The Emergence of Multimodal Concepts : From Perceptual Motion Primitives to Grounded Acoustic Words

Olivier Mangin

Université de Bordeaux
École doctorale de mathématiques et informatique

Thèse pour obtenir le titre de

Docteur en sciences
Spécialité informatique

Émergence de concepts multimodaux

De la perception de mouvements primitifs à l'ancrage de mots acoustiques

Présentée par
Olivier Mangin

Effectuée sous la direction de
Pierre-Yves Oudeyer

Soutenue le 19 mars 2014 devant le jury formé par

Pr. Angelo Cangelosi — *University of Plymouth*	Rapporteur
Dr. Jean-Luc Schwartz — *CNRS*	Rapporteur
Dr. Jacques Droulez — *Collège de France*	Examinateur
Dr. Emmanuel Dupoux — *École des hautes études en sciences sociales*	Examinateur
Dr. Manuel Lopes — *INRIA*	Examinateur
Dr. David Filliat — *ÉNSTA-ParisTech*	Examinateur

À mes parents,
À mes grand-parents,

Résumé

Cette thèse considère l'apprentissage de motifs récurrents dans la perception multimodale. Elle s'attache à développer des modèles robotiques de ces facultés telles qu'observées chez l'enfant, et elle s'inscrit en cela dans le domaine de la robotique développementale.

Elle s'articule plus précisément autour de deux thèmes principaux qui sont d'une part la capacité d'enfants ou de robots à imiter et à comprendre le comportement d'humains, et d'autre part l'acquisition du langage. À leur intersection, nous examinons la question de la découverte par un agent en développement d'un répertoire de motifs primitifs dans son flux perceptuel. Nous spécifions ce problème et établissons son lien avec ceux de l'indétermination de la traduction décrit par Quine et de la séparation aveugle de source tels qu'étudiés en acoustique.

Nous en étudions successivement quatre sous-problèmes et formulons une définition expérimentale de chacun. Des modèles d'agents résolvant ces problèmes sont également décrits et testés. Ils s'appuient particulièrement sur des techniques dites de *sacs de mots*, de factorisation de matrices et d'apprentissage par renforcement inverse. Nous approfondissons séparément les trois problèmes de l'apprentissage de sons élémentaires tels les phonèmes ou les mots, de mouvements basiques de danse et d'objectifs primaires composant des tâches motrices complexes. Pour finir nous étudions le problème de l'apprentissage d'éléments primitifs multimodaux, ce qui revient à résoudre simultanément plusieurs des problèmes précédents. Nous expliquons notamment en quoi cela fournit un modèle de l'ancrage de mots acoustiques.

Mots-clés apprentissage multimodal ; acquisition du langage ; ancrage de symboles ; apprentissage de concepts ; compréhension de comportement humains ; décomposition du mouvement ; primitive motrice ; décomposition de taches ; factorisation de matrice positive ; apprentissage par renforcement inverse factorisé

Université de Bordeaux
School of mathematics et computer science

Submitted in fulfillment of requirements for the degree of

Doctor of Philosophy
Specialized in Computer Science

The Emergence of Multimodal Concepts

From Perceptual Motion Primitives to Grounded Acoustic Words

Presented by
Olivier Mangin

Completed under the supervision of
Pierre-Yves Oudeyer

Defended on March, the 19th 2014 in front the committee composed of

Pr. Angelo Cangelosi — *University of Plymouth*	Reviewer
Dr. Jean-Luc Schwartz — *CNRS*	Reviewer
Dr. Jacques Droulez — *Collège de France*	Examiner
Dr. Emmanuel Dupoux — *École des hautes études en sciences sociales*	Examiner
Dr. Manuel Lopes — *INRIA*	Examiner
Dr. David Filliat — *ÉNSTA-ParisTech*	Examiner

Abstract

This thesis focuses on learning recurring patterns in multimodal perception. For that purpose it develops cognitive systems that model the mechanisms providing such capabilities to infants; a methodology that fits into the field of developmental robotics.

More precisely, this thesis revolves around two main topics that are, on the one hand the ability of infants or robots to imitate and understand human behaviors, and on the other the acquisition of language. At the crossing of these topics, we study the question of the how a developmental cognitive agent can discover a dictionary of primitive patterns from its multimodal perceptual flow. We specify this problem and formulate its links with Quine's indetermination of translation and blind source separation, as studied in acoustics.

We sequentially study four sub-problems and provide an experimental formulation of each of them. We then describe and test computational models of agents solving these problems. They are particularly based on bag-of-words techniques, matrix factorization algorithms, and inverse reinforcement learning approaches. We first go in depth into the three separate problems of learning primitive sounds, such as phonemes or words, learning primitive dance motions, and learning primitive objective that compose complex tasks. Finally we study the problem of learning multimodal primitive patterns, which corresponds to solve simultaneously several of the aforementioned problems. We also details how the last problems models acoustic words grounding.

Keywords multimodal learning; language acquisition; symbol grounding; concept learning; human behavior understanding; motion decomposition; motion primitive; task decomposition; nonnegative matrix factorization; factorial inverse reinforcement learning; developmental robotics

Contents

A developmental robotics perspective **1**

1 Complex and simple, whole and parts **7**
1.1 Understanding and imitation of human behaviors 8
1.2 Structured representations for complex motions 10
1.3 Language acquisition and multimodal learning 13
1.4 Important questions 16

2 Technical background **19**
2.1 Nonnegative matrix factorization 19
2.1.1 Problem description 19
2.1.2 Basic algorithms 22
2.1.3 Variants and improvements 25
2.2 Inverse reinforcement learning 26
2.2.1 Background: reinforcement learning 26
2.2.2 What is inverse reinforcement learning? 29
2.2.3 Algorithms for inverse reinforcement learning 33

3 Learning a dictionary of primitive motions **37**
3.1 Combination and discovery of motion primitives 38
3.1.1 What does combination mean? 38
3.1.2 Motion representations 39
3.1.3 Algorithms to decompose observed motions 42
3.2 Histograms of motion velocity 43
3.3 Discover simultaneous primitives by NMF 45
3.3.1 The choreography data 46
3.3.2 Algorithm 47
3.3.3 Evaluation 48
3.3.4 Results 50
3.4 Concluding perspectives 53

4 Learning a dictionary of primitive tasks **55**
4.1 Previous work 56
4.1.1 Inverse feedback and reinforcement learning 57
4.2 Factorial inverse control 58
4.2.1 Problem definition and algorithm 58
4.2.2 Experiments 62
4.2.3 Discussion 64

4.3 Factorial inverse reinforcement learning . . . 66
4.3.1 Multi-task inverse reinforcement feature learning . . . 66
4.3.2 Experiments . . . 67
4.3.3 Discussion . . . 72

5 Learning a dictionary of primitive sounds 75
5.1 Models of language acquisition . . . 75
5.2 Hierarchical clustering of basic sounds . . . 77
5.2.1 Background and principle . . . 78
5.2.2 Presentation of the framework . . . 78
5.2.3 Implementation . . . 79
5.2.4 Experimental scenario . . . 84
5.3 HAC representation . . . 86
5.3.1 Codebooks . . . 87
5.3.2 Histograms of co-occurrences . . . 87
5.4 Learning words with NMF . . . 88
5.5 Conclusion . . . 89

6 Multimodal learning 91
6.1 Multimodality in perception and learning . . . 91
6.2 Related work . . . 93
6.3 Experimental setup . . . 95
6.4 NMF for multimodal learning . . . 98
6.4.1 Learning a dictionary of multimodal components . . . 98
6.4.2 NMF to learn mappings between modalities . . . 99
6.5 Data and representation . . . 100
6.5.1 Motions . . . 101
6.5.2 Sounds . . . 101
6.5.3 Images and videos . . . 101
6.6 Experiments . . . 102
6.6.1 Learning semantic associations . . . 102
6.6.2 Learning words in sentences . . . 109
6.6.3 Emergence of concepts . . . 113
6.7 Conclusion . . . 115

7 Discussion and perspectives 119

A Non-negative matrices and factorization 125
A.1 Non-negative matrix theory . . . 125
A.1.1 Base definitions and notations . . . 125
A.1.2 Taxonomy of non-negative matrices . . . 126
A.1.3 Perron-Frobenius theorem . . . 127
A.2 Ambiguity in the problem definition . . . 128
A.2.1 Generative model . . . 128
A.2.2 Representation of simplicial cones . . . 129
A.2.3 First case: linearly independent generators . . . 129
A.2.4 Second case: $\mathrm{rk}(W) < K$ . . . 130

B Datasets 131
B.1 The Acorns Caregiver dataset . . . 131
B.2 The first choreography dataset . . . 131

B.2.1 Description of the data 132
B.3 The second choreography dataset 132
B.3.1 Description of the data 134

C Code **135**

A developmental robotics perspective

Late nineteenth and twentieth centuries have witnessed major scientific discoveries in several fields: neurosciences, brain imaging, but also the study of phenomenology by philosophers, the development of psychology, the invention of the theory of computation, and further development of the computer science. It was followed by the technical revolutions that lead to modern computers and robots, and mathematical developments driven by new applications such as machine learning. All these elements played an essential role in the advent of the understanding of cognition and intelligence, in such a way that *cognitive sciences*, the name given to that knowledge and fields of study, is a vastly multidisciplinary domain, that includes computational and robotics models of cognition.

Among these discoveries, develpmental psychology has brought attention on the processes that give rise to intelligence: instead of trying to understand directly the structure of an adult mind, it examines the mechanisms that shape and organize intelligence, starting from early childhood. Later, after a large part of research in both cognitive sciences and artificial intelligence had been mainly focused on the problem of understanding or imitating the adult mind, a similar shift in methodology also appeared in artificial intelligence and robotics. However the idea that the human mind should be studied through the processes of its development and maturation was already clear in Turing's mind sixty years ago.

> " Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child. "
>
> Alan Turing, Computing machinery and intelligence (**?**)

Although Turing was right about the importance of modeling child learning, he was wrong saying the child starts with a *blank sheet*. Developmental and social robotics therefore studies developmental mechanisms that can guide and constrain the learning of robots, animals, and infants. The focus thus moves from building

intelligent robots to a closer study of the behavior and learning mechanisms that make the interaction between these robots and their environment evolve towards structured interactions with persistent patterns. In other words this paradigm pertains to a systemic approach of cognitive development that is directly grounded in Piaget's thoughts.

> « L'intelligence ne débute ainsi ni par la connaissance du moi ni par celle des choses comme telles, mais par celle de leur interaction, et c'est en s'orientant simultanément vers les deux pôles de cette interaction qu'elle organise le monde en s'organisant elle-même.[1] »
>
> Jean Piaget, *La construction du réel chez l'enfant* (**?**)

Developmental robotics is also often denoted as *epigenetic robotics*: a name that emphasises the role of the robot history and environment in determining its final state, in opposition to behaviors fully determined by its initial programming. The motivations behind developmental and social robotics come from two goals, namely building better robots and understanding human development, that are very different in nature, but pointing toward the same area of research.

Firstly developmental robotics is motivated by building better robots. Indeed, science fiction has been exploring for some time now all the possible impacts of robots on our every day life and spreading the idea that robots could, as tools or companions, be a great improvement to our quality of life, provided that security and ethical issues are well dealt with. Actual robots are however still mainly confined to factories or have to be remotely controlled by human operators. The truth is that robots currently are only capable of very poor adaptivity to unpredictable environments they have not been specifically programmed for. Real human environment are such environments and thus this limitation constitutes a strong obstacle to deployment of robots in every day life applications such as domestic assistance or human-robot collaborative work. One reason of these limitations is that the programmer cannot take explicitly into account all possible environments and situations the robot might face. A promising approach to make such programming possible is to implement basic mechanisms that make the robot capable of adapting its behavior throughout its discovery of its environment.

The second motivation is that robots could help us understand the human brain. Robots actually constitute a unique tool to model some biological, psychological or cognitive processes and systems. This motivation lead to some early robotics realizations that had a great impact on biological scientific communities. As one early example, In 1912, Hammond and Miessner developed a robot called *electric dog* (**?**), that happened to be a good model of Jacques Loeb's theory to explain phototropism in insects. The robot had a major impact on the scientific acceptance and diffusion of Loeb's ideas (see **?**). Building robots as models of the child development is an analogous methodology that can help exploring the mainly unresolved question of how children intelligence develops before reaching the adult state. For that purpose developmental robotics research complements the work done in other disciplinary fields that also focus on understanding the functioning of the adult human brain.

[1] *"Therefore intelligence does not start from self-awareness neither from the awareness of its objects in their own, but from the interactions between both. Instead, intelligence organizes the world by organizing itself simultaneously between both sides of that interaction."*

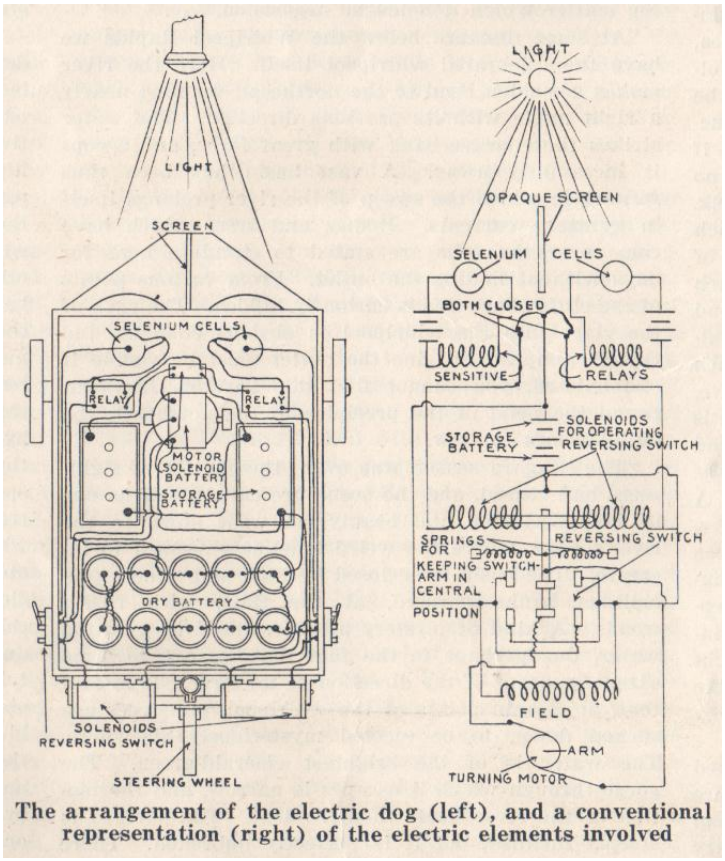


Figure 1: Illustration of the electric dog from **?**.

Reviews on advances in developmental robotics can be found in work from **??**. Among the many questions that are studied by developmental robotics two are of greater importance for this work. First, understanding the mechanisms enabling language acquisition by children is still subject to many open scientific questions (see **?**). Furthermore, building robots with language acquisition capabilities is a promising way of improving human robot or human computer interaction. Although there exist speech acquisition systems and natural language interfaces, these do not feature the adaptability required by the great variety in user preferences and languages. Most of these systems are built for a few target languages and user cases, that have nothing to do with the variety of words and skills people even enjoy teaching to their dog. Furthermore many systems do not focus on the transmission of concepts through natural interaction and the grounding of natural interaction on these concepts, although they are fundamental mechanisms of the emergence of communication between humans or humans and pets. The range of skills that a user would expect from a robot is also very wide and made even wider by the addition of user preferences. Similarly, natural teaching of robots seems a much more plausible solution than specific engineering for each use case; one way to achieve such teaching is known as *imitation learning* or *robot programming by demonstration*. Indeed, to have humans directly demonstrate tasks to a robot, or shaping the robot behavior to

fit their preferences, is much more accessible than actually programming the robot. Furthermore user preferences are not always easy to formulate, either using robots currently very poor language capabilities, or programming languages. While many techniques have been developed toward imitation learning for robots, most of them still target the single task setup. Such limitation is problematic, not only because robots are not necessarily expected to achieve only one task, but also because what seems to be a single task often is, after closer inspection, a combination of several tasks. Also, typical teaching to children does not go directly to complex tasks but through the learning of skills of increasing complexity: such a progressive trajectory might also be beneficial to teaching robots. Furthermore, common skills might be seen as prerequisites to learning complex ones; being able to re-use such skills, as already learnt by a robot, is also a promising way to ease the skill acquisition process.

In this thesis, we embrace the developmental robotics approach and consider the learning and developments that occurs directly on top of the sensori-motor perception. We particularly explore the questions related to the decomposition of motions as well as multimodal perceptual signals and therefore target several aspects of perception and language acquisition. Decomposing the complex into simple parts and composing simple building blocks into complex things are actually processes of great interest for the aforementioned questions. The acquisition of language is a very intuitive example of that decomposition that the structure of language itself implements: spoken language is made of sentences that are composed of words, which, themselves, are produced by sequencing phonemes. On the other side, the study of language acquisition by children shows that they first learn to recognize and produce phonemes, before going to words, and then sentences of increasing complexity (**?**). Similar observations on the production of actions suggests that children first learn to grasp before combining this skill with placement, and before they start building piles of toys. The motivation for roboticists is then to build mechanisms that make a robot capable of similar combinations of simple skills it already masters into more complex ones. Indeed, while it is now possible to teach various tasks to a robot, the evolution over time of the number of tasks a robot masters is typically linear: the time required to learn each new task does not really decrease as experience is accumulated by the robot, even if the task has a lot in common with a task already learnt. Achieving better re-use of knowledge is thus a promising way of improving the amount of tasks a robot can learn and achieving life-long learning (as claimed by **?**): exploiting the combinatorial structure of skills and knowledge makes it possible to learn new skills by efficiently combining mastered competences (see **??**).

From a wider perspective, the question of how learning can happen in an open-ended perspective is well identified as a major challenge for developmental robotics (see **??**). Despite that objective being clearly identified, most experiments in that fields, for example those involving between tens and thousands of repetitions of single grasping tasks, rarely last more than a few hours or days: much less than the scales of animal lives. Precisely, the mechanisms that could drive the shift from simple to complex in such learning, as observed on children, are still not well understood. Often, the tasks on which domestic robots are expected to be used, are too complex to be teachable, even to a state of the art robot; an important part of that complexity actually comes from the highly variable and complex nature of the environment in which they take place.

Computational approaches often implement a notion of primitive elements that

models the simple to complex approach. This follows the intuition of cumulative learning: a learning system gradually acquires a lexicon of elements. At first it acquires elements of very low complexity; then this complexity gradually increases as learnt elements can be combined into more complex ones, that become themselves parts of the lexicon (see **?**, sec. 3 and 4). This model is clearly inspired by the structure of language where the lexicon is initially populated by words before including word groups, propositions, and sentences of increasing complexity. The same idea have been used to model motion and is often behind the notion of motor primitives, as used by roboticists (see the discussion by **?**). A similar idea have been studied extensively in the field of machine learning and often applied to vision (**??**); it is named *dictionary learning*. Therefore one motivation behind the work in this thesis was to explore the application of these ideas and the large literature of associated techniques to the aforementioned questions of developmental robotics.

Despite being quite intuitive, the idea that learning systems first learn basic and local elements, such as words or short and precise motions, and then combine them into complex knowledge, is neither necessarily how it happens for children nor the only way to build artificial learning systems. Indeed such an approach, denoted as *compositional* puts the ability to segment complex motions or sequences into small parts as a prerequisite to learn these parts. As explained further these capabilities often corresponds to solving quite difficult or ambiguous problems. On the other hand the *teleological* approach (**?**) achieves first a global or holistic representation, that enables basic interactions with the world, before understanding the details and parts that compose for example the motion or sentence. As explained further, this thesis provides models of the learning of perceptual components that follows the holistic to decomposed pattern.

Chapter 1 introduces more precisely the central questions studied in this thesis. It explains how concrete issues from the fields of imitation learning, programming by demonstration, human behavior understanding, the learning of representation, structure learning, language acquisition, and multimodality connects with this work. That chapter identifies more precisely three central issues studied in this thesis: “How can the intuition about *simple* and *complex* be made explicit and implemented on a robot or an artificial cognitive system?”, “How can primitive elements emerge or be discovered through interaction of the agent with its physical and social environment?”, and “What mechanisms can overcome the intrinsic ambiguity and indeterminacy that is characteristic of many approaches regarding the learning and emergence of these primitive elements?”

In chapter 2, background on the techniques and algorithms used in this work is provided. It first introduces in details the family of nonnegative matrix factorization algorithms as well as related mathematical theories, but also other affiliated algorithms. Then an introduction to the domain of inverse reinforcement learning and inverse feedback learning is provided, on which new algorithms are grounded in chapter 4.

Chapter 3 introduces contributions of this thesis to the field of motion primitives in more detail (**?**). It explains why it is important to take into account the simultaneous combination of motion primitives, and provides an algorithm and a dataset to illustrate these ideas. Finally it discusses the evaluation of that algorithm with respect to a linguistic weak supervision.

Chapter 4 explores similar questions in the space of intentions that often underly actions: following inverse optimal control and inverse reinforcement learning, demonstrations of actions can be modelled by a demonstrator's intention, that takes the form of an objective function, also called task. Chapter 4 derives new algorithms (one of which was presented in **?**) to decompose a demonstrator's behavior in the task space instead of in the action space.

Chapter 5 reviews methods used to discover primitive acoustic elements that can form basis for word representations. Two approaches are presented, one from **?** based on a hierarchical clustering algorithm, and an other one from **?**, re-used in the following.

Finally chapter 6 brings together the work from chapters 3 and 5 to provide a multimodal learning algorithm (first presented in **?**) that models language grounding and the acquisition of abstract semantic concepts. We explain how that algorithm can model the simultaneous learning of components in several modalities (two or three in the experiments), and of their semantic relations. Furthermore, in the case where one modality contains spoken sentences, we demonstrate that the system focuses more precisely on parts of sentences that bear the semantics, thus illustrating a form of word acquisition, where that segmentation of words is more a consequence than a prerequisite of acquiring semantic knowledge.

A discussion of the contributions and perspectives introduced by this thesis is provided in chapter 7.

Chapter 1

Complex and simple, whole and parts

The main focus of this work is on the learning by a cognitive agent of dictionaries of basic or primitive elements from various potentially multimodal perceptual signals such as motion, images, or sound, which includes language, but also action. By primitive elements we mean, for example, phoneme-like primitive sounds, primitive dance gestures such as raising an arm, primitive objectives in complex tasks such as reaching a body configuration, or patterns in multimodal perception that ground semantic concepts. In the following these elements are denoted as *primitive* elements. This notion does not imply that such element are atomic or indivisible, but rather that they may be combined together to form more complex elements. As an example, if the elements are vectors and combination means linear combination, these elements may form a basis in the sense of linear algebra. However, as discussed further, linear combination is far from the only possible combination.

We claim that the study of primitive elements, together with their combinatorial properties and the algorithms to learn them, is of great interest for developmental robotics in two main aspects. First, composite representations of perception and actions for robots are promising ideas toward overcoming the limitation of many current robotic platforms to a single task in a single context. Indeed some of these platforms have an approach equivalent to learning by heart a whole sentence from many examples, whereas focusing on learning a dictionary of words together with a grammar would enable generalization to new sentences. We believe that, not only robots could benefit from this approach by gaining better versatility in their ability to re-use skills, but composite representations might be more understandable by humans and ease the process of programming or interacting with robots. A simple example of this idea is given by **?** who demonstrate how one can separately teach a robot to pour watter and to hold a glass, before having the robot merge the two tasks to fill a glass of water[1]. Also, building algorithms that are able to learn such composite elements is an attempt to model aspects of human perception that features such properties, and eventually also its limitation. For instance **??** explain that during early language

[1] A video of this experiment can be found on Sylvain Calinon's home page (`http://calinon.ch/showVideo.php?video=11`).

acquisition infants learn classes of sounds that are functionally equivalent, but also that this process makes them unable to differentiate between sounds from the same class: not only the brain learns some primitive sounds but these elements later play a fundamental role in the acoustic perception. In that particular example, the learnt primitive elements not only feature combinatorial properties in the way they are later combined into words, but also become, to a certain extent, atomic elements of perception.

In the remaining of this chapter the relation of this topic with various subjects of developmental robotic, but also psychology and machine learning, is explained in more details.

1.1 Understanding and imitation of human behaviors

Imitation has long been identified as a central mechanism for cultural transmission of skills in animals, primates, and humans, but also for the development of communication (see **???**). It is indeed not only an example of learning from social signal, but also a central aspect of learning to be social. Back to a robotic perspective, programming by demonstration refers to the transposition of the mechanism of learning by imitation as observed in humans to robots. The word programming actually suggests that teaching a skill to a robot by providing demonstrations to that robot may remove the need to actually program the robot, which currently requires advanced technical skills. Another important motivation of such an approach is that tasks are sometimes too complex to be reasonably described, even using natural language, or correspond to preferences of a user that are not fully conceptualized. In addition to learning the whole task from imitation, it is possible to use demonstrations together with a feedback signal and self-refinement of skills (see for example **?**) as well as self-exploration. Finally, designing systems capable of human behavior understanding, even without reproducing such behaviors, is a research field that faces issues very similar to the one presented in this section.

Although children seem to be capable of learning by imitation very easily, the task turns out to be more complicated on a closer examination. As is often the case, the difficulties are easily observed once one try to program a robot imitator. The first important issue faced by the imitator is known as the correspondence problem: how can the imitator relate the motions on the demonstrator's body to its own body? The problem is even made more complicated by the fact that the demonstrator's body is not identical to the imitator's one and might actually be very different. **?** formalize this issue as the one of "finding a *relational homomorphism* between the two bodies".

The correspondence problem might be overcome by using hard coded mappings between the demonstrator's and the imitator's bodies or even by directly demonstrating the skill on the imitator body, a techniques called *kinesthetic demonstration*. However the question of how this capability emerges in children is a very interesting one; some models of early imitation have been developed on that question by **?**. **?** have explained how a *homeostastatic* mechanism, that is to say that tries to reduce the error between prediction and observation, can be sufficient to generate imitation

behaviors. Following that idea **?** showed through a robotic experiment how imitation of facial expressions by a robot can emerge from a predictor that was learnt while the human was first imitating the robot. **?** also provide a model of imitation as emerging from intrinsic motivations: a progress based model of curiosity can drive a robot toward imitation as an efficient learning strategy.

The ability of the brain to relate actions of others to its own has been shown to have a neural manifestation in the existence of *mirror neurons*. Rizzolati and his collaborators have indeed discovered neurons in the premotor cortex of monkeys that fire both when the monkey performs a specific action or observes someone else performing the same action (**?**). Strong evidence from neuro-imaging suggests the existence of areas with similar functions in the human brain. Their discoveries suggest that the ability to recognize an action done by someone else is related, from a neurone point of view, to the one of producing the same action.

The correspondence problem is not the only difficulty brought by imitation learning. Should the exact motions of the demonstrator be reproduced? Or should the imitator rather try to solve the same task? But what is that task? Should the imitator try to fit some kind of preference of the demonstrator while achieving the task? Is every action of the demonstrator relevant to the imitator? Such questions are often summarized as "What, Who, When, and How to imitate?" Although they seem naturally answered by children, it is not clear how to make these choices. See also in fig. 1.1 the *strategy triangle* from **?** that illustrates some of these modelling choices.

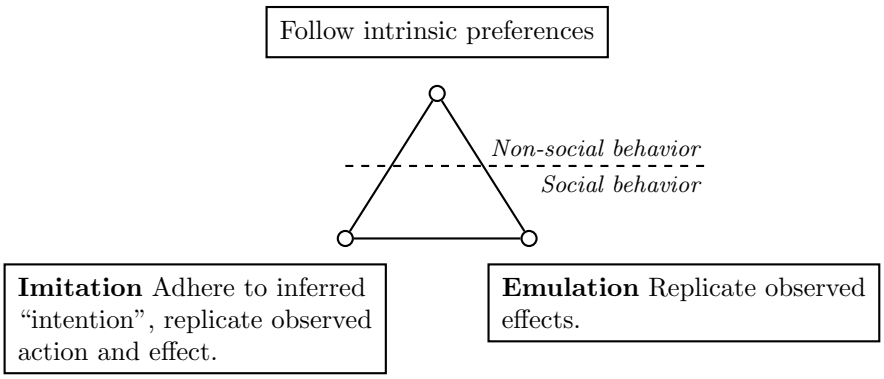


Figure 1.1: The *strategy triangle* from **?** illustrates the potential combination of several simple behaviors into an imitation mechanism.

The work we present is mainly related to the question of "What to imitate?", which, following **?**, includes the *motor 'gavagai'* problem that extends Quine's translation indeterminacy (see section 1.3). A way to clarify this point is to discriminate different levels of imitation. **?** introduced the distinction between *action-level imitation*, which consists in mimicking every action of the demonstrator (for example following the same arm trajectory), and *program-level imitation*, which focuses on reproducing the overall structure of the movement (for example following a different trajectory but featuring the same steps of reaching, grasping, lifting, etc.) Although quite intuitive, the notion of program-level imitation leaves open the question of the nature of the structure that the imitator should preserve, a question very similar to the ones introduced in the following section. **?** further introduce *functional imitation*, also called *effect-level imitation* by **?**, which consists in producing the same effect than the

demonstrator, and *abstract imitation*, which denotes imitation of the demonstrator's internal state (for example the demonstrator laughing could be imitated by smiling). **?** provide a model of some of these imitation mechanism, formulated in a Bayesian framework in which effect-level imitation is named *emulation*.

It follows from that discussion that the question "What to imitate?" closely relates to the question of how to decompose an observation into parts, including the choice of the level of granularity of the decomposition, together with the question of the nature of that decomposition. Furthermore the relevance of a level of decomposition, or a certain part in that decomposition, closely relates to focus, saliency, attention, and mutual attention that are central properties of imitation in humans and animals. While these two sets of questions might seem related but distinct at first sight, they actually are entangled: on one side having a good representation of motion is crucial to distinguish what is relevant within and across motions; on the other side mutual attention and other social signals are essential to achieving that decomposition and solving the indeterminacy that comes with it, as detailed in the followings sections. The work presented in this manuscript provides examples of what can be achieved by addressing both issues at the same time.

1.2 Structured representations for complex motions

The question of the complexity of movements and skills comes from the observation of human learners. It also seems to be both an empirical evidence in pedagogy and a natural way to proceed that learning occurs in a cumulative manner, starting simple and then growing in complexity. It seems quite easy to have an intuition about what a simple or a complex movement is: driving a plane looks *obviously* more complex than grasping simple toys. However, if one looks at what robots can actually do, it seems that currently robots are better at driving planes (or helicopters[2]) than at grasping objects. Actually this example is not really fair, but it points out that actually defining what a complex movement or skill is probably looks easier than it really is. Indeed, the complexity of a motion or an action is highly subjective to the agent's body, experience, culture, etc. Taking another example, why is a newborn gnu capable of walking within a few minutes[3] when human infants need months to achieve a similar behavior? Is this a proof that the gnu brain is more advanced regarding the learning of walking, or the expression of an evolutionary trade-off between the ability to quickly learn one task and the ability to learn a wider variety of tasks? From an epistemological point of view, research in artificial intelligence have long focused on solving the problems that looked difficult from a human point of view, such as playing chess, and considered as more anecdotal problems such as the ability to acquire mental representations of the world and grasp a chess tower or knife (see **?**). It seems now that the latter are much more difficult than the former. In particular we ground our study on the cognitive development starting from the sensori-motor level of perception: we thus take more interest in the shaping of that perception than in solving high level problems in a symbolic world.

A fundamental question behind these realities, both for biologists and roboticists, is to understand how life long learning is possible in a way that enables the efficient re-use

[2]More information and videos can be found at: `http://heli.stanford.edu`

[3]As demonstrated by the online video: `http://www.youtube.com/watch?v=zGaD2DH4evs`

of previously acquired knowledge and skills: answering this question provides at least one definition of complexity or simplicity of skills from a biological point of view. One approach, often labelled as cumulative learning, consists in gradually acquiring a lexicon of elements of increasing complexity, such that new elements can be obtained as the combination of simpler ones (see **?**, sec. 3–4). The concepts of motor synergies and motion primitives have been introduced as a potential implementation of this approach, by both motor control theorists and roboticists (see **?**). Motor primitives represent simple building blocks that can be composed to form complex motions. For example **?** interpret a group of experiments on the control system of frogs and rats as giving strong evidence that the brain features a set of primitive force fields that are combined linearly into more motor commands. **?** provide a more detailed review of that subject.

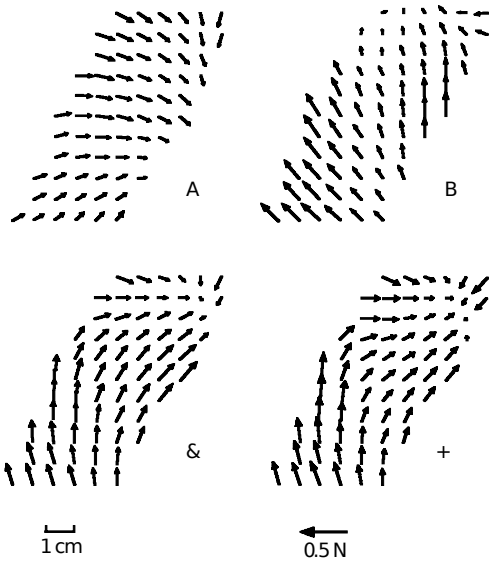


Figure 1.2: Illustration from **?**: A and B are two force fields measured on a frog while stimulating two distinct spinal sites. $\&$ is measured by separately stimulating both sites and $+$ is obtained by summing A and B: the result illustrates the combination in the frog's spine of two *primitive* force fields.

Although it seems natural to try to decompose complex motions into simpler elements, one could ask: "What is a complex movement or skill?" There are actually multiple answers to this question, each of which is of great interest regarding this work.

First, an easy answer to the question is: "A complex motion is a motion composed of simpler parts." This actually leaves us with new questions. The first one is naturally: "What is a simple motion or skill?" Actually, for roboticists, this question is deeply related to the way motions and actions are represented. Indeed simplicity, for a computer, often means efficiency of representation (or compression) whereas, for a human, a motion is often called natural or simple when it seems simple to produce. Unfortunately it seems that these definitions sometime refer to very different or even opposite notions of simplicity. The same difference arises with computers: it is easy for a computer to perform operations on very big numbers that a human could not perform. On the other hand a human is for example able to intuitively detect

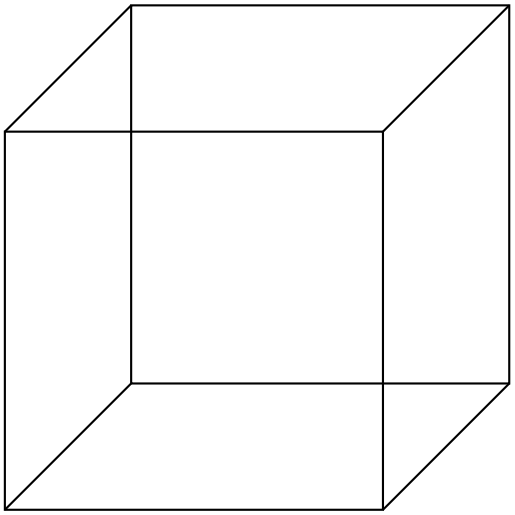


Figure 1.3: Necker's cube is a classical example of multistability in vision.

intersections between a large number of segments, an operation for which a computer requires complex algorithms. Actually in that last example, the problem described is not really the same for the human and the computer: while for the human the segments are represented visually, for the computer they are represented as a list of pairs of points. Therefore the human representation is much closer to both the nature of the geometrical property under consideration and to the processing capabilities of the human visual cortex. From using this analogy for robotics skills or behaviors, it follows that the first question on defining simplicity of motions can be studied through the research of motion representation that lead to efficient representation of human *natural motions*. **?** have shown that learning such representation from a set of observations of human motions also leads to good compression capabilities of natural human motion. Therefore, learning an appropriate representation have brought together both a computational notion of complexity, related to compression capabilities, to a more human centric one.

Looking back to the aforementioned *easy answer*, namely "A complex motion is a motion composed of simpler parts.", raises an other question, that is given by looking at the dual problem: instead of trying to define what a simple and a complex motion are, we could try to define what relates the simple to the complex motion, that is to say how primitive motions can be composed or combined into more complex ones. The study of the different ways of decomposing motions is central to this work and studied in more details in chapter 3. In many cases there is not uniqueness of decomposition; we refer to this issue as *the indeterminacy of the decomposition*. The indeterminacy of decomposition is analogous to phenomenon such as *multistability* in perception (see **???**, and fig. 1.3). Another central question is, knowing what notion of composition is involved, find means of removing the indeterminacy, that is to say find a relevant decomposition into simpler elements.

Although the ambiguity of decomposition is faced by learning systems throughout their development it is not necessarily a prerequisite of that development. Furthermore, we have seen that complex components of perception and action could be defined as

composed of simpler parts, but this notion of complexity may not fit the order in which infants acquire knowledge. In order to clarify this point in the following, we chose a different terminology: we refer to components that are combined into *complex* perceptions or actions as *primitive* or *local* components. **?** contrast *compositional* understanding, that describes an agent that is aware of the local components and their combination into a global perception or action, and *teleological* understanding, that accounts for an agent that only features global perception. More precisely the term teleological refers to a pragmatic emphasis on using the global knowledge even without refined understanding of its structure.

According to **?** the developmental path of infants goes first through teleological understanding before reaching compositional understanding. This developmental path is to contrast to the one stating that compositional understanding occurs first before any usage of the knowledge. Actually if simple and complex are defined with respect to the developmental path, each vision leads to an opposite definition of these notions: according to **?** the global stimulus is simpler, since used before by infants, than the local or primitive stimuli, whose awareness comes later. Conversely stating that compositional understanding comes first means that the primitive stimuli are simple and their composition complex. Because of that antagonism it is important to notice that 'primitive' may not mean simple and to differentiate 'complex', meaning 'composed', and 'complex', meaning harder to learn.

This thesis studies directly relations between local parts and global perceptions. With respect to these questions, we embrace the point of view of **?**, positing that teleological understanding may comes first. Therefore we generally do not assume the understanding of the compositional structure of perception to achieve global understanding. Furthermore we demonstrate that the perception of components can result from an auto-organisation of global perception: we provide models of the emergence of local components of perceptions (primitive motions, words) from the global perception, in contrast to achieving decomposition of perception as a pre-requisite to learning.

1.3 Language acquisition and multimodal learning

We already mentioned the question of the emergence of phonemes and words as examples of primitive elements occurring in the speech signal. Chapter 5 presents in more details technical approaches to the question.

A difficult aspect of the discovery of phonemes and words is the issue of segmentation. In the fields of speech recognition and acquisition, segmentation refers to the task of finding word boundaries from an acoustic stream containing spoken language. This is a difficult problem: unlike written language, spoken signal does not feature easy to detect word boundary cues similar to space characters or silences at the end of sentences (**?**). An illustrative example of the difficulty and ambiguity of segmenting written language without spaces is given by **?**: the sentence "theredonateakettleoftenchips" could be segmented into "the red on a tea kettle often chips" or "there, don ate a kettle of ten chips". Importantly the previous example highlights the ambiguous nature of the segmentation problem.

It has indeed been largely discussed whether the segmentation capability is a prereq-

uisite or a consequence of word recognition, and whether it should play a central role in the word recognition process. Actually experiments on infants performed by **??** have shown that young infant were capable of discovering words from an unknown language after a very short period of exposition (three minutes in their experiment) to acoustic signal only. More precisely their experiment demonstrates that children react differently when hearing sentences containing words they have been exposed to; interestingly this behavior emerges only from statistics on the acoustic signal. Following this experiment a large number of computational models of word discovery have been developed that implement a word segmentation process. A review of early work in that direction is given by **?**. An interesting experiment from **?** proposes a computational approach for an unsupervised setup very close to the one of **?**. On the other hand **?** have also demonstrated that word recognition can be achieved by an artificial learner without an explicit implementation of a segmentation process, but instead some form of symbolic supervision. The work presented in chapter 6 is inspired from the approach of Bosch and colleagues; however, we relax the symbolic supervision and instead study the use of multimodality to address the ambiguity issue.

Similar studies have also been conducted about the important question of grammar acquisition. **?** and later **?** have shown in experiments very similar to the previous one of **?**, that children around twelve months that are exposed for a short time to continuous speech generated from a grammar involving unknown words, react differently at the end of the initial exposure to utterances that are grammatically correct or not.

The notion of multimodal learning, which is a major topic of this thesis, refers to the ability of learning from a stream of mixed data coming from various sensory inputs, with different nature (for example sound, vision, taste). The close relation between language learning and multimodal learning is a central question of the work presented in this thesis. One immediate reason of that relation is that language acquisition is a multimodal problem because the language signal is multimodal. An evidence of that aspect was given by **?** and is referred to as the *McGurk effect*: when someone observes lips pronouncing 'ga' while he hears 'ba', he most of the time reports to have perceived the sound 'da' (see also **?**). Another major reason for claiming language and multimodal learning are closely related is the following: considering learning is taking place on top of multimodal perception, that not only include language signal but also other contextual information, provides a plausible solution to the ambiguity issues that occurs from language learning. One important source of ambiguity in language acquisition is related to the process of associating words to meanings, a process that is also known as *symbol grounding* and was introduced by **?** (see also **?**). Many unsupervised approaches introduced above only model the discovery of *acoustic words*, without relating these words to any kind of meaning, and relying exclusively on acoustic properties of the signal for their discovery. On the other hand the work presented by **?** focuses on the discovery of relations between the acoustic signal and a symbolic contextual information. The discovery of word is then shown to be a side effect of the learning of these correlations. In chapter 6 this idea is taken further and shown to extend to a multimodal signal that do not contain any predefined symbol. Similar ideas, but with some kind of symbolic information, where also featured in several previous works (**???????**).

Word grounding is however not an easy problem but rather an ambiguous one. An

important aspect of such ambiguity is the *indeterminacy of reference* introduced by **?**, which is often illustrated by the *'gavagai' thought experiment*. Quine presents the situation of a linguist who studies an unknown language and observes a speaker pointing toward a rabbit while saying 'gavagai'. In that situation the linguist cannot discriminate between several possible meanings of the word 'gavagai': it could actually mean 'rabbit' as well as 'undetached rabbit parts', 'animal', 'food', or even 'an object out of my reach'. Interestingly very similar issues occur in other modalities; one of them is referred as *motor 'gavagai'* problem in the case of imitation learning (see section 1.1 as well as **?**).

Finally, language is strongly related to action. The actual production of spoken language through the articulatory system is an immediate example of this relation. Indeed a spoken utterance may be seen either on the side of the produced acoustic stream, or on the side of the muscle commands that yielded that acoustic stream. The role of each one of these aspects of sound in its perception is still an actively discussed subject (for more details please refer to **?**). Either way this duality constitutes another important aspect of language multimodality.

Another important facet of the relations between language and action comes with the notion of grammar of action. While similarities between linguistic grammatical constructs and the structure of many common actions is quite straightforward (in the sense that it is perceived without effort), it is less clear to know if this similarity is coincidental or whether we perceive action this way because of our exposure to language, or if language has evolved on top of our perception of action grammar. The latter opinion is actually the subject of a whole theory about language origin called the *syntactic motor system* introduced by **?**.

The talking heads experiment (**??**) has been introduced by Steels and colleagues to study the emergence of language inside a community of robots as a model of the emergence of language in human communities. Each robot possess its own set of visual concepts and its own lexicon of associations between these concepts and initially arbitrary words. The experiment is based on an interaction frame between two robots called *guessing game*. During the game the pair of robots is looking at a visual scene that consists in geometric shapes on a magnetic board. The game follows several stages: to start, the first robot, called 'speaker', chooses a part of the visual scene that is shared with the second robot; then the speaker choses a topic among the objects in that region[4] and vocalizes a verbal description of that object; after perceiving the object description, the hearer robot guesses the described object (communicating its guess with a mechanism equivalent to pointing) and the speaker provides positive or negative feedback on the interaction, depending on whether it was successful or not. The experiment demonstrates that the simple interactions, combined with rotations in the role the robots play in the game, as well as in the pairs of interacting robots, lead to the emergence of a shared language at the scale of the robot community. Steels and colleagues define the *semiotic square* (see fig. 1.4) that illustrates the indirect nature of the communication: both agents in the guessing game have to go through the lexicon and ontological levels to bind the vocalization to its meaning (the referent object). This aspect is closely related to the grounding problem mentioned previously; it emphasizes the distinction between the sign used to communicate, its meaning for one agent, and the grounding of that meaning into perception. Importantly neither the signs, nor the meaning, nor the perception is

[4]In practice, in many experiments, the topic is actually chosen randomly.

exactly shared between the two agents: each of them has its own instances of signs, meanings, and perception. Throughout the interaction, the coherence between the signs, meanings, and perceptions of the agents increases.

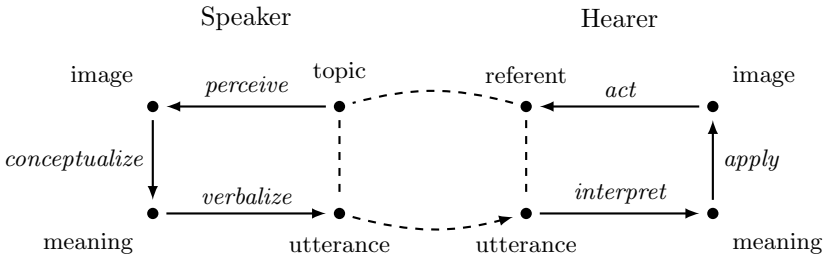


Figure 1.4: Illustration of the semiotic square as presented by **?**. The left part illustrates the generation by the speaker of an utterance describing the topic, and the left part to the interpretation of the utterance, as perceived by the hearer, into an hypothesis about the referent.

In this thesis we provide implementations of language learning experiments that do not assume direct access of the learner concepts as symbols, that is to say that do not shortcut the semiotic square (see chapter 6). The experimental frame we study for the language experiments is very close to the one of the talking heads. Therefore, although the questions studied in this thesis are mainly agent-centric, the implementations we described can be thought as replacement of the agent model in the talking heads experiment. However, a limitation has to be added to that claim, which is that we do not address the question of language production.

1.4 Important questions

In this chapter many subjects amongst those targeted by developmental robotics were introduced in which *primitive elements* play an important role for action, perception, and their acquisition and development. Such primitive elements might be phonemes, words, elementary motions, primitive objectives in tasks, objects or parts of objects in visual scenes, or multimodal elements combining several of these. In this thesis we explore this notion of *primitive elements* and various aspects of it. This study and its presentation is organized along specific axes that are listed below.

How can the intuition about *simple* and *complex* be made explicit and implemented on a robot or artificial system? We have seen that the notions of simple and complex, although being very intuitive, are not so easily formalized and that some perspectives can lead to definitions that are even opposite to their intuitive counterpart. We have discussed that one approach is to define the complex as the combination of several simpler parts. This shifts the question of defining simple and complex to the nature of the algebraic properties of the *simple or primitive elements*. We introduced several ways of combining primitive elements into complex ones. Given one of these, the question of the implementation remains: how can the low-level representation of motion or sound include sufficient algebraic properties so that real complex sounds are well represented as combinations of simpler ones,

under these properties? In other words, the question is to find representations of the perception and associated algorithms that are good substrates to the emergence of behaviors, regarding the simple to complex paradigm, that match the observation of children behaviors.

How can primitive elements emerge or be discovered through interaction of the agent with its physical and social environment? The importance of this question for both the understanding of children development and the conception of robots with similar capabilities is quite straightforward. However the diversity in nature in physical and social interactions that are essential to this emergence suggests a large number of candidate principles that could explain it. We study this question with a focus on the approach stating that the compositional understanding occurs after global representation of stimuli and actions.

What mechanisms can overcome the intrinsic ambiguity and indeterminacy that is characteristic of many approaches regarding the learning and emergence of these primitive elements? Many of the questions regarding the decomposition of perception into meaningful parts are actually very ambiguous. We have introduced several examples of such ambiguity as for example the decomposition of sentences into words or the relation between a word and its meaning. Communication with humans requires that these ambiguities are resolved; furthermore the mechanisms used by humans to resolve similar ambiguous issues seems to be central in their development. Therefore gaining insights on these mechanisms is expected to provide a better understanding of the associated cognitive processes.

Chapter 2

Technical background

As a preliminary to the presentation of experiments in direct relation to the subject of this thesis, we present separately in this chapter a set of technical details, in order to not burden the main presentation. Although the theories and algorithms presented in this chapter are important for a deep understanding and reproduction of the experiments, they are not essential for a high level understanding of these experiments.

2.1 Nonnegative matrix factorization

Experiments presented in chapters 3, 5 and 6 are based on the nonnegative matrix factorization algorithm. Although slightly different versions of this algorithm are used in various experiments, we give here a unified presentation of these algorithms.

2.1.1 Problem description

Non-negative Matrix Factorization (NMF, **??**) is a well-known machine learning algorithm. Given a data matrix $V \in \mathbb{R}^{F \times N}$ which columns are examples, with non-negative coefficients, it approximates it as the product:

$$V \simeq W \cdot H.$$

$W \in \mathbb{R}^{F \times K}$ and $H \in \mathbb{R}^{K \times N}$ also are non-negative. When the inner dimension K of the product is smaller than original dimension M and number of examples N, this achieves data compression by capturing structure in the matrix W.

Furthermore, with previous notations, the reconstructed matrix, $W \cdot H$, is of rank at most K, which makes this process a low rank approximation.

This is a form of dictionary learning in the case of non-negative data, where the columns of W are called *atoms*. The non-negativity constraint fits well in the case of frequency-like coefficients in the data matrix, which happens when histograms are used as data representation.

? presents a broad range of matrix factorization and clustering algorithms in a unified frame that consists in solving a optimization problem.

$$\underset{(W,H)\in C}{\arg\min}\ D(V, f(W \cdot H)) + R(W, H) \tag{2.1}$$

The following modeling components are identified in **?**:

- constraints on W and H represented as set C,
- measure of loss D, often a generalized Bregman divergence,
- transformation f,
- regularization R.

The basic NMF algorithms frame in eq. (2.1) with $R = 0$, f equals to identity and only constraining W and H to have non-negative coefficients, that is to say $C = \mathbb{R}_+^{M\times K} \times \mathbb{R}_+^{K\times N}$.

Problem ambiguity

It is important to notice at that point that the problem addressed by non-negative matrix factorization does not have an unique solution. For example, for any invertible matrix $M \in SL_k$, $W' = W \cdot M$ and $H' = M \cdot H$ yield the same product as W and H: $W \cdot H = W' \cdot H'$.

Some ambiguity in the solution of NMF problem exists for any factorization and is treated in this section. However even other sources of ambiguity can arise, depending on the generative model of the data. A geometrical model and an analysis of these questions are provided in Appendix A.2.

As pointed out in previous paragraph, any non-singular matrix M can transform the terms of the factorization without changing the result: $W \cdot H = (W \cdot M)(M^{-1} \cdot H)$. In addition if both M and M^{-1} are non-negative, then the new factors are still non-negative.

It is easy to show that the converse is true: if a matrix M is such that, for any W and H, $W' = W \cdot M$ and $H' = M \cdot H$ are non-negative and yield the same product as W and H: $W \cdot H = W' \cdot H'$, then M must be non-singular and both M and M^{-1} must be non-negative.

Furthermore, the set of invertible non-negative matrices with non-negative inverse is exactly the set of *monomial* matrices, that is to say matrices that are the product of a diagonal matrix with positive coefficients and a permutation matrix (**?**).

Thus the only transformations that leave unchanged any non-negative factorization are the compositions of a *scaling* of columns and a *permutation* of columns.

In order to compare two non-negative factorizations of the same non-negative matrix, it is useful to be able to normalize or compare such factorizations in a robust way regarding these transformations.

Dealing with scaling The easiest way to remove ambiguity due to scaling indeterminacy is to normalize the columns of W or H, for some norm or metric.

Dealing with permutations At least two approaches can be used to get rid of the permutation invariance of the factorization.

- A first approach is to use a total order on the columns to W or H to sort them increasingly. This yields a canonical form for the factorization. Any total order can be used. However this method might not be robust to noise (for example, using lexical order on coefficients of columns, a small perturbation on the first coefficient would change the ordering regardless the other coefficients).
- For the purpose of comparison between factorizations, it might be useful to define a distance between two such factorizations. For example from a measure of distance d between columns of W one could define the distance:

$$D(A, B) = \arg\min_{\sigma \in \mathfrak{S}_n} \sum_i d(A_i, B_{\sigma(i)})$$

  Computing such a distance would be done in $\mathcal{O}(K!)$ by a naive approach, however this problem is equivalent to finding a coupling of minimum weight in a bipartite graph, which can be solved by Kuhn-Munkres algorithm[1] (a.k.a. Hungarian method) in $\mathcal{O}(K^3)$.

Loss

The factorization of V as $W \cdot H$ is obtained through an optimization process, in which distance between data matrix V and reconstructed matrix $W \cdot H$ is minimized. We call *cost function* the function $\mathcal{L}(V, W \cdot H)$ that we want to minimize.

The first descriptions of NMF algorithms have been using both *Frobenius norm* and *generalized Kullback-Leibler divergence* on matrices.

Most common algorithms however extend well to the broader class of generalized β-divergence, which are separable (i.e. expressed as a sum over functions on coefficients) defined as follows:

$$D_\beta(V|Y) = \sum_{f=1}^{F} \sum_{n=1}^{N} d_\beta(V_{fn}|Y_{fn}) \tag{2.2}$$

$$\text{where } d_\beta(x, y) = \begin{cases} \frac{x}{y} - \log \frac{x}{y} - 1 & : \beta = 0 \\ x \log \frac{x}{y} - x + y & : \beta = 1 \\ \frac{1}{\beta(\beta-1)} \left(x^\beta + (\beta - 1) y^\beta - \beta x y^{\beta-1} \right) & : \beta \in \mathbb{R} \backslash \{0, 1\} \end{cases} \tag{2.3}$$

Cases where β equals 2, 1 and 0 corresponds to *Frobenius* norm, I divergence and *Itakura-Saito* divergence.

The Frobenius norm is one of the most common norms on matrices. The Frobenius norm of a matrix A, denoted by $\|A\|_F$ is defined as the square root of the sum of the square norms of columns of A.

[1] `http://en.wikipedia.org/wiki/Hungarian_algorithm`

$$\|A\|_F = \sum_{i=1}^{m}\sum_{j=1}^{n} {a_{i,j}}^2 \tag{2.4}$$

$$= \mathrm{Tr}(A^T A) \tag{2.5}$$

The Kullback-Leibler divergence is an information theoretic measure of the similarity between two probability distributions. It is denoted by $D_{KL}(p\|q)$ and defined, for to probability distributions p and q as:

$$D_{KL}(p\|q) = \sum_{x\in\mathcal{X}} p(x)\ln\left(\frac{p(x)}{q(x)}\right) \tag{2.6}$$

$$= H(p,q) - H(p) \tag{2.7}$$

where $H(p)$ and $H(p,q)$ are the Shannon entropy of p and the cross entropy of p and q.

For nonnegative matrices A and B a variant called *generalized Kullback-Leibler divergence* or *I-divergence* is often considered. It is defined as follows:

$$D_I(A\|B) = \sum_{i,j}\left(A_{i,j}\ln\left(\frac{A_{i,j}}{B_{i,j}}\right) - A_{i,j} + B_{i,j}\right). \tag{2.8}$$

2.1.2 Basic algorithms

Many algorithms have been designed to solve the NMF problem. While gradient descent can be used to find a local minimum of the joint problem, it is often more efficient (**?**) to use an EM-like approach based on alternating steps solving the problem in W and H. For the latter approach, the most common approached are: *alternate gradient descents* and *multiplicative algorithms*.

Both families share a common optimization approach: *alternate optimization* on matrices W and H. The following method was initially developed to take advantage of the convex nature of $\mathcal{L}(V, W\cdot H)$ both in W and H in cases such as Frobenius norm or generalized Kullback-Leibler divergence (for β-divergences this happens for $\beta\in[1,2]$) to perform *alternate optimization* on W and H. However, since even these cost function are not jointly convex in W and H, these methods can only converge towards local minimums.

Alternate projected gradient descent for β-divergence

The alternate projected gradient descent algorithms are based on alternating updates of the form presented in eqs. (2.9) and (2.10), where η is an update parameter. P is the projection on the first orthant, which means it replaces all negative entries in the matrices by 0.

$$H \longleftarrow P\big(H - \eta \frac{\partial D_\beta(V|WH)}{\partial H}\big) \tag{2.9}$$

$$W \longleftarrow P\big(W - \eta \frac{\partial D_\beta(V|WH)}{\partial W}\big) \tag{2.10}$$

These gradients are easily shown to be given by the following formula where the same notation is used for the real function $\frac{\partial d_\beta(x,y)}{\partial y}$ and its point-wise extension to matrices.

$$\frac{\partial D_\beta(V|WH)}{\partial H} = W^T \cdot \frac{\partial d_\beta}{\partial y}(V, WH) \tag{2.11}$$

$$\frac{\partial D_\beta(V|WH)}{\partial W} = \frac{\partial d_\beta}{\partial y}(V, WH) \cdot H^T \tag{2.12}$$

Actually these formulas are not restrictive to the case of β-divergences but to all *separable* cost function. In the particular case of the family of β-divergence, the element-wise derivatives to use are given in eq. (2.13).

$$\frac{\partial d_\beta}{\partial y}(x,y) = \begin{cases} \frac{1}{y} - \frac{x}{y^2} & : \beta = 0 \\ 1 - \frac{x}{y} & : \beta = 1 \\ y^{\beta-1} - xy^{\beta-2} & : \beta \in \mathbb{R}\backslash\{0,1\} \end{cases} \tag{2.13}$$

In order to get an efficient gradient descent algorithm it is necessary to use a step size adaptation method.

Multiplicative updates

This section is mainly based on presentation by **?**, please refer to this article for a more in-depth description of those algorithms.

Auxiliary function In this section we consider the case of separable cost functions in the particular example of β-divergences, for which the optimization with respect to H can be made separately on each columns of H. The same result is true for columns of W. Furthermore since $D(X|Y) = D(X^T|Y^T)$ each result on H for the optimization problem can be literally *transposed* to W by replacing V by V^T and thus H by W^T and W by H^T.

Targeting alternate optimization process we consider the problem, $\underset{h\in\mathbb{R}^K,\ h\geq 0}{\arg\min}\ C(h)$ with $C(h) = D_\beta(x|Wh)$. To solve it we introduce auxiliary functions as follows.

Definition (Auxiliary function)**.** $G:\ \mathbb{R}_+^K \times \mathbb{R}_+^K \longrightarrow \mathbb{R}_+$ *is said to be an auxiliary function of cost* C *if and only if*

(i) $\forall h \in \mathbb{R}_+^K,\ C(h) = G(h,h)$

(ii) $\forall (h, \tilde{h}) \in (\mathbb{R}_+^k)^2,\ C(h) \leq G(h, \tilde{h})$

The idea behind auxiliary functions is that if a sequence of values (h_t) is such that $G(h_{t+1}, h_t) \leq G(h_t, h_t)$ then

$$C(h_{t+1}) \leq G(h_{t+1}, h_t) \leq G(h_t, h_t) = C(h_t)$$

and thus the sequence $(C(h_t))$ is non-increasing.

To construct such an auxiliary function we use a *convex-concave-constant* decomposition of d_β, that is to say functions $\breve{d}$, $\hat{d}$ and $\bar{d}$ which are respectively convex, concave and constant with respect to their second variable, all differentiable with respect to the second variable and such that

$$d_\beta(x, y) = \breve{d}(x, y) + \hat{d}(x, y) + \bar{d}(x).$$

In the following we also assume that these functions are differentiable with respect to y. Such a decomposition is given in Table 2.1

Theorem 1. *Let $\tilde{h}, h \in \mathbb{R}_+^K$, and $\tilde{v} = Wh$, the function defined by eq.* (2.14) *is an auxiliary function for the β-divergence loss,*

$$G(h|\tilde{h}) = \sum_f \left[\sum_k \frac{w_{fk}\tilde{h}_k}{\tilde{v}_f} \breve{d}\Big(v_f|\tilde{v}_f \frac{h_k}{\tilde{h}_k}\Big) + \hat{d}'(v_f|\tilde{v}_f) \sum_k w_{fk}(h_k - \tilde{h}_k) + \hat{d}(v_f|\tilde{v}_f) + \bar{d}(v_f) \right] \tag{2.14}$$

	$\breve{d}(x\|y)$	$\breve{d}'(x\|y)$	$\hat{d}(x\|y)$	$\hat{d}'(x\|y)$	$\bar{d}(x\|y)$
$\beta < 1,\ \beta \neq 0$	$\frac{-1}{\beta-1}xy^{\beta-1}$	$-xy^{\beta-2}$	$\frac{1}{\beta}y^\beta$	$y^{\beta-1}$	$\frac{1}{\beta(\beta-1)}x^\beta$
$\beta = 0$	xy^{-1}	$-xy^{-2}$	$\log(y)$	y^{-1}	$x(\log(x) - 1)$
$1 \leq \beta \leq 2$	$d_\beta(x\|y)$	$d'_\beta(x\|y)$	0	0	0
$\beta \geq 2$	$\frac{1}{\beta}y^\beta$	$y^{\beta-1}$	$\frac{-1}{\beta-1}xy^{\beta-1}$	$-xy^{\beta-2}$	$\frac{1}{\beta(\beta-1)}x^\beta$

Table 2.1: Auxiliary function for the β-divergence

Maximization-Minimization In this paragraph an update rule for h will be chosen such that the next value h^{MM} solves:

$$h^{MM} = \arg\min_h G(h, \tilde{h}).$$

This is obtained from current value $\tilde{h}$ by equation (2.15). The values $\gamma(\beta)$ are given in Table 2.2.

$$h_k^{MM} = \tilde{h}_k \left(\frac{\sum_f w_{fk} v_f \tilde{v}_f^{\beta-2}}{\sum_f w_{fk} \tilde{v}_f^{\beta-1}} \right)^{\gamma(\beta)} \tag{2.15}$$

	$\beta < 1$	$1 \leq \beta \leq 2$	$\beta > 2$
$\gamma(\beta)$	$\frac{1}{2-\beta}$	1	$\frac{1}{\beta-1}$

Table 2.2: Values $\gamma(\beta)$

The update rule can actually be re-written in terms of matrix operations, in which it takes a relatively simple form. We denotes the Hadamard (i.e. entry-wise) product by $\otimes$ and the entry-wise division with fractions. In this section matrix power is meant entry wise.

$$H \leftarrow H \otimes \frac{W^T\left[V \otimes (WH)^{\beta-2}\right]}{W^T(WH)^{\beta-1}} \qquad W \leftarrow W \otimes \frac{\left[V \otimes (WH)^{\beta-2}\right]H^T}{(WH)^{\beta-1}H^T} \tag{2.16}$$

Heuristics algorithm Another kind of update, that guarantees $G(h_{t+1}, h_t) \leq G(h_t, h_t)$ at least for β in $[0, 1]$ is given by equation (2.17).

$$h_k^{MM} = \tilde{h}_k \frac{\sum\limits_f w_{fk} v_f \tilde{v}_f^{\beta-2}}{\sum\limits_f w_{fk} \tilde{v}_f^{\beta-1}} \tag{2.17}$$

Frobenius norm and I-divergence In Frobenius and generalized Kullback-Leibler cases, $\gamma(\beta) = 1$ thus there is no difference between the two variants of the algorithm. More precisely the associated multiplicative updates were introduced by **?**. $\mathbf{1}$ denotes a matrix which coefficients are all ones.

$$H \leftarrow H \otimes \frac{W^T V}{W^T WH} \qquad W \leftarrow W \otimes \frac{VH^T}{WHH^T} \qquad \text{Frobenius update} \tag{2.18}$$

$$H \leftarrow H \otimes \frac{W^T \dfrac{V}{WH}}{W^T \cdot \mathbf{1}} \qquad W \leftarrow W \otimes \frac{\dfrac{V}{WH} H^T}{\mathbf{1} \cdot H^T} \qquad \text{I-divergence update} \tag{2.19}$$

2.1.3 Variants and improvements

Sparse NMF

Sparseness of the components or the coefficients can be enforced for NMF. However several approaches may be used to achieve this goal.

Projected gradient **?** proposes a measure of sparseness of a nonnegative vector as follows:

$$\text{sparseness}(x) = \frac{\sqrt{n}\|x\|_2 - \|x\|_1}{\|x\|_2(\sqrt{n} - 1)} \tag{2.20}$$

It is defined for any x which is nonzero and takes values in $[0, 1]$.

Hoyer then proposes to solve the NMF problem with the additional constraint that any column of W and row of H has a fixed sparseness. Hoyer also provides an algorithm to project to the set satisfying the constraint.

Regularization **?** present propose algorithms to solve the NMF problem extended with a sparsity inducing regularization of the dictionary or coefficients based on the l_1 norm.

Semi-NMF and convex-NMF

See the paper of **?** for semi and convex algorithms and clustering interpretation; see work from **?** for adaptation of β-NMF algorithm.

2.2 Inverse reinforcement learning

This section provides a general presentation of the fields of reinforcement learning and inverse reinforcement learning whose concepts are supporting the experiments from chapter 4. Further details are also provided on specific inverse reinforcement learning algorithms on which the algorithm presented in section 4.3 is based.

2.2.1 Background: reinforcement learning

This paragraph provides a quick introduction of the concepts grounding reinforcement learning. A more extensive introduction to reinforcement learning is given by **?**.

> " Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making. It is distinguished from other computational approaches by its emphasis on learning by the individual from direct interaction with its environment, without relying on exemplary supervision or complete models of the environment. "
>
> Richard Sutton and Andrew Barto, *Reinforcement learning* (**?**)

One important motivation behind reinforcement learning is to provide a model for trial and error learning, as originally described by **?**. The technical background for such models was developed on top of previous work on *dynamic programming* (**?**) and *Markov decision process* (**??**).

The agent and the world

The reinforcement learning model introduces an important, and sometime counter-intuitive, separation between an *agent* and its *environment*. It also assumes that the agent is involved in a discrete sequence of interactions with the environment. More precisely, at discrete time steps t, the agent measures the state $x_t \in \mathcal{X}$ of the environment and performs an action $a_t \in \mathcal{A}$. The action triggers a change in the

state of the environment and the emission of a reward r_t from the environment to the agent (see fig. 2.1). The distinction between the agent and the environment can be misleading for it does not in general correspond to the physical separation between the body of the agent and its living environment. Instead anything that is not in direct control of the agent's decisions, such as the reward, is considered part of the environment. This differs from a more intuitive model of an animal in which the reward may be a hormonal response to having eaten, thus being emitted by the animal's body and depending on a state of its body. Similarly, an agent may be modifying its mental state which would thus be modelled as part of the environment.

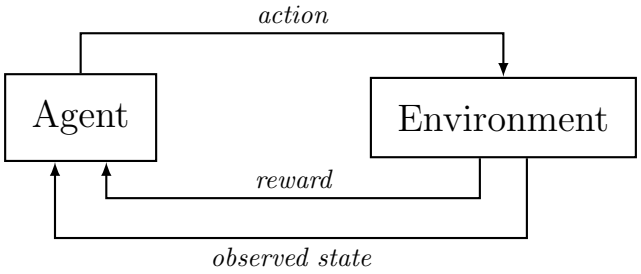


Figure 2.1: The agent and the environment (reproduced from **?**, p. 52).

The Markov property

Classical reinforcement learning assumes that the transitions between states of the environment are Markovian. That is to say the probability distribution over next state x_{t+1}, given all previous states and actions, is such that, for all $t \geq 0$:

$$P(x_{t+1}|x_0, \ldots, x_t, a_0, \ldots, a_t) = P(x_{t+1}|x_t, a_t).$$

That probability distribution is often denoted as the *transition probability*. Similarly, the reward received by the agent at time t is assumed to only depend of a_t and x_t.

The policy

The policy is the model of the agent behavior; it may be understood as an equivalent of the *stimulus-response rule*. The policy is a distribution over possible actions taken by the agent at time t, given the history of the environment state. The policy is called *deterministic* when the distribution degenerates to a function, that is to say when only one action has nonzero probability.

A policy is said to be *stationary* when it only depends on current state of the environment. Under the Markovian hypothesis, since the reward only depends on the current state and the action taken, stationary policies can achieve the same expected rewards as general policies. Therefore, we can safely restrict to the class of stationary policies.

Markovian decision process

A Markovian decision process (MDP) is defined as a state space $\mathcal{X}$, an action space $\mathcal{A}$, a transition probability P, a reward function r, and a *return*. The return defines a notion of accumulated reward. For example one could consider the *average return* over a period T, defined as:

$$R_{\text{average}} = \frac{1}{T}\sum_{t=0}^{T-1} r_t,$$

or the *discounted return* over an infinite horizon, with *discount factor* $\gamma < 1$, defined as:

$$R_{\text{discounted}} = \sum_{t\geq 0} \gamma^t r_t.$$

A MDP is the problem of determining policies that maximize the expected return. Both definition of return lead to similar results; in the following we focus on the discounted return. The MDP is then denoted by a quintuple $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$. P is a mapping from state actions pairs to probability distributions over next states. We denote by $P(x'|x, a)$ the probability to transition to state x' from state x, knowing that action a was taken. Finally, $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function.

Dynamic programming

For a given MDP and a given stationary policy π, that is a mapping from states to probability densities over actions, one can define the value function V^π and the action value function Q^π[2]:

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, A_t)\middle| X_0 = x\right] \tag{2.21}$$

$$Q^\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, A_t)\middle| X_0 = x, A_0 = a\right]. \tag{2.22}$$

The expectations are taken over all possible trajectories $(X_t, A_t)_{t\geq 0}$. Functions verifying the following equations are called optimal value function and optimal action value function:

$$V^*(x) = \sup_\pi V^\pi(x),\ \forall x \in \mathcal{X} \tag{2.23}$$

$$Q^*(x, a) = \sup_\pi Q^\pi(x, a),\ \forall (x, a) \in \mathcal{X} \times \mathcal{A}. \tag{2.24}$$

A policy that is optimal (w.r.t. to eq. (2.23) or eq. (2.24)) over all states is said to be optimal. Greedy policies over Q^*, that is to say policies such that $\pi(\arg\max_a Q^*(x, a)|x) = 1$, are known to be optimal. They are in particular deterministic policies, which state that the existence of that maximum implies the existence of optimal deterministic policies.

[2]For infinite state or action space some additional hypothesis is required for these functions to be finite, such as the boundedness of the reward function.

Equations (2.21) and (2.22) are equivalent to the following fixed point equations, also denoted as Bellman equations.

$$V^\pi(x) = \sum_{a\in\mathcal{A}} \pi(x,a) \left[r(x,a) + \gamma \sum_{y\in\mathcal{X}} P(y|x,a) V^\pi(y) \right] \tag{2.25}$$

$$Q^\pi(x,a) = r(x,a) + \gamma \sum_{y\in\mathcal{X}} P(y|x,a) \sum_{b\in\mathcal{A}} \pi(y,b) Q^\pi(x,b) \tag{2.26}$$

Similarly, eqs. (2.23) and (2.24) are equivalent to the following fixed point equations, denoted as Bellman optimality equations.

$$V^*(x) = \max_{a\in\mathcal{A}} \left[r(x,a) + \gamma \sum_{y\in\mathcal{X}} P(y|x,a) V^*(y) \right] \tag{2.27}$$

$$Q^*(x,a) = r(x,a) + \gamma \sum_{y\in\mathcal{X}} P(y|x,a) \max_{b\in\mathcal{A}} Q^*(x,b) \tag{2.28}$$

The right members of these equations defines the Bellman operators and Bellman optimality operators. The contracting property of these operators is sufficient to prove the existence and uniqueness of solution of these equations from the Banach fixed point theorem[3], in the case of discrete state and action spaces. Interestingly this property provides a Picard iteration algorithm to compute value and optimal value functions, named *value iteration*, which grounds *dynamic programming*.

2.2.2 What is inverse reinforcement learning?

In the formulation of reinforcement learning, the reward function is assumed to be fixed and unknown but observed by the agent. Therefore, in order to build an artificial agent, one must first fully specify that reward function, which turns out to be both a very sensitive and difficult task. That difficulty yields limitations to the accuracy of reinforcement learning models of living agents. Because the manual specification of the reward function results in this limitation, the agent model could instead be more accurately fitted to reality by learning the reward function, from observation of the behavior it produces. That statement lead **?** to formulate the problem of inverse reinforcement learning as follows.

> " It seems clear, however, that in examining animal and human behaviour we must consider the reward function as an unknown to be ascertained. [...] Therefore, to model natural learning using reinforcement learning ideas, we must first solve the following computational task, which we call inverse reinforcement learning:
>
> Given 1. measurements of an agent's behaviour over time, in a variety of circumstances, 2. measurements of the sensory inputs to that agent; 3. a model of the physical environment (including the agent's body). Determine the reward function that the agent is optimizing. "
>
> **?**

[3] `http://en.wikipedia.org/wiki/Banach_fixed_point_theorem`

Another application of inverse reinforcement learning is the field of robot programming by demonstration. Instead of having a robot that copies the actions of a demonstrator, the robot could infer the intention of the demonstrator and then attempt to solve the same task. Indeed, copying the same action in a different context is often not effective, whereas understanding the intention enables better generalization. In that perspective, a reward function models the intention of the agent and is supposed to shape its behaviors through reinforcement learning. In the following we adopt a formulation of the problem in terms of a demonstrator or expert, whom an apprentice learns to imitate.

Inverse reinforcement learning assumes the observation of actions from an agent. This agent is modelled as optimizing an unknown reward function and an apprentice tries to guess the reward function that the agent is optimizing. We make the simplifying assumption that the demonstrator and the apprentice share a common representation of the world, as a set $\mathcal{X}$ of states, a set $\mathcal{A}$ of actions, and transition probabilities P. We represent the demonstrator's intention in the form of a reward function r, which defines a Markov decision process (MDP), that is a quintuple $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ (see section 2.2.1). Finally, the objective of inverse reinforcement learning is to infer the reward function r, given a set of observations that are couples (x_t, a_t) of states $x_t \in \mathcal{X}$ and action $a_t \in \mathcal{A}$ from the demonstrator. The demonstrator or expert is for that assumed to act optimally or nearly optimally. Strictly speaking, the optimality of the demonstrator means:

$$a_t \in \arg\max_{a \in \mathcal{A}} Q_r^*(x_t, a).$$

It is important to notice here, that, in opposition to section 2.2.1, we have written the dependency of the MDP, and in particular the value functions, in the reward.

Limitations of the reward estimation problem

Unfortunately, the problem we have just formulated is ill-posed. First of all the null reward is always a solution since it leads to constant null functions and therefore

$$\arg\max_{a \in \mathcal{A}} Q_r^*(x, a) = \mathcal{A}$$

for all $x \in \mathcal{X}$. Similarly any constant reward function yield constant value functions, which makes any policy optimal.

Also, as explained by **?**, for given state and action spaces, any *potential-based function* (see the definition below) can be added to a reward function without changing the set of optimal policies. Their result also comes with a weaker converse: no other function can be added to any MDP without modifying the set of optimal policies. Finally, the sufficient condition also guarantees stability of the set of *near-optimal* policies.

Definition. *Let F be a real-valued function on $\mathcal{X} \times \mathcal{A} \times \mathcal{X}$. It is said to be* ***potential-baised*** *if there exists a real-valued function $\phi : \mathcal{X} \to \mathbb{R}$ such that for all $x \in \mathcal{X}$, $a \in \mathcal{A}$,*

$$F(x, a, x') = \gamma\phi(x') - \phi(x)$$

Theorem 2. *(?)*

- ***Sufficiency.*** *If F is a potential-based function than any optimal policy of $M = (\mathcal{X}, \mathcal{A}, P, \gamma, R)$ is an optimal policy of $M' = (\mathcal{X}, \mathcal{A}, P, \gamma, R + F)$ (and vice-versa).*
- ***Necessity.*** *If F is not a potential based function, then their exist (proper) transition functions P and a reward function $R : \mathcal{X} \times \mathcal{A} \longrightarrow \mathbb{R}$ such that no optimal policy in $M' = (\mathcal{X}, \mathcal{A}, P, \gamma, R + F)$ is optimal in $M = (\mathcal{X}, \mathcal{A}, P, \gamma, R)$.*[4]

The theorem from **?** considers a definition of reward functions slightly extended with respect to the one we use here: the reward is a function of next state in addition to current state and current action. The notion of potential-based function as defined above does not make sense without that formulation. However, the same sufficiency result holds if we define *in average potential-based functions*, in terms of expected next state, given an action. That is to say a function F such that their exists a potential ϕ for which:

$$F(x, a) = \gamma \mathop{\mathbb{E}}_{x' \sim P(\cdot|x,a)} \big[\phi(x')\big] - \phi(x).$$

Such function is not always potential based in the sense of **?** but the necessity condition holds.[5][6] Following this comment, in the case of non-deterministic transition functions (for which some functions are not potential based but potential based in average), the class of rewards that share the same set of optimal policies can be expanded.

Finally other context dependant functions may, for a given family of transition functions, be added to the reward function without changing the set of optimal policies. Also the previous discussion only covers additions to the reward function but not others transformations. For example we mentioned that the scalar multiplication of the reward leaves invariant the set of optimal policies.

Formulations of the reward estimation problem

From the high level formulation of the inverse reinforcement learning problem by **?**, several more precise formulations have been made in order to bring the problem to the range of algorithmic formulations.

First of all, **?** introduce a characterization of rewards that make an action optimal. The following formulation uses a matrix representation of the transition function: for each action a, $\mathbf{P}_a$ contains the transition probabilities between states. Also r is assumed to only depend on the state and be bounded by $r_{\max}$; it is represented as a vector of one value for each state. Finally, for simplicity, the actions are supposed to be re-ordered for each state in such a way that one action a^* is optimal for all states. The symbol $\succeq$ denotes that each coefficient is greater.

[4] The theorem as formulated above is not true since a constant value can be added to the reward function without changing the optimality of policies. **?** actually mention this fact in the demonstration of the necessity condition because they need to *shift to zero* the F function. However, for this transformation to be made without loss of generality the necessity condition needs to be slightly relaxed.

[5] Actually the proof does not change.

[6] Although this may seem in contradiction with the necessity condition from the theorem, it is not. Indeed, the in average potential based functions depend on the transition function. Therefore they do not leave invariant the set of optimal policies for any MDP.

Theorem 3. *(?) Let a finite state space $\mathcal{X}$, a finite set of actions $\mathcal{A}$, transition probability matrices $\left(P_a\right)_{a\in\mathcal{A}}$, and a discount factor $\gamma \in]0,1[$ be given. Then the policy π given by $\pi(x) = a^*$ is optimal if and only if, for all $a \neq a^*$, the reward r satisfies:*

$$\left(\mathbf{P}_{a^*} - \mathbf{P}_a\right)\left(\mathbf{I} - \gamma \mathbf{P}_{a^*}\right)^{-1} r \succeq 0$$

? use the previous theorem to derive an optimization problem. For that they use the heuristic that the solution reward function should, in additional to make the observed policy optimal, maximize the loss in return induced by diverging from that policy. This is obtained by maximizing the quantity:

$$\sum_s \min_{a\neq a^*} \left(\mathbf{P}_{a^*}(s) - \mathbf{P}_a(s)\right)\left(\mathbf{I} - \gamma \mathbf{P}_{a^*}\right)^{-1} r$$

Finally, they also penalize the norm $\|r\|_1$ of the solution reward and bound its coefficients. The inverse reinforcement learning problem then takes the form of a linear programming problem.

$$\begin{aligned} \text{maximize } & \sum_s \min_{a\neq a^*} \left(\mathbf{P}_{a^*}(s) - \mathbf{P}_a(s)\right)\left(\mathbf{I} - \gamma \mathbf{P}_{a^*}\right)^{-1} r - \lambda\|r\|_1 \\ \text{such that } & \left(\mathbf{P}_{a^*} - \mathbf{P}_a\right)\left(\mathbf{I} - \gamma \mathbf{P}_{a^*}\right)^{-1} r \succeq 0,\ \forall a \neq a^* \\ & \|r\|_\infty \leq r_{\max} \end{aligned}$$

Matching expected features Another interesting formulation of the inverse reinforcement learning problem is expressed in terms of *expected feature vector*. In the case where the reward function is assumed to be linearly parametrized on a feature vector ϕ, that is $r(x) = \theta^T \phi(x)$ for any state s, we define the feature expectation under policy π as

$$f(\pi) = \mathop{\mathbb{E}}_{(x_t)\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \phi(x_t)\right].$$

It follows immediately that the expected return for policy π is given as the scalar product $R(\pi) = \theta^T f(\pi)$. The feature expectation under expert policy, $f(\pi_E)$ can be empirically estimated from the demonstrations as $\hat{f}_E$. **?** state that finding a policy $\tilde{\pi}$ that matches the expert's feature expectation to a certain error is sufficient to achieve a return that is close, at most to the same error, to the expert's empirical return under the real reward.

Proposition 1. *(?) Let $\tilde{\pi}$ be a policy and $\varepsilon > 0$.*

If $\|f(\tilde{\pi}) - \hat{f}_E\|_2 \leq \varepsilon$ then for any parameter $\|\theta\|_2 \leq 1$ and reward $\theta^T\phi$:

$$\|R(\tilde{\pi}) - R(\hat{\pi}_E)\|_2 \leq \varepsilon$$

The proposition is actually a trivial consequence from the Cauchy-Schwartz inequality, which is used as a starting point by **?** to express the problem of matching the expert's expected features. This leads to a new formulation of the inverse reinforcement learning problem back into an apprenticeship learning problem: the goal is not any more to recover the real reward function but to find a policy that behaves as well as the expert's under the real reward.

Matching the policy Finally, another approach is to directly target the imitation of the expert's policy, but parametrized in the reward. That is to say find a reward function r such as the optimal or nearly optimal policy π_r associated to that reward function is as close as possible to the expert's, that is to say solve the following problem for a given norm $\|\cdot\|$.

$$\arg\min_r \|\pi_r - \pi_E\|$$

This approach is followed by **?** and their algorithm to solve it is detailed in section 2.2.3.

2.2.3 Algorithms for inverse reinforcement learning

Various algorithms have been developed to solve the various formulations of the inverse reinforcement learning problem, as introduced in previous section.

Some approaches such as those of **?** and **?** directly attack the ill-posed problem of inferring the *real* reward function. For that they need to introduce additional constraints on the problem. **?** then use linear programming while **?** introduce a Markov chain Monte-Carlo algorithm.

Other approaches focus on finding policies that achieve as well as the expert's on the real, unknown, task. For example, **?** introduce a quadratic programming and a projection algorithm to solve that task.

Finally, several algorithms have been developed to match the expert's policy with a learnt policy that is parametrized in the reward function. Examples include the approach from **?** described below, as well as the one from **?**.

Gradient inverse reinforcement learning

In this section we present the algorithm and results introduced by **?**. For more details and proofs the reader is invited to refer to the original paper.

Inverse reinforcement learning for apprenticeship learning is based in the assumption that the expert's intention is to solve a task, modeled by a reward function. Mimicking the expert behavior therefore consists in using a policy that is optimal for the same reward. The hybrid approach from **?** focuses on learning a reward such that an associated optimal policy matches the expert's actions.

In the following we assume that the model of the world, that is composed of states, actions, and transitions, is fixed and that the discount factor γ is known. However we represent the intention of a demonstrator (called *expert*) as acting optimally with respect to a task that is modelled as a reward function. Since the model of the world is fixed no distinction is made between a reward function r and the associated MDP, $MDP(r)$.

We assume that we observe demonstrations ξ from an expert, that are sequences (x_t, a_t) of states $x_t \in \mathcal{X}$ and action $a_t \in \mathcal{A}$, such that $\xi = (x_t, a_t)_{t\in[|1,T|]}$.

In this single task setup, the expert provides one or several demonstrations of solving the same task. The expert actions are modeled by a stationary policy π_E. The

objective of apprenticeship learning becomes to find a policy π that minimizes the cost function from eq. (2.29), in which μ denotes the average state occupation, that is to say $\mu_E(x) = \lim_{T\to\infty} \mathbb{E}_\pi \frac{1}{T} \sum_{t=1}^{T} \delta_{X_t=x}$.

$$J(\pi) = \sum_{x\in\mathcal{X}, a\in\mathcal{A}} \mu_E(x)\big[\pi(x|a) - \pi_E(x|a)\big]^2 \tag{2.29}$$

For an expert demonstration represented by $\xi = (x_t, a_t)_{t\in[|1,T|]}$ one estimates J by eq. (2.30), in which $\hat{\mu}_{E,\xi}$ and $\hat{\pi}_{E,\xi}$ are empirical estimates of μ_E and π_E from ξ.

$$J_\xi(\pi) = \sum_{x\in\mathcal{X}, a\in\mathcal{A}} \hat{\mu}_{E,\xi}(x)\big[\pi(x|a) - \hat{\pi}_{E,\xi}(x|a)\big]^2 \tag{2.30}$$

In the following the reward r_θ is parametrized by $\theta \in \Theta$ where $\Theta \in \mathbb{R}^d$ and we denote by Q^*_θ the optimal action value function of the MDP associated to r_θ. In practice linear features are used for r. Let G be a smooth mapping from action value functions to policies that returns a close to greedy policy to its argument. Instead of minimizing J_ξ over any subset of policies, **?** suggest to constrain π to be of the form $\pi_\theta = G(Q^*_\theta)$.

Solving the apprenticeship learning problem is then equivalent to finding θ that reaches:

$$\min_{\theta\in\Theta} J_\xi(\pi_\theta) \text{ s.t. } \pi_\theta = G(Q^*_\theta). \tag{2.31}$$

In practice, **?** use Boltzmann policies as choice for the G function, as given by eq. (2.32) where the parameter β is a nonnegative real number. This choice ensures that G is infinitely differentiable. Assuming that Q^*_θ is differentiable w.r.t. θ its first derivate is given by eq. (2.33).

$$G(Q)(a|x) = \frac{\exp\big[\beta Q(x,a)\big]}{\sum_{a'\in\mathcal{A}} \exp\big[\beta Q(x,a')\big]} \tag{2.32}$$

$$\frac{\partial G(Q^*_\theta)(a|x)}{\partial\theta_k} = G(Q^*_\theta)(a|x)\left(\frac{\partial Q^*_\theta(x,a)}{\partial\theta_k} - \sum_{a'\in\mathcal{A}} \pi_\theta(a'|x)\frac{\partial Q^*_\theta(x,a')}{\partial\theta_k}\right) \tag{2.33}$$

The following proposition from **?** provides both guarantees that $\dfrac{\partial Q^*_\theta(x,a)}{\partial\theta_k}$ is meaningful and a practical way to compute it.

Proposition (Neu and Szepesvari)**.** *Assuming that r_θ is differentiable w.r.t. θ and $\sup_{(\theta,x,a)\in\Theta\times\mathcal{X}\times\mathcal{A}} < \infty$, the following statements hold:*

1. *Q^*_θ is uniformly Lipschitz-continuous as a function of θ in the sense that there exist $L' > 0$ such that for any $(\theta,\theta') \in \Theta^2$, $|Q^*_\theta(x,a) - Q^*_{\theta'}(x,a)| \leq L'\|\theta - \theta'\|$*
2. *The gradient $\nabla_\theta Q^*_\theta$ is defined almost everywhere[7] and is a solution of the following fixed point equation, in which π is a greedy policy on Q^*_θ:*

$$\varphi_\theta(x,a) = \nabla_\theta r(x,a) + \gamma \sum_{y\in\mathcal{X}} P(y|x,a) \sum_{b\in\mathcal{A}} \pi(b|y)\varphi_\theta(y,b) \tag{2.34}$$

[7] that is except on a set of measure zero

The previous result thus provides an algorithm, similar to dynamic programming that yields the derivative of Q^*_θ with respect to θ. It it then easy to combine it with the differentiates of J_ξ with respect to π and G to obtain a gradient descent algorithm that finds a local minima of the objective.

? furthermore provide a variant of this algorithm following a natural gradient approach.

Chapter 3

Learning a dictionary of primitive motions

The idea of motion primitive, already introduced above in section 1.2, is both rooted in biology (see **???**) and perceived as an appealing and efficient paradigm for robot programming (see **??**): being able to encapsulate basic motion representation and motor skills in a way that enables their combination is expected to bring combinatorial growth of robot skills in place of the usually observed linear growth. Indeed if a robot is capable of combine a new skill with all the other skills it already knows, it actually has not learnt one skill but as many as the potential combinations it can achieve. That way it scaffolds its knowledge in a way that models the simple to complex learning trajectories observed in humans. The same principle applies to the recognition of human motions (**?**).

However, we have seen in section 1.2 that the concept of motion primitive from previous paragraph is vague and needs to be further defined. For instance, the definition of motion primitive could be closely related to the nature of their potential combinations into more complex motion or skills. In the next section, a closer look is given to that concept of combination of primitive motions, with an emphasis on two questions: "What does it mean to combine motion parts?" and "Where do the primitives come from?"

The question of the nature of the composition is itself connected with issues of motion or skill representation that make them compatible with such combinations, but also with the algorithms that might be necessary to achieve the composition. As for the second question, the origin of motion primitives can vary greatly in the literature: some techniques involve pre-coded skills or motions, others, learning on subproblems provided by a caregiver, learning from self-exploration, learning to decompose a demonstrated motion or skill, etc. Finally the hierarchical nature of a lexicon of primitives in which high order primitives are defined in terms not of low levels primitives but intermediate ones, that in turn are defined on lower level primitives, brings its own set of challenges. However we do not focussed on that aspect in that work.

An approach to the learning of motion parts is to implement an algorithm that tries to closely match a decomposition of motions to the high level representation we,

as grown-up humans, have of that motion. However that high level representation is highly dependant of our perceptual and motor apparatuses, our development history, our culture, which makes that task highly ambiguous. Therefore in the work presented in this section and more generally throughout this thesis we do not build an artificial learner that explicitly constructs a lexicon of motion primitives each of which represents one abstract gesture and then decompose the global motion as the combination of these gestures. Rather we build a learner which internal representation has combinatorial properties that makes it capable of capturing the compositional aspect of the observed motion. Then, instead of analysing the quality of that representation by an *open-skull* approach, we evaluate the learner on a social task which requires the ability to perceive the combinatorial nature of motions. In this example the task is to reconstruct a symbolic linguistic representation of the motion, where the combinations of symbols matches the high level human perception of motions. That way the agent is not designed to have a purely compositional approach to learning but rather the compositional representation is expected to emerge in order to solve the task: which means that the compositional understanding is a consequence of the teleological understanding (using the terminology of **?**, as introduced in section 1.2). Finally that leads to the important distinction between two aspects of decomposition: the intrinsic compositional properties of the representation and the high level decomposition that is encoded in the structure of the social task. The former is expected to act as a substrate for the latter.

Our contribution, presented in this chapter covers three important topics. First we discuss the nature of combinations of motion primitives that are active simultaneously, in contrast to the study of sequential motion primitives. Then, the experiment performed in the following demonstrated how an agent can leverage the compositional structure of motions to learn primitives that were demonstrated only in an ambiguous manner, which means the learnt primitives where not demonstrated alone but always mixed together. Finally this chapter provides an example of language grounding in the case of a simple symbolic language with a combinatorial structure.

3.1 Combination and discovery of motion primitives

This chapter targets the discovery of motion primitives that can be combined together into complex motions. In this section we review and discuss the various meanings of 'combination', properties of several approaches to motion representation, and algorithms that can be used to decompose motions.

3.1.1 What does combination mean?

Sequencing is probably the most common way to combine motion primitives for robotics. A simple example of sequential combination of motor primitives is: moving to a door, turning the door knob, pushing the door and moving forward.

Despite looking quite coherent with our high level perception, the previous decomposition is hiding a lot of complexity. Indeed, the actual motion executed to open the door might vary greatly depending on the kind and shape of the door knob: opening

a door with a round or a bar shaped knob is very different but perceived as the same high level action. In that case both primitive are combined as two *contextual alternative* into the high level action: if the knob is round, action 1 is executed, else action 2 is used. Similarly two motion primitive can be combined through *probabilistic mixing*: motion 1 is used with probability p and motion 2 with probability $1 - p$. In other words the motion primitive is seen as a probability distribution over motions and the combination of the primitives results from the mixture of the associated probability distribution. The case of contextual alternatives can easily be fitted into that formalism by considering probability distributions over motions, conditioned by the context. Such a formalism is quite general and actually covers most of the approaches encountered in the literature and described further.

Even with contextual alternative and probabilistic mixing, we still miss some structure in our motivating example. While moving to the door one generally also looks at the knob, prepare its hand to open the door, avoid obstacles, and eventually say "good bye", in a completely simultaneous way. It thus appears from a closer look that many primitive motion happen in a *concurrent* manner, which may include several degrees of independence or interaction between the motion primitives. For example one can consider two motions happening *independently*, one on each arm (for example while dancing), or superposed such as a perturbation being added to a reaching motion to avoid an obstacle seen at the last moment. However independence is often too strong an approximation and one must consider the interaction between motion primitives in more details. For robotic applications, it is common that motion corresponds to solving tasks that are prioritized such as grasp an object while keeping the user safe. In such cases primitive motions corresponding to one task can be *subordinated* to a more important task or hard constraints imposed by security. When soft constraints are considered instead of subordination motion primitives can be *competing* with one another. For example while walking and holding a cup of coffee, motions targeted to maintaining the balance of the body are competing with those to maintain the balance of the cup. If the coffee is moving too much the walker might modify his gate to keep the coffee in the cup. If by accident the walker slip on a wet floor he might not be able to keep the cup balance while correcting his own. Finally more complex and general interactions between motion primitive might be considered. High level motions like moving a finger in a straight line, actually involve complex muscle synergies but also stabilization mechanisms that might use other parts of the body. Therefore one could represent that by a motion primitive producing the straight line and *plugged into* a control mechanisms for a muscle synergy, which can be seen as a *functional approach* of motor primitives.

Each of these combinations can account for a subset of complex motions but are not incompatible. However, it is still a difficult problem to find representations that are general enough to cover a large range of possible combinations and still enable efficient algorithms for combining and learning such primitive.

3.1.2 Motion representations

Artificial capture systems perceive motion as a signal, that is to say as a time indexed sequence of values, and can be represented as such: it leads to what can be called a *signal processing approach* to motion representation. Sequences of positions can then be cut in subsequences or concatenated. At each time step, body positions can be

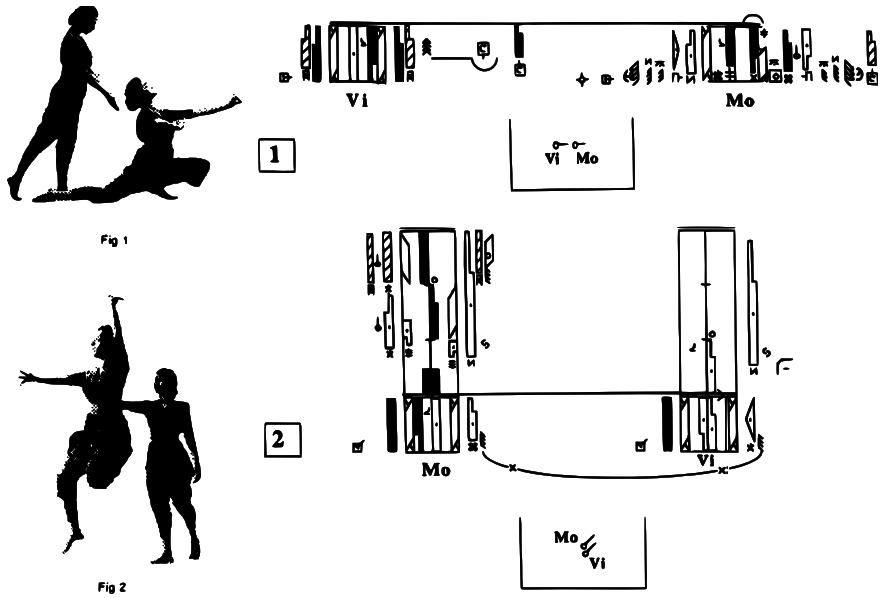


Figure 3.1: Sequential and parallel combinations of motions are well known from choreographers and dancers. This example of dance annotation using Rudolph Laban's system illustrates such structure. Left and right parts correspond to two interacting dancers. Each symbol encodes a gesture, vertical axis corresponds to sequencing of gestures; gestures on the same horizontal line are concurrent. See `http://en.wikipedia.org/wiki/Labanotation` for more information.

the result of the addition of reference body positions encoded as a motion primitives, and small perturbations (for example avoiding an obstacle) coded as other motion primitives. Such decompositions are found in work from **???**.

The usual way to reproduce on a robot a motion recorded and represented that way is to use control in position. Such approach actually hides that forces that need to be applied to body joints to achieve the desired position: they actually transform each position into a low level mechanisms, for examples, a PID[1], that is used to make sure the robot follows the targeted trajectory. Similarly one can use the velocity or the acceleration trajectories to control the robot: for a sufficient sample rate or adequate smoothing, these quantities can be directly derived from the trajectory.

In other words the previous approach is to represent motion primitives as being mappings from time to control commands (for example positions, velocities). Let denote by $\mathcal{T}$ an interval of integers or real numbers and by $\mathcal{A}$ the control space or space of actions; following that view, a motion primitive is a mapping $m : \mathcal{T} \longrightarrow \mathcal{A}$. Actually this is a naive representation of motion that only account for the way it is perceived by simple sensors. That a PID or a similar mechanism is needed to actually produce the motions indeed demonstrates that the nature of motion is more a closed loop interaction between a body and an environment than an open loop

[1] A proportional integral derivative (PID) controller is control loop designed to minimize the error between the measured outcome variable and a desired value. It is named after the three terms used in the feedback function that are proportional to the measured error, its integral, and derivative (See `http://en.wikipedia.org/wiki/PID_controller`).

trajectory. Low-level control to following a predefined trajectory is however not the only aspect of motion that imply closed loop interaction with the environment and the trajectory approach to motion representation is often too restrictive: it does not account for the perceived similarity between two grasping motion that may take the form of completely different body trajectories.

Therefore it is often necessary and useful to represent motion primitives in a way that takes into account the loop between stimulus, reaction of the agent, and its effects on the environment. A common model is to represent the mapping between the *state of the world* and the agent's action. The notion of *world* generally includes both the agent and the environment, or at least what can be perceived by the agent. While defining what the world is is made easy by considering it to be 'everything', it is generally more difficult to define precisely and represent in a computational way its current *state*. The inclusion of the agent in the world is a way to account for internal states of the agent (for example short or long term memory) without considering the whole perceptual history of the agent; in other words, the state representation is assumed to be rich enough for the agent to have a Markovian behavior.

Mappings from time and current state of the system to actions are often referred to as deterministic policies. Denoting the state space by $\mathcal{S}$, a deterministic policy is a mapping $\pi : \mathcal{T} \times \mathcal{S} \longrightarrow \mathcal{A}$. Motion representation are often desired to include some stochasticity; thus the more general notion of stochastic policy (or simply policy) arises, that corresponds to a mapping $\pi : \mathcal{T} \times \mathcal{S} \longrightarrow P(\mathcal{A})$, where $P(\mathcal{A})$ stands for the set of probability distributions over $\mathcal{A}$, assuming it is defined in a meaningful manner. A policy is called stationary when it is constant over time; that is a stationary policy is a mapping $\pi : \mathcal{S} \longrightarrow P(\mathcal{A})$.

An important motivation behind stationary policies comes from decision theory and reinforcement learning (see section 2.2). Reinforcement learning (see **?**, and section 2.2.1) studies agents interacting with a world, abstracted as a state in a state space $\mathcal{S}$, through actions in an action space $\mathcal{A}$, and which ultimate objective is to maximize a reward they get from the world. In the common case where the world and the reward function are assumed to be Markovian, it is known that there exist an optimal deterministic stationary policy. Therefore stationary policies are a relevant representation for the motions of such an agent.

A large set of techniques exist to represent probability distributions and can be applied to policies. The option framework developed by **?**, explains how to build hierarchical policies by extending the space of actions with primitive policies called actions: such approach both represent sequencing and probabilistic mixing of primitives. Similarly, hidden Markov models (HMM, **??**) are used to encode, as a policy, transitions between the agent's internal states and the action they produce; they are also used at a higher level to encode transitions between polices, that is to say their sequential combination (see **??**). In the specific setting where actions are reduced to velocities (assuming an adequate low level controller), **?** chose to instead of directly representing the policy, represent the joint probability over time, positions and velocities: given a position (or a distribution over positions), the policy is obtained as the conditional distribution knowing that position. They use Gaussian mixture models (GMM, **??**, chapter 9) to represent the distribution; the mixture approach is in itself a way to encode probabilistic mixing of very simple primitives, each corresponding to a Gaussian. This approach have been extended to also represent sequencing of primitive by using HMMs to encode transitions between motion primitives (**??**), sometimes

called experts (**?**).

An interesting aspect of the representation of motion primitives as probability distributions over the position-velocity space is that it is a stochastic version of the representation of the phase diagram of the dynamics of the agent. Indeed the agent can be seen as a dynamical system which state s is ruled by a differential equation of the form $\dot{s} = f(s, t)$. The two representations are very close but focus on different aspects of the motion. We have discussed how **?** put in evidence the decomposition of control in a frog limb into control primitive. **??** have extended this idea by building a basis of primitive controller that can be used to provide reaching and tracking abilities to a robot through simple linear combinations of the primitive controllers. The coefficients of the linear combination are then obtained through a simple projection. **?** had previously developed a similar architecture with hand-crafted primitives. **?** and colleagues have studied how the representation as dynamical systems of motions learnt from demonstration can be enforced to satisfy some properties such as the stability of the underlying dynamical system. In **?**, a very simple representation of such a phase diagram for each body joint is shown to be sufficient to recognize dance gestures, as further detailed in section 3.2.

The dynamical system approach to motion representation is also illustrated by **??** who encode motion primitives as a phase[2] indexed perturbation over a predefined dynamical system which encode an elastic attractor dynamics. This representation is called *dynamic motion primitives* (DMP). Stulp and Schaal have further demonstrated how to combine in sequence motion primitive encoded that way (**?**). **?** provide an algorithm to simultaneously learn several alternative policies, represented as DMPs, and use them to solve an episodic reinforcement learning problem.

Finally, since motion primitives often are skills to achieve a certain task, a dual point of view on the problem consist in representing the task itself, eventually together with the associated skill. Such an approach is exemplified in work such as (**?????**). Hart and colleagues further develop the subordinated composition of motion represented in such a way. In our work (**?**) and further wok developed in chapter 4 we focus on the concurrent and competing composition of such tasks.

3.1.3 Algorithms to decompose observed motions

Dictionary learning techniques have been applied to the discovery of motion primitives from sequences. For example, **?** have used orthogonal matching pursuit to decompose complex motions into simple motion patterns activated briefly along the time dimension. The decomposition can then be used to perform both compression and classification. This approach is actually an instance of the *sparse coding* approach to signal processing, which has been extensively studied in other fields (**????**). **?** have also used non-negative matrix factorization to perform a decomposition of globally unstructured motions in low level components and use it in a prediction problem. The use of similar algorithms, but to learn simultaneous motion primitives is detailed in section 3.3. **?** showed that conditional restricted Boltzmann machines can be used to learn a latent representation space for encoding motion primitives.

[2]That notion of phase is a flexible abstraction of time that can evolve non-linearly and easily represent cyclical motions.

Many approaches to representing motion primitives as probability distributions make use of the expectation maximization (EM, **???**, chapter 9) algorithm and its application to HMMs. For example, Kulic and Nakamura have proposed in **?** a method that first performs an unsupervised segmentation of the motion signal into small blocks through a first HMM, and then performs clustering over a HMM representation of the found blocks, thus learning motion primitive as clusters. **?**, have proposed to first discover primitives by clustering action effects on manipulated objects and then use the found clusters and associated segmentation to train parametrized hidden Markov models that allow recognition and reproduction of motions. Finally **?** and **?** use EM for GMMs and HMMs to discover and represent the transitions and sequential combinations of primitives, while **?** use a combination of EM and a custom clustering algorithm.

3.2 Histograms of motion velocity

In our contribution **?** we introduced a simple histogram based representation of motion that can be seen as a rough approximation of the phase diagram of the dynamics of one body joint, as discussed in previous section. This representation is applied to choreographies and is shown to enable the discovery and recognition of primitive gestures.

An important property of such histogram based representation, that makes it usable with techniques like NMF, is that it represents data with vectors of non-negative coefficients which can be combined through non-negative weighted sums.

In this section it is assumed that the motions of a human demonstrator are captured as trajectories in angle and angle velocity spaces of several articulations of the human body. Each trajectory on a specific body articulation (or degree of freedom) is considered separately and the entire sequence of angles and velocities is transformed into a histogram, represented by a fixed length non-negative vector. Vectors obtained for each degree of freedom are then concatenated into a larger vector.

The velocity trajectory are obtained from an angle trajectory that is actually captured: a delayed velocity is used to achieve better robustness to noise in the angle sequences. More precisely $\dot{x}_t = x_t - x_{t-d}$ is used to compute the velocities, instead of being restrained to the case where $d = 1$. It is not necessary to divide by the fixed time step since the histogram representation described in the following is invariant to scaling all the data by the same amount.

In **?**, we explore different approaches for the transformation of angles and velocities sequences into histograms. They differ on two modelling choices:

1. Which data is used to build histograms?
2. Which method is used to build histograms?

Answers to the first question are related to the use of angles and velocities values. While velocities can bring precious information, there are several ways of integrating this information in the histogram representation:

1. consider **only angles**.
2. consider **only velocities**.

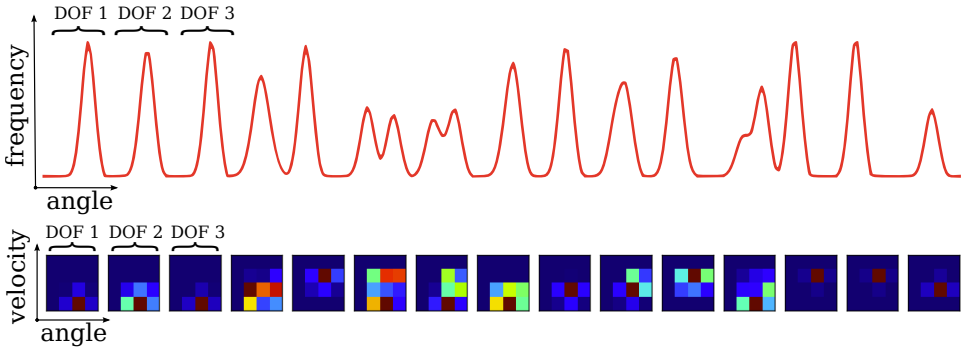


Figure 3.2: Illustrations of concatenated histograms on positions (top) and joint position and velocities (bottom). For the first one, x axis is associated to different possible values for each angles and y axis to frequencies. On the second one frequencies are represented through colors, x and y axis correspond respectively to values of angles and velocities. (Best seen in color)

3. treat **angles and velocities as separate degrees of freedom**.

4. or use the two-dimensional angle-velocity vectors that is to say build histograms on the **joint angle-velocity** space (see fig. 3.2).

We study two methods for building histograms. First, smoothed histograms can be built on **regularly distributed bins**. More precisely we split the angle, velocity or joint angle velocity space into a regular grid of bins. Histograms are built by counting the number of samples from the trajectory falling into each bin and dividing by the length of the trajectory. A Gaussian smoothing kernel is used to make point by point comparison of histograms more robust to perturbations (**?**). These methods are referred to as Kernel Density Estimation (KDE) in the following.

item An alternative approach is to build histograms over a **vector quantization**, which is a more adaptive binning process. Vector quantization (VQ) is performed through a k-means algorithm. Then a histogram is built by counting the proportion of samples falling into each cluster. We explore the use of both hard (each histogram is only counted in one cluster) and softmax (each sample is counted in each cluster with a weight depending on its distance to the cluster's centroid) centroid associations.

Representing motion data by separate histograms on each degree of freedom leads to two approximations:

1. for a given measurement in the trajectory, information about dependency between different degrees of freedom is dropped.

2. the sequential information between measures for a given degree of freedom is dropped.

Similar simplification have however been successfully used in two other fields. **?** have demonstrated that, even if sequential information may appear necessary in language, and especially in speech utterances, very good word discovery can be achieved without considering this sequential information. Both in text classification and in computer vision *bag-of-words* techniques also achieve good performances by dropping positional information of extracted local features (**??**).

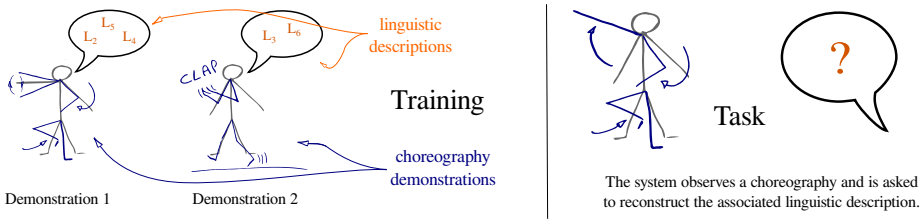


Figure 3.3: In the training phase, the learner observes demonstrations of choreographies composed of several elementary gestures, and the associated set of linguistic labels (left). After that the learner has to reconstruct the set of labels associated to a demonstrated choreography (right). (Best seen in color)

Furthermore using histograms built on joint angle positions and velocities is similar to representing transitions in angle space. By representing the sequence through its transition we approximate it by a Markovian process. Such an approximation is quite common in the gesture recognition and motion planning literature (**??**).

The various methods for building the histogram based representation of motions are compared in the experiment described in next section; relevant results for this comparison are found in section 3.3.4.

3.3 Discover simultaneous primitives by nonnegative matrix factorization

In this section we present results published as **?**. We demonstrate how NMF techniques presented in section 2.1 can enable a system to discover and learn to recognize gestures that are combined simultaneously to form complex choreographies.

More precisely we consider a set of demonstrations each of which is a complex choreography, complex meaning that it is composed of two or three primitive gestures. Each demonstration is associated with a set of symbols that describe the gestures composing the demonstration. Such symbols can be interpreted as an equivalent of the symbols from fig. 3.1 provided to someone unfamiliar with Laban's notation.

A learning system observes the gestures together with the unknown symbols and build an internal (or latent) representation of the data. The symbols are said to be ambiguous since several symbols that describe several gestures demonstrated together, are always given at the same time. Therefore the system has not only to learn a good enough representation of gestures to be able to recognize it, but also has to discover which part of the complex motion is relevant for each symbol.

In a test phase the system is asked, given a new demonstration of a complex dance, to yield symbols corresponding to that dance. The experiment demonstrates that the system is capable of providing correct symbolic descriptions of new demonstrations of choreographies even if the particular combination of gestures was never observed during training: it therefore demonstrates that the system is capable to capture the combinatorial aspect of the observed dance motions. The process is illustrated in fig. 3.3

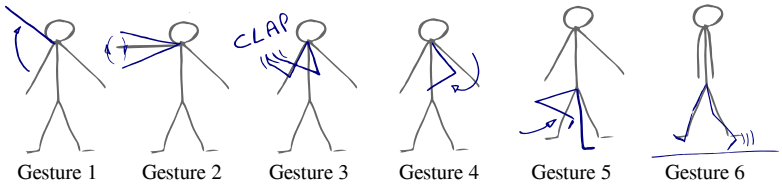


Figure 3.4: A set of the primitive dance movements that are mixed into demonstrated choreographies is illustrated in this figure.

3.3.1 The choreography data

The data used in these experiments is described in more details in section appendix B.2 and available publicly[3]. It has been acquired from a single human dancer through a Kinect™ device and the OpenNI™ software[4] that enables direct capture of the subject skeleton. The device and its associated software provides an approximate 3D position of a set of skeleton points. These points are then converted into 16 angle values representing the dancer position at a specific time. This conversion is achieved through a simple geometrical model of human limbs. It is then converted to position-velocity histograms with the techniques described in section 3.2.

The primitive dance motions used in our gesture datasets and illustrated in fig. 3.4 and table 3.1 are either associated to legs, as for example *squat* and *walk* movements, to both arms, as *clap hands* and *paddle*, or to left or right arm, as *punch*, *wave hand*. Yet this structure is not known by the system initially. They correspond to both discrete and rhythmic movements. The motions were recorded from a single human demonstrator in several sessions. Each motion was selected randomly and the names of the basic gestures that composed it were given to the demonstrator. Recording of the motions occurred through several sessions.

Three motion datasets are considered for these experiments. A first dataset is used to separately study the efficiency of the various representations. In this dataset each demonstration only includes one primitive dance motion. There are 47 different dance primitive and the set contains 326 demonstrations. This dataset is referenced as **single primitive dataset**.

Two other datasets are composed of demonstrations of complex choreographies, composed of two or three randomly chosen compatible (that is spanned over separate degrees of freedom) primitive motions. The first one contains 137 examples of combinations of 16 distinct primitive dance motions. The second one, contains 277 examples with 47 primitive dance motions (the same as in single primitive dataset). These datasets are referenced as **small** and **full mixed dataset**.

Since the datasets only contain a relatively small number of examples we used *leave-one-out* cross validation to build test and train sets. Presented results are averaged over all possible test sets. With the *full mixed dataset* examples presented for testing contain a combination of primitive movements that in 60% of the cases have not been observed during training (see fig. 3.8).

The language description has the following structure: keywords or labels from a

[3]Dataset and examples available at `http://flowers.inria.fr/choreography_database.html`
[4]`http://www.openni.org`

Id	Limb(s)	Description
1	right arm	hold horizontal
5		raise from horizontal to vertical
6		lower from horizontal to vertical
10		hold horizontal and bring from side to front
19	both arms	clap hands (at varying positions)
20		mimic paddling on the left
21		mimic paddling on the right
22		mimic pushing on ski sticks
23	legs	un-squat
24		mimic walking
25		stay still
28	right leg	raise and bend leg to form a flag (or "P") shape
30	left arm	hold horizontal
38		mimic punching
40		lower forearm from horizontal position
43		swing forearm downside with horizontal upper arm

Table 3.1: Primitive dance motions from the *small mixed dataset*

set $\mathcal{L}$ are associated to gestures, and combined into sentences from a set $\mathcal{S} \subset \mathcal{P}(\mathcal{L})$. In this article we only consider symbolic labels. More precisely when the sentence $s = \{l_1, l_2, l_3\} \in \mathcal{S}$ is used to describe a choreography, the system observes a vector $y^i \in \mathbb{R}^L$ (L is the total number of labels, $L = |\mathcal{L}|$) such that for $1 \leq j \leq L$, y^i_j takes value 1 if $l_j \in s$, and 0 elsewhere. For example if 5 labels are considered, the sentence containing labels 1 and 3 would be represented by vector: $(1,\ 0,\ 1,\ 0,\ 0)^T$.

3.3.2 Algorithm

In this section, the NMF algorithm introduced in section 2.1 is applied to the learning problem that we just introduced. This use of NMF in a multi-modal framework was introduced by **??**, and is extended in chapter 6.

In the experiments from this section, we used the plain NMF algorithm based on multiplicative updates, for both errors based on Frobenius norm and Kullback-Leibler divergence (see section 2.1.1).

We assume that we are given a set of examples represented by vectors $v^i \in \mathbb{R}^m$ ($1 \leq i \leq n$), each of which is composed of a part representing a demonstrated choreography and a part representing a symbolic representation of that choreography and use NMF to learn a dictionary of atoms $W \in \mathbb{R}^{m \times k}$ and coefficients $H \in \mathbb{R}^{k \times n}$ such as:

$$V \simeq W \cdot H.$$

Therefore the data matrix V and the dictionary W are composed of a motion and a language part:

$$V = \begin{pmatrix} V_{motion} \\ V_{language} \end{pmatrix} \qquad W = \begin{pmatrix} W_{motion} \\ W_{language} \end{pmatrix}$$

The NMF algorithm only learns, in an unsupervised manner, a transformation of the original data V into an internal representation H. In these experiments it is used in two different ways to first learn the transformation from multi-modal examples to an internal representation, and then use this transformation to reconstruct one modality from another.

In the learning part NMF is applied to a V^{train} data matrix and both W^{train} and H^{train} matrices are learned. The W^{train} matrix is the matrix of most interest since it encodes the structure that has been learned on the data, when H^{train} only encodes the representation of training examples.

The Reconstruction of the linguistic parts associated to demonstrations of motions, that is classifying motions, corresponds to finding the missing $V^{test}_{language}$ given a V^{test}_{motion}. This operation is performed through two steps:

1. reconstructing internal states of the system from demonstrations, which means finding the best matrix H^{test} for the approximation: $V^{test}_{motion} \simeq W^{train}_{motion} \cdot H^{test}$. This step can be performed through NMF algorithms by constraining the W matrix to be constant.

2. once H^{test} has been found, the associated linguistic part can be computed through matrix product: $V^{test}_{language} \simeq W^{train}_{language} \cdot H^{test}$

It should be noted here that the reconstructed matrix $V^{test}_{language}$ is not constrained to take only binary values like the provided linguistic matrix. This issue is addressed by using a thresholding mechanism (where the threshold is learned by cross-validation on training data), as detailed in section 3.3.3.

The value of k is a parameter of the system that is fixed to 150 for the experiments presented in this paper. The number of atoms used by the system to represent observed data is quite important to achieve the desired objective. If k is too big, the algorithm does not have to compress data and can use motion only and language only atoms. On the other hand, if k is too small, the system cannot represent the complexity of the data and may focus on representing components that have bigger effects but less multimodal meaning. In order to demonstrate these capabilities we perform two kinds of experiment.

1. First the system is tested on simple human motions, each containing only one primitive dance gesture. These experiments demonstrate that the motion representation we use is sufficient to perform motion classification, which corresponds to a simple case of multi-modal learning. We also compare different aspects of the representation.

2. Then the system is tested on complex full body human motions to demonstrate its ability to capture the combinatorial structure of the choreographies by exploiting ambiguous linguistic labels.

3.3.3 Evaluation

In each experiment the method based on NMF described in previous section yields a vector of keyword activations, which forms the linguistic reconstruction. The quality of this reconstruction is evaluated by comparison between the reconstructed

$\hat{y}$ (with continuous values) and y from ground truth (with binary values) through the following score functions:

Score function for the single gesture experiment

In that case the good linguistic representation only contains a 1 at the position of the label associated to the demonstrated gesture and 0 elsewhere. The score function is defined as:

$$l_{\text{single}}(\hat{y}, y) = \begin{cases} 1 & \text{if } \arg\max i\hat{y}_i = \arg\max iy_i \\ 0 & \text{else} \end{cases}$$

Score function for mixed gesture: the number of gestures is given

In that case several elementary gestures are present in each example. The reconstructed vector is tested on the fact that gestures that are actually present in the experiment have the best activations.

It can be described by the following equation, where $\#(y)$ denotes the number of gestures present in the demonstration and $\text{best}(n, \hat{y})$ is defined as the set containing the n labels having the best activation in $\hat{y}$.

$$l_{\text{given number}}(\hat{y}, y) = \begin{cases} 1 & \text{if } \text{best}(\#(y), y) = \text{best}(\#(y), \hat{y}) \\ 0 & \text{else} \end{cases}$$

In other words the system is given the number of elementary gestures present in the example and asked which are those gestures.

Score function for mixed gestures: exact reconstruction

This score function evaluates the exact reconstruction of the linguistic description. It requires the reconstructed vector to be converted to a discrete one before comparison.

For that purpose an additional thresholding mechanism is added to the system: after reconstruction of label activations, all values from $\hat{y}$ above a threshold are put to 1, and others are put to 0. The function threshold such that for $1 \leq j \leq L$, $\text{threshold}(y, \eta)_j = \delta_{y_j \geq \eta}$ encodes that mechanism. The threshold η is evaluated through cross-validation on the training data.

The score function is then simply defined as:

$$l_{\text{full}}(\hat{y}, y) = \begin{cases} 1 & \text{if } y = \text{threshold}(\hat{y}, \eta) \\ 0 & \text{else} \end{cases}$$

In each case the score function defined above for one example is averaged over all examples from the test set to yield a final score in $[0, 1]$.

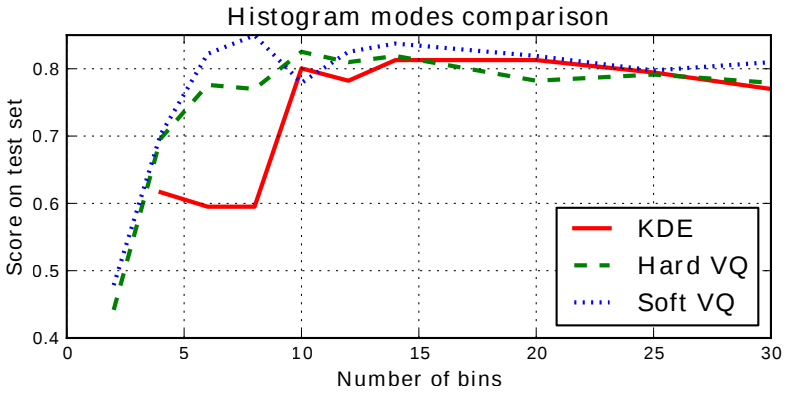


Figure 3.5: When the number of *bins* is small vector quantization (VQ) offers a clear advantage over kernel density estimator (KDE) for representing the angle-velocity distributions (see section 3.2). This advantage however disappear for larger numbers of bins. (Best seen in color)

3.3.4 Results

Demonstrations with a single primitive

We performed a first set of experiments on the *single primitive* dataset in order to evaluate our learning system on a simple multi-modal learning task. In this section primitive dance movements are presented to the learning system with unambiguous labels and the recognition performances are evaluated with the l_{single} score function. We focus on comparisons of the various parameters of the motion representation.

The first experiment compares the use of regular binning with Gaussian smoothing (KDE) and adaptive binning (VQ) with both hard and softmax associations to build the histograms. The comparisons are performed over a range of total number of bins, for **joint angle-velocity** 2D histograms. Results from this experiment in fig. 3.5 outline the advantage of using vector quantization over regular binning (KDE) for small numbers of bins, which corresponds to a low resolution of the input. This difference is however less sensitive for larger numbers of bins. A typical issue of regular binning, that can explain the better results with adaptive binning, is that for the same grid step (the resolution), the number of required bins grows exponentially with dimension. Even with two dimensional histograms, a maximum number of ten bins would lead to a three-by-three (thus low resolution) regular binning. In the same situation adaptive binning can identify ten relevant clusters.

A second experiment is performed to compare the efficiency of histograms built either only on **angles**, only on **velocities**, on **angles and velocities as separate degrees of freedom**, or on the **joint angle-velocity** space. We compare these representations of the motion over a range of values for the delay used in velocity computation, and using KDE histograms with a fixed total number of 15 bins by degree of freedom. The results of the second experiment, presented in fig. 3.6, demonstrate that histograms on joint angle and velocities values capture the most information from the original motions.

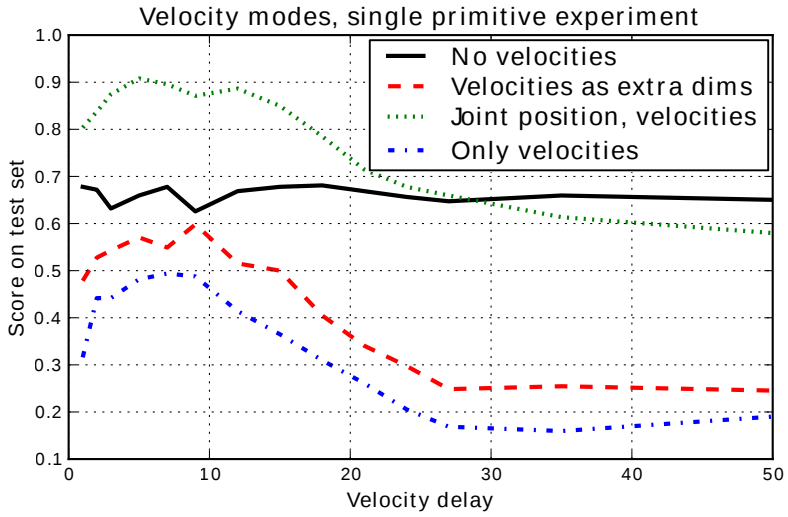


Figure 3.6: For small values of the velocity delay, the representation using jointly the position and the velocity leads to better performances than using only the positions. For high values of the delay, the velocity is mostly noise, therefore using only positions performs better although the noise does not degrade the performance of the joint representation too much. Using only velocities or considering position and velocity histograms as separate dimensions leads to worse performances on the classification task. KDE histograms are used for these results. (Best seen in color)

Demonstrations with complex choreographies

In this paragraph we evaluate the performance of the learning system on the full choreographies with ambiguous labels.

Figure 3.7 illustrates the role of the threshold used to transform vectors of scores produced by the system into vectors with $0, 1$ values representing a set of recognized labels. In the following the threshold (used in l_{full}) is determined through cross validation on the training data.

Table 3.2 presents results obtained on the two mixed datasets for both Kullback-Leibler (DKL) and Frobenius versions of NMF algorithm. The reconstructed label vectors are evaluated by $l_{\text{given number}}$ and l_{full} score functions which enables to understand which part of the error is due to the thresholding mechanism.

For comparison purposes we also tested a method based on support vector machines (SVM) on our dataset. More precisely we trained one linear SVM[5] for the recognition of each label. The SVM method directly yields a set of recognized labels, with no need for thresholding. However this method relies entirely on the symbolic form of the labels and won't generalize to other multi-modal settings with continuous linguistic modalities. There is no such theoretical limitation on our NMF setting (see discussion in section 3.4).

The results in table 3.2 demonstrates that after being exposed to demonstrations

[5]We used the Scikit-learn implementation of SVMs (see `scikit-learn.org/stable/modules/svm.html`).

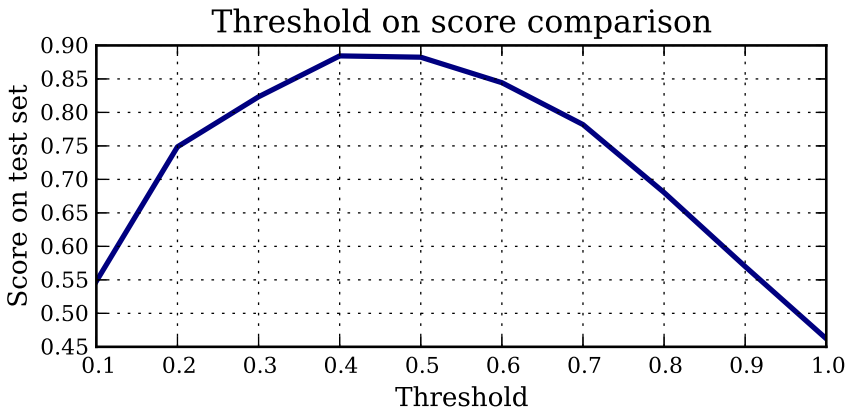


Figure 3.7: The impact of the threshold value on the selection of active labels is limited in a central range of values. Extreme values lead to worse performances.These results are obtained on the small mixed dataset. This experiment is the only one in which the threshold is not automatically adjusted.

	l_{full}	$l_{\text{given number}}$
16 labels (SVM, linear)	0.818	—
16 labels (NMF, Frobenius)	0.854	0.971
16 labels (NMF, DKL)	0.789	0.905
47 labels (SVM, linear)	0.422	—
47 labels (NMF, Frobenius)	0.625	0.755
47 labels (NMF, DKL)	0.574	0.679

Table 3.2: Results on the mixed datasets

of mixed primitive dance motions associated with ambiguous labels, the presented system is able to successfully produce linguistic representations of newly demonstrated choreographies. The second dataset can be considered difficult since each one of the 47 primitive dance motions only appears in an average of 14 demonstrations, which labels are ambiguous.

Handling unknown combinations of primitives

The combinatorial nature of the demonstrated choreographic motions implies that, although the primitive gestures are observed many times, each specific combination of gestures into a choreography is not observed that often. This phenomenon is illustrated in fig. 3.8. An important consequence is that the performance of the system cannot be solely explained by the system recognizing full choreographies; rather the system has captured the combinatorial structure of the data. This ability is illustrated by its behaviour on unknown combinations of motion primitives. For instance in the *full mixed dataset* more than 60% of the examples demonstrates a combination that is not observed in other examples.

In order to get more precise results for this behaviour we set up a slightly different

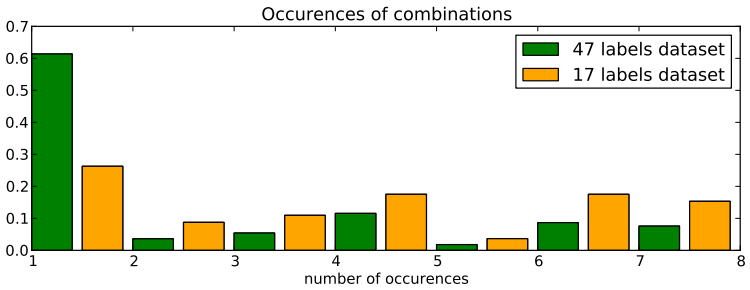


Figure 3.8: In the big dataset 60% of the choreographies (specific combination of primitive gestures) are only observed once. This illustrates the necessity for the system to capture the combinatorial structure of the choreographies instead of representing them in a holistic manner.

	l_{full}	$l_{\text{given number}}$
16 labels (NMF, Frobenius)	0.568	0.800
16 labels (SVM, linear)	0.667	—
47 labels (NMF, Frobenius)	0.406	0.653
47 labels (SVM, linear)	0.206	—

Table 3.3: Results for combination of primitive gestures that were not encountered during training.

experiment where test sets are only composed of combinations of motion primitives that were not observed during training. The results of this experiment are reported in table 3.3.

3.4 Concluding perspectives

In a first experiment from previous section, we demonstrated the efficiency of the position-velocity histogram representation of motion and the NMF algorithm on a dance motion recognition task. **?** had used NMF to perform prediction on learnt motions; in contrast, our contribution extends the scope of NMF usage to a form of structure classification of motions. The motion representation presented in this section enables the application of the architecture developed by **??** for speech learning to motion learning. This constitutes a useful step toward comparison of structural similarities between language and motion learning. That study is further extended in chapter 6 and **?**.

The motion-velocity histogram representation does not make it possible to produce the actual motion from the motion representation. **?** have given an example of motion representation that allows such reproduction. This work may thus be extended, by changing the motion representation, to an imitation learning setting, in which the system could be evaluated on producing gestures on a real robot, corresponding to a given linguistic description. Chapter 4 introduces ideas toward such motion representations, although they are not illustrated on a real robotic setting.

In the second experiment we showed that the architecture presented in this paper is capable of learning to recognize complex gestures, composed of simultaneous motion primitives, while only observing ambiguous symbolic labels. It is demonstrated in the third experiment that the system has captured the combinatorial structure of the observed gestures and is capable of generalization by recognizing combinations that were never observed in training.

We presented a learning system that is capable, after learning from demonstrations of complex gestures and linguistic descriptions, to re-construct the linguistic modality from an example involving only the motion modality. The experiments that we performed only uses a symbolic representation of speech labels. Interestingly this corresponds to a degenerated version a talking heads (see section 1.3 **??**). For this analogy we consider that a first human is performing the gesture. During training, the agent presented here corresponds to a *hearer* from the talking heads, the *speaker* being another human, who names the choreographies. In this experiment the communication channel is discrete and finite and corresponds exactly to the human speaker ontology. The setup we present describes how the learner builds its own ontology and produces utterances when it, in the test phase, plays the speaker. It is however possible to replace this symbolic representation by real acoustic data (for example represented in the same way than in **?**) without changing the learning and reproduction algorithms. Such experiments are performed in our other contribution **?** and detailed in chapter 6.

While in this contribution we focused on primitive motions active at the same time, it is possible to use the same setting to recognize choreographies where motions are composed in sequence and eventually overlaps. A direct application of our method, would however only enable reconstructing the set of active motions and not their order.

Chapter 4

Learning a dictionary of primitive tasks

Despite having potentially many distinct definitions, the notion of *task* or *targeted effect* is very important to apprehend human behaviors or program robots. Actually the common representation of motions as a sequence of positions or basic actions, either captured from an acting agent or implemented on a robot, might be very limiting. For example, in order to grasp an object, an adult would perform some motion of its arm and hand toward the object while a dog would use its head, a child would get on tiptoe and use a tool to reach the object if it is too high for him. Even the same person would use very different motions depending on the position or accessibility of the object. Although the trajectory of body parts are very different in each of these examples, the intention of the agent is generally perceived very clearly as identical in each case by an observer. More generally, the perceived similarity between two motions is often not explained by the similarity of the underlying motions.

Therefore it is of great interest for objectives like human behaviour understanding or programming robots capable of complex tasks to be able to take into account activities at the task or effect level. Learning or perceiving actions at the effect level was already discussed in section 1.1 as *functional level imitation* as introduced by **?**. The motivation behind such representation is to achieve better generalization to unknown situations while preserving the essence of the task or to be able to learn and improve from imperfect demonstrations.

The notion of *affordances*, that is to say the potential actions related to encountered objects, was introduced by the psychologist **?**. Its importance in the perception of objects emphasizes the close relation between the knowledge of achievable effects on the environment and the perception of that environment. Similarly, an interesting analogy can be drawn with the field of *pragmatics* amongst linguistics: pragmatics emphasizes the importance of the speaker's intention over pure semantics for communication (**?**). That approach suggests that sentences or language are not produced as a container for meaning but rather as a way to induce an effect on the person to whom it is directed.

In this chapter several studies related to learning and using the effect space or task space are presented. Some of them address related issues such as the question of

learning the effect of actions or learning relevant aspects of the effects of an action, denoted *task space retrieval*. In the context of complex or composite actions that are studied in this work, we believe that sometime the structure of the behavior is easier to study at the task level than at the motion or trajectory level; contributions about learning task components are therefore presented in section 4.3.

The work presented in the following is similar to the one from previous chapter but study behaviors in the task space instead of the action or motion space. In contrast to previous chapter, the behavior representation studied below is not as destructive and enables reproduction of the learnt behaviors with planning algorithms. Finally the behavior composition introduced in the following can be qualified as 'concurrent', rather than just 'simultaneous'.

4.1 Previous work

Plan recognition and *intention recognition* focus on the study of the structure of tasks abstracted from sequences of symbolic actions (for a review, see **?**). These ideas have been applied to teach robots by demonstration in various ways.

? use a set of static mechanisms to abstract visual sensory input into objects and recognize the effect of predefined actions on the environment, as well as the dependencies between the actions in a given task. From these mechanisms their robotic system is capable of reproducing a demonstrated task by following a sequence of sub-tasks that fit the learnt dependency. **?**, **?**, and **??** have developed similar setups, but including more advanced dependencies between sub-tasks and hierarchies. **?** present an approach in which a robot is capable, from predefined primitive behaviors, to learn the precedence relations and dependencies between these behaviors combined into a solution for a task. The relations are learned following a *specific to generic* process: each time a new demonstration is observed, the model of the task is expanded to account for the new demonstration. Similarly, **??** use a longest common path heuristic to infer a directed acyclic graph representing the relations between predefined behaviors combined to solve task. They also explain how teacher feedback can be efficiently used to correct errors occurring in the learning phase or coming from wrong demonstrations. They also detail the analogy between their task model and regular expressions or equivalently finite automata. **?** also present a similar system that, after learning a set of basic behaviors in a supervised way, can learn a directed acyclic graph, called *task precedence graph* that forms an abstract representation the task structure. The graph is learnt by successive generalization, similarly to the work of **?**.

While the studies presented in previous section mainly focus on methods to learn the structure of a task, knowing the underlying primitive behaviors, other works focus on learning the primitive actions that compose complex tasks; for example **?** use clustering on the effects of various actions on the environment to identify such primitive actions.

4.1.1 Inverse feedback and reinforcement learning

Inverse reinforcement learning and inverse feedback learning, that are presented in section 2.2, both study models of an agent behavior and intention as a process to maximize a reward or feedback function. Both have been developed in order to yield better models of intention of agents performing tasks. For example, **?** demonstrate how applying inverse reinforcement learning to expert demonstrations of aerobatic manoeuvres on a remote controlled helicopter enables a learning system to succeed in accomplishing manoeuvres on which the experts only provided failed demonstrations. Similar ideas have also successively been applied to represent driving styles on a simulator (**?**) and trajectories from taxi drivers (**?**), pedestrians (**?**), and wild animals (**?**).

However, basic inverse reinforcement learning or inverse feedback learning focus on one simple reward or objective function that explains all the data it observes passively data. Such setting does not fit many practical situations.

? consider the situation in which the robot or learning system can request demonstrations at specific states in order to make the learning process require less demonstrations: having human provide demonstrations for robots is indeed often considered an expensive operation. Lopes and Cakmak also presented an optimal teaching strategy for inverse reinforcement learning (**?**): they provide an algorithm to chose optimal demonstration that are to be given to an inverse reinforcement learner.

? present an approach inspired from inverse optimal control that learns a feedback function representing demonstrations of a task that is sparse on some features. They demonstrate how the algorithm is capable to learn a grasping task and explain why the sparseness of the learnt feedback function provides a solution to the task space retrieval problem.

An other common issue of inverse reinforcement learning approaches is that it is often required that adequate features are used to represent the state of the learning agent. **?** introduce an algorithm to overcome this issue that both learn an estimate of the reward optimized by the demonstrator and features to efficiently represent it. They also demonstrate how the learned features can be transfered to a new environment. The work presented in the following of this chapter also goes toward that direction in the more general setup of multi-task demonstrations.

While in the works presented above the expert only demonstrates a single task, **?** present an EM-based algorithm which, from unlabeled demonstrations of several tasks, infers a clustering of these demonstrations together with a set of models. **?** have extended this idea into a reward model based on Dirichlet processes and a Metropolis-Hasting algorithm that represent the demonstrations as being generated from mixture of reward functions without requiring the number of reward functions to be fixed. **?** also recently developed similar ideas in the context of inverse feedback learning.

? suggest that demonstrated behaviors, even in the single task setting, might be better modelled by considering non-stationary rewards functions: they developed an algorithm that learns a set of reward functions such that each sample action is explained by one of these functions.

? have extended the ideas of *modular reinforcement learning* for inverse reinforcement

learning. This approach targets the issue of the exponential growth of state space often observed when real world applications are modelled.

The next sections present two algorithms that extend both inverse reinforcement learning and some inverse feedback learning techniques to build a dictionary of primitive tasks. The primitive tasks are modelled as rewards or feedback functions that can be combined together to model the intention of an expert in demonstrations of complex tasks. They focus on the question of how to represent demonstrated behaviors as the combination of simpler primitive behaviors that can be re-used in other contexts. This approach is similar to feature learning but focus explicitly on the multi-task setup. We present two algorithms that, by observing demonstrations of an expert solving multiple tasks, learns a dictionary of primitive behaviors and combine them to account for the observations. We explain how such an approach can be beneficial for life long learning capabilities in simple toy experiments. Next section uses inverse feedback learning to discover task primitives from demonstrations of an agent in a continuous world. The experiments illustrate the ambiguous nature of the task and how supervision can be used to accurately recovers the original dictionary. The following section presents a similar algorithm in the case of inverse reinforcement learning, in a discrete world. The experiments demonstrate that without supervision the learner can build a dictionary to represent the demonstrated task. The evaluation of the experiments focus on the ability of the learner to solve a task after a few observation of an expert solving that task. The results demonstrate that learning the common structure of composite tasks can make the agent more efficient in solving new tasks that share the same structure.

4.2 Factorial inverse control

? has developed an algorithm based on least square regression to learn *potential functions* modelling the motion of wild animals in natural parks. In this section, and in the publication **?**, we present an algorithm that extends Brillinger's technique to address a new problem: instead of learning a flat representation of a single task, the learner must infer several primitives cost functions that can be composed so that their mixture is a good approximation to the demonstrated task. A very similar behaviour representation is used, but it introduces dictionary learning for solving the new problem. We discuss the use of supervision, such as linguistic supervision, to improve and disambiguate the learning of the dictionary.

4.2.1 Problem definition and algorithm

This section describes a simple synthetic imitation learning experiment in which an imitator learns to reproduce behaviors observed from a demonstrator: the task underlying each behavior is modelled as a cost function on states of the agent (either the demonstrator or the imitator), which can be seen as representing the preferences of the demonstrator. For example the task of filling a glass of water is represented by a cost function giving increasing values to increasing levels of water in the glass. In the case where the "filling the glass" behavior is mixed with the "smiling to someone" behavior, the mixed behavior is be represented by a mixed cost function valuing both full glass and smiling position of the lips.

Each demonstration consists in a trajectory in the demonstrator state space, from a specific initial position. The objective of the imitator is to produce a trajectory that fits the demonstrator preferences, that is minimise the cost function. The imitator may start from the same initial position than the demonstration or another. The latter generally defeats strategies that simply mimic the demonstrator gestures; this issue, that oppose program level imitation to action level imitation, is discussed in section 1.1.

This setup introduces two important difficulties for the imitator. On the one hand each demonstration only presents aspects of the cost function locally, around the trajectory. Each demonstration is thus not sufficient to fully understand the underlying task. On the other hand, each demonstration presents a mixture of several tasks. Thus, while the primitive tasks are observed many time, they are never observed alone and each particular mixture is generally only observed once. It is thus necessary to leverage the compositional structure of the behaviors to be able to understand them, and reproduce them with new initial positions.

Agent and demonstrator models

Both the demonstrator and imitator are assumed to have identical bodies and perceptions of the world. This corresponds for example to the case where demonstrations are performed on the imitator body (kinesthetic demonstrations). Following **?**, the current configuration of the robotic agent q belongs to a state space $\mathcal{Q} \in \mathbb{R}^S$. Each trajectory is denoted by a sequence $(q_t)_{t\in[|1,T|]}$.

The model assumes that there exists a cost function $f: \mathcal{Q} \longrightarrow \mathbb{R}$ such that each task is modeled as the demonstrating agent trying to minimize the cost $f(q)$ to which is added a penalization on the square norm of $\frac{\partial q}{\partial t}$. The penalization term can be seen as a penalization of the energy consumed while moving toward an optimum of $f(q)$.

The following focuses on very simple agents whose actions are motions in the state space and are governed by the local optimization of $f(q) + \alpha \left\| \frac{\partial q}{\partial t} \right\|^2$ which means that each action, at each time step, is chosen such that:

$$q_{t+1} = \arg\min_q f(q) + \alpha \left\| \frac{q - q_t}{\delta_t} \right\|^2,$$

with δ_t the time elapsed between samples t and $t+1$.

The solution of this equation, without additional constraints, and assuming that the cost function f is differentiable, is well known to be proportional to the gradient of f, as $-\frac{1}{\alpha}\nabla f(q)$.

It can be noticed that since the agent previously defined only follows policies driven by local optimization it will only achieve local optimization of the cost function. While this is a simplification of the agent, it also features an important property of real demonstrators: real demonstrators are in general imperfect and do not always succeed in reaching the optimal solution of the task. It is thus important for an imitator to be able to also learn from imperfect demonstrations of behaviors.

Complex tasks are more specifically studied here: each demonstration corresponds to the minimization of a separate cost function f which is only observed through

one demonstration. However f is composed of parts that also occur in other demonstrations and are thus observed several time mixed in various way and in various contexts. We consider N demonstrations, observed as trajectories $(q_t^i)_t,\ i \in [|1, N|]$ in the agent state space. This work assumes that each demonstration corresponds to a given f^i. To model complex demonstrations it also assume that there exists a dictionary of primitive tasks, composed of K cost functions $(g^k)_{k\in[|1,K|]}$, such that, for all demonstration i, there exist coefficients $(a_k^i)_{k\in[|1,K|]}$ such that, for all state q, $f^i(q) = \sum_{k=1}^{K} a_k^i g^k(q)$.

In the following, we first present the inverse feedback learning approach from **?**; then we extend it into a learning algorithm which observes one demonstration associated with each function f^i and learns a dictionary of primitive cost functions g^k, and the coefficients of their combinations into demonstrated tasks f^i.

Inferring a task from a demonstration

The problem of inferring a single task from a demonstration is studied in Brillinger's article (**?**). The cost function is represented by a linear parameter $\beta \in \mathbb{R}^F$ on a space of potentially non-linear features $\varphi :\ \mathcal{Q} \longrightarrow \mathbb{R}^F$. Its minimization is modeled by an agent policy such that:

$$\frac{\partial q}{\partial t} = -\lambda \mathbf{J}(q)^T \beta \tag{4.1}$$

where $\mathbf{J}$ is the Jacobian of φ (lines of $\mathbf{J}$ are gradients of coordinates of φ).

When discrete trajectories are considered, eq. (4.1) may be approximated as: $\frac{q_{t+1}-q_t}{\delta_t} = -\lambda \mathbf{J}(q_t)^T \beta$ for all $t \in [|1, T-1|]$. By denoting $y_{t+1} = \frac{q_{t+1}-q_t}{\delta_t}$, $Y \in \mathcal{R}^{S\times(T-1)}$ the vector obtained by vertically stacking all y_t for $t \in [|2, T|]$, and $\mathbf{\Phi}$ the $S \times (T-1)$ by F matrix obtained by vertically stacking all $-\lambda \mathbf{J}(q_t)^T$, we get:

$$Y = \mathbf{\Phi}\beta \tag{4.2}$$

Equation (4.2) transforms the problem of inferring one task from one demonstration into a linear regression problem, which constitutes an essential contribution of Brillinger's article.

In the case where the Euclidean distance between the vector Y, computed from observations, and its reconstruction through the task model $\mathbf{\Phi}\beta$ is considered, we get the classical least square regression problem. It is solved, assuming $\mathbf{\Phi}^T\mathbf{\Phi}$ is non-singular, by:

$$\beta = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T Y \tag{4.3}$$

More details on the associated derivations can be found in **?**. The algorithm presented above is capable, from one demonstration, to infer the cost function modelling a behavior of the demonstrator. Once the cost function is inferred, the imitator can in turn produce trajectories that minimize it. Such an agent that directly infers all the parameters of the cost function is denoted **flat imitator** in the following.

Learning a dictionary of primitive tasks from mixed demonstrations

The algorithm presented in previous paragraph only applies to a single demonstration generated from a single task model. Here we introduce a matrix factorization algorithm that extends the previous method to a setting where a dictionary of primitive tasks is learnt form several demonstrations.

Each demonstration corresponds to a mixing of primitive tasks which is modeled by a β^i in the feature space. A dictionary that is represented by a F by K matrix $\mathbf{D}$, such that each column of $\mathbf{D}$ is the parameter representing the primitive tasks g^k in the feature space, models the concurrent mixing of primitive tasks. The concurrency between the primitive tasks in a mixing is represented through a weighting coefficient. Coefficients of the i^{th} demonstrated task are given by a vector $a^i \in \mathbb{R}^K$, $\beta^i = \mathbf{D}a^i$.

For each demonstration the vector Y^i and the matrix $\mathbf{\Phi}^i$ are associated with the observed trajectory, by following the method described above. It follows that for each demonstration:

$$Y^i = \mathbf{\Phi}^i \mathbf{D} a^i \tag{4.4}$$

Learning a factored model of the demonstrated tasks that has the minimum Euclidean distance to the demonstrations is equivalent to solving eq. (4.5).

$$\arg\min_{\mathbf{D},\mathbf{A}} \mathcal{L}(\mathbf{D},\mathbf{A}) \quad \text{with} \quad \mathcal{L}(\mathbf{D},a) = \sum_{i=1}^{N} \left\| Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \right\|_2^2 \tag{4.5}$$

We propose an algorithm based on alternate minimisation with respect to $\mathbf{D}$ and $\mathbf{A}$ to solve this problem.

Minimisation with respect to A This sub-problem assumes that the dictionary is known and thus consist in inferring the task decomposition on the dictionary, from the observation of a demonstration. It is similar to the algorithm presented in previous section but the K decomposition coefficients (the vectors a^i) are inferred instead of all the F coefficients of the cost function.

This problem is separable in one sub-problem for each demonstration i, each of which is equivalent to the regression problem from **?** presented previously: the matrix $\mathbf{\Phi}$ is now replaced by the product $\mathbf{\Phi}^i\mathbf{D}$. Thus the solution of the optimisation with respect to $\mathbf{A}$ is given, for Euclidean distance, by eq. (4.6). Other norms or penalization could as well be used to solve the regression (for example methods enforcing non-negativity or sparseness of coefficients).

$$a^i = \left(\mathbf{D}^T {\mathbf{\Phi}^i}^T \mathbf{\Phi}^i \mathbf{D}\right)^{-1} \mathbf{D}^T {\mathbf{\Phi}^i}^T Y^i \tag{4.6}$$

Minimisation with respect to D The second sub-problem assumes that the decomposition coefficients of the demonstrated task are known but not the dictionary $\mathbf{D}$. We use a gradient descent approach to learn $\mathbf{D}$. The differential of the loss with respect to each of the coefficients of $\mathbf{D}$ is given by eq. (4.7).

$$\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D},\mathbf{A}) = -2\sum_{i=1}^{N} {\mathbf{\Phi}^i}^T \left[Y^i - \mathbf{\Phi}^i \mathbf{D} a^i \right] {a^i}^T \tag{4.7}$$

Global algorithm The global algorithm simultaneously learns the dictionary $\mathbf{D}$ and the coefficients $\mathbf{A}$ by alternation of the two procedures from previous paragraphs. Matrices $\mathbf{D}$ and $\mathbf{A}$ are initiated randomly or according to any heuristic. Then $\mathbf{D}$ is learnt, assuming $\mathbf{A}$ contains the correct decomposition coefficients, after which $\mathbf{A}$ is inferred assuming $\mathbf{D}$ is the correct dictionary, and so on. This approach to matrix factorization problems has often proved to be efficient (**??**).

4.2.2 Experiments

To illustrate the algorithm introduced in section 4.2.1 we consider a simple toy experiment. We define an agent which state q belongs to $\mathcal{Q} = [0,1]^2$. Cost functions are parametrized on a 5 by 5 grid of Gaussian radial basis functions, which means $\phi(q)^T = (\dots, \frac{1}{2\pi\sigma}\exp(-\frac{\|x-\mu_f\|^2}{2\sigma^2}), \dots)$ where μ_f are points from a regular 5 by 5 grid on $\mathcal{Q}$ and σ is fixed such that the task parameter space is of dimension $F = 25$. There is no difference between the demonstrator agent and the learner, except that the demonstrator fully knows the task to perform.

We use in this experiment a dictionary of 6 primitive tasks that is represented in fig. 4.1 (first row). Combinations of 2 or 3 concurrent primitive tasks are generated randomly for training and testing. For a given mixed tasks, a starting point is randomly chosen inside $\mathcal{Q}$ and trajectories are generated by the demonstrator or imitator from the initial position, according to eq. (4.1). In the remaining of this section we will describe two separate experiments where a dictionary is learnt by a agent observing mixed combinations of tasks.

Recovering the dictionary from given coefficients

In this section we consider an experiment in which during training the learner both observes demonstrations of mixed tasks and the associated mixing coefficients. This hypothesis models the situation where some labels that describe the task that are mixed together in the demonstration are given to the learner (for example inferred from spoken language). This experiment enables the evaluation of the second part of the algorithm we introduced.

Since the mixing coefficients are known by the learner during training, only the second part of the algorithm presented in section 4.2.1 is used to learn the dictionary $\hat{\mathbf{D}}$. We train such a learner on 200 trajectories generated from a dictionary $\mathbf{D}$. Both the original dictionary of primitive tasks $\mathbf{D}$ and its reconstruction $\hat{\mathbf{D}}$ are represented in fig. 4.1.

Once the imitator has built a dictionary of tasks from observations, it is evaluated in the following way: for a set of coefficients, corresponding to mixed tasks, and a random starting position, the imitator and demonstrator yield trajectories. The demonstrator and imitator trajectories are then compared. Examples of trajectories from both the learner and the imitator are given in fig. 4.2.

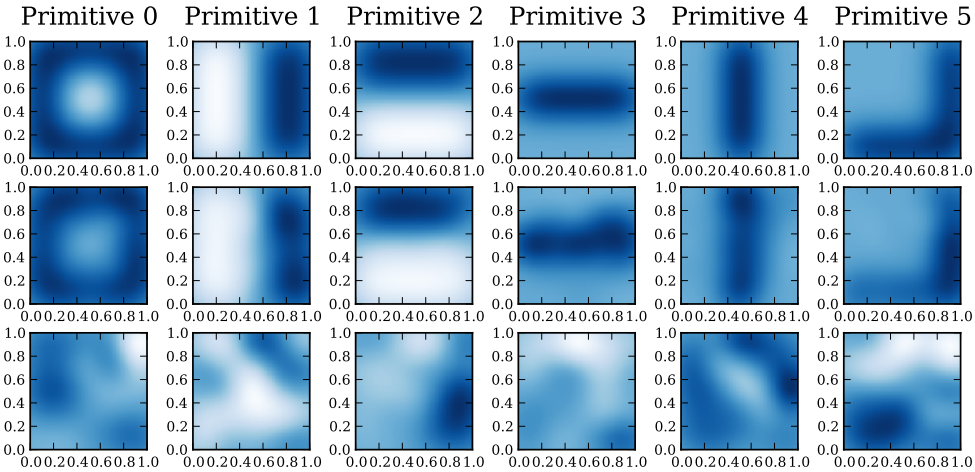


Figure 4.1: The supervised learner achieves accurate reconstruction of the original dictionary of primitive tasks while the unsupervised learner acquire its own representation. Each row present a dictionary of primitive tasks. The tasks are represented as cost functions over $\mathcal{Q} = [0, 1]^2$. Dark areas corresponds to high positive costs and light areas to negative costs. First row corresponds to original primitive tasks (as used by the demonstrator), second row to the one reconstructed by the supervised learner described in and third row to the unsupervised learner. (Best seen in colors)

The relative L_2 error between the trajectories generated by the demonstrator and the imitator is used to evaluate the quality of the reconstruction. An average error of 0.001127 is obtained on the train set (tasks observed while learning the dictionary) and 0.002675 is obtained on the test set (unobserved tasks obtained from the same dictionary).

Learning both primitive tasks and mixing coefficients from concurrent demonstrations

We illustrate the full unsupervised algorithm presented in section 4.2.1 on an experiment where the learner only observes demonstrated trajectories without knowing the coefficients. The bottom row of fig. 4.1 presents an example of dictionary built by such a learner.

Once the dictionary has been learnt, we use the following imitation protocol to test the imitator. A new unobserved combination of primitive tasks is chosen together with an initial position. Then the demonstrator provides a trajectory corresponding to that task. From the observation of the demonstrated trajectory and the its learnt dictionary of primitive tasks, the learner infers the decomposition of the task on the learnt dictionary. For that it uses the first part of the algorithm presented in section 4.2.1. Finally the imitator is asked to produce trajectories that solve to the same task, both from the demonstrator's initial position and new random initial positions. Changing the initial position is a way to evaluate how well the imitator's model of the task generalizes from the context of the demonstration to new ones.

In order to evaluate the impact of learning the dictionary, that is to say the combi-

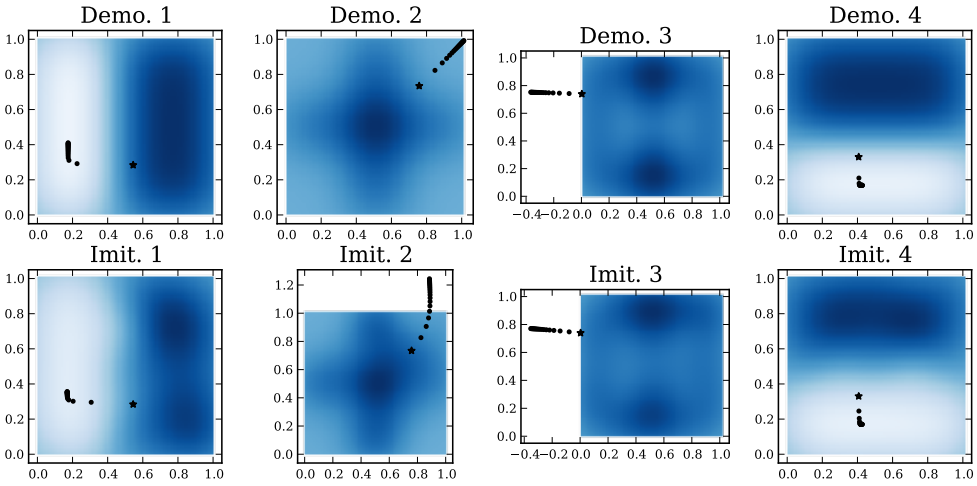


Figure 4.2: The trajectories from the demonstrator are well reproduced by the supervised learner. First row represents four demonstrated tasks (dark color represent areas of high cost) and the demonstrated trajectory (the star represent the initial position and dots further positions). Second row represents the model of the task as inferred by the imitator and the trajectory followed by the imitator solving its model of the task, from the same initial position as the demonstrator. (Best seen in colors)

natorial structure of the demonstrated data, we compare reproductions of the task by an agent that has learnt the dictionary denoted as *full dictionary learner*, to ones by an agent, denoted as *flat imitator*, that directly infers the parameters of the tasks without using a dictionary. We also compare the agent described in the previous section, that has learnt the dictionary from both demonstrated trajectories and mixed coefficients, denoted *dictionary from coefficients learner*. Examples of demonstrated and imitated trajectories are provided in fig. 4.3.

4.2.3 Discussion

The first, supervised, agent is able, by observing motions solving composed tasks and the mixing coefficients, to learn the dictionary of primitive tasks. The acquired dictionary is evaluated in different ways: visually from the plots of the associated cost functions, from trajectories solving a mixed task whose mixing coefficients are given, and from imitation, in random contexts, of a mixed task that is inferred from a single demonstration (this last result is presented together with second experiment).

The second, unsupervised, agent learns a dictionary that enables the factorial representation of demonstrated tasks, without directly observing neither the dictionary nor the mixing coefficients. The factorial representation enables imitation of tasks that are observed through a single demonstration. However the performance evaluation does not validate quantitatively this capability. In particular the least square regression from **?** (described in section 4.2.1) is not performing well on the particular form of mixing of cost functions we have chosen for the illustrative toy example. However our algorithm is compatible with any regression method. Thus, interesting further work could use the comparison of performances between various regression

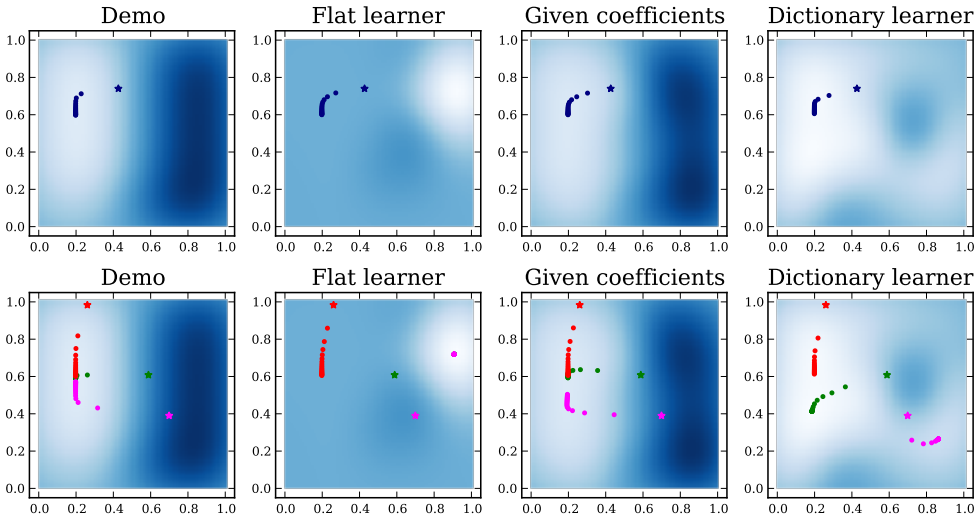


Figure 4.3: Examples of imitated trajectories. First row presents the demonstrated trajectory (first column) and its imitation by the flat learner, the dictionary learner from first experiment (coefficients observed while learning the dictionary) and the full dictionary learner. Second row correspond to imitations of the same task from initial positions that were not observed (the demonstrator trajectories for those positions are given for comparison purpose). (Best seen in colors)

methods, on real human data, to get better insight on the combinatorial properties of human activities. The next section present similar ideas applied using the inverse reinforcement framework; more reliable results are obtained on an other toy experiment.

The dictionary learnt by the agent, illustrated in fig. 4.1, is very different from the one of the demonstrator: actually chapter 1 already discussed the ambiguity of the problem of representing a set of demonstrated mixed tasks as linear combinations of primitive tasks. For example one can scale the primitive cost function by some factor and associated coefficients by its inverse or change the order of the primitive and coefficients without changing the linear combination. Mathematically these difficulties could be solved by adding constraints to the form of the learnt dictionary (for example normalize primitive costs) or by adapting the way to compare dictionaries (for example to make it invariant to re-ordering).

To overcome this difficulty, a particular form of factorisation could also be shaped by information coming from another modality or social interaction. This aspect is demonstrated both in the study from our previous work (**?**) that is described in chapter 3 and in the first experiment, see section 4.2.2, where observing the mixing coefficients, that can be seen as linguistic labels, enables the learner to adapt its internal model (that is the dictionary) to a communication channel. Aspects of social learning have already been shown to improve motor learning by **?**. Solving the ambiguity in the decomposition of human activities thus constitutes a new application for social learning. Further illustrations on learning from several modalities are given in chapter 6.

Extending the algorithm presented above to include constraints or evaluating it on

an online learning experiment would help investigating these questions and thus constitute very interesting future work.

In conclusion, this section studies aspects of the combinatorial structure of behaviors and of their representation as tasks or objectives. We introduced an algorithm to learn a dictionary of primitive tasks from demonstrations of concurrently mixed behaviors. We demonstrated on an illustrative experiment how the dictionary can be used to represent and generalize new demonstrations. Finally we discussed how dealing with ambiguities in factorial representation of behaviors might involve social interactions, multimodality of the sensory experience or intrinsic saliency mechanisms. However the illustrative experiment from this section did not enable to quantitatively demonstrate the advantage of the unsupervised factorial approach. Next section develops similar ideas with inverse reinforcement learning techniques.

4.3 Factorial inverse reinforcement learning

This section present ideas similar to the one introduced above but grounded on different techniques and models. The algorithm presented in this section extends the gradient approach from **?** to learn a dictionary of primitive reward functions that can be combined together to model the intention of the expert in each demonstration. It includes both previous ideas on multi-task inverse reinforcement learning (**???**), but also of feature learning and transfer between tasks (**?**). In its unsupervised study of the multi-task setup, this work is related to those of **?**, **?**, and **?** but differs for the fact that it not only learns several primitive rewards from demonstrations of several tasks but also enables transfer of the knowledge from one task to an other, similarly to what is presented by **?**.

4.3.1 Multi-task inverse reinforcement feature learning

This section presents an extension of the algorithm from **?**, described in section 2.2.3, to a dictionary learning problem. We assume that the learner observes demonstrations ξ from an expert, that are sequences (x_t, a_t) of states $x_t \in \mathcal{X}$ and action $a_t \in \mathcal{A}$, such that $\xi = (x_t, a_t)_{t\in[|1,T|]}$. We call Markov decision process (MDP) a quintuple $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ where γ is the discount factor. P, the transition probability, is a mapping from state actions pairs to probability distributions over next states. We denote by $P(x'|x, a)$ the probability to transition to state x' from state x, knowing that action a was taken. Finally, $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function.

In this single task setup presented in section 2.2.3, the objective is to optimize a parameter θ of the reward function such that the near-optimal policy for the MDP associated to r_θ is as close as possible to the demonstrated policy. This optimization problem is formalized into For an expert demonstration represented by $\xi = (x_t, a_t)_{t\in[|1,T|]}$ one estimates J by eqs. (2.30) and (2.31). The main result provided by **?** and presented as section 2.2.3 is an equation to compute $\nabla_\theta J_\xi$, from which the gradient descent algorithm is derived.

In this section we extend there approach to the case where the expert provides several demonstration of different but related tasks. The demonstrations are denoted ξ_i with index $i \in [|1, n|]$. Each demonstration is modeled by a separate parameter $\theta^{(i)}$ that

represents the tasks solved by the expert. The algorithm presented here focuses on a generative model of mixtures of behaviors or tasks such that the combination of tasks can be represented as a reward function that is a linear combination of the reward functions of the mixed tasks. More precisely, we call *dictionary* a matrix $\mathbf{D} \in \mathbb{R}^{d \times k}$ that represents the dictionary and *coefficients* $\mathbf{H} \in \mathbb{R}^{k \times n}$ a matrix containing mixing coefficients. The columns of $\mathbf{H}$ are denoted $h^{(i)}$ such that the parameters of the ith task are $\theta^{(i)} = \mathbf{D}h^{(i)}$.

The following presents an algorithm to learn the matrix factorization, that is to say, the dictionary matrix $\mathbf{D}$ and associated coefficients $\mathbf{H}$ such that $\theta^{(i)}$s are represented as combinations of k elements from a dictionary. The algorithms minimizes the cumulated cost over all demonstrations denoted by J_Ξ, where $\Xi = (\xi_i)_{i \in [|1,n|]}$, and defined in eq. (4.8). This cost generalizes the average distance from the demonstrator's policy to a nearly optimal policy associated to the inferred representation of the task.

$$J_\Xi(\mathbf{D}, \mathbf{H}) = \sum_{i=1}^{n} J_{\xi_i}(\mathbf{D}h^{(i)}) \tag{4.8}$$

In order to solve the problem $\arg\min_{\mathbf{D},\mathbf{H}} J_\Xi(\mathbf{D}, \mathbf{H})$ the algorithm alternates steps that minimize the cost with respect to $\mathbf{D}$, $\mathbf{H}$ being fixed, and with respect to $\mathbf{H}$, $\mathbf{D}$ being fixed. The second steps actually decomposes in n separate problems similar to the one from previous section. Both steps uses a gradient descent approach where the gradients are given by eqs. (4.9) and (4.10).

$$\nabla_{\mathbf{D}} J_\Xi(\mathbf{D}, \mathbf{H}) = \left(\nabla_\theta J_{\xi_i}\left(\mathbf{D}h^{(1)}\right)\middle| \ \dots \ \middle|\nabla_\theta J_{\xi_i}\left(\mathbf{D}h^{(n)}\right)\right) \cdot \mathbf{H}^T \tag{4.9}$$

$$\nabla_{\mathbf{H}} J_\Xi(\mathbf{D}, \mathbf{H}) = \mathbf{D}^T \cdot \left(\nabla_\theta J_{\xi_i}\left(\mathbf{D}h^{(1)}\right)\middle| \ \dots \ \middle|\nabla_\theta J_{\xi_i}\left(\mathbf{D}h^{(n)}\right)\right) \tag{4.10}$$

In practice the learner performs a fixed amount of gradient descent on each sub-problem (optimization of $\mathbf{H}$ and $\mathbf{D}$), with Armijo step size adaptation before switching to the other sub-problem. The algorithm stops when reaching convergence. It appears that this gradient descent algorithm is quite sensitive to initial conditions. A good empirical initialization of the dictionary is to first learn $\theta^{(i)}$s with the flat approach, perform a PCA on the learnt parameters and use it as an initial guess for the dictionary.[1]

4.3.2 Experiments

In these experiments a *task* refers to a MDP associated to a reward function. We consider *composite tasks* which means tasks that correspond to reward functions obtained by mixing several primitive reward functions.

The algorithm described above is experimented on a simple toy example similar to the one from **?**: a grid world (typically of size 10 by 10) is considered in which actions corresponds to moves in four directions. Actions have the expected result, that is a displacement of one step in the expected direction, in 70% of the cases and results in

[1] Experiments presented further shows that the PCA strategy alone does not provide a good dictionary for our problem, but is an efficient initialization.

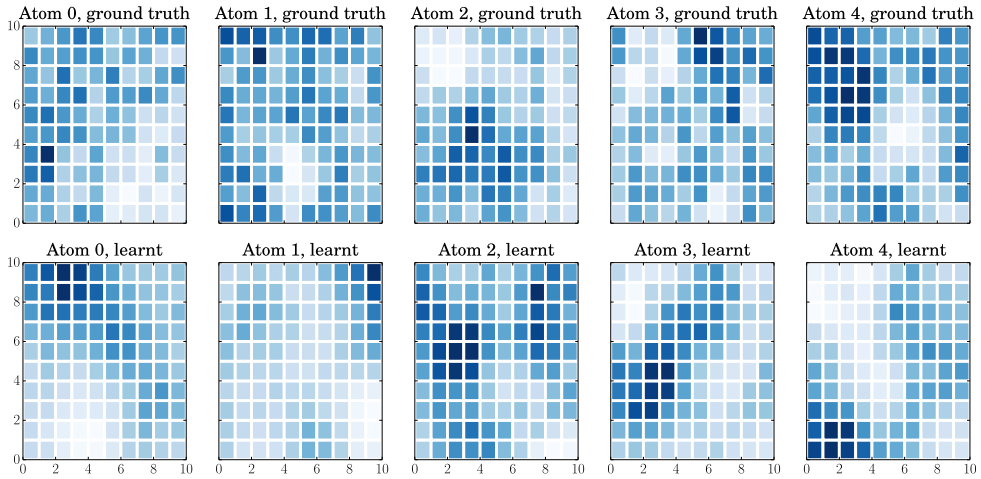


Figure 4.4: The demonstrators generates new tasks by mixing together the five basic reward functions which associated optimal Q-values are represented above. The factorial learner builds its own dictionary by observing the demonstrations; it is represented on the second row. There is no exact match between the two dictionaries however they describe similar reward space as illustrated further in the results. (Best seen in colors)

a random move in the other cases; except when the requested move is not possible from current state (for example going up on top border) in which case the resulting move is drawn uniformly from feasible moves. The following uses a fixed discount factor $\gamma = 0.9$.

Validation

In a first experiment we compare our factorial algorithm to direct learning of the parameter representing a task with Neu and Szepesvari's gradient (GradIRL), that we call *flat learner* to differentiate from the factorial approach.

More precisely a random dictionary of features is chosen, that is unknown from the apprentices, together with mixing coefficients that determine n distinct composite tasks. n experts are then used to generate demonstrations for each tasks (during training the expert may provide several demonstrations of each task). The demonstrations obtained are fed to both flat and factorial apprentices. While the flat learners independently learn a model of each task, the factorial learner reconstructs a dictionary, shared amongst tasks, together with mixing coefficients. fig. 4.4 illustrates the dictionary used by the demonstrator to generate tasks as well as the dictionary reconstructed by the learner.

We evaluate the apprentices on each learnt task by measuring their average performance on the MDP corresponding to the demonstrated task, referred as $MDP(r_{real})$. More precisely the apprentice can provide an optimal policy $\pi^*_{r_{learnt}}$ with respect to its model of the task, that is to say a policy optimal with respect to the learnt reward r_{learnt}.[2] This policy is then evaluated on the MDP corresponding to the real task

[2]This policy is obtained as a greedy policy on the optimal action value function (with respect to

($MDP(r_{real})$). To evaluate the average reward that the apprentice would get on the MDP with random starting positions (not necessarily matching those of the expert) we compute the average value function:

$$\text{score}_{r_{real}}(r_{learnt}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} V_{r_{real}}^{\pi^*_{r_{learnt}}}(s) \tag{4.11}$$

In the results presented here, the demonstrated tasks were generated from a dictionary of 5 primitive reward functions. No feature is used to parametrize rewards: they are represented as deterministic functions from state-action pairs to a real number, which corresponds to a 400 parameters. The expert provides 10 demonstrations for each task, each lasting 10 time steps and 100 tasks are demonstrated.

Results presented in fig. 4.5 show that the factorial apprentice is capable of using information about the common structure of the various tasks to achieve better performance on each task. The performance of the learner therefore increases with the number of demonstrated tasks. When only few demonstrations are given for each task, the demonstrator's behavior is only observed on a subset of the possible state-action pairs. In such cases, the flat learner often fails to achieve good generalization over all the state space. On the other hand, the factorial learner can benefit from other tasks to complete this information.

We also compare the results with flat learners trained with specific features: the ground truth dictionary (*flat, ground truth*) and a dictionary learnt by performing PCA on the parameters learnt by the flat learners (*flat, PCA features*).

Re-use of the dictionary

In order to demonstrate the ability of the factorial algorithm to transfer knowledge to new tasks we performed a second experiment. Apprentices are trained similarly to the previous experiment. In the following we call *train tasks* these tasks. For testing, a new task is generated randomly from the same dictionary of rewards (denoted as *test task*) and apprentices observe a single demonstration of the new task. To get meaningful results, this step is reproduced on a number of independent test tasks (typically 100 in the experiment).

Since the task is different from the previous demonstrations, it is not really meaningful for the flat learners to re-use the previous samples or the previously learnt parameters, so the task must be learnt from scratch. On the other hand, the factorial learner re-uses its dictionary as features to learn mixing coefficients for the new task. We also experimented two alternative, simpler, strategies to build a dictionary in order to re-use information from the training tasks. The first one consists in using a random selection of rewards learnt during training as features for the new tasks (*flat, features from ex.*). We use the learnt parameters of 15 training tasks as features. The other one performs a PCA on the rewards learnt during training and uses the five first components as features (*flat, PCA features*). Similarly to previous experiment the apprentices are evaluated on their score (according to ground truth reward function) on solving the new task.

---

the model of the task, r_{learnt}), computed by value iteration.

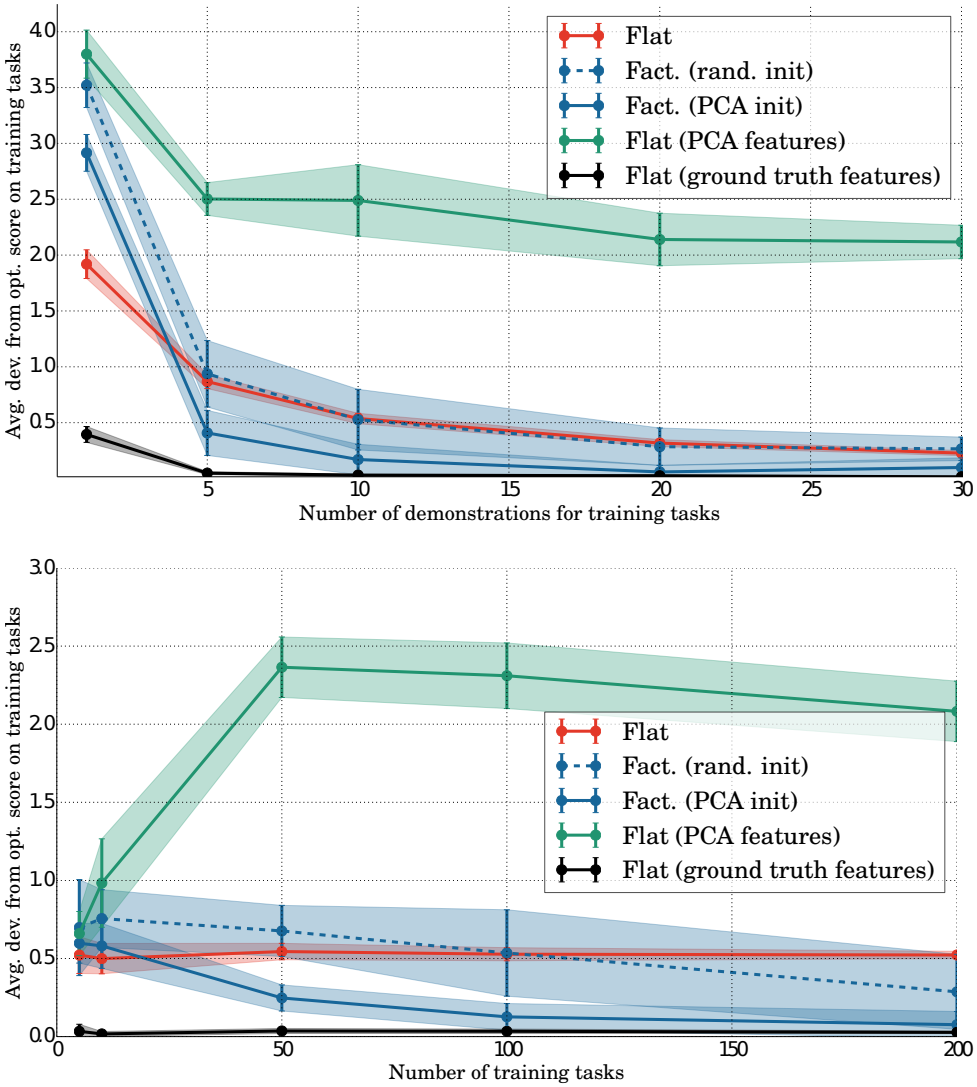


Figure 4.5: **Performance on train tasks** The factorial learner overcomes the flat learner by leveraging the features common to all tasks for high number of demonstrated tasks and moderate number of demonstrations for each task. The curves represent the average deviation (lower is better) from the best possible score (the one obtained with perfect knowledge of the task), that is the average of the optimal value function, for different values of the number of demonstrations per training task (top) for a fixed number of training tasks of 100 and for the number of training tasks (bottom), the number of demonstrations for each tasks being fixed to 10. The algorithm presented in this section is denoted as *Fact. learner*; two alternative initialization heuristics are presented. (Best seen in colors)

Results, presented in fig. 4.6 are compared for various number of training tasks and demonstration per task. They demonstrate that the factorial learner can re-use its knowledge about the combinatorial structure of the task to learn the new task more quickly. The factorial learner also outperforms the other simple feature construction strategies.

The better ability of the factorial apprentice to generalize over the state space is increased in this setting since only a single demonstration is observed from the expert. Often this demonstration only covers a small part of the state-action space. This phenomenon is illustrated in fig. 4.7 that represents the true optimal value function together with the expert's demonstrations, and the learnt value functions by both the flat learner and the factorial one. A typical situation that can be observed in some examples, is that the flat learner's value function is local to expert's demonstration, while the factorial learner, that estimates the task in the space of learnt features, can have a good estimate of the value function in parts of the space where no demonstration was provided.

4.3.3 Discussion

In this section we presented a gradient descent algorithm to learn a dictionary of features to represent multiple tasks observed through an expert's demonstrations with an inverse reinforcement learning approach. The experiments demonstrate that the approach enables the learning of the common structure of the tasks by using transversal information from all the demonstrated tasks. Furthermore it demonstrates and illustrates the fact that this approach enables more accurate representation of new tasks from only one short demonstration, where the classical inverse reinforcement learning approach fails to generalize to unobserved parts of the space due to the lack of adequate features.

The algorithm is compared with naive approaches trying to learn a dictionary from task parameters that were inferred through *flat* inverse reinforcement learning and showed that these approaches fail to learn the relevant structure of the demonstrated tasks. A possible interpretation of this difference is that the PCA approach performs the matrix factorization with respect to the metric of the parameter space, whereas our algorithm uses the more relevant objective cost function. Due to the particular structure of the inverse reinforcement learning problem, namely invariance of the problem with respect to reward scaling, and other transformations (**??**), the metric of the parameter space is not relevant for the objective of apprenticeship learning.

An important limitation of inverse reinforcement learning is that it assumes the knowledge of a model of the dynamics of the environment. Therefore it can either be applied to situations where that model is actually known, meaning it is very simple, or where it can be learnt. However the latter brings the new question of the robustness of inverse reinforcement algorithms to errors or gaps in the learnt model. Furthermore, while regular inverse reinforcement learning outputs both a model of the task and a policy that solves it, the factorial approach presented in this section only provides policies for the observed tasks. This means that although a large variety of tasks may be represented by combining primitive tasks from the learnt dictionary, it is generally not meaningful to combine the policies in the same way: the agent has to train a policy for these new tasks.

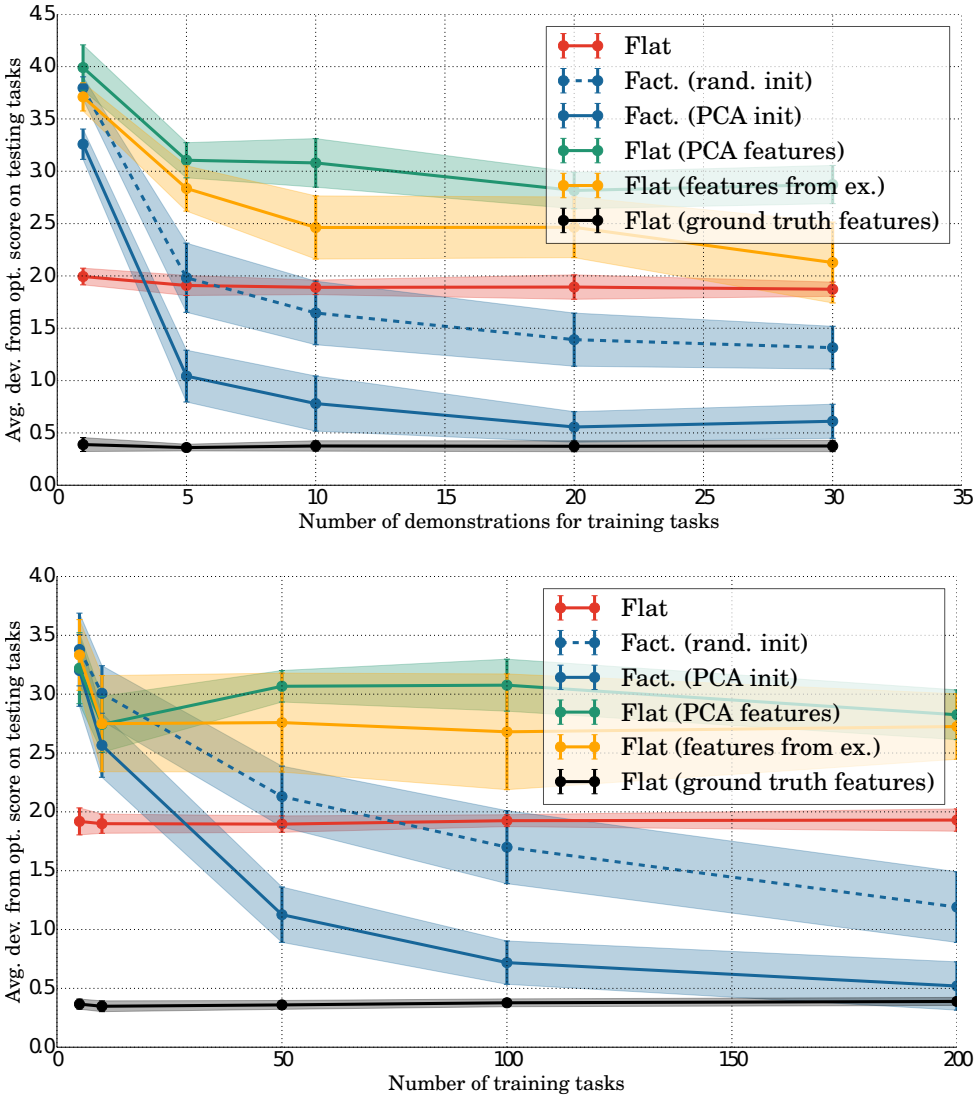


Figure 4.6: **Performance on test tasks** For new task observed through a single demonstration, the factorial learner outperforms the flat learner by re-using previous knowledge on task features. The curves represent the average deviation (lower is better) from the best possible score, for different values of the number of demonstrations per training task (left) for a fixed number of training tasks of 100, and for the number of training tasks (right), the number of demonstrations for each tasks being fixed to 10. (Best seen in colors)

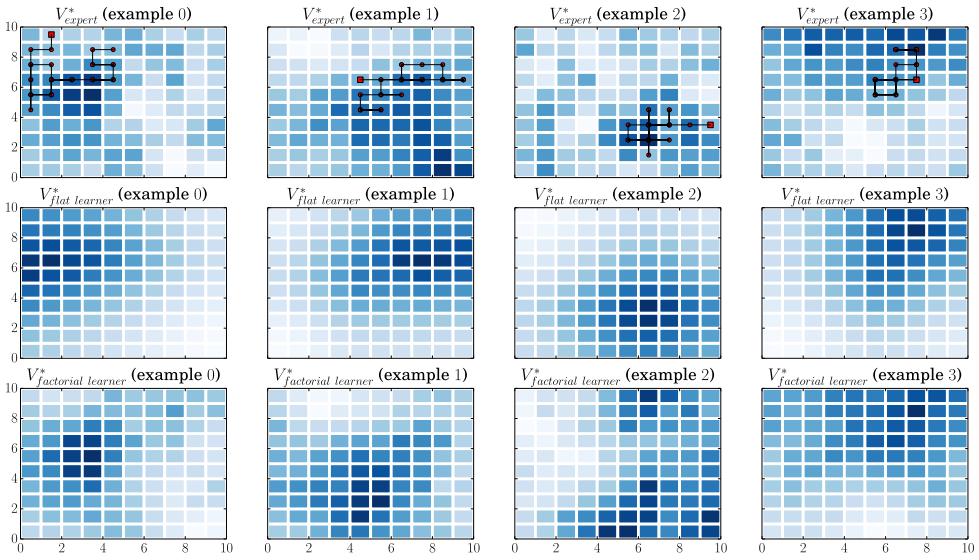


Figure 4.7: The factorial learner achieves a better recognition of new tasks from a single demonstration by using the learnt features. In contrast the flat learner often build a task representation that is local to the demonstration. First row represents the optimal value function (blue is high) for the real task, together with the single demonstration provided by the expert. Second and third row represents the optimal value function for the model of the task as learnt by respectively the flat learner and the factorial learner. Each column corresponds to one of the four first test tasks (from a total of 100). (Best seen in colors)

This algorithm can be considered as a first example of feature learning in the multi-task setup for inverse reinforcement learning. However other approaches should be explored by further work in order to derive more efficient algorithms, by for example extending the natural gradient approach from **?** to the dictionary learning setup, or adopting a Bayesian approach extending **?**.

Finally constraints can be applied to the learnt dictionary to favor some kinds of solutions. Two examples of such constraints for which many machine learning algorithms have been developed are non-negativity and sparsity. Non-negativity of the coefficients would for example focus on representations that allow primitive behaviors to be added to, but not subtracted from an activity in which they do not appear. Such constraints have been successful in many fields to yield decompositions with good properties, in terms of interpretability but also sparsity (see for example **??????**, but also chapters 3, 5 and 6). Sparse coding also focuses on a constraint on decompositions to improve the properties of the learnt elements (**???**). For example. **?** have shown how enforcing sparsity of a task representation can make this task focus only on a few salient features, thus performing task space inference. Other examples are given by **?** and **?**. Exploring the use of these constraints together with the techniques presented in this chapter constitutes important direction for further work.

Chapter 5

Learning a dictionary of primitive sounds

This chapter studies the question of the acquisition of language mainly from the acoustic point of view. More precisely we review and develop techniques that can model the acquisition by a learning system of basic acoustic components of language, like phonemes or words. In this chapter we present approaches that use multimodality or supervision to acquire such components as well as techniques that rely only on the patterns within the acoustic signals. Many of the techniques from the first category use a labels or weak supervision to model multimodality and can therefore be described as symmetric to the experiment presented in chapter 3: instead of having a real scene and symbolic labels representing language, the language is real but the scene is represented by symbols. The NMF techniques presented at the end of this chapter and developed by **??** are the one that inspired work from chapter 3. Finally this chapter as well as chapter 3 present the two experiments that are joint in next chapter.

5.1 Models of language acquisition

Learning dictionary of primitive sounds is an attempt to model language acquisition by featuring what **?** calls *cognitive plausibility* and therefore differs from the static approach that take many automatic speech recognition systems. Brent's cognitive plausibility requires speech recognition methods to be *incremental*, *self-organized* and start with no prior knowledge on the environment it has to explore: properties that are observed on infants who learn the language(s) they are exposed to.

Cognitive plausibility comes with the design of learning processes: a central point in this design is the definition of the interaction between the system and an *outside word*. We already have discussed the central problem of language grounding and the importance of multimodality for language learning, but also the evidence from work of **??** that children can discover patterns in speech signal without relating it to other modalities.

Models of language learning both from multimodal perception and solely from acoustic perception both exist in the literature. Considering the learning of linguistic elements without using any social or multimodal cue highlights the importance of internal regulation systems to develop an internal speech model from extraction of patterns from observed language. **?** have performed an experiment in that direction; it shows that it is possible to extract words or phrases from a set of recorded lectures by studying occurrences of speech patterns. However such learning systems develop representations that are not included in a social interaction and exclude any convention: therefore the models of language that emerges from these experiment cannot be used to communicate since they lack the essential grounding property.

? already underlined the importance of treating the input channel as a communication channel when dealing with user requests and refers to it as the *"How may I help you"* problem: the learning system will develop skills relevant to the client request classification. Gorin thus made a choice between an information theoretic approach or a more action oriented one. Other approaches use both pattern extraction from acoustic perception and multimodal or social information. As an example **?** studies language acquisition from speech, visual and behavioral information. He presents a learning system that includes a first step where sub-lexical patterns in the speech channel and object recognition in the visual channel are separately acquired. In other words, that system is used to bootstrap some preliminary representation before any social interaction. The idea of bootstrapping internal representations have also been studied by **?**. In other works, multimodality is simplified as labels, coding for keywords or more generally for topics, as in **??**.

In order to build larger and more realistic systems, it often is necessary to work with a model of memory; indeed incremental learning systems often bring growth in data size, leading to memory usage and computation. The Acorns[1](**???**) project have explored some of this issues by introducing memory levels. The principle is to separate data storage in different levels, where depth in memory increases with the level of organization of the data. This may, to a certain extent, be seen as a compression mechanism as data is highly structured in the long term memory, but it also introduces an attention mechanism, associated to a model of short term memory.

Pirmitive acoustic elements may be considered at the sub-lexical or lexical level. **?** use hidden Markov models (HMM) to achieve sub-lexical classification: a *universal sound recognizer* is learnt at the sub-lexical level, in a supervised manner before being used to recognize spoken language. **?** uses symbolic labels to build a lexicon of sound segments: similarity measures between sound segments are used to select prototypical examples of such segments that are stored in a model of memory, together with their label. The classification of a new lexical entity is then achieved through a nearest neighbour approach. The experiment from **?** uses a similar approach together with a graph clustering method to group similar sound segments as sub-lexical entities. In the experiment presented in next section, we use a hierarchical clustering method to achieve unsupervised sub-lexical classification on the basis of acoustic similarity; in a second step a bag of word representation is built on top of the sub-lexical units that have been discovered and used to recognize lexical elements learnt in a supervised way.

The exact role of word segmentation is a quite controversial issue in speech recognition: whereas sentence segmentation is relatively easy through silence recognition, word

[1] **A**cquisition of **Co**mmunication and **R**ecog**n**ition **S**kills, http://www.acorns-project.org

segmentation is indeed a difficult task even for standard written text (see **??**). We may distinguish between two approaches: the first one consists on building language acquisition on the ability to segment words, whereas the second one does not rely on word segmentation but might lead to the ability to segment word-like elements as a consequence of word recognition. An example to the first approach is the *segmental dynamic time warping* method which uses dynamic programming to find similar sound segments between speech examples, and defines sub-lexical units as those segments. Such methods are used in **???**.

The second approach which ignores segmentation while recognizing speech may appear less intuitive but also shows great results. Non-negative matrix factorization methods have been used in such experiments. **???** present a method that builds an internal word representation from whole utterances with weak supervision. Similarly **?** explain how NMF can be used to learn models of digit names from sentences that are sequences of such digits. As a consequence, those representations may afterward be used to localize candidate words in examples, and thus, achieve segmentation as a consequence. The bag of words approach presented below uses local descriptors based on a completely arbitrary segmentation. These local descriptors enable a statistical analysis of a whole utterance which leads to keyword recognition without any lexical segmentation.

Many methods have been tried to match a lexical representation of spoken utterances to a more structural representation, a process that may be seen as a grammar extraction. Such methods often use a predefined structure, more or less flexible, to which the utterance is mapped, and statistical inferences in a goal oriented manner. For example **??** use multilayer neural networks to map the recognition of some words to an action; in their experiment the semantics of the environment is based on action choices. In Iwahashi's experiment (**?**), this semantics consists of (object, action, position) associations and is analyzed using a graph structure adapted to this semantics. The difficulty for a system of self extracting such semantics representations, without predefined implementation, is to find an origin for those representations. Other input channels such as the vision channel or motor channels, in the case of action oriented goals, are generally part of this process.

5.2 Hierarchical clustering of basic sounds

In our contribution (**?**) we use a clustering algorithm to discover a dictionary of primitive sounds. More precisely, we use a bag-of-words method in a developmental approach of the learning and bootstrapping of speech recognition skills.

We built a two levels language acquisition system: first an unsupervised clustering level discovers multi-scale acoustic representations of speech invariants from unsegmented speech streams with no prior phonetic knowledge. Then, at a higher level, these low-level representations are re-used to learn to predict a semantic tag associated to whole utterances. The system presented in this section has three main features: there is no explicit segmentation into words or phonemes, sub-lexical units are discovered instead of coming from a static phoneme recognizer, and no supervision is used for the discovery of sub-lexical units.

5.2.1 Background and principle

Our approach is based on the *bag-of-words* idea, that originated in text classification applications (**?**) and have been used with great success in image categorization applications as in the work of **?**. The general idea of bag-of-words approaches is to represent the text or the image as an unordered collection of local elements chosen in a dictionary (the words in a text and local visual features in an image), thus ignoring the global structure. Using this representation, a classification algorithm can then be used to predict the associated category. In computer vision applications, this can lead to very compact representations thanks to the quantization of local features, while preserving the stable local information and ignoring more unstable global geometry. In most applications, the dictionary is static and requires an initial training phase. However **?** has developed an incremental approach that is closer to what developmental systems require. We will therefore transpose this method to the speech recognition problem. Yet, for the sake of clarity, we will use the terminology "bag-of-features" instead of "bag-of-words", since the "words" in the bag-of-words approach are not at all related to "linguistic words" in the speech stream and which constitute important speech invariants to be discovered and learnt in our framework.

5.2.2 Presentation of the framework

The language acquisition system uses three distinct layers to transform the sound representation, as described below.

Continuous Acoustic Feature Vectors (CAF) extraction This layer transforms the input audio signal into a set of vectors, each associated with some position information. The goal of this process is to transform the signal into a set of local descriptors. An important requirement on the used representation is that it must come with a measure of similarity on the verctors. This first layer typically uses time windows static sound processing methods (for example MFCC or RASTA-PLP, as detailed in section 5.2.3).

Unsupervised clustering The role of this layer is to transform each CAF vector from the set obtained above, into a discrete acoustic event, that is to say a single number. This transformation is accomplished through a clustering process. More precisely the clustering builds incrementally a representation of this acoustic event, using the similarity measure inherent to the CAF space. This representation both allows retrieval of the acoustic event corresponding to a given CAF vector and the learning of new acoustic events when a CAF vector does not match any known feature.

Higher level semantic analysis The two previous layers may be seen as a preprocessing, which goal is to transform the input audio signal into a bag of discretized acoustic features, more precisely we get a set of couples, each composed of an acoustic event and its position in the stream. This semantic layer introduces a new representation of the audio signal that allows to efficiently set up higher level

statistical treatment, such as keyword recognition (see following experiments) or more complex analysis.

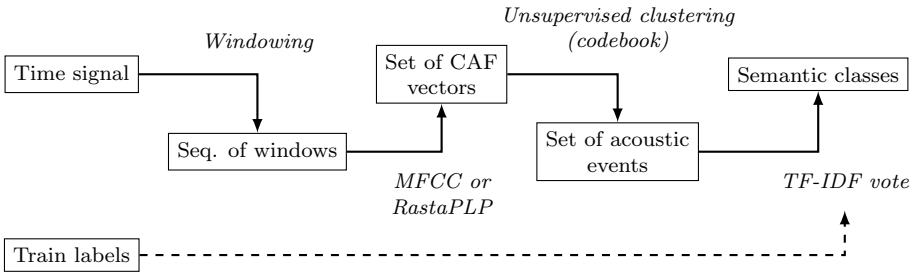


Figure 5.1: Sequence of transformations from raw (time sequence) acoustic signal to the classification into semantic classes.

This process may be described mathematically as follows: given an input audio sequence $a \in \mathcal{A}$, a continuous feature vector space $\mathcal{F}$, a set of localization data, such as time position in the utterance, $\mathcal{P}$, a discrete acoustic feature dictionary $\mathcal{D}$:

- extract CAFs: $a \in \mathcal{A} \longrightarrow (v_i, p_i) \in (\mathcal{F} \times \mathcal{P})^\star$
- find corresponding acoustic events: $(v_i, p_i) \longrightarrow (f_i, p_i) \in (\mathcal{D} \times \mathcal{P})^\star$

Where, i is a free variable, and for any set $\mathcal{E}$, we call $\mathcal{E}^\star = \bigcup_{k\in\mathbb{N}} \mathcal{E}^k$ the set of finite sequences over $\mathcal{E}$. In the case of tag inference, the statistical process is then, given a set $\mathcal{T}$ of tags, a mapping: $(\mathcal{D} \times \mathcal{P})^\star \to \mathcal{T}$.

5.2.3 Implementation

Continuous feature vectors extraction

We use mel-Frequency Cepstral Coefficients (MFCC) and Relative Spectral Transform-Perceptual Linear Prediction (RASTA-PLP) features over a short time window, from **?** implementation. The former feature vectors, which are actually time sequences of successive feature vectors, are compared with respect to a Dynamic Time Warping (DTW) distance (**?**). Such approaches are known to yield efficient acoustic similarity measures for word recognition (see **?**).

Mel-frequency cepstral coefficients These coefficients are computed by first taking the power spectrum of the signal, that is to say the square of the modulus of the Fourier transform of the signal, then averaging it over generally 25 filters, taking the log of the resulting coefficients, and finally applying a cosine transform. The power spectrum of a signal $s(t)$ is given by the following formula.

$$\begin{aligned} p(\omega) &= |\hat{s}(\omega)|^2 \qquad \text{(where } \hat{s} \text{ denotes the Fourier transform of } x\text{)} \\ &= \frac{1}{2\pi} \left| \int_t s(t) e^{-i\omega t} dt \right|^2 \end{aligned}$$

For a set of filters ψ_i over the frequency domain, the mel frequency spectral coefficients (MFSC) are computed as follows.

$$\mathrm{MFSC}(s)_i = \int_\omega |\hat{s}(\omega)|^2 |\psi_i(\omega)|^2 d\omega$$

The filters are chosen according to empirical studies of the human perception of sounds similarities, which is approximated as a transformation of the frequency scale, denoted *mel scale* (**?**); the mel scale is such that frequencies that are perceptually similar are evenly spaced in the mel domain. One then generally use equally spaced triangular filters in the mel domain. Using a finite set of filters accounts for the fact that close frequency cannot be distinguished by humans. An approximation function to the empirical curves from psychology is generally used to convert from the frequency domain to the mel domain. As an example:

$$\mathrm{mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right).$$

The coefficients obtained from this process are meant to model the perception of sound as processed by the human cochlea. The mel frequency cepstral coefficients (MFCC) are computed by applying a discrete cosine transform (DCT, **?**) to the logarithm of the MFSC.

$$\mathrm{MFCC}(s) = \mathrm{DCT}\left[\log\left(\mathrm{MFCC}(S)\right)\right]$$

Dynamic time warping This distance, which inspired from the Levenshtein distance (**?**) distance, takes into account possible insertions and deletions in the feature sequence. It is adapted for sound comparison but does not correspond to an inner product in CAF space, since it is not an Euclidean distance. This prevents the use of the classical k-means algorithm.

A practical benefit of using the DTW distance is that it enables to compare sound feature vectors of different length. However, in our experiments we used fixed length feature vectors: for each acoustic utterance we first compute the MFCC sequence corresponding to this audio stream. After extracting this MFCC sequence, we cut it into fixed length features, using a 80 ms or 150 ms sliding window. The sliding length used in most of the following experiments is one third of the length of the window. However, it is also completely possible to mix several lengths in the same vocabulary or to extract features of random lengths. This would result in more multiscale approach. The window length is here around the scale of a phoneme length and gives a good trade-off between sufficiently long sequences of MFCC vectors and the quadratic complexity in the length of the vectors for DTW computation. Furthermore limiting the window length is necessary in order to obtain local descriptors.

Incremental unsupervised clustering

We use a dictionary structure that groups similar CAF vectors according to their DTW distance into discrete acoustic events. The dictionary implements two operations that are related to its construction and the retrieval of the acoustic event matching a specific CAF. Our approach is adapted from the one used for image processing in **?**.

The dictionary construction is an incremental hierarchical clustering algorithm that is to say new CAF vectors are added incrementally to the dictionary, either in an existing cluster (acoustic event) or by creating a new cluster if the new vector is too far from existing data. The retrieval of the acoustic event that best matches a specific CAF is equivalent to find the closest group in the dictionary for a given vector. Since it is not computationally possible to compare the input vector to each of the clusters we use a tree structure and an associated efficient search algorithm.

More precisely, the acoustic events are defined as hyperspheres in the continuous feature space, and their centers are organised in a tree structure inspired by the one of **?**, where leaves are primitive clusters and nodes represent hierarchical clusters. The tree structure is organised according to the following rules:

1. each leaf or node is a cluster C represented by its centroid: a vector v_C,
2. each leaf (primitive cluster) is actually a hypersphere of radius r_{max} around its centroid. A CAF vector v is therefore part of a primitive cluster C if and only if $d(v, v_C) \leq r_{max}$
3. each node of the tree has a limited number of children N_{max}. The cluster associated to the node is the union of the clusters associated to the children, and the centroids n_C associated to the cluster is the mean of the vectors it contains.

A CAF vector is matched to a cluster by recursively following the child of the node which centroid is the nearest from the searched vector. The dictionary is built by adding these vectors to the tree: we find the nearest leaf (primitive cluster); if the vector matches the radius condition regarding to this cluster, it is added inside this one; if not, a new cluster is created initially containing only this vector. In the case where a new cluster was created, it is added as a child of the same node as the previously found nearest cluster. Then we check if the number of children is below N_{max}; if not, the node is split in k nodes, by a k-means process (see algorithm 2 and **?**, 14.3.6) on the centroids of the leaves. The leaves are then distributed to those child nodes. An example of this mechanism, also described by the following pseudo-code of algorithm 1, is shown in fig. 5.2.

This structure and algorithm implement an approximate nearest neighbour search, and thus the processes of learning a CAF or retrieving the corresponding acoustic event are approximative. Since the CAF vectors are themselves noisy, this approximation is naturally handled by the statistical treatment in layer 3. In order to reduce the impact of orientation errors while exploring the tree, which may result in an important final error, for example, if it occurs near the root of the tree, we added the following improvement to the search algorithm.

The idea is to launch more than one search for each request and then select the best results. This is close to a *branch-and-bound* techniques and may be implemented in many ways. We tried two implementations of this method. In the first one, for each node reached during the search process, the search is launched again on its b children closest to the target, instead of just the closest child. By best children we mean the b children with the lowest distance between their centroid and the requested vector. b is called the backtracking parameter. This method leads to a complexity of $\mathcal{O}\left(n^{\log_k(b)}k\right)$, where n is the number of nodes, k the k-means parameter used to create the tree and b the backtracking parameter. In practice this backtracking

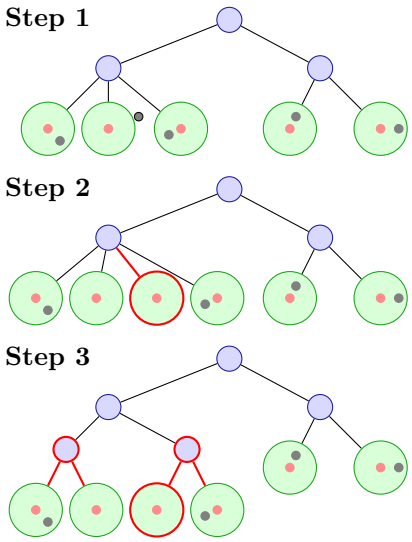


Figure 5.2: Insertion of a new vector in the hierarchical lexicon structure. The nearest leaf is found, but the vector is too far from the center (first step) so a new leaf is created (second step). The new leaf father has now too many children ($N_{max} = 3$) so the node is split in two parts(third step).($k = 2$)

approach make the search very long compared to the $\mathcal{O}\big(k \log(n)\big)$ original complexity.

The second method uses the same idea, but instead of deciding locally which node deserves to be explored, it runs full searches, at the end of which it launches again a search from some node on the tree, where a good candidate path may have been missed. More precisely, during the search, each time a child node is chosen for the proximity of its centroid to the requested vector, its siblings are memorized with some value representing how far they were from the chosen child. When a candidate leaf is finally found, the system is able to reconsider the choices it has made during the search and explore the ignored node which are the closest to the target.

By repeating this process b times, and finally choosing the best candidate nearest neighbor from those found, we are able to minimize the impact of the approximate nature of our structure. The actual complexity of this method is roughly $\mathcal{O}(bk \log(n))$. The second method gave a better trade-off between the number of explored nodes, which corresponds to computation complexity, and the quality of the retrieved approximate nearest neighbor.

Semantic tag inference

While previous steps are able to build an internal representation for the system, based on topological information, this process had no relation to the final goal of keyword classification. Actually, all the semantics related to the classification task is created in the step. We implemented a voting mechanism to score acoustic events and examples regarding semantic tags.

The idea of the voting mechanism is to associate a weight w_i to each acoustic event

Algorithm 1 Adding a vector to the cluster tree node

```
: current_node is the node where the vector is to be added,
: vector is the vector to add,
: k is the k-means parameter,
: r_max is the threshold distance that is used to decide if two vectors are considered
identical,
: N_max is the maximum number of vectors that a leaf may contain.
procedure ADD_VECTOR_TO_NODE(current_node, vector, k, r_max, N_max)
    if current_node is a leaf then
        let v be the nearest vector in current_node
        if distance (vector, v) ≤ r_max then
            add vector to current_node
        else
            let father be the father of current_node
            create a new leaf from vector and add it to father
            let children be the children of father and n their number
            if n ≥ N_max then
                new_nodes ← K_MEANS(k, children)
                set new_nodes as the children of father
            end if
        end if
    else
        let child be the nearest child from vector in current_node
        ADD_VECTOR_TO_NODE(child, vector, k, r_max, N_max)
    end if
end procedure
```

i. Let f_i^t be the frequency of acoustic event i regarding tag t, $f_i^t = \frac{n_{i,t}}{n_t}$ where $n_{i,t}$ is the number of co-appearances of acoustic event i and tag t and n_t the number of appearances of t. For a query utterance q, where acoustic acoustic event i appears q_i times, i votes as $V_i = q_i \cdot f_i^t \cdot w_i$, where w_i are weights. A common way of setting weights w_i is to use a Time Frequency — Inverse Document Frequency (TF-IDF) approach by setting

$$w_i = \log\left(\frac{N_{tags}}{N_{tags}^{(i)}}\right), \tag{5.1}$$

where N_{tags} is the total number of tags and $N_{tags}^{(i)}$ the one of tags whose examples contain acoustic event i at least once.

Additional conditions may be considered such as setting all node weights to zero except from leaves, which rely entirely on the size of clusters, that is to say the r_{max} parameter, in our case, which is chosen a priori. One may also choose to allow only nodes near the leaves to have a nonzero weight or to rely entirely on TF-IDF weights. This kind of modifications may bring more scalability and robustness to the system. It also defines which clusters are acoustic events: either only leaves or all nodes, and thus the use or not of hierarchical and multi-scale acoustic events.

In order to be able to compute this score we store the number of appearances of each acoustic event in an utterance associated to a particular semantic tag: this corresponds to previously introduced $n_{i,t}$. The following process is used: while training, for a given utterance with tag t, transformed in a bag of acoustic events, for each acoustic event i, $n_{i,t}$ is increased by one.

Algorithm 2 The k-means clustering algorithm

: k the number of clusters
: $(x_i)_{1 \leq i \leq N}$ the Nvectors to add,
: c is the vector of assignments (in $[|1, k|]^N$)
procedure K_MEANS(k, (x_i))
  initialize the cluster assignments c
  while assignments change **do**
    for each cluster, let m_j be its mean
    let c be the assignment of each observation to the closest mean:

$$c_i = \arg \min_{1 \leq j \leq k} \|x_i - m_j\|^2$$

  end while
end procedure

During the test phase, we extract the bag of acoustic events corresponding to the utterance. Then, for each tag we compute its score on the utterance, by summing the votes of each acoustic event from the utterance representation. Votes are computed as explained previously, using only the count of co-occurrences, by simple operations over the $(n_{i,t})_{i,t}$ matrix.

5.2.4 Experimental scenario

As explained above, we adopt a framework where the goal is to allow a robot to progressively learn to predict semantic tag(s) associated to a given speech utterance. For example the robot is incrementally provided with examples of associations between speech utterances and semantic tags, and should accordingly incrementally update its internal representations in order to predict better these semantic tags in new utterances. Semantic tags are technically encoded as keywords referring either to general topic(s) of the utterance, sometimes corresponding to the presence of a particular word in the utterance or to the speaker style or language.

Databases and protocols

We restricted our work on labeled classification problems, that is to say, sets of utterances associated with a semantic label. These labels may be words contained in the utterance as well as levels of speech or speaker identities. The system is trained with such a learning dataset and then evaluated on its label prediction performance.

During our experiments we worked with two datasets. The first one was a home made dataset in which utterances were single words. This dataset, which contains twenty three examples of ten different words, was used to evaluate the performances of the nearest neighbor retrieval with word-long continuous features. The second one is the Caregiver dataset (**?**) provided by the ACORNS project, composed of 1000 utterances containing 13 keywords, each spoken by 4 speakers in English adult directed speech; this makes a total of 4000 utterances. An example of sentences used in the dataset is *Angus is lazy today.* where the semantic tag/keyword is *Angus*. Examples of transcriptions from utterances from the dataset are given in table 5.1. More details on the dataset can be found in appendix B.1.

We take a **bath**
To put it in the **bath** isn't funny either
The **shoe** is a symbol
Now **mummy** is losing her patience
Daddy comes closer
Angus takes off her shoe
Daddy never calls
She sits on a **nappy**
Now everybody is in the **car**
Where is the **nappy**

Table 5.1: Transcriptions from ten random examples from the Acorns Caregiver dataset from **?**. Keywords are identified in bold font.

Results

In order to demonstrate the cognitive efficiency of our system we set up the following experiment: for each speaker we randomly split the database in two sets: a train set consisting of 900 examples and a separate test set of 100 examples. The system is trained incrementally with each utterance of the training set; after each 100 train examples, the system is tested on the whole test set. This protocol, which allows us to monitor its progress, is represented in fig. 5.3. In order to characterize the efficiency of the learning process as its improvement through training, that is to say the convergence speed of the algorithm, we regularly test the process during the training and visualize its performance at each step.

The same experiment can be made with the 4000 examples coming from all four speakers, to demonstrate that the method is, in some way, robust to multi-speakers learning. In this experiment, the training sessions are 200 examples long and after each training session the process is tested with a constant set of 400 examples: 100 from each speaker. The training set is a succession of 900 examples from each speaker, presented by order of speakers. Such results are presented in fig. 5.4.

These experiments demonstrate the good accuracy of our system on the keyword recognition problem. We may compare these results with those from **?**, whose database we used. Actually our results are quite similar to the ones they obtained using non-negative matrix factorization, which method is also not centered on segmentation and proved to reach maximal performances among a variety of various competing technical approaches. However matrix factorization has other interesting properties, as discussed in chapters 3, 4 and 6.

Those results demonstrate, first of all, the ability of our system to build an internal representation of speech units, in an unsupervised manner (information about keywords is not used in the building of the dictionary), and then to use this internal representation to achieve a keyword recognition task, performed by a kind of semantic engine, which in our experiments is the score system.

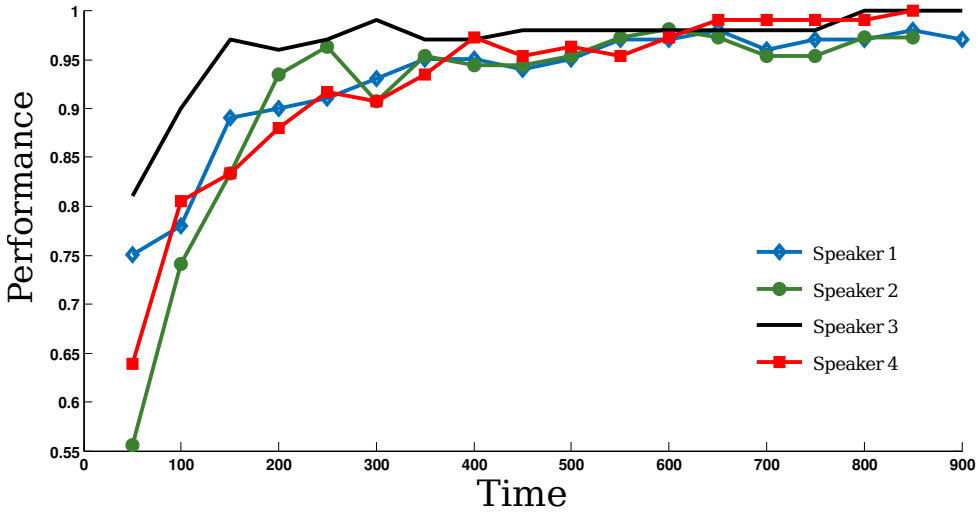


Figure 5.3: Success rate on the utterance classification task with data from a single speaker: each utterance has to be classified as containing one of the ten keywords. Results are plotted at various stages during learning (number of training examples for incremental learning). A separate learner is trained on each speaker; 1000 examples for each speaker are split into 900 for training and 100 for testing. (80 ms MFCC features)

5.3 HAC representation

This section describes the representation of sound used in the works of **???** and that is also used in chapter 6. Histograms of acoustic co-occurrences (HAC) were introduced as a representation of sound that is based on acoustic events. It is similar to the bag-of-words approach from previous section since it discards most of the sequential information of the acoustic events; it however consider co-occurrences of pairs of acoustic events and uses a static (instead of incremental) approach to codebook construction. Two important properties that make HAC representations particularly adapted for the NMF algorithm are that they involve nonnegative values and approximate the sequencing of acoustic patterns in an additive manner.

The outline of the transformation from raw sound to HAC representation is given in fig. 5.5. The steps are explained in more details in the remaining of this section. We start from the representation as sequences of MFCC vectors, which computation is detailed in section 5.2.3. Additionally we consider dynamic information on top of the sequence of MFCC vectors $(\text{MFCC})_t$: time differences are computed with the time difference Δ operator defined following **?** as

$$\Delta\, x_t = 2x_{t+2} + x_{t+1} - x_{t-1} - 2x_{t-2}$$

with the convention $x_i = x_1$ for $i \leq 0$ and $x_i = x_T$ for $i > T$. This transformation is analogous to the delayed velocities used in chapter 3 to represent motions. Similarly the MFCC vectors are extended with additional dimensions from the Δ MFCC and $\Delta\,\Delta$ MFCC. Following **?** we keep 22 dimensions from the MFCC vectors which yields

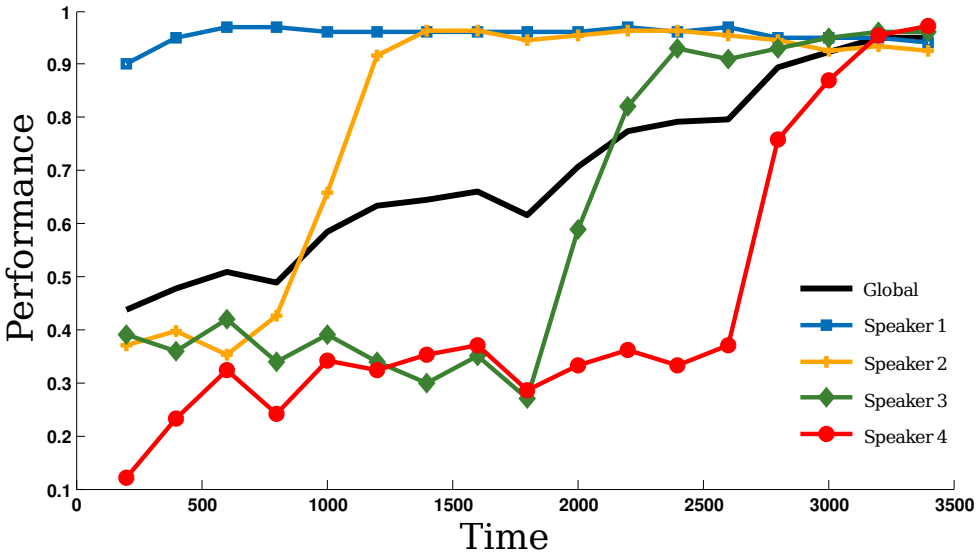


Figure 5.4: Behavior of the learner when facing data from several speakers. The plot displays the success rate on the utterance classification task: each utterance has to be classified as containing one of the ten keywords. Results are plotted at various stages during learning (number of training examples for incremental learning). The same learner is trained on a mixed dataset of 4000 examples from four speakers; 3800 examples are used for training and 200 for testing. The training data is ordered such that the learner encounters first all data from the first speaker, then from the second one, etc. One curve represents global accuracy; the other ones represent the accuracy of the learner on the subset of the test data coming from each speaker. (80 ms MFCC features)

final vectors of 66 dimensions.

5.3.1 Codebooks

There are actually three codebooks, for basic MFCC vectors and there Δ and $\Delta\Delta$ transformations. The codebooks are obtained with the k-means algorithm, described in section 5.2.3. In the following we use the implementation of **?** that builds codebooks of size $k = 150$ for MFCC vectors and Δ and of size $k = 100$ for the $\Delta\Delta$ vectors. The codebooks are used to convert the three sequences of MFCC vectors and their Δ and $\Delta\Delta$ transformations into a sequence of acoustic events: each cluster, that is to say each elements of a codebook defines an acoustic event; each time window is thus transformed into three discrete events corresponding to the clusters in which fall the three vectors associated to that time window.

5.3.2 Histograms of co-occurrences

The last step consists in removing most of the temporal information by building histograms of event occurrences and co-occurrences. This process is used on top of the three sequences of acoustic events obtained from vector quantization of MFCC

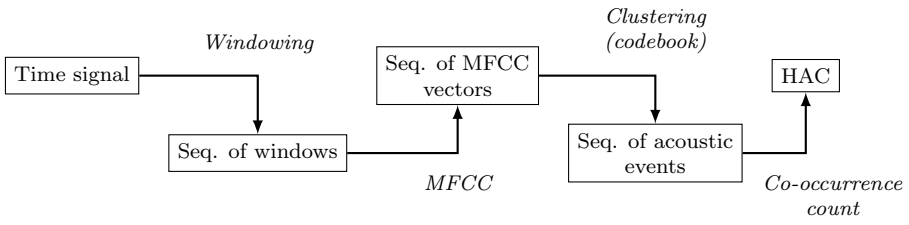


Figure 5.5: Sequence of transformations from raw (time sequence) acoustic signal to histograms of acoustic co-occurrence (HAC) representation.

vectors and their derivatives. It is however not at all restricted to these kind of events: any time indexed sequence of events can be transformed into such histograms. **?** gives a presentation of this construction in the more general case of events not necessarily time indexed but represented in a lattice structure that accounts for the ordering of the events. Finally the process is straightforward to extend to continuous event occurrence probabilities.

For a given stream of event $(e_t)_{1 \leq t \leq T}$ the stream of co-occurences with delay δ is defined as the stream $(c_t)_{1 \leq t \leq T-\delta}$ of pairs $c_t = (e_t, e_{t+\delta})$. Co-occurence histograms are simply histograms of co-occurrences. What is denoted as HAC representation in the following is actually the concatenation of co-occurence histograms for each one of the events categories, that is to say MFCC, Δ MFCC, and $\Delta\Delta$ MFCC events. Two such vectors are then concatenated for two values of δ: 2 and 5.

An important consequence if that representation is a property denoted as the *additive property*. The additive property directly comes from the use of histograms in the representation and states that if two words which HAC representation are w_1 and w_2 are concatenated into an utterance, which HAC representation is denoted as s,

$$s \simeq \lambda w_1 + (1 - \lambda) w_2$$

where $0 < \lambda < 1$. The approximation ignores the events coming from the border between the words. This important property transforms the sequencing operation into a convex combination. It therefore transforms a sentence into a mixture of its words, and similarly a word into a mixture of phonemes.

5.4 Learning words with NMF

In this section we introduce the use of nonnegative matrix factorization to learn words. **??** have performed experiments demonstrating the effectiveness of NMF techniques for word learning in both an unsupervised and a weakly supervised setup.

In their experiment, **?** demonstrate that an unsupervised use of NMF can lead to the learning of words. They use a variation of the HAC representation, described above, on top of a static phoneme recognizer. More precisely they consider 43 distinct phonemes and count the co-occurrences of pairs of phonemes. In their experiment they consider a data matrix V composed of co-occurence vectors built from phoneme recognition on spoken utterances. The utterances are formed as the concatenation of names of digits, such as “014”. Each column of the matrix corresponds to the

representation of one such utterance. **?** apply the NMF algorithm to learn matrices W and H such that $V \simeq W \cdot H$. The algorithm is actually parametrized to learn 11 atoms, that are the columns of W. In their article they demonstrate how these atoms are good representation for the phonemes transitions that are characteristic of the name of each digit. This is done by interpreting the atoms as probabilities on the phoneme transitions and noticing that the phoneme transitions that actually occurs for example in the word "one", that is to say "AH N", "W AH", "_ W", and "N _" have the highest probabilities in one of the atoms.

The experiments from **?** have directly inspired the setup presented in chapter 3 and are therefore very similar. These experiments use NMF in a weakly supervised setup: a data matrix V_{sound} is built using the HAC representation from previous section from spoken utterances. The utterances are simple sentences in containing one or several keywords. Each sentence is associated to a set of labels corresponding to the keywords. These labels can be interpreted as a symbolic representation of objects in a visual scene or basic gestures in a complex motion. A matrix V_{labels} contains binary vectors indicating the presence of each label, as detailed in section 3.3.1. Similarly to chapter 3 sound and label data are concatenated for the training into a matrix

$$V = \begin{pmatrix} V_{sound} \\ V_{labels} \end{pmatrix}.$$

The system is then evaluated on its ability to reconstruct either one or several labels from new utterances.

5.5 Conclusion

In this chapter we introduce several techniques to represent sound and learn sub-lexical or lexical patterns such as phonemes and words. The first technique from **?** accounts for the incremental learning of a phoneme codebook. A bag of acoustic events representation of sound is built from the codebook; the efficiency of that representation is illustrated in a supervised classification task using a simple voting algorithm.

The HAC representation from **?** shares a lot of similarities with the previous techniques: it is based on a static codebook learnt through vector quantization of an audio stream; in addition to the representation as a bag of acoustic events, HAC histograms use co-occurences of events, which capture more temporal information. The bags of paired events are then represented as histograms which provides a representation well fitted to use nonnegative matrix factorization.

The experiments presented illustrate the efficiency of these representations in unsupervised, weakly supervised and supervised problems. In next chapter we extend these use case to multimodal learning as a specific form of unsupervised learning.

The originality of the techniques presented in this chapter is that they discard most of the temporal information of sound. Interestingly they demonstrate that local information is sufficient to achieve simple recognition tasks, a result closely related with similar phenomenon in the field of vision. Therefore these techniques provide an interesting alternative to other techniques based on the process of high level segmentation of the sound signal. Also, it is important to notice that sliding windows

can be used for example to locate the recognized words, using only the aforementioned techniques. However temporal information is crucial for some applications and an important limitation of the techniques presented above is that they do not represent it. It is therefore an important direction for future research to extend such techniques in order to account for the important temporal nature of speech signal.

Chapter 6

Multimodal learning

In this chapter we bring together the work on the discovery of motion primitives presented in chapter 3 and the one on the discovery of sound patterns from acoustic language introduced in chapter 5: we present a multimodal experiment in which learning occurs simultaneously from acoustic language and motions, without any symbolic supervision. That experiment explores aspects of multimodal learning, but also of the issue of language grounding. Although we have identified important issues related to the ambiguity of decomposition in both the problems from chapter 3 and from chapter 5, this chapter demonstrate that the ambiguity that makes each of the problem difficult when taken separately might be much easier to solve when both problems are considered simultaneously.

6.1 Multimodality in perception and learning

Most artificial perceptual systems, as well as humans or other animals, include sensors from various modalities and can therefore take advantage of these multiple modalities to gather more information on their environment. In some situations, the multimodal nature of the signal is of great importance and is not limited to the juxtaposition of information from each modality.

As already mentioned in section 1.3, an important example of multimodality is given by communication: human communication is not in general reduced to speaking or writing; instead full featured communication makes extensive use of facial expressions, physical contact, and eye gaze. A famous evidence of the multimodal nature of communication was given by **?** and is referred as the *McGurk effect*: observing lips pronouncing 'ga' while hearing 'ba' is most often reported as perceiving the sound 'da' (see also **?**). Because human communication is so naturally multimodal, robots or intelligence systems pursuing human assistance or collaboration with humans might greatly benefit from taking into account several modalities. For example, while degraded communication is always possible, as when using a telephone, that only transmit the acoustic modality, it is not as efficient and natural as direct communication. Only taking one modality of communication into account might also make it more difficult to learn.

The question of *language grounding*, as introduced by **?** and discussed by **?**, points out that learning language is not only about learning the signs of communication such as words, but also requires to relate them to their semantic content. Since that semantic content often lies in other modalities, this problem can be seen as another important instance of multimodal learning.

However multimodal learning is not restricted to language learning. The emergence of the concept of 'dog' is not only related to the ability to recognize pictures of dogs but also to the sound of a dog barking and the touch of a dog's fur. Indeed, many concepts cannot be completely characterized without grounding them on several modalities: the concept 'metallic' cannot be characterized without taking into account its perceptual expression on several modalities (for example visual aspect, sound, touch, or taste), together with the recognition of the spoken or written word.

Unlike supervised learning, unsupervised learning, or reinforcement learning, multimodal learning is not a specific class of algorithm. Indeed, multimodal data can be treated as unimodal data on which an unsupervised learning is applied (some examples provided in this chapter fall under this category). It can also bee considered a supervised regression problem that consist in predicting the signal in one modality, knowing the others. Thus, we prefer a presentation of multimodal learning as a focus on several questions or problems. This chapter focuses more precisely on the study of the mechanisms underlying the self-organization of multimodal perception that can explain the emergence of concepts. The notion of concept does not necessarily refers to an explicit representation of that concept but rather on the emergence of behaviors that are interpreted as mastering of that concept. For example a child is said to master the concept 'dog' not by looking into his brain for a neuron spiking each time a dog is seen but rather by its ability to relate the sight of a dog with the sound of a barking dog.

The acquisition of semantic concepts from self-organization of multimodal perception however raises the question of the drives and cues that enable that organization. In the case of language learning, experiments on children performed by **?**, and **?** demonstrate that *cross-situational learning*, which focuses on elements that are persistent in the environment across different uses of a word, might be used by children to learn the meaning of words. Most of the approaches presented in this chapter rely on cross-situational learning to explain or model the acquisition of lexicons. However mechanism such as *the whole object assumption*, *mutual exclusivity* (see **?**), and *conceptual reasoning* (**?**) are also known to play a role in the process of associating linguistic labels to concepts.

Another important aspect of multimodal learning is related to ambiguities and their resolution. As pointed out by **?**: "The challenge which cross-situational learning needs to solve is not only one of mapping a word to a meaning, but of distinguishing that meaning from possible distractors." Indeed, Quine's *indeterminacy of reference* (**?**) states that relating words to meanings when learning a foreign language is intrinsically ambiguous. On the other hand, many models of learning semantic components from one modality also encounter similar ambiguities issues. An example is given by the experiments described both in chapter 3 and section 4.2; another one is encountered with the choice between *thematic* and *taxonomic* association of concepts as explained in **?**[1]. Others analogy can be drawn between this phenomenon and the ambiguity of

[1]Thematic association refers to the association of concepts that are related because they interact together, as *milk* and *cow*. Taxonomic association refers to concepts that belongs to the same class,

word segmentation (see **?**), but also with multistability phenomenon as described by **???**, and the *cocktail party effect* (see **?**).

It comes that ambiguity and the means to overcome it are central aspects of multimodal learning. In a somehow paradoxical manner, many multimodal problems feature ambiguity in one or several modalities, but, as pointed out in chapter 3 and section 4.2, integrating information from several modalities can be efficiently used to overcome such ambiguity. In other words, considering the problem of concept learning separately in each modality suffers from the presence of ambiguity, but looking at the same problem in several modalities at the same time might help resolving that ambiguity instead of increasing it. For example the role of multimodal perceptions relatively to multistability is discussed by **?**. Similarly **?** explores the role of vision of the lips for improving intelligibility of spoken sound. Finally **?** present an algorithm for source separation taking advantage of audio-visual information. In the frame of learning language this emphasises the mutual interaction between the learning of the language itself and the concepts it describes. On the one hand perceptual knowledge is used to identify linguistic signs and structures that are by themselves ambiguous. On the other hand language also plays an essential role in shaping the concepts it describes.

6.2 Related work

Before introducing some works related to the one described later in that chapter, it is important to notice that the boundaries of what may be considered a multimodal learning problem are difficult to draw. As an example, any classification or regression algorithm can be seen as solving a multimodal learning problem, where one modality plays a special role, either by being constrained to a specific strong structure (labels in classification), or by having to be reconstructed. That perspective is taken in chapter 3 and section 4.2 with respect to multi-label classification. Assuming such structure in the data is however often not compatible with th problem of concept emergence from sensori-motor perception; our contribution presented in this chapter thus focuses on the use of unsupervised algorithms.

In their seminal work, **??** introduce a learning architecture called *Cross-channel early lexical learning* (CELL), together with an example implementation, that demonstrates how the problems of learning *linguistic units*, *semantic categories*, and their relations (in the form of *lexical units*) can be achieved at the same time. In CELL, both linguistic information and contextual information, each of which may come from several sensory channels, are segmented according to saliency cues such as utterance boundaries or changes in motions. In a second stage, implemented by a model of short term memory, pairs of recurrent co-occurring linguistic and contextual events are filtered. Finally models of linguistic units and semantic categories are built; they combine clustering of similar language stimuli as well as contextual stimuli and optimize the mutual information between language and context. The pairs of linguistic units and semantic categories with the highest mutual information are kept as lexical units.

? have presented work addressing a similar problem but focusing more precisely on user-centric and multimodal information. They present a learning architecture

such as *cow* and *pig*

that is capable of forming semantic models of both actions and observed objects by using unsupervised learning techniques. First, models of actions are formed by fitting a mixture of hidden Markov models on the observations and models of objects result from an agglomerative clustering algorithm. The models of objects and actions define concepts and together form the contextual information. Then, this contextual information is used to extract word-like units related to these concepts from phoneme transcriptions of the recorded utterances. More precisely longest phonetic sequences are extracted from all utterances related to the same object or action. Then an alignment techniques, that comes from the field of automatic translation, is used to form the lexical units composed of words and concepts.

In **?** the studied language is related to an (object, action, position) semantics which appears to be closely related to the language grammar. More precisely a lexicon is built from data: the lexicon actually represents a mixture of word and meaning pairs, where meanings can either be objects or actions. Specific probability models are implemented to represent the acoustic modality as well as the modality of visual objects and the one of visual actions. The number of elements in the lexicon is automatically chosen in order to maximize the mutual information between the speech and contextual modalities. In their model a representation of the grammar of the language is learnt by identifying in which order the linguistic elements corresponding to the eventual object, action, and landmark appear. In another experiment **?** details how a similar architecture can benefit from the possibility of asking a user for binary feedback when unsure of the novelty of an encountered lexicon pair.

? introduce a recurrent neural network architecture that learns to relate a basic language to corresponding behaviors of a robot. The system is capable of both understanding the words composing the language, that in their experiment are represented by symbols, and their composition, that is to say the syntactic structure of the language. Another aspect of learning action related to language is explored by **?** who provide a model of multimodal learning for symbolic language and real actions. Their experiment demonstrates that learning a compositional structure shared between action and language can allow robotic agents to achieve better generalization of the acquired motor knowledge. More precisely the linguistic input received by the system shapes a model of the structure of actions and makes the system capable of achieving behaviours that were not encountered in training. Furthermore **?** have demonstrated that providing linguistic instructions can facilitate the acquisition of a behavioral skill, in comparison to pure motor learning. Although these experiments are limited to symbolic language, they are good illustrations of the implication of learning multimodal actions and grammars.

Our experiment (**?**), presented in section 3.3, can be seen as a multimodal learning experiment where the language modality is actually symbolic. **??** have presented a similar experiment where the contextual modality is the one that is symbolic and the linguistic one is continuous. Similarly **???** use the NMF or probabilistic latent semantic association (PLSA) algorithm to learn from a continuous and a symbolic modality. **?** have also used the NMF algorithm to learn from two continuous modalities. However their evaluation is based on the reconstruction of a third, symbolic, modality. An interesting aspect of all these approaches is that they use common feature learning algorithms, that are some kind of unsupervised algorithms, instead of relying on explicit models of the lexical units and their relations to language and context.

Another example of the use of feature learning techniques is given by **?** who also present an experiment based on a similar multimodal setup. They introduce an architecture based on sparse restricted Boltzmann machines that learns from two continuous modalities: one is acoustic and the other corresponds to the observation of the speaker's lips. They demonstrate how in certain conditions the algorithm reproduces the McGurk effect. Their algorithm actually learns a new representation of the input in an unsupervised setup and is then evaluated combined with a standard supervised classifier trained on top of this representation. Their work can also be described as a sensor-level multimodal fusion: several modalities are used to build a common representation that is later used to solve a classification problem. Actually multimodal fusion has already been used to improve supervised classification: **?** discuss the use of both sensor-level fusion and decision-level fusion for speech recognition. **?** also implement decision-level fusion and demonstrate that it improves the recognition of objects.

In the following we present and extend a multimodal learning experiment (**?**) based on the use of the NMF algorithm. The setting and the algorithm are closely related to the one of **?**; the experimental setup also shares many similarities with the one from **?**. However in these experiments we do not evaluate the learning through a standard classification task: instead of testing the reconstruction of symbolic labels, the system is tested on a behavior based classification task, as encountered by children. We show that fitting an explicit representation of a lexicon is not necessary to produce behaviors that are considered on children as evidence of the mastering of lexicons understanding. That aspect is an important novelty of our work in comparison of the aforementioned previous work. More precisely we do not build a system with mechanisms for explicit decomposition into concepts and words, which would make the decomposition capability a pre-requisite to the learning of words, concepts, and their relation. The latter approach is described by **?** as targeting *compositional understanding* first, which they oppose to *teleological understanding*[2]. Indeed the system presented below self-organizes until it is capable of solving a simple behavioral classification task; it therefore achieves teleological understanding of sentences without word segmentation and recognition being implemented as a pre-requisites. We however illustrate the fact that the compositional understanding also emerges at the same time. These aspects constitute an important difference between the work presented in this chapter and the ones from **??** and **?**.

Similarly to all the aforementioned approaches, ours use the cross-situational heuristic to discover the semantic concepts: a form of compression is performed on the sensory input that favor the representation of events that occur simultaneously.

6.3 Experimental setup

This chapter presents a system that learns to link elements from one modality of perception to related elements in other modalities. We perform several experiments in order to explore the learner's ability to represent semantic relations between the modalities. These semantic relations may correspond to either an essential relation

[2]As explained in section 1.2, compositional understanding consists in understanding a complex concept as the combination of the simple parts that compose it. On the other hand teleological understanding is the understanding of the concept as a whole, generally with respect to a specific interaction task.

as the one relating the barking to the image of the dog, or conventional relation as the one relating the name 'dog' to images of dogs.

The origin of the essential relation comes from the reality of an object that has manifestations in several modalities. Their exists such a thing as a dog that has manifestations in the visual modality as images of the dog, in the touch modality as the touch of the dog's fur or its claws, or in the acoustic modality as the sound of the dog barking. Although not all of these manifestations occur each time the dog is encountered, they are often perceived simultaneously since they corresponds to the actual presence of the dog.

On the other side, the conventional relation is characteristic of language: it corresponds to the fact that the word 'dog' is often pronounced when a dog is present and is the object of attention. It is extensively used by parents to teach new words to children.

An important element is that both relations are characterized by that cross-situational property, therefore a mechanism leveraging such information would be able to learn both. In the following we denote by *semantic concept* the set of manifestation of such an object, either related essentially or by convention. Additionally a semantic concept may have several manifestations in a single modality. For instance a dog is associated to both the touch of its fur and claws, or to the sound of the dog barking and the word 'dog'. When the essential relation is the same as the convention, the word actually takes the form of onomatopoeia. Importantly this is an example of a mechanism for symbol grounding; more generally the semantic relations we consider actually include Peirce's icon, index, and symbol **?**, 3.1. In the following, the semantic relations are only characterized in the stimuli by the relatively simultaneous occurrence of the related elements in the various modalities, that is, the cross-situational relation.

In this setup we consider the situation in which objects or motions are perceived by an intelligent system while sentences describing the scene are pronounced. Such a setup is illustrated in fig. 6.1.

The modalities presented can vary from one experiment to the other, but a semantic relation exist between some elements of the different modalities. These elements might be of several natures: gestures in motions, object in visual scenes, or words in spoken utterances. We consider semantic relations as mappings between these elements: for example a word is related to a gesture, or a gesture to an object in a scene. An example of such a mapping is given in table 6.1. During training the learning agent observes examples of scenes as observations in several modalities. The scenes are such that in each of them one multimodal concept is present and observed in several manner in the modalities. For example a sentence is heard containing the word 'dog' and a picture of a dog is seen. However not all perceived elements are meaningful, that is to say related to elements in other modalities. For instance many words appear in each utterances that are not semantically related to anything in other modalities. Similarly other objects may appear in the visual scene that are not related to the subject of the sentence. Therefore the association between elements of the several modalities is ambiguous in each example and the system has to use several observations to solve that ambiguity. The learning system is then tested by observing only one modality and having to chose between several examples in another modality the best match. For example the system hears a sentence talking about a dog and has to chose between several pictures the one containing a dog.

Training The learner observes a set of examples of gestures each of which is paired with a spoken descriptions of the gesture.

Testing The learner hears a new spoken utterance and is asked to chose a gesture from a small set of demonstrated gesture that best fit to the description.

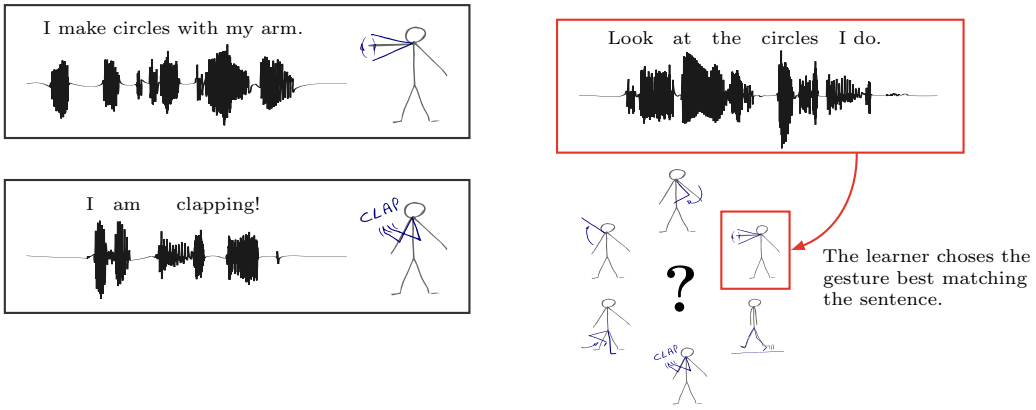


Figure 6.1: Illustration of the cross-modal classification task on which one learner presented in this chapter is tested. The transcriptions of the spoken utterances are represented on the figure to illustrate the keyword semantics. However, the learner does not observe these transcriptions.

Interestingly this experiment is very similar to the one performed by develepmental psychologists to study the role of various heuristics used by children for the acquisition of words, including cross-situational information, as in the works from **??**. Unlike many approaches presented in previous works on multimodal learning, we do not evaluate the performance of the learner on a regular classification task. Instead the learner is evaluated on its ability to relate elements from distinct modalities, in a way similar to the one that would be used on a children. We also evaluate the emergence of words recognition as well as the emergence of a representation of the semantic concepts.

The interactions mechanism between the learning agent and the caregiver that provides the demonstration actually shares many similarities with the one from the talking heads experiment as described by **??** (see also section 1.3). More precisely the agent we present in this chapter plays the role of the 'hearer' from the talking heads, while the caregiver takes the role of the 'speaker'. There are however important differences between our setup and the one from **?**. First, there is no turn in the role taken in our experiment: the learner only plays the hearer and the caregiver only plays the speakers. Importantly this means that the language is taught to the learner by the caregiver, instead of evolving and emerging from their interaction. Also, in our experiments, the naming game, that consists for the hearer in guessing which object the hearer is talking about, is only played during the evaluation stage. During the training the learner passively observes the caregiver teaching and does not receive any other feedback. Finally, the similarity with the talking head agent is mainly behavioral; our agent uses a different architecture and mechanisms as the ones implemented in the talking head, as explained and discussed in more details in

next sections.

6.4 NMF for multimodal learning

This section presents the algorithmic tools that are used in the following multimodal experiments. They are based on the nonnegative matrix factorization algorithm (see section 2.1), that is used in a very similar way than in the experiments presented in sections 3.3 and 5.4.

The first part of this section presents the learning of a multimodal dictionary; it is then explained how the learned dictionary provides a representation of data that is not bound to any modality; in the following this representation is referred to as the learner's *internal representation* of data. Finally we explain how the learner can transform data from one or several modalities to an internal representation or to an expected representation in unobserved modalities.

The following assumes that each input from the modalities is available as a set of samples, each of which is represented by a nonnegative vector. We consider a setting in which the learner observes samples in several modalities. For example, the system visually observes objects while hearing a spoken description of the scene. We represent the perception of the samples in each modality by a vector v_a, where a denotes the modality (for example the system observes the objects as v_{image} and the sound description as v_{sound}). Details about such representations for the modalities used in the experiments are given in section 6.5.

6.4.1 Learning a dictionary of multimodal components

We call *components* primitive elements that are mixed together into observations, in the same way that phonemes can be seen as mixed together into a word or a sentence. Compared to the common context of clustering, this notion of component is more general: observations are mixtures of several components at the same time, instead of being just a noisy observation of one centroid.

The learner presented here builds a dictionary of multimodal components according to the following model: it searches k components, each represented by a vector w_j (j from 1 to k), such that each observed example v^i verifies:

$$v^i \simeq \sum_{j=1}^{k} h_i^j w^j \tag{6.1}$$

where h_i^j are coefficients and $\simeq$ denotes a notion of similarity between matrices that is defined below. This is equivalent to clustering when the w_j are the centroids and for each i only one h_i^j is nonzero and equals 1. We consider a more general case where w_j and h_i^j are only constrained to be nonnegative.

In the following, the set of n examples is represented by a matrix v of shape $d \times n$ (each example is a column of V), the set of components by a matrix W of shape $d \times k$, called 'dictionary', and the coefficients by a matrix H of shape $k \times n$. The

previous equation, that models the objective of our learner, can thus be re-written as:

$$V \simeq W \cdot H \tag{6.2}$$

In the following, the generalized Kullback-Leibler divergence (also known as I-divergence) is used to characterize the reconstruction error between V and $W \cdot H$. The I-divergence is denoted as $D_I(A\|B)$, is defined by eq. (2.8). In order to minimize $D_I(V||W \cdot H)$, the algorithm, based on multiplicative updates of W and H, that was originally presented in **?**, is used. A description of the algorithm is given in section 2.1.2.

6.4.2 NMF to learn mappings between modalities

Previous section explains how, for a given set of observations from several modalities that is represented by a matrix V, the NMF algorithm can learn a dictionary W and a coefficient H matrices such that training examples are well approximated by the product $W \cdot H$.

We actually consider the case of data coming from several modalities (three in the example). More precisely we assume the data matrix V is composed of column vectors v such that:

$$v = \begin{pmatrix} v_{mod1} \\ v_{mod2} \\ v_{mod3} \end{pmatrix} \text{ and thus } V = \begin{pmatrix} V_{mod1} \\ V_{mod2} \\ V_{mod3} \end{pmatrix}.$$

The minimization of the I divergence induces a trade-off between error in one modality relatively to others. In order for the error in each modality to be treated on a fair level by the algorithm it is important that the average values in the representations are of similar magnitude. It can be easily obtained by normalizing data in each modality. In the following experiment data in from each modality is normalized according to its average L_1 norm.

Since the observations, that is to say the columns of V are composed of several modalities, the dictionary W can also be split into several parts each corresponding to one modality. That is to say each components can be seen as the concatenation of several parts: one for each modality. For example if the data is composed of three modalities: $mod1$, $mod2$, and $mod3$, there exist matrices W_{mod1}, W_{mod2}, and W_{mod3} such that:

$$W = \begin{pmatrix} W_{mod1} \\ W_{mod2} \\ W_{mod3} \end{pmatrix}.$$

In the following we interpret the columns of the matrix H, as an internal representation of the data by the learner. For example, an internal representation h is induced by an observation in modality one such that $v_{mod1} = W_{mod1} h$ or one in both modality one and modality three by:

$$\begin{pmatrix} v_{mod1} \\ v_{mod3} \end{pmatrix} = \begin{pmatrix} W_{mod1} \\ W_{mod3} \end{pmatrix} h.$$

Also, for a given internal representation h we say that the learner expects the observations given by the previous formulae.

Interestingly, it is possible to use the learned dictionary to compute an internal representation of an example, even if the example is only observed in a subset of the modalities. Given an example observed only in one modality, v_{mod1}, one can search for an h such that v_{mod1} is well approximated as $W_{mod1}h$. More precisely this is equivalent to finding an h solution of:

$$\arg\min_h D_I(v_{mod1}, W_{mod1}h) \tag{6.3}$$

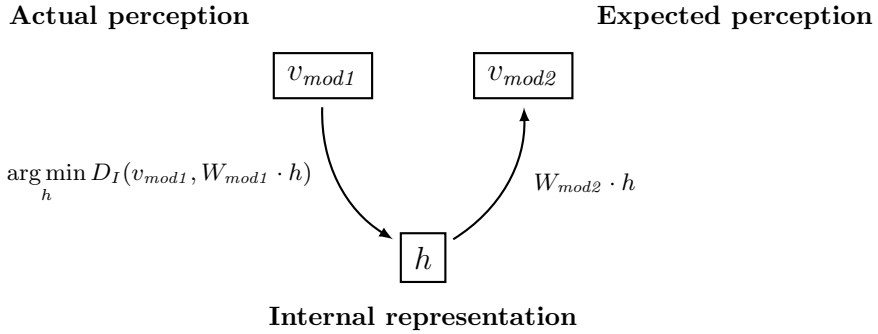


Figure 6.2: Once the system has learnt the dictionary (W_{mod1} and W_{mod2}), given an observation v_{mod1} in one modality it can reconstruct the corresponding internal representation as well as the expected perception in another modality.

The NMF algorithm used in these experiments actually alternates steps minimizing $D_I(V||W \cdot H)$ with respect to W and H. Solving eq. (6.3) is equivalent to the NMF problem with respect to H only; therefore, it can be obtained with the same algorithm, but only using the steps that update H. In theory this approach scales to any number of modalities although the experiments presented here only test it on numbers from two to four.

Finally it is also possible to reconstruct a representation of the data that the system would expect in a modality, given observations in other modalities. For that, from an observation featuring a subset of the modalities, the system fits an internal representation h using the method described previously. Then it can reconstruct the expected representation in an unobserved modality (for example the third modality, *mod3*) by computing the product $W_{mod3}h$. This forms a framework, illustrated in fig. 6.2, that uses a learned multimodal dictionary to transform data from modalities to internal representations or expected data in other modalities. It enables a large set of experiments as illustrated in section 6.6.

6.5 Data and representation

In the following experiments three raw modalities are used: motion, sound, and image. For comparison purposes, a symbolic modality is sometime also used. It is represented in the same way than explained in section 3.3.1. The multimodal data

is obtained by taking examples from three datasets of motions, sounds, and images as explained in next sections. In some of the experiments time windows are built from images, that are actually frames from videos, and the recorded utterances. The process used to obtain these examples is explained in section 6.6.2.

In each experiment an arbitrary random mapping between elements from one modality to the others is chosen; these elements are objects in images, keywords in sentences, and gestures in motions. More precisely the semantic concepts occurring in one modality are associated to the one of the others. For example the keyword 'shoe' from the sound dataset is associated with the gesture 'squat' from motion dataset. The associations are both random and arbitrary, which means they are purely conventional and do not correspond to intrinsic similarities of the corresponding data.

6.5.1 Motions

The motion dataset was recorded from a single human dancer with a Kinect™ device and the OpenNI™ software[3] that enables direct capture of the subject skeleton. The device, accessed using the ROS framework[4], provides an approximate 3D position of a set of skeleton points together with angle values representing the dancer's pose at a given time.

We recorded a motion dataset composed of a thousand examples of ten dance gestures, similar to the one used in the dataset presented in section 3.3.1. The gestures are listed in table 6.1. The gestures are either associated to legs as for example *squat* and *walk* movements, to both arms as *clap hands* and *paddle*, or to left or right arm as *punch* or *wave hand*. Yet this structure is not known by the learner initially. They correspond to both discrete and rhythmic movements. This datased named *Choreo2* is publicly available at `http://flowers.inria.fr/choreo2`.

6.5.2 Sounds

The following experiments use the Acorns Caregiver dataset (**?**), that is described in appendix B.1. The acoustic data is represented according to the HAC representation presented in section 5.3.

In the following experiments, we use recordings from the first speaker from the Caregiver dataset; it includes 1000 utterances containing 10 keywords; in English adult directed speech. An example of sentences used in the dataset is *Angus is lazy today.* where the semantic tag/keyword is *Angus*.

6.5.3 Images and videos

Pictures used in the experiments were acquired as frames from an interaction with an iCub robot, through an RGBD sensor (red, green, and blue camera coupled with a depth sensor). Both the acquisition of the frames and their processing is described in more details by **?**. The processing of the image stream goes through the following steps.

[3] `http://www.openni.org`
[4] Robotic Operating System (`http://ros.org`)

1. *Proto-objects* are segmented using information from motion, depth sensors, and an agglomerative clustering of local descriptors. From there each proto-object is processed independently.
2. Two types of *local features* are extracted: SURF descriptors (**?**) and HSV (hue, saturation, value) of superpixels (obtained by grouping of similar adjacent pixels). Once extracted features of each type are quantized by incrementally learning growing dictionaries of features. This process is very similar to the one presented for sound in section 5.2.
3. Closest SURF points and superpixels are grouped into pairs or triplets of feature vectors denoted as *mid-features*. These mid-features are quantized similarly to the features.
4. At this point, and following the bag-of-word principle (**???**), each view is represented as an histogram of quantized features or mid-features. A dictionary of *object views* and their models are learned incrementally using the TF-IDF score, expressed in eq. (5.1), to track the probability of a feature to appear in a given view.
5. Finally a dictionary of *objects* is built from recognition of views and tracking information.

In the following experiments, one or a combination of several of the representations computed in the aforementioned process are used. More precisely the representation used include (always in the quantized form): SURF features (SURF), SURF couples (SURF mid-couples), HSV superpixels (color), HSV superpixels couples (color mid-couples), and triplets (color mid-triplets).

6.6 Experiments

This section describes several experiments that explore the capacity of the algorithm from section 6.4 to learn semantic concepts and their grounding in several modalities in the setting that was introduced by section 6.3.

6.6.1 Learning semantic associations

In order to investigate the learning of semantic associations between elements of the acoustic, visual, and motion modalities, we use an artificial mapping between acoustic words, visual objects, and gestures. An example of such a mapping is given in table 6.1. Each triplet of word, gesture, object forms a semantic concept. The data used to train the system is composed of sentences, motions, and images; each sentence contains one of the keywords, each motion features one gesture, each image an object. Finally the gesture, the word, and the object from an example belong to the same semantic concept, which implements the cross-situational manifestation of the semantics.

The system is trained on various combinations of either two or three modalities. The modalities might be denoted as *Motion* or *M*, *Sound* or *S*, and *Image* or *I*. After being exposed to a set of training multimodal examples, the system is tested as follows: it observes a new example, called *test example* in a subset of its modalities and has to

Name	Limb(s)	Motion
shoe	both legs	squat
nappy		walk
book	right leg	make a flag/P on right leg
daddy	both arms	clap
mummy		mimic paddling left
Angus	right arm	mimic punching with right arm
bath		right arm horizontal goes from side to front
bottle	left arm	horizontal left arm, forearm goes down to form a square angle
telephone		make waves on left arm
car		say hello with left arm

Table 6.1: List of associations between keywords from the acoustic dataset (names) and gestures from the motion dataset. The limbs on which the motions occur are also mentioned.

chose the best match among several examples observed in other modalities, which are denoted as *reference examples*. An illustration of that process is given by fig. 6.1. For example, the system is trained on sound and image and tested by hearing a sentence (the test example) and having to chose among a set of images (the reference examples) the one that is best described by the heard sentence. Another possibility is to train the system on motions, sounds, and images, and test it on its ability to chose from several sentences the one that best describes a pair of a motion and an image that it observes. We denote such settings by the notation: $M1 \rightarrow M2$, where $M1$ represent the modality or modalities in which the test example is observed, called *test modalities*, and $M2$ the modality or modalities, denoted as *reference modalities*, in which a best matching example must be chosen among a set of reference examples. For example hearing a sentence and choosing the best matching object from images is denoted by $Sound \rightarrow Image$ or $S \rightarrow I$. Viewing an object and a gesture and finding the best matching sentence amongst examples is denoted by $M, I \rightarrow S$. The testing process is illustrated in fig. 6.3. As mentioned above, the testing process is analogous to an instance of the language game from the talking head experiment form **?**.

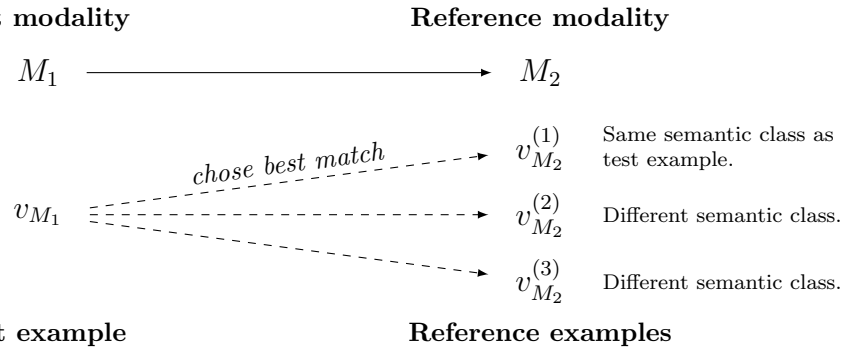
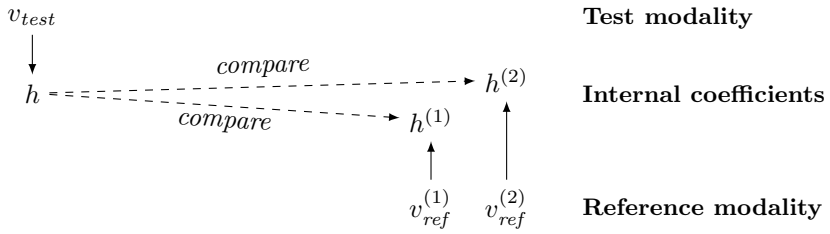


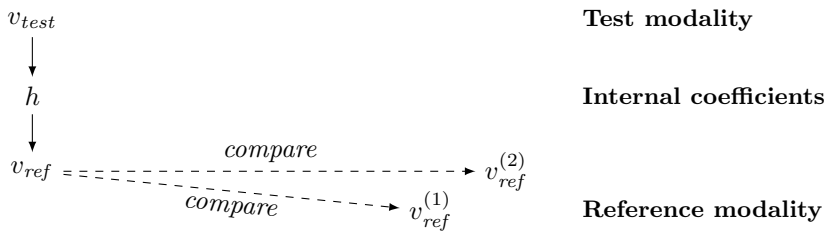
Figure 6.3: The learner is tested on its ability to relate an observation of a test example in one modality to the right reference example in another modality.

Section 6.4.2 explains how to use NMF on multimodal data, to learn a dictionary and the associated *internal representation* and finally how to transform data either from one modality to another, or from a modality to the internal representation (see also fig. 6.2). We use that mechanism as a basis to implement a classification behavior for the learner. For a given example the system uses the learned multimodal dictionary to produce an internal representation of the example (coefficients h) and eventually also an expected transcription of this example in another modality. It then compares an example from the test modalities to those in the reference modalities. To perform the comparison the system can either:

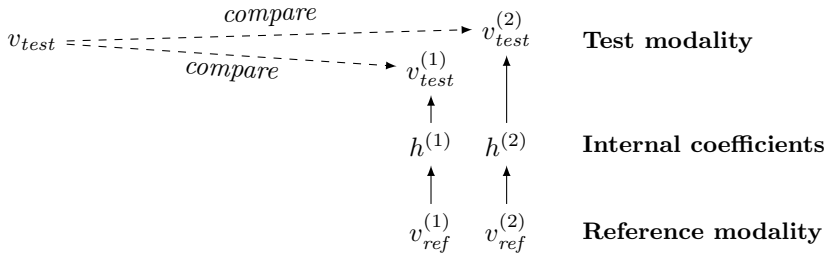
- compute an internal representation of the test example, compute internal representations of the reference examples, and then compare these internal representations.



- compute an internal representation of the test example, use it to generate an expected representation in the reference modality, and compare it to the reference examples.



- compute internal representations of reference examples, for each of them compute an expected representation in the test modality, and compare then the test example.



The choice of one of these methods is referred as *the modality of comparison*. In the following we mainly use the comparison on internal representation. The main interest of proceeding that way is that the comparison is the same, regardless of what the test and reference modalities are.

Choosing the modality of comparison is not sufficient to fully define the system: in order to be able to chose a best matching reference example, the system needs a metric to perform the comparison. Several alternative metric could be chosen to perform the comparison. More importantly, the choice of the metric and its efficiency is highly dependant on the modality of comparison, as shown by the following results. We considered the following common metrics.

- **Euclidean distance**
- **Kullback-Leibler or I-divergence** The Kullback-Leibler and I-divergences are introduced by eq. (2.8). In the following we denote its usage as Kullback-Leibler (KL), although when the data is not normalized we use the I-devergence. By default the divergence from the test example to a reference example is computed; however since it is not symmetrical, we also experimented with the reversed divergence (that is to say the divergence from a reference example to the test example) and a symmetrized divergence obtained as: $D_{sym}(x\|y) = \frac{1}{2}\big[D(x\|y) + D(y\|x)\big]$. None of the three approaches was systematically better in our experimentation.
- **Cosine similarity**[5] The cosine similarity is no a metric but can be used to compare vectors; it ranges between -1 and 1 and the biggest the value is, the most similar the vectors are. It is defined for two vectors x and $y \in \mathbb{R}^d$, as:

$$\text{cosine\_similarity}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

Other choices are possible. In our experiments, many modalities are represented by histograms, or concatenation of histograms, that are of high dimension. In these modalities the Euclidean norm is not necessary meaningful, this is why we use other measures of similarity such as the I-divergence and the cosine similarity.

In the following, the system is evaluated on its recognition success rate. It is defined as the proportion of correct recognition of the system; a recognition is correct when the system choses a reference example matching semantic concept from the test example. In the following the system is always presented with one reference example per class.

In the following experiments we consider 10 semantic classes; when not specified otherwise, we use a default value of $k = 50$ as the number of atoms for the NMF algorithm with 50 iterations, although a number of 10 is generally already close to convergence.

Motion and spoken utterances

Table 6.2 compares the influence of the modality of comparison and metric on the classification success. These result show for example that the sound modality, mainly because of its very high dimension, is not a good choice for the comparison, specially when the comparison is performed with the Euclidean metric. Indeed the choice of the metric to use is highly dependant on the nature of the data in the modality; therefore, using the internal representation is a way to only chose one metrics.

[5] `http://en.wikipedia.org/wiki/Cosine_similarity`

Modality			Success rate		
Test	Reference	Comparison	KL	Euclidean	Cosine
Sound	Motion	Internal	0.608	0.612	0.646
		Motion	0.552	0.379	0.444
		Sound	0.238	0.126	0.208
Motion	Sound	Internal	0.610	0.704	0.830
		Sound	0.106	0.090	0.186
		Motion	0.676	0.642	0.749

Table 6.2: Success rates of recognition of the right reference example from a test example. The values are given for many choices of the reference test and comparison modalities and various measures of similarity. The results are obtained by averaging on a ten fold cross-validation, baseline random is in that case 0.11.

The results from table 6.2 demonstrate that the system is capable of learning aspects of the semantic associations. If the system is trained on a dataset where no semantic semantic association exists between the two modalities (such a dataset can be obtained by choosing a random motion and a random utterance for each demonstration), it generally scores around 0.11[6].

Table 6.3 presents very similar results in the case were symbolic labels are included in the training data. This is done similarly to the experiments from section 3.3 and **?**: the label of each example is transformed to a vector of 10 binary values with zeros everywhere except for a one at the index corresponding to the label. The binary vector is concatenated to the vector representing the example. This setup is equivalent to adding a third modality, which contains unambiguous symbols, in order to improve the learning. The symbols are said to be unambiguous in comparison to utterances that contains several sounds where only some sequences of specific sounds form words, and generally only one word per sentence is relevant. The results from table 6.3 illustrate the fact that the system does not clearly takes advantage of this additional information. An interpretation of these results is that the system is already capable of dealing with the ambiguity and is not helped by such additional symbolic information. However the relevance of such comments is limited to the current algorithm and its implementation.

Images and sound: comparison of the image representations

In this section we perform similar experiments with the image and sound modalities. In order to get more meaningful results, the experiments are run several times for various combinations of train, test examples, and reference examples. The latter being always taken outside the two previous sets.

Figure 6.4 presents results each using a different subset of image descriptors. For each of them, results are given both for the $I \rightarrow S$ and $S \rightarrow I$ settings. The results indicate that on the dataset, the *color* and *color-pairs* representations work best. It also demonstrates that the system is rather robust to adding information: in the

[6]This is not 0.1 because the distribution of sound examples from the Caregiver dataset is not exactly uniform.

Modality			Success rate		
Test	Reference	Comparison	KL	Euclidean	Cosine
Sound	Motion	Internal	0.387	0.699	0.721
		Motion	0.543	0.261	0.424
		Sound	0.136	0.089	0.131
Motion	Sound	Internal	0.573	0.620	0.702
		Sound	0.114	0.090	0.122
		Motion	0.519	0.469	0.552

Table 6.3: There is no significant improvement of the recognition rate when unambiguous symbols are added to the training data. The table represents the same success rates as previously (see table 6.2) but with a learner that observed symbolic labels representing the semantic classes during training. The results are obtained by averaging on a ten fold cross-validation, baseline random is in that case 0.11.

results, the concatenation of several representations generally behaves nearly as well as the best of the representations, taken alone.

Other modalities

In this section we present similar results using various combinations of the motion, sound, and image modalities. Results are presented together with box plots corresponding to 20 repetitions of the experiment with random label associations, test set, train set, and reference examples. For the image modality, the color descriptors (see section 6.5.3) are used, that give the best results, as illustrated in previous section and fig. 6.4.

More precisely several setup are presented, including learning from motion and sound, as well as from image and sound, as previously, but also learning from motion and image, and finally learning from the three modalities at the same time. For each of these choices of learning modalities, several setup are possible for the test phase, specially when the three modalities are present during training: these include testing on the recognition of one modality from another (for example $I \rightarrow S$) but also from two modalities to another (for example $I, M \rightarrow S$), or conversely one modality to two (as in $M, S \rightarrow I$).

Figure 6.6 compares various one modality to one modality setups for the case where only the two modalities that are used for testing are present in the training and the case where an additional modality was also present during testing. The results demonstrate that the system is capable of learning the semantic concepts event when more than two modalities are present. There is no significant difference between the cases of two and three modalities: the system neither benefit noticeably from the third modality nor does it suffer from the increased dimensionality of the data. However, since the number of atoms k is fixed, the results could come from the fact that when the system is trained on three modalities, the dimension of the dictionary becomes insufficient to encode non-meaningful aspects of the three modalities. Therefore fig. 6.6 present the same experiment for various values of k in order to interpret more precisely the previous result. The comparison confirms the fact that the system mainly behaves similarly with two or three modalities.

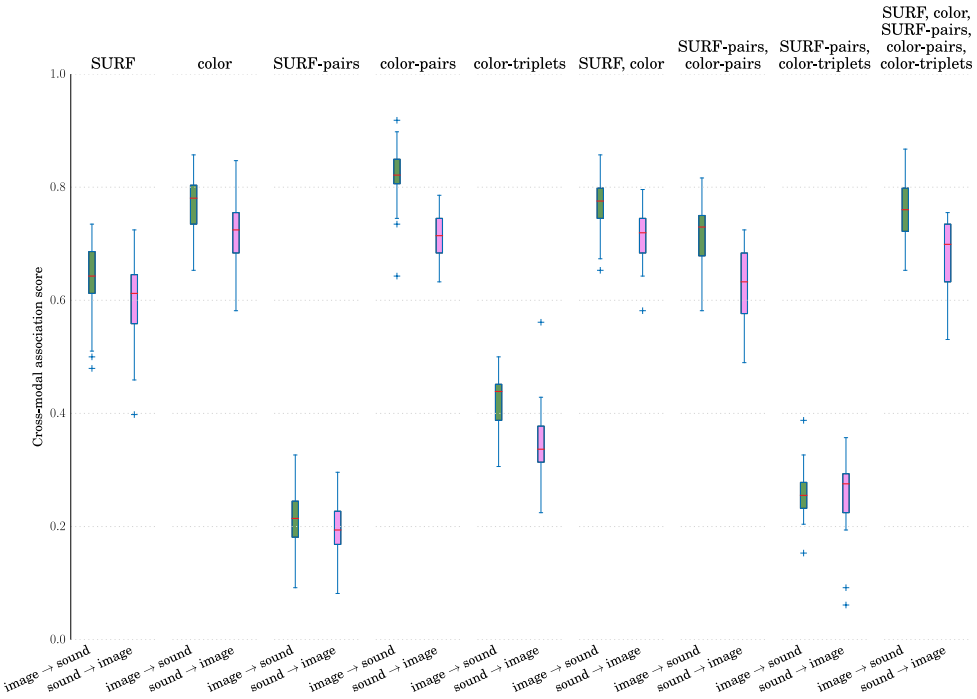


Figure 6.4: Box plot of classification success rates for various image features in the experiments $I \rightarrow S$ and $S \rightarrow I$. Each plot corresponds to the use of a subset of image features, each plot contains two boxes representing the average success as well as quantiles and extreme values through cross-validation. The features used for each experiment are presented on top of the figure. The random success rate are around 0.1.

Figure 6.7 present the results on many possible test setups int the case where all modalities are present during training. The results demonstrate that the system is capable of using information contained in more than one modality in the test or reference example. Although the results are slightly better when using more modalities as input (as in $M, I \rightarrow S$ in comparison to $M \rightarrow S$ or $I \rightarrow S$), the improvement in performance is not really significant in the experiment.

Regular classification with the symbolic modality

This section leaves the non-symbolic setup considered previously, in order to compare properties of the system described above with results obtained in previous works, such as the one of **?**.

? present a learner that is trained on multimodal examples of phonemes, either perceived through their acoustic manifestation or through the motions of the lips that pronounce them. In their experiment they show that the learner can benefit the observation of several modalities and improve its recognition success in comparison to the case where only one modality is observed.

We consider a regular classification setup, similar to the one presented in section 3.3

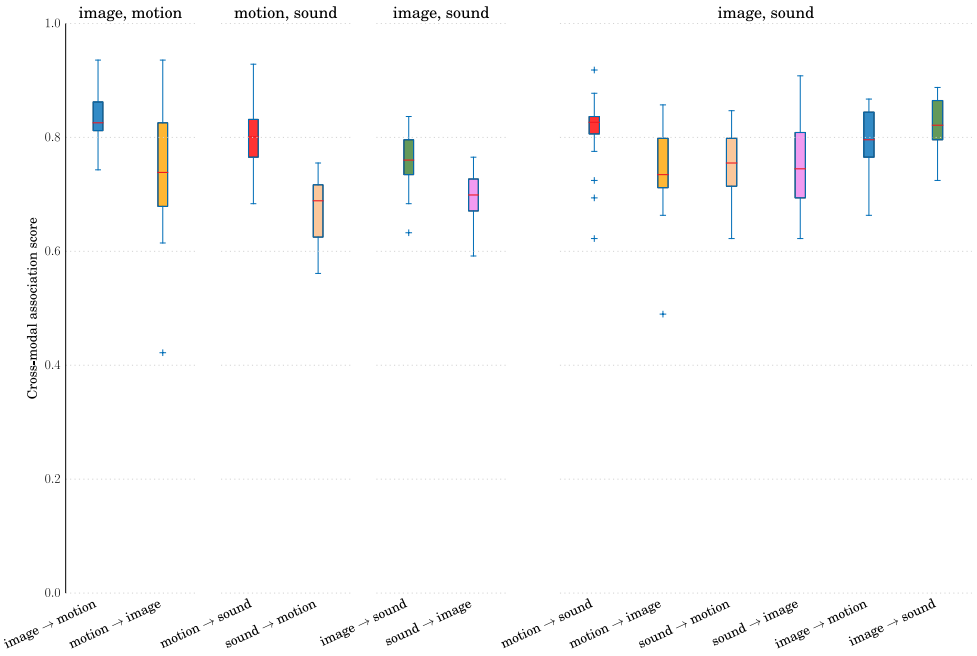


Figure 6.5: Box plot of classification success rates for various experiments where two or three modalities are used for training. Each plot corresponds to the use of a subset of modalities during training: the first three plots use two modalities and the last one use three modalities. Each plot contains boxes representing the average success as well as quantiles and extreme values through cross-validation for various testing setups, using only two modalities. There are only two testing setups when only two modalities are used for training, and six when two modalities are used for training.

but also **??**. More precisely we introduce a symbolic modality represented by a binary vector as already explained in previous section and in section 3.3. The system is trained by observing examples both in the symbolic modality and in one or several other modalities. Then results are compared between various testing setups to explore the ability of the learner to improve its classification performance in the case where several modalities are observed. Such an experiment can be described as a classification task with multimodal input unified through *sensor fusion*.

Table 6.4 present the results for such an experiment for the sound and motion modalities. The symbolic modality is denoted as L. Interestingly training with the two modalities (sound and motion) does not significantly change the performance of the learner, and that when tested on sound, motion or both. In that case the benefit of having two non-symbolic modalities is not an increase in performance, but rather that the same learner can use either acoustic perception or motion perception to classify an example.

6.6.2 Learning words in sentences

The previous experiments demonstrate that the artificial learner studied in this chapter is capable of learning the semantic connection between utterances and the

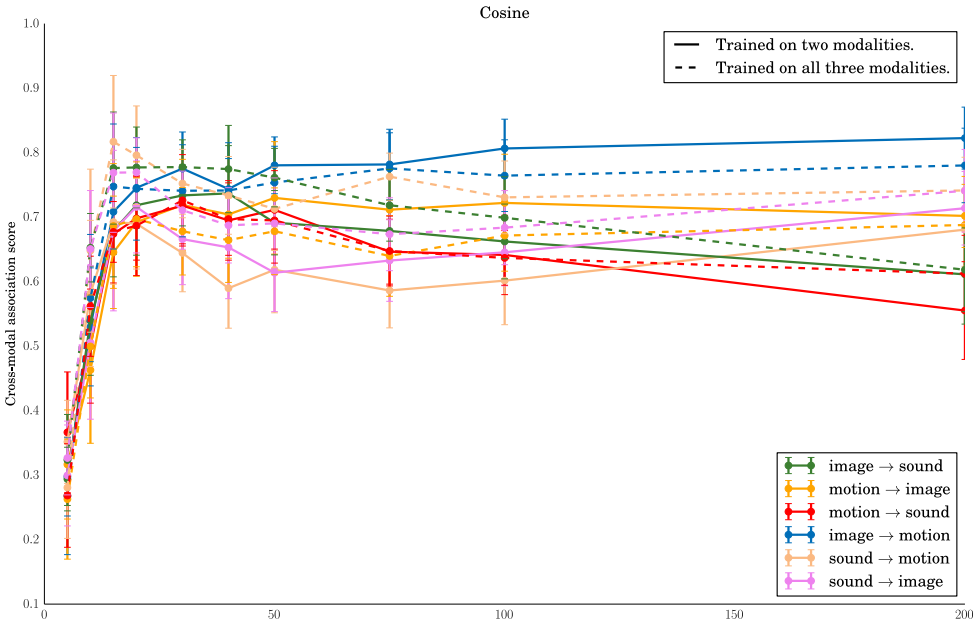


Figure 6.6: With both two (full lines) and three (dashed) modalities during training, the classification success rates are similar and good for high enough value of the number of atoms k. The plots demonstrate that the success rate is quite stable above a minimum value of k.

objects or motions they describe. The meaning of the sentences is modelled in our experiment by the presence of a keyword; more precisely the association between sentences and images of objects or motions are based on the presence of keywords in the utterances. However the learner is not aware of the fact that all the meaning of the sentence is actually localized in one word; instead it only exploits cross-situational learning to discover relations between modalities. The task solved by the learner actually only involve holistic understanding and classification of the sentences. Therefore it is not completely clear what information the learner actually exploits in the sentence and whether the learner discovers word-like units from the acoustic stream. Indeed the previous experiments only demonstrate that the learner achieves teleological understanding of the sentences; however the question remains to know if it starts to understand compositionally the sentences. We further explore this question in the experiment presented in this section.

Actually the grammar used to generate the utterances, as described quickly by **?** and in more details by **?**, chap. 2, introduces additional structure. For example the utterance: 'Now *mummy* is losing her patience.', which meaning is related to the word 'mummy' also contains the pronoun 'her' which makes it more likely that the sentence is about a feminine keyword (considering that the sentences are quite short). Furthermore it appears that the expression 'losing her patience' is always used in the dataset in the aforementioned sentence. Therefore the expression is also a relevant cue of the presence of the keyword 'mummy', although the keyword is also used in many other sentences. That example shows that it is not completely clear what cues the learner is using to recognize the meaning of sentences, and whether this cues are localized, as words, or unlocalized elements. In order to explore this question,

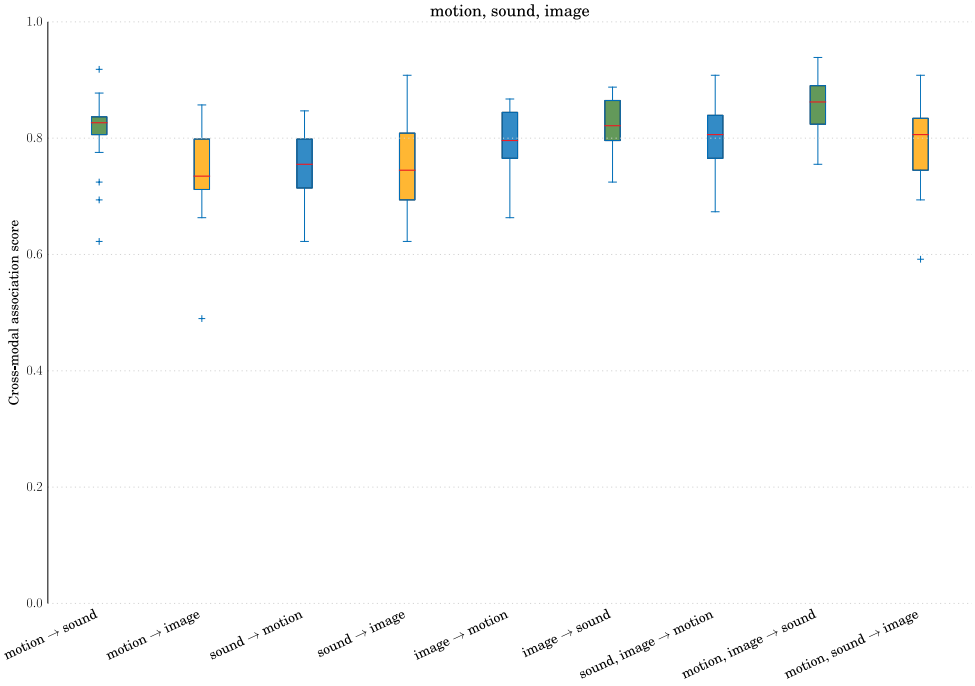


Figure 6.7: The system is capable of relating information from many modalities to on. There is however no substantial improvement in performance from the use of two modalities as input for the recognition. The figure presents box plots of classification success rates for various experiments where three modalities are used for training. There are boxes representing the average success as well as quantiles and extreme values through cross-validation for various testing setups.

another experiment was designed that uses sliding windows taken from a video-like stream composed as explained in next section. In that experiment we take a closer look to the recognition along time in the utterance of each of the semantic concept. It is somehow an extension to the multimodal setup of the experiment presented by **?**, 4.C.

Sliding windows

We consider simulated video streams, generated in the following way: acoustic records of utterances are concatenated and images are sequenced at a fixed frame rate such that the semantic concept associated to the image is the same as the one of the utterance at the starting time of the image. This construction simulates the setting where a caregiver shows objects to a learner and at the same time pronounces a sentence that describes the object.

In order to build the videos, utterances from the Caregiver dataset are concatenated in a random order. Then pictures are chosen from the dataset presented in previous section, in order to form a sequence of frames that verifies two conditions. First the sequence has a fixed frame rate close to the one used for the capture of the pictures. Also the pictures are chosen such that they represent the object corresponding to the

Training	Testing	Success rates
$S + L$	$S \rightarrow L$	0.916 ± 0.034
$M + L$	$M \rightarrow L$	0.906 ± 0.052
$S + M + L$	$S \rightarrow L$	0.896 ± 0.043
$S + M + L$	$M \rightarrow L$	0.910 ± 0.054
$S + M + L$	$S + M \rightarrow L$	0.917 ± 0.055

Table 6.4: Success rate for the label recognition experiment. In this experiment an additional modality containing labels, L, is considered. The results are computed on average for a cross-validation of the train and test sets; standard deviations are also given.

subject of the current sentence. Actually the frames having a fixed duration they may start during one utterance and end during another. This property actually introduces additional ambiguity in the data, since a sentence may start while a different object than the one described in the sentence is observed.

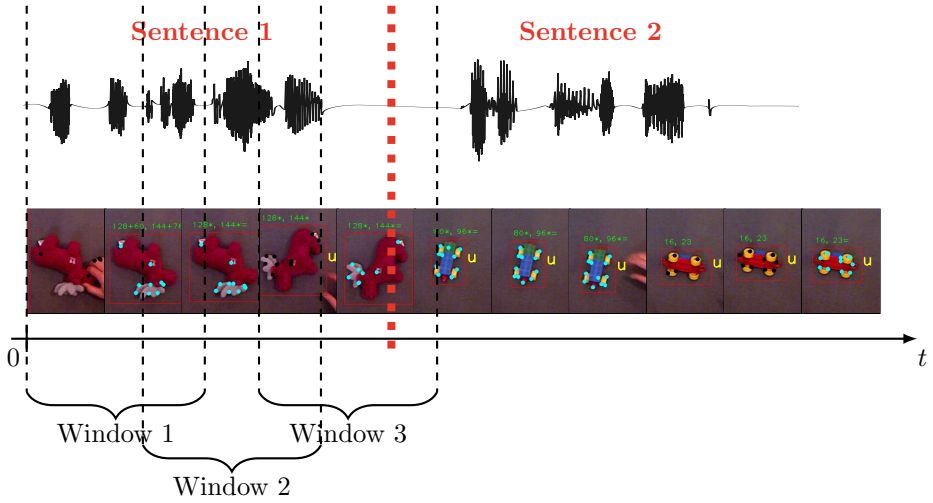


Figure 6.8: Illustration of the process used to generate video data and split into sliding windows.

The learning algorithm does not directly operate on the sound and frame streams that compose the video. Instead sliding windows of given length are extracted from the video. The sliding windows extraction process actually takes two parameters: the width of the windows and the shift between two windows. The latter is often expressed as a fraction of the former. Once a sliding window has been extracted, it is represented using similar features than regular sound and pictures. For the acoustic part, the subsequence of sound corresponding to the time window is extracted and converted to the HAC representation using exactly the same process as previously. Regarding the visual part, an histogram representation is first extracted for each frame intersecting with the time window. Then the representations of these frames are mixed using a weighted averaged with weights proportional to the duration of the intersection between the frame and the time window of interest. The length of

the sliding windows is typically between 0.05 s and 1 s, therefore between the length of a phoneme and a short sentence. One important aspect of this experimental setting is that for windows of small length, with a sliding amount of the same magnitude than the window length, one word or less is contained in the acoustic stream from the window: thus many windows contain only a part of the keyword or even do not intersect at all with the keyword. Finally many sliding windows are actually across the utterance boundaries and thus are associated with a mixture of the representation of several objects. These elements make the learning in such a context more ambiguous and more noisy since a lot of the sliding window do not contain meaningful associations.

Results

The results obtained from the sliding window training set however demonstrate that the learning system is robust to the additional ambiguity and noise brought by the setup. **??** presents success rates obtained as previously on the learning task, for various values of the window length. The overall decrease of performance is coherent with the above analysis of the increased difficulty of that task. In these results full sentences and single images are used as test and reference examples. It is indeed not really meaningful to evaluate quantitatively the system on sliding windows that do not contain any keyword and whose affectation to semantic concepts might be ambiguous.

The behavior of the recognition of the system on sliding windows taken inside utterances is however very interesting on a qualitative point of view. In the following, acoustic sliding windows are computed on utterances outside the training set and the system is evaluated on the value of similarity it returns between each acoustic sliding window and an image. The results of this experience, as presented in fig. 6.9 provide a better insight of which parts of the utterances are more strongly associated with the underlying semantic concepts. Actually the results show that the recognition of the object are often localized in the sentence around the temporal occurrence of the keyword.

The results also illustrate the fact that in some sentences, as the example 'Now *mummy* is losing her patience', the keyword is not the only part of the sentence that is meaningful regarding the semantic concept, but other elements such as the expression 'is losing her patience' or the pronoun 'her' are also associated to the semantic concept. Figure 6.9 illustrate this effect.

6.6.3 Emergence of concepts

In previous sections we evaluated the learner on concrete tasks that emphasis its ability to relate information from one modality to another. A natural question that follows is whether the learner develops an internal representation of the semantic concepts from the data, although it does not observe the symbolic information. The question is actually non-trivial since it is not immediate to interpret the internal representation that the system builds, that is to say, the role of the various components of the dictionary matrix. However some insight can be gained that suggests that at least some components are more specialized into some of the semantic classes.

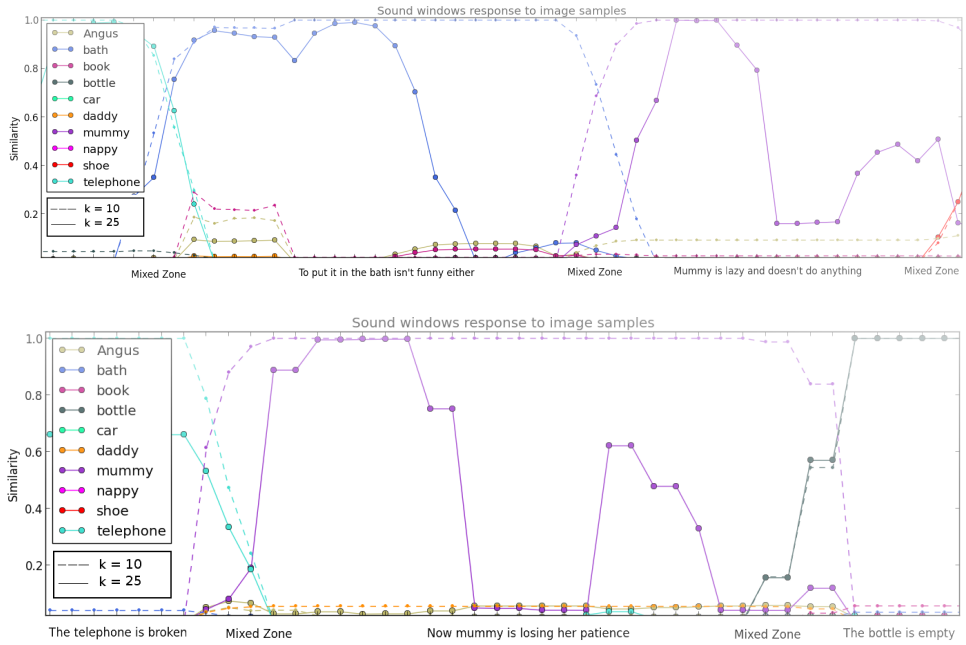


Figure 6.9: Examples of similarity to images around a time window sliding through the utterance. The similarity is represented at the time index of the beginning of each window. Interestingly this example demonstrates an association between the pronoun 'her' and the concept 'mummy', since 'her' is only used as a synonymous of 'mummy' in the set of utterances.

In order to investigate that aspect we quantified the mutual information between the semantic concepts and the coefficients of the internal representations of samples featuring the concepts. For each semantic concept l and sample i we consider the random variables X_l such that $X_l^i = 1$ if and only if the concept l appears in sample i. For each dimension j of the internal representation and each sample i we define the random variable $Y_j = h_j^i$. We then assume that $(X_l^i)_i$ are independent and identically distributed, as well as the $(Y_j^i)_i$. In the following we quantify the dependency between these two variables by looking at the mutual information between them. In information theory, the mutual information I is an information theoretic measure defined for two random variables X and Y as "the relative entropy [or Kullback-Leibler divergence] between the joint distribution $[p(x, y)]$ and the product distribution $p(x)p(y)$" by **?**.

$$I(X;Y) = D_{KL}\left(p(x,y) \| p(x)p(y)\right)$$

The X_l variables takes binary values but the Y_j are continuous. Therefore we use a discrete approximation of the values of the coefficients h with 10 bins in order to be able to compute the mutual information by estimating the probability distributions $p(X_l)$, $p(Y_j)$, and $p(X_l, Y_j)$ by using the samples for $1 \leq i \leq N$. From this process we obtain a value $I(X_l; Y_j)$ for each pair (l, j) that quantifies how much information the coefficient j captures from the concept l.

Figure 6.10 represents, for each semantic class and each coefficient of the internal

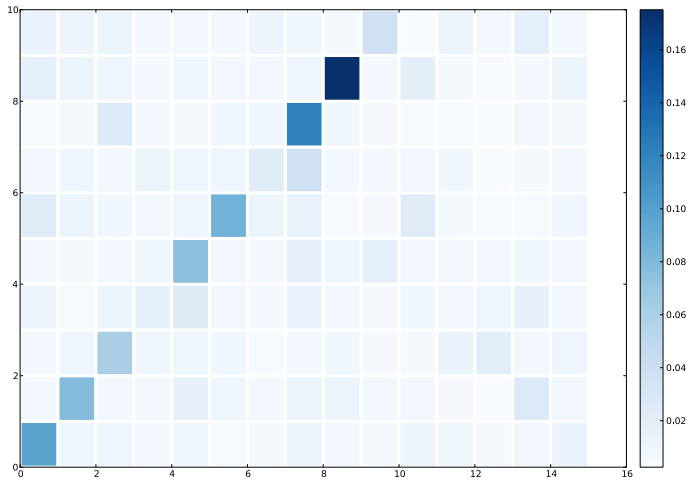


Figure 6.10: Illustration of the specialisation of some components with respect to some semantic labels. The figure represents the mutual information between (vertically) semantic classes (that are not observed by the learner) and (horizontally) each internal coefficient used by the learner to represent pairs of motion demonstration and acoustic descriptions from the training set. A value of $k = 15$ was used in this experiment.

representation, the mutual information between the belonging of examples to that class and the value of a given coefficient of the internal representations of these examples. To emphasis the specialisation of some internal coefficients we re-ordered internal coefficients so that classes and coefficients that have high mutual information are aligned. More precisely, the best alignment was computed by a Kuhn-Munkres algorithm and we plotted first the coefficients that are highly associated to one class and then the one that are less meaningful. The figure does not display a perfect one to one relationship between labels and coefficients; however some coefficients are highly specialized, the information about other labels is spread over several coefficients, and some information is not clearly localized.

6.7 Conclusion

This chapter presents a truly multimodal learning experience in the sense that a learner is trained from several subsymbolic modalities and demonstrates a classification behavior similar to the one that would be expected to a young child. More precisely it is shown that the result can be obtained from the auto-organization of the multimodal perception driven by the compression mechanism on which nonnegative matrix factorization is based. These experiments demonstrate the effective exploitation by the learner of the cross-situational information from the data.

The behavior of the learner is explored on several setups such as learning from three modalities or with a symbolic modality. It is also explored whether, when it is used as a regular classifier, additional modalities during training and testing improve the results (as in **?**). It is also shown that the system can be implemented, using sliding time windows, in a more realistic setting where utterances do not need to be segmented

beforehand. This setting demonstrates the robustness of the system to demonstrations that are not meaningful, that is that do not feature the cross-situational semantic association between modalities. Furthermore the setting is also used to gain better insight on the temporal localization of the learnt concept among sentence. It is illustrated that the meaning of sentences, regarding the semantic objects, is localized mainly around keywords but also around other meaningful grammatical elements that where in the training set associated with the concepts. Finally the experiment show that the system not only learns semantic associations between words, visual objects, and gestures, but also that a representation of the semantic classes, although never observed directly, emerges in the system's representation.

The learner presented in this chapter specifically targets and is tested on the cross-situational learning setup. Although cross-situational information is not the only one that might be involved in the learning of multimodal concepts. Indeed children have been shown to rely on other important elements in the process of word acquisition. For example, **?** have shown that children might rely on spatial localization of objects and words to relate words to their meanings. **?** provided a computational model to compare cross-situational learning to social learning and shown that the latter outperforms the former. The interaction of the learner with the caregiver or other agents is indeed an important aspect of word learning. Actually the word learning process used in the *Talking heads* experiment (**?**) mainly rely on the interaction between the agents. Finally **??** details various aspect of conceptual thinking that seems to play an important role learning words by using additional heuristics such as the *whole object assumption*, *taxonomic assumption*, or *mutual exclusivity*.

With regard to the question of the precedence of *teleological* or *holistic* understanding over *compositional understanding* as discussed by **?**, the experiments presented in this chapter do not assume that the sentences and more generally the semantic concepts has to be understood first in compositional manner. Indeed it instead focuses more on global understanding: the learner we present does not include mechanism to segment the perception in parts and then try to relate these parts from one modality to the other, as in previous works from **????**. Instead the system learns a representation of whole sentences and the main task demonstrates that the representation is efficient. We further refine the learning process to demonstrate that representation learnt by the system actually focuses on recognizing the keywords that concentrate the cross-modal information. Therefore aspects of compositional understanding also emerge from the learning process. The compositional aspect of the semantics presented in our experiment is actually quite limited; in order to investigate more these aspects, as well as the question of grammar learning, experiments including a more complex and structure semantics are required. One such experiment results from the combination of several concepts in each samples: this corresponds to the semantics used in the experiment from chapter 3 but presented with real sentences instead of symbolic labels. Finally such an experiment would also explore aspects of the similarities between the grammar of language the grammar of motions as described by **?**.

As already mentioned, the learner presented in this chapter can be seen as an agent from the *Talking heads* experiment from **??**. The main difference between our implementation and the original one is that our agent uses cross-situational information instead of feedback from the other agent. As studied by **?** these two approaches are both valid to explain the learning of words, and might be used complementary, for example using cross-situational information to exploit language

exposure when no interaction is available. The system we present in this chapter features other important differences with the original *talking head*: the fact that the algorithm does not rely on a preliminary process of word segmentation and concept segmentation opens new perspectives on the study of the interaction between the formation of concepts and the formation of words.

From a technical point of view, the algorithm and setup presented above could however be improved in many ways. One direction for further work is the development of algorithm more adapted to the setup. Indeed the NMF algorithm is built to take independent samples as input and therefore information must be sliced into such samples in order to be learned by the system. On way to slice the data is to detect utterance boundaries or manually annotate them. Another is to use sliding windows of fixed or varying lengths. However, other algorithms could be used that directly model the temporal dynamics of the signal. Such algorithms could be derived from existing models, as hidden Markov models, or more recent work by **?** that extends NMF.

The NMF algorithm used in these experiment implements the optimization of one specific criterion, namely reconstruction error under the constraint of the factorization. Many other properties of the learnt representation could be used to improve the results. For example, sparsity of the representation have been shown to provide more meaningful results in many application fields (as **????**). Interestingly other metrics have been explicitly developed in the past that target multimodal learning. Such examples are given by extensions of independent component analysis to multimodal settings by **??**. Actually these technical aspects introduce the deeper question of the mechanism that drive the learning of multimodal concepts: the experiment we present demonstrate that an algorithm based on a compression mechanism can capture semantic information by exploiting the cross-situational information from the data. Many other heuristics could eventually be used to capture that information. Would they behave similarly? Would they enable the extraction of other information than the cross-situational one? Several important questions are actually raised by these possibilities, that might help to model more precisely what it means to achieve multimodal learning.

Chapter 7

Discussion and perspectives

In this thesis we explore the problem of learning lexicons of primitive elements from perception and their association across modalities. This study involves the question of the definition of 'simple', in particular with respect to what complex is, more specifically through mechanisms that enable the discovery or emergence of such simple elements from perception, and how these mechanisms can handle the ambiguity often inherent to the definition of simple elements.

It is explained in chapter 1 that the notion of 'motion primitive' is far from being straight-forward and unique. Then, the example of dance motions is given in which choreographies are composed of parts combined simultaneously. We have thus explored the question of learning simple elements from observation of complex motions, where the simple elements are active at the same time. Learning this kind of combination of motion primitives is actually an instance of the source separation problem, in the fields of imitation learning and human behavior understanding. The approach developed in chapter 3 therefore uses nonnegative matrix factorization, a technique commonly used on source separation problems (for example for sound in **?**). A lot of work related to learning motion primitives actually focus on learning sequences of motion primitives. However the experiments presented in this thesis demonstrates that it is also possible to decompose motions in simpler parts that are active simultaneously. This question is actually orthogonal to the sequence decomposition but less addressed in the literature. In this work we explained the ambiguity issues deriving from this problem and demonstrate that NMF can handle such ambiguity, as appearing in the motion dataset, when it is coupled with weak supervision in the form of linguistic data.

Chapter 4 explores the use of similar ideas to decompose observations of humans solving complex tasks into a dictionary of simple tasks. In that chapter, the novel idea is to consider the combinatorial properties of human actions not in the policy space, that is the space of actions, but in the task space, that is the space of objectives the actions are pursuing. Chapter 4 demonstrates on synthetic problems that it is possible to exploit the combinatorial structure of tasks modelled by objective functions from the observation of artificial agents solving these tasks. Interestingly the new matrix factorization algorithm derived in these experiments are based on mechanism very similar to the ones behind nonnegative matrix factorization algorithm.

Chapter 5 reviews mechanisms that can be used to learn primitive elements from sound. More precisely we introduce an unsupervised algorithm based on clustering and *bag-of-words* representation, that can be used to learn primitive elements. On top of these primitive elements, interpreted as phonemes, words can be learnt in a supervised manner, using a voting technique based on TF-IDF score. We also detail how nonnegative matrix factorization have been used on top of similar basic primitive elements in an unsupervised setup by **?** and in a supervised manner by **??** to also achieve word learning.

Finally, in chapter 6 we explain how the mechanisms used for learning words from acoustic observation of utterances or gestures from observation of motions can be extended to a multimodal learning setting. More precisely we demonstrate that a NMF based algorithm is capable of learning simultaneously primitive elements in a multimodal setting without any symbolic input nor explicit models of words and meanings. Actually a learner is shown to yield the same classification behavior that would be expected from a child after being exposed only to subsymbolic data. The learner presented in chapter 6 exploits cross-situational learning to relate words to gestures and/or visual objects. The presented learner is further analysed in order to show that the semantic relations it learns between modalities make it able to localize the information contained by keywords among utterances. The representation of multimodal data learnt with NMF is also shown to yield, to some extent, a specialisation of some of the dimensions to the recognition of semantic concepts from the training data. In these experiments the semantic concepts that the system learns are the result of a convention, that maps keywords in utterances to objects in visual scenes and gestures demonstrated through a motion acquisition system. The experiment therefore demonstrates to what extent the mechanisms of nonnegative matrix factorization can recover the correlations that characterize these semantic concepts, when embodied in a cross-situational learning setup. These results comfort those of **??**; we however use a setup in which a single algorithm treats all modalities in a unified way. Furthermore the setup we present starts from a representation that does not involve explicit segmentation and static phoneme recognition. Finally chapter 6 discuss how other cues might be included in similar synthetic learners, such as spatial localization of objects (see **?**), interaction with the caregiver or other agents (see **?**), or conceptual thinking (see **??**).

The contribution of this thesis can be summarized as using matrix factorization techniques to model the emergence and acquisition of primitive elements in the perception of motions, complex behaviors, objects vision, and spoken language. First this thesis illustrates the similarity of the problem in various fields of perception by using the same family of algorithm in all of them. It then shifts to the specific study of the language grounding problem and more generally the question of multimodal learning. More particularly multimodal learning is not only a way of solving the same problem in several modalities at the same time, but to make less ambiguous, in the multimodal setting, several problems that, taken alone, are ambiguous. Regarding the three questions that were asked in the introduction of this thesis, this works provides the following answers and new questions.

How can the intuition about *simple* and *complex* be made explicit and implemented on a robot or artificial system? For developmental roboticists an appealing aspect of the concept of *motion primitives* is that they can implement

the gradual growing of complexity observed for example in tasks executed by children. This corresponds to intuitions such as: mastering grasping and placing of objects is a prerequisite to being able to assemble cubes into towers. However an explicit representation of motion primitives is not required to explain such increase of complexity; also, as explained previously, it is very difficult to formulate a definition of these motion primitives: the intuition of motion decomposition may lead to many different approaches, each of which is not necessarily useful or even well defined. An alternative is proposed that identifies and represent the combinatorial properties of motion skills without explicit definition of what the basic motions are. The work presented in chapter 3 is actually introducing three important ideas on that subject, that are then followed in different directions. First, many work have been done to represent motion primitives and their combination in sequence. Thus we introduced similar work for the simultaneous combination, using the NMF algorithm. Then, although the NMF algorithm is based on the linear combination of elements from a dictionary into observed examples, it is not claimed that these atoms corresponds as motion primitives as perceived by humans. Actually our preliminary experiments following that idea show that it is often not the case, one reason for that being the intrinsic ambiguity or indeterminacy of the decomposition problem. In other words, the system learns and represents motions in a form that implements simultaneous combination, but do not presuppose the ability to segment or separate basic motions, as perceived by humans amongst the data. Indeed, building motion representations that are compatible with some notion of combination is a distinct approach from trying to first represent parts of motions and then implement their combination. Both enable the representation of composite motions but the former does not involve the pre-requisite of motion segmentation. As discussed in next paragraph the former approach is more compatible with the idea that holistic understanding may emerge before compositional understanding (see **?**). Finally, in order to evaluate the system, we chose to test its ability to represent similar combinations than a human does, at the behavioral level. For that, we use a tasks based on a communication channel modelled by the symbols. In other words the ambiguity is only addressed by the addition of a linguistic channel that models social interaction and provides an input for social conventions. The other experiments explore similar ideas. For example, the experiment on tasks decomposition gives a visual example of the multiplicity of the solution and the non-explicit representation of tasks that where used to generate the examples. However the learning of the structure is demonstrated at the behavior level by the score on imitating the demonstrator task.

This thesis however does not treat other combinatorial properties that are central both for motion or language. Examples of such combinations have already been discussed in section 3.1.1 and includes time sequences and hierarchies of primitives. As these have often been explored individually in previous works, the main challenge is to be able to combine several forms of combinations, which requires the development of representations and algorithms that implement them as well as new experimental setup. For example on possibility is relax the hypothesis of sample independence in NMF to represent the temporal dynamics of the data, following work from **?**.

How can primitive elements emerge or be discovered through interaction of the agent with its physical and social environment? The experiments developed in this thesis are based on matrix factorization or clustering algorithms. These algorithms can all be interpreted as the minimization of a form of reconstruction

error of the perceived signal, under the constraint of compression. That mechanism is the one that yields the emergence of primitive elements, at the behavioral level. While this mechanism is the only one studied through the experiments from this thesis, many other candidates exist. First structural constraints can be added to the compression performed in the algorithms. For example, sparsity inducing constraints were already combined with matrix factorization approaches (**????**). Interestingly **?** also achieve both sparsity and hierarchy of dictionary elements. The work on deep belief nets achieve experiments that are conceptually similar to the one presented here, both in the field of motion composition (see **???**), and multimodal learning (**?**). However the mechanisms behind the learning of such representations are not exactly the same as the one underlined matrix factorization. Other metrics that have been explicitly developed to target multimodal learning were already given as examples in previous chapter, such as extensions of independent component analysis to multimodal settings by **??**. The comparison of these technical approaches with the one taken in this work on a unique multimodal setup actually constitutes an interesting direction for future research. We explain in chapter 6 that an important novelty of our approach in comparison to previous work on similar questions is that it does not presuppose the decomposition of perception in concepts and words to the learning of their relation. More precisely we present a system that learns to relate words to concepts without explicit segmentation of acoustic input into phonemes or words, nor images into relevant objects. It thus demonstrate that a task of language grounding can be solved in holistic way, thus featuring teleological understanding without requiring compositional understanding. Indeed in previous work from **????** but also in the *Talking head* experiment from **??** the algorithm include explicit mechanism to segment the sensor input into either phonemes and then words or concepts. Even if the segmentation is in some cases learnt by the system, these architecture encode compositional understanding as a prerequisite to language grounding and more generally multimodal learning. In our work we provide an alternative which open new perspectives on the kind of questions introduced by **?**. It is important to notice that an important shortcoming of our approach with respect to the study of *emergence* is that it is not incremental. However there exist online versions of the NMF algorithms on which we base our experiments that can be used to study the framework presented in this thesis in a more incremental manner.

What mechanisms can overcome the intrinsic ambiguity and indeterminacy that is characteristic of many approaches regarding the learning and emergence of these primitive elements? In chapter 3 the mechanisms used to learn motion representation do not alone solve the indeterminacy of decomposition. However we show that adding weak supervision to the perception through a linguistic channel, symbolic in that experiment, resolve the ambiguity at the behavioral level. Chapter 6 actually demonstrates the same idea from real multimodal perception; the learner can then exploit cross-situational and cross-modal information to achieve word grounding or the learning of other kinds of multimodal concepts. Many other mechanisms can be implemented at the data collection level and the way to represent it. The multimodal experiment represents data in such a way that makes learning possible from cross-situational information; however, as mentioned previously, other sources of information are available such as localization of objects or interactions with the caregiver. Furthermore, in the experiments in this thesis we use data coming from three distinct dataset to simulate multimodal perception. This constitute an

important limitation of this work which hides important questions. For example whether the object is shown by the caregiver or the learning system is an important aspect that may greatly change the structure of the collected visual data. Other questions include the differences between infant and adult directed speech as discussed by **?**, or the process of autonomous acquisition of data (see **?**).

This thesis presents the simultaneous combination of motion primitives as an important aspect of imitation learning and human behavior understanding. Following that idea, an experiment demonstrating such learning is performed, where symbolic linguistic input is used to relate the motion representations learned by the system to a human based representation. In chapter 6 the symbolic linguistic modality is replaced by a continuous acoustic input. However theses last experiments only study the learning of gestures demonstrated alone. An important extension of the work presented here is therefore to bring together multimodality from only continuous perception and fully ambiguous demonstrations: which means relating gestures that are observed combined in complex motions to words that are observed combined in complex sentences. Such an experiment actually is a very interesting first approach to the question of similarities between the combinatorial structure of words and motions, that is to say the grammar of language and the grammar of motions and actions. It would be even more interesting to relate that experience to the analysis and hypothesis from **?**.

The experiments presented in this thesis are performed either on simple toys problems or on fixed datasets. It is therefore important, in order to consolidate the results, to experiment similar ideas both on more diverse and advanced datasets, and in more interactive setups. Indeed, while the issues addressed by this work are clearly grounded in developmental robotics, no real robotic experiment was performed in this thesis. This aspect actually corresponds to the fact that this work focusses specifically on a perception based point of view. We actually believe that many of these questions are also very relevant to action but the link is still to be made. Although chapter 4 introduces the ideas of task decomposition in a way that enables the generation of motions to solve new tasks, it is limited to simple toy problems and suffers from the weaknesses of the inverse reinforcement learning approach it is based on: the relative novelty of the field and the algorithmic cost that still makes it a challenge to reach the complexity of real robotic environment. Also, the algorithm we propose focuses on the problems of learning by imitation and human behavior understanding: a demonstrator is required to be able learn tasks; this leaves apart the important mechanisms for autonomous discovery of new motor skills and new goals. Other interesting questions emerge from the duality between goals or tasks and the skills that solve them. For example, is one of them explicitly represented and the other inferred or both represented simultaneously? Is the autonomous discovery of skills motor based or task based? This question is for example related to the study of motor babbling and goal babbling by **?**. Can the learning of primitive tasks model the emergence of affordances in perception? Regarding language learning, the work we present also focusses mainly on perception and word recognition. Extending this work to language production is also an interesting direction for future research, specially through the studies of the interactions between the learning of language perception and language production. Aspects of this interaction have already been studied by **??**.

Appendix A

Non-negative matrices and factorization

A.1 Non-negative matrix theory

This section summarizes some result from the Perron-Frobenius theory that are useful to study theoretical aspects of the non-negative matrices and factorization.

Its purpose is to present main properties of these matrices, centered on Perron-Frobenius theorem, that enables basic theoretical discussions. These elements are adapted from presentation given by **??**. These references might be looked into for proofs and more detailed presentations of this theory.

A.1.1 Base definitions and notations

We only consider square matrices in this part.

We call positive (resp. non-negative) a matrix whose coefficients are all positive (resp. non-negative), we use notations $A > 0$ for positivity and $A \geq 0$ for non-negativity.

We denote by $\sigma(A)$ the set of all A's eigenvalues. We call **spectral radius**, denoted $\rho(A)$, the quantity defined as follows:

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| \tag{A.1}$$

We also denote by $\chi_A(X)$ the characteristic polynomial of A.

Multiplicities The **algebraic multiplicity** of an eigenvalue refers to its multiplicity as root of the charateristic polynomial whereas **geometric multiplicity** refers to the dimension of the associated eigenspace. In general the algebraic multiplicity is greater or equal to the geometric multiplicity. When these two quantities are equal the eigenvalue is said to be **semi-simple**. When both are equal to 1, the eignevalue is said to be **simple**.

Two matrices A and B are said **equivalent** if and only if there exists an invertible matrix P such that $B = P^{-1}AP$.

A.1.2 Taxonomy of non-negative matrices

Amongst all non-negative matrices, a few categories are of great importance regarding the Perron-Frobenius theory. These are **positive**, **primitive** and **irreducible** matrices.

Definition (Primitivity)**.** *A square non-negative matrix is said to be primitive if and only if some power of it is positive*

$$i.e.\ \exists k \in \mathbb{N},\ A^k > 0 \tag{A.2}$$

$$i.e.\ \exists k \in \mathbb{N},\ \forall (i,j) \in [|1,n|]^2,\ (A^k)_{i,j} > 0 \tag{A.3}$$

Definition (Irreducibility)**.** *A square non-negative matrix is said to be irreducible if and only if*

$$\forall (i,j) \in [|1,n|]^2,\ \exists k \in \mathbb{N},\ (A^k)_{i,j} > 0 \tag{A.4}$$

Given a square non-negative matrix A let $G(A)$ be the directed graph with n vertices and such that an array exists between vertex i and vertex j if and only if $A_{i,j} > 0$. Then for some $k \in \mathbb{N}$, $(A^k)_{i,j} > 0$ is equivalent to the existence of a path of length k between i and j in $G(A)$.

Definition (Period of a square non-negative matrix)**.** *The period of a square non-negative matrix A, is the greatest common divisor of all length of loops in $G(A)$.*

For an irreducible matrix A of period p, we define the relation:

$$i \sim_A j \text{ if and only if } \exists k \in \mathbb{N},\ (A^{kp})_{i,j} > 0 \tag{A.5}$$

Following this definition, $i \sim_A j$ if and only if p divides all path lengths between i and j in $G(A)$. That this relation is an equivalence easily follows the irreducibility of A. Using the partition of $[|1,n|]$ into equivalence classes for $i \sim_A j$, we can re-order the columns of A, which makes it equivalent to B where:

$$B = \begin{pmatrix} 0 & A_1 & 0 & \cdots \\ 0 & 0 & A_2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ A_p & 0 & 0 & \cdots \end{pmatrix} \tag{A.6}$$

This form is referred as **cyclic block form** or **Frobenius form**. This formulations makes it easy to get the following results.

Proposition 2. *A is primitive if and only if A is irreducible of period 1.*

Proposition 3. *Let A be an irreducible matrix of period p. Then A^p is equivalent to a block diagonal matrix which blocks are all primitive. Moreover all blocks have the same non-zero spectrum.*

Proposition 4. *Let A be an irreducible matrix of period p and ξ a primitive p^{th} root of unity. Then*

- *A and ξA are similar,*
- *if r is a root of the characteristic polynomial of A, with multiplicity α, then $r\xi$ is a also a root of A of multiplicity α.*

Theorem 4. *Let A be an irreducible matrix of period p and D a diagonal block from B^p where B is a cyclic block form of A. Then, $\chi_A(X) = \chi_D(X^p)$. More precisely if ξ is a primitive p^{th} root of unity and*

$$\chi_D(X) = \prod_{j=1}^{k} (X - \lambda_j^p),$$

$$\chi_A(X) = \prod_{i=1}^{p} \prod_{j=1}^{k} (X - \xi^i \lambda_j).$$

Those results enable to relate the spectrum of any irreducible matrix of period p to the spectrum of primitive matrices. This relation makes it possible to generalize some of the results for primitive matrices to irreducible matrices, as stated in Section A.1.3.

A.1.3 Perron-Frobenius theorem

Case of positive matrices

Theorem 5 (Perron theorem)**.** *Let A be a positive matrix, then,*

(i) *$\rho(A)$ is a simple eigenvalue of A,*

(ii) *there exists a unique unit norm positive eigenvector u,*

(iii) *it is associated with $\rho(A)$ and called the **Perron vector** of A,*

(iv) *$\rho(A)$ is the unique eigenvalue of A of norm $\rho(A)$.*

Case of irreducible matrices

When generalized to irreducible matrices the previous result takes the following form.

Theorem 6 (Perron-Frobenius theorem)**.** *Let A be an irreducible matrix of period p, then,*

(i) *$\rho(A)$ is a simple eigenvalue of A,*

(ii) *there exists a unique unit norm positive eigenvector u,*

(iii) *it is associated with $\rho(A)$ and called the **Perron vector** of A,*

(iv) *A has exactly p eigenvalues of norm $\rho(A)$, furthermore A is similar to $e^{\frac{2i\pi}{p}} A$ and thus $\sigma(A)$ is invariant by rotation of angle $\frac{2\pi}{p}$*

Perron projection

Previous results does not change when a matrix A is replaced by A^T since all the considered properties are invariant by transposition. However the Perron vectors of A and A^T are in general not the same. It is thus useful to distinguish between left ($v > 0$) and right ($u > 0$) Perron vectors of matrix A such that:

$$Au = \rho(A)u \quad \text{and} \quad v^T A = \rho(A)v^T \tag{A.7}$$

Proposition 5. *Let A be a primitive matrix with left and right Peron vectors v and u.*

$$\left(\frac{A}{\rho(A)}\right)^n \underset{n\to\infty}{\longrightarrow} \frac{uv^T}{u^T v}$$

*This quantity is a projection onto the eigenspace associated to $\rho(A)$, which is called **Perron projection**.*

Collatz-Willandt formula

Proposition 6. *Let A be an irreducible matrix and $f(x) = \min_{i\in[|1,n|],\ x_i\neq 0} \frac{[Ax]_i}{x_i}$, then*

$$\rho(A) = \max_{x\leq 0,\ x\neq 0} f(x).$$

A.2 Ambiguity in the problem definition

A.2.1 Generative model

A geometric interpretation of the non-negative matrix factorization has been presented by Donoho and Stoden **?**, it is based on the notion of simplicial cones and the following link to non-negative factorization.

Definition (Simplicial cone)**.** *The **simplicial cone** generated by the vectors $(w_k)_{k\in[|1,K|]}$ is defined as:*

$$\Gamma_W = \left\{ \sum_{k=1}^{K} h_k \cdot w_k : h_k \geq 0 \right\} \tag{A.8}$$

For a given factorization $W \cdot H$ that generates exactly the data, W yields K generators such that all data vectors $(x_i)_{i\in[|1,N|]}$ lie in the simplicial cone Γ_W.

The factorization of a non-negative matrix is thus equivalent to providing such a simplicial cone as a generative model of the data. It is however not true in general that, even if the data is generated filling a simplicial cone, there is uniqueness of such a model. In such a situation one would like to chose the *simplest* or the *smallest* fitting model. In some cases even defining such a simplicity is ambiguous.

In the following we analyse separately various sources of ambiguity in the problem of finding a factorization:

- ambiguity in the **representation** of simplicial cones,

- ambiguity in notions of *simplest* solution.
- ambiguity from **lack of data**.

Since the data is non-negative it lies in the positive orthant $\mathcal{P}$ which is the convex simplicial cone generated by the canonical basis vectors, the problem thus always admit $\mathcal{P}$ as a solution.

Furthermore it also lies in $\mathrm{Span}\Big((w_i)_{i\in[|1,K|]}\Big)$, and $\mathcal{P}\cap\mathrm{Span}(W)$ is also a solution.

A.2.2 Representation of simplicial cones

Definition (Extreme rays)**.** *An **extreme ray** of a convex cone Γ is a line $R_x = \{ax : a \geq 0\}$ such that $x \in \Gamma\backslash\{0\}$ and there is no x_1 and x_2, linearly independent such that $x = x_1 + x_2$.*

When the generators are linearly independent, the set of generators corresponds to the set of extreme ray, thus a convex simplicial cone is uniquely defined by its set of extreme rays.

This shows that solving non-negative matrix factorization is equivalent to finding a set of extreme rays generating a simplicial cone containing the data. The set of rays is represented by a list of generators. Scaling of the generators and re-numbering of the rays does not change the found simplicial cone, which is another formulation of the invariance introduced in section 2.1.1.

Definition (Primal simplicial cone)**.** *Given a set of points $\mathcal{X}$ and an integer r, a **primal simplicial cone** associated with r, $\mathcal{X}$ is a simplicial cone Γ such that $\mathcal{X} \subset \Gamma \subset \mathcal{P}$.*

The invariance by scaling and permutation is thus a representation invariance of the simplicial cone underlying the factorization.

Dual formulation This formulation of the NMF problem can equivalently be made in the dual space (in terms of complex duality, see **?**).

Definition (Dual simplicial cone)**.** *Given a set of points $\mathcal{Y}$ and an integer r, a **dual simplicial cone** associated with r, $\mathcal{Y}$ is a simplicial cone Γ such that $\mathcal{P} \subset \Gamma \subset \mathcal{Y}$.*

Proposition 7. *Every primal simplicial cone is the dual of a dual simplicial cone and reciprocally.*

A.2.3 First case: linearly independent generators

We first consider the somehow simpler case in which the generators of $\mathcal{X}$ are linearly independent, which is a sufficient condition for these generators to be extreme rays of $\mathcal{X}$.

However, if $\Gamma_X \in \mathcal{P}\cap\mathrm{Span}(X)$ where $X = (x_i)_{i\in[|1,r|]}$ is such that $\Gamma_X \neq \mathcal{P}\cap\mathrm{Span}(X)$, there are an infinity of simplicial cones Γ' with r generators such that $\Gamma_X \subset \Gamma' \subset \mathcal{P}$ ($\mathcal{P}\cap\mathrm{Span}(X)$ is one of them).

The following lemma from **?**, however limits the solution of such simplicial cones inclusion.

Proposition 8. *If* Γ *and* G *are convex cones such that* $\Gamma \subset G \subset \mathcal{R}^n$ *where* Γ *is a simplicial cone with* r *generators and* $\Gamma \cap G$ *contains exactly* r *extreme rays of* G, $\left(Rx_i\right)_{i\in[|1,r|]}$, *then:*

- *the* $\left(Rx_i\right)_{i\in[|1,r|]}$ *are extreme rays of* Γ,
- *there is no simplicial cone* Γ' *with* r *generators such that* $\Gamma' \neq \Gamma$ *and* $\Gamma \subset \Gamma' \subset G$.

So in general such a cone might be *widened* towards $\mathcal{P}$ and is thus not unique. However under some conditions the cone is already *maximal*, such a case happens under sufficient conditions given in **?**. In that case the primal simplicial cone is unique and so is the solution to the NMF problem (still with invariance by dilatation and permutations).

A.2.4 Second case: $\mathrm{rk}(W) < K$

This case arises even when all generators are extreme rays, for example in three dimensional space when the simplicial cone section (for example section by a plan orthogonal to first diagonal) is a convex polygon with more than three vertices.

In that case two extreme points of view can be taken:

- searching a simplicial cone minimal regarding inclusion, which in the 3D case means which section has minimal area, i.e. finding the convex hull of the section of the *projection* (in the conic sense) of the data on some plane,
- searching a simplicial cone with the minimum number of generators, which leads to chose the rank of W as the number of generators, thus in our example choosing a cone *too big* since the whole first octant is chosen.

Appendix B

Datasets

B.1 The Acorns Caregiver dataset

The Caregiver dataset (**?**) provided by the ACORNS project, is composed of 1000 utterances containing 13 keywords, each spoken by 4 speakers in English adult directed speech; this makes a total of 4000 utterances. An example of sentences used in the dataset is *Angus is lazy today.* where the semantic tag/keyword is *Angus*. Examples of transcriptions from utterances from the dataset are given in table B.1.

We take a **bath**
To put it in the **bath** isn't funny either
The **shoe** is a symbol
Now **mummy** is losing her patience
Daddy comes closer
Angus takes off her shoe
Daddy never calls
She sits on a **nappy**
Now everybody is in the **car**
Where is the **nappy**

Table B.1: Transcriptions from ten random examples from the Acorns Caregiver dataset from **?**. Keywords are identified in bold font.

B.2 The first choreography dataset

The *first choreography dataset* contains choreography motions recorded through a kinect device. These motions have a combinatorial structure: from a given set of primitive dance motions, choreographies are constructed as simultaneous execution of some of these primitive motions. The data is publicly available at `http://flowers.inria.fr/choreography_database.html`.

Primitive dance motions are chosen from a total set of 48 motions and are spanned

over one or two limbs, either the legs (for example walk, *squat*), left or right arm (for example *wave hand*, *punch*) or both arms (for example *clap in hands*, *paddle*).

Complex choreographies are produced as the simultaneous demonstration of two or three of these primitive motion: either one for legs and one for both arm, or one for legs and one for each arm. Each example (or record) contained in the dataset consists in two elements: the motion data and labels identifying which primitive motions are combined to produce the choreography.

The dataset actually contains three separate sets of examples:

1. **primitive**: in each example, only one primitive motion is demonstrated, the set of labels associated to each example is thus a singleton (326 examples).

2. **mixed small**: demonstrations of complex choreographies composed of primitive motions taken in a subset of 16 possible motions (137 examples).

3. **mixed full**: demonstrations of complex choreographies composed of primitive motions taken in all the possible motions (277 examples).

B.2.1 Description of the data

The data has been acquired through a kinect camera and the OpenNI drivers[1], which yields a stream of values of markers on the body. Each example from the dataset is associated to a sequence of 3D positions of each of the 24 markers. Thus for a sequence of length T, the example would corresponds to $T * 24 * 3$ values.

The kinect device recognizes and provides positions of the following list of markers: *head*, *neck*, *waist*, *left_hip*, *left_shoulder*, *left_elbow*, *left_hand*, *left_knee*, *left_foot*, *left_collar*, *left_wrist*, *left_fingertip*, *left_ankle*, *right_hip*, *right_shoulder*, *right_elbow*, *right_hand*, *right_knee*, *right_foot*, *right_collar*, *right_wrist*, *right_hand*, *right_fingertip*, *right_ankle*.

These markers are however not tracked with the same accuracy and it might be better to filter to keep only a subset of these markers. In the experiments from chapter 3 we use: *head*, *neck*, *left_hip*, *left_shoulder*, *left_elbow*, *left_hand*, *left_knee*, *left_foot*, *right_hip*, *right_shoulder*, *right_elbow*, *right_hand*, *right_knee*, *right_hand*, *right_foot*.

B.3 The second choreography dataset

The *second choreography dataset* contains choreography motions recorded through a kinect device. It contains a total of 1100 examples of 10 different gestures that are spanned over one or two limbs. The data is publicly available at `http://flowers.inria.fr/choreo2`.

[1] `http://openni.org`

Id	Limb(s)	Description
1	right arm	hold horizontal
2	right arm	hold vertical (down)
3	right arm	hold vertical (up)
4	right arm	from horizontal on side, bend over the head
5	right arm	raise from horizontal to vertical
6	right arm	lower from horizontal to vertical
7	right arm	from horizontal side, bend in front of the torso
8	right arm	from horizontal side, bent elbow to get vertical forearm toward up
9	right arm	mimic punching
10	right arm	hold horizontal and bring from side to front
11	right arm	from horizontal side, bend elbow to get vertical forearm toward down
12	right arm	from horizontal side, bring hand to shoulder (elbow moving vertically)
13	right arm	hold horizontal and bring from right side to left side
14	right arm	swing forearm downside with horizontal upper arm
15	right arm	draw circles with arm extended on the right
16	right arm	wave motion of the arm held, horizontal on the side
17	right arm	wave hand (shoulder level)
18	right arm	wave hand (over the head)
19	both arms	clap hands (at varying positions)
20	both arms	mimic paddling on the left
21	both arms	mimic paddling on the right
22	both arms	mimic pushing on ski sticks
23	legs	un-squat
24	legs	mimic walking
25	legs	stay still
26	legs	step on the right
27	legs	step on the left
28	right leg	raise and bend leg to form a flag (or 'P') shape
29	left leg	raise and bend leg to form a flag (or 'P') shape
30	left arm	hold horizontal
31	left arm	hold vertical (down)
32	left arm	hold vertical (up)
33	left arm	from horizontal on side, bend over the head
34	left arm	raise from horizontal to vertical
35	left arm	lower from horizontal to vertical
36	left arm	from horizontal side, bend in front of the torso
37	left arm	from horizontal side, bent elbow to get vertical forearm toward up
38	left arm	mimic punching
39	left arm	hold horizontal and bring from side to front
40	left arm	from horizontal side, bend elbow to get vertical forearm toward down
41	left arm	from horizontal side, bring hand to shoulder (elbow moving vertically)
42	left arm	hold horizontal and bring from left side to right side
43	left arm	swing forearm downside with horizontal upper arm
44	left arm	draw circles with arm extended on the left
45	left arm	wave motion of the arm held, horizontal on the side
46	left arm	wave hand (shoulder level)
47	left arm	wave hand (over the head)

Table B.2: List of gestures composing the motions of the first choreography dataset. The *small dataset* only uses the following subset of labels: $1, 5, 6, 10, 19, 20, 21, 22, 23, 24, 25, 28, 30, 38, 40, 43$.

Id	Limb(s)	Description
1	both legs	squat
2	both legs	walk
3	right leg	make a flag/P on right leg
4	both arms	clap
5	both arms	mimic paddling left
6	right arm	mimic punching with right arm
7	right arm	right arm horizontal goes from side to front
8	left arm	horizontal left arm, forearm goes down to form a square angle
9	left arm	make waves on left arm
10	left arm	say hello with left arm

Table B.3: List of gestures composing the motions of the second choreography dataset.

B.3.1 Description of the data

The data has been acquired through a kinect camera and the OpenNI drivers through its ROS[2] interface, which yields a stream of values of markers on the body. Each example from the dataset is associated to a sequence of 3D positions of each of the 15 markers. Thus for a sequence of length T, the example would corresponds to $T * 15 * 7$ values. The 7 successive values for each marker are there 3D coordinates together with a representation of the rotation of the frame between previous and next segment. The rotation is encoded in quaternion representation as described on the ROS time frame page[3].

The position of the following list of markers was recorded: *head*, *neck*, *left_hip*, *left_hip*, *left_shoulder*, *left_elbow*, *left_hand*, *left_knee*, *left_foot*, *right_hip*, *right_shoulder*, *right_elbow*, *right_hand*, *right_knee*, *right_foot*, *right_hand*.

[2]Robotic operating system, `http://ros.org`
[3]`http://www.ros.org/wiki/tf`

Appendix C

Code

The code used in the experiments from **?** is available publicly on `http://github.com/omangin/multimodal`. It consists in a set of tools and experimental scripts used to achieve multimodal learning with nonnegative matrix factorization (NMF). This code is distributed under the new BSD license.

Bibliography

P. Abbeel, A. Coates, and A. Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13): 1608–1639, June 2010. ISSN 0278-3649. doi: 10.1177/0278364910371999.

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International conference on Machine learning*, 2004.

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: Design of dictionaries for sparse representation. In *Proceedings of SPARS*, number 5, pages 9–12, 2005.

Nasir Ahmed, T Natarajan, and Kamisetty R Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 100(1):90–93, 1974.

Guillaume Aimetti. Modelling early language acquisition skills: Towards a general statistical learning mechanism. In *EACL (Student Research Workshop)*, pages 1–9, 2009.

S. Akaho, Y. Kiuchi, and S. Umeyama. Mica: multimodal independent component analysis. In *International Joint Conference on Neural Networks, (IJCNN 99)*, volume 2, pages 927–932, 1999. doi: 10.1109/IJCNN.1999.831077.

Zeynep Akata, Christian Thurau, and Christian Bauckhage. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In *Computer Vision Winter Workshop*, number 16, Mitterberg, Autriche, 2011.

Nameera Akhtar and Lisa Montague. Early lexical acquisition: The role of cross-situational learning. *First Language*, 19(57):347–358, 1999.

Javier Almingol, Luis Montesano, and Manuel Lopes. Learning multiple behaviors from unlabeled demonstrations in a latent controller space. In *International conference on Machine learning (ICML)*, 2013.

Toomas Altosaar, Louis ten Bosch, Guillaume Aimetti, Christos Koniaris, Kris Demuynck, Henk van den Heuvel, Signal Proc, P O Box, and Fi Tkk. A speech corpus for modeling language acquisition: Caregiver. In *Language Resources and Evaluation - LREC*, pages 1062–1068, 2008.

Pierre Andry, Philippe Gaussier, Sorin Moga, Jean-Paul Banquet, and Jacqueline Nadel. Learning and communication via imitation: An autonomous robot perspective. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(5):431–442, 2001.

Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. Cognitive developmental robotics: A survey. *Autonomous Mental Development, IEEE Transactions on*, 1(1):12–34, 2009.

Monica Babes-Vroman, Vukosi Marivate, Kaushik Subramanian, and Michael Littman. Apprenticeship learning about multiple intentions. In *International conference on Machine learning (ICML)*, number 28, 2011.

Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, January 2013. doi: 10.1016/j.robot.2012.05.008.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957a.

Richard Bellman. A markovian decision process. *Indiana University Mathematics Journal*, 6:679–684, 1957b. ISSN 0022-2518.

Tony Belpaeme and Anthony Morse. Word and category learning in a continuous semantic domain: Comparing cross-situational and interactive learning. *Advances in Complex Systems*, 15(03n04), 2012. doi: 10.1142/S021952591200312.

Jaafar BenAbdallah, Juan C. Caicedo, Fabio A. Gonzalez, and Olfa Nasraoui. Multimodal image annotation using non-negative matrix factorization. In *International Conference on Web Intelligence and Intelligent Agent Technology*, pages 128–135. IEEE / WIC / ACM, August 2010. ISBN 978-1-4244-8482-9. doi: 10.1109/WI-IAT.2010.293.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

Randolph Blake. A neural theory of binocular rivalry. *Psychological review*, 96(1): 145–167, 1989.

Sofiane Boucenna, Philippe Gaussier, Pierre Andry, and Laurence Hafemeister. Imitation as a communication tool for online facial expression learning and recognition. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5323–5328. IEEE, 2010.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. ISBN 978-0521833783.

M. Boyle. Notes on the perron-frobenius theory of nonnegative matrices. Technical report.

Holger Brandl, Frank Joublin, and Christian Goerick. Towards unsupervised online word clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 5073–5076, Las Vegas, NV, 2008.

Michael R. Brent. Speech segmentation and word discovery: A computational perspective, 1999.

David R. Brillinger. Learning a potential function from a trajectory. *IEEE Signal Processing Letters*, 14(11):867–870, November 2007. ISSN 1070-9908. doi: 10.1109/LSP.2007.900032.

Rodney Brooks. Intelligence without representation. *Artificial Intelligence*, 47: 139–159, 1991.

Jesse Butterfield, Sarah Osentoski, Graylin Jay, and Odest Chadwicke Jenkins. Learning from demonstration using a multi-valued function regressor for time-series data. In *International Conference on Humanoid Robots*, number 10, Nashville, 2010. IEEE Comput. Soc. Press.

Richard W. Byrne and Anne E. Russon. Learning by imitation: a hierarchical approach. *Behavioral and brain sciences*, 21(5):667–721, 1998.

Maya Cakmak and Manuel Lopes. Algorithmic and human teaching of sequential decision tasks. In *AAAI Conference on Artificial Intelligence (AAAI-12)*, 2012.

S. Calinon and A. Billard. A probabilistic programming by demonstration framework handling skill constraints in joint space and task space. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, September 2008.

Sylvain Calinon and Aude G Billard. Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics*, 23:2059–2076, 2009.

Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 37(2):286–98, April 2007. ISSN 1083-4419. doi: 10.1109/TSMCB.2006.886952.

Sylvain Calinon, Florent D'Halluin, Eric L Sauser, Darwin G Caldwell, and Aude G Billard. Learning and reproduction of gestures by imitation: An approach based on hidden markov model and gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17(2):44–54, 2010.

Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Francesco Nori, Luciano Fadiga, Britta Wrede, Katharina Rohlfing, Elio Tuci, Kerstin Dautenhahn, Joe Saunders, and Arne Zeschel. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3):167–195, September 2010. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2053034.

Luigi Cattaneo and Giacomo Rizzolatti. The mirror neuron system. *Archives of Neurology*, 66(5):557, 2009.

Thomas Cederborg and Pierre-Yves Oudeyer. From language to motor gavagai: Unified imitation learning of multiple linguistic and non-linguistic sensorimotor skills. *Transactions on Autonomous Mental Development (TAMD)*, 5, 2013.

E. Colin Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25:975, 1953. doi: 10.1121/1.1907229.

Jaedeug Choi and Kee-Eung Kim. Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems 25*, pages 314–322, 2012.

Thomas M Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 1991.

Christian Daniel, Gerhard Neumann, and Jan Peters. Learning concurrent motor skills in versatile solution spaces. In *Proceedings of the International Conference on Robot Systems (IROS)*, 2012.

John Demiris and Gillian Hayes. Do robots ape. In *AAAI Fall Symposium*, pages 28–30. American Association for Artificial Intelligence, November 1997.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

Chris Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, January 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2008.277.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts ? *Advances in neural information processing systems*, 16, 2003.

Joris Driesen. *Discovering words in speech using matrix factorization*. PhD thesis, KU Leuven, ESAT, 2012.

Joris Driesen, Louis ten Bosch, and Hugo Van Hamme. Adaptive non-negative matrix factorization in a computational model of language acquisition. In *Interspeech*, pages 1–4, 2009.

Joris Driesen, Hugo Van Hamme, and Bastiaan W. Kleijn. Learning from images and speech with non-negative matrix factorization enhanced by input space scaling. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1–6, Berkeley, California, USA, 2010. IEEE. doi: http://dx.doi.org/10.1109/SLT.2010.5700813.

Joris Driesen, Jort F. Gemmeke, and Hugo Van Hamme. Data-driven speech representations for nmf-based word learning. In *SAPA-SCALE*, Portland, USA, 2012.

Staffan Ekvall and Danica Kragic. Learning task models from multiple human demonstrations. *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 358–363, September 2006. doi: 10.1109/ROMAN.2006.314460.

Daniel P. W. Ellis. Plp and rasta (and mfcc, and inversion) in matlab, 2005.

K. Farrell, R.J. Mammone, and A.L. Gorin. Adaptive language acquisition using incremental learning. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1993)*, volume 1, pages 501–504. IEEE, 1993.

David Filliat. Interactive learning of visual topological navigation. In *International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 248–254. IEEE, 2008.

Sadaoki Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):52–59, 1986.

Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 29(9):2421–2456, 2011. doi: http://dx.doi.org/10.1162/NECO_a_00168.

Cédric Févotte, Jonathan Le Roux, and John R. Hershey. Non-negative dynamical system with application to speech and audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013.

Mrugesh R. Gajjar, R. Govindarajan, and T. V. Sreenivas. Online unsupervised pattern discovery in speech using parallelization. In *InterSpeech*, pages 2458–2461. ISCA, 2008.

Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in Neural Information Processing Systems 6*, pages 120–127. Morgan Kaufmann, 1994.

James Jerome Gibson. *The Ecological Approach To Visual Perception*. Taylor & Francis, 1979. ISBN 9781135059729.

Arthur M Glenberg and Michael P Kaschak. Grounding language in action. *Psychonomic bulletin & review*, 9(3):558–65, September 2002. ISSN 1069-9384.

Rebecca L. Gomez and Lou-Ann Gerken. Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition*, 70(2):109–135, 1999.

A L Gorin, D Petrovska-Delacretaz, J Wright, and G Riccardi. Learning spoken language without transcription, 1999.

Allen L. Gorin, Stephen E. Levinson, and Ananth Sankar. An experiment in spoken language acquisition. *Speech and Audio Processing, IEEE Transactions on*, 2(1): 224–240, January 1994. ISSN 1063-6676. doi: 10.1109/89.260365.

Jacqueline Gottlieb, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 17(11):585–593, 2013.

Herbert Paul Grice. *Studies in the Way of Words*. Harvard University Press, 1989.

Daniel H Grollman and Odest Chadwicke Jenkins. Incremental learning of subtasks from unsegmented demonstration. In *IROS*, Taipei, Taiwan, 2010.

Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.

Stephen Hart and Roderic Grupen. Learning generalizable control programs. *Autonomous Mental Development, IEEE Transactions on*, 3(3):216–231, 2011.

Stephen Hart, Shijaj Sen, and Roderic A. Grupen. Intrinsically motivated hierarchical manipulation. In *International Conference on Robotics and Automation (ICRA)*, pages 3814–3819, Pasadena, California, USA, 2008.

Trevor J. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, second edi edition, 2001.

Sven Hellbach, Julian P Eggert, Edgar Körner, and Horst-michael Gross. Basis decomposition of motion trajectories using spatio-temporal nmf. In *Int. Conf. on Artificial Neural Networks (ICANN)*, pages 597–606, Limassol, Cyprus, 2009. Springer.

Ronald A. Howard. *Dynamic Programming and Markov Processes*. Published jointly by the Technology Press of the Massachusetts Institute of Technology and Wiley and Sons, 1960.

Patrik O Hoyer. Non-negative sparse coding. Technical report, February 2002.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in neural information processing systems*, pages 1547–1554, Cambridge, 2003. MIT Press.

Naoto Iwahashi. Language acquisition through a human–robot interface by combining speech, visual, and behavioral information. *Information Sciences*, 156(1):109–121, 2003.

Naoto Iwahashi. Active and unsupervised learning for spoken word acquisition through a multimodal inteface. In *IEE International Workshop on Robot and Human Interactive Communication*, 2004.

Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

Nikolay Jetchev and Marc Toussaint. Task space retrieval using inverse feedback control. In Lise Getoor and Tobias Scheffer, editors, *International Conference on Machine Learning*, number 28 in ICML '11, pages 449–456, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, UniversUniversität Dortmund, LS VIII-Reportität Do, 1997.

Frédéric Kaplan and Pierre-Yves Oudeyer. The progress-drive hypothesis: an interpretation of early imitation. *Models and mechanims of imitation and social learning: Behavioural, social and communication dimensions*, pages 361–377, 2007.

Juha Karhunen and Tomas Ukkonen. Extending ica for finding jointly dependent components from two related data sets. *Neurocomputing*, 70:2969–2979, 2007. ISSN 0925-2312. doi: 10.1016/j.neucom.2006.10.144.

S Mohammad Khansari-Zadeh and Aude Billard. Learning stable non-linear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics2*, 27 (5):943–957, 2011.

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems*, pages 849–856, Red Hook, NY, USA, 2009. Curran.

Jürgen Konczak. On the notion of motor primitives in humans and robots. In *International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, volume 123, pages 47–53. Lund University Cognitive Studies, 2005. ISBN 91-974741-4-2.

V Kruger, Danica Kragic, Aleš Ude, and Christopher Geib. The meaning of action: a review on action recognition and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007.

Volker Kruger, Dennis Herzog, Sanmohan Baby, Ales Ude, and Danica Kragic. Learning actions from observations. *Robotics and Automation Magazine*, 17(2): 30–43, 2010.

Patricia K Kuhl. Early language acquisition: cracking the speech code. *Nature reviews. Neuroscience*, 5(11):831–43, November 2004. ISSN 1471-003X. doi: 10.1038/nrn1533.

Patricia K. Kuhl. Brain mechanisms in early language acquisition. *Neuron*, 67(5): 713–727, 2010.

Patricia K. Kuhl, Karen A. Williams, Francisco Lacerda, Kenneth N. Stevens, and Björn Lindblom. Linguistic experience alters phonetic perception in infants by 6 months of age. *Science*, 255(5044):606–608, 1992.

D Kulic, Hirotaka Imagawa, and Yoshihiko Nakamura. Online acquisition and visualization of motion primitives for humanoid robots. *Symposium on Robot and Human*, pages 1210–1215, 2009.

Dana Kulic and Yoshihiko Nakamura. Incremental learning of human behaviors using hierarchical hidden markov models. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4649–4655. IEEE Comput. Soc. Press, 2010.

Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *Robotics and Automation, IEEE Transactions on*, 10(6):799–822, 1994.

Barbara Landau, Linda Smith, and Susan Jones. Object perception and object naming in early development. *Trends in cognitive sciences*, 2(1):19–24, 1998.

Daniel D. Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91, October 1999. ISSN 0028-0836. doi: 10.1038/44565.

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562. MIT Press, 2001.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, number 19, 2006.

Augustin Lefèvre, Francis R. Bach, and Cédric Févotte. Itakura-saito nonnegative matrix factorization with group sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, number 1, pages 21–24. IEEE, 2011.

David A. Leopold and Nikos K. Logothetis. Multistable phenomena: changing views in perception. *Trends in cognitive sciences*, 3(7):254–264, 1999. doi: 10.1016/S1364-6613(99)01332-7.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

Sergey Levine, Zoran Popovic, and Vladlen Koltun. Feature construction for inverse reinforcement learning. *Advances in Neural Information Processing Systems*, (24): 1–9, 2010.

Yi Li, Cornelia Fermuller, Yiannis Aloimonos, and Hui Ji. Learning shift-invariant sparse representation of actions. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2630–2637, San-Francisco, June 2010. IEEE. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539977.

Rainer Lienhart, Stefan Romberg, and Eva Hörster. Multilayer plsa for multimodal image retrieval. In *International Conference on Image and Video Retrieval -CIVR*, number April, page 1, New York, New York, USA, 2009. ACM Press. ISBN 9781605584805. doi: 10.1145/1646396.1646408.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–79, October 2007. ISSN 0899-7667. doi: 10.1162/neco.2007.19.10.2756.

Manuel Lopes, Francisco S. Melo, Ben Kenward, and José Santos-Victor. A computational model of social-learning mechanisms. *Adaptive Behavior*, 17(6):467–483, 2009a.

Manuel Lopes, Francisco S. Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *European Conference on Machine Learning (ECML)*, 2009b.

Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

Natalia Lyubova and David Filliat. Developmental approach for interactive object discovery. In *Proceedings of the 2012 IJCNN International Joint Conference on Neural Networks*, 2012.

Bin Ma and Haizhou Li. Spoken language identification using bag-of-sounds. In *International Conference on Computational Cybernetics*. IEEE, 2005.

Julien Mairal, Francis Bach, Jean Pnce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning (ICML)*, number 26, pages 689–696, 2009.

Olivier Mangin and Pierre-Yves Oudeyer. Learning to recognize parallel combinations of human motion primitives with linguistic descriptions using non-negative matrix factorization. In *International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Algarve (Portugal), 2012a. IEEE/RSJ.

Olivier Mangin and Pierre-Yves Oudeyer. Learning the combinatorial structure of demonstrated behaviors with inverse feedback control. In *International Workshop on Human Behavior Understanding*, number 3, Vilamoura, Algarve (Portugal), 2012b.

Olivier Mangin and Pierre-Yves Oudeyer. Learning semantic components from subsymbolic multimodal perception. In *the Joint IEEE International Conference on Development and Learning an on Epigenetic Robotics (ICDL-EpiRob)*, number 3, August 2013.

Olivier Mangin, David Filliat, and Pierre-Yves Oudeyer. A bag-of-features framework for incremental learning of speech invariants in unsegmented audio streams. In Birger Johansson, Erol Sahin, and Christian Balkenius, editors, *Tenth International Conference on Epigenetic Robotics*, pages 73–80, Öorenäas Slott, Sweden, 2010.

Ellen M. Markman. Constraints children place on word meanings. *Cognitive Science*, 14:57–77, 1990.

Gianluca Massera, Elio Tuci, Tomassino Ferrauto, and Stefano Nolfi. The facilitatory role of linguistic instructions on developing manipulation skills. *IEEE Computational Intelligence Magazine*, 5(3):33–42, 2010.

Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264 (5588):746–748, December 1976. ISSN 0028-0836. doi: 10.1038/264746a0.

C.D. Meyer. *Matrix analysis and applied linear algebra: solutions manual*. Society for Industrial and Applied Mathematics, 2000. ISBN 0-89871-454-0.

Bernard Michini and Jonathan P How. Bayesian nonparametric inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2012.

Benjamin Franklin Miessner. The electric dog. *Scientific American Supplement*, (2267):376–377, June 14th 1919.

Clément Moulin-Frier. *Rôle des relations perception-action dans la communication parlée et l'émergence des systèmes phonologiques : étude, modélisation computationnelle et simulations*. PhD thesis, Université Pierre Mendes-France, Grenoble, 2011.

Ferdinando A. Mussa-Ivaldi and Emilio Bizzi. Motor learning through the combination of primitives. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 355(1404):1755–69, December 2000. ISSN 0962-8436. doi: 10.1098/rstb.2000.0733.

Jacqueline Nadel. *Imitation et communication entre jeunes enfants*, volume 13. Presses Universitaires de France - PUF, 1986.

Yukie Nagai. Joint attention learning based on early detection of self-other motion equivalence with population codes. *Journal of the Robotics Society of Japan*, 25 (5):77, 2007.

C Nehaniv and K Dautenhahn. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. *World Scientific Series in Robotics and*, pages 1–26, 2000.

Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, number 23, pages 295–302, Vancouver, Canada, 2007. AUAI Press, Corvallis, Oregon. ISBN 0-9749039-3-00-9749039-3-0.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. *International Conference on Machine Learning*, 2000.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, number 16, pages 278–287, 1999.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, and Andrew Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning*, number 28, Bellevue, Washington, USA, 2011.

Monica N. Nicolescu and Maja J. Matarić. A hierarchical architecture for behavior-based robots. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02*, page 227, 2002. doi: 10.1145/544741.544798.

Monica N. Nicolescu and Maja J. Matarić. Natural methods for learning and generalization in human-robot domains. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: http://dx.doi.org/10.1109/CVPR.2006.264.

Francesco Nori and Ruggero Frezza. Biologically inspired control of a kinematic chain using the superposition of motion primitives. In *Decision and Control (CDC)*, volume 1, pages 1075–1080. IEEE Comput. Soc. Press, 2004a.

Franceso Nori and Ruggero Frezza. Nonlinear control by a finite set of motion primitives. In *Nolcos*, 2004b.

Pierre-Yves Oudeyer. On the impact of robotics in behavioral and cognitive sciences: From insect navigation to human cognitive development. *IEEE Transactions on Autonomous Mental Development*, 2(1):2–16, March 2010. ISSN 1943-0604. doi: 10.1109/TAMD.2009.2039057.

P Paatero and U Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994.

Michael Pardowitz, Steffen Knoop, Ruediger Dillmann, and Raoul D Zöllner. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 37(2): 322–32, April 2007. ISSN 1083-4419.

Alex S. Park and James R. Glass. Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech and Language Processing*, 16(1):186–197, 2008.

Jean Piaget. *La construction du réel chez l'enfant.* Delachaux & Niestlé, 1937.

RJ Plemmons and RE Cline. The generalized inverse of a nonnegative matrix. *Proc. Amer. Math. Soc*, 31(1):46–50, 1972.

Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

Willard Van Orman Quine. *Word and object*. MIT press, 1960.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI'07)*, number 20, pages 2586–2591, 2007.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum margin planning. In *International conference on Machine learning - ICML '06*, number 23, pages 729–736, New York, New York, USA, 2006. ACM Press. ISBN 1595933832. doi: 10.1145/1143844.1143936.

Constantin A. Rothkopf and Dana H. Ballard. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics*, pages 1–14, 2013.

Alice C. Roy and Michael A. Arbib. The syntactic motor system. *Gesture*, 5(1):7–37, January 2005. ISSN 15681475. doi: 10.1075/gest.5.1.03roy.

Deb Roy. *Learning from Sights and Sounds: A Computational Model*. PhD thesis, Massachussetts Institute of Technology, September 1999.

Deb K. Roy and Alex P. Pentland. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146, January 2002. ISSN 03640213. doi: 10.1207/s15516709cog2601_4.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "the earth mover's distance as a metric for image retrieval": None. *International Journal of Computer Vision*, 40 (2):99–121, 2000.

Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103. ACM, 1998.

Kate Saenko and Trevor Darrell. Object category recognition using probabilistic fusion of speech and image classifiers. In *Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms - MLMI*, number 4, Brno, Czech Republic, 2007.

Jenny R. Saffran and Diana P. Wilson. From syllables to syntax: Multilevel statistical learning by 12-month-old infants. *Infancy*, 4(2):273–284, April 2003. ISSN 15250008. doi: 10.1207/S15327078IN0402_07.

Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055.

Larissa K. Samuelson, Linda B. Smith, Lynn K. Perry, and John P. Spencer. Grounding word learning in space. *PloS one*, 6(12):e28095, 2011.

Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive Motion of Animals and Machines*, pages 261–280, Los Angeles, 2006. Springer.

Jean-Luc Schwartz. A reanalysis of mcgurk data suggests that audiovisual fusion in speech perception is subject-dependent. *The Journal of the Acoustical Society of America*, 127:1584–1594, 2010. doi: 10.1121/1.3293001.

Jean-Luc Schwartz, Frédéric Berthommier, and Christophe Savariaux. Seeing to hear better: evidence for early audio-visual interactions in speech identification. *Cognition*, 93(2):B69–B78, 2004.

Jean-Luc Schwartz, Anahita Basirat, Lucie Ménard, and Marc Sato. The perception-for-action-control theory (pact): A perceptuo-motor theory of speech perception. *Journal of Neurolinguistics*, 25(5):336–354, 2012a. ISSN 0911-6044. doi: 10.1016/j.jneuroling.2009.12.004.

Jean-Luc Schwartz, Nicolas Grimault, Jean-Michel Hupé, Brian C.J. Moore, and Daniel Pressnitzer. Multistability in perception: binding sensory modalities, an overview. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1591):896–905, 2012b.

Jihene Serkhane, Jean-Luc Schwartz, and Pierre Bessiere. Building a talking baby robot: A contribution to the study of speech acquisition and evolution. *Interaction studies*, 6(2):253–286, 2005.

Ajit P. Singh and Geoffrey J. Gordon. A unified view of matrix factorization models. In *ECML PKDD*, pages 358–373, 2008.

Josef Sivic and Andrew Zisserman. Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566, 2008.

Linda Smith and Chen Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106(3):1558–1568, 2008.

David Sodoyer, Laurent Girin, Christian Jutten, and Jean-Luc Schwartz. Developing an audio-visual speech source separation algorithm. *Speech Communication*, 44(1):113–125, 2004.

Luc Steels. *The Talking Heads Experiment*, volume 1. Laboratorium, 1999.

Luc Steels. The symbol grounding problem has been solved. so what's next? In Manuel de Vega, Arthur M. Glenberg, and Arthur C. Graesser, editors, *Symbols and Embodiment: Debates on Meaning and Cognition*, chapter 12. Oxford University Press, Oxford, 2008.

Luc Steels and Frederic Kaplan. Bootstrapping grounded word semantics. In Ted Briscoe, editor, *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, chapter 3. Cambridge University Press, 2002.

Stanley S. Stevens and John Volkmann. The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, 53(3):329–353, 1940.

Veronique Stouten, Kris Demuynck, et al. Discovering phone patterns in spoken utterances by non-negative matrix factorization. *Signal Processing Letters, IEEE*, 15:131–134, 2008.

Freek Stulp and Stefan Schaal. Hierarchical reinforcement learning with movement primitives. *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 231–238, October 2011. doi: 10.1109/Humanoids.2011.6100841.

Y. Sugita and J. Tani. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, 13(1):33, 2005.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. ISBN 978-0-262-19398-6.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.

Graham W. Taylor and Geoffrey E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *International Conference on Machine Learning (ICML 2009)*, number 26, pages 1–8, New York, New York, USA, 2009a. ACM Press. ISBN 9781605585161. doi: 10.1145/1553374.1553505.

Graham W. Taylor and Geoffrey E. Hinton. Products of hidden markov models : It takes n ¿ 1 to tango. In *Twenty-fifth Conference on Uncertainty in Artificial Intelligence*, pages 522–529. AUAI Press, 2009b.

Graham W. Taylor, Geoffrey E. Hinton, and Sam Roweis. Modeling human motion using binary latent variables. *Advances in neural information processing systems (NIPS)*, 19:1345–1352, 2006. ISSN 1049-5258.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains : A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

Louis ten Bosch, Lou Boves, Hugo Van Hamme, and Roger K. Moore. A computational model of language acquisition: the emergence of words. *Fundamenta Informaticae*, 90(3):229–249, 2009.

Louis F.M. ten Bosch, Hugo Van Hamme, and Lou W.J. Boves. Unsupervised detection of words questioning the relevance of segmentation. In *Speech Analysis and Processing for Knowledge Discovery*, ITRW ISCA. Bonn, Germany : ISCA, 2008.

Edward L. Thorndike. *Animal intelligence; experimental studies*. New York, The Macmillan company, 1911.

Sebastian Thrun and Tom M. Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1):25–46, 1995.

Michael Tomasello. *Origins of human communication*. MIT press Cambridge, 2008.

Matthew C Tresch and Anthony Jarc. The case for and against muscle synergies. *Current opinion in neurobiology*, 19(6):601–7, December 2009. ISSN 1873-6882. doi: 10.1016/j.conb.2009.09.002.

Elio Tuci, Tomassino Ferrauto, Arne Zeschel, Gianluca Massera, and Stefano Nolfi. An experiment on behaviour generalisation and the emergence of linguistic compositionality in evolving robots. *IEEE Transactions on Autonomous Mental Development*, 3(2):1–14, 2011.

Alan Mathison Turing. Computing machinery and intelligence. *Mind*, 59(236): 433–460, 1950.

Hugo Van Hamme. Hac-models: a novel approach to continuous speech recognition. In *Interspeech ISCA*, pages 2554–2557, 2008.

Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, Ida Stockman, Mriganka Sur, and Esther Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.

Andrew Whiten and R. Ham. On the nature and evolution of imitation in the animal kingdom: reappraisal of a century of research. *Advances in the Study of Behavior*, 21:239–283, 1992.

Matthew M. Williamson. Postural primitives: Interactive behavior for a humanoid robot arm. In *International Conference on Simulation of Adaptive Behavior*, number 4, pages 124–131, 1996.

Britta Wrede, Katharina Rohlfing, Jochen Steil, Sebastian Wrede, Pierre-Yves Oudeyer, and Jun Tani. Towards robots with teleological action and language understanding. In Emre Ugur, Yukie Nagai, Erhan Oztop, and Minoru Asada, editors, *Humanoids 2012 Workshop on Developmental Robotics: Can developmental robotics yield human-like cognitive abilities?*, Osaka, Japon, November 2012.

Chen Yu and Dana H Ballard. A multimodal learning interface for grounding spoken language in sensory perceptions. *Transactions on Applied Perception*, (1):57–80, 2004.

Chen Yu and Dana H. Ballard. A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, 70(13-15):2149–2165, August 2007. ISSN 09252312. doi: 10.1016/j.neucom.2006.01.034.

Daniel Yurovsky, Chen Yu, and Linda B. Smith. Statistical speech segmentation and word learning in parallel: scaffolding from child-directed speech. *Frontiers in psychology*, 3, 2012.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAI Conference on Artificial Intelligence*, number 23, pages 1433–1438, 2008.