf{J}_w$ is singular and the inverse does not exist.

All partial solutions are then used to calculate the belief on the parameters $b(\mathbf{w})$ as follows

$$b(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \hat{\mathbf{w}}^{new}, \hat{\mathbf{W}}) \tag{2.42}$$

$$\hat{\mathbf{w}}^{new} = \hat{\mathbf{w}}^{old} + \sum_t \mathbf{J}_H^{\#}(\hat{\mathbf{x}}_{\triangleleft} - \hat{\mathbf{x}}_{\triangleright}) \tag{2.43}$$

$$\mathbf{J}_H^{\#} = \hat{\mathbf{W}} \mathbf{J}_w^T \mathbf{A}_t^{-1} \tag{2.44}$$

$$\hat{\mathbf{W}} = \left(\sum_t \mathbf{J}_w^T \mathbf{A}_t^{-1} \mathbf{J}_w + \mathbf{H}\right)^{-1} \tag{2.45}$$

The matrix $\mathbf{J}_H^{\#}$ can be interpreted either as a normalization costant that weights each contribution, or as the so-called singularity-robust inverse, where the learning rate matrix $\mathbf{H}$ acts as a regularizer.

2.4.3 Algorithm summary

A standard algorithm would resolve the recursive equations by first iterating forward over time to compute the forward messages, and then iterate backward over time to compute the backward messages. The algorithm we propose is given by the following message passing scheme:

1. Initialize all beliefs uniformly, except for $\mathbf{x}_0$.

2. Update the state space beliefs

$$b(\mathbf{x}_t) = \mu_{g_t \to x_t} \mu_{f_{t-1} \to x_t} \mu_{f_t \to x_t} \tag{2.46}$$

   first iterating forward for $t = 1, .., T$, then iterating backward for $t = T-1, .., 1$. We use local linearization at $\hat{\mathbf{w}} = b(\mathbf{w})$ and $\hat{\mathbf{x}}_t = b^{old}(\mathbf{x}_t)$. This will yield a preliminary belief over possible state.

3. Update the parameter space beliefs

$$b(\mathbf{w}) = P(\mathbf{w}) \prod_t \mu_{f_t \to w} \tag{2.47}$$

using local linearizations at $\hat{\mathbf{w}} = b^{old}(\mathbf{w})$ and $\hat{\mathbf{x}}_t = b(\mathbf{x}_t)$. This generates a new estimate of forward model parameter.

4. Iterate until convergence.

Since the messages depend on the point of linearization we have to iterate the forward and backward sweeps until convergence. We can make a direct comparison with the EM algorithm. The step (1) corresponds to the E-step, which estimates the hidden variables of the model and the step (2) corresponds instead to the M step which computes the new parameters.

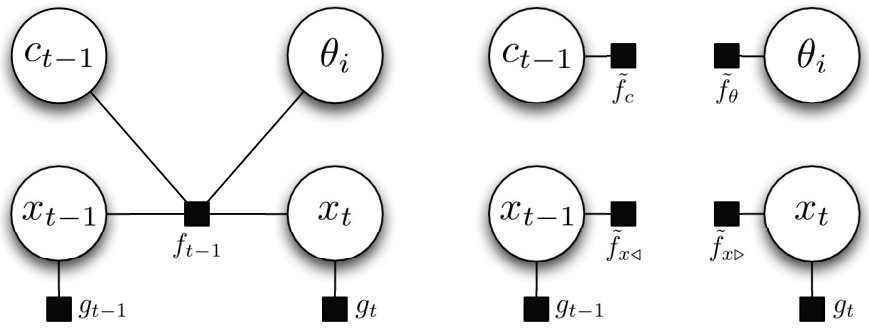
Iterating steps (2) and (3) also means to propagate up $\mu_{f_t \to w}$ and down $\mu_{w \to f_t}$ messages between the state-level and the parameter-level until coherence between both levels is achieved. In the first case, the prediction error $\mathbf{y} - \hat{\mathbf{y}}$ is propagated from the output space to the parameter space and is used to calculate a new estimate; in the second case, the new estimate is used to recalculate the prediction error. When this error is small enough, the algorithm converges to a solution. This process is mediated by the state space, which acts as a glue. The prediction error is initially projected in the state space (to correct the state prediction made by the forward model) and then projected into the parameter space (to correct the its current parameters).

2.5 Multiple paired forward-inverse model

Now, we assume that the dynamics of the system f are not fixed over time but can take on a possibly infinite number of different forms. These different forms correspond to the *context* of the control and include such factors as interactions with objects or changes in the environment. This can either be parameterized by assuming there is a set of system dynamics f_i or by including a context parameter c as part of the dynamics

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, c_t = i) = f_i(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \tag{2.48}$$

where c_t encapsulates the context at time t. The aim of the overall controller is to learn to control the system under different and unknown contexts. We address this problem using a Bayesian approach.



(a) (b)

Figure 2.3: On the left is the graphical model for multiple forward-inverse model learning. On the right is the corresponding factorized approximation.

Suppose we have a training set consisting of pairs $\{\mathbf{u}_{t-1}, \mathbf{y}_t\}$ and to know the number of models to be learned, N. We can model the new system with the Bayesian network as shown in Figure 2.3(a), which is a hybrid network, ie a network of discrete and continuous variables. In this hybrid network, the messages exchanged between variables and factors are no longer Gaussian distributions, but are mixture of Gaussians (MOG) distribution. To compute the belief of a single variable, we must combine a Gaussian prior with n mixture of Gaussians likelihoods that yields an MOG posterior with $O(2^n)$ modes. Hence, the full posterior over a single variable cannot be represented exactly or compactly. We may approximate the posterior with a single Gaussian (moment matching).

2.5.1 Expectation Propagation

Expectation propagation exploits the fact that the likelihood is a product of simple terms. If we approximate each of these terms well, we can get a good approximation to the posterior. EP is like Belief Propagation, except that the messages it sends only contain expectation of features, instead of full (marginal) belief states. For many probabilistic models, the joint distribution of data $\mathcal{D}$ and hidden variables (including parameters) $\boldsymbol{\theta}$ comprises a product of factors in the form

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}_i)$$

where $\boldsymbol{\theta}_i$ represents the subset of variables associated with factor f_i. It apply to any model defined by a directed probabilistic graph in which each factor is

a conditional distribution corresponding to one of the nodes, or an undirected graph in which each factor is a clique potential. We approximate this using a fully factorized distribution of the form

$$q(\boldsymbol{\theta}) = \prod_k q_k(\theta_k) \tag{2.49}$$

where θ_k corresponds to an individual variable node. Each factor $q_(\theta_k)$ represents an approximated belief of node θ_k and is defined as

$$q_k(\theta_k) = \prod_i \tilde{f}_{ki}(\theta_k) \tag{2.50}$$

In order to obtain a pratical algorithm, we need to constrain the approximated factor $\tilde{f}_{ki}(\theta_k)$ in some way. We shall assume that they come from the exponential family.

Suppose that we wish to refine the particular term $\tilde{f}_i(\boldsymbol{\theta}_i) = \prod_{k\in\theta_i} \tilde{f}_{ki}(\theta_k)$ keeping all other terms fixed. We first remove the factor $\tilde{f}_i(\boldsymbol{\theta}_i)$ from the current approximation to the posterior by defining the unnormalized distribution

$$q^{old}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\tilde{f}_i(\boldsymbol{\theta}_i)} = \prod_k \frac{q_k(\theta_k)}{\tilde{f}_{ki}(\theta_k)} = \prod_k q_k^{old}(\theta_k) \tag{2.51}$$

We see that each message $q_k^{old}(\theta_k)$ is a prior term and is equivalent to the message $\mu_{x_k\to f_i}$. Note that we could instead find $q_k^{old}(\theta_k)$ from the product of factors $j \neq i$, although in practice division is usually easier. This is now combined with the exact factor $f_i(\boldsymbol{\theta}_i)$ to give a distribution

$$\frac{1}{Z_i} f_i(\boldsymbol{\theta}) q^{old}(\boldsymbol{\theta}) \tag{2.52}$$

where Z_i is the normalization constant given by

$$Z_i = \int_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i) q^{old}(\boldsymbol{\theta}) \tag{2.53}$$

which involves taking the marginal of $f_i(\boldsymbol{\theta}_i)$ multiplied by any terms from $q^{old}(\boldsymbol{\theta})$ that are functions of any of the variables in $\boldsymbol{\theta}_i$. Expectation propagation chooses each approximation such that the posterior using the term exactly and the posterior using the term approximately are close in KL-divergence

$$KL\left(\frac{1}{Z_i} f_i(\boldsymbol{\theta}) q^{old}(\boldsymbol{\theta}) \,\middle|\, q^{new}(\boldsymbol{\theta})\right) \tag{2.54}$$

It is straightforward to obtain the required expectations for any member of the exponential family, provided it can be normalized, because the expected statistics can be related to the derivatives of the normalization coefficient

$$\mathbf{m}_k^{new} = \mathbf{m}_k^{old} + \mathbf{V}_k^{old} \Delta_{m_k} Z_i \tag{2.55}$$

$$\mathbf{V}_k^{new} = \mathbf{V}_k^{old} - \mathbf{V}_k^{old} \left(\Delta_{m_k} Z_i \Delta_{m_k} Z_i^T - 2\Delta_{V_k} Z_i \right) \mathbf{V}_k^{old} \tag{2.56}$$

We see that the revised factor $\tilde{f}_i(\boldsymbol{\theta}_i)$ can be found by taking the new posterior approximation and dividing out the remaining factors so that

$$\tilde{f}_i(\boldsymbol{\theta}_i) = K \frac{q^{new}(\boldsymbol{\theta})}{q^{old}(\boldsymbol{\theta})} = \bar{K} \prod_k \frac{q_k^{new}(\theta_k)}{q_k^{old}(\theta_k)} \tag{2.57}$$

The coefficient K is given by

$$K = \int_{\boldsymbol{\theta}_i} \tilde{f}_i(\boldsymbol{\theta}_i) q^{old}(\boldsymbol{\theta}) = \bar{K} \prod_{j \notin \theta_i} q_j^{old}(\theta_j) \tag{2.58}$$

Terms that correspond to other factors $\tilde{f}_j(\boldsymbol{\theta}_j)$ will cancel between numerator and denominator when we divide by $q^{old}(\boldsymbol{\theta})$ in Eq. 2.57.

2.5.2 Learning from trajectory

We consider the joint probability distribution

$$\begin{aligned} &P(\mathbf{x}_{1:T}, \mathbf{u}_{1:T}, \mathbf{y}_{1:T}, c_{1:T}, \mathbf{w}) \\ &= \left[\prod_{t=1}^{T} P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, c_{t-1}, \mathbf{w})\, P(\mathbf{y}_t \,|\, \mathbf{x}_t) P(c_{t-1}) \right] P(\mathbf{x}_0) P(c_0) P(\mathbf{w}) \end{aligned}$$

where c_t is a discrete random variable and and the parameter $\mathbf{w}$ depends on the context variable.

We seek an approximation $q(\mathbf{x}_{1:T}, c_{1:T}, \mathbf{w})$ that has the same factorization, so that

$$q(\mathbf{x}_{1:T}, c_{1:T}, \mathbf{w}) = \prod_{t=1}^{T} \tilde{f}_{t-1}(\mathbf{x}_t, \mathbf{x}_{t-1}, c_{t-1}, \mathbf{w}) \tag{2.59}$$

Now we restrict attention to approximations in which the factors themselves factorize with respect to the individual variables so that

$$q(\mathbf{x}_{1:T}, c_{1:T}, \mathbf{w}) = \prod_{t=1}^{T} \tilde{f}_{t-1}(\mathbf{x}_t) \tilde{f}_{t-1}(\mathbf{x}_{t-1}) \tilde{f}_{t-1}(c_{t-1}) \prod_{i=1}^{N} \tilde{f}_{t-1}(\mathbf{w}_i) \tag{2.60}$$

which corresponds to the factor graph shown on the right in Figure 2.3(b). Because the individual factors are factorized, the overall distribution is itself fully factorized

$$q(\mathbf{x}_{1:T}, c_{1:T}, \mathbf{w}) = \prod_{i=1}^{N} q(\mathbf{w}_i) \prod_{t=1}^{T} q_t(\mathbf{x}_t) q_t(c_t) \tag{2.61}$$

Now we apply the EP algorithm using the fully factorized approximation. Suppose that we have initialized all of the factors and that we choose to refine factor $\tilde{f}_{t-1}$. We see that the only factors in $q(\mathbf{x}_{1:T}, c_{1:T}, \mathbf{w})$ that change when we update $\tilde{f}_{t-1}$ are those that involve the variables in f_{t-1}. We remove this factor from the approximating distribution to give

$$q^{old}(\mathbf{x}_t, \mathbf{x}_{t-1}, c_{t-1}, \mathbf{w}) = q^{old}(\mathbf{x}_t) q^{old}(\mathbf{x}_{t-1}) q^{old}(c_{t-1}) \prod_{i=1}^{N} q^{old}(\mathbf{w}_i) \tag{2.62}$$

where the partial approximation are defined as

$$q^{old}(\mathbf{w}_i) = \frac{q(\mathbf{w}_i)}{\tilde{f}_{t-1}(\mathbf{w}_i)} = \mathcal{N}(\bar{\mathbf{w}}_i, \mathbf{\Lambda}_w^i) \qquad q^{old}(\mathbf{x}_t) = \frac{q(\mathbf{x}_t)}{\tilde{f}_{t-1}(\mathbf{x}_t)} = \mathcal{N}(\bar{\mathbf{x}}_t, \mathbf{\Lambda}_{x_t})$$
$$q^{old}(\mathbf{x}_{t-1}) = \frac{q(\mathbf{x}_{t-1})}{\tilde{f}_{t-1}(\mathbf{x}_{t-1})} = \mathcal{N}(\bar{\mathbf{x}}_{t-1}, \mathbf{\Lambda}_{x_{t-1}}) \quad q^{old}(c_{t-1}) = \frac{q(c_{t-1})}{\tilde{f}_{t-1}(c_{t-1})} = \bar{p}_{t-1}^i$$

and we then multiply this by the exact factor to give

$$\hat{p}(\mathbf{x}_t, \mathbf{x}_{t-1}, c_{t-1}, \mathbf{w}) = P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, c_{t-1}, \mathbf{w}) q^{old}(\mathbf{x}_t, \mathbf{x}_{t-1}, c_{t-1}, \mathbf{w})$$

We now find q^{new} by minimizing the Kullback-Leibler divergence $KL(\hat{p} \,|\, q^{new})$. This involves taking the marginal of $\hat{p}$ for any of the variables involved in f_{t-1}. We therefore evaluate the normalization constant Z as follows

$$Z = \sum_{i=1}^{N} q^{old}(c_{t-1}) \int_{\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{w}_i} P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{u}_{t-1}, c_{t-1}, \mathbf{w}) q^{old}(\mathbf{x}_t) q^{old}(\mathbf{x}_{t-1}) q^{old}(\mathbf{w}_i) \tag{2.63}$$

It's easy to show that the result is a mixture of Gaussians distribution, where $\bar{p}_{t-1}^i$ are the weights of each mixture components and each Gaussian distribution $\mathcal{N}(\bar{\mathbf{x}}_t \,|\, \bar{\mathbf{x}}_{i,t}, \mathbf{\Lambda}^i)$ is defined as

$$\bar{\mathbf{x}}_{i,t} = f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, \hat{\mathbf{w}}_i) + \mathbf{J}_{i,x} (\bar{\mathbf{x}}_{t-1} - \hat{\mathbf{x}}_{t-1}) + \mathbf{J}_{i,w} (\bar{\mathbf{w}}_i - \hat{\mathbf{w}}_i) \tag{2.64}$$
$$\mathbf{\Lambda}^i = \mathbf{C} + \mathbf{J}_{i,x} \mathbf{\Lambda}_{x_{t-1}} \mathbf{J}_{i,x}^T + \mathbf{J}_{i,w} \mathbf{\Lambda}_w^i \mathbf{J}_{i,w}^T + \mathbf{\Lambda}_{x_t} \tag{2.65}$$

We compute the mean and covariance of q^{new} by finding the mean and covariance of $\hat{p}$. Here we show only the mean update equation. First we compute the derivatives of Z for each factor in q^{old} and than, using the equation 2.55, we obtain the new approximation.

For example, we want to update the factor $q^{old}(\mathbf{x}_t)$. First we calculate the derivative of Z for $\bar{\mathbf{x}}_t$ to obtain

$$\Delta_{\bar{x}} \log Z = -\frac{1}{Z} \sum_{i=1}^{N} \bar{p}_{t-1}^{i} \mathcal{N}(\bar{\mathbf{x}}_t \,|\, \bar{\mathbf{x}}_{i,t}, \mathbf{\Lambda}^i)\ {\mathbf{\Lambda}^i}^{-1} (\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{i,t}) \tag{2.66}$$

To simplify the notation, we introduce the variable λ_i, defined as

$$\lambda_{t-1}^{i} = \frac{\bar{p}_{t-1}^{i} \mathcal{N}(\bar{\mathbf{x}}_t \,|\, \bar{\mathbf{x}}_{i,t}, \mathbf{\Lambda}^i)}{\sum_{i=1}^{N} \bar{p}_{t-1}^{i} \mathcal{N}(\bar{\mathbf{x}}_t \,|\, \bar{\mathbf{x}}_{i,t}, \mathbf{\Lambda}^i)} \tag{2.67}$$

Using the equation 2.55, we get

$$\mathbf{x}_t^{new} = \bar{\mathbf{x}}_t - \sum_{i=1}^{N} \lambda_{t-1}^{i} \mathbf{\Lambda}_{x_t}\ {\mathbf{\Lambda}^i}^{-1} (\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{i,t}) \tag{2.68}$$

Below we list the updates for other factors

$$\mathbf{w}_i^{new} = \bar{\mathbf{w}}_i - \lambda_{t-1}^{i} \mathbf{\Lambda}_w^i \mathbf{J}_{i,w}^T\ {\mathbf{\Lambda}^i}^{-1} (\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{i,t}) \tag{2.69}$$

$$p_{t-1}^{i} = \lambda_{t-1}^{i} \tag{2.70}$$

Then we use the equation 2.57 to compute the refined factor $\tilde{f}_{t-1}$. This refinement process is repeated until a suitable termination criterion is satisfied, for instance that the maximum change in parameter values resulting from a complete pass through all factors is less than some threshold.

2.5.3 Controlling the system

For each behavior captured by a forward model we would wish to learn a control strategy, i.e. an inverse model. Therefore, for each forward model there would be a paired inverse model. Each inverse model would have as input the desired next output $\mathbf{y}_t^*$ and would produce a motor command $\mathbf{u}_{t-1}^i$ The aim is that each inverse model learns to provide suitable control signals under the context for which its paired forward model provides good predictions.

The Bayesian framework allows direct calculation of the control $\mathbf{u}_t$ without further assumptions. We don't need to define parametric functions associated with the inverse models. Instead, we can directly calculate the

individual contributions associated with each forward model and the entire sequence of controls $\{\mathbf{u}_t\}_{t=1}^{T}$.

We introduce a new factor to the equation 2.62

$$q^{old}(\mathbf{u}_{t-1}) = \frac{q(\mathbf{u}_{t-1})}{\tilde{f}_{t-1}(\mathbf{u}_{t-1})} = \mathcal{N}(\mathbf{u}_{t-1} \mid \bar{\mathbf{u}}_{t-1}, \mathbf{\Lambda}_u) \tag{2.71}$$

We use the linearization of f and g functions, as defined in the single-step case. Equations 2.64 and 2.65 become

$$\bar{\mathbf{x}}_{i,t} = f(\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{u}}_{t-1}, \mathbf{w}_i) + \mathbf{J}_{i,x}\left(\bar{\mathbf{x}}_{t-1} - \hat{\mathbf{x}}_{t-1}\right) + \mathbf{J}_{i,u}\left(\bar{\mathbf{u}}_{t-1} - \hat{\mathbf{u}}_{t-1}\right) \tag{2.72}$$

$$\mathbf{\Lambda}^i = \mathbf{C} + \mathbf{J}_{i,x}\mathbf{\Lambda}_{x_{t-1}}\mathbf{J}_{i,x}^T + \mathbf{J}_{i,u}\mathbf{\Lambda}_u\mathbf{J}_{i,u}^T + \mathbf{\Lambda}_{x_t} \tag{2.73}$$

The total motor command generated by the whole set of N forward models is given

$$\mathbf{u}_{t-1}^{new} = \bar{\mathbf{u}}_{t-1} - \sum_{i=1}^{N} \lambda_{t-1}^i \mathbf{\Lambda}_u \mathbf{J}_{i,u}^T \, {\mathbf{\Lambda}^i}^{-1} \left(\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{i,t}\right) \tag{2.74}$$

Again the variables λ_{t-1}^i would be used to weight the prediction error $\bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{i,t}$ of each forward model and to correct the current estimation of the control $\mathbf{u}_{t-1}$.

2.5.4 Discussion

In this section, we have proposed a Bayesian model which can solve the motor learning and selection problems in a computationally coherent manner. We applied the distal learning principles to multiple model learning. The key to this model is the variable λ_{t-1}^i which reflect, at any given time, the degree to which each pair of forward and inverse models should be responsible for controlling the current behavior. If we look at equation 2.67, we notice that $\bar{p}_{t-1}^i$ acts as a prior, updated by the likelihood $\mathcal{N}(\bar{\mathbf{x}}_t \mid \bar{\mathbf{x}}_{i,t}, \mathbf{\Lambda}^i)$ that depends on the prediction error of i-th forward model. Those forward models which capture the current behavior, and therefore have low errors, will have high responsibilities. In MOSAIC ([Haruno et al., 2001]), this variable is known as *responsibility signal*. This responsibility signal both couples the inverse and forward model pairs, guides learning in each pair of the inverse and forward models, and gates the contribution of each inverse model's output to the final output. The variable λ_{t-1}^i is the probabilistic equivalent of this signal.

As in MOSAIC, the variables λ_{t-1}^i are used in three ways: (1) to gate the learning of the forward models, (2) to gate the learning of the inverse models

and (3) to gate the contribution of the inverse models to the final motor command. In summary, each forward model receives the total motor command, and each models' prediction is compared with the true outcome. Only those forward models with small errors should adapt and those with large errors should learn little. Conceptually speaking, if one forward model's prediction is good, its corresponding inverse model receives the major part of the motor error signal and its output contributes significantly to the final motor command. On the other hand, if the forward model's prediction is poor, its corresponding inverse model does not receive the full error and its output contributes less.

2.6 Experiments

2.6.1 Object dynamics

To examine motor learning and control, we simulated a task in which the hand had to track a given trajectory $\{\mathbf{y}_t^*\}_{t=1}^T$, while holding different unknown object. Each object had a particular mass, damping, and spring constant (M, B, K). We model the dynamics of each object as a next-state linear system, defined as follows

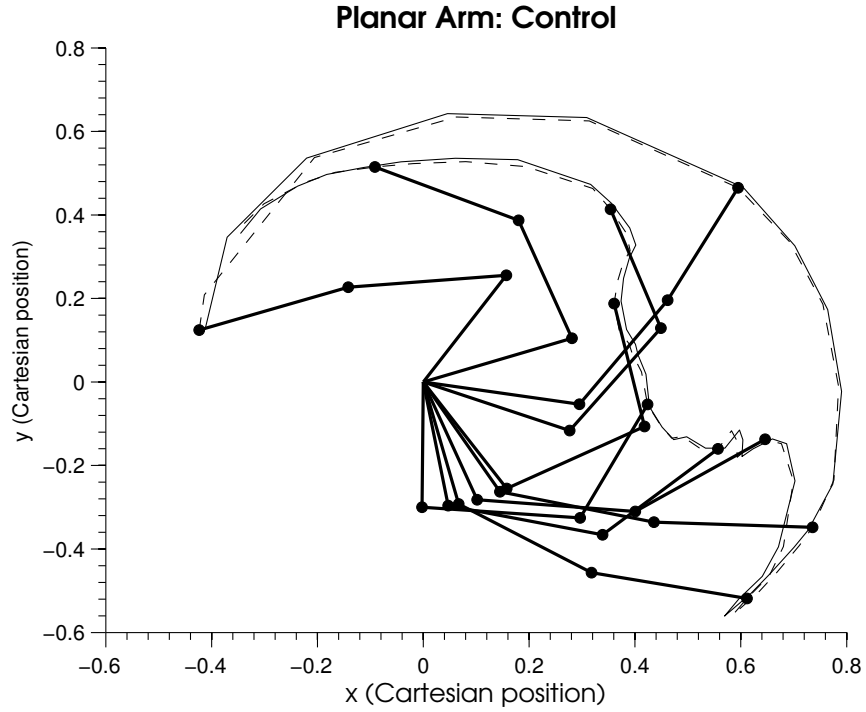
$$\begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ -\frac{K}{M}\Delta t & 1 - \frac{B}{M}\Delta t \end{bmatrix} + \begin{bmatrix} x_{t-1} \\ \dot{x}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta t}{M} \end{bmatrix} u \tag{2.75}$$

and an output function

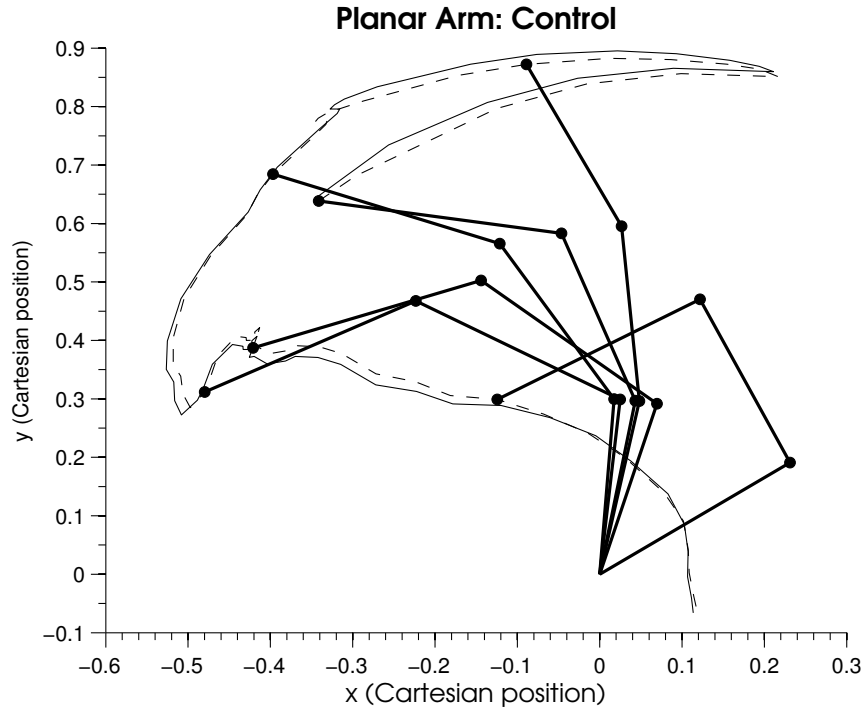
$$\begin{bmatrix} y_t \\ \dot{y}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} \tag{2.76}$$

The experiment goals are: (1) to learn the motor commands to compensate for the dynamics of the different objects and (2) to learn the dynamics of each objects. Forward models were implemented as a linear neural network. The use of linear networks allowed M, B, and K to be estimated from the forward model weights.

In the first simulation, we used the BP method to train forward-inverse model pair. The learning rate matrix $\mathbf{H}^{-1}$ was set to 0.01. The sampling rate was 100 Hz, and a trial was a 5 second run. The initial forward model is used to generate the sequence of commands $\{\mathbf{u}_t\}_{t=1}^T$. We used AICO framework to generate the controls. These controls are then used to generate the training set, consisting of pairs $\{\mathbf{y}_t, \mathbf{u}_{t-1}\}$. The output signal $\mathbf{y}_t$ was corrupted with Gaussian noise (var = 0.1, 0.3, 0.5). Figure 2.5 shows an evolution of the forward model estimates of M,B, K during learning. The parameters started

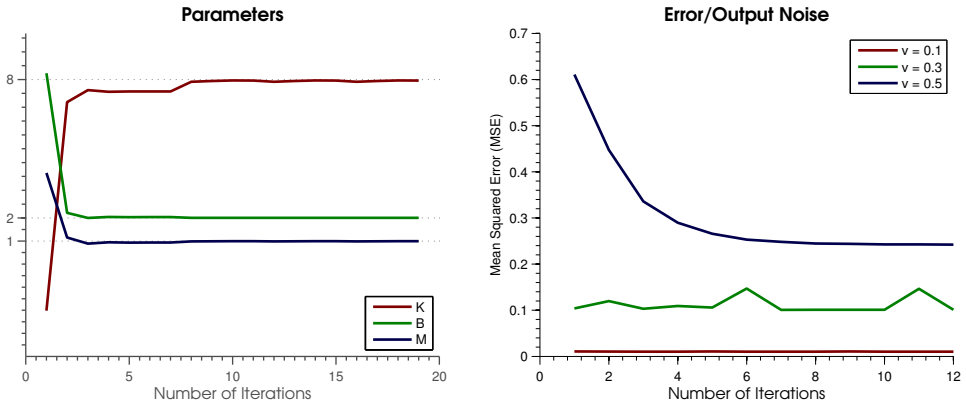


(a)



(b)

Figure 2.4: The belief over the endeffector trajectory (approximate forward model). Also the mean joint configurations for some time step are displayed. The dotted lines represents the approximated trajectories and the solid line represents the true trajectories.



(a) Parameters convergence (var = 0.1) (b) Total error (MSE) during the training phase

Figure 2.5: Learning single dynamic

from randomly selected initial condition and converged over 15 trials to very good approximations. Figure 2.5(b) shows the performance of the model during the learning phase. It is interesting to note that the method compensates for the output noise and, at the same time, calculates with reasonable accuracy the parameters of the forward model.

2.6.2 Planar arm kinematics

In the second simulation, we try to learn the forward and inverse kinematics of a three-joint planar arm. As shown in Figure 2.4 the configuration of the arm is characterized by the three joint angles q_1, q_2 and q_3 and the corrisponding pair of Cartesian variables x_1 and x_2. The function that relates these variables is the forward kinematic function $\mathbf{x} = g(\mathbf{q})$. It is obtained in closed form using elementary trigonometry:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \end{bmatrix} \tag{2.77}$$

where l_1, l_2 and l_3 are the link lengths.

The forward kinematic function g is a many-to-one mapping: for every Cartesian position that is inside the boundary of workspace, there are an infinite nuber of joint angle configurations to achive that position. This implies that the inverse kinematic relation g^{-1} is not a function: rather, there are an infinite number of inverse images of each of the Cartesian positions.

In the simulation, the joint angles of the arm were represented using the vector $(q_1, q_2, q_3])^T$. The motion of the joints was restricted to the intervals

$[-\frac{\pi}{2}, \frac{\pi}{2}]$, $[0, \pi]$ and $[0, \pi]$. The Cartesian variables x_1 and x_2 were represented as real numbers. Forward kinematics was implemented as a feedforward neural network with sigmoidal activation function. A single layer of 50 units was used. No attempt was made to optimize the number of the hidden units or their connections. These variables were represented directily as real-valued activations of units in the network. Thus, three units were used to represents joint-angle configurations and two units were used to represent Cartesian positions.

Initially, the forward kinematics was trained. The simulation provided input vectors to the network by sampling randomly from an uniform distribution in joint space. Target vectors were obtained by mapping the input vectors into Cartesian space according to Equation 2.77. Initial parameters of the network were chosen randomly from an uniform distribution on the interval $[-0.5, 0.5]$. The initial value of the root-mean-square (RMS) Cartesian position error was 31.66. After 4 iterations of BP algorithm, the RMS error reached asymptote at value of 0.01.

In Figure 2.4, we demonstrate that the Bayesian approach can find particular inverse kinematic mapping. A trajectory in Cartesian space was used to provide targets during the second phase of the simulation. After 12 iterations, the resulting joint-angle sequence was plotted.

Chapter 3

Action Understanding

3.1 Introduction

Action understanding, or the process of how understand actions performed by others and their goals when we observe them, has recently received great attention in cognitive psychology and social neuroscience.

Historically, two main theories have been proposed for explaining action understanding (and, more broadly, mindreading abilities, which also include the understanding of another's distal intentions, beliefs, emotions, etc.). On the one hand, followers of 'theory theory' argue that, to understand another's action, human rely on an explicit 'theory of mind', or a set of hypotheses (possibly expressed in a propositional format) of what causes actions, or on a rationality principle (i.e., what would be rational in a given situation) [Carruthers, 1996, Csibra and Gergely, 2007]. On the other hand, followers of 'simulation theory' argue that, to understand another's action, humans put themselves into another's shoes, and use their own action repertoire rather than explicit (or propositional) knowledge [Gallese and Goldman, 1998, Goldman, 2006, Gordon, 1986].

The discovery of mirror neurons, or neurons in the F5 area of the macaque brain, which are active both during (transitive) action and the perception of the same action performed by others [Rizzolatti and Craighero, 2004], has boosted a great deal of interest in the topic of action understanding in social cognitive neuroscience, and, at the same time, has provided (prima facie) support to the simulative view, at least for what concerns the understanding of the immediate goals of observed actions.

Several years of research on neural substrate of action understanding and intersubjectivity (which has focused on additional brain areas that form the so-called 'social brain' [Grafton, 2009]) has revealed a number of interesting

features of the (monkey and human) action recognition mechanism. First, the motor resonance phenomenon of mirror neurons, which align one's own and another's action representations, is an *automatic* process[1], in which the activation of motor representations is selective for specific intentional actions. Numerous studies show that populations of mirror neurons in F5 fire only when a specific action is executed (e.g., grasping but not reaching), and do not fire when random patterns of movements (i.e., non intentional actions) are observed [Rizzolatti and Craighero, 2004] (but note that the 'motor vocabulary' of neurons in F5 code for actions at different hierarchical levels, e.g., some of them are effector-specific, and some other effector-independent). Additional evidence for specificity of action understanding comes from study of expert performance, in which action understanding and prediction accuracy depend on the similarity of the motor repertoire of of observed and observing agents [Aglioti et al., 2008, Calvo-Merino et al., 2006]. Second, action is recognized at the *goal* rather than the movement level, as evidenced in studies that dissociate movements and goals in action observation [Csibra and Gergely, 2007, Umiltà et al., 2008]. Third, automatic action understanding mechanisms can explain the recognition of *distal intentions* rather than only proximal action [Fogassi et al., 2005, Iacoboni et al., 2005].

3.1.1 Mirroring and action understanding at the computational level

Despite a large deal of empirical research in physiology and cognitive neuroscience, it is still unclear what are the cognitive processes and computations in the mirror neurons system and the wider brain network devoted to social understanding. One popular idea is that mirror neurons implement a direct *motor resonance* mechanism, in which observed actions are directly mapped into goal representations. A possible computational process behind direct resonance is Hebbian learning [Keysers and Perrett, 2004].

Alternatively, it has been proposed that actions are recognized via *predictive processes* rather than via direct resonance (or other passive or reconstructive processes). In this view, the motor system produce predictions in

[1]Automatic motor simulation has to be distinguished from deliberate processes, such as for instance *motor imagery* [Crammond, 1997, Jeannerod, 1994, Jeannerod and Decety, 1995] or mechanisms of prospection [Schacter et al., 2007, Suddendorf and Corballis, 2007]. In addition, in the social cognition literature, it is believed that automatic processes of action understanding are accompanied and augmented by more complex mindreading mechanisms based on the *intentional stance* [Dennett, 1987, Gergely and Csibra, 2003], which permit to explicitly attribute mental states to others, and to flexibly reason about them [Keysers and Gazzola, 2007].

the service of object and action perception, realizing what has been called *action simulation* or *motor simulation*. This idea connects well with a large body of theories of *internal modeling* in computational motor control [Jeannerod, 2006, Miall and Wolpert, 1996, Wolpert et al., 1995], according to which the CNS uses internal forward models to predict the perceptual effects of motor commands, being them really executed or only imagined. Internal forward models can be used during action observation, too, and permit to *emulate* what is observed so as to enhance on-line perception [Gallese and Goldman, 1998, Grush, 2004, Wilson and Knoblich, 2005], understand of the goal of the action [Wohlschlaeger et al., 2003], or both [Pezzulo, 2008, Wolpert et al., 2003]. Importantly, this is done by reusing one's own motor repertoire, and this explains motor involvement during action perception. Given this theoretical framework, in the last years many experiments have been conducted with the aim to find evidence for predictive processes and forward modeling during action observation. Direct evidence that the time course of mirror-neuron activation anticipates motor observation comes from a MEP (Motor-Evoked Potentials) study [Borroni et al., 2005]. Evidence that action understanding is a predictive process comes from studies in which (the final) part of an action is not visible [Umiltà et al., 2001] or in which motor activation precedes perception when the context is predictable [Kilner et al., 2004]. In addition, a recent TMS study shows that motor facilitation is higher during observation of ongoing action than in its last phase, suggesting that an anticipatory simulation rather than a reconstructive process is in play [Urgesi et al., 2010]. Motor involvement in perceptual processing is not confined to the visual modality, but has been reported also in auditory processing [Gazzola et al., 2006] and speech understanding [D'Ausilio et al., 2009, Meister et al., 2007].

3.2 A Bayesian model of action understanding

3.2.1 An inverse-forward modeling scheme for action understanding

An authoritative view in computational motor control is that to act in a goal-directed manner the brain employs internal models [Kawato, 1999, Wolpert et al., 1995], which are fundamental for understanding a range of processes such as state estimation, prediction and context recognition. Internal models could explain the human's ability to generate appropriate movement patterns

under uncertain environmental conditions. For example, we can pick up an object without knowing its dynamics. To solve this task, it has been proposed that the CNS uses a modular approach in which multiple controllers coexist and are selected based on the movement context or state, so that each inverse-forward pair becomes an *expert* of a given motor action and context [Wolpert and Kawato, 1998].

In computational motor control there are two types of internal models. Internal models that predict the sensory consequences of a motor command are known as *forward models* as they model the causal (forward) relationship between actions and their consequences, $x_t = \phi(x_{t-1}, u_t, c_t)$. A forward model, therefore, can be used to predict how the motor system's state x_t changes in response to a given motor command u_t. An *inverse model* performs the opposite transformation to a forward model, determining the motor command u_t required to achieve some desired outcome x^*. In engineering it is also known as controller, $u_t = \psi(x_{t-1}, x^*)$. When a motor command is generated, an efference copy of the motor command can be used to simulate the sensory consequences under the possible contexts. These predictions are compared with actual sensory feedback. Each predictor can, therefore, be regarded as a *hypothesis tester* for the context that it models. The smaller the error in prediction, the more likely the context. Moreover, each predictor is paired with a corresponding controller forming a predictor-controller pair. The sensory prediction error is used to weight the outputs of the paired controllers: $\delta = x_t - \hat{x}_t = x_t - \phi(x_{t-1}, \psi(x_{t-1}, x^*), c_t)$. We can therefore describe a goal-directed action as: (a) a controller, or inverse model, which determines the appropriate motor command to reach a certain goal, coupled with (b) a predictor, or a direct model, which predicts the consequences of the action in a given context.

It has been proposed that inverse and forward models used in motor control can be reused for motor simulation in support of action prediction and understanding [Wilson and Knoblich, 2005, Wolpert et al., 2003]. Given the system's ability to produce and predict actions via inverse and forward models, action recognition can be described as an *inverse inference* process in which the generative model is "inverted" so as to pass from the observation of movements to the inference of which of the observer's inverse model(s) could have generated the observations [Wolpert et al., 2003]. Since inverse models have associated goals, recognition of inverse models entails recognition of the (more plausible) action goals. Note that action recognition is not a one-step process, but can be as dynamic as action performance itself. This is why motor simulation and prediction can help in the recognition process. Put in simple terms, the same internal models that an organism uses for performing goal-directed action can be re-enacted "in simulation" and used for inferring

others' goals (and possibly imitating them). Forward models, or predictors, can be used as simulators of the consequences of an action, and when paired with an inverse model, a controller, a degree of discrepancy between what I observe and what I do (or just "imagine" of doing) can be produced which helps finessing the initial hypotheses about the observed action goals (and which inverse model could have produced it). Importantly, with this method action are recognized in terms of the observer's action repertoire; this is the key insight to relate this recognition method to the activation of mirror neuron system.

3.2.2 The role of priors and contextual information

One limitation of the aforementioned scheme is that when there are multiple possible action goals associated with the same movement, or multiple distal intentions associated with the same action, our inverse inference method for action understanding is under-determined. For instance, how to know if one clicks a button to turn a light on or off (same movement but different action goal) or if one turns the light on to have light or to check if the electrical system works well (same action but different distal intention) by simply observing the ongoing action? Our system solves this problem by (i) casting it as a probabilistic inference process and (ii) by adding contextual information to the generative architecture. First, in probabilistic terms, a probability distribution of possible action goals is computed rather than a single goal, and multiple hypotheses are maintained in parallel and updated when novel evidence is accumulated. Second, our generative architecture takes into account the context in which the action takes place to help the (probabilistic) estimate. In the preceding example, the context (light is on or of) disambiguates the proximal goal (turning light off or on, respectively) and distal intention (if the actor is an electrician, a checking intention becomes more plausible).

Contextual information and prior knowledge has been manipulated in many empirical studies, and plays a key role in mirroring and action (or intention) recognition. For instance, in the study of Iacoboni et al. [2005], the position of objects provides the necessary contextual information (beginning or end of breakfast) to pick up the right distal intention. In the study of Kilner et al. [2004], a color cue serves as context for the more plausible goal to be executed, and in the study of Umiltà et al. [2001] the presence or absence of food behind the screen (which is known by the monkey) disambiguates between a goal-directed action and a simple movement. Our model accounts for the role of multiple sources of contextual information: goal information (i.e., prior information of the actor's preferences over possible goals, or what

goals are more plausible in the context), environmental information (e.g., before or after breakfast, like in the experiment of Iacoboni et al.), and object affordances (e.g., orientation of handle of a coffee cup, which makes some approach movements more plausible than others). The role of affordances in providing priors is compatible with the idea that canonical neurons [Murata et al., 1997, Rizzolatti et al., 1988], which encode affordances [Tucker and Ellis, 2004], can bias mirror neuron activation. Evidence that canonical responses can interfere with the mirror mechanism comes from a recent study, in which action prediction accuracy decreased when the observed action was incongruent with the action representation elicited in the observer [Craighero et al., 2008].

3.2.3 Action understanding as approximate inference using particle filters

In principle, modular approach based on internal models gracefully solves the problem of action recognition and motor control. However, although the prior information (i.e. context, goal, action preferences) could restrict the number of models to consider, it is unlikely and impractical that all models are maintained in parallel for the entire period of recognition. For each context, there might be hundreds or thousands of internal models (and their number could easily grow in ambiguous situations where the environment does not provide sufficient information for recognition) making the problem of action recognition intractable with scarce resources. We must also consider the inherent diversity during the execution of the same action among different individuals, and the diversity in action execution when performed by the same individual during different trials. Problems associated with the possibly huge number of internal models to take into consideration during action observation, and their inherent stochastic nature, hinder the development of efficient analytical solutions for the problem of action understanding. However, casting the problem of action understanding in a Bayesian framework permits to adopt efficient techniques for *approximate* probabilistic inference under the constraint of limited resources. We adopt *particle filters*, a Monte Carlo technique for sequential simulation [Doucet et al., 2000], where each particle represents a weighted hypothesis of an internal model activation in the action recognition task. The weight of each particle is computed according to the divergence between the predicted state of the internal model the particle belongs to and the observed state; intuitively, severe discrepancies between predictions produced by coupled internal models and observed percepts will lead to assigning low weights to internal models less involved

in explaining the current action observation. The particle filter resampling step (see section 3.3.2 below for technical details) will gradually exclude those models as hypotheses testers, thus focusing computational resources on more accurate models. Interestingly enough, neurological evidence for the revision of goal hypotheses during perception comes from a recent findings by Gangitano et al. [2004], in which motor excitability was initially high but suppressed once the course of the observed action was no longer compatible with a biological movement.

3.3 Computational model

Action recognition, when described as a motor simulation process, is influenced by three main factors: agent's repertoire of actions (represented as coupled forward-inverse models), contextual information, and observation of the movements of the demonstrator (i.e., agent performing the action). Suppose we can extract the noisy measurements of the true state of the demonstrator, z_t, through some predefined perceptual process. The objective of the recognition is to determine the goal-directed action, i_t, that the demonstrator is doing based on the observed state z_t. The action i is associated with a paired inverse-forward model, and it implicitly encodes the demonstrator's goal. The initial choice of which internal models to activate is biased by the a priori given contextual information, c_t, and the contextual induction process described through the probability distribution $p(i_t|c_t)$. Each action i_t is responsible of both generating a motor control u_t, given the (hidden) state x_{t-1} (inverse model), and predicting the next (hidden) state x_t, given the motor control u_t and the previous state x_{t-1} (forward model).

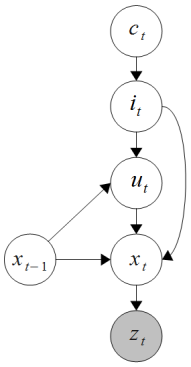
The entire process of action understanding can be cast into a Dynamic Bayesian Network (DBN) shown in Figure 3.1(a)[2]. As usual, shaded nodes represent observed variables while others are hidden and need to be estimated through the process of probabilistic inference. Details of our probabilistic model are given in the Figure 3.1(b-c).

3.3.1 Probabilistic inference for action understanding

Let us denote with $\mathcal{X}_t$ the set of hidden variables at time t, and with $\mathcal{Z}_t$ the set of observed variables at the same time step. In general, we want

[2]DBNs are Bayesian networks representing temporal probability models [Murphy, 2002]. Directed arrows depict assumptions of conditional (in)dependence between variables.

(a) Graphical model



(b) Stochastic variables

c_t	context	$[0, ..., C]$
i	goal-directed action index	$[0, ..., I]$
u_t	motor act	v_x, v_y
x_t	state	x, y
z_t	observed state	x', y'

(c) Probability densities

$p(c_t)$	contextual information
$p(i_t\|c_t)$	contextual induction
$p(u_t\|x_{t-1}, i)$	inverse model
$p(x_t\|x_{t-1}, u_t, i)$	forward model
$p(z_t\|x_t)$	observation model (prediction error)

Table 3.1: Graphical model (DBN) for action understanding based on coupled forward-inverse models

to recursively estimate the posterior distribution $p(\mathcal{X}_t|\mathcal{Z}_{1:t})$ from the corresponding posterior one step earlier, $p(\mathcal{X}_{t-1}|\mathcal{Z}_{1:t-1})$. The usual Markovian assumptions lead to the following equation which, together with an a priori distribution $p(\mathcal{X}_0)$, provides the recursive formulation of the inference task [Murphy, 2002].

The first step involves the application of Bayes rule on the target posterior $p(\mathcal{X}_t|\mathcal{Z}_{1:t})$:

$$\eta p(\mathcal{Z}_t|\mathcal{X}_t, \mathcal{Z}_{1:t-1}) \cdot p(\mathcal{X}_t|\mathcal{Z}_{1:t-1}) \tag{3.1}$$

As clearly shown in the graph in the figure 3.1, the assumption of completeness of $\mathcal{X}_t$ allows us to simplify the equation above. If we knew $\mathcal{X}_t$ and were interested in predicting the evolution of $\mathcal{Z}_t$, no past observed state or contexts would provide us any additional information. $\mathcal{X}_t$ is sufficient to explain the observed variables $\mathcal{Z}_t$, so the above equation can then be further simplified in:

$$\eta p(\mathcal{Z}_t|\mathcal{X}_t) \cdot p(\mathcal{X}_t|\mathcal{Z}_{1:t-1}) \tag{3.2}$$

We can now expand the probability distribution $p(\mathcal{X}_t|\mathcal{Z}_{1:t-1})$:

$$\int p(\mathcal{X}_t|\mathcal{X}_{t-1}, \mathcal{Z}_{1:t-1}) \cdot p(\mathcal{X}_{t-1}|\mathcal{Z}_{1:t-1}) d\mathcal{X}_{t-1} \tag{3.3}$$

Once again, we exploit the assumption that our state is complete. If we know $\mathcal{X}_{t-1}$, past actions and contexts provide no addition information regarding $\mathcal{X}_t$. The above equation can therefore be further simplified in:

$$p(\mathcal{X}_t|\mathcal{Z}_{1:t}) = \eta p(\mathcal{Z}_t|\mathcal{X}_t) \cdot \int p(\mathcal{X}_t|\mathcal{X}_{t-1}) \cdot p(\mathcal{X}_{t-1}|\mathcal{Z}_{1:t-1}) d\mathcal{X}_{t-1} \tag{3.4}$$

In our graphical model for action understanding (fig. 3.1) the task is to recursively compute the posterior distribution over possible forward-inverse action pairs, $p(i_t|z_{1:t})$. This distribution can be obtained by marginalizing the posterior distribution over all hidden variables in the model[3]. The following equations describe the observation and transition models, together with the a priori distribution over the set of hidden variables:

$$p(\mathcal{Z}_t|\mathcal{X}_t) = p(z_t|x_t) \tag{3.5}$$

$$p(\mathcal{X}_t|\mathcal{X}_{t-1}) = p(x_t|x_{t-1}, u_t, i) \cdot p(u_t|x_{t-1}, i) \tag{3.6}$$

$$p(\mathcal{X}_0) = p(x_0) \cdot p(c_0) \cdot p(i|c_0) \tag{3.7}$$

It is worth noting how the coupled forward-inverse models naturally appear in the prediction model (equation 3.6 above).

However, in order to compute the most likely observed action, the recursive propagation of the posterior density $p(\mathcal{X}_t|\mathcal{Z}_{1:t})$ in equation 3.4 is only a theoretical possibility, and in general it cannot be determined analytically. Next section provides a brief technical overview of the particle filters which allow to efficiently perform approximate computation of the posterior density with limited resources.

3.3.2 Particle filters

Functions that describe probability densities in real-world problems are typically nonlinear, and an analytical solution of the Bayesian inference problem is intractable. The key idea of particle filters is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. Each random sample is therefore a distinct *hypothesis* that the agent tracks during the action recognition process. Let $\{x_t^k, u_t^k, i^k, c_t^k, w_t^k\}_{k=1}^{N_s}$ denote a random measure that characterizes the target posterior. The importance weights w_t^k represent an approximation of the relative posterior probabilities of the particles. The weights are normalized such that $\sum_k w_t^k = 1$. A particle filtering algorithm consists of three steps:

Initialization: drawn N_s samples from prior probability density $p(\mathcal{X}_0)$:

[3]Marginalization can be efficiently performed by Monte-Carlo integration methods.

$$x_0^k \sim p(x_0); c_0^k \sim p(c_0); i^k \sim p(i|c_0^k) \tag{3.8}$$

Importance sampling: draw N_s samples from proposal probability density $p(\mathcal{X}_t|\mathcal{X}_{t-1})$:

$$u_t^k \sim p(u_t|x_{t-1}^k, i^k); x_t^k \sim p(x_t|x_{t-1}^k, u_t^k, i^k) \tag{3.9}$$

Update importance weight: recursively update the importance weights up to a normalizing constant:

$$\hat{w}_t^k = w_{t-1}^k \cdot p(z_t|x_t^k) \tag{3.10}$$

and than compute the normalized importance weights: $w_t^i = \frac{\hat{w}_t^i}{\sum_i \hat{w}_t^i}$.

Resampling: draw (with replacement) a set of particles according to the weights $\hat{w}_t^k$. The resampling step, while not strictly necessary, is used to avoid the *particle impoverishment* problem in which the majority of particles' importance weights are close to zero. In addition, resampling allows to focus computational effort on models providing plausible hypotheses (i.e. hypotheses in accordance with the observations) by pruning out less probable models.

3.4 Experimental setup and results

To verify the adequacy of our model to explain action understanding, and assess the robustness of the proposed approximate inference method, we performed a series of experiments. First, we compared the performance of our system with human behavior during action observation in real world. Second, we realized a series of simulations in a scenario that includes key characteristics of experiments in social cognitive neuroscience (as outlined in sec. 3.1), and permits to manipulate the amount of contextual information provided to the system.

Each intentional action is represented as a coupled forward-inverse model, described through the stochastic discrete variable i_t. Inverse models, $p(u_t|x_{t-1}, i_t)$, are implemented as potential fields producing a velocity vector, u_t, for a given target position. Velocities produced by internal models are corrupted by a Gaussian noise with fixed variance, σ_i. For every target object the system automatically instantiate inverse models for *approaching* it. We have also implemented the models for staying still (*stop*), and moving away towards the initial position (*down*). Forward models predict the next 2D position of the observed agent by applying a velocity model $p(x_t|x_{t-1}, u_t, i_t) =$

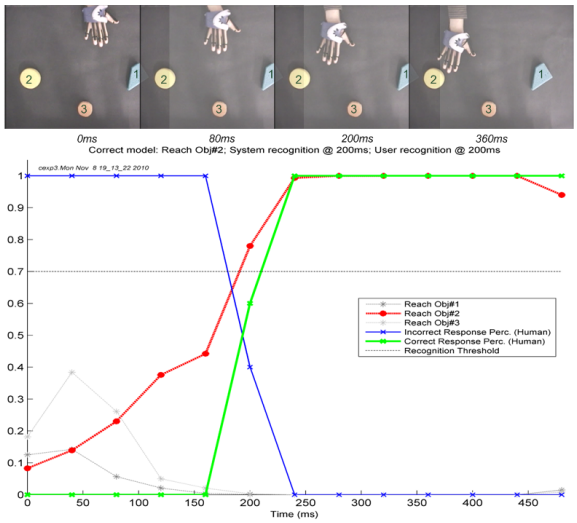


Figure 3.1: Experimental results: comparing human action recognition with our system; The blue curve depicts the average uncertain (or wrong) user response rate, the green one depicts the positive (e.g. correct) user response rate, while the red curve is the posterior probability of the winning action as computed by our system; Note that other two internal models, namely *stop* and *down*, are not shown in the figure

$\mathcal{N}(x_{t-1} + \Delta t \cdot u_t, \sigma_f)$. Predicted positions are therefore corrupted by a Gaussian noise with the fixed variance, σ_f. Without the loss of generality we assume that each inverse model is coupled with the identical forward model. Finally, the observation model is given by $p(z_t|x_t) = \mathcal{N}(x_t, \sigma_o)$, providing the prediction error. For each experiment we compute the posterior distribution $p(i_t|z_{1:t})$ through Monte Carlo integration. In all but the simulation settings, human data is provided by a low-cost commercial motion capture device. The number of particles in all experiments was set to 500.

3.4.1 Comparison with human action recognition

To test whether the recognition ability of our system was correlated with the similar ability of humans to predict the goal of a demonstrated action, we performed a human experiment in demonstrator-observer scenario, with participants (and our model) playing the role of observers.

We recorded 30 video clips at 25 frame-per-second (fps), showing the demonstrator approaching one of several possible objects on a table (average video length was 1,5 s). We have divided recordings into two groups depending on the number of possible target objects (each group contains the same

number of recordings). The 'simple' group contains exacly two target objects, while the 'complex' group can contain up to five target objects. At every frame (40ms) the demonstration was interrupted and we asked participants if they were able to recognize the target of the action corresponding to the goal-directed action (*reaching-object#1*, *reaching-object#2*, ...) by pressing a key on a computer keyboard corresponding to the recognized goal-directed action or an uncertain '*I-dont-know*' response. In order to ease the recognition task we numbered each object in the video. Individual participants ($n = 5$) were randomly selected members of the student population unaware of the purpose of the experiment. Upon the succesful completition of each experiment users were given a reward (a candy).

We measured the performance of the computational model described in sec. 3.3 on the same recordings. The goal of the system, which plays the role of an "observer", is to infer which of its internal models (i_t) provides the best explanation of the perceived demonstration. The 'response' of the system is measured as its current belief (posterior probability of the winning action), and the model having the probability above a fixed threshold (in the current experiment we set it to 0.7) for at least 200ms (5 frames) is elected as the winning model. The contextual distribution $p(c_t)$ is uniform in all experiments indicating that all objects are equally probable action targets.

We were interested in comparing the instant in which our computational system makes the correct prediction to the instant in which the majority of users recognizes the same goal-directed action. At each frame we measured the average number of correct and wrong/uncertain responses provided by partcipants. Fig. 3.1 is a plot of the results for a recording from the 'complex' group; the blue curve depicts the average uncertain (or wrong) user response rate, the green one depicts the positive (e.g. correct) user response rate, while the red curve is the posterior probability of the winning action as computed by our system. Results show that response time our system is qualitatively comparable to that of human participants.

In order to quantify the results we have measured the differences in response times between our system and human users. For each recording we computed the difference between the instant in which the majority of users recognized the correct action, and the instant in which the posterior probability of the correct internal model were above the recognition threshold for five consecutive frames (200ms). Table 3.2 shows the results of the comparison phase for two groups of recordings. The results indicate that response time are quantitavely similar. However, it consistently turns out that humans perform better in 'simple' tasks, while being outperformed by the system in 'complex' tasks. This can be explained by the absence of contextual information: when all the goal-directed actions are equally probable a cluttered

environment provides an additional source of uncertainity for humans which prefer to wait until they are sure of guessing the right target object and earn a reward.

Table 3.2: Comparison between user and system response time

Condition	Avg. response time difference (ms)	STD
Simple	-70	150.9967
Complex	40	56.5685

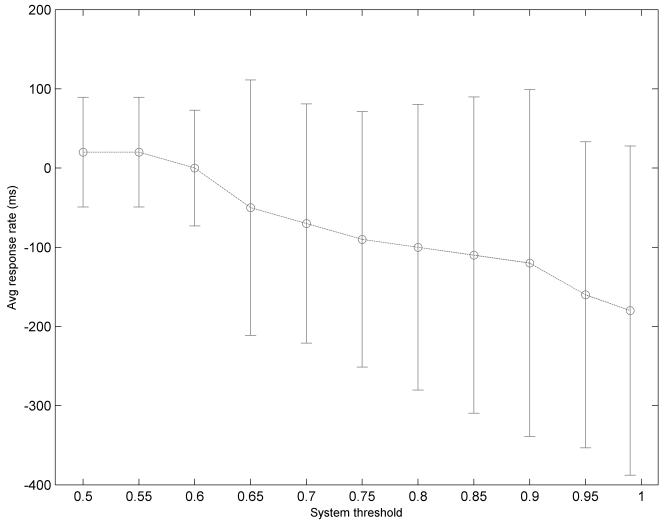
To further investigate the effect of uncertainity on the motor simulation process, we have performed the sensitivity analysis of our system with respect to the action recognition threshold. Fig. 3.2 shows the results obtained in both 'simple' and 'complex' experimental conditions. It is evident that, by increasing the threshold, the behavior of the system tends to that of human participants which prefer making complex decisions with little or no uncertainity. We will provide a detailed discussion of this phenomena in the next section.

Discussion

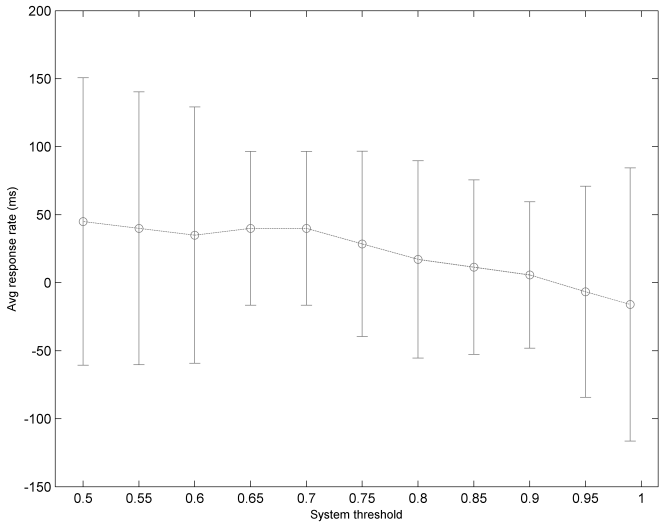
A central hypotheses of our model is the reuse of one's own motor repertoire during action perception. This hypotheses has been tested in several studies, which revealed motor involvement during perception [Bosbach et al., 2005, Kilner et al., 2003], and that action recognition is significantly modulated by the observer's motor repertoire and skills [Aglioti et al., 2008, Calvo-Merino et al., 2005, 2006, Pezzulo et al., 2010].

Casting this hypothesis in Bayesian terms permits to study how uncertainty in the internal models influences uncertainty in the action recognition. Although there are many studies on how uncertainty influences motor planning and execution [Kording and Wolpert, 2006], the influence of uncertainty on the confidence of action recognition has been less studied. According to our view, the timing of action recognition depends significantly on uncertainty of the internal models. In other words, it is only when one of the "hypotheses" has low uncertainty that an human feel confident enough in its recognition. Our set-up permits to test this hypothesis and to compare quantitatively the timing of human and system action recognition. Our results suggest that uncertainty in the internal models significantly affects confidence in the choice and thus response time.

Note that an assumption of this setup is that demonstrator and observer have the same vocabulary of motor acts, whose performance is (near) optimal



(a) Simple condition



(b) Complex condition

Figure 3.2: Effects of system threshold on the recognition accuracy

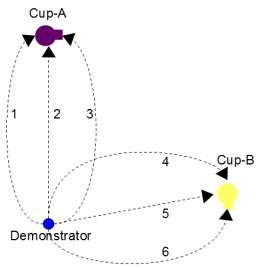


Figure 3.3: Setup used for the simulations. The goal of the demonstrator (blue circle) is to approach one of the two target objects (coffee cups) through the repository of motor acts (internal models). Each motor act is implemented as a goal-directed motor schema. Demonstrator and observer share the same vocabulary of motor acts, currently set to: *approach-from-left* (1 or 4), *approach-straight* (2 or 5), *approach-from-right* (3 or 6).

for achieving a given goal. The requisite of optimal control is plausible for highly-trained human skills such as reaching and grasping, and is achieved. An consequence of this assumption is that each action has a specific and explicit goal, and thus inferring what internal model could have produced the observed action counts as recognizing the demonstrator's goal as well.

3.4.2 Assessing the role of prior contextual information

As discussed in sec. 3.2.2, social neuroscience experiments have shown that action recognition is significantly modulated by prior information, such as probability of events (as dictated by their affordances Craighero et al., 2008) and the context in which they take place (e.g., beginning or end of breakfast, Iacoboni et al., 2005). To test the ability of our model to capture qualitatively similar phenomena, we performed three action recognition tasks on simulated data.

In all the simulations, the system is presented with a 2D experimental setup (see fig. 3.4.2) consisting of two coffee cups (cup-A and cup-B), and a demonstrator, which approaches one of the two cups with one of three possible actions: `approach-straight`, `approach-from-left` or `approach-from-right`. The simulations vary for the kind and amount of contextual information and prior knowledge provided to the system.

The contextual variable, c_t, represents every possible configuration of target objects in the scene (e.g. `cup-B-handle-left`, ...). Agent's *a priori* knowledge is represented in the distribution over the contextual variable, $p(c_t)$ which directly biases the choice of internal models through the process of contextual induction $p(i_t|c_t)$. This distribution implicitly encodes the prior knowledge on which actions are most likely to be observed in a given context (e.g. if the coffee cup has its handle oriented to the left, the most likely

action to perform is `approach-from-left`)[4]. In order to simulate a variety of approaching actions we have introduced an imaginary obstacle in front of each object forcing the agent to turn either to left or to right.

For each experiment we provide the posterior distribution $p(i_t|z_{1:t})$ over time, and the distribution of particles representing both the state (2D position of the demonstrator) and the internal model. A demonstration lasts 250 time steps.

Simulation 1: preferences over actions

In the first simulation, the observer can see that the handles of both cups are oriented toward the right. Although it cannot predict which cup will be grasped, it has strong prior information any cup will be taken from the right side (i.e., that the demonstrator will perform `approach-from-right`, corresponding to 3 or 6 in fig. 3.4.2).

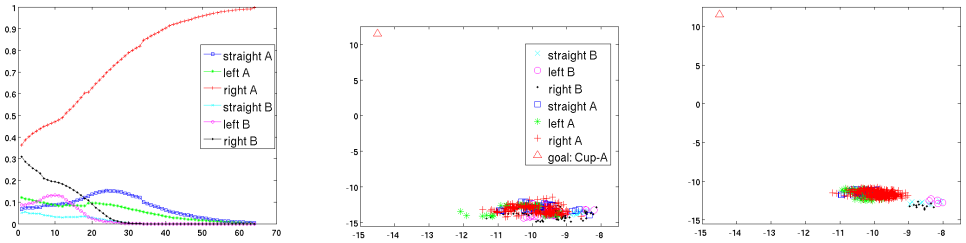
In keeping with studies showing that canonical neurons can influence mirror neuron activation [Craighero et al., 2008], this setup models the role of object affordances in providing priors to the process of action recognition. In our model, object affordances are modeled through the contextual induction distribution, $p(i_t|c_t)$, providing an initial bias towards particular actions (e.g. `approach-from-right`). The results shown in Fig. 3.4 show how, thanks to contextual information, the correct action is recognized from the very beginning of the demonstration. (It is worth mentioning that we performed additional experiments in which we provided wrong contextual information to the agent, and we successfully disambiguated the action based on the stream of data from the observation.)

Simulation 2: ambiguous contextual information

In the second simulation, the observer knows that there are two cups, but cannot see them (neither they know the orientation of their handles), so it has not priors on which action will be executed[5]. This situation is analogous to the experiment of Umilta' et al. [Umiltà et al., 2001], in which the target is not observable. Here, all available internal models equally compete for explaining the observation from the beginning of the demonstration. During the experiment, the demonstrator approaches cup-A from the right. Our results (see fig. 3.5) show that after approximately 20 time steps models relative to goal-directed actions towards the target B are not taken into

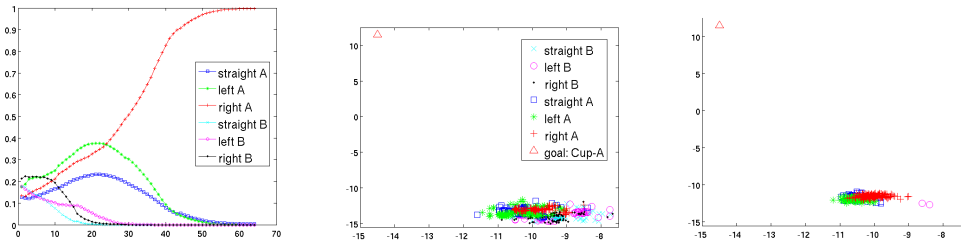
[4]This distribution could be easily learnt through a life-long supervised inductive mechanism.

[5]Technically speaking, this implies sampling from a uniform distribution $p(c_0)$.



(a) Probability distribution $p(i_t)$ (b) Particles at timestep #0 (c) Particles at timestep #15

Figure 3.4: Experimental results: contextual information provides a strong prior over possible actions; in this context the two *approach-from-right* actions, one for each target, are prefered



(a) Probability distribution $p(i_t)$ (b) Particles at timestep #0 (c) Particles at timestep #15

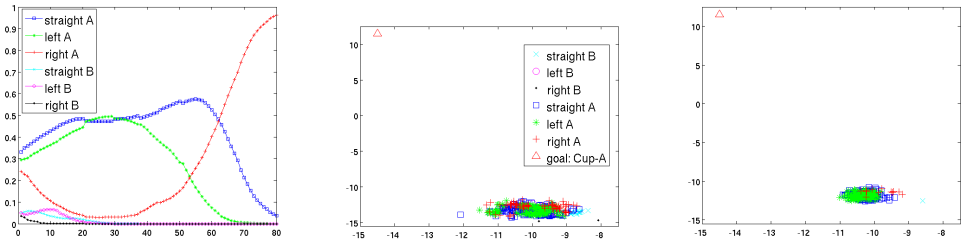
Figure 3.5: Experimental results: contextual information provides no prior over actions or goals

consideration. Since the initial part of goal-directed actions towards the target B is the same, there is an ambiguity until the time step #30 when the demonstrator starts to turn to right in order to correctly approach the object.

Simulation 3: preferences over goals

In the third simulation, the situation is the same as simulation 2, but the observer has additional prior information: it knows that cup-A contains decaffeinated coffee, while cup-B contains coffee. Since it is evening, it has reason to believe that the demonstrator will take cup-A, although it does not know which action it will perform[6]. This situation resembles experimen-

[6]Technically speaking, this implies sampling from a uniform distribution $p(i_t|c_t = cup_A)$.



(a) Probability distribution $p(i_t)$ (b) Particles at timestep #0 (c) Particles at timestep #15

Figure 3.6: Experimental results: contextual information provides a strong prior towards a goal (cup-A in this case); actions pertaining to that goal are prefered over others

tal scenarios used in neuroscience studies, in which monkeys instructed to recognize the action context showed preferences for certain action goals [Kilner et al., 2004] (here, proximal goals). Our results show that (see fig. 3.6) prior information biases the initial allocation of particles, too.

Discussion of simulation results

A qualitative comparison of the results in three simulation settings permits to measure how contextual information affects the dynamics of action recognition. The results indicate that action recognition in our system is modulated by the same contextual factors as those manipulated in social neuroscience experiments, and the result is a qualitatively similar behavior (we discuss the issue of comparing the performance of our model with neural data in the Conclusions).

A second results of our simulations is that the recognition of the goal-directed action (measured as the probability of the best model) is usually achieved long before the action is terminated; this is consistent with neural evidence indicating that action recognition and mirror neuron activation is highly predictive [Borroni et al., 2005, Urgesi et al., 2010].

3.5 The problem of switching actions

A limitation of the model that we have presented is its poor performance when the demonstrator changes its action abruptly (e.g., a “feint” in a soccer game). The reason is that the resampling step of the particle filter usually assigns a high probability to a unique model, and thus a whole set of par-

ticles will follow a unique dynamics. Although theoretically admissible, in many practical cases this behavior can induce misleading results. Consider for example the case when several actions have distinguishable features in the final stage of observation only. In such an ambiguous situation, the particle filter could start tracking a wrong dynamics. Even if the observations will at some point assign low weights to the predicted state, the system has no means to recover the true belief. A similar problem arises with switching dynamics (i.e. when the demonstrator rapidly changes its proximal intention). This problem is intimately connected to the particle impoverishment and the resampling step, and clearly a more sophisticated solution is needed in order to prevent the system from tracking an implausible dynamics.

A potential solution is to explicitly model the transition between different models (i.e. $p(i_t|i_{t-1})$). At every time step, several particles will jump from one dynamics to another thus preventing the impoverishment by tracking a huge number of hypotheses. However, this random walk in the action space forces the recognition algorithm to process most of the actions at each time step, even those that have a low probability of being observed in a given context, thus making the algorithm computationally prohibitive and unsuitable to operate in real time with limited resources. A more sophisticated approach to the "*switching intention*" problem is to populate the space with particles having different dynamics as soon as the particle impoverishment is detected. In this way, the computational burden of the overall algorithm is constrained and the particle filter can recover the true belief.

In order to detect the particle impoverishment, we use the informational theoretic measure based on *Kullback-Leibler (KL)* divergence [Bishop, 2006] between the current belief, represented by the set of weighted particles, and the probability distribution induced by the current observation. The algorithm will inject random particles from the state space when the KL divergence is larger than a given threshold. We represent the belief distribution, $\mathcal{N}_x$, with the first and the second moment (mean and variance), computed as below:

$$\mu_t^x = \sum_k w_t^k x_t^k \qquad \Sigma_t^x = \frac{1}{N} \sum_k w_t^k (x_t^k - \mu_t^x)(x_t^k - \mu_t^x)^T \tag{3.11}$$

$$N = 1 - \sum_k (w_t^k)^2$$

In the same way, we summarize the present observation distribution ($z_t = \hat{x}_t$) as:

$$\mu_t^z = \hat{x}_t \qquad \Sigma_t^z = \sum_k (x_t^k - \mu_t^z)(x_t^k - \mu_t^z)^T \tag{3.12}$$

The impoverishment measure is KL divergence between these distributions:

$$D_{\mathrm{KL}}(\mathcal{N}_x \| \mathcal{N}_z) = \frac{1}{2}\left(\log_e\left(\frac{\det \Sigma_z}{\det \Sigma_x}\right) + \mathrm{tr}\left(\Sigma_z^{-1}\Sigma_x\right) + (\mu_z - \mu_x)^\top \Sigma_z^{-1}(\mu_z - \mu_x) - N\right) \tag{3.13}$$

When particle impoverishment is detected, the algorithm will inject a preset percentage of random particles by sampling i from $p(i|c_t)$ and x_t from $\mathcal{N}_x$. The injected particles, representing different system dynamics, will thus cover a subset of entire action repertoire conditioned on the current context c_t and present belief $\mathcal{N}_x$, without making use of the observation distribution $\mathcal{N}_z$ [7].

3.5.1 Comparison with human action recognition in switching tasks

To test whether the recognition ability of our system was correlated with the similar ability of humans to predict the goal of a demonstrated action even in presence of changing actions we performed a similar set of experiments as described in the sec. 3.4.1. We recorded 20 video clips at 25 frame-per-second (fps), showing the demonstrator approaching one of several possible objects on a table and then suddendly changing the target object of its action (average video length was 2 s). At every frame (40ms) the demonstration was interrupted and we asked participants ($n = 5$) if they were able to recognize the target of the action corresponding to the goal-directed action by pressing a key on a computer keyboard. As in the previous experiment, the goal of the system, which plays the role of an "observer", is to infer which of its internal models (i_t) provides the best explanation of the perceived demonstration at each time step in absence of any contextual information. At each frame we measured the average number of correct and wrong/uncertain responses provided by partcipants. For each recording we collected the differences between the response time of the system and the instant in which the majority of users provided the correct answer.

Fig. 3.7 is a plot of the results for a randomly selected recording. As in the experiments where no switching occures our results show that response time our system is qualitatively comparable to that of human participants. The mean difference in response times (at 0.7 threshold) is $33.3ms$ with a standard deviation of $99.33ms$. we have also performed the sensitivity analysis of our

[7] Indeed, z_t and x_t can not be compared directly since their dependence is usually modelled through a complex non-linear functional relation $z_t = \phi(x_t)$.

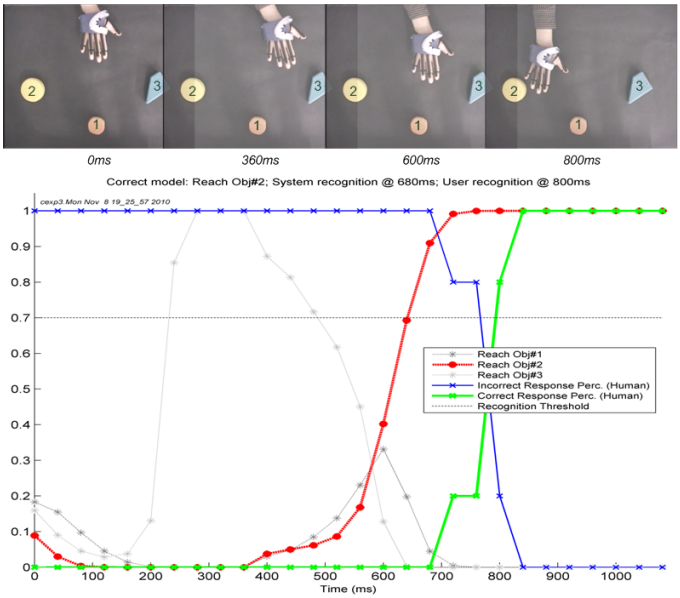


Figure 3.7: Experimental results: comparing human action recognition with our system in the switching condition; The blue curve depicts the average uncertain (or wrong) user response rate, the green one depicts the positive (e.g. correct) user response rate, while the red curve is the posterior probability of the winning action as computed by our system; Note that other two internal models, namely *stop* and *down*, are not shown in the figure

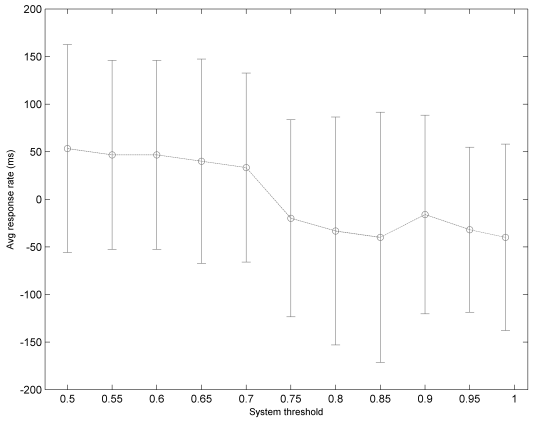


Figure 3.8: Effects of system threshold on the recognition accuracy in the switching condition

system with respect to the action recognition threshold. Fig. 3.8 shows the results obtained in the switching condition. The results, similar to those obtained in the non-switching condition, indicate that the behavior of the system tends to that of human participants which prefer making complex decisions with little or no uncertainity.

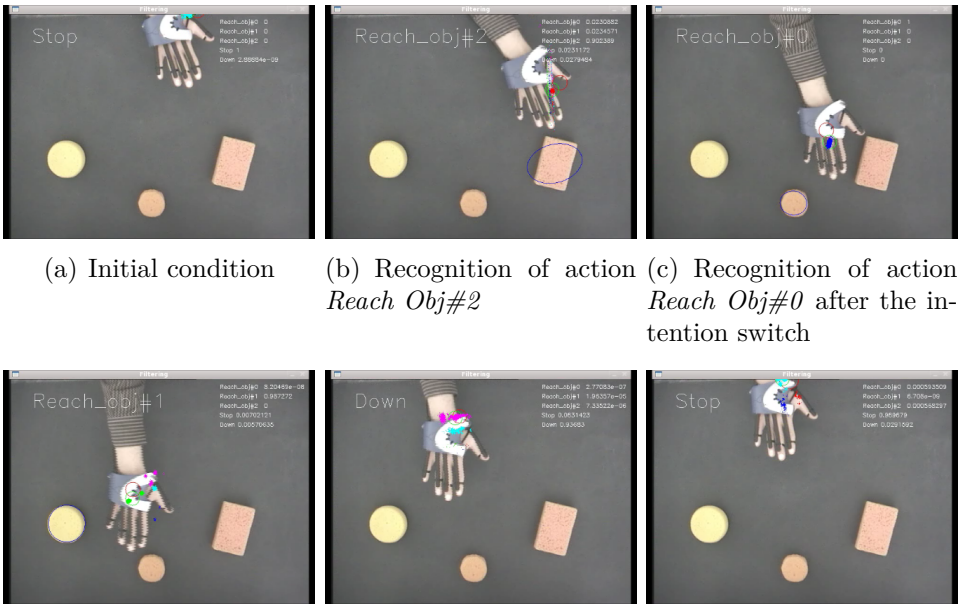
3.5.2 Recognizing sequences of actions

The same system can be exploited for recognizing sequences of actions in the same experimental setup[8] (e.g., reach, stop and then retreat). In this particular case, the goal of the observer is not to recognize a single goal-directed action, but to infer which sequence of internal models explains best the perceived demonstration. Thanks to the KL divergence, the system is able to recognize sudden change in demonstrator's behavior. Our results show that the system was able to correctly recognize demonstrated actions. Some results are shown in Fig. 3.9.

3.6 Conclusions

We have proposed that action understanding can be cast as an approximate Bayesian inference, in which inverse-forward models are hypotheses that are

[8]Note that this is conceptually different from recognizing distal intentions; see the Conclusions for a discussion of this point.



(a) Initial condition

(b) Recognition of action *Reach Obj#2*

(c) Recognition of action *Reach Obj#0* after the intention switch

(d) Recognition of action *Reach Obj#1* after another intention switch

(e) Recognition of action *Move-away*

(f) Recognition of action *Stop*

Figure 3.9: Recognizing a sequence of goal-directed actions. The most probable action, as recognized by the system, is shown in foreground, while the posterior probabilities of each action are shown at the top-right corner. Figures also show the evolution of particles for each internal model.

tested in parallel, using bounded computational resources. The result is a *motor simulation* process, which is initially biased by prior knowledge of plausible goals and affordances and successively refined when novel information is collected that is in agreement or disagreement with the ongoing predictions of the inverse-forward models.

In this framework, motor simulation serves both for action prediction and recognition at the goal level. Indeed, action goals are implicitly encoded in the coupled inverse-forward models, since each pair maps to one goal, or to a probability distribution over possible goals. This reconciles apparently contrasting views, which emphasize on the one hand the benefits of prediction and emulation during perceptual processing [Wilson and Knoblich, 2005], and on the other hand the fact that actions are recognized and imitated at the goal level rather than as movement patterns [Bekkering et al., 2000].

3.6.1 Main contributions

Efficient processing of multiple forward-inverse models

In computational motor control, the idea that the brain adopts multiple forward-inverse models to control and recognize actions is quite popular (see e.g., Demiris and Khadhouri, 2005, Oztop et al., 2005, Wolpert et al., 2003). However, the huge computational complexity of this method prevents its use in most real-world scenarios and risks to hinder its plausibility as a scientific hypothesis as well.

Our model is able to efficiently handle the problem of action recognition via a simulative approach based on many internal models and using limited computational resources. This is achieved by using prior information over possible contexts and goal-directed actions, and by adopting an approximate inference procedure (sequential Monte Carlo simulation) for tracking several competing hypotheses. In particular, the particle filter framework permits to easily embed a huge number of internal models by exploiting all the prior knowledge available to the agent and to limit the activation of internal models.

Motor simulation as a suitable computational-level explanation of action recognition

The fact that our model is efficient and compares positively with human performance is not only remarkable from a technological viewpoint, but it also provides support for motor simulation as a suitable (computational-level) candidate explanation of action recognition, complementing existing

empirical evidence that we have reviewed in sec. 3.1.

As explained in the Introduction, there is a vivid debate on action recognition in the psychological and neuroscience literature. This debate spans on computational modeling as well, and indeed the explanation of action recognition that we offer is not certainly the only possible one.

In a series of computational studies, [Baker et al., 2006, 2009] describe mindreading as a rational Bayesian process of *inverse planning*, which is close to the ideas of "theory theory". A drawback of this method is its computational cost and its huge demands in terms of prior knowledge. To solve these problems, [Ramirez and Geffner., 2010] propose a similar computational method, but use (cheaper) classical planners rather than bayesian inference.

Differently from these studies, the method that we have proposed is based on forward modeling (via the re-enactment of one's own motor repertoire) rather than on the calculation of what is rational. Besides the possibility that simulative and rational methods are complementary for action recognition, the adequacy of our proposal can be evaluated by considering its computational requirements and the specific empirical predictions that it entails.

From a computational viewpoint, what is required is that the observer has a suitable repertoire of motor actions, not that it calculates what is rational. This assumption entails that the maturation of action performance and recognition abilities should follow a similar trajectory, whilst the maturation of a more sophisticated "theory of mind" about others should be less relevant.

A second kind of prediction is related to motor involvement during action perception. On the one hand, our model predicts that specific motor programs in the motor apparatus of observers should be active during action observation. In addition to that, in our model, action observation can be influenced by the embodiment and skills of the observer, the actions it is currently performing (since in that case the same action system is recruited for both execution and performance at the same time) and more in general from a variety of perceptual and motor phenomena that can potentially interfere with the active motor processes. All these influences should have minor effects on action recognition if it does not recruit motor processes.

Motor simulation as approximate Bayesian inference

The idea of using Bayesian models for explaining cognitive processing is not novel, as many researchers have proposed formal theories (e.g., of vision and speech processing) that make use of models belonging to the broad Bayesian family and that have been extensively used in the engineering of visual and speech analysis tasks, such as for instance analysis by synthesis, predictive

coding, or their variants [Friston, 2005, Lee and Mumford, 2003, Rao and Ballard, 1999]. In addition, the idea that cognitive processing is essentially generative and predictive[9] is being widely recognized, leading to the hypotheses of predictive brains [Bar, 2009, Pezzulo and Castelfranchi, 2007, Pezzulo, 2008] and Bayesian brains [Doya et al., 2007].

A recent view in cognitive science is that the CNS uses approximate Bayesian inference in vision, reasoning or other cognitive domains [Chater et al., 2006, Lee and Mumford, 2003]. The emphasis on approximate methods explains how psychological processes come close to the optimal solutions that are postulated by rational theories of cognition, using limited resources; therefore, this method has implications at the cognitive processing level (or Marr's *algorithmic level* [Marr, 1982]) rather than only at the Marr's computational level, which is more common for rational models of cognition.

Recently several studies have investigated the psychological plausibility of approximation algorithms. [Sanborn et al., in press] have compared multiple methods of approximate inference (Gibbs sampling, multi-particle particle filter, single-particle particle filter) in category learning, and found evidence in favor of single-particle particle filter. A study of conditioning reported a good match with a particle filtering method using a single particle [Daw and Courville, 2007].

In a similar vein, we argue that action understanding is an inferential mechanism, in which multiple hypotheses of plausible action goals are maintained and tested in parallel via a sampling method (although we cannot, with our data, provide direct support for particle filtering). Initially, it is the prior information that biases the allocation of particles. Motor simulation (and the reactivation of inverse-forward models) serves to generate hypotheses-related predictions, whose (in)accuracy determines changes in the hypotheses and the way they are tested (i.e., it changes the allocation of particles).

Not only approximate inference is a valid method from a computational viewpoint, but it can be viewed as a mechanistic hypothesis and can explain the response of populations of neurons Doya et al. [2007]; for instance, MCMC methods are compatible with the presence of "broadly congruent" mirror neurons (two-thirds of all MNs) that respond to a broad class of (related) actions rather than a single action, in addition to mirror neurons that respond specifically to a single action.

[9]Note that technically speaking generative and predictive models are not the same. Generative models produce predictions relative to present, not future stimuli, in which priors are propagated top-down. Predictive models produce predictions relative to future stimuli (or rewards). Although important, this distinction is not crucial for the sake of our analysis, since the graphical models that we use afford both kinds of predictions.

The role of priors and contextual information

An additional contribution of our study is clarifying the (computational) role of contextual information. As we have discussed in sec. 3.1, contextual knowledge was manipulated in many social neuroscience experiments. Our study indicates that the different sources of information —affordances, context and preferences— can be treated in a homogeneous way as *Bayesian priors* (although they can have distinct neural underpinnings) that bias the initial allocation of particles, and all modulate the dynamics of our system.

3.6.2 Future work

Extending the model to distal intention recognition

Experiments in which monkeys observe sequences of actions having some elements in common but associated with different distal intentions (e.g., grasping-to-eat vs. grasping-to-place) have emphasized that, first, the mirror neuron mechanism supports the recognition of sequences of actions and not only of individual acts, and second, that the recognition of distal intentions entails the activation of different neurons for the same basic actions, thus revealing dedicated "action chains" in the monkey brain [Fogassi et al., 2005].

The recognition of distal intentions is currently beyond the capability of our model. Future work will include the extension of the model for the recognition of distal intention through a hierarchy of inverse and forward models, where higher-level pairs encode increasingly abstract actions [Wolpert et al., 2003]. Within this architecture, distal intentions (when known) can serve as priors to recognize proximal actions; conversely, the recognition of proximal actions can serve as prior for inferring the demonstrator's distal intention[10].

Comparison with neural data and experimental predictions

Behind the obvious differences between the experimental and simulative setups, comparing the results of our simulation results to neural data is problematic, given the difficulties in finding neural equivalents of the probability

[10]Note however that this process does not exactly address the monkey experiment that we have described so far. The study was conducted by firstly training monkeys (for several weeks) in the execution of the action sequences (also because one of the sequences, grasping-to-place, does not belong to the standard action repertoire of monkeys). These sequences can therefore be treated as highly routinized and habitual. On the contrary, the extension of our model will address the execution of goal-directed movements (for a distinction of habitual and goal-directed actions, see Balleine and Dickinson, 1998, Niv et al., 2006.

distributions $p(i_t)$ and uncertainties as computed by our model. The first aspect of this problem is what neural populations to consider in the comparison. In our model, $p(i_t)$ represents forward-inverse pairs; however, the neural representation of forward-inverse models plausibly involves a wide neural network that spans over several brain areas; for a discussion of this issue, see [Kawato, 1999, Kilner et al., 2007, Imamizu et al., 2003]. Most social neuroscience experiments that we have mentioned measured activity of (single cells of) the mirror neurons system, which are only a part of the brain circuit realizing internal modeling. In other words, the encoding of our system is more 'abstract' than a single neural population. A second, related aspect of the problem is that it is still unclear if and how the critical variables of our model, and namely probability distributions over actions/goals and uncertainty, are represented at the neural level, (with plausible alternatives being population coding, temporal population coding, etc.), and, in case, whether they are coded together or separately [Doya et al., 2007].

Despite so, assuming that the mirror neurons system encodes the goals of actions, which in a sense "summarize" an entire forward-inverse model, it is plausible to assume that the probability distribution $p(i_t)$ should be highly correlated with neural activation in the mirror neurons system. Under the assumption that populations of neurons in the mirror neurons system can be tuned to specific actions, it is plausible to assume that wider populations should be active when the observed movements are compatible with more than one action or more in general when uncertainty is higher; or, in other terms, that the total amount of activation of the mirror neurons system represents uncertainty implicitly.

Chapter 4

Language Grounding

4.1 Introduction

Visually grounded human-robot interaction is recognized to be an essential ingredient of socially intelligent robots and the integration of vision and language increasingly attracts attention of researchers in diverse fields, from robotics and artificial intelligence to computational linguistics and neuroscience Dautenhahn [2007] .

We have investigated the lexical acquisition problem, particularly how a robot can be bootstrapped into communication and what are the necessary prerequisites for robots in order to learn a language. In particular, we focused on grounded systems that learn to generate and understand contextualized spoken descriptions of objects in visual scenes. Unlike other systems we did not address word-boundaries learning in speech signal [Steels and Kaplan, 2001][Roy, 2005].

The process of lexical acquisition in infants seem to be innately driven by the principle of reference: words refer to objects, actions, and attributes of the environment. Observational learning may be used to deduce word meanings from cross-situational experiences. The child is assumed to receive as input a large number of example cases of specific situations and is able to isolate through an inductive learning process what is essential of these situations and learn the appropriate categories underlying language. A well-known problem in observational learning, is the Quine's paradox: an infinite number of possible meanings can be inferred from a finite set of utterance-context pairs. A likely solution to this problem is that all infants have certain biases which constrain the set of possible meanings of words [Fisher et al., 1994][Baldwin, 1995]. For example, the whole object assumption proposes that children will assume a novel label refers to a whole object rather than

its parts. The mutual exclusion assumption proposes that they prefer to assign only one label to a concept. These assumptions are considered good strategies for bootstrapping the inference process, but don't solve completely the referent's problem. The disadvantages of this learning strategy are clear: (a) the passive role of the language learner and (b) little feedback given by the speaker. So in observational learning, the interaction between the learner and the speaker plays no role and is cut off from the learning process. The purpose of language is precisely the communication between individuals. Learning is not only grounded in reality through a sensori-motor apparatus but also through interactions with others [Tomasello et al., 2005]. For example, joint attention plays an important role in learning terms of reference. Infants are more likely to connect words with their referents when engaged in joint attention with their caregivers [Bloom, 1997]. In addition, caregivers can sets constraints on the situation to make it more manageable (scaffolding) and provides feedback upon the infant's actions.

We have used these assumptions to bootstrap the lexical acquisition process. We have partially addressed the problem of finding the referent in early learning stage, endowing robot of the basic attention skills. Originally, the system will acquire a minimal grounded language model without any prior semantic and syntactic information. From this initial model, the system will be able to acquire novel meanings and more complex language models (i.e., spatial clauses). Unfortunately, many words also refer to proprieties that can not be directly observed through sensor, and can't be clarified with pointing gestures and gaze direction. For example, the presence or absence of "aboveness" in the environment is weakly related to the presence and absence of the word "above" [Gold et al., 2009]. Our approach uses the context of the sentence in order to determine the reference of spatial terms.

Our ultimately goal is to take advantage of acquired concepts (representing physical properties and spatial relation of objects in a scene), and language model (encoding all syntactic and semantic constraints) to engage in simple verbal interaction with human partner. All concepts underlying acquired language model are used to initialize dynamic fluents as predicate calculus terms and update robot's logic database representing the state of the world from sensor data. Language acquisition therefore proceed in parallel with concept acquisition. Concepts acquired from lexical acquisition are used to initialize a logic representation of several observable entity of world. For example, the meaning of the word "red" is used to seed the logic representation of the "red" concept. Concepts underlying acquired language model can be considered as independent from language acquisition process and can be reused for other cognitive tasks. We have used these concepts for resolving perceptual ambiguities of description-context pairs. Fig. 4.1

provides an overview of our system.

4.1.1 Learning visually-grounded meanings

Grounding can be considered as the process whereby internal representations are connected to external percepts. This is based on the principle that cognitive agents and robots learn to name entities, individuals and states in the external (and internal) world at the same time as they interact with their environment and build sensorimotor representations of it [Harnad, 1990]. Our approach addresses two interrelated question in lexical acquisition: how semantic categories can be learned and how symbols/words can be associated with appropriate semantic categories. These problems are known as semantic categorization (or semantic clustering) and word-to-meaning mappings, respectively. Categorization produces semantic categories which serve as referents of words. No *a priori* knowledge about innate semantic categories is assumed by the model; instead, through repeated experience, appropriate categories must be learned from positive examples and the model must associate linguistic units with appropriate semantic categories (see Fig. 4.3).

4.1.2 Syntactic bootstrapping

Categories, such as adjective, verbs and nouns, form the basic units for learning the rules of grammar including grammatical relations, cases, and phrase structure configurations. Without syntactic categories a learner will be unable to acquire the rules of the language. However, one-to-one mappings between syntactic and semantic classes does not seem to exist.
Bootstrapping theories provide strategies for deriving syntactic categories from perceptual input. Semantic bootstrapping [Pinker, 1984] proposes that the language learner uses semantic categories to seed syntactic categories. This theory assumes that the learner has already acquired words and their semantics without the use of any syntax.

Basic syntactic categories will be acquired directly through words' meaning, and the words associated with similar perceptual categories will be included in same syntactic category. However, complex adjectives, such as "above" and "below", can not be learnt in the same fashion. Instead, we need to take words' order into account. Treating sentences as unordered set of words tends to lose syntactic contents and its semantic implications. We have exploit word ordering constraint and semantic bootstrapping assumption to acquire spatial terms and explore weak generalization mechanisms.

4.1.3 Joint attention

Joint attention is the process by which one manipulate the attentional behavior of another to attend to the same aspect of the environment via nonverbal means, such as gazing or pointing. An intentional agents must be able to detect attentional behaviour of other agents and influence it to environment features which are relevant for his current activity. In literature, these skills are named as attention detection and manipulation, respectively. Joint attention seems to be necessary for functional speech and is a main prerequisite for early language development. In our work, we stressed only detection capabilities and equipped the robot with some basic skills to detect declarative pointing gestures.

4.1.4 Resolving ambiguities

Another problem we have addressed is how to resolve the ambiguities contained in a description through simple verbal interactions. For example, suppose that the robot hears the statement "Take the red object!" while facing a scene containing several red objects. In order to fulfill the task, the robot must disambiguate the context and select one of the red objects. The process of ambiguity resolution is based on a context-specific human-robot dialogue through specific questions directly related to the properties and spatial relationships of the objects in the observed scene. We have considered only those ambiguities related to perceptual quality shared between objects of the scene. The robot must be able to detect ambiguous statement and select the least ambiguous question to be asked to. If all the red objects have different shape, the robot can resolve easily the ambiguities asking for the shape of a specific object, i.e. "Is the rectangle?". However, if all items have the same shape, the robot must solve the ambiguity by using relative spatial terms between objects, that usually are more ambiguous than the previous. For example, some objects may share particular spatial relationships with other objects (i.e. more objects below the yellow rectangle), making the question partially ambiguous, or even the phrase could not be fully understood (either by a human or by the robot) because the landmark object is indistinguishable from the target.

4.2 Related works

There has been a huge interest in grounded language acquisition in the past years. Visual Translator system [Herzog and Wazinski, 1994] (VITRA) is a natural language generation system which is grounded directly in perceptual

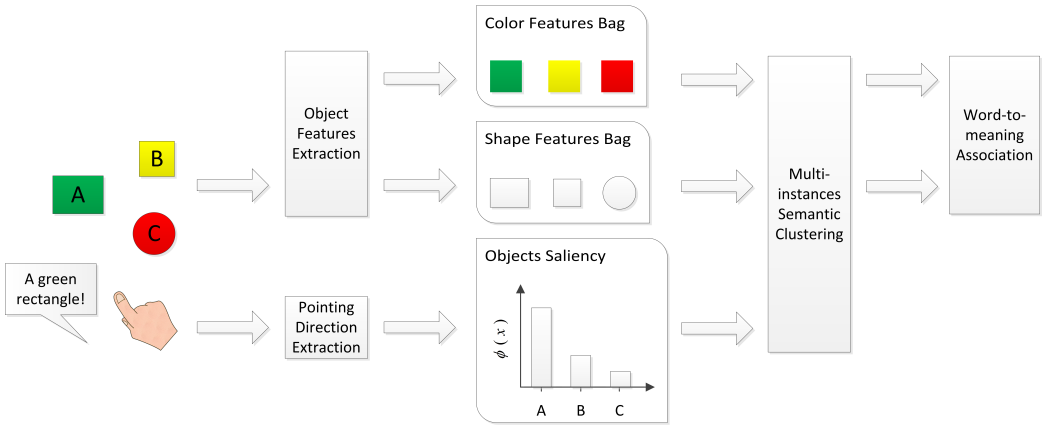


Figure 4.1: General model overview.

input. From a sequence of digitized video frames low-level sensory processes perform recognition and tracking of visible objects. Authors provide a geometrical reconstruction of the perceived scene. Detailed domain knowledge is used to categorize spatial relations between objects, and dynamic events. Higher level propositions are formed from these representations which are mapped to natural language using a rule-based text planner. In contrast to other works, VITRA is not designed as a learning system. Roy implemented a system, CELL [Roy, 2005] (Cross-channel Early Lexical Learning), able to learn object names from a corpus of spontaneous infant-directed speech and process single and two-word phrases which referred to the colour and shape of objects. CELL is the first implemented model of language acquisition which learns words and their semantics from raw sensory input without any human-assisted preparation of data. In DESCRIBER [Roy, 2002], the same authors address the problem syntactic structure acquisition within a grounded learning framework. Learning algorithms acquire probabilistic structures which encode the visual semantics of phrase structure, word classes, and individual words. Using these structures, a planning algorithm integrates syntactic, semantic, and contextual constraints to generate natural and unambiguous descriptions of objects in novel scenes.
Another interesting system is TWIG [Gold et al., 2009], a word learning system that allows a robot to learn compositional meanings for new words that are grounded in its sensors. TWIG allows a robot (1) to learn the meanings of deictic pronouns, (2) to contrast new word definitions with existing ones, thereby creating more complex definition, and (3) to use word learned in an unsupervised manner for production, comprehension, or referent inference. The techniques that TWIG introduces are extension inference and

word definition tree. Its technique are more generally applicable to other word categories, including verbs, prepositions and nouns. Inamura et al. [Inamura et al., 2004] focus on the communicative behavior on a humanoid platform in which the robot narrows down the focus candidate against users' vague instructions based on the certainty factor and degree of localization of target object. The localization is defined as 'a situation in which the target object is distinguished without any perceptual confusion'.

Our work while not making significant advances compared to the systems presented, puts more emphasis on one fundamental problem in language acquisition process: the search for the referent. We endow the system with a real model of attention and formalize a multi-instance learning algorithm for the acquisition of semantic categories. Our goal is to create robust learning algorithms, able to build knowledge even in absence of important pragmatic information.

4.3 System Overview

In our previous work we focused on the learning of grounded language models from examples [Dindo and Zambuto, 2009]. In the experiment we proposed, the demonstrator could chose one of the objects of the scene and provide its, more or less detailed, description. In that case, the referent of a descriptive phrase was directly given to the robot. This information, while on one hand simplifies the learning process and allows the robot to discard the majority of the incorrect associations, on the other reduces the applicability of the technique to more complex environments and does not allow any level of interaction with the demonstrator. In this work, we wanted to relax this constraint by making the interaction between the demonstrator and a robot more natural in the teaching phase.

Without knowing the referent of the sentence (e.g. the object being described by the demonstrator), the language learning problem becomes more difficult. Indeed, the robot should maintain a huge number of assumptions, many of which are incorrect, that would make the association of possible meanings to available words computationally impossible. In the new experimental setting, the teacher first tries to capture the attention of the robot by pointing the object of interest, and once obtained the attention of the robot, she describes the object. The robot is unable to determine with certainty the object, but she may exploit the pointing direction of the demonstrator to filter the possible referents, without other a priori given knowledge. We equipped the robot with some basic skills to simulate the process of joint attention. The robot is able to recognize the demonstrator (face, hand) and

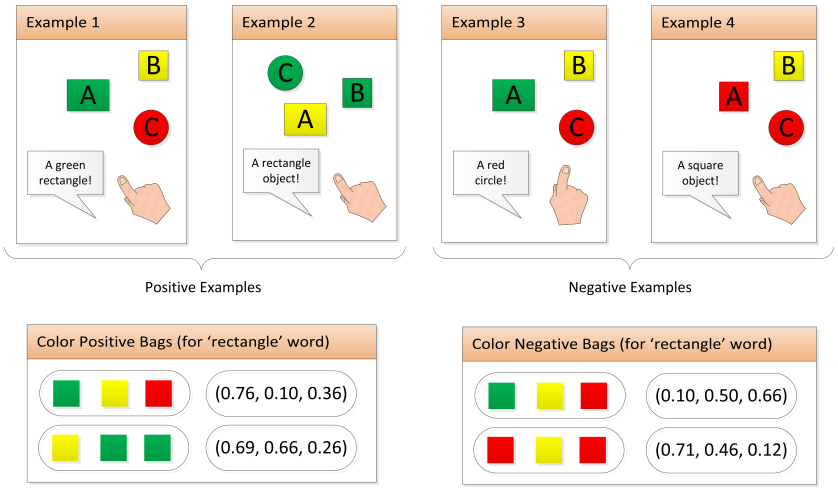


Figure 4.2: Positive and negative bags extraction.

to detect any activity (speech, hand movements, etc.). Moreover, it is able to estimate the direction of the pointing actions. These information is merged in order to determine the salience of each object in the scene. The robot first tries to determine the area to the maximum salience on the work surface, by following hand gestures, then observes and stores the objects that correspond to this area. The robot can also indicate the area of interest for further feedback from the demonstrator. When the demonstrator says something (presumably the object description), the robot stores the most salient objects and the transcript of the statement that he heard, which will constitute the training set for learning. The sample will be discarded when the degree of salience is not high enough, and the robot was unable to identify with certainty the most salient objects. Using a humanoid robot platform, the demonstrator can guess the state of the robot by using the same mechanism of joint attention and correct it, if necessary, so to minimize the ambiguity in the training set.

Before describing the problems addressed in the present work we provide operational definitions of several terms which are used throughout this article. A *semantic category* (or semantic unit) specifies a range of sensory inputs which can be grouped and associated with a word/symbol. For example, a semantic category might specify a portion of the color spectrum. Such a semantic category could be used to ground the semantics for a color term such as "red". A *semantic class* specifies a set of semantic categories grounded in the same sensory channel. For example, a semantic class could be used to associate acquired color terms (color class). A *lexical item* encodes the association between a word and its corresponding semantic category.

All semantic categories are derived from visual sensory signal. Feature extractors computes visual features from the sensors (video). Each extracted feature encodes relevant, non-redundant, information from the visual sensory stream about observable proprieties of the world (semantic class). Potential visual features include categories of shape, color, size, and spatial relation (see Table 4.1). Any word may potentially be paired with any semantic category which is derived from the same utterance-context pair. These pairs are clustered to generate a set of lexical items.

The hand data are collected through low-price glove-like peripheral device, the P5 Virtual Reality Glove, suitable for gaming and 3D virtual environments. The user moves their hand in front of a receptor 'tower', which contains two infrared sensors. They detect the visible LEDs on the glove (there's eight altogether), and convert them into an (x, y, z) position for the glove, and an orientation in terms of pitch, yaw, and roll. The pointing gesture recognition is achieved by simply projecting the hand position and direction, represented by yaw angle, on 2D plane. The position of each object in the scene is shifted to the same 2D plane. The salience of each object is proportional to the angle formed between the direction of the hand and the relative direction between object and hand position.

Each sample collected composed of a bag of words provided in the description, the sensory characteristics of multiple objects and a level of salience associated with each object. The salience can be regarded as an a priori estimate of the possible referent of the statement. Imagine that the robot should determine the degree of associability between the word red and the color of the object. If the estimated salience is correct in most cases, each example containing the word red contain at least one instance of the color red (RGB feature), but it will also contain instances of different colors that are an additional source of ambiguity. In order to correctly infer the meaning of the word red, the robot must be able to isolate from each example the proper instance, discarding all others. This problem in literature is known as *multi-instance learning*. In multi-instances learning the labels are only assigned to bags of instances (i.e. labels are not assigned to individual instances). In the binary case, a bag is labeled positive if at least one instance in that bag is positive, and the bag is labeled negative if all the instances in it are negative.

In our previous work [Dindo and Zambuto, 2009], a semantic distortion metric was used to select appropriate semantic category from several hypothetical ones. The meaning of each word (i.e. it's semantic category) was treated as a random variable and modeled with a multivariate Gaussian distributions. These distributions are estimated for each semantic class (shape, color, size). Taking the example above, the algorithm estimates a semantic category (Gaussian density) for each sensory channel (shape, color, size)

from the set of positive examples associated with the word red. Each of these categories represents a hypothesis about the possible meaning of the word, and a hypothetical association between the semantic class and word. Similarly, the algorithm estimates a probabilistic model from the negative examples associated with the word red, i.e. those examples where the word is not present. The semantic grounding is done with the semantic class (and the associated semantic category) that maximizes the semantic distortion measure between the two probabilistic models. The previous algorithm then consists of two basic steps: (a) the estimation of semantic categories and their negative models (background probability) for each acquired word and (b) the association of meaning-word obtained by probabilistic measures on the estimated probabilities.

In this work, we have maintained the same structure as the previous algorithm. Again, we first estimate the semantic categories and negative models and then use probabilistic methods to determine the correct association. The first difficulty, as already mentioned, is precisely in the estimation of densities: treating them with bags of instances and not with individual instances complicates the learning problem, which becomes multi-instance problem. The estimation process must take into account a priori information obtained from the attentional system (salience), and at the same time find the set of redundant instances for each class.

We present a new algorithm for learning semantic categories, inspired by some multi-instance learning techniques [Maron and Lozano-Pérez, 1998]. In this work, we also present a new algorithm for semantic association that, compared to the previous work, also integrates information related to the learned syntax, as well as those related to sensory observations alone (semantics). Words that belong to the same semantic class, must follow the same syntactic rules, and thus should belong to the same syntactic class. The system is schematically outlined in Fig. 4.1.

4.3.1 Semantic clustering as multi instances learning

The first phase of the algorithm deals with the estimation of semantic categories and negative models. For each word w recognized by the system, there is a set of training data, consisting of positive and negative examples, i.e. examples where the word is used or not used. Each sample consists of a number of instances and the degree of salience associated with them. For example, the word red, assuming we have three semantic classes, get three sets of positive examples, and three sets of negative examples, one for each class (Figure 4.2). We denote positive bags as $\mathbf{x}_n$, and the ith instance in that bag as $\mathbf{x}_{ni}$. Suppose each instance can be represented by a real-

valued feature vector. Likewise, $\mathbf{x}_n^-$ denotes a negative bag and $\mathbf{x}_{ni}^-$ is the ith instance in that bag. For each semantic class-word pair, we then estimate two probability density: the distribution of feature values conditioned on the presence of word $p(\mathbf{x}|c, w)$ (hypothetical semantic category) and the distribution of feature values conditioned on the absence of word $p(\mathbf{x}|c, \overline{w})$ (background distribution).

Learning background distribution

We want to estimate a parametric probability density $p(\mathbf{x}|c, \overline{w}, \boldsymbol{\theta}^-)$ from all negative bags within each class. Unlike the classical paradigm of multi-instance learning, we can not be sure that the bag contain only negative instances of the concept to be learned. For example, if we want estimate the negative model of the word red, we can use examples that describe green objects, but we can not be sure that the bags do not also contain instances of red object (that is, the process of attention may have estimated a high degree of salience for one red object next to the object described). We must carefully select the bags to be used for estimation of the negative model. A simple procedure to minimize this type of error is to select examples where the degree of salience is concentrated on few objects only.

We assume that the data points $\mathbf{x}_{ni}^-$ are drawn independently from the distribution. The likelihood function is given by:

$$p(\mathbf{X}^-|\boldsymbol{\theta}^-) = \prod_n \prod_i p(\mathbf{x}_{ni}^-|\boldsymbol{\theta}^-)$$

If we assume a unique Gaussian density, the Maximum Likelihood (ML) solution [Bishop, 2006] is:

$$\boldsymbol{\mu}^- = \frac{1}{N} \sum_n \sum_i (\mathbf{x}_{ni}^-)$$

$$\boldsymbol{\Sigma}^- = \frac{1}{N} \sum_n \sum_i (\mathbf{x}_{ni}^- - \boldsymbol{\mu}^-)(\mathbf{x}_{ni}^- - \boldsymbol{\mu}^-)$$

Learning positive distribution

A more difficult problem is to estimate a parametric model $p(\mathbf{x}|c, w, \boldsymbol{\theta}^+)$ from positive bags. As we know, a bag is labeled positive if at least one instance in that bag is positive. However, we do not know which instance is the positive one. The knowledge of positive instance in each bag is modeled by using a set of hidden variables, which are estimated using the Expectation Maximization algorithm. We denote positive bags simply as $\mathbf{x}$, and the ith

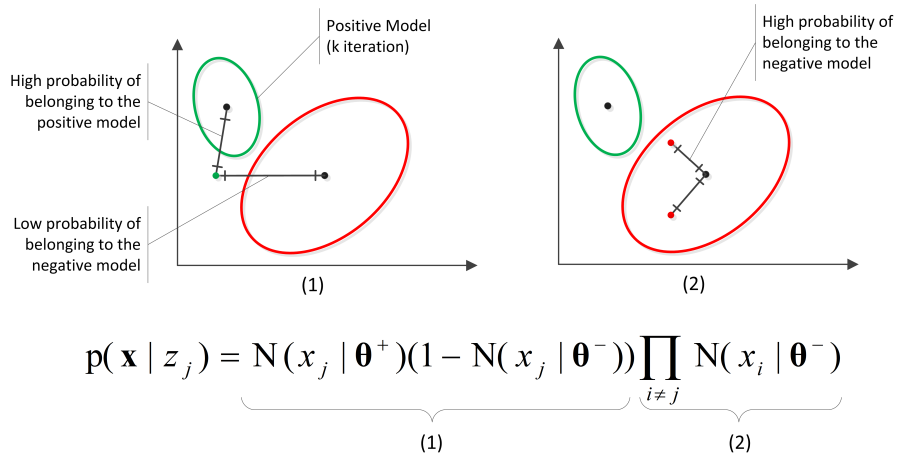


Figure 4.3: Positive model learning.

instance in that bag as $\mathbf{x}_i$. We suppose that each bag have same number of instances, I. We introduce a I-dimensional binary random variable $\mathbf{z}$ having a 1-of-I representation. There are I possible states for the vector $\mathbf{z}$. The value of z_i therefore satisfy $\sum_i z_i = 1$. The hidden variable z models the missing information: the learning process so try to estimate the semantic category and at the same time, approximately what is the proper instance for each bag. We can thus define the likelihood function:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^{I} p(\mathbf{x}|z_j)^{z_j} = \prod_{j=1}^{I} p(\mathbf{x}_1, ..., \mathbf{x}_I|z_j)^{z_j}$$

The likelihood function depends only on one of the instances in the bag. We can then rewrite the previous equation as follows:

$$p(\mathbf{x}_1, ..., \mathbf{x}_I|z_j) = \prod_{i=1}^{I} p(\mathbf{x}_i|z_j) \tag{4.1}$$

At this point, we must quantify the degree of "positivity" of the instance, which depends on two main factors: the instance should not belong to the negative model previously estimated and at the same time it must be like to at least one instance of any other positive example. We can rewrite the equation 4.1 as follows:

$$p(\mathbf{x}|z_j) = \mathcal{N}(\mathbf{x}_j|\boldsymbol{\theta}^+)(1 - \mathcal{N}(\mathbf{x}_j|\boldsymbol{\theta}^-)) \prod_{i \neq j} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\theta}^-) \tag{4.2}$$

Like Maron's *Diverse Density*, the equation 4.2 represents a measure of the intersection of the positive bags minus the union of the negative bags [Maron

and Lozano-Pérez, 1998]. By maximizing that measure, we can find the redundant points distribution (the desired concept). The equation 4.2 quantifies the possibility that a specific instance of the bag is positive and at the same time depends on the degree of negativity of the other instances, seeking instances farthest from the negative examples, but closer to other positive instances.

$$p(\mathbf{z}) = \phi(\mathbf{z}) = \prod_j \phi(z_j)^{z_j}$$

The ϕ function compute a prior estimate of "positivity" of each instance of a bag. In this work, we do not model this probability (attention). For each bag $\mathbf{x}$ of the training set, we know a I-dimensional real-value vector, $\boldsymbol{\phi}$. The value of ϕ_i satisfy $\sum_i \phi_i = 1$. The posterior probability integrates the salience of the individual instance, which depends on the attention process, with its degree of positivity, which depends on the entire training set and is therefore more generic. The posterior probability is defined as follows:

$$p(z_j|\mathbf{x}) = \frac{\phi_j p(\mathbf{x}|z_j)}{\sum_{i=1}^{I} \phi_i p(\mathbf{x}|z_i)} \approx \phi_j p(\mathbf{x}|z_j)$$

Now consider the problem of maximizing the likelihood for the complete data set $\mathbf{X}, \mathbf{Z}$:

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}^+) = \sum_n \sum_i z_{ni} \ln \mathcal{N}(\mathbf{x}_{ni}|\boldsymbol{\theta}^+)$$

During expectation step, we estimate the expected value of the variable z_{ni}.

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^+) \approx \prod_n \prod_j [\phi_{nj} p(\mathbf{x}_n|z_{nj})]^{z_{nj}}$$

$$E[z_{ni}] = \frac{\phi_{ni} p(\mathbf{x}_n|z_{ni})}{\sum_{j=1}^{I} \phi_j p(\mathbf{x}_n|z_{nj})} = \gamma(z_{ni})$$

We can now proceed as follows.

$$\boldsymbol{\mu}^+ = \frac{1}{N} \sum_n \left[\max_i \gamma(z_{ni}) \right] \mathbf{x}_{ni}$$

$$\boldsymbol{\Sigma}^+ = \frac{1}{N} \sum_n \left[\max_i \gamma(z_{ni}) \right] (\mathbf{x}_{ni} - \boldsymbol{\mu}^+)(\mathbf{x}_{ni} - \boldsymbol{\mu}^+)$$

$$N = \sum_n \left[\max_i \gamma(z_{ni}) \right]$$

Instead of using all the instances of each positive bag for density estimation, we use only the instance that maximizes the expected value on the hidden variable. In this way, we consider the hypothesis made initially, namely that each positive bag contains only one positive instance.

4.3.2 Word-to-meaning association

In the previous section we presented the algorithm used for the estimation of parametric distributions that describe the positive and negative instances for each semantic class. We must now associate each word with an estimated semantic category, and then force the system to make the more correct association. We can apply our previous algorithm and evaluate the degree of association between semantic class and word. We use a distortion measure (Bhattacharyya distance) between the positive and negative distributions as a measure of association between word and semantic categories. Once we have determined the semantic category that maximizes the probability measure for each word (treated as the more correct association), we can estimate a pseudo syntax that generalizes the results obtained. Exploiting only the sensory information to determine the word-to-meaning association still leads to partially correct results. Densities used are in fact estimated from extremely noisy data, and the association procedure will be successful only for those words whose training set is less ambiguous. We must try to integrate into the process other information that will enable us to correct those ambiguous cases (Figure 4.4).
One possibility is to add more examples in those training sets which are corrupted by more noise, and re-apply the learning algorithm. Another option is to use the position of words in the sentence along with the semantic information to improve the associations: words associated with similar perceptual categories will be included in same syntactic category and will follow same syntactic rules. But we must first solve the problem of the referent, which remains unknown. Our previous learning algorithm allows us to approximate partially correct language model that can be used together with the salience to determine the object of interest in the description.

Bootstrap word-to-meaning association

In our previous work, we used a distortion between the word-conditioned and background distributions as a measure of association between word and semantic categories. The Bhattacharyya distance measures the similarity of two probability distributions. The Bhattacharyya distance between two Gaussian distributions is defined as follows:

$$d_{bhat} = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T + \frac{1}{2}ln\left(\frac{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_1|\,|\boldsymbol{\Sigma}_2|}}\right)$$

The first term of equation gives the class separability due to the difference between class means, while the second term gives the class separability due

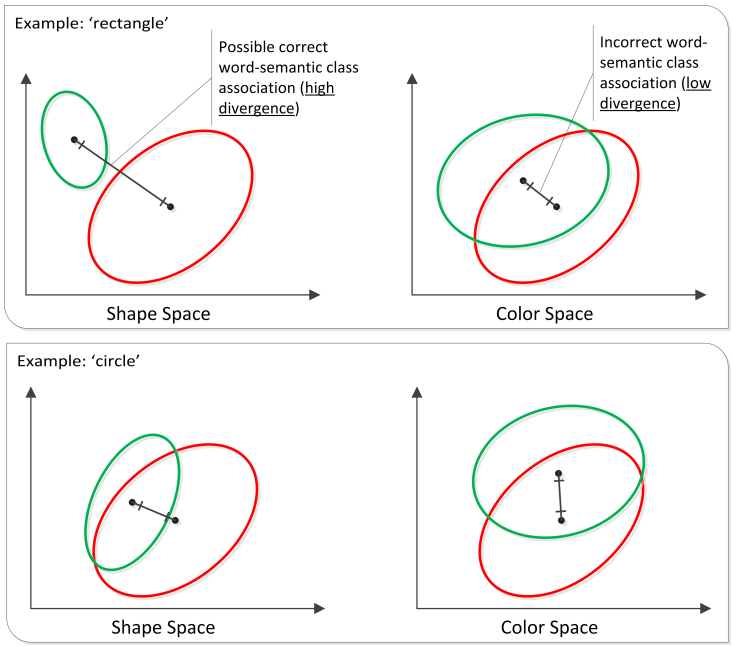


Figure 4.4: Two example of estimated positive and negative model. In the first case, we obtain correct association exploiting only semantic information. In the latter case, the semantic information don't give any additional information. and can lead to incorrect results.

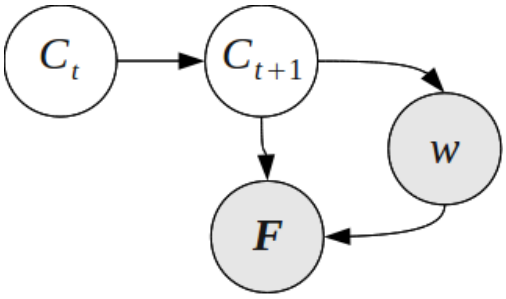


Figure 4.5: Word-to-meaning association: probabilistic model

to the difference of the class covariance matrices. Furthermore, the optimal Bayes classification error between the two class is bounded by the following expression: $\varepsilon \leq \sqrt{P_1 P_2} e^{-d_{bhat}}$. We refer to the upper bound of the error probability as the Bhattacharyya error.

In our case, the Bhattacharyya error provides *a measure of association between words and individual semantic categories*. A clustering algorithm estimates hypothetical semantic categories of words from co-occurring contexts and associates each word with semantic categories that maximize the classification error outlined above [1]. Categories below a fixed threshold are discarded, and the word is associated to a special ungrounded class. Ungrounded class gathers all the words that can not be directly grounded in sensory input. In our model, no semantic category (Gaussian density) is associated with these words. We have then defined a fitting function which measures the similarity of an utterance to an object based on Mahalanobis distance and saliency, which will be explained later. The object of the scene that minimizes this measure is selected as a possible referent of the sentence. This process can be repeated each time the language model change. In some cases, this process might fail by selecting the wrong referent.

Probabilistic model

Each example of the (positive) training set then consists of a sequence of words $w_{1:T_k}$ (utterance) and a set of features $\mathbf{F}_k = \mathbf{f}_k^1, \ldots, \mathbf{f}_k^M$ describing the alleged target object (M feature type or semantic class). C latent variable models the relationship between word w and one of the feature classes $\mathbf{f}^m$, and is time-dependent. We want to find the sequence of semantic classes (hidden variable) that gave rise to a certain sequence of words and a given

[1] We force the clustering algorithm to associate only one semantic category, without considering multi-class meanings (semantic categories that depend on several classes of perception)

set of features (observations). Figure 4.5 shows the graphical model that summarizes the dependencies between random variables used. The probability $p(C_{1:T_k}|w_{1:T_k}, \mathbf{F}_k)$ can be decomposed in a manner similar to HMMs, as follows:

$$p(C_0)\prod_{t=1}^{T_k} p(C_t|C_{t-1})p(w_t|C_t)p(\mathbf{F}_k|C_t, w_t)$$

As in Hidden Markov Models (HMM), we recognize the transition probabilities of semantic categories $\mathbf{A}$, the emission probability of the word given a particular semantic category $\mathbf{B}$, and the probability of the features given the semantic category and word. This last probability coincides with the semantic category $p(\mathbf{x}|c, w, \boldsymbol{\theta}^+)$ estimated at the previous step for each class-word pair.

$$p(\mathbf{F}_k|C_t = m, w_t = i) = \mathcal{N}(\mathbf{f}_k^m|\boldsymbol{\mu}_{mi}^+, \boldsymbol{\Sigma}_{mi}^+)$$

The discrete variable C can take M values, while w can take W values. We want to estimate the parameter of the model $\theta = \{\mathbf{A}^{M\times M}, \mathbf{B}^{M\times W}\}$. The transition probability $\mathbf{A}$ encodes a pseudo-syntax that depends on the semantic classes. The emission probability $\mathbf{B}$ measures the degree of belonging of a word to a particular semantic class. We used a modified version of Baum-Welch algorithm to learn the parameters of the model. In the expectation step, we calculate first the transition from class j to class i given a word s of the sentence and the set of features describing the target object F_k, as follows:

$$\epsilon_t(i, j) = p(C_t = i, C_{t+1} = j|w_t = s, \mathbf{F}_k) = \tag{4.3}$$

$$= \frac{\alpha_t(i)a_{ij}b_{js}\mathcal{N}(\mathbf{f}_k^j|\boldsymbol{\mu}_{js}^+, \boldsymbol{\Sigma}_{js}^+)\beta_t(j)}{\sum_i\sum_j \alpha_t(i)a_{ij}b_{js}\mathcal{N}(\mathbf{f}_k^j|\boldsymbol{\mu}_{js}^+, \boldsymbol{\Sigma}_{js}^+)\beta_t(j)} \tag{4.4}$$

then the probability of being in state i, given the observations sequence and the model:

$$\lambda_t(i) = p(C_t = i|w_t = s, \mathbf{F}_k) = \sum_j \epsilon_t(i, j)$$

Maximization with respect to $\mathbf{A}$ and $\mathbf{B}$ is easily achieved by using appropriate Lagrange multipliers with the following result:

$$a_{ij} = \frac{\sum_t \epsilon_t(i, j)}{\sum_t \lambda_t(i)}$$

$$b_{js} = \frac{\sum_{t,w_t=s} \lambda_t(j)}{\sum_t \lambda_t(j)}$$

The EM algorithm requires initial values for the parameters of the emission distribution. We can initialize these probability as follows:

$$a_{ij}^0 = \frac{1}{M}$$

$$b_{js}^0 = \frac{d_{bhat}^{js}}{\sum_s d_{bhat}^{js}}$$

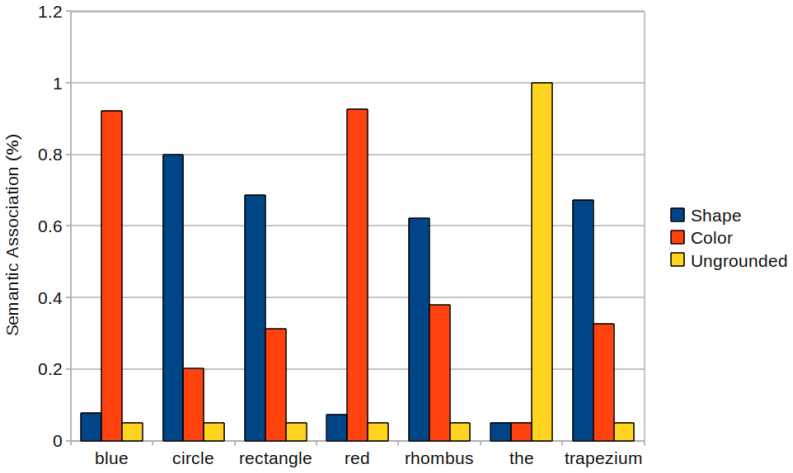
Note that the algorithm is not guaranteed to converge at the global maximum.

4.3.3 Syntactic Constraints

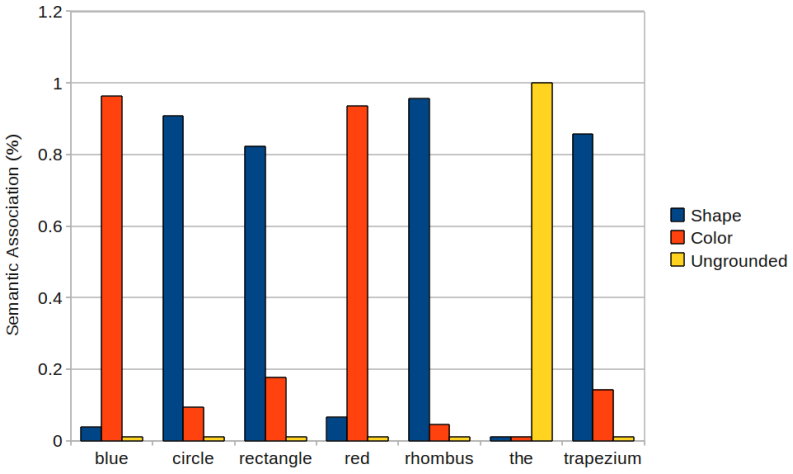
We also need to learn syntactic component that encodes word order constraints. Initially, we have described these constraints as a Markovian chain. Lexical items generated in the previous phase clustered words into groups that depend on the associated meaning. We use these groups, and associated probabilities, to construct a more general representation of syntactic constraints in terms of a finite state automaton (FSA) based on semantic classes. These FSA are used as the basis for a deterministic parser which identifies object phrases embedded in complex utterances.

4.3.4 Grounding relative spatial terms

To learn the meaning of spatial terms, we need to find all the objects to which the phrase refers. We consider only those phrases describing spatial relationships between two objects (i.e., "the rectangle above the red circle"). A deterministic parser based on previously acquired FSA, identifies object phrases embedded in complex utterances. We must decide which of two phrases refers to the target object, and which of the remaining objects in the scene should be linked to the other phrase. We have defined a fitting function which measures the similarity of an utterance to an object based on Mahalanobis distance and object salience, which will be explained later. Once the focal objects of the sentence has been determined, we can calculate the features that describe the spatial relationship between these. We must also encode the order of the phrases to learn the difference between spatial terms (i.e., difference between "above" and "below") and to calculate correctly the spatial features. The parser replace the object phrases with (ungrounded) labels to facilitate the task. The only information that really matters in our case, is which of the two sentences refers to the target object. Finally, the learning process that associate a meaning to the word describing a spatial



(a) Semantic association measured from Gaussian density estimated with multi-instance algorithm



(b) Semantic association obtained after having integrated the syntactic information.

Figure 4.6: Results of the word-to-meaning association algorithm. Words associated with similar perceptual categories will be included in same syntactic category and will follow same syntactic rules.

feature continues as previously described. New lexical items are simply added to the vocabulary of model, and the learnt FSAs are merged into a single syntactic model.

4.3.5 Understanding and generating descriptions

Thanks to the Mahalanobis distance, we can measure the degree of association ("semantic distance") between a visual feature and a semantic category previously acquired. For example, we consider the case of a sentence like "the red rectangle". The acquired lexical items allow the parser to assign a semantic category (and hence a Gaussian density) to each word of the sentence. Given the visual features of an object, we can calculate the degree of association between sentence and object's visual features as:

$$\frac{\sum_{i=1}^{T} \sqrt{(\mathbf{f} - \boldsymbol{\mu}_i)\boldsymbol{\Sigma}_i^{-1}(\mathbf{f} - \boldsymbol{\mu}_i)}}{\phi}$$

The object of the scene that minimizes this measure is selected as a possible referent of the sentence. We have already used the fitting function to find phrase-to-object matches. The same procedure can be used for understanding spatial clauses. The deterministic parser guides the extraction of spatial features which depends on the order of object phrases. The referent of the sentence is searched among all possible pairs of objects.

One way to facilitate the understanding of descriptions and decrease the computational burden is to use logical terms to represent the state of the world (i.e., knowledge about the properties and spatial relationships of objects), and then use query in a logic programming language for inferring the reference object. When the parser begins to analyse an utterance, it queries the system for the state of the world described in the form of a list of all atomic sentences that can be produced from available sensors at the time of query. For each object in the scene, some logical terms are generated through the acquired lexical items. In particular, the system generates a logical term for each visual feature extracted from a object by selecting the lexical item that minimizes the Mahalanobis distance for each semantic class. Deterministic parser has been modified to generate logical query to be submitted to the system. Some examples of possible queries are: "red(X)", "rect(X), above(X,Y), red(Y)", etc.

The generation of the description of an object is quite simple. We can generate a description as the most likely path of words (in the main FSA) that produces a given observation. Given a set of visual features extracted from an object we generate a description of object through a modified Viterbi-like

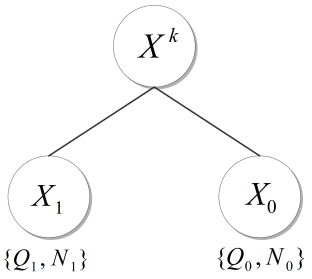


Figure 4.7: Disambiguation tree reconstruct the robot's decision process in choosing a question to disambiguate a context.

algorithm. We seek the sequence of words that minimizes the fitting function for that object, and follows the syntactic rules implicitly encoded in the FSA. We need a method to determine the length of a description. The generation algorithm include a stop criterion based on acquired ending probability.

4.4 Resolving perceptual ambiguities

Another problem we have addressed in this work is how to resolve the ambiguities contained in a description through simple verbal interactions based on yes/no questions. The robot does not use generic questions like "What is the colour of the object?", but specific questions directly related to the properties and spatial relationships of the objects in the observed scene. Obviously the questions will change depending on the context. Some questions would allow the robot to effectively clarify the ambiguities, while other might be too specific and will not contribute to the disambiguation process.

Disambiguation tree reconstruct the robot's decision process in choosing a question to disambiguate a context. They are essentially decision trees: the possible target objects are stored in the leaves, a disambiguation strategy is given by the path from the root to the object's leaf. The interior nodes can be questions about referent's proprieties itself, or relations to other objects. When the robot finds a reference ambiguity, it builds a disambiguation tree that resolves it. Figure 4.7 shows an example of disambiguation tree. Path to the left after question indicate that the predicate is satisfied (yes answer), while the right branch indicates that it is not (no answer). Each question consists of attempting to satisfy a logical predicate, and appropriate subtree is explored. This process continues until a leaf is reached and the ambiguities is resolved.

Disambiguation tree are costructed using the output of an ambiguous query (logic queries with more than one solution). The construction method

minimizes an entropy-based disambiguation measure. At each decision node, the algorithm chooses a possible question and splits the available objects into two groups. This process then occurs recursively until all objects are selected. Clearly, the algorithm must decide which of these question is most informative. We have used a entropy-based disambiguation measure, characterizing the average amount of uncertainty in a single question. This measure is computed as:

$$\frac{N_0}{N_1}H(X_0) + \frac{N_1}{N_0}H(X_1) + \sum_i C_i(X^k)$$

The first two members of the equation measures the uncertainty related to possible questions in the children nodes (future scenarios). We have calculated the uncertainty linked to the set of possible questions Q in a node, as the entropy calculated on a random variable X generated by this set. The algorithm selects one of the possible questions associated with the current node (represented by X^k) and split the set of objects into two groups. For each group generates all the possible questions Q_i ($i = \{0, 1\}$) that have not been previously selected in other high level decision node. The discrete random variable X_i encodes the information related to each question in set Q_i. We can compute a probalility mass function $p(X_i)$ as follows:

$$p(X_i = j) = \frac{< j >}{\sum_k p(X_i = k)}$$

where the counter operator $< ... >$ return the number of the objects that respond positively to that question. N_i represents the number of the objects of each group. We weight the first two terms with their ratio to ensure that the tree is always well-balanced and the algorithm does not choose questions that have a clear answer in first tree level.

The last term measures the uncertainty related to the question selected, depending on the encoding of the question in natural language (i.e. the number of words used) and its contextual ambiguity (i.e. ambiguity in the understanding of the sentence by a listener). The greedy algorithm selects the question that minimizes this measure based on entropy. The trees generated by the disambiguation algorithm, are well-balanced. Figure 4.8(b) shows an example of disambiguation tree generated by the algorithm. In this example, the first selected question is the term "blue(X)". The question "Is blue?" allows the system to divide the objects of the scene into two groups: blue objects and not blue objects. The construction method consider future scenarios, and hence possible question and ambiguities in future contexts, to select appropriate question. The method does not select the questions

Type	Feature	Description
Shape	a, s, b_x, b_y, t	deformable superellipse
Color	R, G, B	RGB color space
Area	A	superellipse area %
Spatial Relation	v_x, v_y	relative orientation

Table 4.1: Features extracted from visual sensory stream

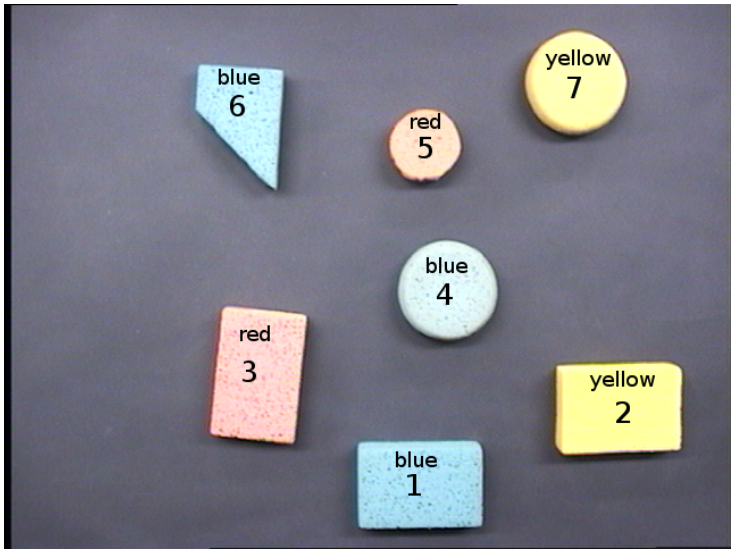
that have an immediate and obvious answer (i.e. "trapezium(X)"), but more general questions that allow us to isolate and partially resolve the perceptual ambiguity and reduce the number of questions to ask. The subsequent scenarios, as mentioned, are those with a lesser degree of ambiguity.

4.5 Evaluation

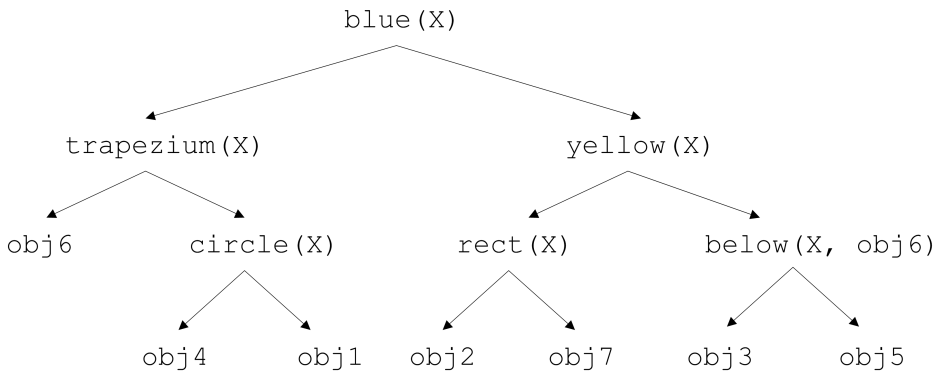
The experiments reported in this paper are based on the object description task. The description task consists of generating phrases which best describe target objects and must be context sensitive since often depend on the other objects in the observed scene. The variation of objects is limited to shape, color, size and position. Each training example is comprised of an utterance and a context representing the semantics of the word sequence. Utterances consist of phonetic transcripts of spoken sequences recorded by the auditory sensor. Context consists of visual sensory input which co-occurs with the utterance, and usually it contains instances of multiple semantic categories. The system's sensors consists of a color camera for color image processing (Sony EVI-D3I, f=5.4-64.8 mm, 12X, 320x240) and a microphone for speech dialogue. The experimental setting consists of a set of objects of different shape and color placed on a table. The camera is placed above the table and ensures a comprehensive view of the scene. An example is shown in the figure 4.8. The implementation of the system was tested on a PC with the following specifications: Intel Core 2 Duo T8100 2.10 GHz with 2 GB ram.

4.5.1 Experiments: training phase

As previously mentioned, the system has been tested on the NAO robotic platform. NAO is a humanoid robot equipped with Force Sensitive Resistors (FSR) located on the feet, sonars, bumpers, tactile sensors, an IR emitter/receiver, a stereo camera and a pair of microphones. The robot has a number of built-in machine vision modules used in the experimental setup. In addition, we have implemented a set of perceptual and motor schema for



(a) A sample image from training corpus



(b) A disambiguation tree inferred from the scene

Figure 4.8: Disambiguation trees are used to resolve the ambiguities contained in a description through simple Yes / No questions. A greedy algorithm generate well-balanced trees by minimizing an entropy-based measure.

(a) (b) (c)

Figure 4.9: An example of human-robot interaction via learned language model. The demonstrator asked the NAO to take the object to the left of a blue object. (a) NAO points the first object and says "Is the yellow rectangle?"; (b) NAO points the second object and says "Is the blue circle"; (c)NAO grasps the correct object.

basic behaviors such as *pointing* and *grasping*. A typical scene is provided in the Fig. 4.9.

A training corpus from two participants unfamiliar with the project has been collected. The acquisition process is, as already mentioned, interactive: the demonstrator stimulates the attention of the robot on one or more objects of the scene and verbally describes them. Participant were asked to generate simple utterances related to the observed scene such that a listener could later select the same target from the identical scene. Simple utterances contain reference to exactly one object (target object). The training corpus was composed of 266 utterances of which 171 are simple and 95 complex. In the first learning phase only simple utterance are used. The results of the algorithm are promising. Semantic clustering algorithm is able to isolate more than 80% of positive instances and then to estimate correctly the semantic category associated with the word. Figure 4.6(a) shows the degree of associability the word calculated in the first phase of the algorithm with respect to semantic classes. In most cases we can still get partially corrected results with our previous algorithm. However there are ambiguities, as in the case of the word "circle", which can be minimized by considering syntactic information. Figure 4.6(b) shows the final results obtained in the second phase of the algorithm.

4.5.2 Experiments: Human-Robot interaction

All concepts underlying acquired language model are used to initialize dynamic fluents as predicate calculus terms and update robot's knowledge base representing the state of the world from sensor data. Only the actions of the

robot can modify the values of the fluent associated with the objects. The demostrator can't modify the scene. For this reason, the knowledge base is updated after every action of the robot. The only fluent to be updated are those associated with spatial relationships between objects that change with every action. Every time the robot completes the move or grasp actions, updates the database with new spatial relationships. Obviously there will be some of the logic terms that will not vary at all (eg, color). We have tested the capabilities of the robot to understand the descriptions provided by the users and to conduct a dialog in case of ambiguities. The robot was given concrete instructions, such as "*Point the green object!*", or "*Grasp the object to the left of the yellow circle!*"[2]. The whole human-robot interaction is driven by gestures and language. An example of dialogue is shown below, while the robots actions are depicted in Fig. 4.9:

> *Robot*: looks at the object 1.
> *Human*: "NAO, grasp the object to the left of the blue one!"
> *Human*: points the object 1 (50%), 2 (20%), 3 (50%).
> *Robot*: looks at the object 3.
> *Robot*: "Is it the yellow rectangle?"
> *Robot*: points the object 3 (Fig. 4.9 left).
> *Human*: "No!"
> *Robot*: "Is it the blue circle?"
> *Robot*: points the object 2 (Fig. 4.9 center).
> *Human*: "Yes! That's right!"
> *Robot*: Grasps the blue circle (Fig. 4.9 right).

We used the disambiguation trees to solve some perceptual ambiguities present in the scene [Dindo and Zambuto, 2009]. In this experiment have been learned only the terms that refer to colors and shapes of objects in the scene. For the spatial relationships were used categories learned in our previous work [Dindo and Zambuto, 2009]. A set of external observers were judging the goodness of the system with respect to the following factors:

- Naturalness of the robot's linguistic and motor behavior;
- Differences between the expected behavior and that observed.

[2]In the present model, the meaning of verbs "*to point*" and "*to grasp*" is hand-coded, and it is not learned by the system. Future releases will address the problem of grounding dynamic terms through the same computational framework.

Participants	Generated by human	Generated by system
A	87,3 %	82,6 %
B	85,5 %	80,1 %
C	88,3 %	79,3 %
Media	87,1 %	80,5 %

Table 4.2: Results of an evaluation of human and machine generated descriptions.

About ten people were involved in a full-day evaluation session. The overall score was positive in about 80% of collected forms. While these results have no scientific foundation, they however show a positive impact of our computational model in a human-robot interaction system.

4.5.3 Experiments: Model accuracy

The goal of the evaluation phase is both to measure the semantic and syntactic accuracy of generated description, and to evaluate the performance of the system in the referent search problem. Two human participants unfamiliar with the technical details of the generation system participated in the evaluation. An evaluation program was written which presents images on a computer screen and allows participants to select a target object and to digit a description of a selected object. Different experiment was performed to measure the performance of the system.

In the first experiment participants were asked to describe an object of the observed scene providing both generic and detailed descriptions. All participants evaluated the same sets of images. Responses were evaluated by comparing the selected object for each image to the actual target object which was selected by the system. All perceptual information was integrated to find the correct referent of a description.

Participants were then asked to select the object which best fit the description generated by the system. Responses were evaluated by comparing the selected object for each image (the target object described by the system) to the actual target object which was selected by participant as the referent of the description. The collected data was used to measure the accuracy of system generated description (showed in Table 4.2). Participants were also asked to select a landmark object and to generate a detailed description of the target object in the scene. Responses were evaluated by comparing the selected objects for each image to the actual target object and landmark object which were selected by the system. Table 4.3 shows the result of evaluation of system understanding capabilities.

Participants	Training corpus	Testing corpus
A	91,0 %	85,7 %
B	87,2 %	87,2 %
C	93,0 %	84,3 %
System	84,2 %	79,2 %

Table 4.3: Results of an evaluation of machine understanding capabilities.

The system was able to describe scenes it has not encountered during training, and exhibit sequences of words which have never occurred in the training data. The results presented in this section demonstrate the effectiveness of the learning algorithms to acquire and apply grounded structures for the visual description task.

4.6 Conclusion

The algorithms presented in this thesis extends our previous work on the grounded language model learning. We focused on some limitations of the previous technique while maintaining the same algorithmic structure. In particular: (a) we endowed the system with a real model of attention and formalized a multi-instance learning algorithm for the acquisition of semantic categories, (b) we have improved the word-to-meaning association algorithm, by linking the choice not only to semantic information but also to syntactic constraints encountered, and (c) we have made demonstrator-robot interaction more natural.

However, a set of important questions still remain to be solved. As presented, the system learns “simple” concepts involving a single perceptual channel. Ongoing work is focused on learning complex concepts from the interaction data. The same computational framework will be employed recursively in order to assign meanings to words by hierarchically describing complex concepts as composed of simpler ones in a Bayesian network. Another issue is related to the process of learning and understanding verbs as words that usually involve an observable action. The work presented here represents the first steps in this direction.

Chapter 5

Conclusions

We started the journey by exploring the potential of the Bayesian inference for estimating the parameters of internal models. As opposed to classical view, optimizazion constraints was explicitly formulated as conditional dependencies in a joint distribution. This view provided the basis for the formulation of an efficient Belief Propagation (BP) algorithm for forward model estimation. The method was demonstrated on linear and non-linear feedforward networks. A link between the presented framework and gradient-based optimization methods was also established. Using the expectation propagation (EP) algorithm, a similar algorithm was derived for multiple internal models scenario.

The thesis went on by proposing a Bayesian model to efficiently handle the problem of action recognition using limited computational resources. This was achieved by using prior information over possible contexts and goal-directed actions, and by adopting an approximate inference procedure for tracking several competing hypotheses. In particular, the particle filter framework permitted to easily embed a huge number of internal models by exploiting all the prior knowledge available to the agent and to limit the activation of internal models. The system was exploited for recognizing single action and sequences of actions. Our experimental results showed that the system was able to correctly recognize demonstrated actions. The recognition of distal intentions is currently beyond the capability of our model. Future work will include the extension of the model for the recognition of distal intention through a hierarchy of inverse and forward models, where higher-level pairs encode increasingly abstract actions.

Finally, the thesis investigated how the robot can learn a grounded language model in order to be bootstrapped into communication. We presented a system which generates and understands spoken descriptions of objects and actions in visual scenes. We formalized a multi-instance learning algorithm

for the acquisition of semantic categories and a word-to-meaning association algorithm which intergates semantic information and syntactic constraints. We also addressed the problem of ambiguity resolution contained in a verbal description through simple interactions based on yes-no questions. However, a set of important questions still remain to be solved. As presented, the system learns "simple" concepts involving a single perceptual channel. Future work will focus on learning complex concepts from the interaction data. The same computational framework will be employed recursively in order to assign meanings to words by hierarchically describing complex concepts as composed of simpler ones in a Bayesian network.

Appendix A

Gaussian Identities

A.1 Gaussian identities

We define a Gaussian over x with mean a and covariance matrix A as the function

$$\mathcal{N}(x \,|\, a, A) = \frac{1}{|2\pi A|^{1/2}} \exp\{-\frac{1}{2}(x-a)A^{-1}(x-a)^T\} \tag{A.1}$$

with property $\mathcal{N}(x \,|\, a, A) = \mathcal{N}(a \,|\, x, A)$. We also define the canonical representation

$$\mathcal{N}[x \,|\, a, A] = \frac{\exp\{-\frac{1}{2}a^T A^{-1} a\}}{|2\pi A^{-1}|^{1/2}} \exp\{-\frac{1}{2}x^T A x + x^T a\} \tag{A.2}$$

with properties

$$\mathcal{N}(x \,|\, a, A) = \mathcal{N}[x \,|\, A^{-1}a, A^{-1}] \tag{A.3}$$

$$\mathcal{N}[x \,|\, a, A] = \mathcal{N}(x \,|\, A^{-1}a, A^{-1}) \tag{A.4}$$

The product of two Gaussians can be expressed as

$$\mathcal{N}[x \,|\, a, A]\,\mathcal{N}[x \,|\, b, B] = \mathcal{N}[x \,|\, a+b, A+B]\mathcal{N}(A^{-1}a \,|\, B^{-1}b, A^{-1}+B^{-1}) \tag{A.5}$$

$$\mathcal{N}(x \,|\, a, A)\,\mathcal{N}(x \,|\, b, B) = \mathcal{N}[x \,|\, A^{-1}a + B^{-1}b, A^{-1}+B^{-1}]\mathcal{N}(a \,|\, b, A+B) \tag{A.6}$$

$$\mathcal{N}(x \,|\, a, A)\,\mathcal{N}[x \,|\, b, B] = \mathcal{N}[x \,|\, A^{-1}a + b, A^{-1}+B]\mathcal{N}(a \,|\, B^{-1}b, A+B^{-1}) \tag{A.7}$$

Linear transformations in x imply the following identities

$$\mathcal{N}(Fx + f \,|\, a, A) = \frac{1}{|F|}\mathcal{N}(x \,|\, F^{-1}(a - f), F^{-1}AF^{-T}) \tag{A.8}$$

$$= \frac{1}{|F|}\mathcal{N}[x \,|\, F^T A^{-1}(a - f), F^T A^{-1} F] \tag{A.9}$$

$$\mathcal{N}[Fx + f \,|\, a, A] = \mathcal{N}[x \,|\, F^T(a - Af), F^T AF] \tag{A.10}$$

The division of two Gaussians in canonical form can be expressed as

$$\frac{\mathcal{N}[x \,|\, a, A]}{\mathcal{N}[x \,|\, b, B]} = \mathcal{N}[x \,|\, a - b, A - B] \tag{A.11}$$

The derivative of a Gaussian distribution can be expressed as

$$\partial_\theta \mathcal{N}(x \,|\, a, A) = \mathcal{N}(x \,|\, a, A) \left[-h^T \partial_\theta x + h^T \partial_\theta a - \frac{1}{2}\operatorname{tr}(A^{-1}\partial_\theta A) + \frac{1}{2}h^T(\partial_\theta A)h\right] \tag{A.12}$$

Bibliography

Salvatore M Aglioti, Paola Cesari, Michela Romani, and Cosimo Urgesi. Action anticipation and motor resonance in elite basketball players. *Nat Neurosci*, 11(9):1109–1116, Sep 2008.

Chris Baker, Josh Tenenbaum, and Rebecca Saxe. Bayesian models of human action understanding. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 99–106. MIT Press, Cambridge, MA, 2006.

Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, September 2009. ISSN 00100277. doi: 10.1016/j.cognition.2009.07.005. URL `http://dx.doi.org/10.1016/j.cognition.2009.07.005`.

D.A. Baldwin. Understanding the link between joint attention and language. *Joint attention: Its origins and role in development*, pages 131–158, 1995.

B. W. Balleine and A. Dickinson. Goal-directed instrumental action: contingency and incentive learning and their cortical substrates. *Neuropharmacology*, 37(4-5):407–419, 1998.

Moshe Bar. The proactive brain: memory for predictions. *Philos Trans R Soc Lond B Biol Sci*, 364(1521):1235–1243, May 2009. doi: 10.1098/rstb.2008.0310. URL `http://dx.doi.org/10.1098/rstb.2008.0310`.

H. Bekkering, A. Wohlschlager, and M. Gattis. Imitation of gestures in children is goal-directed. *Q J Exp Psychol A*, 53(1):153–164, Feb 2000.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

P. Bloom. Intentionality and word learning. *Trends in cognitive sciences*, 1 (1):9–12, 1997.

Paola Borroni, Marcella Montagna, Gabriella Cerri, and Fausto Baldissera. Cyclic time course of motor excitability modulation during the observation of a cyclic hand movement. *Brain Res*, 1065(1-2):115–124, Dec 2005. doi: 10.1016/j.brainres.2005.10.034. URL `http://dx.doi.org/10.1016/j.brainres.2005.10.034`.

Simone Bosbach, Jonathan Cole, Wolfgang Prinz, and Gï¿½nther Knoblich. Inferring another's expectation from action: the role of peripheral sensation. *Nature Neuroscience*, 8:1295–1297, 2005.

B. Calvo-Merino, D. E. Glaser, J. Grezes, R. E. Passingham, and P. Haggard. Action observation and acquired motor skills: an fmri study with expert dancers. *Cereb Cortex*, 15(8):1243–1249, Aug 2005. doi: 10.1093/cercor/bhi007. URL `http://dx.doi.org/10.1093/cercor/bhi007`.

Beatriz Calvo-Merino, Julie Grezes, Daniel E Glaser, Richard E Passingham, and Patrick Haggard. Seeing or doing? influence of visual and motor familiarity in action observation. *Curr Biol*, 16(19):1905–1910, Oct 2006. doi: 10.1016/j.cub.2006.07.065. URL `http://dx.doi.org/10.1016/j.cub.2006.07.065`.

P. Carruthers. Simulation and self-knowledge: a defence of the theory-theory. In P. Carruthers and P.K. Smith, editors, *Theories of theories of mind.* Cambridge University Press., Cambridge, 1996.

Nick Chater, Joshua B Tenenbaum, and Alan Yuille. Probabilistic models of cognition: conceptual foundations. *Trends Cogn Sci*, 10(7):287–291, Jul 2006. doi: 10.1016/j.tics.2006.05.007. URL `http://dx.doi.org/10.1016/j.tics.2006.05.007`.

Laila Craighero, Francesco Bonetti, Luca Massarenti, Rosario Canto, Maddalena Fabbri Destro, and Luciano Fadiga. Temporal prediction of touch instant during observation of human and robot grasping. *Brain Res Bull*, 75(6):770–774, Apr 2008. doi: 10.1016/j.brainresbull.2008.01.014. URL `http://dx.doi.org/10.1016/j.brainresbull.2008.01.014`.

D. J. Crammond. Motor imagery: never in your wildest dream. *Trends Neurosci*, 20(2):54–57, February 1997. ISSN 0166-2236. URL `http://view.ncbi.nlm.nih.gov/pubmed/9023871`.

G. Csibra and G. Gergely. 'obsessed with goals': Functions and mechanisms of teleological interpretation of actions in humans. *Acta Psychologica*, 124: 60–78, 2007.

Alessandro D'Ausilio, Friedemann Pulvermüller, Paola Salmas, Ilaria Bufalari, Chiara Begliomini, and Luciano Fadiga. The motor somatotopy of speech perception. *Current Biology*, 19(5):381–385, March 2009. ISSN 09609822. doi: 10.1016/j.cub.2009.01.017. URL `http://dx.doi.org/10.1016/j.cub.2009.01.017`.

K. Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

N. Daw and A. Courville. The pigeon as particle filter. In *Advances in neural information processing systems*, volume 20, 2007.

Y. Demiris and B. Khadhouri. Hierarchical attentive multiple models for execution and recognition (hammer). *Robotics and Autonomous Systems Journal*, 54:361–369, 2005.

D. Dennett. *The intentional stance*. MIT Press, 1987.

H. Dindo and D. Zambuto. Resolving ambiguities in a grounded human-robot interaction. *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium*, pages 408–414, 2009.

A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

K. Doya, S. Ishii, A. Pouget, and R. P. N. Rao, editors. *Bayesian Brain: Probabilistic Approaches to Neural Coding*. The MIT Press, 1 edition, January 2007. ISBN 026204238X. URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/026204238X`.

C. Fisher, D.G. Hall, S. Rakowitz, and L. Gleitman. When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. *The acquisition of the lexicon*, pages 333–375, 1994.

L. Fogassi, P.F. Ferrari, F. Chersi, B. Gesierich, S. Rozzi, and G. Rizzolatti. Parietal lobe: from action organization to intention understanding. *Science*, 308:662–667, 2005.

Karl Friston. A theory of cortical responses. *Philos Trans R Soc Lond B Biol Sci*, 360(1456):815–836, Apr 2005. doi: 10.1098/rstb.2005.1622. URL `http://dx.doi.org/10.1098/rstb.2005.1622`.

Vittorio Gallese and Alvin Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998. URL `http://www.unipr.it/%7Egallese/Gallese-Goldman%201998.pdf`.

Massimo Gangitano, Felix M Mottaghy, and Alvaro Pascual-Leone. Modulation of premotor mirror neuron activity during observation of unpredictable grasping movements. *Eur J Neurosci*, 20(8):2193–2202, Oct 2004. doi: 10.1111/j.1460-9568.2004.03655.x. URL `http://dx.doi.org/10.1111/j.1460-9568.2004.03655.x`.

Valeria Gazzola, Lisa Aziz-Zadeh, and Christian Keysers. Empathy and the somatotopic auditory mirror system in humans. *Curr Biol*, 16(18):1824–1829, Sep 2006. doi: 10.1016/j.cub.2006.07.072. URL `http://dx.doi.org/10.1016/j.cub.2006.07.072`.

G. Gergely and G. Csibra. Teleological reasoning in infancy: the naive theory of rational action. *Trends in Cognitive Sciences*, 7:287–292, 2003.

K. Gold, M. Doniec, C. Crick, and B. Scassellati. Robotic vocabulary building using extension inference and implicit contrast. *Artificial Intelligence*, 173 (1):145–166, 2009.

A. Goldman. *Simulating Minds: The Philosophy, Psychology, and Neuroscience of Mindreading*. Oxford University Press, 2006.

M.A. Goodrich and A.C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

R.M. Gordon. Folk psychology as simulation. *Mind and Language*, 1:158–171, 1986.

Scott T Grafton. Embodied cognition and the simulation of action to understand others. *Ann N Y Acad Sci*, 1156:97–117, Mar 2009. doi: 10.1111/j.1749-6632.2009.04425.x. URL `http://dx.doi.org/10.1111/j.1749-6632.2009.04425.x`.

Rick Grush. The emulation theory of representation: motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27(3):377–96, Jun 2004.

S. Harnad. The symbol grounding problem. *Physica. D*, 42(1-3):335–346, 1990.

M. Haruno, D. M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Comput*, 13(10):2201–2220, Oct 2001. doi: 10.1162/089976601750541778. URL `http://dx.doi.org/10.1162/089976601750541778`.

G. Herzog and P. Wazinski. VIsual TRAnslator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review*, 8(2):175–187, 1994.

Marco Iacoboni, Istvan Molnar-Szakacs, Vittorio Gallese, Giovanni Buccino, John C Mazziotta, and Giacomo Rizzolatti. Grasping the intentions of others with one's own mirror neuron system. *PLoS Biol*, 3(3):e79, Mar 2005. doi: 10.1371/journal.pbio.0030079. URL `http://dx.doi.org/10.1371/journal.pbio.0030079`.

Hiroshi Imamizu, Tomoe Kuroda, Satoru Miyauchi, Toshinori Yoshioka, and Mitsuo Kawato. Modular organization of internal models of tools in the human cerebellum. *Proc Natl Acad Sci U S A*, 100(9):5461–5466, Apr 2003. doi: 10.1073/pnas.0835746100. URL `http://dx.doi.org/10.1073/pnas.0835746100`.

T. Inamura, M. Inaba, and H. Inoue. Dialogue control for task achievement based on evaluation of situational vagueness and stochastic representation of experiences. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3:2861–2866, 2004.

M. Jeannerod. The representing brain: Neural correlates of motor intention and imagery. *Behav Brain Sci*, 17:187–245, 1994.

M. Jeannerod and J. Decety. Mental motor imagery: a window into the representational stages of action. *Curr Opin Neurobiol*, 5(6):727–732, Dec 1995.

Marc Jeannerod. *Motor Cognition*. Oxford University Press, 2006.

M. I. Jordan and D. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354, 1992.

M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–27, 1999.

C. Keysers and V. Gazzola. Integrating simulation and theory of mind: from self to social cognition. *Trends in Cognitive Sciences*, 11:194–196, 2007.

C Keysers and DI Perrett. Demystifying social cognition: a hebbian perspective. *Trends in Cognitive Sciences*, 8:501–507, 2004.

James M. Kilner, Karl J. Friston, and Chris D. Frith. Predictive coding: An account of the mirror neuron system. *Cognitive Processing*, 8(3), 2007.

J.M. Kilner, Y. Paulignan, and S.J. Blakemore. An interference effect of observed biological movement on action. *Current Biology*, 13:522–525, 2003.

J.M. Kilner, C. Vargas, S. Duval, S.J. Blakemore, and A. Sirigu. Motor activation prior to observation of a predicted movement. *Nature Neuroscience*, 7:1299–1301, 2004.

K.P. Kording and D.M. Wolpert. Bayesian decision theory in sensorimotor control. *Trends Cogn. Sci.*, 10:319–326, 2006.

F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2): 498–519, 2001.

Tai Sing Lee and David Mumford. Hierarchical bayesian inference in the visual cortex. *J Opt Soc Am A Opt Image Sci Vis*, 20(7):1434–1448, Jul 2003.

O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems*, pages 570–576. MIT Press Cambridge, MA, USA, 1998.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982. ISBN 0716715678. URL `http://portal.acm.org/citation.cfm?id=1096911`.

Ingo G Meister, Stephen M Wilson, Choi Deblieck, Allan D Wu, and Marco Iacoboni. The essential role of premotor cortex in speech perception. *Curr Biol*, 17(19):1692–1696, Oct 2007. doi: 10.1016/j.cub.2007.08.064. URL `http://dx.doi.org/10.1016/j.cub.2007.08.064`.

R. C. Miall and D. M. Wolpert. Forward models for physiological motor control. *Neural Networks*, 9(8):1265–1279, 1996. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/S0893-6080(96)00035-4.

A. Murata, L. Fadiga, L. Fogassi, V. Gallese, V. Raos, and G. Rizzolatti. Object representation in the ventral premotor cortex (area f5) of the monkey. *J Neurophysiol*, 78(4):2226–2230, Oct 1997.

Kevin P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002. URL `http://www.worldcat.org/oclc/52827959`.

Y. Niv, D. Joel, and P. Dayan. A normative perspective on motivation. *Trends in Cognitive Science*, 8:375–381, 2006.

E. Oztop, D. Wolpert, and M. Kawato. Mental state inference using visual control parameters. *Cognitive Brain Research*, 22:129–151, 2005.

Giovanni Pezzulo. Coordinating with the future: the anticipatory nature of representation. *Minds and Machines*, 18(2):179–225, 2008. doi: http://dx.doi.org/10.1007/s11023-008-9095-5.

Giovanni Pezzulo and Cristiano Castelfranchi. The symbol detachment problem. *Cognitive Processing*, 8(2):115–131, 2007.

Giovanni Pezzulo, Laura Barca, Alessandro Lamberti Bocconi, and Anna M. Borghi. When affordances climb into your mind: Advantages of motor simulation in a memory task performed by novice and expert rock climbers. *Brain and Cognition*, 73(1):68–73, 2010.

S. Pinker. *Language learnability and language development*. Harvard University Press Cambridge, Mass, 1984.

M. Ramirez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proc. AAAI-10*, Atlanta, USA., 2010.

Rajesh P. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat Neurosci*, 2(1):79–87, January 1999. doi: 10.1038/4580. URL `http://dx.doi.org/10.1038/4580`.

G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192, 2004.

G. Rizzolatti, R. Camarda, L. Fogassi, M. Gentilucci, G. Luppino, and M. Matelli. Functional organization of inferior area 6 in the macaque monkey. ii. area f5 and the control of distal movements. *Experimental brain research.*, 71(3):491–507, 1988. ISSN 0014-4819. URL `http://view.ncbi.nlm.nih.gov/pubmed/3416965`.

D. Roy. Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3):353–385, 2002.

D. Roy. Grounding Words in Perception and Action: Insights from Computational Models. *Trends in Cognitive Science*, 9(8):389–396, 2005.

A. N. Sanborn, T. L. Griffiths, and D. J. Navarro. Rational approximations to rational models: Alternative algorithms for category learning. *Psychological Review*, in press.

Daniel L. Schacter, Donna R. Addis, and Randy L. Buckner. Remembering the past to imagine the future: the prospective brain. *Nat Rev Neurosci*, 8(9):657–661, 2007. doi: 10.1038/nrn2213. URL `http://dx.doi.org/10.1038/nrn2213`.

L. Steels and F. Kaplan. AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32, 2001.

T. Suddendorf and M. C. Corballis. The evolution of foresight: What is mental time travel and is it unique to humans? *Behavioral and Brain Sciences*, 30(3):299–313, 2007.

M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28(05):675–691, 2005. ISSN 0140-525X.

M. Toussaint and C. Goerick. A bayesian view on motor control and planning. *From Motor Learning to Interaction Learning in Robots*, pages 227–252, 2010.

M. Tucker and R. Ellis. Action priming by briefly presented objects. *Acta Psychol.*, 116:185–203, 2004.

M. A. Umiltà, E. Kohler, V. Gallese, L. Fogassi, L. Fadiga, C. Keyers, and G. Rizzolatti. I know what you are doing. a neurophysiological study. *Neuron*, 31(1):155–65, 2001.

M.A. Umiltà, L. Escola, I. Intskirveli, F. Grammont, M. Rochat, F. Caruana, A. Jezzini, V. Gallese, and G. Rizzolatti. How pliers become fingers in the monkey motor system. *Proceedings of the National Academy of Science*, 105:2209–2213, 2008.

Cosimo Urgesi, Marta Maieron, Alessio Avenanti, Emmanuele Tidoni, Franco Fabbro, and Salvatore Maria Aglioti. Simulating the future of actions in

the human corticospinal system. *Cereb Cortex*, Jan 2010. doi: 10.1093/cercor/bhp292. URL http://dx.doi.org/10.1093/cercor/bhp292.

M. Wilson and G. Knoblich. The case for motor involvement in perceiving conspecifics. *Psychological Bulletin*, 131:460–473, 2005.

A. Wohlschlaeger, M. Gattis, and H. Bekkering. Action generation and action perception in imitation: An instance of the ideomotor principle. *Philosophical Transactions of the Royal Society of London*, 358:501–515, 2003.

D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.

D. M. Wolpert, Z. Gharamani, and M.I. Jordan. An internal model for sensorimotor integration. *Science*, 269:1179–1182, 1995.

Daniel M Wolpert, Kenji Doya, and Mitsuo Kawato. A unifying computational framework for motor control and social interaction. *Philos Trans R Soc Lond B Biol Sci*, 358(1431):593–602, Mar 2003. doi: 10.1098/rstb.2002.1238. URL http://dx.doi.org/10.1098/rstb.2002.1238.