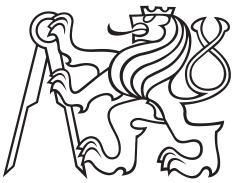
Doctoral Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Hierarchical probabilistic model of language acquistion

Mgr. Karla Štěpánová

Supervisor: Mgr. Michal Vavrečka, Ph.D.
Supervisor–specialist: Doc. Ing. Lenka Lhotská, CSc.
Ph.D. Programme: Electrical Engineering and Information Technology
Branch of study: Artificial Intelligence and Biocybernetics
Prague, September 2016

Acknowledgements

On the first place, I would like to thank my supervisor, Mgr. Michal Vavrečka, PhD., for his patience, encouragement and valuable suggestions he has provided throughout my work on the thesis. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions so promptly. I would also like to thank doc. Ing. Lenka Lhotská, CSc. for providing me very pleasant working environment and giving me feedback on my work, and Prof. Ing. Václav Hlaváč, CSc. for his proofreading of my thesis which lead to its significant improvement. My great thanks belong to people of CRNS lab in Plymouth, headed by Prof. Angelo Cangelosi, for their personal and technical support while I was dealing with implementation of my architecture into physical and simulated iCub robot. Their friendship is irreplaceable and it made from the time spent with work something very pleasant which I will always recall with a smile. Namely, I would like to thank Dr. Michael Klein for his suggestions on language models, Alexander Antunes for help with iCub robot, and last but not least Frederico Belmonte Klein for his critical but supportive approach, long discussions covering potential applications of my research and also for an initial idea of sequential mapping. Finally, I would like to express my gratitude to my family, especialy to my two small daughters, who have been the best motivation and support during my PhD studies and who were always able to make me laugh even when I felt desperate. I also owe them thanks for all those ideas which I got when observing their cognitive development.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all used information sources in accordance with Methodical instruction about ethical principles in the preparation of university theses.

In Prague on 19. September 2016

Karla Štěpánová

Abstract

In this thesis, I propose an unsupervised computational model of language acquisition through visual grounding. I especially focus on a case where the language input is in a form of variable length sentences. The state-of-the-art cognitive architectures with the focus on grounding language in vision are explored. I take an advantage of probabilistic Bayesian models which are besides neural networks one of the main tools used in a computational cognitive modeling. The probabilistic (Bayesian) models have been used in the tasks such as language processing, decision making or causality learning. In the first part of the thesis newly proposed method for estimating a number of clusters in data is described. In the second part of the thesis I focus on the description of the cognitive architecture itself. The developed hierarchical cognitive architecture processes separately visual (static) and language (time-sequence) data and combines them in a multimodal layer. The important feature is a compositionality of the system - ability to derive meaning of previously unheard sentences and unseen objects and its ability to learn all features describing the object from sentences of variable length. The proposed architecture was implemented into the humanoid robot iCub and tested on both artificially generated data and on the real-world data.

Keywords: unsupervised learning, language acquisition, symbol grounding, Gaussian mixture model, probabilitic Bayesian model

Supervisor: Mgr. Michal Vavrečka, Ph.D.
Czech Institut of Informatics, Robotics and Cybernetics,
Zikova 1903/4,
166 36 Praha 6

Abstrakt

V této disertační práci se zabývám návrhem výpočetního modelu osvojování jazyka skrz ukotvení symbolů ve vizuálních vjemech. Speciálně se zaměřuji na případ, kdy je jazykový vstup ve formě vět variabilní délky. V rámci práce byla provedena rešerše recentních kognitivních architektur se zaměřením na ukotvení jazyka ve vizuálních vjemech. Navrhovaná architektura využívá pravděpodobnostních Bayesovských modelů, které jsou vedle neuronových sítí jedním z hlavních nástrojů používaných ve výpočetním kognitivním modelování. Pravděpodobnostní (Bayesovské) modely byly využity v takových úlohách jako zpracování jazyka, rozhodování nebo kauzální učení. V první části práce je popsán nově navržený algoritmus pro odhad počtu shluku v datech. Druhá část se již zabývá popisem samotné kognitivní architektury. Předkládaná hierarchická kognitivní architektura zpracovává samostatně vizuální (statická) a jazyková (časová) data a kombinuje je v multimodální vrstvě. Důležitou vlastností je komposicionalita systému - jeho schopnost odvodit význam předtím neslyšených vět a neviděných objektů z předchozích pozorování a jeho schopnost naučit se přiřadit význam k jednotlivým vizuálním vlastnostem na základě vět s variabilní délkou. Navrhovaná architektura byla implementována do humanoidního robota iCub a testovaná jak na uměle generovaných datech, tak na datech reálných.

Klíčová slova: učení bez učitele, osvojování jazyka, ukotvení symbolů, směs Gaussiánů, pravděpodobnostní Bayesovský model

Překlad názvu: Hierarchický pravděpodobnostní model akvizice jazyka

Contents

Figures

Tables

Abbreviations

CCN	Computational cognitive neuroscience
GMM	Gaussian mixture model
gGMM	greedy Gaussian mixture model
GMEM	greedy merge EM algorithm
EM	Estimation-Maximization algorithm
AIC	Akaike information criterion
BIC	Bayesian information criterion
AWE	Approximate weight of evidence
NMF	Neural Modeling Fields theory
SOM	Self-organizing map
GWR	Growing-when-required neural gas
MCMC	Markov Chain Monte Carlo
HMM	Hidden Markov Model
fMRI	Functional magnetic resonance imaging
PCA	Principal component analysis
PC	Principal component
YARP	Yet another robotic platform

Nomenclature

$P(h\|d)$	conditional probability of h given d
N	number of input data points
$\vec{x_n}$	data point n
LL	overall log-likelihood (similarity measure)
$\vec{r_k}$	mixing proportions of the mixture component k
$\vec{S_k}$	covariance matrix of the mixture component k
$\vec{m_k}$	cluster centre of the mixture component k
$\vec{\theta_k}$	estimators (approximate model parameters) of the mixture component k
$l_k(\vec{x_n}\|\vec{\theta_k})$	Gaussian density (similarity of data point $\vec{x_n}$ with the component k)

Chapter 1

Introduction and Motivation

The emergence of language in human prehistory enabled people to expand knowledge quickly and pass it from generation to generation, something humans have been unable to do up to this point or it was done mainly through genetics. The amazing language capacity caused a quick development of the language and the langauge evolved through ages from the system of few sounds describing simple objects or situations to a very complex and abstract symbol system with a complex grammatical structure. As it was getting more and more complex, it became more powerful tool, which was able not only to describe the sensoric data, but also situations, express metaphorical meaning or some abstract concepts.

The essential (and still not fully answered) question in language acquisition is how percepts are anchored in some arbitrary symbols. In other words, how words (symbols) get their meanings. This is a so called symbol grounding problem. For many years, there has been a joint attempt of cognitive modeling, neuroscience, psychology and machine learning to understand how human solve this 'problem'. There are many questions which are left open – *how is the language acquired* by a newborn child? How do babies find to which object or property should the perceived word be assigned to in sensorically very rich world? How do they deal with the *noise* and *uncertainity* in received data? How do they find out *how many different words* or properties should be discriminated? And what about the case when the *unambiguous mapping* between the language and sensorical data does not exist? Furthermore, what about cases when the language inputs describe *abstract concepts* or very *complex situations*, to which we cannot simply point by a finger? In this Ph.D. thesis, I aim to answer some of these questions although some of them are beyond the scope of this thesis.

The difficulty of the task was well described in a well-known experiment done by Quine [1] who imagined the anthropologist meeting a native who pointed at the scene and said "*gavagai*". When the anthropologist is stimulated in a situation by seeing a rabbit, he will suppose that the word represents running rabbit in front of him, even though it could mean as well "*ground*", "*sun*", "*hello*", or whatever else. This problem is related to language relativity, as there are several objects and their features that are described by words. The simple version of this problem consists of simple visual scene and separate words that are grounded based on statistical co-occurrence (cross-situational learning). A more difficult version of this problem requires cognitive mechanism for grounding visual scenes described by sentences with variable structure.

Computational cognitive models of grounding language are primarily based on the psychological experiments which have studied relation between perception and language [2] and language and action [3] . There are two main streams of these models: one focuses on

developing models based on highly adaptive neural networks [4, 5], others have developed probabilistic models [6], which can handle better noisy and incomplete data. One of the main long-term objectives of many teams worldwide is building the conversational robots, which will be able to participate in cooperative tasks mediated by a natural language. It has been shown how robots can learn new symbols using already grounded symbols and their combination [4] and how to transfer knowledge between agents [7]. Cangelosi [4] has presented their research on language emergence and grounding in sensorimotor agents and robots. This model was further extended by Tikhanoff [8], who did iCub simulation experiments and focused on integration of speech and action. Grounding of higher order concepts in action was also explored by Stramandinoli at al. [9], who made use of recurrent neural networks. Sugita and Tani [10] in their paper describe the experiment dealing with semantic compositionality – the capability of a robot to use the compositional structure to generalize novel word combinations. The current state-of-the-art on grounding variable length sentences is very restricted and deals only with static scenes [6].

The ability to learn language through perception and especially through visual grounding is not only important for understanding human cognition but is also applicable in many areas such as verbal control of interactive robots [11], automatic sports commentators [12], car navigation systems, for visually impaired, situated speech understanding in computer games [13], automated generation of weather forecasts [14], tutoring children foreign language [15], etc. It should be mentioned that there is also a recent interest in European and other institutions to provide industrial and rescue robotics with cognitive capabilities – especially language – so those can easier communicate with a human operator.

Despite the growing number of studies, there is still not available fully unsupervised architecture, which would be able to deal with language grounding [16], particularly language grounding in a case where we don't have each sentence with fixed structure and when there is more than one object in a scene.

The overarching goal of the research proposed herein is to extend the capabilities of robotic systems to provide more autonomous and adaptive behaviours and to allow a more natural communication between human and robots not dependant on the recognition of sentences with fixed structures. The hierarchical probabilistic architecture for the language acquisition is proposed with the focus on *grounding the language* (in a form of variable length sentences) *in a visual input*. Because the focus of the thesis is mainly on how to find *mapping between language and vision*, I used very simple datasets consisting of one object in a scene with varying properties, which are described by a sentence with fixed or variable length.

The goals of the thesis are following:

- Propose the probabilistic hierarchical architecture for the language acquisition, which will be fully unsupervised.
- Propose algorithms, which will enable working in this unsupervised enviroment (finding unknown number of clusters in a data autonomously, deal with non-unambiguous mapping or variable length sentences etc.).
- Test this architecture on artificial data (both visual and language).
- Test this architecture in a robotic scenario on the real-world data.

1.1 Personal background and Motivation

As an introduction to this thesis, I would like to mention here some background information about my journey towards the research described herein. After finishing master studies at Mathematical-Physical faculty at Charles University, where I focused on magnetic properties of lanthanoid compounds [17], my first daughter was born and I got fascinated by incredibly quick progress that she was doing every day. This motivated me to start reading first books about neuroscience and cognition and I soon decided that I would like to dedicate my life to understanding how these cognitive abilities are created and evolved. By that time I found one of the few cognitive scientist in Czech Republic, Mgr. Michal Vavrečka, Ph.D., in BioDat group (biomedical data processing group) at Faculty of Electrical Engineering, who kindly took me with doc. Lhotská as their Ph.D. student. I started to process EEG data during cognitive tasks (my works from this research were published in [18, 19]) and slowly moved to cognitive modeling and its application in robotics [20], where my multidisciplinary background (and especially mathematical background) turn to be an advantage. My research focused mainly on models of language acquisition and grounding language in perception – the results of this research are described in this thesis. My own architecture was published in [21] and I implemented it into the humanoid iCub robot during my research stay in Plymouth, UK (resulting video can be seen at [22]). To be able to design fully unsupervised architecture, I had to deal with problems in unsupervised learning such as ability to find unknown number of clusters in data [23] or finding the best mapping between two clusterings. It was a long way, but I finally know, that I found the field where I could hopefully contribute to the nowdays knowledge, apply my multidisciplinary background and which will never stop to fascinate me.

1.2 Task specification

The more detailed specification of individual thesis goals is described here.

- Propose the probabilistic hierarchical architecture for the language acquisition, which will be fully unsupervised. This task consists of following subtasks:
  - Explore the state-of-the-art cognitive architectures with the focus on grounding language in vision and clustering algorithms used for processing visual and language information.
  - Propose and implement the algorithms for processing data in individual modalities. Visual (static) and language (time sequence) data should be processed separatelly using unsupervised clustering algorithms.
  - Propose the mechanism for assigning information from individual layers in the multimodal layer. The proposed algorithms should be able to deal both with sentences having fixed and variable structure.
- Propose algorithms, which will enable working in this unsupervised enviroment.
  - Propose the algorithm which will be able to detect the number of clusters in observed data. Especially focus on finding optimal number of components in a mixture of Gaussians which should be used for processing visual data.

  - Propose an algorithm which will be able to find mapping between individual modalities in a case of variable length sentence and when there is non-equal number of clusters in individual subdomains.

- Test the performance of the architecture on artificial data. The artificially generated visual data should have varied several visual features such as color, shape, orientation, size and texture.

  - Test performance of the proposed algorithms on the data from individual subdomains (vision and language) and compare the performance to other state-of-the-art algorithms.
  - Test dependency of the recognition accuracy on the level of noise in data.
  - Test compositionality of the architecture - ability to derive meaning of unknown sentences and combinations of percepts from that of known ones.
  - Test ability to find mapping between visual and language layer on these data.

- Test this architecture in a robotic scenario on the real-world data. Implement the proposed architecture into the robotic simulator and test its performance in a real-world scenario (real images and voice). For this task, number of objects and its differing visual properties will be restricted. Architecture will be first implemented into the robotic simulator which will enable easier control of experimental conditions. Afterwards, the architecture will be implemented into the physical robotic platform. As a robotic platform will be used humanoid robot iCub. Similarly as for the artificial data, performance of individual modalities should be tested as well as the ability to find mapping between individual modalities. The effect of noise in language data should be also investigated.

1.3 Thesis organization

The rest of the thesis is organized as follows. The thesis consists of two major parts. The first part describes a technique for estimating number of components in Gaussian mixture models (GMM). This is an important issue in tasks that take advantage of unsupervised learning, including symbol grounding where we don't know the number of visual and language categories beforehand. This proposed algorithm is subsequently used in the multimodal cognitive architecture, which is described together with experimental results in the second part of the thesis. Chapter 2 gives an overview of the state-of-the-art probabilistic models of cognition, namely general Estimation-Maximization (EM) algorithm and GMM are described. In Chapter 3, the proposed algorithm for unknown number of components is described and its performance on both artificial nad real-world datasets is shown. Second part of the thesis starts with Chapter 4, where state-of-the-art cognitive models of vision, language and symbol grounding are summarized, focusing on implementation of these models into the robotic platform. The proposed multimodal cognitive architecture for language acquisition is presented in Chapter 5. Datasets used for testing the performance of this architecture are described in the Chapter 6 and in Chapter 7 are presented experimental results for unimodal layers as well as for the top-most multimodal layer both on artificial data and on the real-world data from humanoid

iCub robot. Results of mapping language-to-vision are emphasized. These results are discussed in Chapter 8. In Chapter 9 is provided an overview of thesis contribution and in Chapter 10 are outlined possible extension and future prospects of the presented work.

1.4 Computational cognitive modeling

A *computational cognitive modeling* covers simulations of complex mental processes in different areas of cognition, especialy in human problem solving, based on computational model. The goal of cognitive modeling is not only to understand, describe and model observed human behavior, but also to predict it.

Cognition can be defined as the *mental process of knowing*, including aspects such as awareness, perception, reasoning and judgement [24]. The term itself originates from the Latin word cognitio, from cognoscere, which is composed from *-co* (intensive) + *nosecere* (to learn).

The history of interest in human cognition is untrackable. The first documented remarks can be found in the antic philosophy where the human *psyché* is discussed. The *dualism* of a body and mind was firstly proposed by Plató who believed that the mind is located within the brain. Eventhough the Antic era was followed by centuries of Christian dogmatism during the Middle Ages, the philosophical focus has shifted from God to the humankind during the Reneissance and the Plato's idea of mind-body dualism was recovered in the 17th century by Descartes who believed in the introspective methods.

Immanuel Kant, the great philosopher of the 18th century, was the first to realize that the understanding requires synthesis of two distinct types of knowledge – the general truth (*a priori*) and experience-based (*a posteriori*) knowledge. Kant had a huge impact on philosophy and his ideas in connection with knowledge of human body (physiology) of that times were the base stones for establishing psychology as a separate discipline. The first half of the 20th century was an era of behaviorism which believed that all the basis for knowledge is a sensory perception.

In 1950s, primitive computers have been constructed, George Miller summarized studies showing that the mental capacity of human is limited, Herbert Simon, Marvin Minsky, John McCarthy, Allen Newell and others found artificial intelligence and the linguist Noam Chomsky rejected ideas of behaviorism [25]. That was the beginning of the *cognitive science* as an interdisciplinary study of mind, which interconnects knowledge from philosophy, neuroscience, linguistics, psychology, artificial intelligence and anthropology. Oppositely to the behaviorism, cognitivists believe that the human behavior can be understood mainly by grasping how the human mind works. This is a reason why many detailed *psychological models* of human thinking were developed and computerized using artificial intelligence methods (which should be based on neuroscience findings).

The majority of these models are focused on specific cognitive areas – e.g. visual perception [26], implicit and explicit learning [27] or language processing [28].

Analysis can be done on different levels of abstraction. There are several theories what are these levels. Well known is division to the three levels defined by David Marr [29]:

1. *Computational* level
2. *Algorithmic* level

3. *Implementation* level

In my work, I use the differentiation to four levels, which was defined by Sun et al. in 2005 [30]. The levels are:

1. *Sociological* level – inter-agent processes, collective behavior of agents
2. *Psychological* level – individual behavior of agents
3. *Componential* level – intra-agent processes, modular construction of agents
4. *Physiological* level – biological implementation

Most of the computational analysis is performed on the componential level where agents' functions and internal processes are defined.

These levels correspond to four bands defined by Newell [31]: biological, cognitive, rational and higher bands (social, historical and evolutionary band). Each of them being divided into separate levels and operating on a different time scale.

Comparing different models of cognition

Traditional models of cognition can be divided into three major groups – *connectionistic modeling*, *Bayesian parametric models* and *rule-based modeling* proposed by Minsky [32]. In the last years, *Bayesian nonparametric models* became very popular. This method was proposed to suppress limitations of simple parametrical models, but the volume of classification space grows exponentially with the dimensionality. Training requirements for nonparametric paradigms have thus often exponential complexity [33].

The above mentioned models combine different degree of *adaptivity* and apriority, their *neurorelevance* differs widely and all of the concepts face combinatorial explosion of *computational complexity* [33].

The number of cognitive models is big. It is useful to give some restrictions and rules for them to be able to choose the best one.

Constraints on complex cognitive system (mind, cognitive architecture) summarized in Newell [31] are following: flexibility, adaptivity, autonomy, self-awarness, operation in real-time and in complex environments, usage of symbol and abstractions, usage of language, learning from environment, acquiring capabilities through development, be realizable as a neural system, be constructable by an embryological growth process and arise through evolution. The extended version of these desiderata for cognitive architectures includes [34]: ecological realism, bio-evolutionary realism, cognitive realism and eclecticism of methodologies and techniques.

In [35], four properties are discussed, which should satisfy every *computational cognitive neuroscience* (CCN) *model*:

- The *neuroscience ideal*: A CCN model should not make any assumptions that are known to contradict the current neuroscience literature.
- The *simplicity heuristic*: No extra neuroscientific detail should be added to the model unless there is data to test this component of the model or the model cannot function without this detail.

- The *Set-in-Stone Ideal*: Once set, the architecture of the network and the models of each individual unit should remain fixed throughout all applications.
- The *Goodness-of-Fit Ideal*: A CCN model should provide good accounts of behavioral data and at least some neuroscience data.

These neuroscience requirements increase a number of constraints for the wide field of cognitive models, which focuses mainly on behavioral aspects. The above mentioned properties will enable to find relations between seemingly unrelated behaviors through same neuroscience aspects or to predict some behavioral aspects, which could be hidden from a strictly cognitive perspective.

Models can be compared based on their degree of adaptivity and apriority, neural and bilogical plausability and computational complexity.

We could also focus on whether the model integrates *online* or perform *incremental learning*, which would enable the model to continuously update with an incoming information.

Part I

Estimating number of components in mixture models

Chapter 2

Probabilistic (Bayesian) models of cognition

Probabilistic models of cognition describe learning and reasoning as inference in complex probabilistic models. The history of *probability theory* dates as far back as 18th century. The theory was developed to analyse games of chance but quickly became a formal account of rational reasoning in a case of uncertainity [36]. The Bayesian statistics was used in cognitive science in many different ways, which are discussed in [37]. We can separate them to three main flows: statistician view uses Bayesian approach for conducting standard analyses of data sampling distributions and null hypothesis testing (e.g. [38]); the theoretician view uses it to describe how inferences are made by a human mind on a computational level [39, 40] or as an theoretical methaphor for behavior at the implementation and algorithmic levels [41]; the last approach relates models of psychological processes to data (e.g. [42]).

Among other topics, *Bayesian models* have addressed animal learning [43], visual scene perception [44], sensorimotoric tasks [45, 46], semantic memory [47], language processing and acquisition [28, 48], and social cognition [49]. Recently, also *nonparametric Bayesian models* became increasingly used. The probabilistic models try to find the answer to the question how could the human mind learn so much from such a sparse and noisy data, which we observe through our senses [40].

The biggest advantage of the parametric-based models is that their parameters can capture variablities and uncertainities in the data because probability distributions are used instead of frequencies or sampling distributions. Models based on a priori logic rules [32] require no training but cannot adapt. On the other hand, neural networks use no a priori knowledge and learn only through the adaptivity (e.g. [50, 51]). Parametric model-based algorithms combine *adaptivity of parameters* with *apriority of models* (e.g. [40, 28, 48, 52, 53, 54]) and can adapt the models to the *variabilities in data*.

The main problem of neural networks and other adaptive models learning from data is the *"the curse of dimensionality"* [55]. It addresses the problem that the number of necessary training examples is increasing combinatorically with the dimensionality of the problem. On the other side, rule-based models with a priori rules face the combinatorical growth of the number of rules necessary to teach the system [33]. The parametric models, which combine both apriority and adaptivity, face the combinatorial explosion of the computational complexity because the segmentation requires evaluation of combinatorially many data subdivisions into subsets corresponding to the individual models [52].

Concerning the *behavioral* and *neural plausability*, it is reasonable to ponder that the organism's response to the signals received from afferent sensoric fibres is selected by choosing an option from a list of all possibilities, which is most appropriate considering a current state of an organism [56]. It should be also mentioned that during certain tasks

people make use *Bayesian inference*. These tasks include combination of haptic and visual information about an object [57], perceptual [58, 59] or sensorimotoric [45, 46] tasks. On the other hand, eventhough there exist proposals how could neural populations performing Bayesian inference [60], up to date the neuroscience evidence is only limited; a high-level perception and many other biological mechanisms probably do not implement Bayesian inference (or do not rely only on it). Also some processes at algorithmic or neurocomputatinal levels and these on computational level with no induced inference are not suitable for Bayesian analysis [40]. Rather than advocating a monolithic and exclusively probabilistic view of the mind, Chater, Tenenbaum and Yuille in [28] suggest instead that probabilistic methods have a range of valuable roles to play in understanding cognition.

2.1 Bayesian modeling

Bayesian models are based on *Bayes' rule* [61, 62], which is an elementary result of the probability theory:

$$P(h|d) = \frac{P(d|h) * P(h)}{\sum_{h' \in H} P(d|h') * P(h')}. \tag{2.1}$$

As can be seen, a posterior probability $P(h|d)$ depends only on prior probabilities and their likelihoods. Likelihoods $P(d|h')$ reweight the hypothesis from the hypothesis space H and describe how well they match the data.

Adaptive models describing data $\vec{X}$ can be described as follows in general:

$$\vec{X}_N = M_k(\vec{\theta}_k, N), \tag{2.2}$$

where $\vec{X}$ is data, $\vec{\theta}_k$ are parameters of model M_k, and N is number of input data vectors.

The learning in such a model is provided by maximizing the similarity measure between data and a model.

Association (or segmentation) between the input data vector and objects can be described mathematically as a subdivision Ξ of the inputs $(\vec{x}_1, \ldots, \vec{x}_n)$ to the subsets ξ_k, which correspond to the objects $\Xi = \{\xi_1, \xi_2, \ldots, \xi_k\}$. The overal likelihood (similarity measure) between model and data can be defined as follows:

$$LL(\vec{\theta}) = \sum_{j=1}^{K} \sum_{n=1}^{N} ll(\vec{x}_n|M_j) = \sum_{j=1}^{K} \sum_{n=1}^{N} \log(l(\vec{x}_n|M_j)), \tag{2.3}$$

where $l(\vec{x}_n|M_j)$ is a conditional similarity measure between data $\vec{x}_n$ and model M_j, K is number of components (objects), and N is number of data points.

The aim is to find such a segmentation Ξ and models parameters $\vec{\theta}_k$ to maximize the similarity LL:

$$\max_{\Xi} \sum_{j=1}^{K} \max_{\vec{\theta}_{M_j}} \sum_{n=1}^{N} ll(\vec{x}_n|M_j). \tag{2.4}$$

The positives of the models are the following: they provide a link between human cognition and the normative prescriptions of a theory of rational inductive inference, combine statistical learning with symbolic structure and enable communication with

other fields studying computational principles [40]. The models mainly work at Marr's computational level rather than on the algorithmic or process level.

There exists no closed form solution for finding optimal parameters $\vec{\theta}^* = \text{argmax}_{\vec{\theta}} LL(\vec{\theta})$. Therefore parameters $\vec{\theta}^*$ have to be estimated either by the numerical optimization methods such as *Markov Chain Monte Carlo* (MCMC) method or learned using an iterative *EM algorithm* [63, 64].

EM algorithm maximizes instead of $LL(\vec{\theta})$ its lower bound $F(\vec{\theta})$. At each iteration we find an optimal lower bound $F(\vec{\theta^t})$ at the current guess of parameters $\vec{\theta^t}$ and then maximize this bound to obtain an improved estimate $\theta^{\vec{t+1}}$. Initial values of parameters $\vec{\theta^0}$ may be chosen randomly or using more sophisticated method.

1. *Expectation step (E-step)*: Calculate the expected value of the log likelihood function with the respect to the unknown data $\mathcal{Y}$ given the observed data $\mathcal{X}$ and the current estimate of the parameters $\vec{\theta^t}$:

$$Q(\theta, \theta^t) = E[log(p(\mathcal{X}, \mathcal{Y}|\theta)|\mathcal{X}, \theta^t].$$

2. *Maximization step (M-step)*: Find parameters that maximize the expected value of the log likelihood estimate:

$$\theta^{\vec{t+1}} = \underset{\vec{\theta}}{\text{argmax}}\, Q(\vec{\theta}, \vec{\theta^t}).$$

General Gaussian mixture model algorithm

General mixture models are defined as a convex mixture (with mixing proportions r_k) of some probability distribution $l_k(\vec{x_n}|\vec{\theta_k})$:

$$f_k(\vec{x_n}) = \sum_{k=1}^{K} \vec{r_k} l_k(\vec{x_n}|\vec{\theta_k}). \tag{2.5}$$

In the case of *Gaussian mixture model* (GMM), probability distributions $l_k(\vec{x_n}|\vec{\theta_k})$ are d-dimensional Gaussian densities (where parameters $\vec{\theta_k}$ are cluster centres $\vec{m_k}$ and covariance matrices $\vec{S_k}$):

$$l_k(\vec{x_n}|\vec{m_k}, \vec{S_k}) = (2\pi)^{-d/2}|\vec{S_k}|^{-1/2} \exp[-0.5(\vec{x_n} - \vec{m_k})^\top \vec{S_k}^{-1}(\vec{x_n} - \vec{m_k})]. \tag{2.6}$$

The overall likelihood (or log-likelihood) between data and model is described by the following equation:

$$LL(\vec{\theta}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} r_k l_k(\vec{x_n}|\vec{\theta_k}) \right), \tag{2.7}$$

where K is number of components, N is number of data points, $\vec{\theta}$ are estimators (approximated parameters of model k), $l_k(\vec{x_n}|\vec{\theta_k})$ are Gaussian densities (similarities of data point $\vec{x_n}$ with the component k), and r_k is mixing proportion of component k.

Mixing proportions must satisfy:

$$\sum_{k=1}^{K} r_k = 1 \text{ and } r_k > 0 \text{ for } k = 1, \ldots, K. \tag{2.8}$$

Estimating GMM parameters

Since there exist no closed form solution for finding optimal parameters $\vec{\theta}^*$, we have to find the parameters by the numerical methods or by using simple iterative EM algorithm. The *EM algorithm* for GMM iterates between two steps:

1. *E-step*: Estimating all probabilities :

$$f_k(\vec{x_n}) = \frac{r_k l_k(\vec{x_n}|\vec{m_k}, \vec{S_k})}{\sum_{k'=1}^{K} r_{k'} l(\vec{x_n}|\vec{\theta})}, \tag{2.9}$$

2. *M-step*: Choosing the parameters that maximize the log-likelihood when the probabilities are known:

$$r_k = \frac{1}{N} \sum_{n=1}^{N} f_k(\vec{x_n}), \tag{2.10}$$

$$\vec{m_k} = \frac{\sum_{n=1}^{N} f_k(\vec{x_n}) \vec{x_n}}{\sum_{n=1}^{N} f_k(\vec{x_n})}, \tag{2.11}$$

$$\vec{S_k} = \frac{\sum_{n=1}^{N} f_k(\vec{x_n})(\vec{x_n} - \vec{m_k})(\vec{x_n} - \vec{m_k})^\top}{\sum_{n=1}^{N} f_k(\vec{x_n})}. \tag{2.12}$$

2.1.1 Convergence of EM algorithm

EM algorithm is an iterative algorithm. Its convergence is assured by the fact, that the likelihood change is nondecreasing at each iteration. Lets suppose that current estimate of parameters θ is θ_n. Parameters in a next iteration of the algorithm θ_{n+1} are chosen to maximize $L(\theta)$: $L(\theta_{n+1}) - L(\theta_n) \geq L(\theta_n) - L(\theta_n) = 0$. Therefore $L(\theta)$ is nondecreasing at each iteration. The monotonical convergence to a fixed point is guaranteed. Anyway it is not guaranteed that the fixed point reached for θ^* will be global or local maxima. It has been shown [63, 65], that θ^* can be either local maximum, local minimum or stationary point (see example in Murray [66]).

It was also shown, that EM algorithm is generally first-order or linearly convergent algorithm [67]. Rate of EM algorithm convergence can be found by calculating the information matrices for the missing data and for the observed data [**?**, 68].

Xu and Jordan [69], proved the general dominance of EM algorithm over gradient method for Gaussian mixtures. Anyway the convergence rate strongly depends on the overlap of components [**?**, 63, 70]. The methods aiming to increase the convergence are usually based on superlinear optimization theory [71, 72].

2.2 Time Sequences – HMM

Hidden Markov Model (HMM) is a simple and effective tool used to model time-varying signals. The Hidden Markov Model is a special case of a *probabilistic finite-state machine* that was sucessfully aplicated for example in speech recognition task [73], hand-written text processing [74] or gesture recognition [75]. The assumption that the probability of an observation at time n only depends on the observation at time $n-1$ is called a *first-order Markov assumption*. HMM with a continuous output probability density (CHMM) is characterized by the following parameters described in [73]:

$$
\begin{aligned}
&\mathrm{T} = \text{length of the observation sequence}\\
&\mathrm{N} = \text{number of states in the model}\\
&\mathrm{M} = \text{number of observation symbols}\\
&\mathrm{S} = \{s_0, s_1, ..., s_{N1}\} = \text{distinct states of the Markov process}\\
&\mathrm{V} = \{v_0, v_1, ..., v_{M1}\} = \text{set of possible observations}\\
&\mathrm{A} = \{a_{ij}\} = \text{state transition probabilities, } \sum_{j=1}^{N} a_{ij} = 1,\\
&\qquad a_{ij}\text{: probability of going from } s_i \text{ to } s_j\\
&\mathrm{B} = \{b_i(v_k))\} = \text{observation probability matrix,}\\
&\qquad b_i(v_k)\text{: probability of generating } v_k \text{ at } s_i\\
&\Pi = \{\pi_1, \ldots, \pi_N\} = \text{initial state probabilities, } \sum_{i=0}^{N-1} \pi_i = 1,\\
&\qquad \pi_i\text{: probability of starting at state } s_i\\
&\mathrm{O} = (O_0, O_1, ..., O_{T1}) = \text{observation sequence.}
\end{aligned}
\tag{2.13}
$$

There are *three fundamental problems* that can be solved using HMMs:

1. **Problem 1**: Given a sequence of observations O and a model $HMM = (S, V, B, A, \Pi)$, determine the likelihoods $P(O|HMM)$ of the observed sequence O (solved by the the forward algorithm [73]).

2. **Problem 2**: Given a sequence of observations O and a model $HMM = (S, V, B, A, \Pi)$, find an optimal sequence of states S (solved by Viterbi algorithm [76]).

3. **Problem 3**: Given a sequence of observations O, find the model parameters that maximizes the probability of observing a sequence O (solved by learning procedures such as Baum-Welch algorithm).

The most known learning procedure used to find unknown parameters of HMM is a *Baum-Welch algorithm* which is a variant of the iterative EM algorithm.

Clustering HMMs

Clustering of HMMs provides a way to identify similar sequences and group these data sequences based on the transition patterns [77]. There are variety of algorithms for

discovering clusters within data (both parametric and non-parametric), ranging from traditional clustering methods, such as the k-means or hierarchical clustering to spectral clustering that has got an increasing attention recently. Anyway, the application of these techniques for clustering sequential data poses a number of additional challenges. For example, we have to detect similar hidden properties between sequences of different lengths etc. [78]. There are also several methods, which do not cluster HMMs but make use of HMM to cluster data.

Application of traditional *k-means* algorithm for clustering HMMs is described in Algorithm 1.

Algorithm 1 Clustering HMM using k-means

```
for each data point x_n^→ do
    draw an data point x_n^→ at random (without replacement)
    create an HMM and train on data point x_n^→
end for
while cluster membership change do
    for each data point x_n^→ do
        for each cluster c do
            compute log likelihood of x_n^→ under c
            assign x_n^→ to its highest likelihood cluster
        end for
    end for
    for each cluster c do
        retrain HMM on items in c
    end for
end while
```

Jabera et al. [78] developed the *semi-parametric method* for clustering HMMs and applied recent work of spectral clustering to the time-series data. They use probability product kernels [79] to measure similarity between HMMs and laid down the foundations for the use of the *Bhattacharyya affinity* in this area. Fan in his dissertation thesis [77] followed up on this approach and proposed a semi-parametric method using Bhattacharyya affinity to measure the pairwise similarity between sequences. He used sequential model to extract the features of the data, constructs the distance matrix based on the features and finally he applies existing clustering algorithms to obtain the cluster assignment.

Another method is the *agglomerative HMM clustering*. Smyth in [80] used the log-likelihood to measure the discrepancy between two sequences and then applied hierarchical clustering on the resulting distance matrix. Butler [81] found that the hierarchical HMM is an effective tool for unsupervised learning of sound patterns. Initially there are N singleton clusters $c_i, i \in \langle 1, N\rangle$ each modelled by one HMM $M_i, i \in \langle 1, N\rangle$ each trained on a single data item. These singleton clusters are sequentially merged according to the distance measure $LL_{ij} = log(l(x_j|M_i))$ (each item x_j is evaluated under each model M_i), which is normalized (column-wise [81]) and symmetrized $LL_{ij} = \frac{LL_{ij}+LL_{ji}}{2}$. The pair of clusters with the lowest distance measure is merged in each iteration. The combined cluster replaces the lower-numbered merged cluster $k = \min(i, j)$, with $c_k = c_i c_j$ and j discarded. Afterwards the distance measure LL_{ij} is updated: $\forall l\colon L_{kl}, L_{lk}$ are computed. The singleton initialisation is intended to scatter the cluster models randomly through the data. Items are then moved among clusters until a stable arrangement is found (see

Algorithm 2) [80, 81].

Algorithm 2 Agglomerative HMM clustering

for each data point $\vec{x_n}$ **do**
    init a HMM with k states
    learn a single HMM
    compute $LL(i)$
end for
for each data point $\vec{x_n}$ **do**
    for each data point x_j, $j \neq n$ **do**
        loglik(n, j) – compute loglikelihood of data point $\vec{x_n}$ in a HMM learned using data point x_j
    end for
end for
while number of $HMM > K$ (K – number of clusters to detect) **do**
    find loglik maximum value to detect the two HMMs, which will be merged
    relearn merged HMM and update loglik variable
    delete merged HMM from all variables and update indices vector
end while

Coviello in [82] presented the algorithm to cluster HMMs based on the *hierarchical EM* (HEM) algorithm and applied it to clustering of motion capture sequences. A given collection of HMMs is clustered into groups based on distributions they represent and each group is characterized by a cluster centre. That is, a novel HMM that concisely and appropriately represents each cluster. A *model-based method* was also developed by Panuccio et al. [83] or Garcia et. al [84] and a similarity-based method was proposed by Bicego et al. [85]. These methods do not scale well for large data problems because constructing the distance matrix based on the pairwise likelihood of HMMs is computationaly expensive. Alon et al. [86] considers EM algorithm for clustering HMM. His method has a problem when time series data are not radially distributed.

There are several other methods such as a clustering method using HMM parameter space and *eigenvector decomposition* proposed by Porikli et al. [87] or sequence clustering method with HMM setting based on the transition matrix induced in a common HMM which was proposed by Garcia et al. [88]. Porikli et al. [87] has shown that the number of eigenvectors is proportional to the number of clusters.

2.3 Hierarchical Bayesian Models

Hierarchical models structure data into groups, which can be represented by a set of modules or sub-modules. Different parameters are used for each group. Some of these parameters describing the model are conditionally dependent on other parameters. The example of such a case is measuring data from a group of subjects, where each subject characterized by its own parameters is an element from superior population distribution (parameters of this superior distribution are called hyperparameters) [89].

Hierarchical Bayesian model is a model written in a hierarchical form that is estimated using Bayesian methods [90]. Usually, we can not analytically determine the posterior distribution because we can not find analytically the normalizing constant in Bayes' theorem. Parameters of hierarchical models can be determined by the maximum likelihood

estimation method. There has been a dramatic increase of interest in the recently developed MCMC (*Monte Carlo Markov Chain*) methods to sample from the posterior, which work in hierarchical models particularly well.

Lee [37] has described the basic design of hierarchical Bayesian models and their use for *modeling cognitive functions*. Hierarchical Bayesian models are based on the simplest Bayesian non-hierarchical model, which is visualized in Fig. 2.1. In this type of model, data is generated directly from parameters through a likelihood function f. Despite its simplicity this model was used in Signal Detection Theory [91] to model memory [92], in Generalized Context Model for category learning [93] or for finding distribution of accuracy and response times of simple decisions [94].

Figure 2.1: General non-hierarchical Bayesian model [37]: d – data, $f(.)$ – processes, $\mathbf{\Theta}$ – parameters.

More complex hierarchical models can be of many types. Three basic models are visualized in Fig. 2.2 [37].

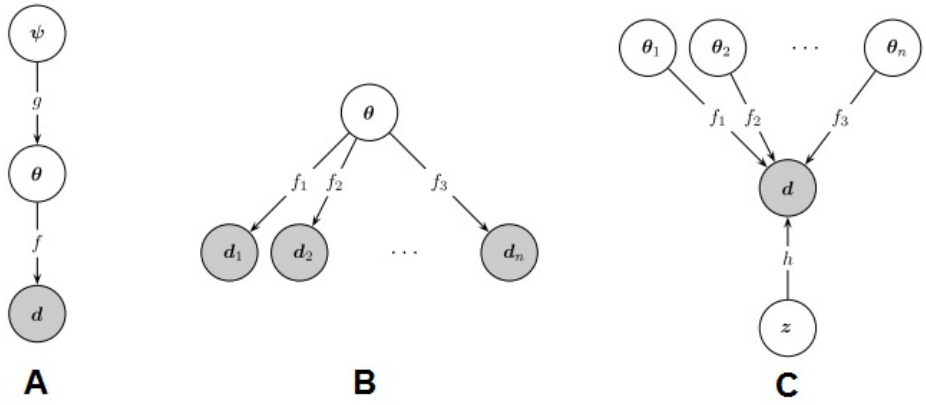


Figure 2.2: Three basic types of complex hierarchical models [37]: d – data, $f(.), g(.)$ – processes, $\mathbf{\Theta}$ – parameters, h – unification rule, z – bias.

Model A reflects the situation where combination of 2 mappings is used and parameters Θ are generated from basic parameters ψ using a process $g(.)$. This model was used for accommodation of individual differences, for modeling memory retention [89], memory [95] or emotional states [96].

Model B is a visualization of the situation where the same parameters Θ can lead to different data $d_1, \ldots, d_n$ through the processes $f_1, \ldots, f_n$. Based on this model, a

joint model for recognition, free recall and serial recall was designed, assuming that all processes work on the same memory system [97].

The last presented model C describes the opposite situation where one set of data d is influenced by different sets of parameters $\Theta_1, \ldots, \Theta_n$ through processes $f_1, \ldots, f_n$. This model was applied by Ratcliff [98] for modeling the accuracy and reaction times distributions for simple decision-making or in the Topics model [99].

In Figure 2.3 [37], the most complex hierarchical model can be seen which interconnects models of type A, B and C from Fig. 2.2.

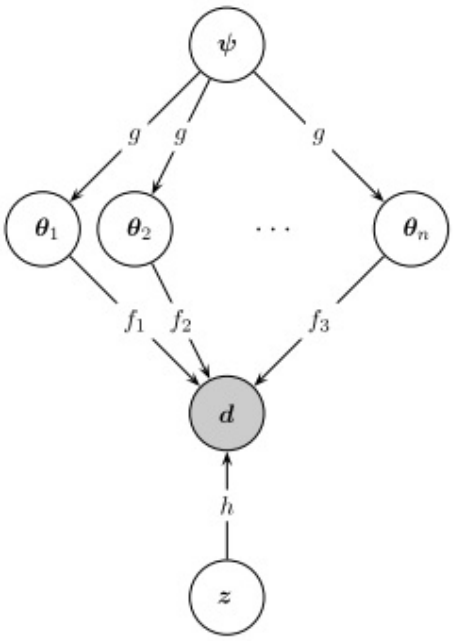


Figure 2.3: A hierarchical model where data d are generated from different models which have parameters $\Theta_1, \ldots, \Theta_n$ (which are generated from basic parameters ψ through process $g(.)$) through processes $f_1, \ldots, f_n$. The unification with bias z is governed by the rule h [37].

Chapter 3

Proposed technique for estimating number of components in Gaussian mixture model

As a part of the proposed cognitive architecture I have designed the algorithm for finding the number of clusters in a data. The problem of *unknown number of clusters* is a common problem in a clustering analysis, which has not been solved yet. The following chapter is mainly based on the article published in [23].

The ability to find the number of concepts (clusters) in a dataset is essential to many cognitive tasks, spanning from the analysis of objects in a visual scene to word categorization during language acquisition. In the field of machine learning, this is still an unsolved problem. One possible solution is to take inspiration from humans.

In the traditional machine learning approach, the visual input is processed in a bottom-up manner, and cortical regions analyse increasingly complex information. On the other hand, recent *neurological and fMRI neuroimaging studies* have confirmed that visual processing is performed by a *combination of top-down and bottom-up processes* in the brain [100]. Moreover, another research [101, 102] has concluded from fMRI data that top-down signals in attentional experiments are initially vague. This leads us to the conclusion that it should be useful to use this knowledge in the field of machine learning.

The abovementioned processes have already been transformed into mathematical equations by Perlovsky [33, 52] who called them the "*knowledge instinct*", which is defined as an unsupervised cluster analysis, where data comes from unknown sources with some probability distribution (in the simplest case, the Gaussian distribution). The whole problem can be modelled on a mixture model, specifically, on a Gaussian mixture model for Gaussian distribution. The *Gaussian mixture model* (GMM) is a very powerful model-based unsupervised clustering method [103].

Perlovsky hypothesizes that the process of learning for a given number of concepts and parameters is based on the adaptive convergence from vague, highly fuzzy concepts to crisp and deterministic ones using *dynamic logic equations* [104, 105, 106]. This approach was tested in such complex tasks as pattern recognition [107], tracking [108] or acquisition of language in cognitive robotics [109]. The dynamic logic is an unsupervised model-based learning technique that maximizes similarity over the model parameters and the number of models. The correspondence of dynamic logic to the estimation-maximisation (EM) algorithm was shown in [33]. The *EM algorithm* is a standard method used to fit finite mixture models to observed data. When fitting mixture models, we face two main problems. First is the fact that the EM algorithm is a method converging to a local optimum and thus it is very sensitive to initialisation. The second problem is the selection of the number of components.

The standard GMM algorithm can be used only in cases where the number of clusters is known, at least approximately. However, there exist several *modifications of the standard GMM algorithm* that are suitable for an unknown number of components. The most widely used of these are the *split-merge method* [110, 111, 112] and the *greedy Gaussian mixture model* (gGMM) method [113]. The greedy approach starts with a single component and new components are sequentially added into the mixture. In [114], it is shown that this approach can be superior to a normal GMM even in cases where the number of components is known. Both the GMM and gGMM ability to find the optimal solution depends strongly on the initialisation of the cluster centres.

Also, the *stopping criterion* used for finding the optimal number of clusters plays an important role in this process. These are mostly information criteria that penalise the log-likelihood function, e.g. AIC [115], BIC [116], NEC [117, 118]. Both of these problems (initialisation and criteria for assessing the optimal number of clusters) are discussed in the next few chapters in greater detail.

In the following sections, a proposed novel *greedy GMM method with merging* (gmGMM) is described. It is shown that it performs generally better than existing algorithms when searching for the optimal number of components. The important feature of the proposed method is that it combines the greedy and the merging approaches. This results in a better accuracy in the estimation of the number of components (compared to the simple greedy GMM) without the necessity to know the upper bound for the number of components (which is necessary for merging algorithms). The performance of the greedy and normal GMM approaches is compared on unified datasets and further the proposed method is compared to other well-known methods for learning GMM, using both artificial and real-world datasets. Different stopping criteria are confronted and some enhancements to them are proposed.

3.1 Initialization techniques

In the past few years, several methods for the initialisation of component parameters and the localisation of optimal new components have been proposed. Methods used for the initialisation of new components can be divided into those that use only input data and those that also use the parameters of components that are already included in the mixture. Below, a brief overview of the existing initialisation methods is provided.

A simple but, in many cases, very efficient method is the *random initialisation* [110] of a known number of components K. Centres of clusters are randomly chosen vectors from the sample dataset, priors are set to $1/K$ and covariance matrices $\vec{S_k}$ are initialised as:

$$\vec{S_k} = \frac{1}{10d}\mathrm{Tr}(\vec{S}), \text{ for } k = 1, \ldots, K, \tag{3.1}$$

where d is a number of dimensions, and S is covariance matrix of the data set.

For the greedy GMM, a *partial random initialisation* can be used. After the EM converges, a new cluster centre is randomly chosen with a covariance matrix that has smaller values on the diagonal than the covariance matrices of components already included in the mixture [113].

A very commonly used method for initialisation in the GMM with a known number of parameters is a *k-means or fuzzy c-means* [119] algorithm that, unfortunately, faces

high computational demands and tends to find local optima. Hence, it is not a suitable method for the gGMM. The incremental k-means algorithm for the gGMM was used in [120]. A new cluster centre is found using the global search k-means algorithm described in [121]. Other methods include the *optimally smoothed kernel estimate* with local maxima search [122] or *resampling* [113, 123, 124].

The initialisation of components in was described in [52]. The clusters are initialised with large covariance matrices and random centres. The method is limited to tasks with an approximately known number of clusters.

Moreover even a very careful initialisation of the gGMM cannot prevent the system from converging to the local optima. Therefore, *multiple initialisations*, a *genetic algorithm*, and *split and merge* criteria have been used [125, 110]. In the split and merge method proposed in [126], centres of the new components are initialised with the old centre values.

3.2 Criteria for assessing the number of components

Several methods have been proposed for finding the optimal number of components in a mixture. The simplest method is to test different number of clusters and compare the overall like-lihood among the data and model. Any method, which aims to find the optimal number of components in a mixture must and which is based on log-likelihood among data and model, deals with the problem that log-likelihood increases with the number of components. There have been various criteria proposed to overcome this problem. Hence *information criteria* are applied to penalise the term $-LL(\vec{\theta})$ which provides an asymptotically unbiased estimate of the relative entropy loss (its essential part). The general form of the information criteria can be described as:

$$-2LL(\vec{\theta}) + P(K, N, E, \mathrm{F}^{-1}(\mathrm{K}), \vec{r}), \tag{3.2}$$

where P is the *penalisation function* for models that are too complex.

P is an increasing function of the number of components. The information criteria differs in this P function. P function is always dependent on the number of components K and can be further dependent on the number of data points N, entropy E, M - number of parameters specifying each component, inverse-Fisher information matrix $\mathrm{F}^{-1}(\mathrm{K})$, or mixing proportions $\vec{r}$. An overview of the most-used information criteria is in Table 3.1.

Criteria can be divided into those based on the *log-likelihood* $LL(\vec{\theta})$ (AIC, BIC, AWE, etc.) and those based on the *classification log-likelihood* (ICL, etc.). Classification log-likelihood is defined as:

$$CML(\vec{\theta}) = LL(\vec{\theta}) + \sum_{k=1}^{K}\sum_{n=1}^{N} z_{nk} \log f_k(\vec{x_n}), \tag{3.3}$$

where $z_{nk} = 1$, if and only if $\vec{x_n}$ arises from component k.

Criteria AIC, BIC, MML, ICOMP, AWE and MDL are dependent on the number of free parameters in the mixture model with K components $\eta(K)$, which is for the d-dimensional GMM with full covariance matrix:

$$\eta(K) = (K-1) + dK + \frac{dK(K+1)}{2}. \tag{3.4}$$

Criterion	Formula	Reference
AIC (*Akaike information criterion*)	$-2LL(K) + 2\eta(K)$	Akaike (1974)[115], Bozdogan (1984)[127]
BIC (*Bayesian information criterion*)	$-2LL(K) + \eta(K)\ln(N)$	Roberts (1998)[128], Schwarz (1978)[116]
LEC (*Laplace-empirical criterion*)	$-2LL(K) - 2\log r + log\|I(K)\| - \eta(K)\log(2pi())$	McLachlan (2000)[124]
AWE (*Approximate weight of evidence*)	$-2LL(K) + 2\eta(K)(\frac{3}{2} + \ln N)$	Bafield (1992)[129]
ICOMP (*Informational complexity criterion*)	$-2LL(K) + \eta(K)\ln\left(\frac{\mathrm{Tr}(\mathrm{F}^{-1}(\mathrm{K}))}{\eta(K)}\right) - \ln\|\mathrm{F}^{-1}(\mathrm{K})\|$	Bozdogan (1990)[130]
MIR *Minimum information ratio*	$\frac{1-\|\|\theta_{m+1}-\theta_m\|\|}{\|\|\theta_m-\theta_{m-1}\|\|}$	
AMIR	$MIR(K)(LL(K) - LL(1))$	Windham (1991)[131]
NEC (*Normalised entropy criterion*)	$\frac{E(K)}{LL(K)-LL(1)}$	Celeux (2006)[117]
ICL (*Integrated classification likelihood*	$-2CL^*(K) + \frac{\eta(K)}{2}\ln(N)$	Biernacki (1999)[132]
MML (*Minimum message length*)	$-2LL(K) + \eta(K)\ln(N)$	Oliver (1996)[133]
MDL (*Minimum description length*)	$-2LL(K) + \frac{K}{2}\ln\frac{N}{12} + \frac{\eta(K)+K}{2} + \frac{\eta(K)}{2K}\sum\lim_{k=1}^{K}\ln\frac{Nr_k}{12}$	Rissanen (1989)[134]
L	$\frac{N}{2}\sum\lim_{k:r_m>0}\log\left(\frac{Nr_k}{12}\right) + \frac{k_{nz}}{2}\log\frac{N}{12} + \frac{k_{nz}(M+1)}{2}$	Figueiredo (2002) [110]

Table 3.1: An overview of information criteria

In this equation, the first term corresponds to the estimated priors, the second to the estimated means, and the third to the estimated parameters of covariance matrices. If we had used diagonal covariance matrices instead, the number of free parameters would have been lowered from $\frac{K(K+1)}{2}$ to K for each covariance matrix.

The MIR criterion measures how the data are able to model densities of components. $(\theta_m, m > 1)$ is a sequence of parameters generated by the EM algorithm. The entropic criterion NEC is derived from the linking between the maximum likelihood (ML) and the classification maximum-likelihood (CML) approaches. It is defined as follows: $LL(K) = CL(K) + E(K)$. The entropic term $E(K)$ measures the overlapping of the clusters.

The *minimum encoding length criteria* (MDL, MML and L) estimate parameters so that they minimise the length of the message Length$(\theta, \vec{v})$. Message length consists of two parts, Length$(\vec{\theta})$ and Length$(\Delta|\vec{\theta})$, i.e. Length$(\vec{\theta}, \Delta)$ = Length$(\vec{\theta})$ + Length$(\Delta|\vec{\theta})$. The first part, Length$(\vec{\theta})$, is the length needed to estimate parameters $\vec{\theta}$ (a priori unknown for dataset Δ), and the second part, Length$(\Delta|\vec{\theta})$, is the data code length.

Biernacki in his study [135] *compared the performance* of these information criteria. Performance of all criteria strongly depends on the fitted data, especially the NEC criterion [118] as well as the criteria based on the classification likelihood (CLM, CLM2, CL and CL2). In case of equal mixing proportions, AIC and ICOMP criteria show the tendency to overestimate the right number of components, while BIC and AWE tend to underestimate the number of components. In case of different mixing proportions, AIC, BIC and ICOMP highly overestimate the number of components, while CLM, AWE and NEC perform better.

There are also different methods [136, 120] based on the mutual relationship between the components. Mutual information measures the statistical dependency of the components. The mutual relationship between components i and k is:

$$\phi(i,k) = p(i,k) \log_2 \frac{p(i,k)}{p(i)p(k)}, \tag{3.5}$$

where

$$p(i) = \frac{1}{N} \sum_{n=1}^{N} \lim f_i(\vec{x_n}), \tag{3.6}$$

$$p(i,k) = \frac{1}{N} \sum_{n=1}^{N} \lim f_i(\vec{x_n}) f_k(\vec{x_n}). \tag{3.7}$$

This criteria can be effectively used in gGMM as described in [120].

The general *NMF theory* [33] assumes that there exists a rough idea of how many components are in the mixture system. Components are initialised with high variances and with random centres. The algorithm forms a new concept or eliminates an old one after a fixed number of iterations to find the optimal number of components [52]. After the optimal number of clusters are found, the parameters describing the concept can be changed from a general Gaussian distribution to a more precise one (e.g. a parabolic shape) that will further increase the overall log-likelihood [104].

Split and merge algorithms can also be used for assessing the optimal number of components. These algorithms can work either in an agglomerative or divisive way. The drawback of the agglomerative approach is the necessity to know the maximum number of components to be expected k_{max} [137]. In [110], an agglomerative method is described

where components are initially generated with random centres and the same covariance matrices (number of components > expected number of clusters in the data). These components are iteratively merged until the stopping criterion L is met.

An opposite approach of working in divisive way is proposed in [126]. Components to be split are found via a multivariate normality test based on the Mahalanobis distance of each sample measurement vector from the component centre to which it belongs. In each step, the cluster that deviates the most from the Mahalanobis distance distribution, is split. When no cluster deviates from this distribution, the optimal number of clusters is found.

3.3 The proposed gmGMM algorithm

In the *proposed gmGMM algorithm* (Gaussian mixture model with merging), I deal with the problem of prior knowledge by considering that the information about the number of components is not always accessible to humans as they perform various tasks. In many cases, a human has no initial idea of how many clusters there are in a given dataset. This problem can be tackled by using the *greedy GMM*, which avoids the necessity of knowing the upper bound of the number of components and tests the criteria for all possible numbers. The algorithm creates one highly fuzzy component and sequentially adds the others to find the optimal number of clusters. The *merging of clusters* can be done from time to time, as suggested in [52]. This idea is implemented in the proposed algorithm in following manner. When the algorithm stops adding new clusters, the dependent components are sequentially merged. Let us look at the algorithm in a greater detail in the next chapter. Figure 3.1 shows an overview of the proposed gmGMM algorithm.

Initialisation of component centres and covariance matrices

In the GMM, convergence to the *local optima* is a frequent reason why the algorithm fails to find the best partition of the data (the correct number of clusters). The *diagonal initial variance matrices* are widely adopted to suppress any direction preferences. The newly added component is *initialised vaguely* (with a large variance), which corresponds to the high fuzziness of a newly searched concept. Concepts that are already included in the mixture should also be made more vague than their previous convergence.

The newly added GMM components are initialised in the following manner. At the start, the mean vectors of K_{min} components are initialised to randomly chosen data points from the sample dataset. The component priors are set to $\frac{1}{K_{min}}$ and the covariance matrices are initialised as:

$$\vec{S_k} = \mathrm{Tr}(\mathrm{cov}(\vec{v})), \text{ for } k = 1, \ldots, K_{min}. \tag{3.8}$$

Components that are added to a mixture subsequently at each iteration are initialised in the same way. The initialisation of a newly added component is described in Algorithm 3.

The number of components

The general idea of the proposed technique is described in the flowchart shown in Fig. 3.1. The whole algorithm is presented in Algorithm 4.

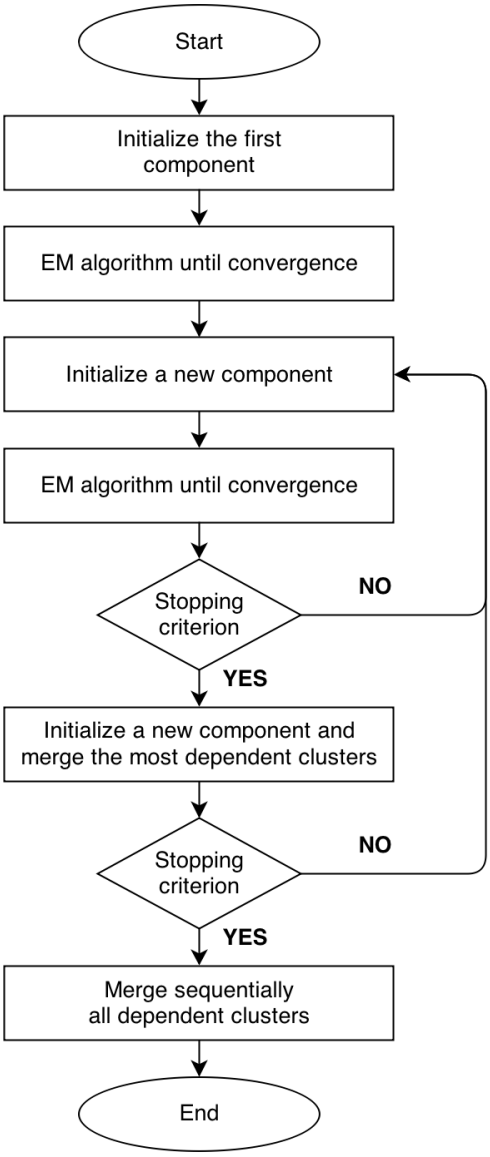


Figure 3.1: The general scheme of the proposed algorithm.

Initially, k_{min} components are initialised and classical EM steps are performed (see Eqs. (2.9–2.12)). A new component is added after each iteration until the selected stopping criterion is not met (see Algorithm 4). In the opposite case, a local convergence of the EM algorithm is avoided to allow the improvement. This is accomplished by initialising a new component and then removing the most dependent one. The *mutual information* (cluster statistical dependency) adopted from Ueda [111] (see Eq. (3.5)) serves as a merging criterion. Centres and covariance matrices of the merged components were computed as proposed in [111]. The initial parameter values for the merged model (r_m, $\vec{S_m}$ and $\vec{m_m}$) are set as a linear combination of the original ones before the merge:

$$r_m = r_i + r_j \tag{3.9}$$

Algorithm 3 Initialisation of a new component by initTech "random"

Inputs:
the input data $\vec{v}$, approximated model parameters $\vec{\theta}$ (cluster centres $\vec{m}$, covariance matrices $\vec{S}$ and mixing proportions $\vec{r}$)
Input parameters:
the number of components already included in a mixture K, a number of dimensions d
Output:
the new approximated parameters of mixture model $\vec{\theta_{new}}$

$\vec{m_{K+1}} \leftarrow$ random data point from a dataset $\vec{v}$
$\vec{S_{K+1}} = \mathrm{Tr}(\mathrm{cov}(\vec{v}))$
$r(K+1) = \frac{1}{K+1}$
$\vec{\theta_{new}} = \{\vec{m_1}, \ldots, \vec{m_{K+1}}, \vec{S_1}, \ldots, \vec{S_{K+1}}, \vec{r}\} \leftarrow$ add a newly initialized component into a set of approximated model parameters

$$\vec{m_m} = \frac{r_i * \vec{m_i} + r_j * \vec{m_j}}{r_m} \tag{3.10}$$

$$\vec{S_m} = \frac{r_i * \vec{S_i} + r_j * \vec{S_j}}{r_m} \tag{3.11}$$

The *partial EM algorithm* can be used for reestimating parameters to make the total algorithm more efficient [111]. After the EM steps (Eqs. (2.9–2.12)) are performed, the selected stopping criterion is tested once again after the EM convergence. If the stopping criterion is not met, we accept the newly proposed mixture and its parameters $\vec{\Theta_{new}}$ and continue with adding new components to the mixture. Otherwise, we reject it and stop the algorithm with the previous set of parameters $\vec{\Theta_{best}}$. This process avoids stacking in the local minima and can be done repeatedly.

The selected stopping criterion should *slightly overestimate the number of clusters* to enable the *consequential reduction* of the number of clusters (dependent clusters are merged). As a testing criteria we have compared AIC, BIC, AWE and the likelihood L proposed by Figuiredo [110] (see Table 3.1 for computational details). These include both underestimating and overestimating criteria.

In the final stage, when the stopping criteria are met or the maximum number of components k_{max} is reached, all dependent components can be merged in one iteration or alternatively, they can be merged sequentially until there is no dependent cluster left in the mixture (the parameters of the merged components are computed using Eqs. (2.9–2.12)). Finally, the classical EM steps are performed until the convergence to get the final estimates of mixture parameters.

3.4 Computational complexity of the proposed algorithm

Convergence of EM algorithm is guaranteed at each iteration. In addition, removal or addition of a component is done only when: $F(\theta_{n+1}) \geq F(\theta_n) + \epsilon$, where $F = -2LL(\theta) + P(K, N)$ is a criterion function. Therefore F is nondecreasing during the algorithm and the convergence of the algorithm to some stationary point is guaranteed although the convergence to the global optima is not guaranteed.

Algorithm 4 An overview of the gmGMM (greedy GMM with merging) algorithm

Inputs:
the input data $\vec{v}$
Input parameters:
the initialisation technique $initTech$, minimum number of components $minCmp$, maximum number of components $maxCmp$, stopping criterion $stopCrit$, threshold T, number of initialisation candidates when dependency $nmbTrials$, $new = 1$
Output:
the mixture model at $\vec{\theta_{best}}$ (cluster centres $\vec{m}$, covariance matrices $\vec{S}$ and mixing proportions $\vec{r}$)

$\vec{v} \leftarrow normalize(\vec{v})$
$\vec{\theta} = \{\vec{m_1}, \ldots, \vec{m_{minCmp}}, \vec{S_1}, \ldots, \vec{S_{minCmp}}, \vec{r}\} \leftarrow$ initialize $minCmp$ components using an $initTech$
$f, l, ll, \vec{\theta} \leftarrow$ EM steps until convergence
if $minCmp == maxCmp$ **then**
  $stop \leftarrow 1$
else
  $stop \leftarrow 0$
end if
$\vec{\theta_{best}} \leftarrow \vec{\theta}$
while not $stop$ **do**
  $K \leftarrow$ number of components
  while new **do**
    $\vec{\theta} = \{\vec{m_1}, \ldots, \vec{m_{K+1}}, \vec{S_1}, \ldots, \vec{S_{K+1}}, \vec{r}\} \leftarrow$ initialize new component using an $initTech$
    $f, l, LL, \vec{\theta} \leftarrow$ EM steps until convergence
    if ($trial < nmbTrials$ **and** any dependent component) **then**
      $\vec{\theta} = \{\vec{m_1}, \ldots, \vec{m_K}, \vec{S_1}, \ldots, \vec{S_K}, \vec{r}\} \leftarrow$ merge the most dependent components (mutual information)
      $trial \leftarrow trial + 1$
    else
      $new \leftarrow 0$
    end if
  end while
  $\vec{\theta_{best}} \leftarrow \vec{\theta}$
  $stop1 \leftarrow$ test stopping criterion $stopCrit$: $stopCrit(iter) - stopCrit(iter-1) < T$
  if $stop1$ **then**
    $\vec{\theta} = \{\vec{m_1}, \ldots, \vec{m_K}, \vec{S_1}, \ldots, \vec{S_K}, \vec{r}\} \leftarrow$ merge the most dependent component (mutual information)
    $f, l, LL, \vec{\theta} \leftarrow$ EM steps until convergence
    $stop2 \leftarrow$ test stopping criterion: $stopCrit(iter) - stopCrit(iter-1) < T$
    if $stop2$ **then**
      $stop \leftarrow 1$
    end if
  end if
  if ($stop$ **or** $K \geq maxCmp$) **then**
    while there is any dependent component **do**
      $\vec{\theta} \leftarrow$ merge the most dependent components (mutual information)
      $f, l, LL, \vec{\theta} \leftarrow$ EM steps until convergence
    end while
  end if
  $iter \leftarrow iter + 1$
end while

Analysis of convergence rate of the algorithm is beyond the scope of this thesis since it depends on many parameters such as overlap of the components, initialization technique for adding a new component, number of potential candidates for new component, etc.

Algorithm 5 The testing stopping criterion "stopCrit"

Inputs:
the criterion value at each iteration $CritValue$, log-likelihood at each iteration $\vec{LL}$, approximated model parameters $\vec{\theta}$ (cluster centres $\vec{m}$, covariance matrices $\vec{S}$ and mixing proportions $\vec{r}$)
Input parameters:
the number of components already included in a mixture K, number of data points N, threshold T, iteration $iter$
Output:
the decision of whether to stop adding new components or to not $stop$

$CritValue(iter) = -2 * LL(iter) + P(K, N, \vec{r})$ ← compute a criterion value
if $CritValue(iter) - CritValue(iter-1) < T$ **then**
$stop = 1$
end if

The computational complexity of each iteration can be derived from the fact that EM algorithm is a linear method. Therefore computational complexity of the proposed algorithm is $O(m * N)$, where m is a number of potential candidates for a new component and N is number of datapoints.

3.5 Experimental evaluation

Datasets

The selected algorithms were tested on *artificially generated datasets* S_1-S_4 adopted from [138]. These are synthetic 2-D data with 5000 vectors having 15 predefined Gaussian clusters with varying complexity in terms of spatial data distributions. With an increasing index number of the dataset, the degree of the clusters overlap increases, and thus the task complexity also increases. We have further tested the algorithms on the *real-world 4-d dataset Iris* (150 instances, three classes) and on the *real-world 16-d dataset Letter recognition* (20 000 instances, 26 classes) [139]. The character images (black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet) in the Letter recognition dataset were based on 20 different fonts and each letter within these 20 fonts, was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15.

All data were *normalized* using a standard score normalization:

$$x_{n,\vec{norm}} = \frac{\vec{x_n} - \text{mean}(\vec{x_n})}{\text{std}(\vec{x_n})} \tag{3.12}$$

($\vec{x_n}$ – each data point; mean($\vec{x_n}$) – the average of all the sample data points; std($\vec{x_n}$) – the sample standard deviation of all the data points)

Comparison of a greedy and normal GMM

Initially, the greedy and normal GMM algorithms were compared to verify results in [114] and to see whether the greedy approach is more effective than the conventional EM algorithm (measured by computational complexity and achieved accuracy). The comparison for all datasets can be seen in the Figure 3.2. Both algorithms were initialised randomly and run 20 times. To compute the accuracy, each cluster is assigned to the class that appears most frequently in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned data points and dividing this by the total number of data points.

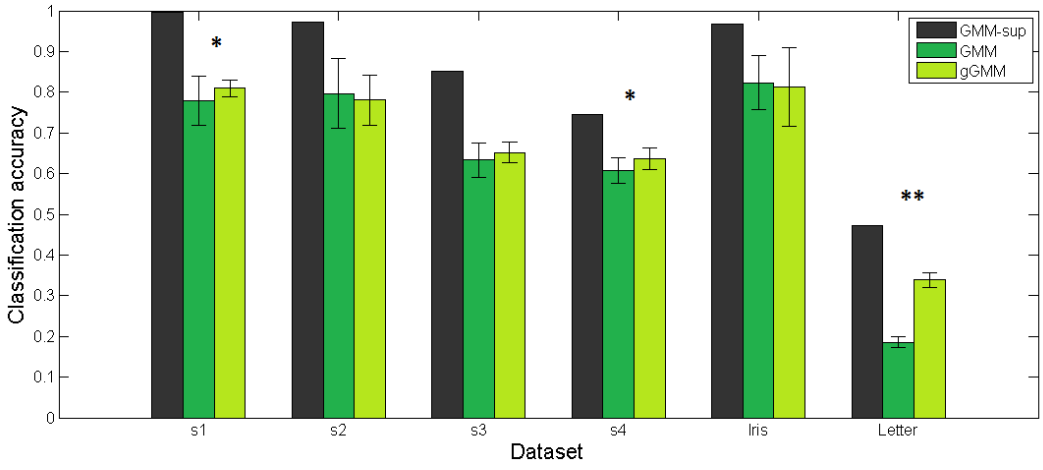


Figure 3.2: Comparison of greedy and normal GMM: Both greedy (gGMM) and normal GMM (GMM) algorithms are initialised randomly, results averaged over 20 repetitions. Algorithms are compared also to the supervised GMM (GMM-sup). The mean and standard deviation from 20 repetitions is visualised. * (resp. **) signifies that gGMM and GMM differ on the significance level p=0.05 (resp. p=0.01) (pair-sample t-test).

The greedy algorithm (gGMM) performed significantly better than GMM for artificial datasets S_1, S_4 (p=0.05, pair-sample t-test) and in the real-world dataset Letter recognition (p=0.01, pair-sample t-test). For other datasets, the difference was insignificant.

Number of components

The algorithm ability to find the optimal number of clusters with different stopping criteria (AIC, BIC, AWE and likelihoood L proposed by Figueiredo [110]) was tested. A number of components found for different datasets is visualised in Fig. 3.3 (averaged over 20 trials).

The best results were achieved for the AIC criterion followed by the L criterion. The BIC and AWE criteria resulted in a strong underestimation of the number of clusters, because they tended to underestimate the correct number of clusters even without the final merging.

The performance of the proposed algorithm was compared to the algorithm proposed by Figuieredo [110], to the greedy algorithm proposed by Verbeek [113], and to the greedy merge learning algorithm (GMEM) proposed by Li [112].

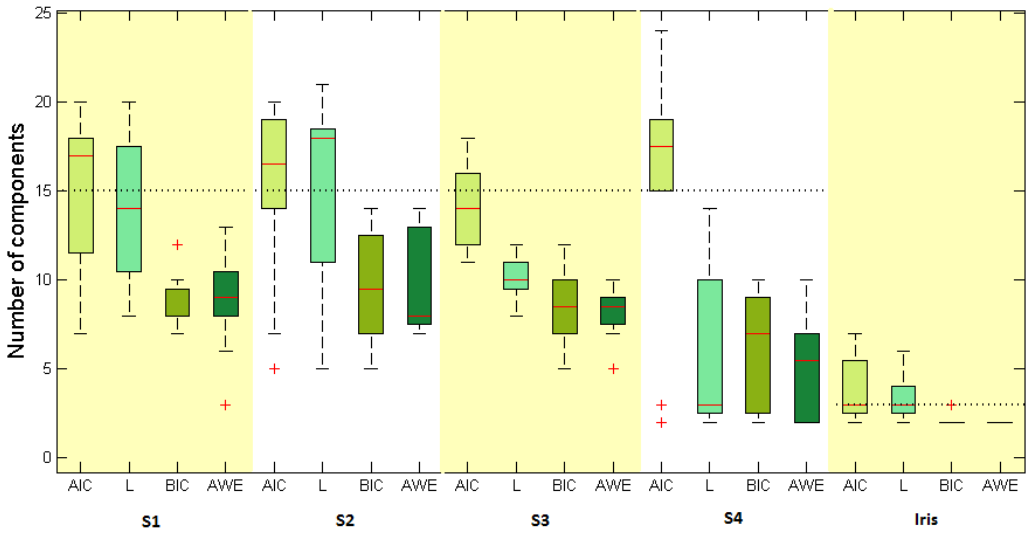


Figure 3.3: An ability of gmGMM algorithm to find the optimal number of components when integrating different stopping criteria: Results are visualized for artificial datasets S_1-S_4 (averaged over 20 trials) and real-world dataset Iris (averaged over 100 trials). Dashed lines denote the correct number of components. (On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

It is important to notice, that the GMEM algorithm [112] was not applicable in the version presented in the article because there are several local optima on the AIC (and also L criterion) curve during the merging process and the algorithm tends to stack as it detects an abnormal number of clusters. The GMEM algorithm had to be modified to overcome the small local optima. The algorithm was modified by adding ϵ into the stopping criteria S (AIC or L stopping criterion) so the algorithm stops merging components when: $S(i) > (1+\epsilon) * S(i+1)$, $\epsilon = 0.01$, where i goes from the initial number of components (100) to 0.

The results for all compared algorithms are visualised in Fig. 3.4.

The achieved results for all tested algorithms were compared using the standard deviation from the theoretical correct value (for the artificial datasets, the theoretical correct value was 15; for the real-world dataset Iris, it was three; and for the Letter dataset it was 26). The standard deviations for all datasets and both stopping criteria can be seen in Fig. 3.5.

The proposed gmGMM algorithm outperformed all compared algorithms (gGMM, Figueiredo and GMEM) for datasets S_1-S_3 for both stopping criteria. The gGMM algorithm proposed by Verbeek using the L stopping criterion strongly overestimated the number of components in the data, more so for well separated clusters (dataset S_1, S_2). On the other hand, for the AIC stopping criterion gGMM algorithm achieved very good results (for dataset S_4 it performed best among the compared algorithms). There was no correction used as was in GMEM algorithm. When the stopping criterion L was modified the same way as in GMEM algorithm, the gGMM algorithm achieved better results than GMEM algorithm. The algorithm proposed by Figueiredo also strongly overestimated

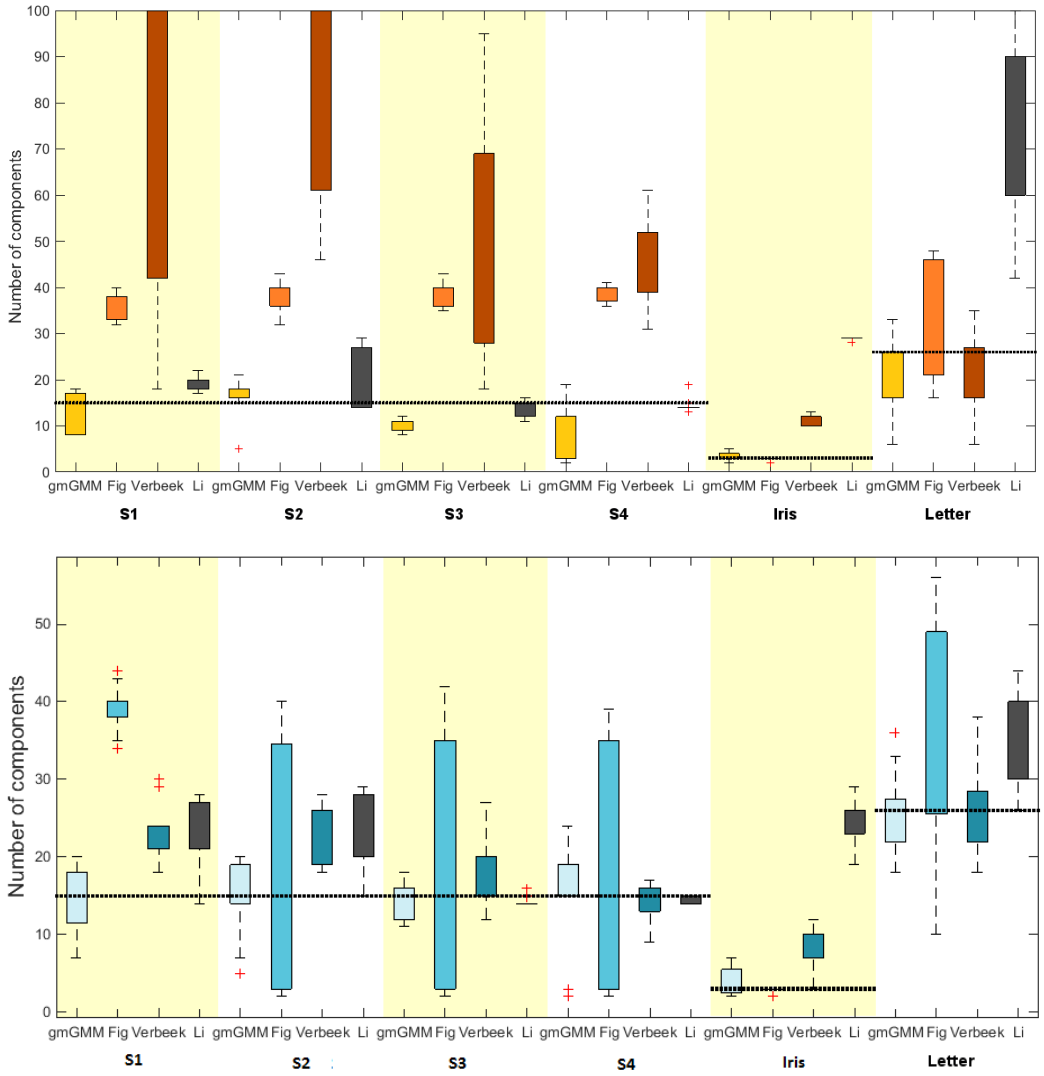


Figure 3.4: Comparison of different algorithms for finding optimal number of components in GMM: Results are visualized for artificial datasets (averaged over 20 trials) and real-world dataset Iris (averaged over 100 trials). Dashed lines denote the correct number of components. Compared algorithms are: gmGMM (proposed algorithm), Verbeek (gGMM algorithm proposed by Verbeek [113]), Fig (merging algorithm proposed by Figueiredo [110]) and Li (a greedy merge learning algorithm proposed by Li [112]). As a stopping criteria are used AIC (upper) and L stopping criterion [110] (lower) (on each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually).

the number of components in data and was outperformed for all artificial datasets by both gmGMM and GMEM algorithm for both AIC and L stopping criteria. It was also outperformed by gGMM algorithm when the AIC criterion was used.

For the real-world dataset Iris, the Figueiredo algorithm achieved the best results for both stopping criteria. The number of clusters detected (averaged over 100 trials) was: 2.7 (0.7), the classification error: 18(10)%, followed by gmGMM algorithm resp. gGMM

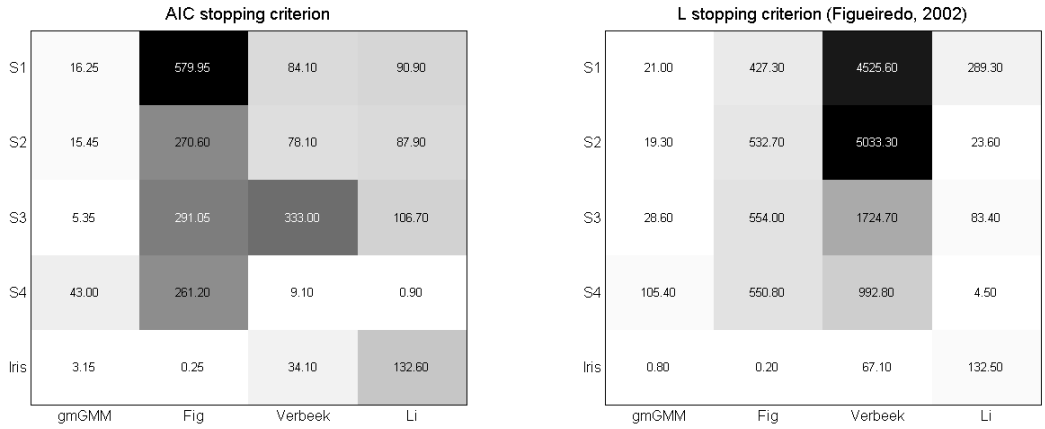


Figure 3.5: Finding optimal number of components in GMM (standard deviations): the achieved results (see Figure 3.4) were compared to the correct theoretical value using a standard deviation: standard deviations for all datasets and compared algorithms for AIC criterion (left) and L stopping criterion (right).

algorithm for L resp. AIC stopping criterion.

The question that arises is why did we use the modified version of the stopping criterion for the GMEM algorithm and not for the gGMM algorithm. The reason for this is that we would not be able to compare the GMEM algorithm with the others without modification, because they stack in the local optima after two or three iterations. On the other hand, for the gGMM algorithm, the learning curve of L criterion was smooth and by using the modified version of the stopping criterion, we would only said that we are not interested in the small improvements of the criterion. The results for gGMM confirm that the L criterion overestimates the number of clusters.

Also, the time efficiency of the algorithms is not directly comparable. The resulting time and the number of EM iterations is dependent on the initial and final number of components. The initial number of components for Figueiredo and GMEM algorithms further depends on the number of data points in a dataset.

The evolution of models is visualised in Fig. 3.6.

3.6 Summary to novel gmGMM algorithm

This section presented the greedy Gaussian mixture model with merging, which is the algorithm that is capable of finding the optimal number of components in the mixture without any prior information [23]. The novel step in the greedy algorithm allows the improvement of performance when the stopping criterion is met, as the most dependent component is removed and the new component is initialised. The second novelty lies in the merging of all the dependent clusters in the final stage.

Enhancement of the method, which detects the optimal number of clusters, led to an improvement of its effectiveness compared to the method based on stopping criteria. This can be seen in more accurate estimation of the number of clusters and the lower classification error for most of the datasets. The effectiveness of the proposed method

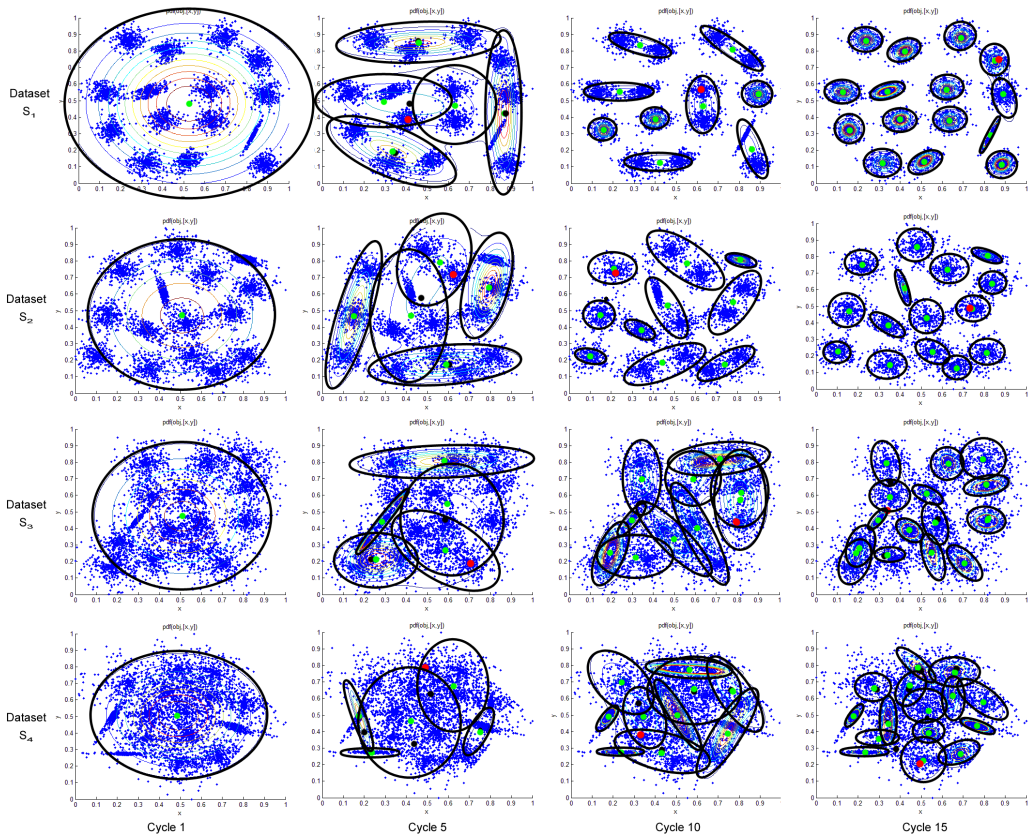


Figure 3.6: Initialisation and algorithm progress: Here there are shown normal distributions of components with the black line for covariances. The red point shows the initial position of the newly inserted component. Green points are final positions of centres after the EM algorithm.

originates from the fact that it does not have to go through all possible numbers of components.

Initially, results in [114] were verified and it was shown that the gGMM algorithm is generally more effective than the conventional EM algorithm. The comparison of the achieved accuracy for both real-world and artificial datasets can be seen in the Fig. 3.2.

In the second stage, the properties of the proposed gmGMM algorithm were investigated in detail. Its ability to find optimal number of components in data for different stopping criteria was compared. The best results were obtained for AIC criterion in highly overlapping mixtures, while it was more effective to use mutual information criterion without final merging for well separate clusters.

The final components in the merging process should be treated carefully. The administration of this process in one final step should lead to the deletion of important components. Hence performing the final merging sequentially is recommended. This will lead to little increase in time demands but the classification error will be lowered significantly.

Finally, the performance of the proposed algorithm to other similar algorithms was

compared. The deviation from the correct theoretical number of clusters was used as the comparison criterion. The proposed gmGMM algorithm achieved the lowest error variability (compared to other algorithms) for most of the datasets.

Part II

Multimodal cognitive architecture for language acquisition

Chapter 4

Existing cognitive models of vision and language

Brain is studied by neuroscientists on different levels of processing – starting from molecular to psychological level [56], which corresponds to appropriate computational cognitive models. These models are not focused mostly on all levels of processing simultaneously. They rarely incorporate all cognitive modalities and rather focus on processing only one of them. Furthermore, current models of language and vision can be divided based on the used computational framework to connectionistics, probabilistic and symobolic models. Anyway, this division is definitely not strict since some models tend to use a combination of computational frameworks, try to incorporate more levels of signal processing or combination of modalities.

4.1 Main computational cognitive models of vision

The vision is *the most important sense* for human (in contrast to many animals) and approximately half of the cortex is involved in vision. What makes the vision difficult is the large number of objects in the surrounding world (approximately 20 000), which in addition have many specific features such as orientation, texture, shading etc. The development of infant vision is shown in Figure 4.1.

The *brain areas* responsible for processing separate parts of the visual information were detected both by studying effects of brain lesions in human beings and primats and by neuroimaging methods. The most important findings of the exhaustive neuroscience research are that in the brain there are apart of primary visual cortex also separate higher visual centra responsible for processing colour, movement, shape, face recognition, orientation etc., all retinotopically organized. The signal which is captured by our eyes is firstly processed by retina's fotoreceptors and ganglion cells. These cells are collected by optical nerve which transfers the signal to the primary visual cortex in the occipital lobe (the left half of the field of vision is processed by the primary visual cortex in the right hemisphere). The information is further processed by two distinct neural pathways both responsible for specific tasks: *ventral pathway* which extends to the temporal lobe is responsible for processing the information about the shape, colour and other information important for object recognition (also known as a WHAT pathway) and *dorsal pathway*, which extends to the parietal lobe is responsible for processing the information about the position and movement of the objects (known as a WHERE pathway) [140]. Vision is fully developed to adults' level between ages 3 and 5, which corresponds to the myelinization of neural fibres, development of photoreceptors or synchronization of muscles controlling eye movements.

As a response to the progression in the neuroscience research, the focus to cognitive

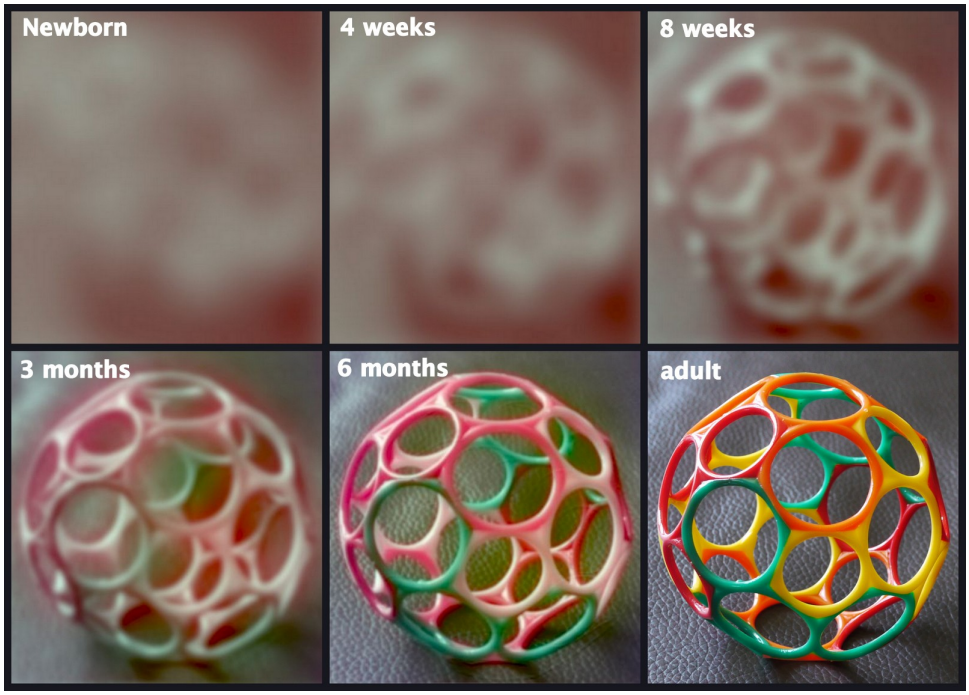


Figure 4.1: The development of infant vision: the vision progresses from seeing just fuzzy blob at birth to being able to see the toy as it is by 10 month. Kids start also to be more interested in toys by 3 months, which can be connected to their vision development. Source: "https://lasermom.wordpress.com/2012/06/24/infant-vision-research/".

models of vision have been gradually shifting: from the models primarily inspired by the psychological research to the neuroinspired cognitive models.

Early attempts to model vision were mainly based on *template matching*. The problem of such an approach is that you need as many templates as many possible objects should the model learn and different viewpoints must be covered by many templates. Current state of the art object recognition systems [141, 142] are primarily based on local image descriptors. These can be distinct features (such as edges, colours,...) which are composed together and are invariant under transformations.

"Neocognitron" [143] is the model of object recognition where hierarchical neural network is used to achieve the position-invariance, the invariance to moderate changes in size and orientation and also the invariance to a moderate noise. The limitations of the model are that it is not fully view-invariant and is designed only for a limited domain (text recognition). In the SEEMORE [144], an extension of Neocognitron, more feature detectors are used and the training set is variable. The model is able to recognize objects across changes in position, size, orientation, noise, etc.

Following extensions of cognitive models focused on the *biological plausability of specific features*. The new features should match the nowadays neuroscience knowledge and be comparable to features, which are processed by neural pathways in primary and higher visual cortices. The examples of these features are MEX or C2 features [145, 146]. There were also attempts to extend *invariance of the features* to be able to recognize objects seen from different viewpoints and to detect objects which are representatives of

the same class (e.g. the class dog) [147, 148, 149, 150].

Various sets of *visual descriptors*, which are rich descriptors of objects, were created: SIFT (scale invariant features) [151], HoG, Haar features, Spin Images over 3D point clouds [152], or kernel descriptors [153, 154] (which have been shown to be equivalent to a type of match kernel that performs similarly to sparse coding [142, 155] and deep networks [156] on many object recognition benchmarks). Features of the image are computed after segmentation of the image. There are various *segmentation methods* ranging from simple thresholding method, clustering methods, methods based on histograms or detected edges to more sophisticated methods such as methods using graph partitioning or solving partial differential equation. The selected type of segmentation plays an essential role in object recognition and it will, for example, influence the sharpness and softness of detected objects boundaries.

Special attention is given in object detection to the so called *occlusion problem*. Ogale and Aloimonos [157] take advantage of torque and FFT of descriptors to detect the occluded moving objects. They used normalized cut for scene segmentation, colour and texture are used to define edge pixels and motion is used to detect object boundaries.

Many cognitive models of vision have taken advantage of neural networks. *Neural networks* have shown good performance in many pattern recognition tasks [158] such as object recognition and classification [159] or hand-written letters recognition [160]. Their disadvantage is that they need many training examples to adjust weights. On contrary, humans are in some cases able to learn from very limited number of examples.

In recent years, *probabilistic models of vision* have become increasingly popular [28, 161, 162, 163, 164]. These models are taking into account the necessity of *compositionality/modularity of the system*, which enables an unsupervised incremental learning of the visual scene. Thanks to them, traditional theories of visual perception have been revolutionized, ranging from low-level models such as shape perception or motion prediction to higher-level models, which resamble probabilistic parsing in natural language and operate over hierarchically organized representations of objects (generated by probabilistic grammar for natural scenes) [165, 166, 167, 168].

Yuille in [163] deals with the enormous complexity and ambiguity of the images using a Bayesian inference on structured probability distributions, and that use "analysis by synthesis" strategies with intriguing similarities to the brain. Moghaddam and Pentland in [169] applied probabilistic models (probability densities – multivariate Gaussian and mixture of Gaussians) to the *visual search* and *target detection* for automatic object recognition and coding, especially to the probabilistic visual modeling, detection, recognition, and coding of human faces and non-rigid objects such as hands. Bayesian vision system was also used as a model of vision when modeling intuitive physics for scene understanding [170].

4.2 Main computational models of language

In this chapter, I will summarize main computational models of language. Firstly, I will briefly mention the current knowledge about language acquistion, processing and development in human brain to show challenges which every cognitive model must face.

The development of *language perception* and *production* is visualized in the Fig. 4.2. As can be seen, ability of perception precedes the production by few months. Baby starts

with learning phonems (sounds) at 2 months, followed by learning words (capability of sound segmentation is very important in this stage) at 8 months, the first words are produced at 12 months, and grammar is last to occur (at 2 years). The increased abilities correspond to the myelination (the process by which a fatty layer, called myelin, accumulates around axons, which increases speed and specificity of neural transmission) of axons which is reflected in a rapid increase of the brain weight during the first year (from 350 g at birth to approximately 1 kg at 1 year, adult brain weight 1350 g is reached by 15 years) [171], see Table 4.1 for detailed information.

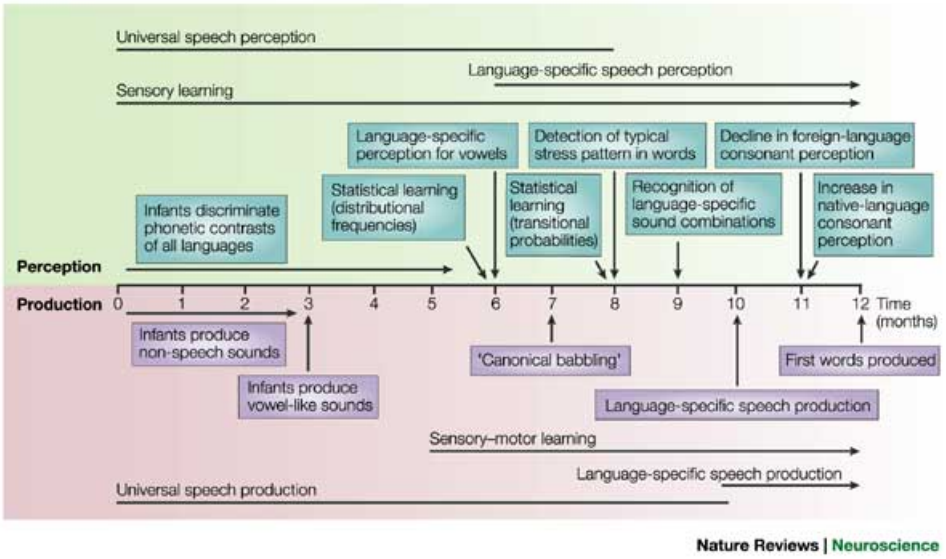


Figure 4.2: Timeline of speech-perception and speech-production development: changes in human infant during the first year of life [172].

We should keep in our mind, that different cognitive areas in brain are not maturated at the same time. Neural pathways important for the language processing (acoustic areas) achieve its developmental maxima (fibres become fully myelinated) few months later than pathways responsible for processing visual information. *Myelination* is like a slow wave, which spreads in each lobe of cortex gradually. In each lobe, the first areas to myelinate are the primary areas (the classical projection areas that mediate functions in one specific sensory or motor modality), followed by unimodal and multimodal association areas (some are not fully myelinated until the late adulthood) [174].

There are *4 basic language skills* needed for complete communication: language comprehension, speech, reading and writing. Two well-defined areas in the brain are considered to be the most important for the language processing: Wernicke's area (responsible for the language comprehension) and Broca's area (responsible for the speech production). These areas communicate directly with the tonotopically organized primary auditory cortex and are usually located in the dominant hemisphere (the left hemisphere for 97% of people). However, the less-dominant hemisphere also participates in language processing and can fully adopt language skills when the left hemisphere areas are removed before the age of two (this ability weakens strongly after the age of six) [175]. Broca's area further transmit the signals from primary auditory cortex and communicate with

Age	Language production	Brain weight
5-7 months	Sounds similar to speech	660 g
7-8 months	Putting syllables together (babbling)	750 g
10 months	Babbling similar to speech flow (intonation)	800 g
1 year	First words	900 g
18 months	Increased speed of words learning, combinations of words	1050 g
2 years	Complex phrases, origin of grammar (articles, prefixes,. . .)	1100 g
3 years	Correct usage of grammar, advanced language construction, fluent conversation, only minor and systematical errors (”why did he dis it appear?”)	1270 g

Table 4.1: Brain weight and corresponding language production during the first years of life [173].

the primary motorical cortex (see Fig. 4.3).

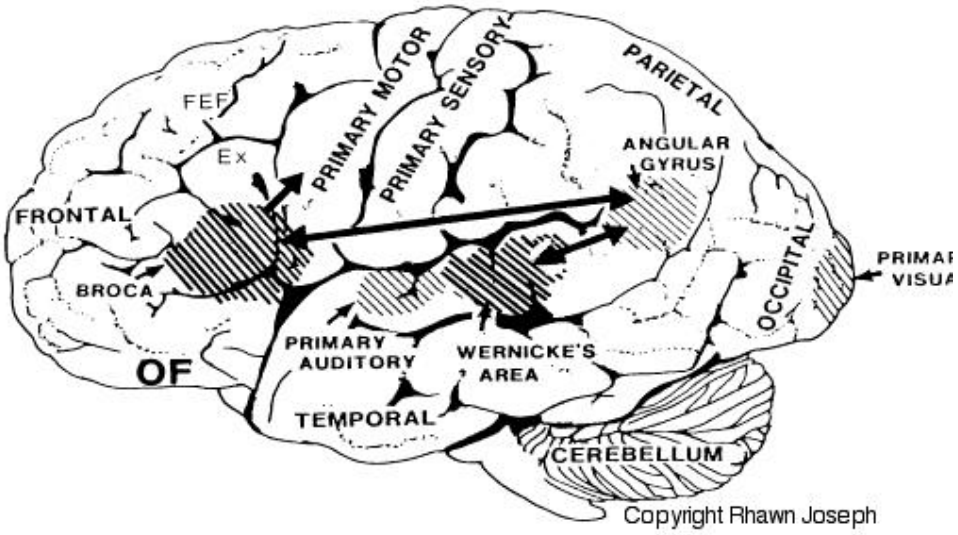


Figure 4.3: The most important areas for language processing [176].

The neuroscience research which in last two decades massively uses neuroimagining methods did not bring any clear conclusion about the *location of the semantic system* (the system that brings meaning to all verbal and non-verbal stimuli). There are two main theories concerning this topic.

The first theory is that apart of processing of separate modalities (e.g., colour/smell/shape of an object) there is also a transmodal ”hub”, which activates the intermodal-

ity information. Neuroscientific experiments provide some evidence that this "hub" could be supported primarily by the regions within the anterior temporal lobe, bilaterally [177].

The second school believes that semantics is basically just the world representation in the brain (so the semantics of a cup would be the visual representation of the cup in the visual areas, the affordance represenation in motor and somatosensory cortex, etc.) and no such "hub" exists, because even this multi-modal conceptualization cannot provide coherent, generalizable concepts and other additional inter-modalities processes are necessary [178]. Binder's meta-analysis of 120 fMRI studies [179] focusing on semantic processing showed some higher activated regions which form a distinct, left-lateralized network, but mainly points out that the processing is highly distributed in the brain and does not come to any consensus about the identity of the neural systems that should store and retrieve semantic information.

As with the models of vision, progression in the neuroscience research had strongly influenced the cognitive models of language. Models had been gradually shifting from symbolic models to neural networks and probabilistic models and these all different types of models have been recently combined on different levels of processing. There are wide range of domains in language processing where specific models should be applied. These are processing incoming signals (speech processing) [180], word recognition, phonology, morphology, syntax, lexical semantics and language acquisition [28]. The main *computational models of language* are well summarized in Chater [181]. The basic frameworks are:

1. **Chomskian (symbolic) models** are early models where the knowledge of the language is embodied in a set of *declarative rules* and a set of *processing operations*. Two main processes are *parsing* (used to find a syntactic derivation that yields to the observed sequence of words) and *production* (uses the rules to construct a derivation and output the resulting sequence of words). These models are unable to fully model psycholinguistic data which indicate that purely structural features of language are just one othe factors and experimental results are also influenced by probabilistic and world-knowledge factors. Among the first symbolic models of words is the Forster's model [182]. Forster implemented the model where symbolically represented word forms are matched against acoustic or visual input. In this model, the sequential search in memory is necessary. Parallel search of multiple word forms was proposed (but not implemented) by Morton [183].

   Symbolic models of sentence processing have been extensively used in computational linguisitics [184, 185, 186, 187]. Nowadays, are often replaced by (or combined with) connectionistic and probabilistic models. Vasishth in [188] divided these models into *grammar-based approaches*, which presuppose a syntactic theory [184, 189, 190], *symbolic models* involving complexity metrics and ambiguity resolution principles which can be defined without reference to a particular architecture for the competence grammar [191, 192] and approaches where *fixed cognitive architecture* is used as a starting point (for example READER model [193] or utilization of independent cognitive architecture such as ACT-R [194]).

   To explain *language acquisition*, Chomsky hypothesizes that child has a hypothesis space of candidate grammars and choose on the basis of experience one of these grammars [195]. His argument for this hypothesis is the *poverty of the stimulus*

(this includes, among others, absence of negative feedback and presentation of only positive examples) which should lead to the production of the overgeneral grammar (but we do not observe it).

2. **Connectionist models** are *bottom-up models* of learning. Connectionism is often used in a combination with a symbolic approach when modelling language processing while each of them models different level of processing (symbolic approach at a psychological level and connectionism at an implementation level) [196, 181, 197].

   There exist detailed cognitive models of speech processing, which capture a wide range of empirical data and have also made novel predictions (e.g. TRACE model [198] consisting of a sequence of layers for phonetic features, phonemes and words). Despite the doubts, these bottom-up connectionist models can also accomodate apparently "top-down" effects [199]. Among the first connectionist models of language processing was also the *Rumelhart's model* [200] where the connectionist model of the acquisition of the past tense in English is developed. These first models and their claims about the dispensability of rules was criticized by Pinker and Prince [201].

   The model by Christiansen [202] is a simple *recurrent network* (SRN) trained to integrate phonetic features with information about lexical stress and can capture also infant speech segmentation. Takáč [203] proposed a connectionist model of the language acquisition and sentence generation where the messages, which form the input to the network, are structured as sequences, so that message elements are delivered to the network one at a time. Models of reading aloud have focused mainly on naming single words – e.g. Sejnowski model NETtalk [204] or model of Seidenberg [205] who proposed the feedforward network for mapping from a distribubted orthographic representation to a distributed phonological representation which can achieve human levels of performance on both word and non-word pronunciation when trained on actual word frequencies.

   In sentence processing, there have been proposed two main classes of models. Models from the first class are related to *stochastic context-free grammars* and learn to parse "tagged" sentences [206, 207] where networks are trained on sentences associated with a particular grammatical structure and appropriate grammatical structures should be assigned to novel sentences [196]. The second class of models tries to learn syntactic structure from sequences of words using SRN [208, 209]. Besides simple recurrent networks, also two-route networks [210] or bidirectional fully recurrent networks [211] have been used to model processing of words and sentences in current models.

   Connection of symbolic models and connectionistic models is briefly mentioned in Vasishth [188]. Nowadays, the *parallelism* is included in all prominent symbolic cognitive architectures (SOAR [31], ACT-R [212] etc.) and also cognitive models of sentence processing involve parallelism. Furthermore, eventhough we consider symbolic cognitive models as systems that manipulate with discrete representations they are able to manipulate, in some degree, also with some continuous aspects. This means that distinction between purely "symbolic" or "connectionistic" model is being progressively decreased.

Over the past several years, *Deep Neural Networks* have shown remarkable success in many computer science areas such as image classification [159] and revolutionized the field of speech recognition [**?**]. These multi-layer networks were recently incorporated as an improvement to Google Voice transcription instead of GMM acoustic models.

3. **Probabilistic models** differ from the Chomskian tradition in one main point. They merely try to find any derivation but try to find the most probable derivation. The probabilities can be added to the existing linguistic rules to indicate usage frequency of rules or more competitive approach can be used when language structures itself are viewed probabilistically. The probabilistic models of language are well summarized in the work of Chater and Manning [28]. The state of the art speech recognition algorithms used in probabilistic models are *hidden Markov models* [213], *vector quantization* or *dynamic programming*. One of the leading speech recognition system *Sphinx 4* is based on fully-continuous hidden Markov models [214]. Bayesian word learning is described in the work of Xu and Tenenbaum [48]. Araki et al. [215] proposed the online version of multimodal latent Dirichlet allocation (MLDA) using *Gibbs sampling* for the multimodal categorization together with the unsupervised word segmentation method based on the hierarchical Pitman-Yor Language Model (HPYLM).

   Probabilistic models can be used for a theoretical analysis of connectionist models behavior. Their performance can be understood as depending on the regularity of *orthography-phonology mapping* at different levels of analyses (phonemes, n-grams, onsets/rimes,...) [181]. Furthermore, probabilistic models can produce predictions about pronunciation of non-words [216] and can also provide a model of optimal eye movements to maximize information coming into the reading system [217]. The question why we do not hear any of the possible partitions of the speech can be explained by the fact that we build a huge network of brain cells, which is capable of computing probabilities of possible partitions [218].

   The probabilistic model of sentence processing is a model that will *predict the following words on the basis of previous experience*. These models make a significant simplification when the probabilities of words are calculated on the previous n words but real language depends on entire sentence (or even texts). Models are trained on a *corpus* of language data. The models of the sentence processing engage *Bayesian mixtures*, HMM, *Suprisal theory*, *n-gram*-based models or *context-free phrase-structure grammar*. Probabilistic models of sentence processing have been used to study variations in observed corpus frequencies across languages [219, 220]. Regularities between words should be taken into account to capture the probabilistic influence of the lexical information as the computational parsing performance is substantially improved when the co-occurence of words is considered [221, 222]. The model of general knowledge [223] and "theory of mind" [224] should be also engaged into the models. Probabilistic symbolic models [188] are models where a set of symbolic rules is used to generate syntactic structures [185, 225]. The probability of these rules is computed using a corpus (a Penn treebank). This enables these models to capture the role of the experience and frequency in the language processing.

4.3 Language acquisition and symbol grounding

There is an ongoing debate in the *language acquisition* research on the question whether the langugage is acquired through learning and *interaction with environment* or there are *innate structures* in brain. The *behaviorist theory* proposed by Lado [226], Skinner [227] and Weinreich [228] describes language as an unconsious, automatic process acquired by the stimulus-response condition method. The *nativist theory* was proposed by Chomsky [195, 229] who posits that language abilities must be innate and these innate grammatical structures are evolved through interaction with the world. As mentioned above, his argument for the innate grammar is the poverty of stimulus and also the generativity of the language. The *cognitive theory* developed by Piaget [230] sees the language acquisition as a consciouss process (cognitive development preceeds language development as well as semantics preceeds syntax) composed of the following periods of development: Sensorimotor (birth to 18-24 months), preoperational (18-24 months to 7 years), concrete operational (7 through 12) and formal operational (adolescence through adulthood). These stages can be passed in different ages by some children, but any of them cannot be skipped. Another researchers such as Gopnik [231] gives much more emphasis to the human factor when parents helps to give sense of the situations to children.

A more modern theory is the *social-interactionist theory* [232] combining importance of social influences with Vygotsky's socio-cultural theory. Vygotsky defined the *zone of proximal development* in every learner which is "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" [233]. Social-interactionists criticize Chomsky's clame about lacking a negative feedback and corrections from parents on childrens' errors (see meta-analysis conducted by Moerk [234]). The importance of the interaction with the environment is also highlighted in the *Relational-frame theory* developed by Hayes and Barnes-Holmes [235, 236] who stated that psychological events are predicted and influenced by the environment. Opposed to Skinner they identified the derived relational responding (a particular type of operant conditioning to be found only in human). *Emergentism* [237] provides a theory of how language learners come to identify and prioritize the various competing cues such as word order, animacy or agreement that are relevant to sentence comprehension. Other theories of language acquisition are Usage-based theory [238], Optimality theory introduced by Prince and Smolensky [239, 240] or Native language magnet model [241, 242].

Chomsky argued that the *absence of the negative feedback* would lead to the production of the over-general grammar. In contrast, the probabilistic language acquisition hypothesis proposed for example by Hsu and Chater [243, 244] have shown that statistical methods are capable to learn restrictions of general rules only from the positive feedback [245, 246, 247, 248, 249, 250] when it is sufficient to learn language only with a high probability and within the statistical tolerance [251, 252]. Klein and Manning [253] combined the extended distributional phrase clustering model for learning word classes (where left and right word context is taken into account) with a dependency-grammar-based model. This model was able to learn high-quality parses from little unlabeled text from a wide range of languages with 80% accuracy.

The essential question in language acquisition is how symbols are anchored in some arbitrary symbols – this is a so called *symbol grounding problem* (SGP). This problem was firstly described by Harnad in his work [254] and since then several attempts to solve this problem have been presented. It has been shown how to learn new symbols using already grounded symbols and their combination (e.g. Cangelosi [4, 255]) and how it is possible to transfer the knowledge between agents (e.g. Vogt's model [256]) but the question how capacities of any agent to ground symbols are evolved at the first place is still unanswered.

Taddeo and Floridi [16] defined the *zero semantical commitment condition*, which must be satisfied by any model which claim to solve the symbol grounding problem:

- *no innatism* allowed (no semantic resources are preinstaled in the artificial agent)
- *no externalism* (no semantic resources are uploaded from outside)
- artificial agent has its *own capacities and resources* to ground symbols

Taddeo and Floridi [16] further divide current models to representationalist, semi-representationalist and non-representationalist approaches and show that all of them break some of the zero semantical commitment conditions.

- **The representationalist approaches** [254] such as *symbolic theft hypothesis* by Cangelosi [4], *functional model* by Mayo [257] or *intentional model* by Sun [258] use representations that cannot be presupposed without begging the question. This means that they presuppose availability of semantic capacities or resources that the approach is trying to show to be evolvable by an artificial agent.
- **The semi-representational approaches** such as *epistemological model* by Davidsson [259], *physical symbol grounding* by Vogt [256] or model utilizing *temporal delays and predictive semantics* [260] use representations while relying on principles imported from *behavior-based robotics*. These models cope with problems illustrated in a detail on the Vogt's model example. The Vogt's solution of SGP combines Harnad solution with situated robotics and semiotic definition of symbols (symbols are defined as pairs of sensorimotor activities and environmental data). The symbol grounding problem is transformed to the *physical symbol grounding problem* (grounds meaning of the symbols in the sensorimotor activities while the precedent models ground symbols only in sensoric domain) and is solved by usage of semiotic symbol systems and Guess game. *Guess game* [261] is a game where a common language is developed by two robots (speaker and hearer) in a common environment: the speaker firstly names an object and hearer tries to find the object by the trial and error. This leads (in 4 separate stages of the game) to development of a common semiotic symbolic system. Problems of this solution are the following: signs are meaningfull symbols in the eyes of the interpreter without begging the question and the guess game is not meant to ground the symbols. But the two agents share the same grounded vocabulary by iterated communication which only multiplies the number of agents who need to learn grounded symbols [16].
- **In non-representational approaches** such as *communication-based models* [262] or *behavior-based models* [263] only sensorimotor couplings are considered and

symbolic representations are thought to be unnecessary. These models face the problem that SGP is rather postponed than avoided: after developing even an elementary protolanguage and higher cognitive capacities it will have to be able to manipulate some symbols.

The conclusion of Taddeo and Floridi [16] is that the current models solve mainly the problem how to transfer the knowledge of grounded symbols among agents (breaking the condition of externalism) but does not solve how an artificial agent evolves such capacities at the first place (breaking the condition of innatism). The *valid solution* of the symbol grounding problem will need to combine at least: bottom-up (sensorimotor) approach, top-down feedback, representational, categorical and communication capacities of artificial agents, evolutionary approach and satisfaction of the zero commitment condition mentioned above.

Current approaches to symbol grounding in robotics and intelligent systems were well summarized by Coradeshi [264] and the *key challanges* for symbol grounding research area were summarized by Roy [265] and particulary in Cangelosi [266] who reviewed what has been done, what has been negiotated and what they expect that will be done in next 2, 4, 6, 8,10 and 20 years in developmental robotics with the focus on integration of action and language.

4.4 How children acquire language?

Children have to solve the *primary symbol grounding problem* when learning language although the agent (parent) with the knowledge of the grounded symbols (language) is available. Therefore the question how the word-to-meaning mapping is learned remains open. How are the first words separated from the speech and how is their meaning understand? This is the so called "chicken-egg-problem" because after learning first words children could derive meaning of other words (including verbs) from situations.

Snow in her paper [267] found out that *mothers' speech* to 2-years-olds is much simpler and less redundant than their speech to 10-years-old. The same results were observed for mothers and nonmothers, which indicates that young children have available a sample of speech, which is simpler, more redundant, and less confusing than normal adult speech.

There is also evidence that children are sensitive to word meaning even within the first year of life, 6-months olds could pair word "mommy" with the videos of their mother [268] and 9-, 10- and 12-month old infants *accumulate receptive lexical knowledge* [269, 270]. 13- or 14-months old can link a sound to an object when unambiguous pairings are repeated in one session [271]. On the other hand, some studies showed that children even as old as 18 months sometimes do not make the right *inferences of novel words*, e.g. [271, 272, 273]. The Woodward's obeservation that 20-months did not link the novel sound to the object can reflect some awarness of how sound and name differ. One of the explanations provided by Woodward is that 20-months old have a strong *expectation that spoken words serve as names*, and thus resist learning about signals, which do not have this form. This idea is supported by Namy and Waxman's [274] study with similar conclusions. The 26- and 27- month-olds in their studies resisted learning gestural labels even given pragmatic and syntactic cues that the gestures were being used as labels while 18-month-olds were able to learn gesture as a label to a novel object. The second factor

discussed by Woodward [271] can be that sounds in the study were not treated as a grammatical unit and occured outside the utterance boundaries. Overall, these findings raise the possibility that there might be special way how infants learn word-referent mapping compared to the simple associationist theory [275].

Blooms' [273] arguments against the simple associationist theory are that the language is not only used when referent is present, ostensive labelling does not occur at all in some cultures. The first words learnt by children often denote more abstract concepts than the simple associationists would suppose – e.g. "kiss", "brother" etc.

The empirical test (refutation) of simple *associationism* was provided by *Baldwin* et al. [276, 277, 278] in his experiments where a novel object in combination with novel word was presented to children. In the first experiment [276, 277], there were two buckets both containing novel objects and exprimenter said *"It's a modi!"*. Child did not associated novel word with the object it was focused on but with the object the experimenter was looking at. In the second experiment [278] the experimenter said *"It's a modi"* when the child was interacting with the novel object. The children was able to learn the novel word only when there was a direct interaction with the experiementer and not in the situation when the experimenter was outside the room and only the voice was present. Similar experiments were summarized by Tomasello in his article [279, 280].

One possibility how could be word-to-meaning mappings learned by children is *cross-situational learning* [281, 282, 283, 284] which is based on the idea that a learner can determine the meaning of a word by finding something in common across all observed uses of that word. [282] (see Fig. 4.4).

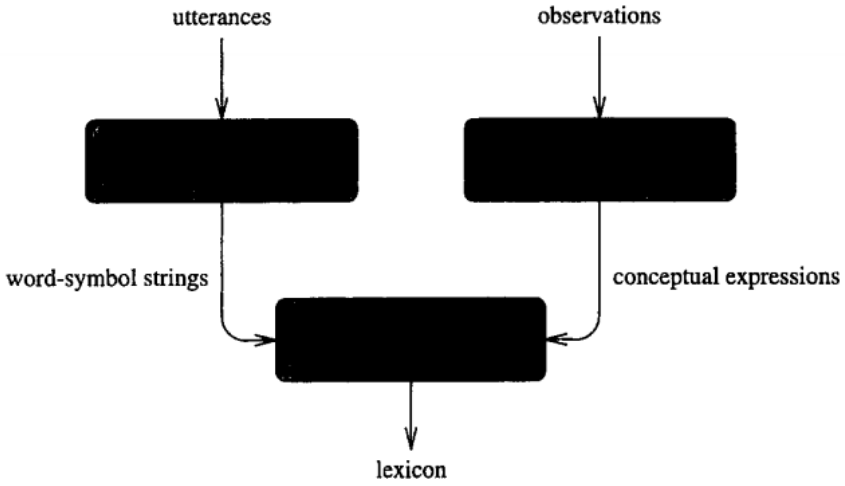


Figure 4.4: The lexical acquisition task and its interaction with other cognitive faculties [282].

Another researcher who has attempted to answer the question how children acquire language is Deb Roy with his ambicious *project Speechome* [285]. He has recorded 230,000 hours of audio-video recordings spanning the first three years of one child's life (his son) at home. The project addresses questions such as where and when different words were learned (the word birth is defined as "the moment of the first reliably transcribed utterance of a new word type by the child") [286] and how the pronunciation and utterance has been changing over time. The curve of a number of word births per month showed

an interesting peak at 20 months. One explanation for these results is that words learned later are less likely to show at the production. The another explanation is that when a child finds out the combinatiorical power of a vocabulary it starts to combine already learned words into sequences to produce new meanings and words.

4.5 Learning language through visual grounding

Ability to learn the language through the perception and especially through *visual grounding* is not only important for understanding human cognition. It is also applicable in many areas such as automatic sports commentators [12], situated speech understanding in computer games [13], car navigation systems (based on map routes), for visually impaired, automated generation of weather forecasts [14], large-scale image database retrieval by natural language query, verbal control of interactive robots [11], in search engines where language and visual information can be combined while finding the best matches for the language query etc.

Computational cognitive models of grounding language are primarily based on the *psychological experiments* which have studied relation between perception and language [287, 2] and language and action [3]. The *computational models* have been developed by Deb Roy [288, 289, 290, 6, 291], Angelo Cangelosi [292, 293], Nicholaos Mavridis [294, 11], team of Yannis Aloimonos [295, 296], Michal Vavrečka [5] and others. While some researches have focused on modeling language grounding by *neural networks* [297, 4, 298, 5], others have developed *probabilistic models* [299, 48, 300, 6]. Researchers who have grounded language in *interactive robots* see symbol as a structural coupling between an agent's sensorimotor activations and its environment [301, 256, 7, 302] and as well reasearchers who deal with the *evolution of the language* [303] are also particularly interested in the language grounding problem.

Roy [291] have highlight issues to take into account when grounding language in perception. These are for example *context dependency* (difference in colour we imagine under "red wine" vs. "red hair"), *functional dependency* (difference between "clean behind the couch" vs. "hide behind the couch") or the fact that larger models should cover not only words but also *phrases and sentences*.

Conversational robotics

One of the main long-term objective of many teams worldwide are the attempts to build the *conversational robots*, which will be able to participate in *cooperative tasks* mediated by a *natural language*. The corresponding cognitive architecture must be developed. This architecture should process synchronously and online visual and language input and solve to some extend the symbol grounding problem. Recently, several European projects with these aims were sponsored such as iTalk [304] or Poeticon [305].

As summarized in [292], *adaptive agent models* and *evolutionary and epigenetic robotics* focus on four main issues: understanding *interdependence among language and perceptual, motor and cognitive capabilities*; understanding *psychological and cognitive bases of language and its grounding*; development of *autonomous interactive systems*; and simulation of *evolutionary emergence of language*. There have been a shift from the fully or partly supervised models [4, 294] to the models which are fully based on the

unsupervised learning [5, 306, 293]. The adaptivity of models is also emphasized. The disadvantage of these models is a very *limited vocabulary* and a mainly *fixed grammar*. Moreover, the number of objects to appear on the scene is limited.

In [292], Cangelosi has presented their research on *language emergence and grounding in sensorimotor agents and robots*. The sensory system of the robot is composed of a contact sensor on the body, which detects when body collides with another one and proprioceptive sensors which give an information about the current position of each joint of the arm. The output layer of their model controls actuators and could be compared to the functionality of motor neurons. Evolution of the agents' behavior is modeled by a genetic algorithm, which is based on the assumptions that emergence of signaling brings direct benefits to the agents and the populations (increased behavioral skills and comprehension ability); that there is a benefit in direct communication between parents and children (while parents produce more stable and reliable input signals); and that the preevolution of good sensorimotor and cognitive abilities permits the establishment of a link between production and comprehension abilities, especially in the early generations when signaling is introduced. They have also studied *imitation and language in epigenetic robotics*. Two robots (12 degrees of freedom) were placed into the virtual world, the physics of the environment was controlled by open dynamic engine (ODE) and the online mimicking algorithm was applied. Agents, in absence of a linguistic input, performed a different default action for every object, which caused the object recognition area to have a double function: categorizing the objects and bootstrapping a default action in absence of linguistic input (see Fig. 4.5).

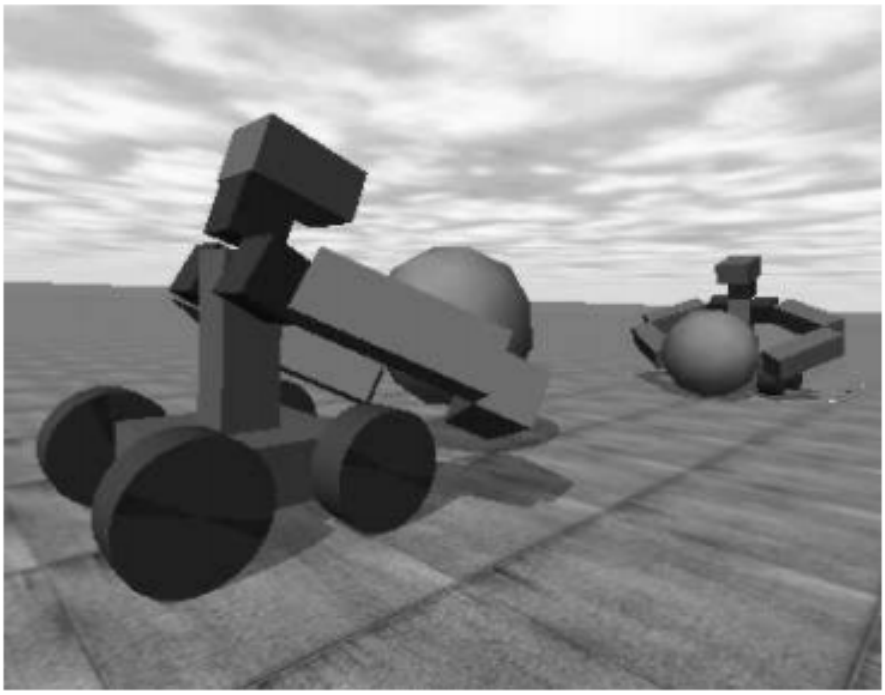


Figure 4.5: Simulation setup for the model of imitation and communication in epigenetic robots (see [292] for more details).

This model was further extended by Tikhanoff [8], who did an *iCub simulation experiments* and focused on *integration of speech and action*. The humanoid robot was able to learn to handle and manipulate objects autonomously, to understand basic instructions, and to adapt its abilities to changes in internal and environmental conditions. Artifical neural networks were used as a feed-forward controller for solving the task of reaching for an object and another control system consisting of a neural controler (Jordan neural network) was used to actually grasp the object. In the experiment described in [306], mixture of multivariate Gaussians (Neural modeling field theory [33, 52]) was applied to the data on the classification of the posture of robots, as in an imitation task.

Crangle and Suppes in [307] proposed a *natural-model semantics* which they applied to the interpretation of robot commands. Two experimental projects are described, which provide natural-language interfaces to robotic aids for the physically disabled. They also examine the use of explicit verbal instruction to teach the robot new procedures or the interpretation of spatial prepositions. This model lacks a perceptual system and there is also manually inserted model of background into the knowledge based. McGuire et al., in [308], built a *hybrid architecture* that combines statistical methods, neural networks, and finite state machines into an integrated system for instructing *grasping tasks* by man-machine interaction. This system incorporates gestures interpretation as well as visual attention or language interpretation. Sofge et al., in [309], utilized an *agent-based architecture* to achieve a multimodal human-centric interface for *controlling a dynamically autonomous mobile robot*. An occupancy map was created using various sensorical data, which enabled the robot to separate the individual objects and communicate about their spatial relationships which was further exploited in interpretation of separate robotic actions.

Another researcher, Cynthia Breazeal has focused on *sociable robots*. The first of them was a *Kismet* [310, 311], which was developed to explore social and emotional aspects of human-robot interaction and is able to speak protolanguage and express emotions. Robot *Leonardo* [311] is a successor of Kismet, which has an implemented cognitive architecture created on the bases of its own database. Language, visual and motoric data are interconnected using central model, so called "Belief system". Sensorical data are captured in separate moments, classified using the hierarchical structures and subsequently sent into the central system which decides whether new beliefs about an object will be created or not. The model also incorporates human beliefs about objects and attentional mechanisms.

Mavridis developed the multimodal architecture and implemented it into the robotic arm called *Ripley* [294, 11], which enables the robot to transfer spoken commands into situated action and vice versa. This architecture is partially similar to the cognitive architecture implemented in Leonardo robot but extends it by providing an ability to quantify and express owns beliefs and confidence. The core of the architecture is dynamic mental model, which updates itself using haptic, visual and language input. This model is able to ground verbs, adjectives and nouns referring to physical referents using an *unified representational framework*. Eventhough many parts of the architecture are adaptive – combination of neural networks and probabilistic models is used, some parts are supervised or fixed – for example the fixed grammar and only predefined number of words and categories.

While the models above have focused mainly on acquisition of nouns and adjectives

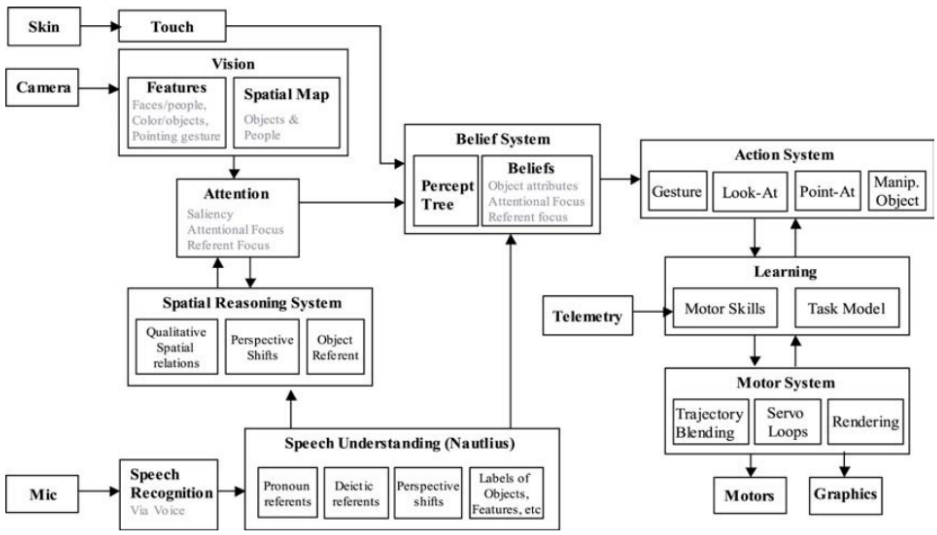


Figure 4.6: Leonardo's cognitive architecture for learning and performing cognitive tasks and motor skills. It is built on top of the c5 codebase (see [310] for more details).

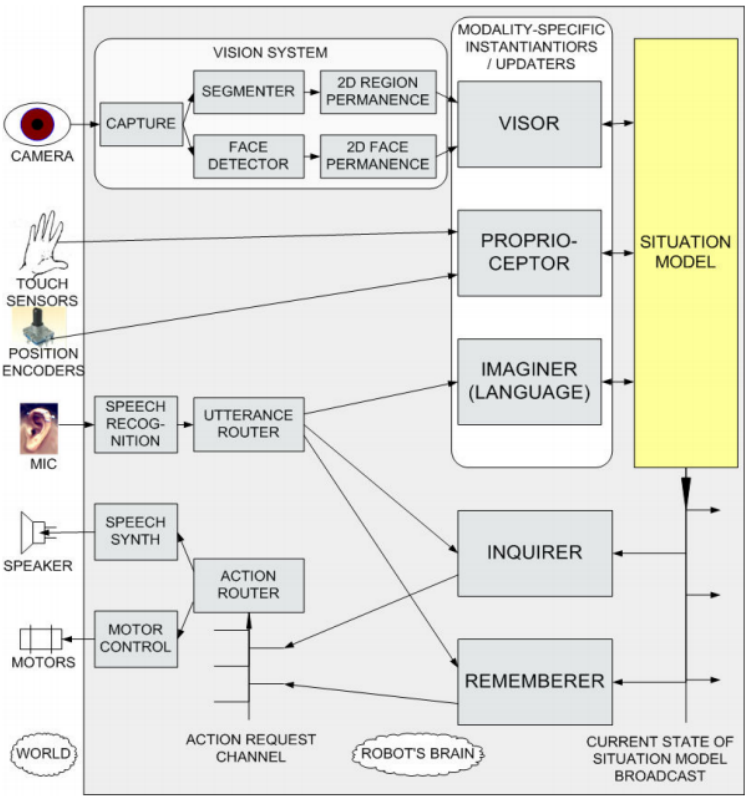


Figure 4.7: Modular software implementation of grounded situation model (GSM) for a Ripley robot (see [294] for more details).

referring to physical objects and their properties, I would like to summarize briefly main research directions in the area of *verbs acquistion*. Bailey [312] have proposed the concept of *x-schemas* which are control sequences of movements of simulated manipulator arm. Action primitives can be organized using these schemas into networks. Each verb is defined by its associated x-schema and control parameters (e.g. each verb "pick up" and "push down" has its own x-schema while "push" and "shove" differs only in control parameters of identical x-schemas – forces and velocity).

Yiannis Aloimonos and his team have focused on *manipulation actions* for robotics. In [295] they propose that *tree banks* are an effective and practical way to organize semantic structures of manipulation actions for robotics and they have introduced the manipulation action context-free grammar which was used to parse semantic tree structures. In another research [296], they have studied occlusions and defined 6 action consequences: assemble, divide, consume, create, transfer, deform. *X-bar schema* was used to describe robotic action. The colaboration of agents was necessary to solve some presented problems. In [313], human actions extracted from video sequences were represented as short sequences of atomic body poses. *Probabilistic context-free grammar* (PCFG) was constructed based on these sequences, which enables the model to recognize new actions and changes from a new single viewpoint video.

Another approach was used by Siskind [314] who analysed video sequences of *human hands manipulating coloured blocks*. Subsequently, he has extracted visually derived features that express contact, support and attachement relationship between hands, blocks and tabletops. Temporal relations between *force dynamic features* were described using *Allen relations* which are 13 possible logical relations between time intervals.

Recent research indicates that sensory and motor cortical areas could play a significant role in the *neural representation of concepts*. In a recent fMRI study [315] 900 words with five sensory-motor attributes (colour, shape, motion, sound and manipulation) were presented and associated activation was examined. The results indicate involvement of multimodal and higher-level unimodal areas.

Roy in review [291] proposes to combine these two approaches to model verbs acquisition – Siskind perceptually grounded verb learning and Baileys x-schemas. In review he also summarizes questions for future research to create a framework for grounding words in terms of structured networks of motor and sensor primitives. Verbs acquisition has been recently also incorparated into the developed conversational robots (mainly through experiments with iCub humanoid robot).

Special part of the grounded models of language are *infants models*, which deal with sensor-grounded language. Roy have developed the *CELL model* [316], where spoken words of shape and colour are segmented and associated. In the model, only one object was presented in corresponding video sequence. Yi and Ballard [317] have combined spoken input with the visual input composed of *multiple objects* and they also have taken advantage of eye tracker when studying eye movements.

Probabilistic models

Finaly, I will focus on probabilistic models of grounding language in perception.

The team of Josh Tenenbaum has been dealing with probabilistic models of cognition and developed also probabilistic model of language grounding. In the article [299, 48] they have proposed the *Bayesian framework for word learning* and then tested the predictions

in three experiments, with both adult and child learners. The model is formulated within the Bayesian framework for concept learning and generalization introduced by Tenenbaum and his colleagues [299, 300, 318] (see Fig. 4.8). Bayesian approach naturally explains the spectrum of generalization behavior observed given one or a few positive examples. Also hypotheses about word meanings are evaluated by Bayesian probability theory which means that the hypotheses are scored according to their probability of being correct.

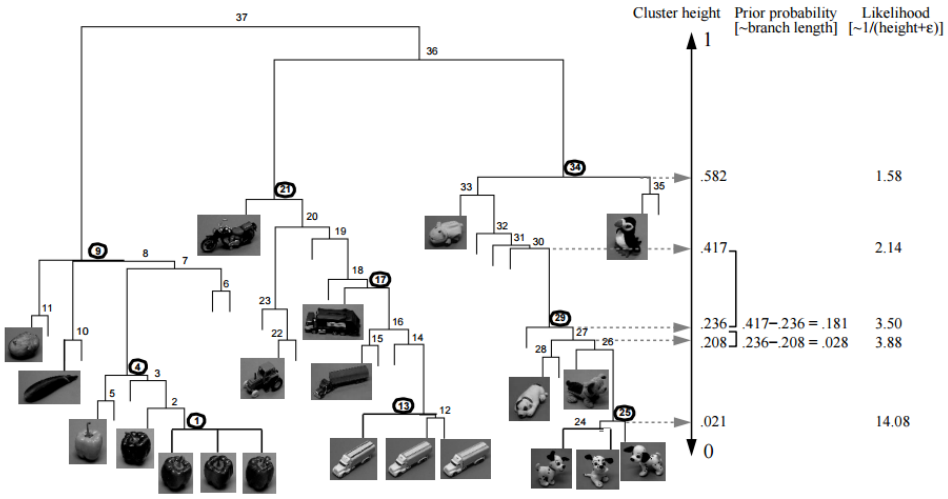


Figure 4.8: Hierarchical clustering of similarity judgments yields a taxonomic hypothesis space for word learning (see [299] for more details).

In [319], they presented a Bayesian model of *cross-situational word-learning* and an extension of this model that also learns which *social cues* are relevant to determining reference. This model should answer the question which word has been most probably used in the presence of specific social cues. They also examined several important phenomena in word learning: *mutual exclusivity* (the tendency to assign novel words to novel referents), *fast-mapping* (the ability to assign a novel word in a linguistic context to a novel referent after only a single use), and *social generalization* (the ability to use social context to learn the referent of a novel word). The proposed model was tested on a corpus of mother-child interactions where each utterance was annotated with the set of objects visible to the infant and with a social coding scheme (infants eyes, infants hands, infants mouth, infant touching, mothers hands, mothers eyes, mother touching). The model has outperformed association models as well as translation models learning from noisy corpus data.

Fontanari et al. [284] used a *mixture of Gaussians* which learn by an EM algorithm, specificaly *Neural modeling fields* categorization mechanism [33], for *cross-situational learning of object-word mapping*. The work aimed to show that a general purpose categorization algorithm can be used as a mechanism to acquire a lexicon in an unsupervised learning scenario. The language is viewed as a mapping between sounds (words) and objects. After learning was the model sensitive to the frequency of co-occurrence of objects and words. They have also showed that the inclusion of a *clutter detection* module

can identify and automatically discard the inputs representing wrong object–word associations. So the model created distinct categories for all correct object–word associations and dumped all wrong associations in a single category.

Another researchers have focused on learning probabilistic language models from *natural language input* [154, 320], some of them also include a visual component [321]. However, these approaches ground the language into predefined language formalisms, rather than extending the model to account for entirely novel input.

Cynthia Matuszek et al. presented in [154] an approach for joint learning of language and perception models for *grounded attribute induction*. The approach builds on existing work on visual attribute classification [322] and probabilistic categorial grammar induction for semantic parsing [323, 324]. The goal is to map automatically a natural language sentence x and a set of scene objects O to the subset $G \subset O$ of objects described by x. The individual objects are extracted from the scene via segmentation and learning is performed via optimizing the data log-likelihood using an online, EM-like training algorithm. This system is able to learn the accurate language and attribute models for the object set selection task, given data containing only language, raw percepts, and the target objects. To bootstrap the learning approach, they first train a limited language and perception system in a fully supervised way (each example additionally contains labeled logical meaning expressions and classifier outputs). The system can be taught to recognize previously unknown object attributes by someone describing objects while pointing out the relevant objects in a set of training scenes. The approach was evaluated on *Amazon Mechanical Turk*.

Tellex [321] introduced a novel model called *Generalized Grounding Graphs* (G3), which is a type of grounding graph – a probabilistic graphical model that is instantiated dynamically according to the compositional and hierarchical structure of a natural language command. Each grounding is taken from a *semantic map of the environment*, which consists of a metric map with the location, shape and name of each object and place, along with a topology that defines the environment connectivity. The model was trained using a corpus of commands collected by crowdsourcing and paired with groundings for each part of the command. This enabled the system to automatically learn meanings for words in the corpus (including complex verbs such as ”put” and ”take”). The system was evaluated in the specific domain of natural language commands given to a robotic forklift.

There are also several researchers who have used probabilistic models to represent *sentences describing videos, scenes and pictures*. However, these approaches are mainly supervised (e.g. [325, 326]) and do not solve the symbol grounding problem eventhough the meaning of a phrase in a description is implicitly grounded by the relevant content of the image.

Kulkarni et al. presented in [325] the supervised automated system for generating and understanding *simple image descriptions*. A *Conditional random field* (CRF) was used to predict the best labeling for an image where nodes of the CRF correspond to several kinds of image content (objects, attributes which modify the appearance of an object, and prepositions which refer to spatial relationships between pairs of objects). The language model was trained using Wikipedia pages that describe objects the system can recognize, and evaluated by the UIUC PASCAL sentence dataset [327]. Sentences were represented using n-grams and HMM. After learning the system automatically

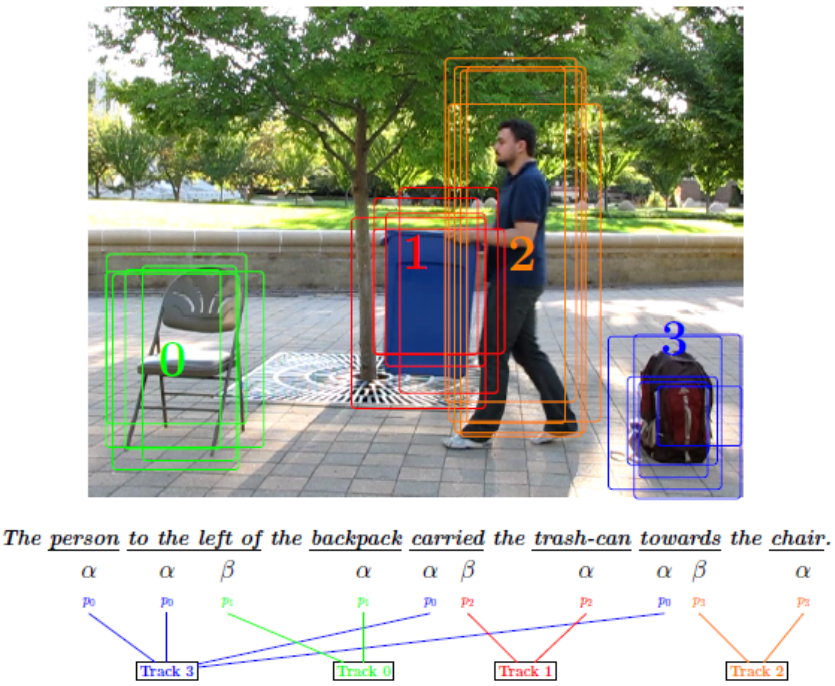


Figure 4.9: Grounded language learning from video sequences. Suppose that each word in the sentence has one or more arguments (α and possibly β), each argument of each word is assigned to a participant $(p_0, ..., p_3)$ in the event described by the sentence, and each participant can be assigned to any object track in the video. Here is shown one of the possible interpretation of the sentence (erroneous). (see [328] for more details).

generates the descriptive text for the presented image (e.g. "*This picture shows one person, one grass, one chair, and one potted plant. The person is near the green grass, and in the chair. The green grass is by the chair, and near the potted plant.*").

The unsupervised learning was used in the model presented by Yu and Siskind [328]. They applied *factorial HMM* to learn representations for word meanings from *short video clips* (people interacting with multiple complex objects in outdoor environments) paired with sentences. The model learns the entire lexicon, including nouns, verbs, prepositions, adjectives, and adverbs, simultaneously from video described with whole sentences. "Compositionality is handled by linking or coindexing the arguments of the conjoined HMMs which were were parametrized with varying arity. Thus a sentence like '*The person to the left of the backpack approached the trashcan*' would be represented as a conjunction of person (p_0), to-the-left-of (p_0, p_1), backback (p_1), approached (p_0, p_2), and trash-can (p_2) over the three participants p_0, p_1, and p_2. This whole sentence is then grounded in a particular video by mapping these participants to particular tracks and instantiating the associated HMMs over those tracks, by computing the feature vectors for each HMM from the tracks chosen to fill its arguments." (see Fig. 4.9)

The most relevant research to my work was done by Deb Roy and his team who have developed a series of techniques for grounding words in visual scenes [6, 294, 329, 330]. In [6], Roy has focused on learning visually grounded language and spatial relationships

between the objects represented by relative spatial clauses. Several visual features were used to represent objects (radius, width, colour, x and y coordinate etc.). Two speakers described the observed objects and their spatial relationships using utterances and clauses with varying difficulty (e.g. "*The red square*", "*The purple rectangle to the left of the pink square*"). Subsequently, these speaker recordings (2x 90 min) were manually transcribed. Language was represented using hidden Markov models. Words and phrases were separated into clusters based on their semantic features associations. Parameters of multivariate Gaussian models associated with each word were learned by EM algorithm and KL distance was used as a similarity measure. Word order constraints were modeled by class-based bigram transition probabilities (see Fig. 4.10). This stochastic network was able to capture spatial relationships between objects. Compared to our model, the class-based bigram is trained on the corpus of sentences.

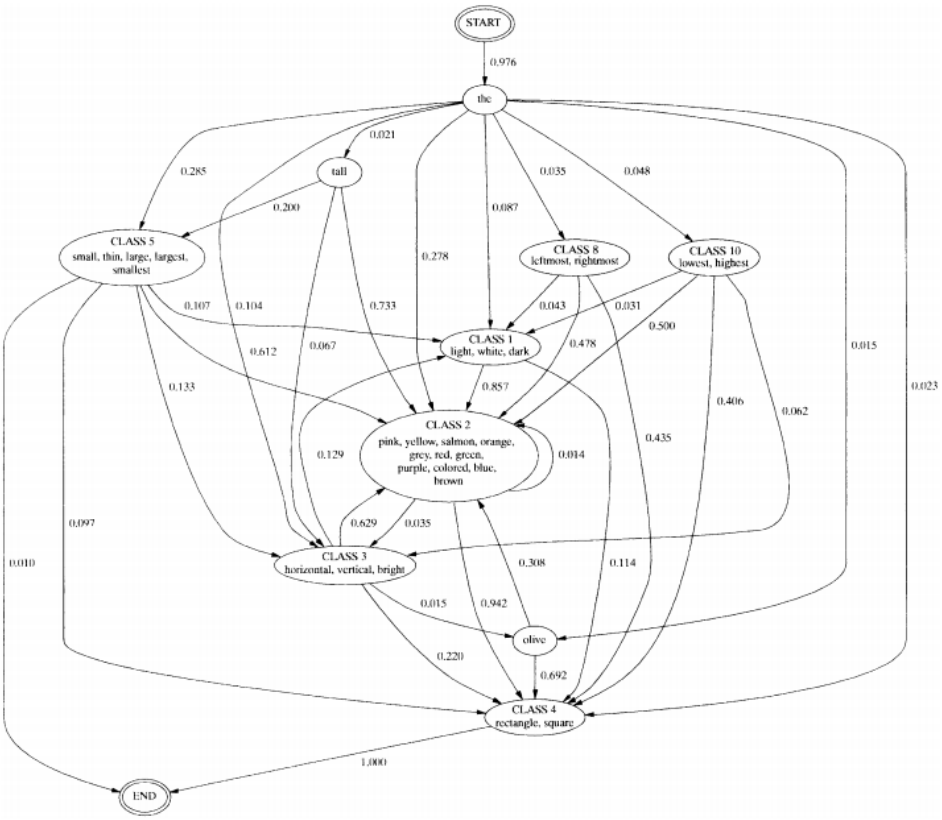


Figure 4.10: Word-class based statistical bigram for simple utterances (see [6] for more details).

Chapter 5

Proposed multimodal cognitive architecture

5.1 General overview

In the thesis, mainly the architecture, which was further used for processing artificial data with five separate visual features (size, colour, texture, orientation and shape), is described. The architecture can be as well adapted for specific cases such as was done for experiments with iCub (see architecture shown in Section 5.5). My own architecture was published in [21] and the video from implementation into humanoid robot can be seen at [22].

5.2 Visual layer

The sensory input is captured by the visual layer which serves as an artificial retina. The visual layer is represented by a set of mixture models, specificaly mixture of Gaussians.

Firstly, the visual input is divided based on the different visual features (colour, orientation, shape, etc.) (see Fig. 5.2) and these visual features are subsequently processed separately. From the neuroanatomic point of view, this corresponds to the processing of the visual input in the separate higher visual centres in the brain, specifically to the independent processing of the information about position and indentification of an object in the ventral ("what") and dorsal ("where"/"action") neural pathways respectively [331, 332]. Individual object properties are identified in the separate visual centres of the occipital lobe.

The input to the first visual layer can be both vector of features extracted from a raw image data or the raw data transformed from the matrix to the vector form (e.g. y_n^{size}, y_n^{colour}, $y_n^{orientation}$, $y_n^{texture}$ and y_n^{shape}) and serves as an input to the first layer. Outputs of all unimodal modules from the first layer are concatenated and serve as an input vector to the second layer where the identification of an object is performed. Object recognition is realized by an unsupervised learning, specifically data are modeled by a mixture of gaussians learned by the EM algorithm. This approach is compared to the k-means algorithm, SOM, GWR and supervised learning of the mixture of Gaussians.

The first layer – processing of individual visual features

Each visual feature (e.g. size, colour, etc.) of input data point is processed separately using the mixture of Gaussians in the first visual layer (see Fig. 5.3). The number of models in the mixture model matches the number of expected classes. In the first experiments, the known number of classes is supposed (e.g. 10 for shapes, 5 textures, 3

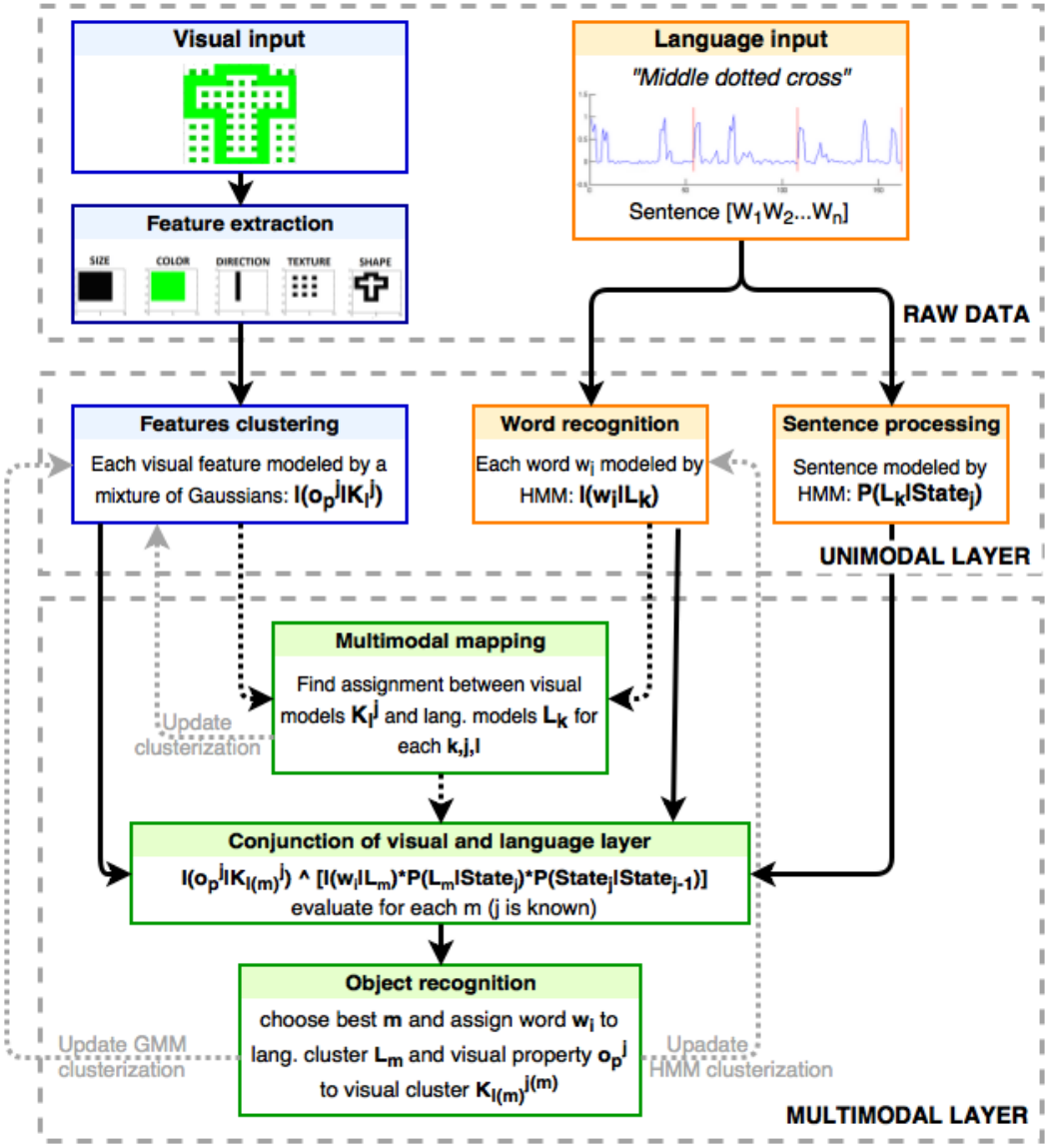


Figure 5.1: General overview of the proposed architecture

sizes, 3 orientations and 9 colours) and simple GMM is used. Subsequently, the number of Gaussians is found automatically (greedy GMM or the newly proposed greedy GMM with merging are used).

The used Gaussian mixture model is a convex mixture of d-dimensional Gaussian densities $l_k(\vec{x_n}|\vec{\theta_k})$:

$$f_k^{feature}(\vec{x_n}) = \sum_{k=1}^{K_{feature}} r_k l_k^{feature}(\vec{x_n}|\vec{\theta_k}), \forall feature, \tag{5.1}$$

(e.g. $feature \in$ {size, colour, orientation, texture, shape}),
where $\vec{x_n}$ is d-dimensional continuous-valued data vector, r_k are the mixture weights,

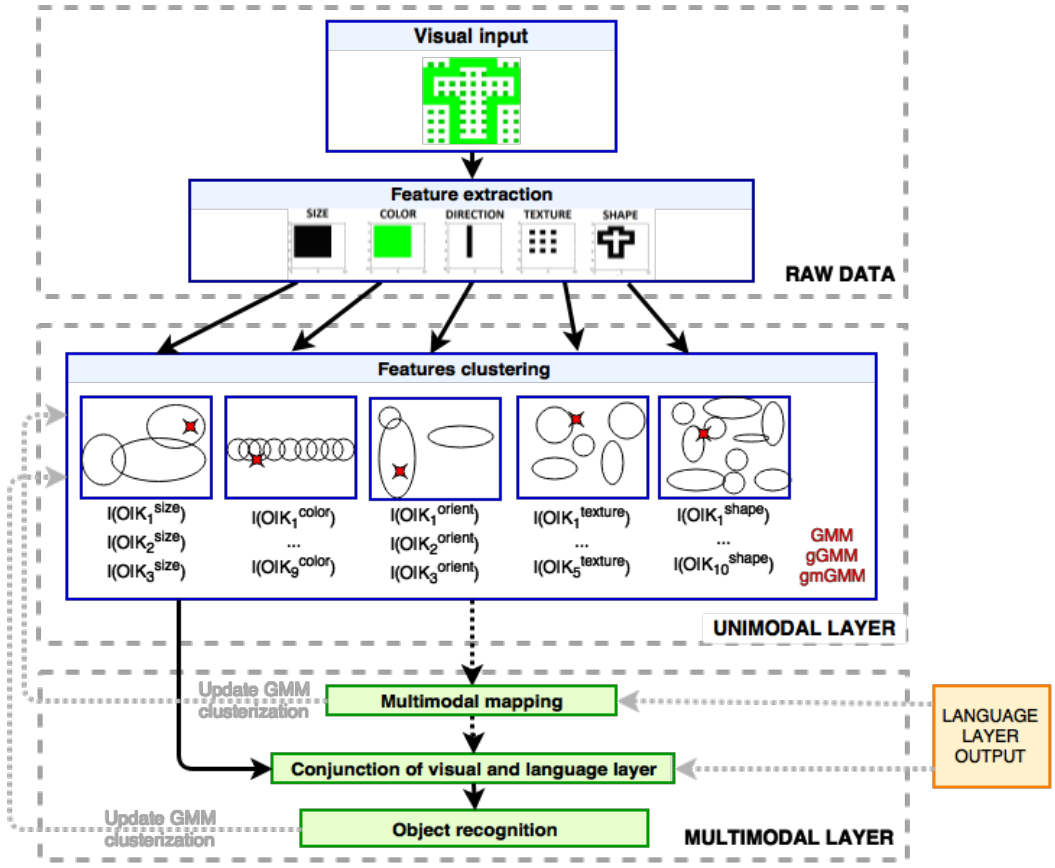


Figure 5.2: Processing of the visual information in the proposed architecture.

parameters $\vec{\Theta_k}$ are cluster centres $\vec{m_k}$ and covariance matrices $\vec{C_k}$,and $K_{feature}$ is number of clusters in data associated with a given visual $feature$.

Mixture of Gaussians is trained by the EM algorithm (see Eq. (2.9–2.12)).

An output of this layer is the vector $\vec{y}$ of $\sum_{feature} K_{feature}$ output parameters describing the data point – the likelihood that the data point belongs to each cluster (we get the information about redness, stripeness and horizontalness of an object or how much is the object star-like).

$$\begin{aligned} y_n = \quad & [l(O|K_1^{size}) \dots l(O|K_3^{size}), l(O|K_1^{colour}) \dots l(O|K_9^{colour}), \\ & l(O|K_1^{orient}) \dots l(O|K_3^{orient}), l(O|K_1^{texture}) \dots l(O|K_5^{texture}), \\ & l(O|K_1^{shape}) \dots l(O|K_{10}^{shape})] \end{aligned} \tag{5.2}$$

An overview of the algorithm used in the first layer is presented in Algorithm 6.

The second layer – Evaluating and integrating visual features of an observed object

Output parameters from the first layer are being integrated in the second layer. The identification of an observed object is based on these concatenated parameters. In this

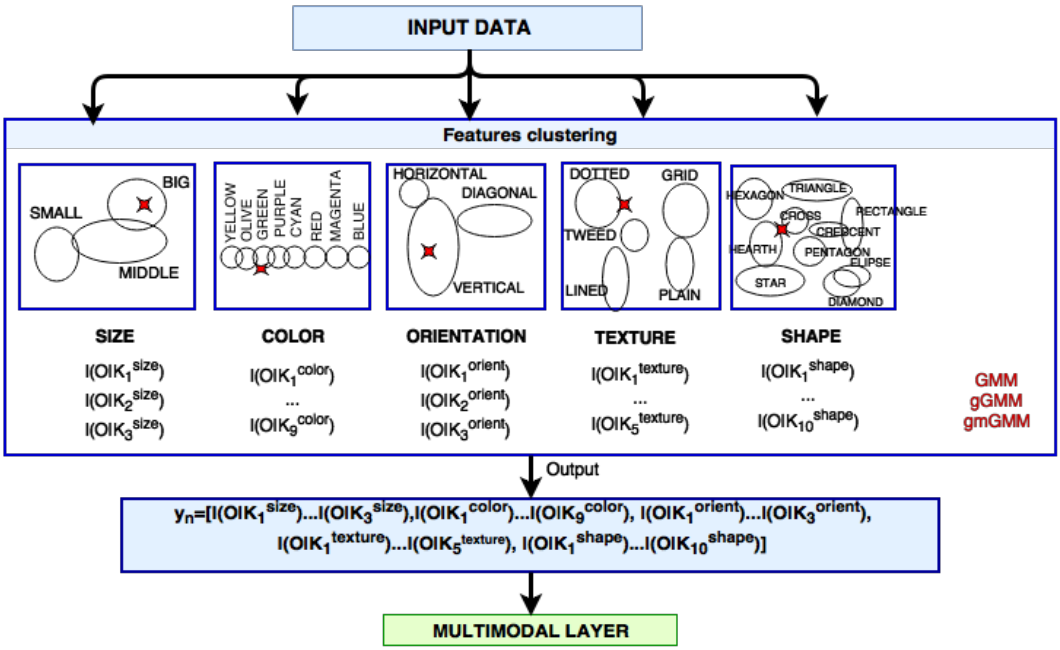


Figure 5.3: The first visual layer of the proposed architecture.

Algorithm 6 Architecture – the first visual layer.

Inputs:
input visual data $\vec{x_j^i}$ for each feature i and each data point j
Input parameters:
number of classes $NmbCl^i$ for each feature i (optional)
Output:
matrix of likelihoods $\vec{y}$ that the given data point j belongs to the given cluster k of the given mixture i

for $i \in$ {size, colour, orientation, texture, shape} **do**
if known number of classes $NmbCl^i$ **then**
$\theta^i = \{\vec{M^i}, \vec{C^i}\}$ initialize parameteres of $NmbCl^i$ components of a mixture
$[LL^i, \vec{F^i}, \vec{l^i}] \leftarrow \text{EMalg}(\vec{x^i}, \vec{\theta^i})$ train a mixture by EM algorithm, LL^i – log-likelihood, $\vec{F^i}$ – memberships of each data point to each cluster in a mixture, $\vec{L^i} \leftarrow l(\vec{x_j^i}|K_n^i)$ – matrix of likelihoods that the given data point $\vec{x_j^i}$ belongs to the given cluster K_n^i in a mixture i
else(unknown number of classes $NmbCl^i$)
$\Theta_{best} \leftarrow \text{gmGMM}(\vec{x^i})$ estimate components' parameters and number of classes by gmGMM algorithm (or another similar algorithm for unknown number of classes in GMM)
$\vec{L^i} \leftarrow l(\vec{x_j^i}|K_n^i)$ compute likelihoods that the given data point $\vec{x_j^i}$ belongs to the given cluster K_n^i in a mixture i
$\vec{F^i} \leftarrow$ compute memeberships of each data point to each cluster in a mixture
end if
$\vec{y_j^i} = [l(\vec{x_j^i}|K_1^i), l(\vec{x_j^i}|K_2^i), \ldots, l(\vec{x_j^i}|K_{NmbCl_i}^i)]$ output of unimodal layer i
end for
$\vec{y} = [\vec{y^{size}}, \vec{y^{colour}}, \vec{y^{orientatiton}}, \vec{y^{texture}}, \vec{y^{shape}}] \leftarrow$ concatenate outputs of unimodal layers

level, the symbol grounding problem must be solved as well as so called binding problem. The binding problem refers to the segregation problem how to retroactively assign decomponed features to the individual objects or how "to bind together all the features of one object and segregate them from features of other objects and the background" [333]. This problem is solved by interconnecting visual and language layer and could be also solved by utilizing the feature integration theory of attention developed in 1980 by Anne Treisman and Garry Gelade [334]. They suggested that during the decomposition of an object to specific features attentional resources are used to bring the various independent feature maps into register with respect to a master map of location. This master map of locations will indicate what combinations of features coexist at each location in the map which will enable subsequent feature integration.

There are two basic options how to transfer the information of data point clusterization from the first to the second layer:

1. **Winner-takes-all** (localist representation) – the strategy where only the cluster with the highest cluster membership probability is considered (the data point is assigned to this cluster):

$$M(O|K_k^{feature}) = \begin{cases} 1 & \text{if } k = \text{argmax}_j\, l(O|K_j^{feature}) \\ 0 & \text{if } k \neq \text{argmax}_j\, l(O|K_j^{feature}) \end{cases} \tag{5.3}$$

   $\forall feature$ (e.g. $feature \in$ { size, colour, orientation, texture, shape}).

   Example: in the case that output likelihoods that data point corresponds to the given texture are: 0.01, 0.3, 0.5, 0.1 and 0.05 for dotted, lined, tweed, grid and plain respectively, we will transform the output to the vector: 0 0 1 0 0 and assign the data point only to the 3rd cluster (tweed).

$y_j =$ [0.1 0.2 0.4 0.8 0.7 0.01 0.04 0.2 0.12 0.001 0.002 0.001 0.2 0.05 0.3 0.01 0.3 0.5 0.1 0.05 0.04 0.2 0.1 0.03 0.5 0.6 0.1 0.002 0.003 0.04]
(Outputs of the 1st layer)

$->$

$z_j =$ [0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0]
(Inputs to the 2nd layer)

2. **Fuzzy memberships** (distributed representation) – the fuzzy membership to different clusters are considered and further combined with the language output and with prior expectations. The prior expectation corresponds to the situation when a red apple is more likely to be observed than a red banana. The likelihoods to different clusters are used directly as an input to the second layer.

Outputs of the 1st layer = Inputs to the 2nd layer

$$y_j = z_j$$

The final object identification can be done directly based on the winner-takes-all outputs from the 1^{st} layer (Eq. (5.3)). These data will tell us an assinment of a data point to individual clsuters. Therefore we can generate the "prototypical object" representing an observed object. This prototypical object would be generated from the mean values of winning clusters. Furthermore, if we have available the true labels, we can label the clusters (based on the most occured label in a cluster) and consequently evaluate the recognition accuracy.

In next chapters there is described how the visual clusters can be labeled in an unsupervised manner by mapping vision to the language data. In this case both winner-takes-all and fuzzy membership can be applied.

We could alternatively use fuzzy outputs from the 1^{st} layer and cluster these data once again in the second layer. The outputs from the first layer (n dimensional vector of output parameters) would serve as an input to the self organizing neural network with the number of neurons equal or bigger than the expected number of classes (for 10 shapes, 5 textures, 3 sizes, 3 orientations and 9 colours this will give total amount of 4050 classes). The clusterization error will be tested in the following manner: after learning the neural network, neurons will be labeled by the class to which they respond the most and then in the testing stage. Alternatively GMM can be used for data clustering. Anyway, it is worth mentioning that this approach is very inefective since we would need enourmous number of training data and it also partly goes against the reason why we did decomposition of an object to separate visual features.

5.3 Language layer

Language (or auditory) inputs (English sentences) in a vector form are processed in the following manner: firstly individual words are processed and subsequently full sentences are processed in the second layer. Hidden Markov models (HMM) are used in both layers of hierarchy. Outputs of both layers are combined into one output, which describes the likelihood that the language input corresponds to the appropriate sequence of words. Input sentences can have fixed or variable length and are encoded as high-dimensional patterns.

Word processing

Words are modeled by m-state Hidden Markov model (HMM) with one output Gaussian distribution for each hidden state (see Chapter 2.2).

In this case, each hidden state can represent either one phoneme or bunch of the phonemes. Depending on it, number of hidden states can either correspond to:

- number of individual phonemes, or
- bunch of phonemes can be represented by one hidden state where emission probabilities of this hidden states identify the used phoneme. In this case, the optimal number of hidden states will be estimated based on the clusterization error.

If the lexicon is given, we can construct separate HMM models for each lexicon word W_i. After fitting all models l_i, we can generate a corresponding log-likelihoods $ll_{ij} = \log l(W_j|L_i), 1 \leq i, j \leq N$ (evaluate the log-likelihood of each of the N sequences

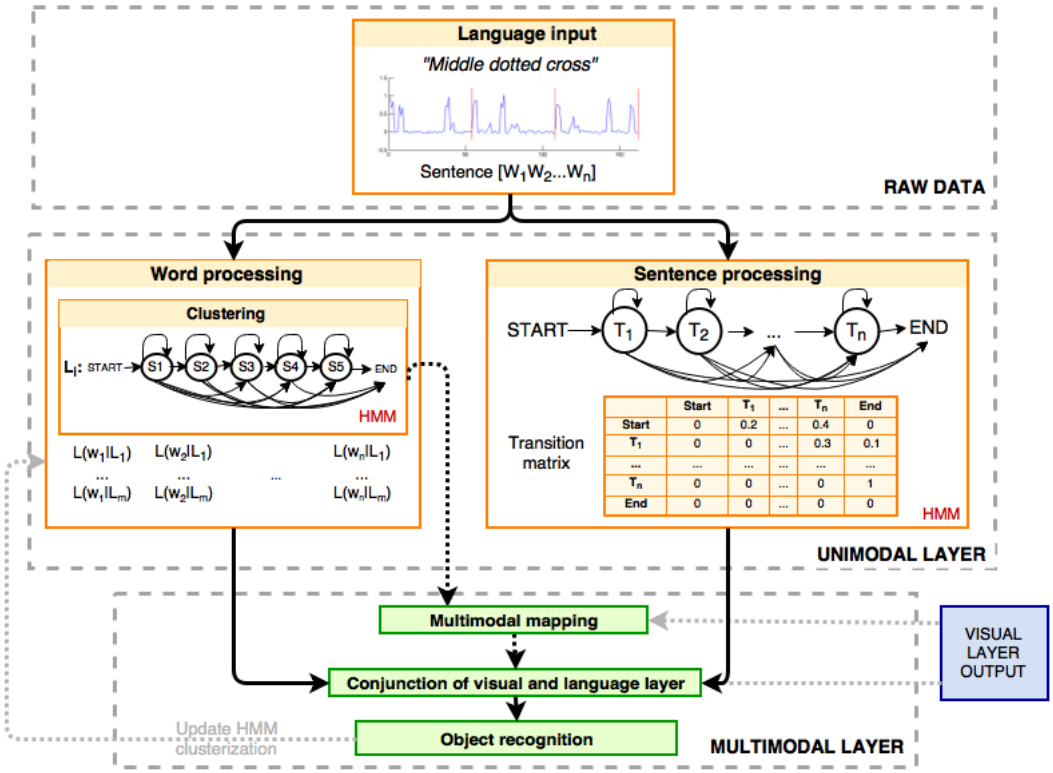


Figure 5.4: Processing of the language data in the proposed architecture.

given model L_i) that could be computed either by Baum-Welch or by Viterbi algorithms (see Fig. 5.5).

A variety of clustering methods can be used to cluster the sequences into K groups using log-likelihood distance matrix. The symmetrized distance:

$$ll_{ij} = \frac{\log(l(S_i|L_j)) + \log(l(S_j|L_i))}{2}, \tag{5.4}$$

can be used as an appropriate measure of dissimilarity between models L_i and L_j [335, 80]. There are also other methods based on different kernel to measure the pairwise similarity between sequences such as Bhattacharyya affinity [78] (for more details see Chapter 2.2).

In this thesis, two clustering techniques are compared:

- k-means clustering (implementation described in [81]) (see Section 2.2, Algorithm 1),
- agglomerative clustering [80] (see Section 2.2, Algorithm 2)

Cluster redistribution is deterministic, assigning each item $\vec{x_n}$ to the cluster c_j that gives it the highest posterior $l(\vec{x_n}|c_j)$.

The first implemented algorithm was clustering of HMM using k-means (see Chapter 2.2, Algorithm 1) which served as a referential method for more sophisticated algorithms.

Two alternative designs of the first layer are compared:

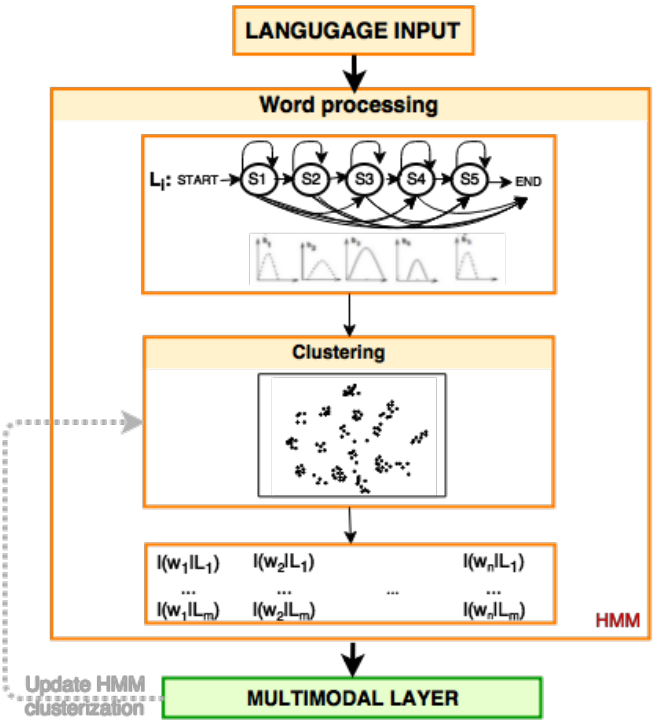


Figure 5.5: The first language layer of the proposed architecture – word processing.

1. All possible words are clustered altogether. This method enables variable length of the sentence as well as variable grammar. On the other hand clustering will achieve worse results in a case of a fixed grammar where the corresponding feature to the given word is known (see Fig. 5.6).

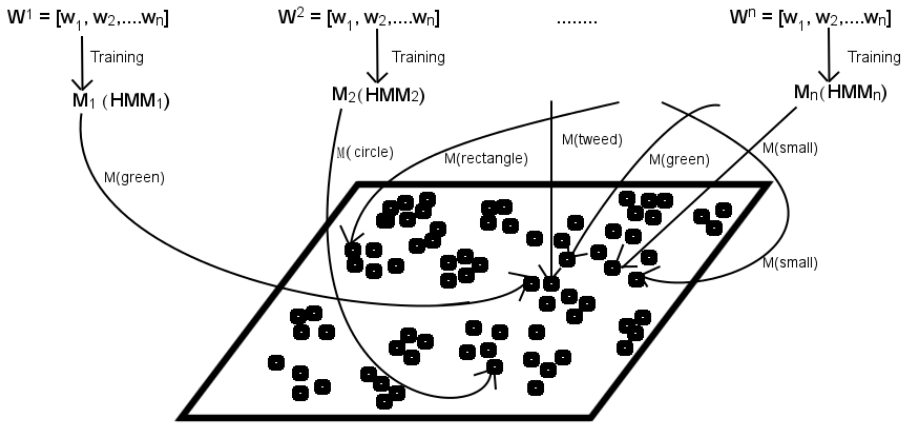


Figure 5.6: Word processing – visualization of distance measure between sequences represented by HMMs (all words clustered altogether).

The process of learning is as following:

- 1. $ll(k,j,i) = l(w_k^j|L_i), \forall_{j,k,i}$
- 2. assign word w_j^k to the language model $L_{im}, \forall_{j,k} : im = \text{argmax}_i\, ll(k,j,i)$
- 3. relearn $L_i, \forall_i$
- 4. repeat from 1.

2. Similarly to the visual layer, words corresponding to each feature (or word type) are clustered separately. This is possible when we use the fixed grammar and length of the sentences or by utilizing the probabilities of transition between different features computed in the second layer (see Fig. 5.7). It has to be mention, that in this case the number of language clusters will be different generally from the number of visual ones because we cannot restrict the recognized words only to words used for the given feature (e.g. even though we say "*Blue*" and in front of us is a cube, we can hear "*Cube*" so the word "cube" will create a new cluster in language).

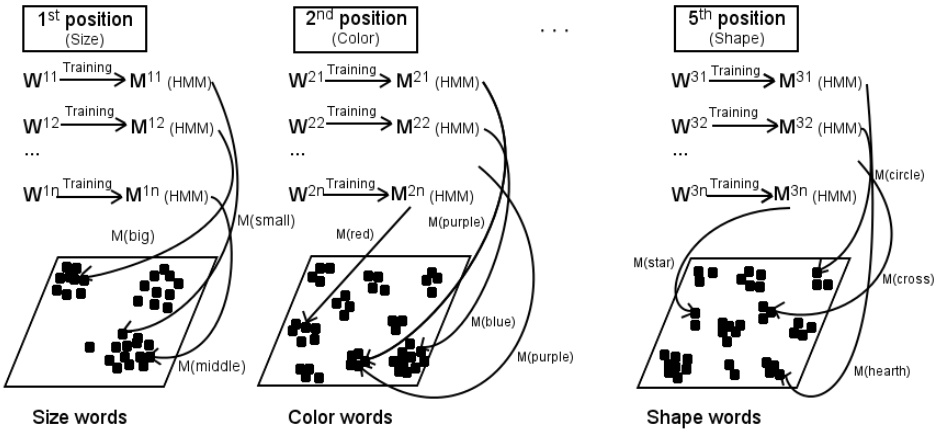


Figure 5.7: Word processing – visualization of distance measure between sequences represented by HMMs (words corresponding to each feature are clustered separately).

Notes:

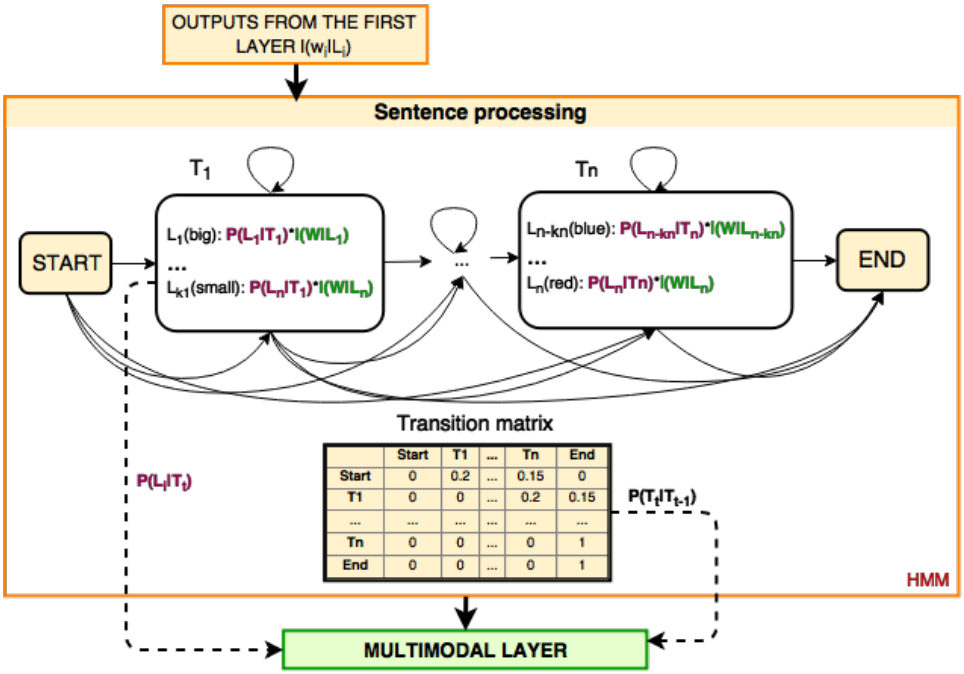
- What would happen when a two-state HMM with discrete or continuous output probabilities would be used? This model would be able to only differentiate between a small number of words. The discrete model would not be able to differentiate between words, which have different level of energy in a given hidden state.
- What if we considered words as static n-D patterns. Even though this model would be easier to implement and would achieve higher clusterization accuracy, we would not be able to use this model for speakers varying in prosody or phrasing of the words as well as it would not be able to differentiate among words of different length.
- In the algorithms used in this thesis, every item is assigned unambiguously to a single cluster (represented by a single HMM) on each iteration, and the HMM

parameter updates are influenced only by those items currently in the associated cluster. This leads to much quicker convergence of the algorithm [81]

- Cluster priors which could reflect the cluster size are ignored in these implementations
- There are other clustering algorithms which can be used such as divisive clustering or model-based clustering

Sentence processing

In the second layer, sentence processing is modeled by a probabilistic model, specificaly HMM. The probabilistic model of sentence processing will predict the following words on the basis of previous experiments. While in word processing we solved the "Evaluation problem": given a HMM model L_i and sequence of observations $O_1, O_2, \ldots, O_n$ what is the probability that the observations are generated by the model: $p(\vec{O}|L_i)$, in sentence processing we are mostly interested in a "decoding problem": Given a model L_i and sequence of observations $O_1, O_2, \ldots, O_n$ what is the most likely state sequence in the model that produced the observation? In both cases (word and sentence processing), learning problem must be solved because HMMs should be trained online while newly data are continuously added to the training set.



	Start	T1	...	Tn	End
Start	0	0.2	...	0.15	0
T1	0	0	...	0.2	0.15
...	...	...	...	...	...
Tn	0	0	...	0	1
End	0	0	...	0	1

Figure 5.8: The second language layer of the proposed architecture – sentence processing.

The simplification that the probability of an observation at time n only depends on the observation at time $n-1$ is called the first-order Markov assumption. Eventhough the proposed model could be used to the sentences of an arbitrary length and grammar

(see Fig. 5.11), in the thesis I will focus only on very simple sentences of variable or fixed length describing one observed object to be able to .

Each hidden state corresponds to one feature (the 5-states HMM is used) and the transition between hidden states is described by the transition matrix. Each hidden state can produce a bunch of observations which output probabilities are described by an emission matrix. (e.g. hidden state "size" will produce output observation "big", "small" or "medium"). The output probabilities defined by transition and emission matrices can evolve during the learning (e.g. more one-word sentences are produced at the start and longer sentences can be added as the learning progresses) (see Fig. 5.9).

I compare two alternative designs of the sentences:

1. The full-length sentence having a fixed grammar and fixed length which has a following structure:

   <Size> <Colour> <Orientation> <Texture> <Shape>
   (e.g. "*Big red horizontal dotted cross*")

   Example of transition matrix for sentences with fixed length and grammar is shown in Table 5.10a.

   In this case, the model and its tranisition matrix is given and only emission matrix can be relearned after each observation.

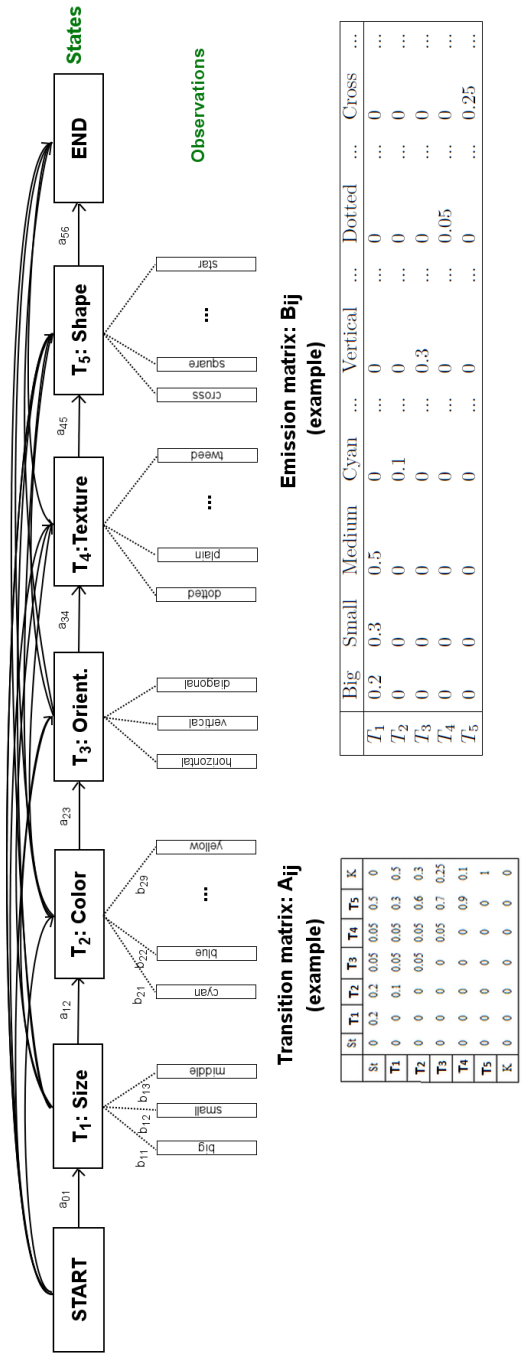
2. The sentences with a variable length. The examples of these sentences are following:

   "*Rectangle*"
   "*Red triangle*"
   "*Big striped circle*"
   etc.

   In this case, parameters of transition matrix must be derived from the input data. The structure of transition matrix can evolve through learning (e.g. at the start of learning there are only short one-word sentences describing only shape or colour of an object, in the next stage, sentences describing more features of an object can arrive). Specific observations of the specific state are described by an emission matrix.

   Example of transition matrix for sentences with a variable length is in Table 5.10b.

   When we would like to consider also non-fixed order in the sentence corresponding to a variable grammar, transition matrix would change so that there are non-zero elements below the diagonal.



	Big	Small	Medium	Cyan	...	Vertical	...	Dotted	...	Cross	...
T_1	0.2	0.3	0.5	0	...	0	...	0	...	0	...
T_2	0	0	0	0.1	...	0	...	0	...	0	...
T_3	0	0	0	0	...	0.3	...	0	...	0	...
T_4	0	0	0	0	...	0	...	0.05	...	0	...
T_5	0	0	0	0	...	0	...	0	...	0.25	...

	St	T1	T2	T3	T4	T5	K
St	0	0.2	0.2	0.05	0.05	0.5	0
T1	0	0	0.1	0.05	0.05	0.3	0.5
T2	0	0	0	0.05	0.05	0.6	0.3
T3	0	0	0	0	0.05	0.7	0.25
T4	0	0	0	0	0	0.9	0.1
T5	0	0	0	0	0	0	1
K	0	0	0	0	0	0	0

Figure 5.9: The second language layer of the proposed architecture – sentence processing, illustrative Transition and Emission matrix.

	T_1	T_2	T_3	T_4	T_5	F
St	1	0	0	0	0	0
T_1	0	1	0	0	0	0
T_2	0	0	1	0	0	0
T_3	0	0	0	1	0	0
T_4	0	0	0	0	1	0
T_5	0	0	0	0	0	1

(a) : Fixed length

	T_1	T_2	T_3	T_4	T_5	F
St	0.2	0.2	0.1	0.1	0.4	0
T_1	0	0.1	0.1	0.1	0.2	0.5
T_2	0	0	0.1	0.1	0.5	0.3
T_3	0	0	0	0.1	0.7	0.3
T_4	0	0	0	0	0.9	0.1
T_5	0	0	0	0	0	1

(b) : Variable length

Figure 5.10: Transition matrices for fixed (a) and variable length (b) (St – Initial state, T_1 – Size, T_2 – Colour, T_3 –Orientation, T_4 – Texture, T_5 – Shape, K – Final state).

In Fig. 5.11, I propose an extension of the sentence processing model for more complex sentences with variable grammar and length. This model would capture more complex sentences, which describe multiple objects and their spatial relation including action verbs (e.g. "*Big red cross on the left is moving quickly towards the dotted triangle in the front.*")

5.4 Multimodal Cooperation Between Visual and Auditory Layers

After individual modalities are procesed separatelly, they are interconnected in a top most multimodal layer of the architecture. To be able to correctly join the information from the separate unimodal areas, corresponding clusters in both modalities have to be mapped to each other. From a neuroscientific point of view, this can be viewed as a process of finding mapping between primary unimodal visual and language brain areas. Where the integration is performed is still the subject to research and existing literature provides conflicting accounts of the cortical location of this convergence. For example, the study of [336] provides evidence for the involvement of the left basal posterior temporal lobe (BA37) in the integration of language and visual information. On the contrary, e.g. Spitsyna et al. [337] propose that access to the verbal meaning depends on both anterior and posterior heteromodal cortical systems within the temporal lobe.

The mapping solves the problem how to correctly assign words to individual object properties (e.g. that word "*triangle*" describes the shape of an object and not its colour). The mapping task is non-trivial even for the fixed length grammar sentences. In a case of variable length/grammar sentences, we have to deal with even more incomplete information. Good sentence model (see Section 5.3) including the previously learned transition matrix (see Table 5.10) describing the sentence can help. Nevertheless, the

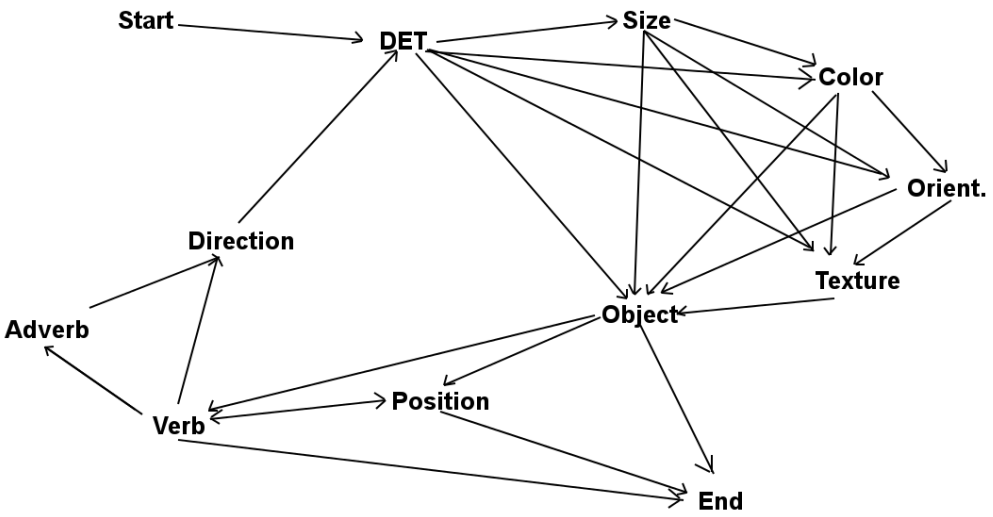


Figure 5.11: Sentence processing – scheme of HMM used for processing sentences with a variable length (DET – the, a, an,...; Adverb – slowly, quickly,...; Position – on the left, at the bottom,...; Direction – to, from,...; Verb – move, bounce, jump,...; Object – cross, chair, wall, triangle,...).

amount of data necessary for finding the best mapping is in this case obviously increased compared to the case of a fixed grammar.

Finding the best mapping between visual and language layer – fixed lenght and grammar sentence

After both visual and language data are clustered, the mapping between the two layers must be found. For each cluster L_i in language layer corresponding cluster $K_j^{feature}$ in visual layer for each feature *feature* (e.g. *feature* $\in$ {size, colour, shape}) is found or vice versa. In a case of fixed length and grammar sentence the situation is slightly simpler since we know to which visual feature the heard word corresponds to. Therefore we can find the number of coocurence of the visual feature with the given word without any need for a sentence model (or the sentence model is described by a diagonal transition matrix).

The basic model for a fixed grammar and length sentence is visualized in Figure 5.12. You can see that in the scheme there is also proposed that the clustering of $K_j^{feature}$ and L_i and their mapping could be updated with the newly added datapoints and based on the result of their clusterization.

The mapping is found as follows: for each j and $feature$ we find cluster $L_{kmax_{jf}}$ from language layer which will be assigned to the cluster $K_j^{feature}$ from the visual layer. Two different models how to find indices $kmax_{jf}$ are compared.

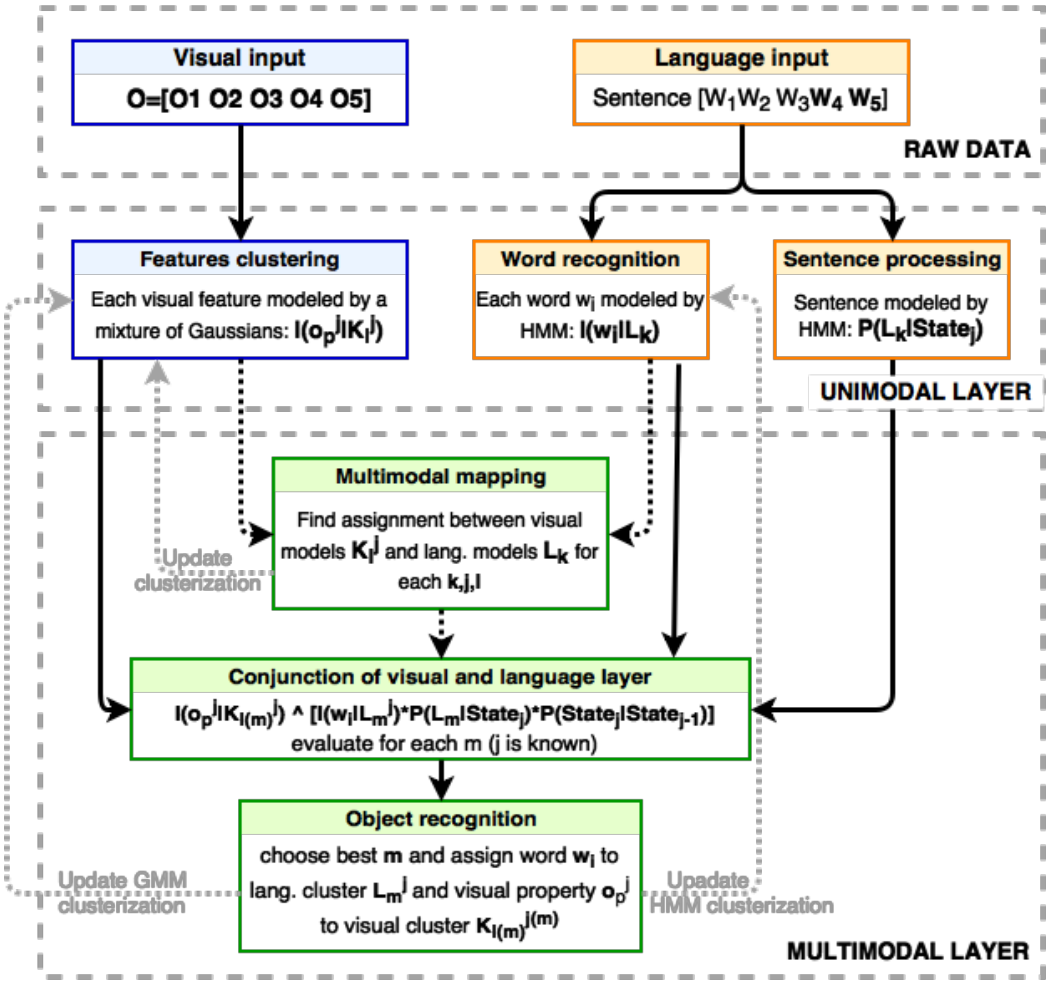


Figure 5.12: Scheme of the architecture for the fixed length sentences. Communication between visual and language layer is visualised by dashed lines. In the case of the fixed length sentences, type of the feature (j) which is described by the given word is known from the position of the word in the sentence.

Model 1 (one-step mapping): Indices $kmax_{in}$ are found as following:

$$\forall_{j,feature} kmax_{jf} = \max_{k} \sum_{i \in K_j^{feature}} l(L_k, x_i^{feature}), \tag{5.5}$$

where $x_i^{feature}$ are selected data from dataset $x^{\vec{feature}}$ which based on the vision should be assigned to the cluster $K_j^{feature}$ and $l(L_k, x_i^{feature})$ is likelihood that data point $x_i^{feature}$ will be assigned to the cluster L_k.

Model 2 (sequential mapping): Indices $kmax$ are found sequentially so that always the best mapped data are excluded and the rest is reclustered using GMM. Afterwards one-step mapping is performed (see Alg. 7).

In the ideal case, the unambiguous mapping between the two clusterizations will be found. In the real case (when clusterizations in visual and language layer are not

Algorithm 7 Sequential mapping

Inputs:
language clusters L_i ($i \in 1:M$), visual clusters $K_j^{feature} \sim N(\vec{m_k}, \vec{S_k})$, $j \in 1:N^{feature}$, input data $x^{\vec{feature}}$ for each feature $feature \in$ {size, colour, orientation, texture, shape}, number of clusters $NmbCl^{feature}$ for each feature $feature$

Output:
mapping between all visual classes $K_j^{feature}$ and language classes L_i

for $feature \in$ {size, colour, orientation, texture, shape} **do**
  $NCl \leftarrow NmbCl^{feature}$
  while $NCl > 0$ and $x^{\vec{feature}}$ is not empty **do**
    assign each data point from $x^{\vec{feature}}$ to visual and language cluster (Winner-takes all, see Eq. (5.3))
    for $j = 1 : N_{feature}$ **do**
      for $i = 1 : M$ **do**
        $T_{ij} \leftarrow$ how many times was class i actually classified as j
      end for
    end for
    $[im, jm] \leftarrow \text{argmax}_i \, \text{argmax}_j \, T_{ij}$
    $x_{NCl_{del}}^{\vec{feature}} \leftarrow$ data points assigned to both $K_{jm}^{feature}$ and L_{im}
    $\Theta_{new_{NCl}}^{feature} \leftarrow N(x_{NCl_{del}}^{\vec{feature}})$ learn Gaussian on the to be deleted data
    $x^{\vec{feature}} \leftarrow$ delete all data points assigned to both $K_{jm}^{feature}$ and L_{im}
    $NCl \leftarrow NCl - 1$
    relearn $K_j^{feature} \sim N(\vec{m_k}, \vec{S_k})$ on new data $x^{\vec{feature}}$ with NCl number of clusters
  end while
end for
cluster visual data using new $\Theta_{new}^{feature}$ parameters (cluster centres $\vec{m_k}$ and covariance matrices $\vec{S_k}$ and perform One-step mapping (Model 1)

optimal), none or more than one cluster from language layer will be assigned to one cluster $K_j^{feature}$ in visual layer or vice versa .

To compute the accuracy, each cluster is assigned to the true (manual) class that appears most frequently in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned data points and dividing this by the total number of data points. It is important to notice, that true (manual) labels can be partly subjective.

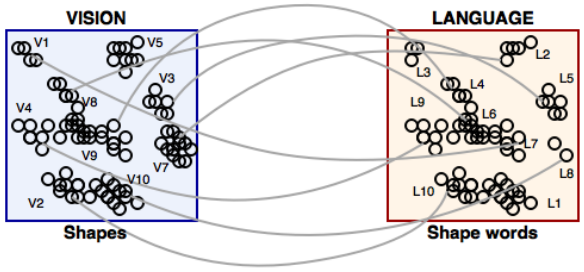


Figure 5.13: Finding mapping between visual and language layer (Shape feature).

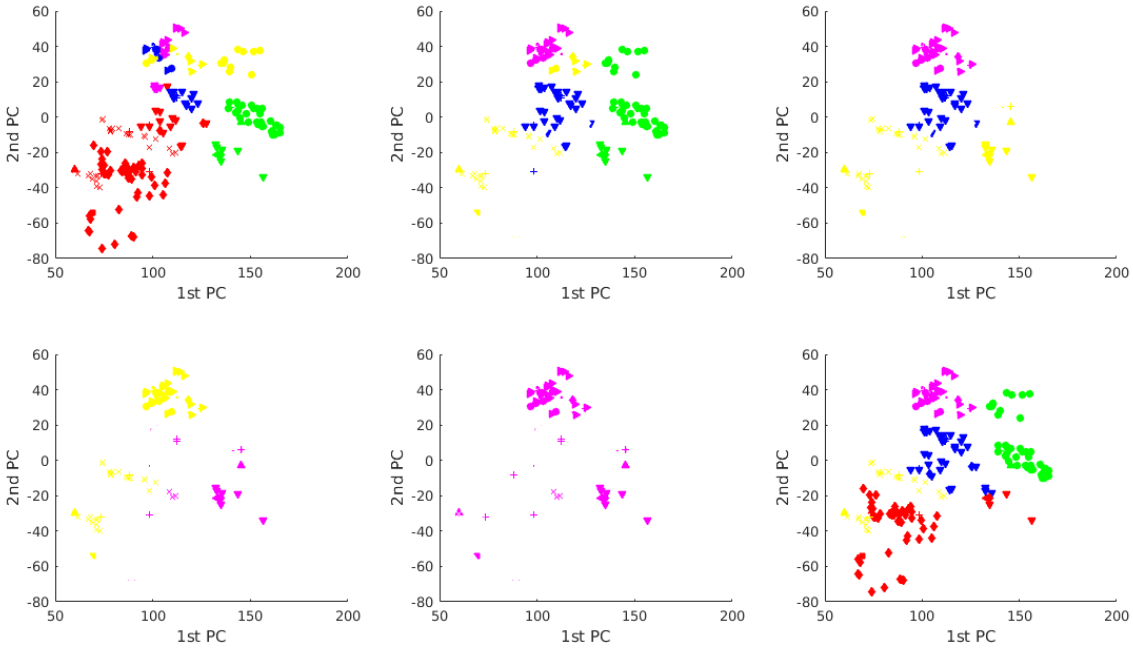


Figure 5.14: Sequential mapping - in each iteration one mapping is found and corresponding datapoints are removed, resting points are reclustered. Different colors in each iteration correspond to individual clusters.

Finding the mapping between the visual and language layer – variable lenght/grammar sentence

In a case of a fixed grammar, each word position corresponds to the individual visual feature (e.g. sentence is in a form <Size> <Colour> <Orientation> <Texture> <Shape>). On the contrary, in a case of variable-length/grammar sentence, every word in a sentence can be assigned to any visual feature (when taking into account restrictions given by a sentence model). This makes mapping a little bit more tricky. Nevertheless, the basic idea how the mapping is done remains same.

In this case, product of probabilites $P(w_k^m|L_i)$ that the given word is modelled by the language model L_i and probabilites that the given model L_i will occur on the specific

	o	o	o	o	o
o	29	2	4	0	13
>	0	11	4	0	6
▼	6	4	11	15	0
♦	1	0	0	45	1
×	0	0	0	28	0
✳	1	0	0	1	4
.	0	1	2	0	0
▲	1	0	0	1	0
<	1	1	0	0	1
+	2	0	3	5	0

(a) : Model 2-a,

	o	o	o	o
o	40	7	0	1
>	0	17	0	4
▼	6	0	30	0
♦	1	0	0	1
×	0	0	0	28
✳	1	0	0	1
.	0	6	0	1
▲	1	0	0	1
<	1	0	1	0
+	2	1	4	4

(b) : Model 2-b,

	o	o	o
o	8	0	0
>	21	0	0
▼	0	30	6
♦	1	0	1
×	0	0	28
✳	0	0	2
.	7	0	0
▲	0	0	2
<	0	1	1
+	2	3	6

(c) : Model 2-c,

	o	o
o	0	8
>	0	21
▼	6	0
♦	1	1
×	3	25
✳	1	1
.	0	7
▲	1	1
<	2	0
+	7	4

(d) : Model 2-d,

	o
o	8
>	21
▼	6
♦	2
×	3
✳	2
.	7
▲	2
<	2
+	11

(e) : Model 2-e,

	o	o	o	o	o
o	40	8	0	0	0
>	0	21	0	0	6
▼	0	0	33	3	0
♦	0	1	2	44	0
×	0	0	0	0	28
✳	1	0	0	1	0
.	0	7	0	0	0
▲	1	0	0	0	1
<	0	0	1	1	0
+	2	2	3	2	2

(f) : Model 1

Figure 5.15: Mapping: columns correspond to the visual clusters and rows to different language classes. Model 2 a,–e, corresponds to the sequential removing of the clusters based on the mapping which is found in each step (by red colour are denoted points which will be deleted from the dataset corresponding to mapped language cluster i and visual cluster j). After each cluster removal, remaining points are clustered based on the visual features and the second best assignment is found. Model 1 (f) shows how the one-step mapping would be performed.

position in a sentence is summed up over all coocurence of a given visual cluster $K_j^{feature}$ and language model L_i. These values are stored in a variable $T(i, j, feature)$. From these values we select the highest number over all clusters and features:

$$[im, jm, fm] = \underset{i}{\operatorname{argmax}} \underset{j}{\operatorname{argmax}} \underset{feature}{\operatorname{argmax}} T(i, j, feature), \tag{5.6}$$

and assign language model L_{im} to a visual cluster K_{jm}^{fm}. Afterwards, datapoints belonging to both L_{im} and K_{jm}^{fm} are deleted from the dataset and visual feature fm is reclustered.

The whole algorithm is described in Alg. 8.

The Algorithm 8 can be further extended when we incorporate language model and corresponding probabilities of sequence of individual visual features. In that case, instead of only counting coocurence of visual and language classes, I do the following computation:

$$T_{i,j}^{feature} = \sum_{k; v_k^{feature} == j} l(w_k^m | L_i) * P(L_i | T_t, L_{i-1}) * P(T_t | T_{t-1}), \tag{5.7}$$

where t is time.

The mapping in an ideal case when there is 100% word-recognition accuracy is shown in Fig. 5.17. The more realistic case with variable accuracy of word-recognition and also some misclassification errors in visual clustering is shown in Fig. 5.18. These are

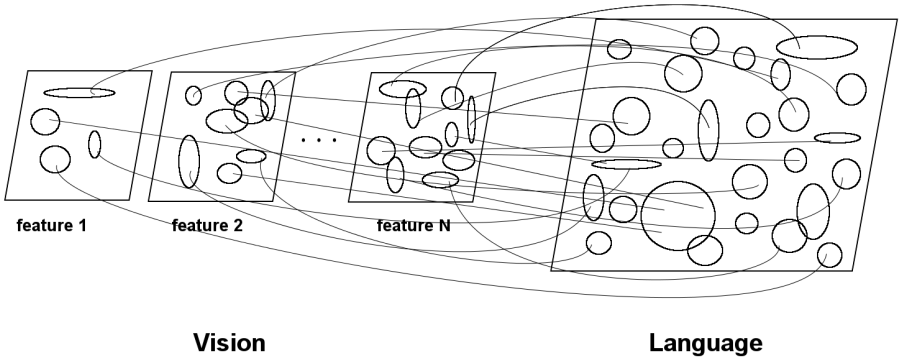


Figure 5.16: Mapping vision to language in a case where all language data are clustered altogether.

Algorithm 8 Sequential mapping – variable sentence

Inputs:
language clusters L_i $(i \in 1:M)$, visual clusters $K_j^{feature} \sim N(\vec{m_k}, \vec{S_k})$,
$j \in 1 : N^{feature}$ visual input data $\vec{x}$ and corresponding language data
$W_i = w_1, \ldots, w_k$, number of clusters $N^{feature}$ for each feature $feature$
Output:
mapping between all visual classes $K_j^{feature}$ and language classes L_i

while $\sum(NmbCl^n) > 0$ and $x^{\vec{feature}}$ is not empty **do**
$l_k^m \leftarrow$ assign each word w_k^m from each sentence k to language cluster (Winner-takes all, see Eq. (5.3), $l_k^m = \text{argmax}_i(P(w_k^m|L_i))$
for $feature \in$ {size, colour, orientation, texture, shape} **do**
$v_k^{feature} \leftarrow$ assign each datapoint $x_k^{feature}$ to a visual cluster (Winner-takes all, see Eq. (5.3), $v_k^{feature} = \text{argmax}_j(P(x_k^{feature}|K_j^{feature}))$
for $j = 1 : N^{feature}$ **do**
for $i = 1 : M$ **do**
$T_{ij}^{feature} \leftarrow$ how many times did visual class i coocurred with language class j $(T_{ij}^{feature} = \sum_{k;v_k^{feature}==j} \sum_m (l_k^m | l_k^m == i))$
end for
end for
end for
$[fm, im, jm] \leftarrow \text{argmax}_{feature}\, \text{argmax}_i\, \text{argmax}_j\, T_{i,j}^{feature}$ (visual cluster K_{jm}^{fm} is mapped to language cluster L_{im})
$\vec{x_{NCldel}^{fm}} \leftarrow$ data points assigned to both K_{jm}^{fm} and L_{im}
$\Theta_{new}^{feature}{}_{NCl} \leftarrow N(\vec{x_{NCl}^{feature}}_{del})$ learn Gaussian on the to be deleted data
$x^{\vec{feature}} \leftarrow$ delete all data points assigned to both K_{jm}^{fm} and L_{i_m}
$NmbCl^{fm} \leftarrow NmbCl^{fm} - 1$
relearn $K_j^{fm} \sim N(\vec{m_k}, \vec{S_k})$ on new data $x^{\vec{feature}}$ with $NmbCl^{fm}$ number of clusters
end while
cluster visual data using new $\Theta_{new}^{feature}$ parameters (cluster centres $\vec{m_k}$ and covariance matrices $\S_k$ and perform One-step mapping (Model 1)

only illustrative examples to explain how the mapping is performed, restricted on a low number of visual classes described with the low number of words. In these examples, it

is also imposible to see how the sequential mapping is performed in a case of variable grammar because the number of each visual feature was restricted to two. Therefore, only one-step mapping is performed in this case. However, the idea of sequential mapping is described in a detail in the Alg. 8 and remains the same as for the fixed legth/grammar sentence. The real examples with results are described in the Section 7 of this thesis.

In a case of clusters with non-equal size, it is also important to normalize the sumed values by the number of datapoints assigned to the corresponding visual cluster.

No.	Vision						Language					
	Size		Colour		Shape							
	1	2	1	2	1	2	1	2	3	4	5	6
1	1	0	1	0	0	1	0	0	0	1	0	0
2	0	1	0	1	0	1	0	0	1	1	1	0
3	1	0	0	1	0	1	0	1	0	1	0	0
4	1	0	0	1	1	0	0	0	0	0	1	1
5	0	1	1	0	0	1	1	0	1	1	0	0
6	0	1	0	1	1	0	0	0	1	0	1	1
7	0	1	1	0	1	0	0	0	1	0	0	1
8	1	0	1	0	1	0	1	1	0	0	0	1

(a) : Clustering results for individual datapoints (No.) (winner-takes all) (1 denotes that the datapoint is assigned to the corresponding cluster). E.g. 1st datapoint is image of a small blue cube with the sentence "*Cube*", 2nd datapoint is the image of a big red cube with the sentence "*Big red cube*" (therefore there is 1 for a language cluster 1, 3 and 4 which corresponds to 3 words in the sentence).

Size		Language					
		1	2	3	4	5	6
Vis	1	1	2	0	2	1	2
	2	1	0	4	2	2	2

(b) : Confusion matrix (visual feature Size).

Colour		Language					
		1	2	3	4	5	6
Vis	1	2	1	2	2	0	2
	2	0	1	2	2	3	2

(c) : Confusion matrix (visual feature Colour).

Shape		Language					
		1	2	3	4	5	6
Vis	1	1	1	2	0	2	4
	2	1	1	2	4	1	0

(d) : Confusion matrix (visual feature Shape).

Figure 5.17: Mapping vision to language in a case of variable length/grammar sentence – ideal case. In the ideal case all datapoints are clustered correctly (all individual visual features as well as all words). Clustering results are shown in a subfigure a, (in this simplified example we consider only 2 different colours – red/blue, 2 shapes – cube/shere and 2 sizes – big/small). Confusion matrices (b,–d,) show coocurence of visual and language clusters for individual visual features. In this case one-step mapping can be performed since clustering cannot be further improved. The shades of grey show the time-sequence of mapping indivudual clusteres (darker shades indicate mapping which is performed earlier). In particular, first are mapped clusteres with the highest coocurence (visual clusters 2 (Size), 1 (Shape), and 2 (Shape) are mapped to language clusters 3, 5, and 6 respectively) then the second highest coocurence is found and visual cluster 2 (feature Colour) is mapped to the language cluster 5. From the resting clusters the highest coocurence is found so the visual clusteres 1 (feature Size) and 1 (feature Colour) are mapped to the language clusters 2 and 1 respectively.

No.	Vision						Language					
	Size		Colour		Shape							
	1	2	1	2	1	2	1	2	3	4	5	6
1	1	0	1	0	1	0	0	.6	0	.4	0	0
2	1	0	0	1	0	1	.1	.2	.8	.9	.7	.3
3	1	0	0	1	0	1	0	1	.6	.4	0	0
4	1	0	1	0	1	0	0	.1	0	0	.9	1
5	0	1	1	0	0	1	1	.2	.8	.7	0	.3
6	0	1	0	1	1	0	0	.2	1	0	.8	1
7	0	1	0	1	1	0	.1	.2	.7	0	0	1
8	1	0	1	0	1	0	.7	.5	.1	.6	.1	1

(a) : Clustering results for individual datapoints (No.) (winner-takes all). Vision: 1 denotes that the datapoint is assigned to the corresponding cluster, darker shade of grey denotes incorrectly classified datapoints. Language: probability that the datapoint belongs to the given model is a product of probability that an individual word is described by the given model $P(w_k^m|L_i)$ and probability from a sentence model that the given model L_i will occur on this position in a sentence $P(L_i|T_i) * P(T_i|T_{i-1})$.

Size		Language					
		1	2	3	4	5	6
Vis	1	.8	2.4	1.5	2.3	1.7	2
	2	1.1	.6	2.5	.7	.8	2.3

(b) : Confusion matrix (visual feature Size).

Colour		Language					
		1	2	3	4	5	6
Vis	1	1.7	1.4	.9	1.7	1	2.3
	2	.2	1.6	3.1	1.3	1.5	2.3

(c) : Confusion matrix (visual feature Colour).

Shape		Language					
		1	2	3	4	5	6
Vis	1	.8	1.6	1.8	1	1.8	4
	2	1.1	1.4	2.2	2	.7	.6

(d) : Confusion matrix (visual feature Shape).

Figure 5.18: Mapping vision to language in a case of variable length/grammar sentence – non-ideal case. In this case not all data are clustered correctly. Clustering results are shown in a subfigure a, (in this simplified example we consider only 2 different colours – red/blue, 2 shapes – cube/sphere and 2 sizes – big/small). Confusion matrices (b,–d,) show coocurence of visual and language clusters for individual visual features (similarly to confusion matrices in Fig. 5.17 here are sumed probabilities of language models). The shades of grey show the time-sequence of mapping indivudual clusteres (darker shades indicate mapping which is performed earlier). Eventhough in this simplified case it is not obvious, the mapping should be in this case performed sequentially. Which means that after any mapping is found, corresponding visual feature is reclustered (see Alg. 8).

Conjunction of visual and language layer

After the mapping between visual and language layer is found, the resulting likelihoods are sent to the second layer. In the second layer, results from auditive and visual layer are combined (see Fig. 5.19). The assignment to the class is found for each data point from the conjunction of visual *(Vis)* and auditive *(Aud)* likelihoods to the appropriate clusters (representing types of features).

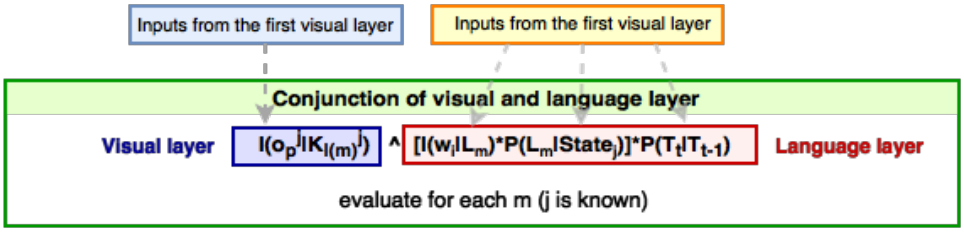


Figure 5.19: Combining visual and language input in a second layer. The conjunction ($\wedge$) between likelihoods from visual and language first layers is computed to get the final assignment of the data point.

For a fixed length/grammar sentence the transition probabilities from the sentence model are restricted to the diagonal case (see Fig. 5.10a).

$$P_{fin}(im, fm) = l(O|K_{jm}^{fm}) \wedge [l(W_k|L_{im}) * P(L_{im}|State)], \quad \forall jm, fm \tag{5.8}$$

3 types of fuzzy conjunction (triangular norm) are compared:

- Standard (Gődel) fuzzy conjunction:

$$Vis \wedge Aud = \min(Vis, Aud) \tag{5.9}$$

- Algebraic product fuzzy conjunction:

$$Vis \wedge Aud = Vis * Aud \tag{5.10}$$

- Lukasiewicz fuzzy conjunction:

$$Vis \wedge Aud = \begin{cases} Vis + Aud - 1 & \text{if } Vis + Aud - 1 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

The observed datapoint will be assigned to the visual clusters K_{imax}^{fm}:

$$imax = \underset{im}{\operatorname{argmax}}\, P_{fin}(im, fm), \forall fm \tag{5.12}$$

and corresponding language cluster.

5.5 Architecture used for processing data from an iCub simulator

The multimodal hierarchical architecture used for processing data from an iCub simulator and real iCub is a specific case of the general architecture described above. It consists of separate processing of visual and language information, which are consequently mapped one to the other (see Fig. 5.20). Individual steps of visual and language data processing are described in a Chapter 6, Section 6.2.

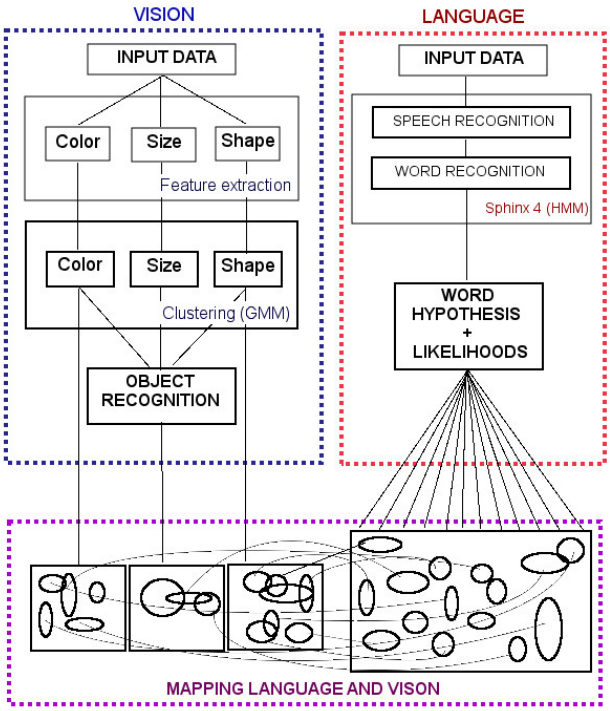


Figure 5.20: Multimodal architecture used for experiments with iCub simulator and real iCub.

Chapter 6

Datasets

6.1 Input data description and visualisation – Artificial data

The input data consist of visual and language inputs. The Gaussian noise was added to both visual and language data to enable to investigate the relation between fuzziness of the input and the error in the visual and language layer clusterizations.

Visual input

The visual scene is composed of an object in a centre of the scene. The scene size (retina) is 10x10 pixels and the size of each object is 7x7 pixels. The position of an object can be varied or fixed. Each of the presented objects has five visual features: size, colour, orientation, texture and shape (see Table 6.1).

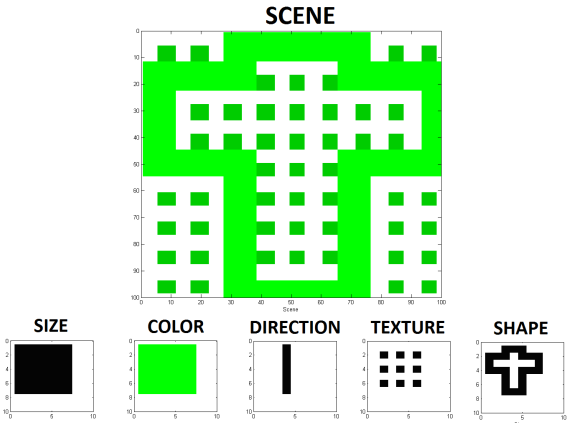


Figure 6.1: Visual input data: An example of visual input and visualization of features it is composed of.

Altogether this gives 4050 possible combinations of features. The example of a visual input can be seen in Figure 6.1 and a complete set of used visual features is visualized in Fig. 6.2. The stimuli with variable fuzziness of the objects features (gaussian noise was added) and position were presented to study the relation between the noiseness of the data and the resulting clusterization error of the visual and multimodal layer.

Feature	Number of classes	Number of data points/class
Shape	10	405
Colour	9	450
Texture	5	810
Size	3	1350
Orientation	3	1350

Table 6.1: Overview of input data.

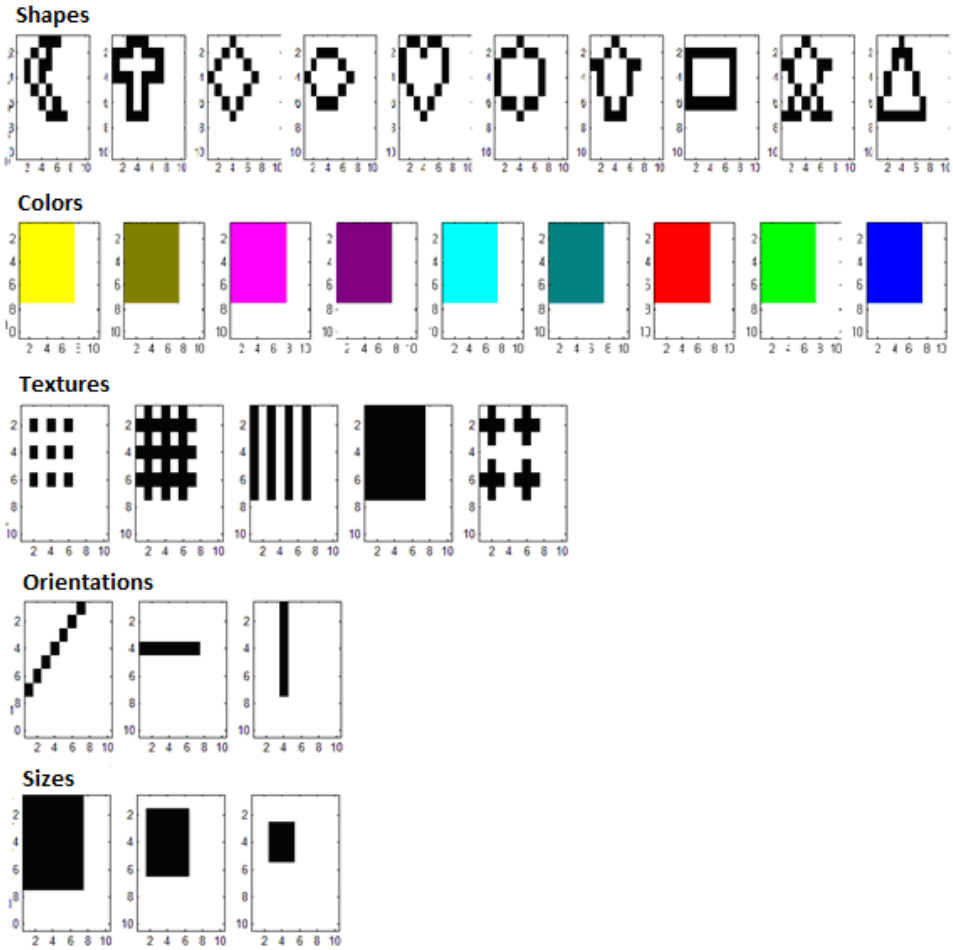


Figure 6.2: Visual features – A complete set of features used for generating a visual input.

Firstly, Principal Component Analysis (PCA) was applied to investigate separability of the visual data. Fig. 6.3 shows a projection onto the first two principal components (PCs) separately for each visual feature. I have also plotted contributions of individual PCs to the overall variance in the data (based on the eigenvalues of the principal vectors). We can see that in the case of shape feature, the first two PCs capture about 73% variance and 6 PCs are needed to capture 90% of variance generated by individual features.

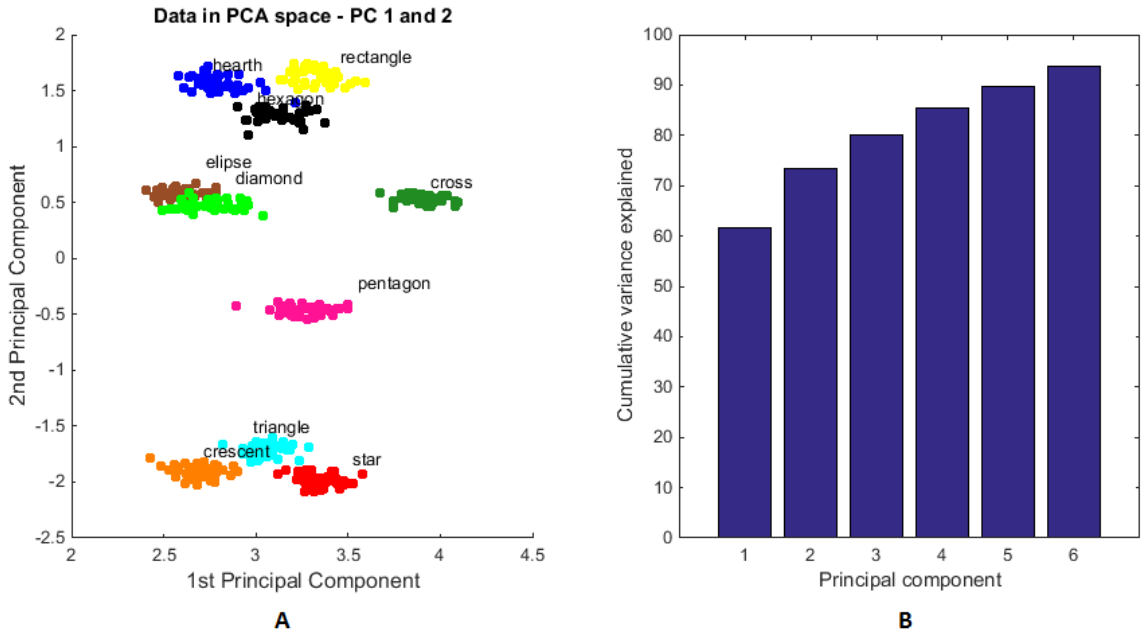


Figure 6.3: Feature space visualization — (A) variability due to shapes with a 10% noise level. Data are visualized in the space of first two principal components. (B) Contributions of first n principal components to explain 90% of variance in the data. Note: For visualization purposes, only 1/10 of the data points were plotted.

In order to get a more quantitative understanding of the problem, I computed Bhattacharyya distance matrix among each feature type separately for each feature and visualised the normalized histogram of these distances (see Fig. 6.4).

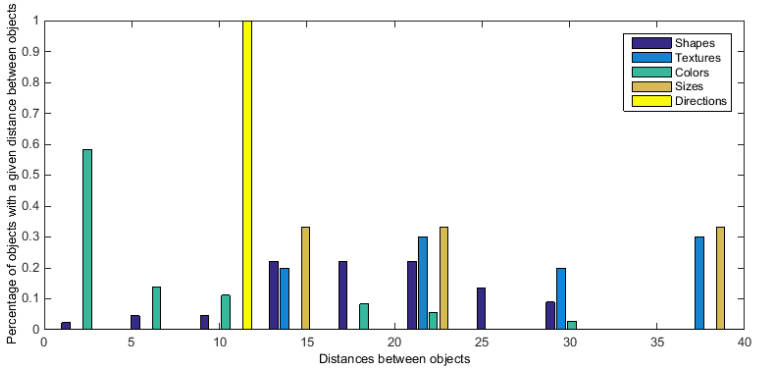


Figure 6.4: Bhattacharyya distances. Histogram of inter-distances between feature types shown separately for each visual feature (normalized to a number of object).

Bhattacharyya distance is a more general case of Mahalanobis distance, which can be used also in cases where the variance of the data is not same for all distributions (to enable generalization of this particular case, Bhattacharyya distance is used instead of Mahalanobis distance). The higher distances the better clusterization results we can achieve. As can be seen, only orientations have the same inter-distances for all 3 feature types. Colours have the highest number of small inter-distances (the neigbours are very

close which results in the overlap of the distributions and higher clusterization error). Also as can be seen in the PCA space, colours are distributed linearly which results in small inter-distances between neigbours and large distances between the farthest ones. Eventhough there are 9 types of shape, shapes are distributed more uniformly, compared to colours, with the closest shapes "diamond" and "elipse".

Language input

Language data (English sentences) (see Fig. 6.6) are processed simultaneously with the visual input (see Figure 6.1). These auditory inputs were encoded to a high-dimensional vector form using a PatPho, a phonological pattern generator, which parsimoniously captures the similarity structures of the phonology of monosyllabic and multisyllabic words [338]. PatPho uses the concept of a syllabic template: a word representation is formed by combinations of syllables in a metrical grid, and the slots in each grid are made up by bundles of features that correspond to consonants and vowels. The length of all words in a vector form after encoding is 54-d, however the same length of words is not a need. The number of presented words corresponds to the number of observed visual features (in our particular case this gives 3 (size) + 10 (shape) + 9 (colour) + 5 (texture) + 3 (orientation) = 30 words). An example of language input for sentence "*Big dotted cross*" is shown in Fig. 6.5.

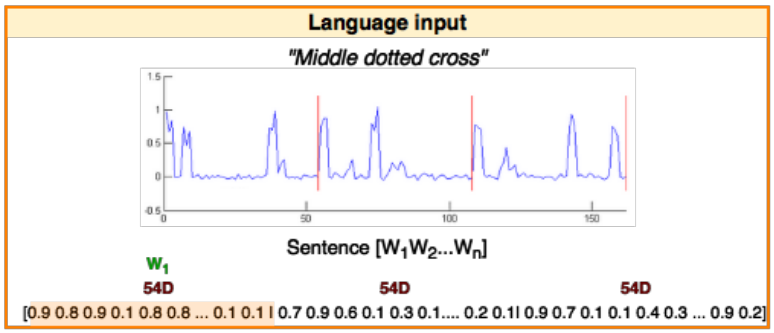


Figure 6.5: Input language data – sentences with variable length in a vector form (encoded to a high-dimensional vector form using a PatPho). Example shows language input for the sentence "*Small yellow horizontal plain rectangle.*".

If we consider language data as d-dimensional vectors with a same fixed length and we do not consider data as a time-series, we can visualize them in a same manner as visual data. Firstly, PCA was applied to language data (see Fig. 6.7) and all words are visualized altogether. In this case, five PCs were needed to explain 90% of data variance.

In order to get a deeper insight into the problem, I also visualised closeness of words by a dendrogram (see Fig. 6.8).

6.2 iCub simulator and physical iCub

iCub robotic platform and iCub simulator

The experiment used a simulated [339] and a physical [340] iCub robot. The iCub (Fig. 6.9c) is an open-source humanoid robot with the size of a three and a half year-old

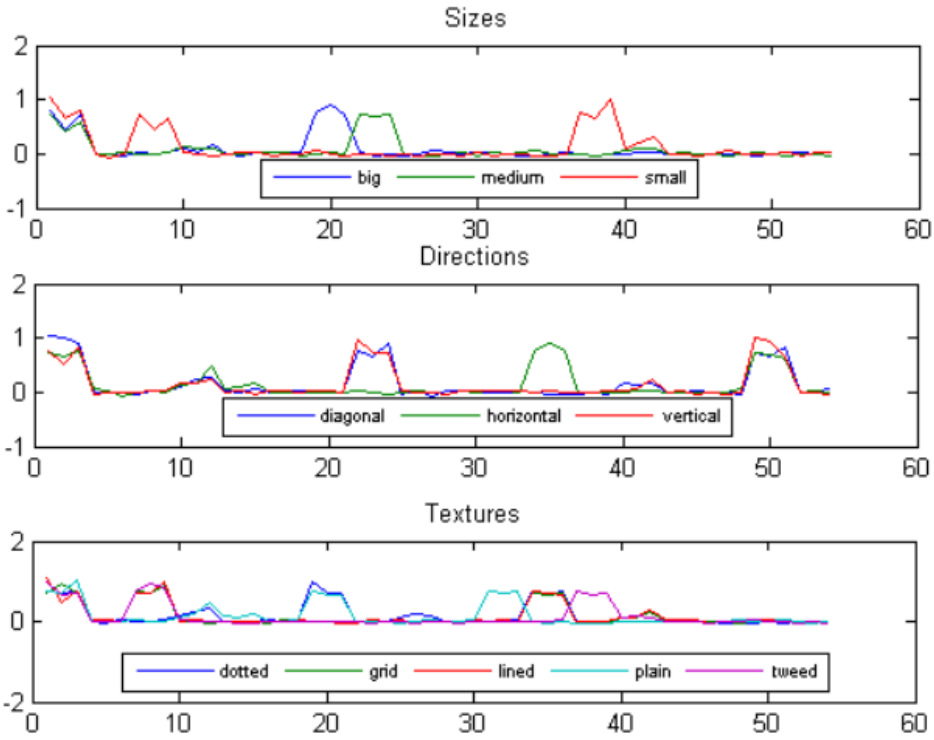


Figure 6.6: Language input data – words after encoding by a PatPho generator (with an added noise, joined by visual features).

child, fully articulated hands as well as a head-and-eye system which makes him ideal for cognitive experiments. The iCub simulator has been designed to reproduce, as accurately as possible, the physics and the dynamics of the robot and its environment [339]. The simulator and the actual robot have the same interface supporting YARP [341], which is a robot platform for interprocess communication and control of the physical and simulated robot in a real-time.

Visual and language inputs

The visual scene was composed of an object in a centre of the scene with slightly varying position. The visual features (size, shape and colour) of an object were varied. We compared two separate datasets in the paper. Real-world dataset: visual sensoric data were acquired from the cameras of the physical iCub robot who observed the simple objects placed on the whiteboard in front of his eyes (see Fig. 6.9, (c) and (d)) (204 instances, 3 sizes, 5 colours and 7 shapes). Simulated dataset: data were acquired from the iCub simulator (see Fig. 6.9, (a) and (b)) where we placed objects generated in the Blender software (432 instances, 3 sizes, 6 colours and 6 shapes). In both cases, the full-colour images were saved in the .ppm format and further processed in MATLAB.

The spoken language input were sentences pronounced by a non-native English speaker describing the image in the format: <size> <colour> <shape> (e.g. "*Small red triangle*") and were processed simultaneously with the visual input.

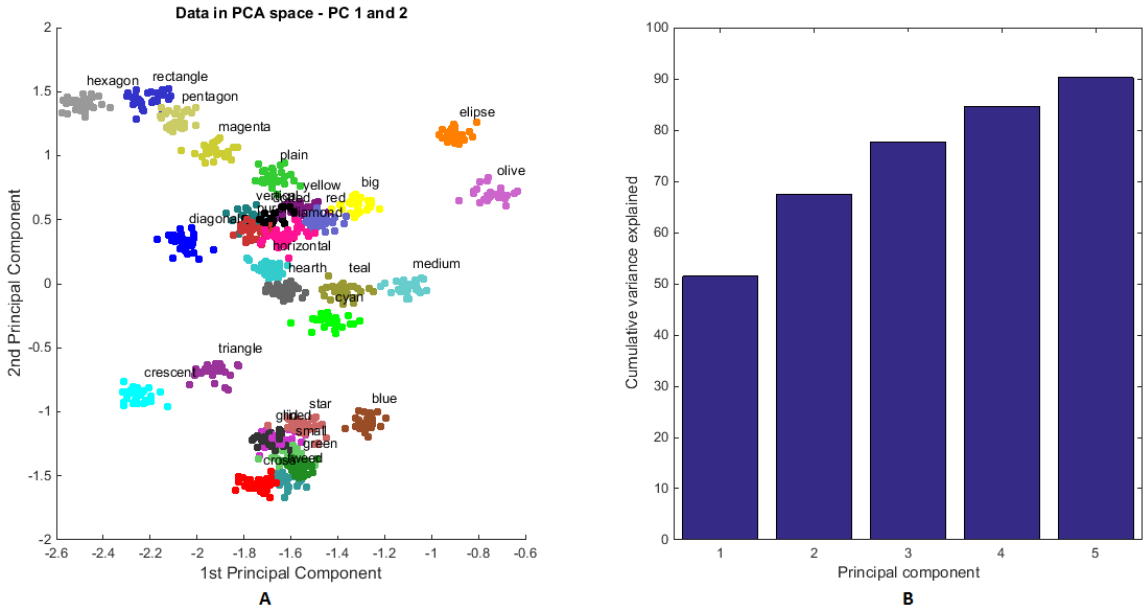


Figure 6.7: Feature space visualization — (A) variability of language data (30 different words) with a 10% noise level. Data are visualized in the space of first two principal components. (B) Contributions of first n principal components to explain 90% of variance in the data. Note: For visualization purposes, only 1/20 of the data points were plotted.

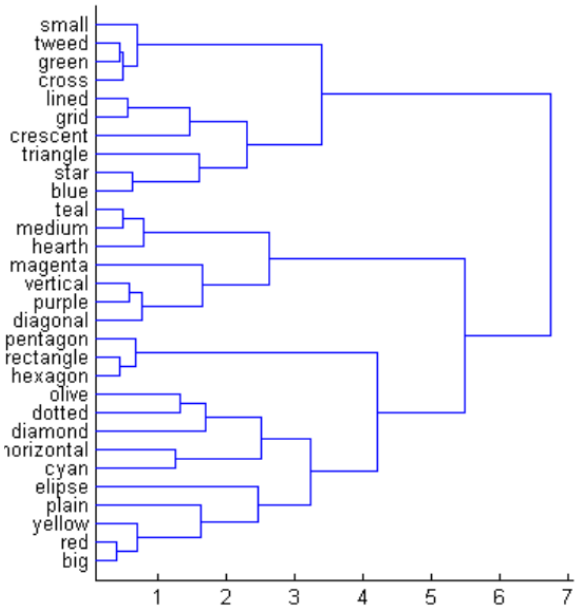
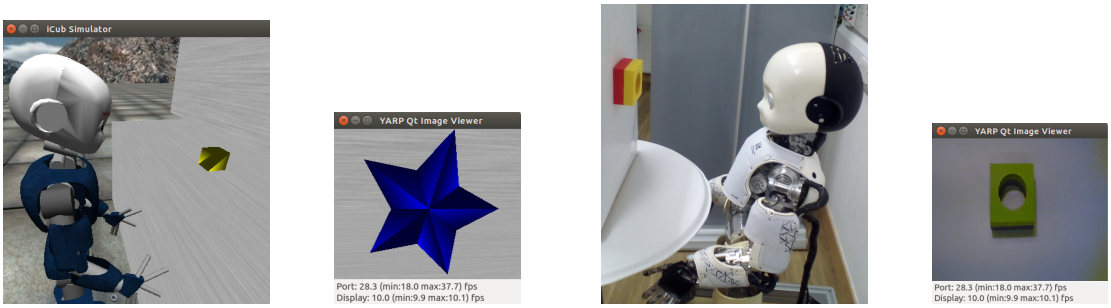


Figure 6.8: Visualising closeness of individual words by a dendrogram.

Speech recognition

CMU Sphinx was used for speech recognition, which is an open-source flexible Markov model-based speech recognizer system [342]. Sphinx itself offers large vocabulary, but we



(a) : iCub simulator. **(b) :** Blender object. **(c) :** Physical iCub. **(d) :** Real object.

Figure 6.9: Experiment design and corresponding input data.

created our own task-specific smaller vocabulary using online IMtool that produces a dictionary based on a CMU dictionary and matches its language model.

The 10 best hypothesis with corresponding scores were saved for each utterance (those are log-scale scores of the audio matching the model). Because the scores for hypothesis of each word in the sentence were needed for further evaluation, the words were pronounced with the large pauses and the end of the sentence was marked by the word "*STOP*".

This corresponds to the findings of Werker et al. [343] that infant-directed words are usually kept short with large pauses between words and to the study of Brent and Siskind [344] which showed that frequency of exposure to a word in isolation predicts better whether that word will be learned than the total frequency of exposure to that word.

Image processing

The image inputs are processed using standard MATLAB functions. First, the image is morphologically opened with a disk-shaped structuring element (*imopen*) to remove the noisy background of an image, then all greyish pixels are removed and the image is converted from the true colour RGB to the greyscale intensity image (*rgb2gray*). Finaly, the intensity image is converted to a binary image (*threshold*).

Figure 6.10: Image processing – original image, removal of the background, converting to BW image and filling the holes for the images acquired through iCub simulator and physical iCub.

Afterwards the properties of image regions are measured using the function *regionprops*. Individual visual features (shape, colour, size) are subsequently processed separately. Following features were used: Colour (3: Average RGB of the selected region), Size (6: Parimeter of an object, distance from the centroid to the left corner of the bounding

box, width and length of the bounding box), Shape (13: Area, centroid, major axis length, eccentricity, orientation, convexArea, FilledArea, EulerNumber, EquivDiameter, Solidity, Extent, Perimeter). To obtain shape features we cropped and resized the image to equalize the size of objects. All data were normalized using a standard score normalization.

Chapter 7
Results

7.1 Results - visual layer

Comparing classification accuracy of different algorithms

Final classification accuracy of individual features is listed for all compared algorithms in Table 7.1 (10 repetitions). To compute accuracy, each cluster is assigned to the class that appears most frequently in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned data points and dividing this number by the total number of data points.

Accuracy [%]	Size	Colour	Orientation	Texture	Shape
GMM_{sup}	100 ± 0	81 ± 0	100 ± 0	100 ± 0	100 ± 0
GMM_{unsup}	96 ± 10	63 ± 8	77 ± 16	74 ± 13	76 ± 9
k-means	100 ± 5	56 ± 10	66 ± 11	80 ± 8	30 ± 5
SOM	97 ± 14	87 ± 7	86 ± 17	85 ± 10	79 ± 6
GWR	100 ± 0	82 ± 4	99.9 ± 0.1	100 ± 0	100 ± 0

Table 7.1: Classification of visual input: Comparison of GMM supervised, GMM unsupervised, k-means, SOM and GWR (Growing when required neural gas) [345, 346] algorithms. Listed means and standard deviations are computed from 10 repetitions.

The GMM algorithm performance is compared to the supervised GMM algorithm, k-means, SOM and GWR algorithm (growing when required neural gas, implementation in MATLAB [346]) [345] (both SOM and GWR had 100 nodes) . The comparison for real-world dataset and simulated dataset with Blender objects can be seen in the Table 7.2. It should be mention that even though SOM and GWR are considered to be unsupervised algorithms, the way we are labeling them corresponds to the model with highly overestimated number of clusters (number of nodes corresponds to the number of clusters – in this case 100), which means that this algorithm is partly overfitting the data (which was omitted by dividing the set to testing and validation data) and its performance is much closer to supervised algorithms.

Impact of added noise on classification accuracy

The classification accuracy of shape and colour feature for different levels of noise is visualized in the Figure 7.1. Results for supervised GMM, unsupervised GMM, k-means

Accuracy [%]	Real-data Size	Real-data Colour	Real-data Shape	Blender Size	Blender Colour	Blender Shape
GMM sup.	83.3 ± 0	99.0 ± 0	81.4 ± 0	98.6 ± 0	97.9 ± 0	93.1 ± 0
GMM unsup.	76.2 ± 6.8	76.1 ± 9.1	56.1 ± 6.2	74.2 ± 10.1	60.9 ± 9.0	64.3 ± 7.2
K-means	67.8 ± 6	81.2 ± 1	53.1 ± 4.2	66.3 ± 0.2	77.1 ± 10.7	72.8 ± 6.9
SOM	69.6 ± 5.6	78.9 ± 6.8	54.2 ± 4.1	66.1 ± 4.2	81.7 ± 7.6	59.3 ± 6.2
GWR	89.9 ± 2.1	99.5 ± 0.4	76.6 ± 1.4	88.9 ± 0.7	98.1 ± 0.9	94.2 ± 0.6

Table 7.2: Comparison of clusterization accuracy of visual data. The mean and standard deviation from 100 repetitions is visualised.

and SOM algorithms are compared. The noise added to the raw data has a Gaussian distribution.

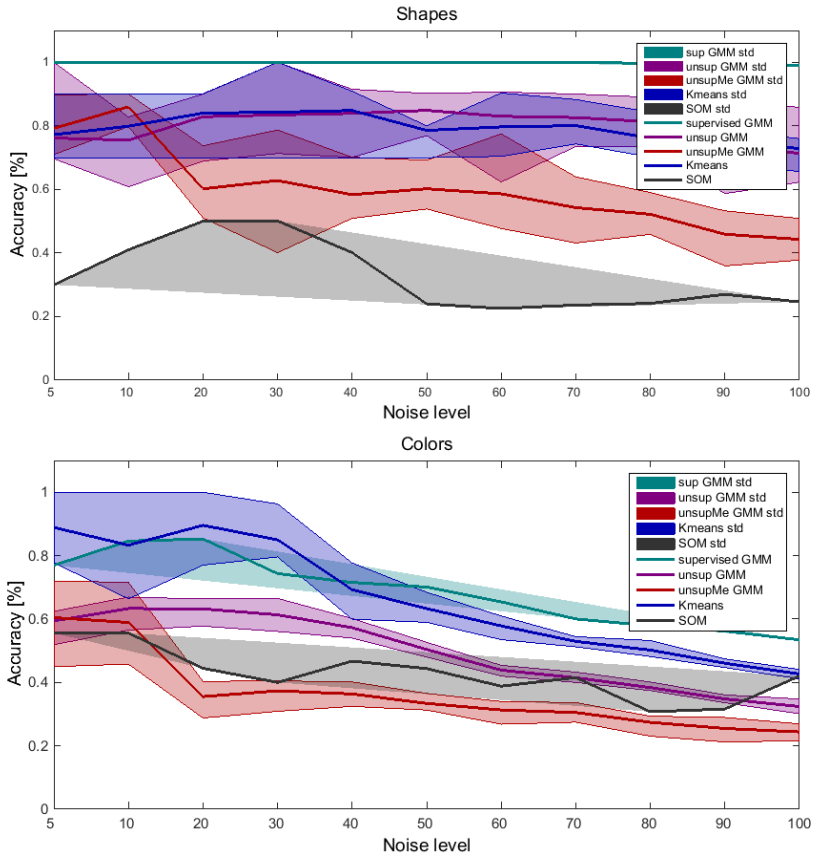


Figure 7.1: Impact of added noise on the classification accuracy for individual algorithms (supervised GMM, unsupervised GMM, k-means and SOM). The mean accuracy is visualised by the coloured line and std of accuracy is visualised by a coloured area (averaged over 10 repetitions). Data are visualised for shapes (upper) and colour (lower) features.

Object recognition – compositionality

The test of compositionality during the object recognition was performed only for artificially generated data. To test the compositionality of the architecture, I will perform the following experiment. In the training stage, we will not present to the architecture an object with a specific combination of features and then we will test whether the architecture is able to recognize the previously unseen objects.

For example there will be no *small hearth* (small (feature *size*) + hearth (feature *shape*)) presented in training dataset. The architecture can see *small rectangles*, *small triangles*, *middle hearths*, *big hearths*, etc. but no *small hearth*. The data in training and testing dataset for this specific example are visualized in the Fig. 7.2 and Fig. 7.3.

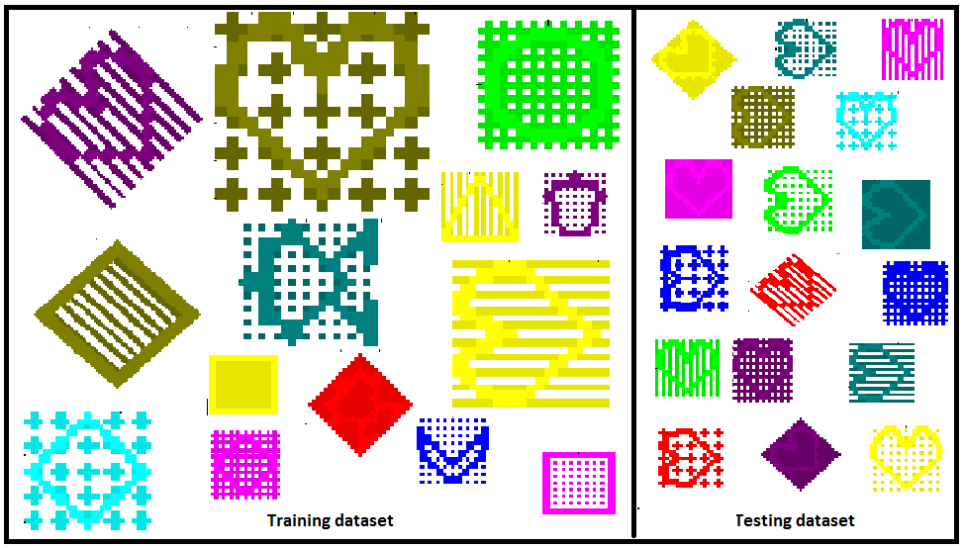


Figure 7.2: Training and testing dataset for an example where no "small hearts" are presented to the architecture during training

The accuracy for each eliminated combination of two features was computed (for the used set of features, it is altogether 338 combinations). The results for all individual combinations of specific type of feature are averaged over each feature. Results can be seen in Table 7.3.

After the reduction of data was done to test the compositionality of architecture for two features (during the training stage an object with a specific combination of features is not presented), the performance of clusterization for individual features was tested. The results are listed in the Table 7.4 (those are accuracies averaged over all 338 combinations grouped by specific combination of features).

In Figure 7.4 are listed similar results for combinations of three features (e.g. averages over all combinations for colour-texture-shape which are omitted in training stage and their recognition subsequently tested are listed under the label "CTSh"), four features and all five features.

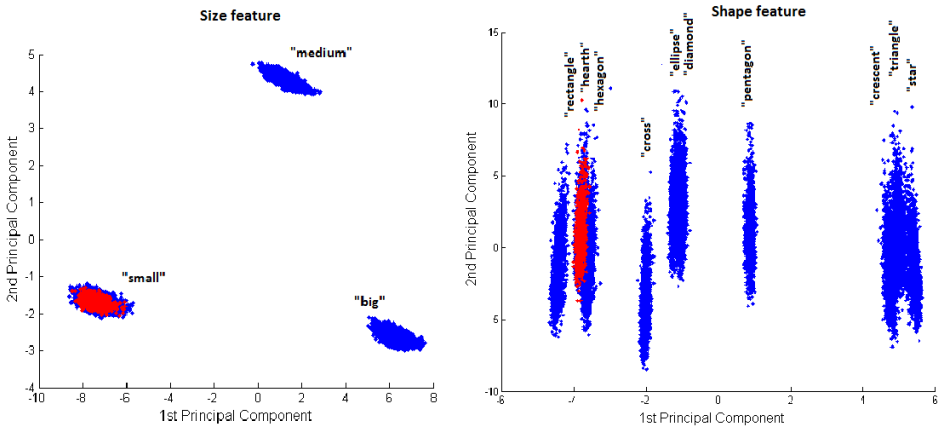


Figure 7.3: Compositionality – Visual data space visualization using PCA (training and testing dataset, noise level 5). Size feature data are on the left and shape feature data are on the right. Data are divided into training (blue) and testing (red) set. In this case "small hearts" are excluded from training data and are used for testing.

Accuracy [%]	Size	Colour	Orientation	Texture	Shape
Size	-	58 ± 7	76 ± 22	82 ± 20	70 ± 14
Colour	58 ± 7	-	48 ± 12	53 ± 9	45 ± 6
Orientation	76 ± 22	48 ± 12	-	68 ± 19	58 ± 14
Texture	82 ± 20	53 ± 9	68 ± 19	-	64 ± 14
Shape	70 ± 14	45 ± 6	58 ± 14	64 ± 14	-

Table 7.3: Testing compositionality of visual layer – testing for previously unseen combination of two features: In this Table, accuracy of testing stage is listed for each combination of features (accuracy for shape + size means average accuracy for all combination of shape-sizes which were missing during training stage. For example testing for "small hearts" and no "small hearts" were presented during training stage. The testing accuracy is computed for all combinations of features values.) – 20 repetitions. Results for standard GMM are shown.

Accuracy [%]	Size	Colour	Orientation	Texture	Shape
	96 ± 14	61 ± 6	79 ± 18	89 ± 19	74 ± 9

Table 7.4: Testing compositionality of visual layer – performance of clusterization of individual features in a case where specific combination of 2 features is not presented to the architecture, so the number of data is adequatelly reduced. Results for standard GMM are shown (20 repetitions).

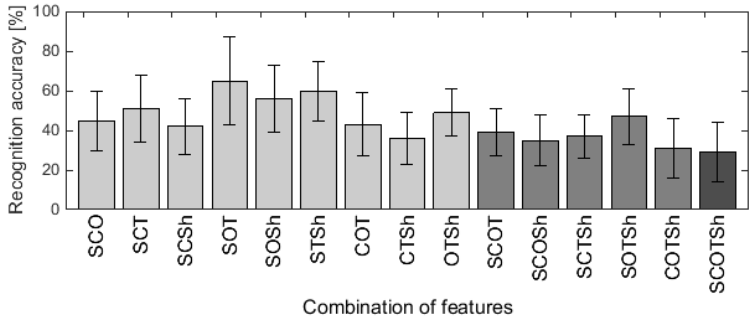


Figure 7.4: Testing compositionality of visual layer – testing for previously unseen combination of 3, 4 or 5 features: In Tables accuracy of testing stage is listed for each combination of features (S – size, C – colour, O – orientation, T – texture, Sh – shape). Accuracy for SCOSh means average accuracy for all combination of sizes-colour-orientation-shape which were missing during training stage. For example testing for "small red vertical hearts" and no "small red vertical hearts" were presented during training stage. The testing accuracy is computed for all combinations of features' values. – 20 repetitions. Results for standard GMM are shown.

7.2 Results - language layer

Ability to recognize separate words

In the first language layer, separate HMM models were constructed for each lexicon word and those were subsequently clustered. Firstly, HMMs with different number of hidden states were compared (see Fig. 7.5).

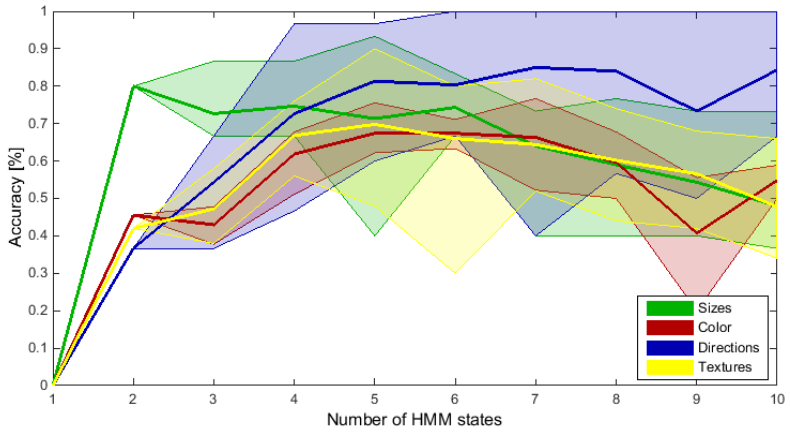


Figure 7.5: Compare accuracy of word recognition for HMMs with differing number of hidden states

Based on the preliminary results, I decided to use the HMM model with five hidden states in the first language layer. For the given lexicon, separate HMM models were constructed for each lexicon word w_i (the length of all sequences are set to be T = 54). There are 3, 9, 3, 5 and 10 clusters for words describing size, colour, orientation, texture and shape feature respectively. After fitting all models L_i, corresponding log-likelihoods

$ll_{ij} = \log l(w_j|L_i), 1 \leq i, j \leq N$ were generated (evaluate the log-likelihood of each of the N sequences given model L_i). K-means and agglomerative clustering method were used to cluster the sequences into K groups using log-likelihood distance matrix and their performance was compared. The case where words describing individual features are clustered separately was compared to the case when all features are cluster altogether (see Fig. 7.2).

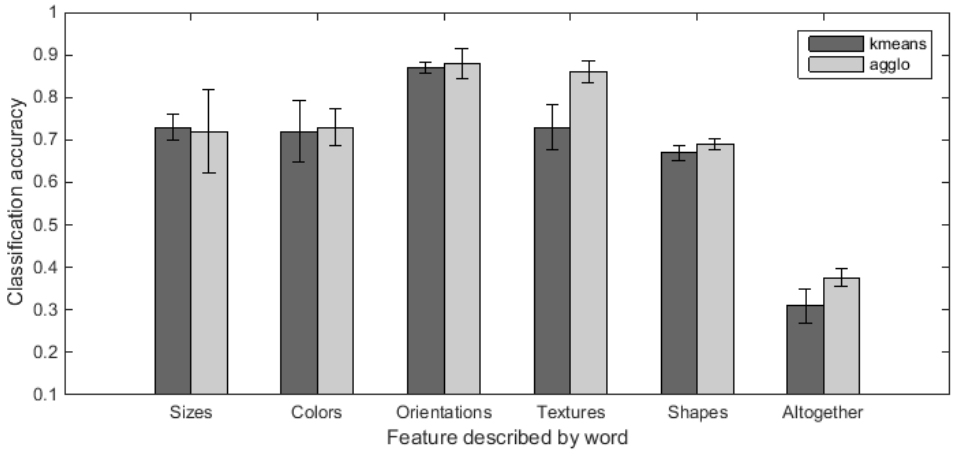


Figure 7.6: Accuracy of word recognition – clustering of HMMs representing words. K-means 1 and agglomerative 2 clustering methods are compared. Comparison of the case when words describing individual features are clustered independently and altogether. (10 repetitions)

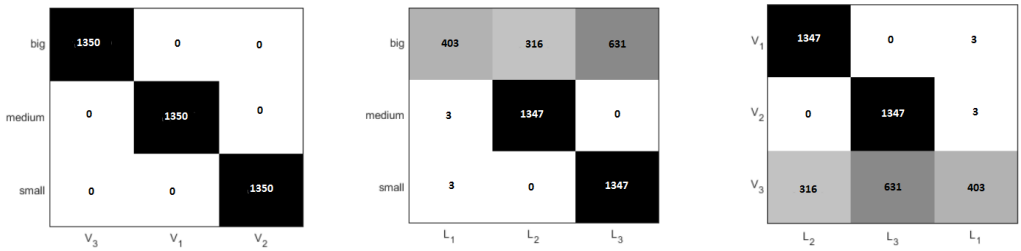
As can be seen from the Figure 7.6, the agglomerative clustering achieved comparable or significantly better results than k-means clustering in all cases. We can also see from the Figure 7.6, that clustering words altogether achieves for 30 words quite low accuracy (28% and 37.6 % for k-means and agglomerative clustering respectively). When the noise is added to the language data, these values are getting even lower.

7.3 Results – multimodal layer

Mapping visual and language layer – fixed length sentence

Firstly, both visual data and corresponding language data are clustered separately and the resulting outlabels are compared to true labels. In Figures 7.7a and 7.7b, confusion matrices of the visual and language layer clusterings respectively for the feature Size are shown. This means that the clusterings obtained from unimodal layers (*Predicted labels*) are compared to true labels (*Targets*) which are previously known, but hidden. The confusion matrix is created based on these outlabels where each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class (target).

Afterwards, to find mapping between visual and language clusterings, outlabels from one unimodal clusterization are called *Targets* and outlabels from the second are called *Predicted labels*. Assignment between the visual and language clusterings will enable us to assign correct ”names” or ”labels” to visual inputs or vice versa to find out which feature of visual input does the language word describe. Resulting confusion matrix



(a) : Visual layer clustering

(b) : Language layer clustering

(c) : Assignment between clusterings from language and visual layer

Figure 7.7: Finding mapping between vision and language (feature Size) – confusion matrices of resulting individual clusterings in Visual and language layer are visualized as well as assignment between visual (*Targets*) and language outlabels (*Predicted labels*). The number of datapoints assigned to the given class is shown (total is 4050 datapoints).

of assignment between clusterings from language and visual layers for a feature Size is shown in the Figure 7.7. In this case, outlabels from clustering in visual layer are *Targets* and outlabels from language layer are *Predicted label*. This can be changed throughout the experiment when our believes to individual modalities may vary as our knowledge or experience increases.

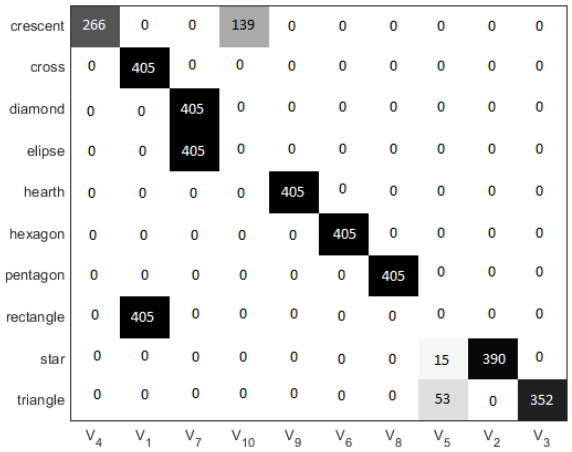
The mapping between outlabels and targets is found using the confusion matrix using either one-step mapping or sequential mapping. Both mappings are described in Section 5.4, specifically one-step mapping is formalized in Eq. 5.5 and sequential mapping is described in an Algorithm 7 (fixed grammar) and Algorithm 8 (variable grammar).

From Figure 7.7c, we can see that when we use one-step mapping, the corresponding assignment between unimodal visual and language clusterings in this particular case will be: $L_2 - V_1$ (medium), $L_3 - V_2$ (small) and $L_1 - V_3$ (big). This mapping is a bijection in this case, but does not have to necessarily be, since more visual clusters can be assigned to the same language cluster (or potentially vice versa).

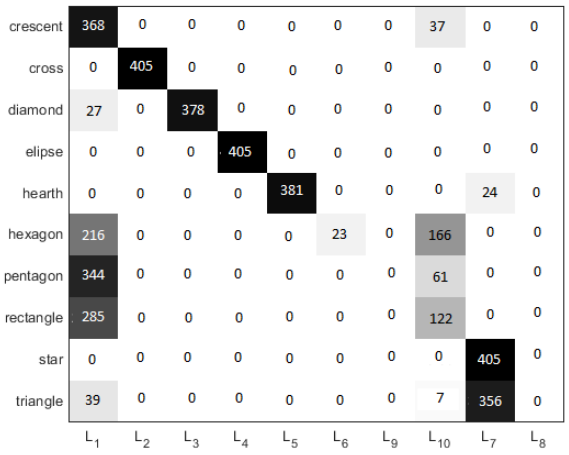
Confusion matrices for visual, language and multimodal layer for feature Shape are shown in Figures 7.8a, 7.8b and 7.8c, respectively.

For the feature Shape (see Fig. 7.8c), the mapping is harder to find. The assignment between clusterings found using an equation Eq. 5.5 is not bijective mapping. The best bijective mapping in this particular case would be: L_2 – V_1 (cross), L_7 – V_2 (star), L_{10} – V_4 (crescent), L_3 – V_7 (diamond), L_8 – V_5 (rectangle), L_6 – V_6 (hexagon), L_4 – V_7 (elipse), L_1 – V_8 (pentagon), L_5 – V_9 (hearth), L_9 – V_3 (triangle).

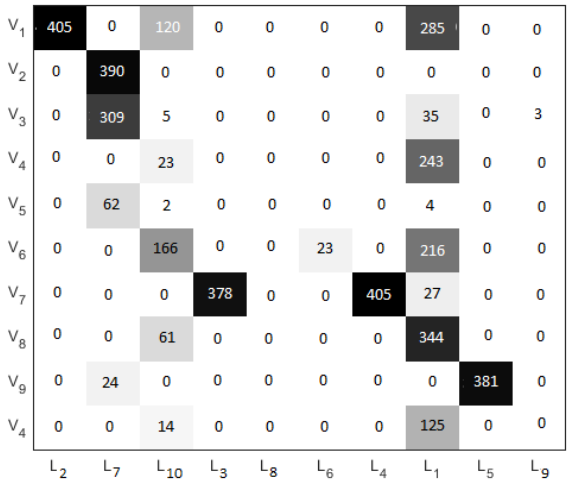
As can be seen, when we have achieved high-quality clustering (meaning well-separated clusters corresponding to respective true labels) in both (or at least one) unimodal layers, the unambiguous mapping between the clusters from unimodal layers can be found easily (see confusion matrix for feature Size in Fig. 7.7). On the other hand, when the clusterings from one or both unimodal layers are of a poor quality - e.g. highly overlapping clusters (see Fig. 7.8c), the unambiguous mapping between the two clusterings cannot be found (or is hard to find) which will result in further problems in object recognition and description, mainly in a case of the sentences with variable length.



(a) : Visual layer clustering.



(b) : Language layer clustering.



(c) : Assignment between clusterings from language and visual layer.

Figure 7.8: Finding mapping between vision and language (feature Shape) – confusion matrices of resulting individual clusterings in Visual and language layer are visualized as well as assignment between visual (*Targets*) and language outlabels (*Predicted labels*). The number of datapoints assigned to the given class is shown (total is 4050 datapoints).

In Table 7.5 and Table 7.6, accuracies for unimodal Vision, Language and vision to language mapping are listed (one-step mapping to sequential mapping is compared). The comparison was done for artificially generated visual data with corresponding language data generated by the PatPho generator (see Table 7.5). In the case of fixed grammar, the diagonal transition matrix was used while the generation of the sentences (see Fig. 5.10a). In Table 7.6, results for for real-world data from iCub and for Blender objects used in the iCub simulator are compared, specifically accuracies for unimodal vision (GMM), language (Sphinx), one-step mapping and sequential mapping (where mapping between vision and language is found in a stepwise mode (see Alg. 7)) are listed.

	Artificial data				
Accuracy [%]	Size	Colour	Orientation	Texture	Shape
Vision	76 ± 10	63 ± 8	77 ± 16	74 ± 13	76 ± 9
Language	73 ± 2	72 ± 4	87 ± 7	73 ± 2	67 ± 2
One-step mapping	70 ± 17	69 ± 4	68 ± 3	75 ± 6	61 ± 9
Sequential mapping	91 ± 2	88 ± 8	97 ± 6	75 ± 6	77 ± 5

Table 7.5: Comparison of One-step mapping and Sequential mapping for artificial data. The mean and standard deviation from 100 repetitions is visualised.

	Real-data			Blender		
Accuracy [%]	Size	Colour	Shape	Size	Colour	Shape
Vision	76 ± 7	76 ± 9	56 ± 6	74 ± 10	61 ± 9	64 ± 7
Language	71 ± 0	82 ± 0	78 ± 0	98 ± 0	96 ± 0	98 ± 0
One-step mapping	54 ± 4	58 ± 10	52 ± 5	67 ± 8	56 ± 6	62 ± 3
Sequential mapping	74 ± 15	87 ± 10	73 ± 5	96 ± 31	95 ± 1	92 ± 1

Table 7.6: Comparison of One-step mapping and Sequential mapping for data from iCub simulator (Blender) and physical iCub (real-data). The mean and standard deviation from 100 repetitions is visualised.

As can be seen, for artificial data, the sequential mapping reaches better results for all features than both unimodal language and vision as well as than one-step mapping. For real-data, it reaches better results than vision for the features colour and shape and outperform language for colour. For data from Blender, it outperforms vision for all features and is comparable to language for size and colour.

Dependence of the accuracy of the sequential mapping on the noise in the language data is visualized in the Figure 7.9 for data from iCub simulator in combination with language data processed by Sphinx 4. The noise to the language data is added subsequently and evenly to all classes (given proportion of language inputs was randomly changed to the random word). The noise was added artificially, but can be interpreted either as a noise in the data or mistakes in labeling perceived objects. I grouped them together into a misclassification variable. The visual data are let intact so the only cause of the observed variations in the accuracy is initialization. As can be seen, the accuracy of sequential mapping remains very stable even though the accuracy of language decreases and outperforms both language and vision for almost all values of the misclassification.

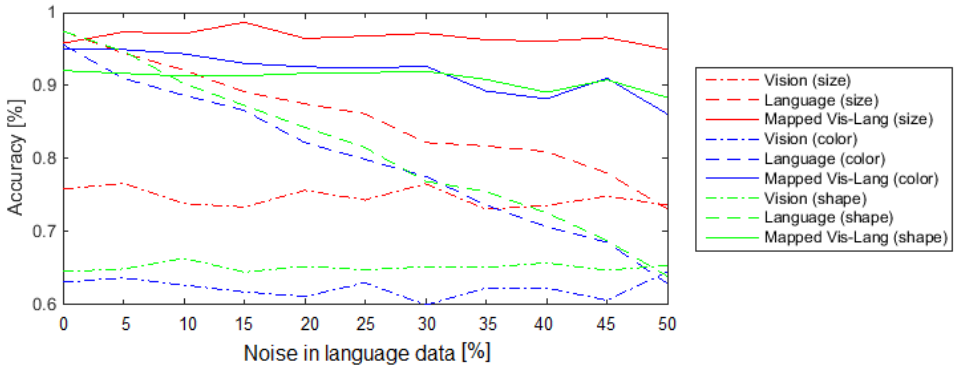


Figure 7.9: Dependence of mapping accuracy on the misclassification in the language data for fixed length sentence (mean values over 50 repetitions are visualized). Different colours correspond to different visual features (red – size, blue – colour, green – shape). Visual data are generated in Blender and acquired through iCub simulator, language data are processed using Sphinx 4.

The same comparison was done for artificially generated data. The dependence on the misclassification in language data is visualized in the Figure 7.10. Misclassification of language data is added equally to all classes.

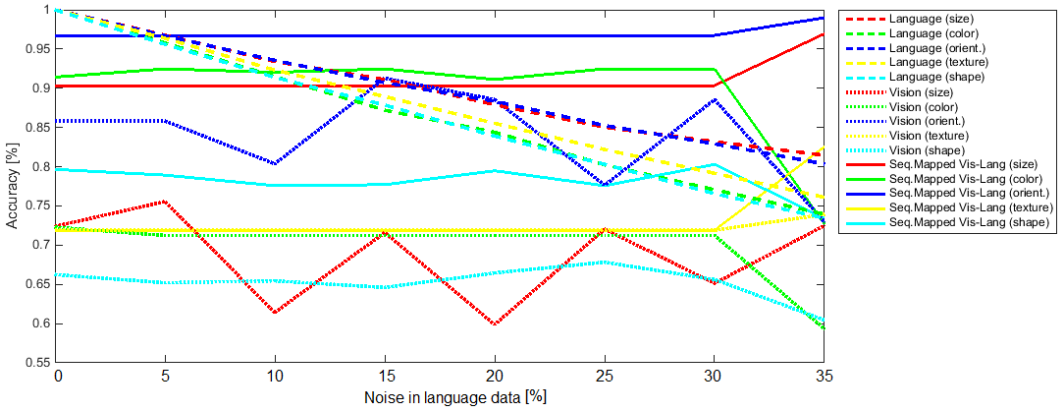


Figure 7.10: Dependence of mapping accuracy on the misclassification in the language data for fixed length sentence and artificially generated data (mean values over 10 repetitions are visualized). Different colours correspond to different visual features (red – size, blue – colour, green – shape). Visual data are generated artificially and language data are generated using a PatPho generator.

Mapping visual and language layer – variable length and grammar sentence

For the variable length/grammar sentence, only the artificially generated visual and language data were used. In this case, language data are clustered altogether and mapped to the visual features (those are clustered separately). The sequential mapping for a variable length sentence (see Algorithm 8) was used.

Accuracy [%]	Artificial data				
	Size	Colour	Orientation	Texture	Shape
Vision	75.8 ± 15.8	61.2 ± 6.0	78.9 ± 17.8	59.2 ± 15.1	74.4 ± 9.8
Language			31.02 ± 0.04		
One-step mapping	77.8 ± 19.2	33.8 ± 2.9	33.7 ± 0.0	76.2 ± 12.5	57.4 ± 4.6
Sequential mapping	78.4 ± 18.7	40.2 ± 7.6	76.9 ± 15.9	78.1 ± 14.7	70.3 ± 8.2

Table 7.7: Comparison of One-step mapping and Sequential mapping for artificial data and variable length sentence. The mean and standard deviation from 20 repetitions is shown.

Language [%] \ Vision [%]	Size	Colour	Orientation	Texture	Shape
	76 ± 16	61 ± 6	79 ± 18	59 ± 15	74 ± 10
100	100 ± 0	69 ± 7	89 ± 17	75 ± 18	89 ± 8
95	100 ± 0	76 ± 9	100 ± 0	56 ± 11	88 ± 7
90	100 ± 0	68 ± 7	89 ± 17	75 ± 18	91 ± 6
86	100 ± 0	68 ± 11	100 ± 0	70 ± 14	92 ± 7
82	100 ± 0	71 ± 8	96 ± 11	70 ± 14	88 ± 10
78	100 ± 0	66 ± 10	93 ± 15	75 ± 18	86 ± 9
75	100 ± 0	65 ± 6	100 ± 0	50 ± 15	95 ± 5
71	100 ± 0	69 ± 7	93 ± 15	53 ± 10	95 ± 7
68	100 ± 0	65 ± 11	100 ± 0	52 ± 14	97 ± 5
56	96 ± 11	69 ± 11	96 ± 11	53 ± 13	94 ± 7
46	89 ± 33	66 ± 26	78 ± 33	52 ± 17	82 ± 4
39	96 ± 11	58 ± 13	100 ± 0	53 ± 15	95 ± 5
33	89 ± 16	67 ± 9	100 ± 0	53 ± 10	90 ± 1
9	85 ± 17	61 ± 11	74 ± 15	52 ± 14	69 ± 11

Table 7.8: Comparison of accuracy of Sequential mapping for different levels of misclassification in language data – variable length sentence. Visual data are artificially generated data, language data are correct labels with artificially added equally distributed misclassification. The mean and standard deviation of accuracy from 10 repetitions is listed.

First, the ability to map together both vision and language was tested. The results can be seen in the Table 7.7.

Since the accuracy of the language clustering for artificially generated language data is very low when all words are clustered altogether, the found mapping between the two modalities is not very reliable and the effect of the different levels of language misclassification would be really hard to see. Furthermore, the noise added to the language data is not evenly distributed, which results in a fact that for some cases it decrease the accuracy compared to vision (initialization of clustering in visual layer using the found mapping is worse then the random initialization). Therefore I decided to use true labels and add a given amount of noise (misclassification) evenly to all classes and afterwards evaluate performance of the resulting sequential mapping. As can be seen from the Figure 7.11, the found mapping between the two modalities achieves higher accuracy compared to both subdomains and is very stable to misclassification in a language data.

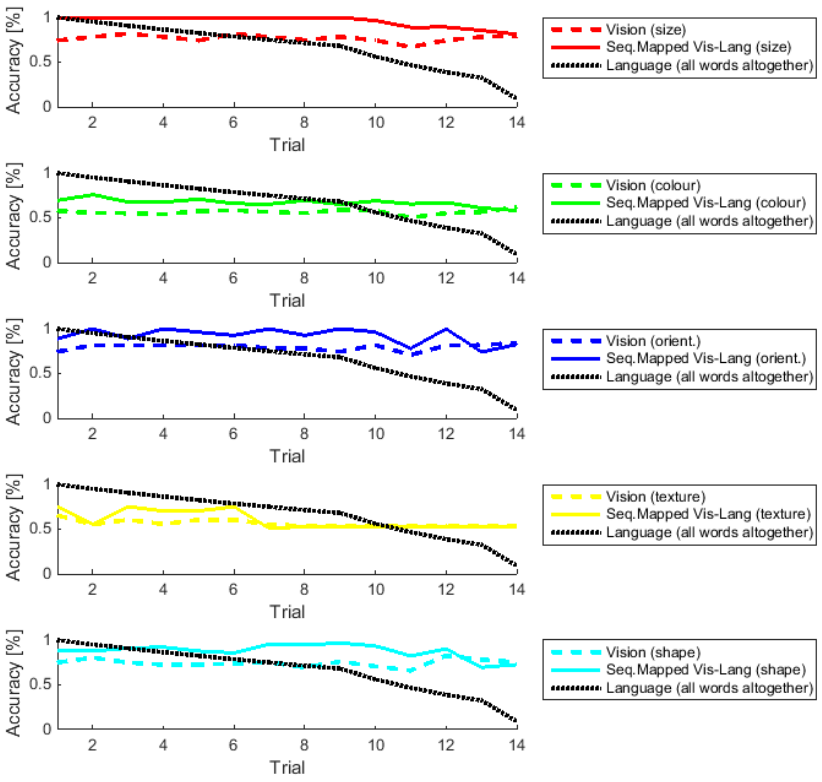


Figure 7.11: Dependence of mapping accuracy on the misclassification in the language data for variable length sentence (mean values over 10 repetitions are visualized). Different colours correspond to different visual features (red – size, green – colour, blue – orientation, yellow – texture, cyan – shape). Visual data are artificially generated data, language data are correct labels with artificially added equally distributed misclassification. As can be seen, with the decreasing accuracy of language, accuracy after mapping approaches the accuracy of the visual subdomain.

Conjunction of visual and language input – fixed sentence

After the assignment between clusterings from the unimodal visual and language layer is found, the resulting likelihoods are combined in a multimodal layer using a fuzzy conjunction. In following figures, I will mainly visualize the likelihood to individual clusters (and not the probability or an accuracy of final assignment), to enable a deeper insight into the whole process of putting together data from unimodal layers. Likelihoods are also used as input values for conjunction. As will be seen, likelihoods in a visual layer and likelihoods in language layer differ a lot. This is given by the fact that different models are used in a case of language (HMM) and vision (GMM), and that there is also different number of clusters in both subdomains. As can be seen, the assignment in language layer is much more fuzzy (an observed word has quite high likelihood to a lot of HMM models) while in the visual layer it is quite the contrary (mostly one model show high likelihood while others have very low likelihood). This is not only given by the fact

that there is used different metric for likelihood in a case of GMM than for HMM, but also because the visual data are much easier to cluster.

An example of one data point assignment is shown in Figure 7.12 where the likelihoods to individual clusters in the visual layer, language layer and multimodal layer after Lukasiewicz fuzzy conjunction are visualized. As can be seen, the resulting multimodal likelihood will assign the data point to the cluster $VL_{1,8}$ (corresponding to the visual cluster V_1 and language cluster L_8), which was selected neither by visual nor language layer. The visual layer would select the cluster V_{10} and language layer would select cluster L_3 (corresponding to the cluster V_3).

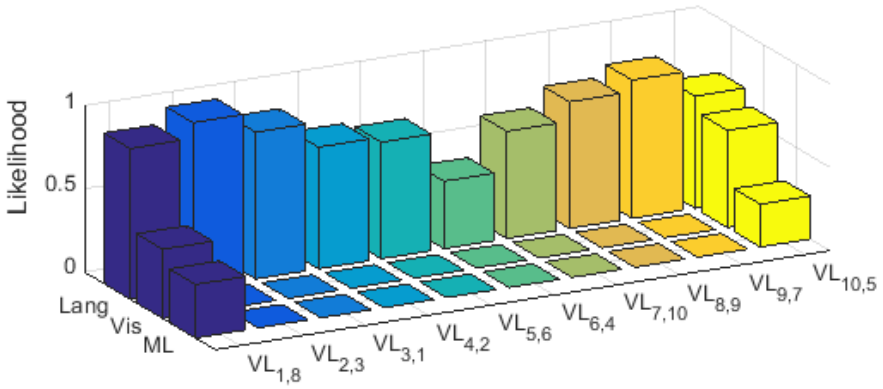


Figure 7.12: Combining likelihoods from visual and language layer in a multimodal layer by Lukasiewicz fuzzy conjunction for 1 datapoint (feature Shape) – likelihoods to individual clusters are shown (language clusters are reordered based on the assignment of clusters between visual and language layer described in previous Section 7.3). The likelihood for individual clusters for Vision (Vis), Language (Lang) and Multimodal layer (ML) is shown for individual clusters (multimodal likelihood $V_{1,8}$ corresponds to the visual cluster V_1 and language cluster L_8).

In Table 7.9, results for 3 types of fuzzy conjunction are listed (Product, Gödel and Lukasiewicz). The resulting clusterization accuracy in multimodal layer is compared to the clusterization accuracy obtained in visual and language layer.

In the computations above, emission probabilities for each possible emission are considered to be equal.

Accuracy [%]	*Product* $\hat{}$	*Gödel* $\hat{}$	*Lukasiewicz* $\hat{}$	Seq.mapping	Vis.	Lang.
Size	92 ± 2	93 ± 4	91 ± 3	91 ± 2	76 ± 10	73 ± 2
Colour	89 ± 3	89 ± 4	90 ± 2	88 ± 8	63 ± 8	72 ± 4
Orientation	97 ± 6	97 ± 6	97 ± 6	97 ± 6	77 ± 16	87 ± 7
Texture	76 ± 7	75 ± 5	76 ± 5	75 ± 6	74 ± 13	73 ± 2
Shape	80 ± 5	80 ± 6	81 ± 5	77 ± 5	76 ± 9	67 ± 2

Table 7.9: Results of multimodal layer – comparison of fuzzy conjunction: Product, Gödel and Lukasiewicz conjunction of visual and language clusterings is compared to the results from the first unimodal layers (averaged over 10 repetitions).

Conjunction of visual and language input – variable sentences

As well as for a fixed length sentence, the mapping between language and vision must be found when dealing with a variable length sentence. Afterwards, likelihoods from both unimodal layers can be combined using a given conjunction. The whole process will be shown on a simple example. Let us suppose that we have a sentence "*Small cross*" and corresponding image with a small purple diagonal lined cross.

After processing the visual layer, we get the corresponding likelihoods for each cluster in each visual feature. These are visualized in Figure 7.13 and reordered based on the found mapping to the language layer.

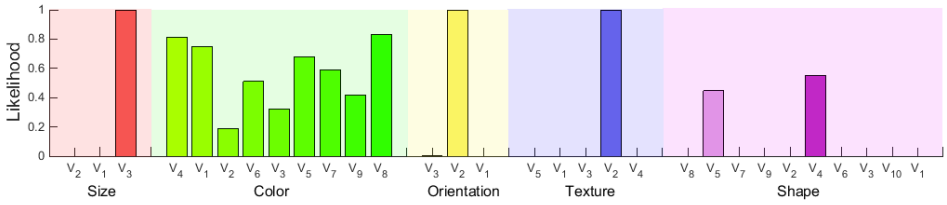


Figure 7.13: First visual layer – Likelihood to individual clusters for image "*Small purple horizontal lined cross*".

As well, likelihoods to the clusters in a language layer can be computed (in this case we suppose that words describing different visual features are clustered separatelly). These likelihoods to each language model are subsequently processed in a second language layer where they are multiplied by the probability of a given state in a sentence. This probability is found from an appropriate sentence model described by a given transition matrix. Lets suppose that the transition matrix shown in a Table 7.14 is used in this example.

	T_1	T_2	T_3	T_4	T_5	F
St	0.2	0.2	0.1	0.1	0.4	0
T_1	0	0.1	0.1	0.1	0.2	0.5
T_2	0	0	0.1	0.1	0.5	0.3
T_3	0	0	0	0.1	0.7	0.3
T_4	0	0	0	0	0.9	0.1
T_5	0	0	0	0	0	1

Figure 7.14: Transition matrices for variable length (St – Initial state, T_1 – Size, T_2 – Colour, T_3 – Orientation, T_4 – Texture, T_5 – Shape, K – Final state).

The **Problem 1** (described in Section 2.2) must be solved: Given a sequence of observations W and a model $HMM = (S, V, B, A, \Pi)$, determine the likelihoods $P(W|HMM)$ of the observed sequence W. To find out the most probable sequence of a given length (having a given observation sequence W), likelihoods $P(W|HMM)$ must be computed for all possible sequences of a given length and the most probable one is found.

Firstly suppose that we have one-word sentence. The incoming word is processed by the first language layer. Subsequently, the transition matrix for sentence processing (the second language layer) is taken into account and the output of the first language layer is multiplied by the transition probabilities. This means that likelihoods for each feature are multiplied by values in a first row of the transition matrix (for example likelihoods for feature Size are multiplied by 0.2 and similarily likelihoods for feature Orientation are multiplied by value 0.1). Subsequently, conjunction with the visual layer (Fig. 7.13) is performed. Output likelihoods from unimodal visual and language layer, language layer values after multiplication by transition matrix and results from the multimodal layer for sentence "*Small*" are shown in Fig. 7.15 and for sentence "*Cross*" in Fig. 7.16.

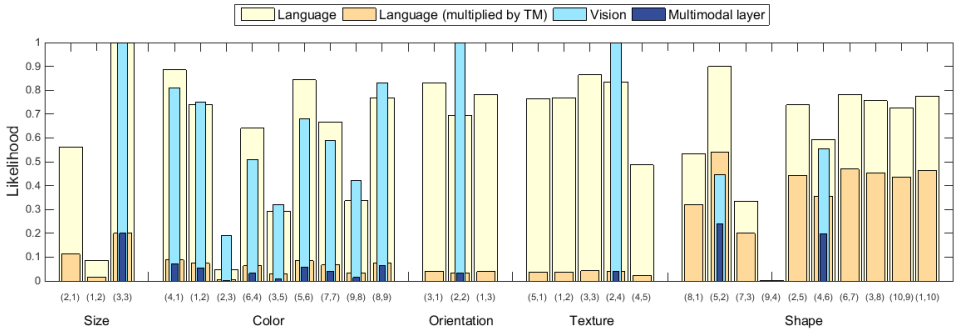


Figure 7.15: Likelihood to individual clusters in visual layer, likelihood to individual clusters in second language layer after multiplication by transition probabilities and the resulting values after Algebraic product fuzzy conjunction with visual layer for sentence "*Small.*" presented together with an image "*Small purple horizontal lined cross.*". X axis label (X,Y) (e.g.(1,3) means that the given column (likelihood) corresponds to the visual cluster V_1 and language cluster L_3 (to which the visual cluster is mapped) from a given feature (e.g. Shape).

As can be seen from Fig. 7.15 and Fig. 7.16, likelihoods of language layer for selected sentences "*Small.*" and "*Cross.*" are quite similar which will result – after multiplication with sentence transition matrix and conjunction with visual layer – to the final decision that both sentences are "*Cross.*". This is an incorrect decision for the first sentence and a correct decision for the second sentence "*Cross.*". The incorrect decision in the first case (sentence "*Small.*") is caused by the fact that based on our sentence model described by transition matrix we expect that one-word sentence. One-word sentence describing the feature Shape is two-times more probable than the sentence describing the feature Size.

Now have a look at the case when the sentence with more words is coming. For example we got two-word sentence "*Small cross.*" (see Fig. 7.17). Because outputs from the first language layer are only likelihoods that the observed word was generated by the given model (which should represent individual word), our aim is to use also visual information and the sentence model λ (sentence is modeled by HMM represented by transition matrix (see Table 7.14) and emission matrices. In this particular case, we have to compute for each object likelihood as follows: that the object has visual features i and

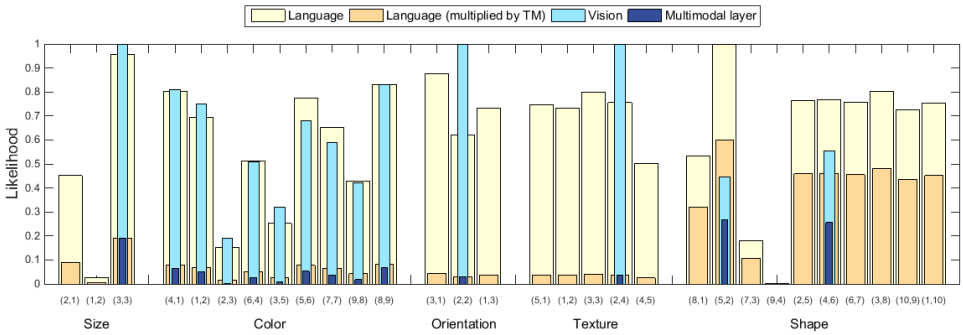


Figure 7.16: Likelihood to individual clusters in visual layer, likelihood to individual clusters in second language layer after multiplication by transition probabilities and the resulting values after Algebraic product fuzzy conjunction with visual layer for sentence "*Cross.*" presented together with an image "*Small purple horizontal lined cross.*". X axis label (X,Y) (e.g.(1,3) means that the given column (likelihood) corresponds to the visual cluster V_1 and language cluster L_3 (to which the visual cluster is mapped) from a given feature (e.g. Shape).

j $(i, j \in 1, ..., 30)$ described by sequence of words W_1W_2:

$$\begin{aligned} ML(i,j|O,W) =& [LL(O|K_i) \hat{F} (l(W_1|L_i) * P(W_iW_jW_F|\lambda))] * \\ & [LL(O|K_j) \hat{F} (l(W_2|L_j) * P(W_iW_jW_F|\lambda))], \end{aligned} \tag{7.1}$$

where K_i is the model of the type of feature i in the first visual layer, L_i is the HMM model of individual word i in the 1st language layer, $l(W_1|L_i)$ is the likelihood that the word W_i was generated by the model L_i, $P(W_iW_jW_F|\lambda)$ is a likelihood of observing sequence of words $W = W_iW_jW_F$ where W_F is the end of a sequence and $\hat{F}$ is the arbitrary fuzzy conjunction.

To compute the probability $P(O_iO_jO_F|\lambda)$, the forward algorithm is used [73].

The algorithm will find likelihood of every combination of observation sequences of the given length after evaluating all possible combinations of observation sequences O_iO_j (here the length of the sentence is 2 for the case when we do not consider Start and End of the sentence or 4 if we do).

The likelihood of the sentence *Start* O_i O_j *End* for each i and j is listed in Fig. 7.17.

As can be seen, the model will select correctly "*Small cross*" as the most probable sentence corresponding to the observed object. The second and the third most probable sentences would be *"Horizontal cross"* resp. "*Small hexagon*".

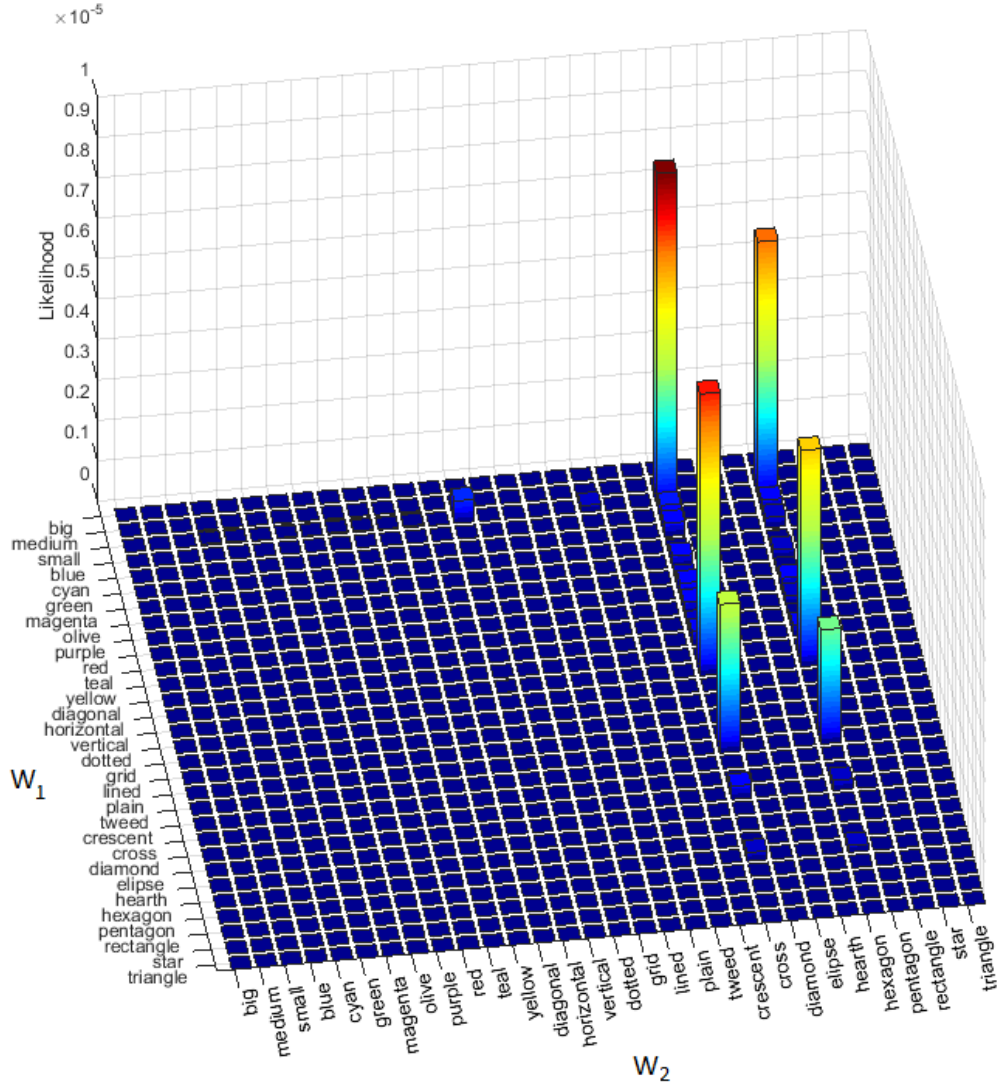


Figure 7.17: Likelihoods for two-word sentences. The likelihood of the sentence *Start* W_1 W_2 *End* for each posible word $W_1, W_2 \in \{\text{big}, ...,\text{rectangle}\}$ is visualized.

Chapter 8

Summary to the proposed architecture

The bioinspired unsupervised hierarchical multimodal architecture for parallel language and vision processing was proposed. Particularly, I have focused on how is the vision-to-language mapping performed with the focus on an unequal number of classes in both modalities. I also investigated how the mapping performance is affected by noise in the data. Results for artificially generated data (both visual and linguistic), data from iCub simulator and from real iCub robot with language analysed by Sphinx 4 were shown. Artificially generated data enabled creating a bigger dataset for testing purposes as well as higher variability of features or possibility to control different parameters of input data (such as added noise, movement, size of objects etc.). Real data acquired from physical iCub robot gave the more realistic distribution of noise in data as well as processing of real speech signals.

First, the ability of different algorithms to classify unimodal visual data was compared. As expected, for data which are well separated and mainly spherically distributed (this is generally a case for simulated and artifically generated data), k-means algorithm outperformed GMM algorithm. On the other hand for non-sperical real data performed generally better GMM algorithm (see Table 7.1 for artificially generated data and Table 7.2 for simulated data placed in an iCub simulator and data from real iCub robot cameras). Impact of added noise on classification accuracy is shown in Figure 7.1. Since the unsupervised algorithms are highly dependent on the initialization, it can be seen, that the standard deviation of data is quite high even though 20 repetitions were averaged. Furthermore, compositionality of algorithm was tested which means that the performance on previously unseen combination of features was evaluated. The modularity of the architecture ensured that the performance in this case is comparable to the case when the objects are seen during the training phase (see Table 7.3 and Fig. 7.4 for results with combination of two features and more features respectivelly).

In the speech recognition layer, Hidden Markov Models (HMM) were used to model separate words. Two clustering algorithms were compared for clustering these HMM models – namely k-means and agglomerative clustering. As can be seen from the Figure 7.6, agglomerative clustering outperformed k-means clustering in most of the cases. The most important parameter of For HMM is the number of hidden states. Therefore performance of the clusterization algorithm was analysed for different number of hidden states and based on these preliminary results (see Table 7.5) five hidden states were selected for further analysis. As can be seen, performance of clusterization is not very high compared to standadard speech recognition tools. This is given by the fact that in this case, to simulate real language acquisition, no prelearned vocabularies are used. Hidden Markov Models (HMM) require a large amount of training data to obtain reliable

probability estimates. For isolated word recognition (100 words or more), we cannot expect a user to speak each word more than several times. This problem can be partly overcome when we focus on learning individual phonemic groups while each state will incorporate the whole group of different phonems or by using online clusterization.

One of the main concerns of this thesis is how to find the mapping between the language and vision in a case of both fixed and variable length sentence and its resistance against the noise or misclassification in separate subdomains. In the thesis, two approaches are compared: one-step mapping to sequential mapping which in a stepwise manner finds the best mapped clusters while constantly relearning clusterization of visual data. The novel sequential mapping (see Algorithm 7 and Algorithm 8) led to an improvement of effectiveness compared to the method which maps vision to language directly in one step. This can be seen in finding more accurate mapping which leads to better estimation of the labels of the clustered data and consequently to the lower classification error for all of the evaluated datasets and features (see Table 7.5 for artificially generated data and Table 7.6 for data from iCub simulator and real iCub robot). The accuracy after mapping (initialization for clustering visual data is based on the found mapping to language layer) outperformed either vision or language and in some cases both of them. This is an important finding, since the sequential mapping does not improve only accuracy of visual clustering, but can as well fix mistakes in the language recognition, which provides the labels.

The ability to find the mapping in a case when we have to deal with the increasing volume of misclassificiation in a linguistic domain was tested for both fixed length sentence (see Figure 7.10 for artificially generated data and Figure 7.9 for data from iCub simulator) and for variable length sentence (see Figure 7.11). As can be seen, the ability to find the mapping is very resistant to the misclassification in language subdomain. It remains almost intact when the noise is equally distributed to all classes. For data from iCub simulator, the mapping accuracy decreases only very slightly and remains around 90% while the accuracy of language recognition drops from original approx. 95% to approx. 70% (depends on the specific visual feature). Furthermore, when the sequential mapping is used and the found mapping is used for the new initialization of visual data, the clustering achieves better results compared to both subdomains. Similar results as for iCub simulator were obtained for artificially generated data with fixed grammar.

Ability to find mapping for a variable length sentence was tested only on artificially generated data. Since the accuracy of the language clustering for artificially generated language data is very low and the misclassification of language data is not evenly distributed among classes, in some cases the accuracy after mapping is decreased compared to individual visual and language subdomains (initialization of clustering in visual layer using the found mapping is worse then the random initialization). Therefore the effect of different levels of language misclassification on sequential mapping would be hard to see. To see the effect of misclassification on performance of mapping, true labels were used and a given amount of noise (misclassification) was added evenly to all classes. As can be seen from the Figure 7.11, the found mapping between the two modalities achieves higher accuracy compared to both subdomains and is very stable to misclassification in a language data. With the decreasing accuracy of language, accuracy after mapping approaches the accuracy of the visual subdomain.

The found mapping between the language and vision is used for evaluating the most

probable class to which the observed data should be assigned to in the top most layer of the architecture. The information from individual subdomains is combined using the fuzzy conjunction. Different types of fuzzy conjunctions were compared (see Table 7.9 for results). As can be seen from the table, improvement after using fuzzy conjunction is very small and non significant compared to clusterization based on simple visual inputs initialized using information from sequential mapping. This does not necessarily mean that the better method for joining information in multimodal layer should be proposed. More probably, sequential mapping improves clusterization accuracy so much that there is only very small space for further improvement of accuracy. Anyway, these results will be basis for further investigations.

An example of likelihoods to individual clusters in individual layers (including multimodal layer after algebraic fuzzy conjunction) for sentence "*Small*" and "*Cross*" is shown in Figure 5.16 and Figure 5.16 respectively. The second sentence "*Cross*" is an example of the case where multimodal layer improves the decision of unimodal visual layer and correctly sentence "*Cross*" is correctly selected. An example of likelihoods for a two-word sentence is shown in Figure 7.17.

Part III

Thesis contribution and future research

Chapter 9

Thesis contribution

Most of the current work on language grounding assumes a fixed grammar and length sentence and uses at least partly supervised approach (e.g. image recognition or speech recognition is based on deep neural networks, which are pre-trained on large labeled databases; actions are pre-learned; number of clusters in data is hardcoded; etc.). Only few models investigate language compositionality and its grounding in multimodal perception [10]. The research on grounding variable length sentences is very restricted and deals only with static scenes [6]. The synergy of fully unsupervised approach, variable length sentence and multimodal grounding is virtually nonexistent.

The approach proposed here go beyond the current state of the art in several key aspects. First, the thesis focuses on creating fully unsupervised cognitive architecture. *Unsupervised approach* was used both for processing individual modalities (vision and language) as well as for their mapping in the multimodal layer. To avoid neccessity of prior knowledge, algorithm for finding optimal number of components in a mixture was proposed. The unsupervised approach is important in the situations where the adaptive and autonomous behaviour is required. Secondly, the *variable length sentence* containing visual properties was used as a language input. To find mapping between individual modalities, sequential mapping was proposed. The mapping is able to map also non-equal number of clusters from individual subdomains and assign words from variable length sentence to observed visual features. The comparison of proposed algorithms to other state-of-the-art algorithms was provided on both simulated and real-world data. The presented dissertation thesis shows how is it possible to combine information from different modalities (static visual data and time-varying language data) and give a meaning to observed visual scenes. Furthermore it shows, that it is possible to increase the overall accuracy by combining the information from individual subdomains.

9.1 A novel clustering algorithm – gmGMM

In the Chapter 3, a novel clustering algorithm was proposed [23]. Specifically, greedy Gausian mixture model with merging (gmGMM), which is an algorithm that is capable of finding the optimal number of components in the mixture without any prior information. The novel step in the greedy algorithm allows the improvement of performance when the stopping criterion is met, as the most dependent component is removed and the new component is initialised. The second novelty lies in the merging of all the dependent clusters in the final stage. The ability of gmGMM algorithm to find optimal number of components in data for different stopping criteria was evaluated and the performance of an algorithm was compared to other similar algorithms using four artificial and two

real-world datasets.

Many methods other than the gGMM have been proposed for the detection of the number of components in the data. Most of the algorithms nowadays [347, 348] stems from non-parametric Bayesian methods. For example, the Bayesian approach, based on the reversible Markov Chain Monte Carlo, was proposed in [349], but the method is computationally very demanding. Generally, parametric methods are more efficient than non-parametric ones when the number of estimated parameters is relatively low.

The gmGMM algorithm is suitable for tasks with an unknown number of components (e.g. image recognition, feature selection or speakers identification) and could be further improved by taking advantage of more sophisticated initialisation methods and incremental GMM methods. The proposed method can be also generalised for mixture models of an arbitrary probability distribution.

9.2 Proposed multimodal cognitive architecture

In the Chapter 5, hierarchical architecture of language acquisition was proposed [21]. The architecture aims to replicate organization of the similar processes in the brain. Therefore the language and visual information are processed separately, while mapping between these two modalities is found subsequently. This architecture is fully unsupervised, which enables autonomous behavior, adaptation to the gradually changing environment, as well as ability to learn things for which it wasn't trained before. Even though in many fields supervised teaching is possible at least to some extend, in many others, autonomous behavior and unsupervised behavior is a big advantage.

The main novelty of this thesis lies in the fact that I investigated how to *ground variable length sentence in a perception*. In my case, I used as a visual input static scene consisting of object with varying visual features. The variable length sentence and unsupervised approach makes the task of finding mapping between the modalities (in this case vision and language) much harder.

In cross-situational learning, the mapping between modalities is usually found by one-step mapping – by directly using frequencies of referent and meaning co-occurrences (the ones with the highest co-occurrence are mapped together) [350]. When the problem is extended to the more realistic case where meanings and referents are recognized with some uncertainty, or when there is non-equal number of classes of those clusterings, more advanced methods are needed. How to map in an unsupervised manner several clusterings (e.g. for vision, action, language) is not only important question in cognitive modeling, but also *in general machine learning, where data acquired from different sensors or in different situations can be independently clustered and mapped one to each other*. In this thesis, I tested newly proposed method called sequential mapping for mapping language to vision on both artificial and real data from humanoid robot iCub. It was shown, that the method is able to find mapping between language and vision, it improves the accuracy of both individual subdomains and shows very good resistance to noise in language. This is an important result which says that we are able not only to find mapping between more clusterings, but we can also *improve clusterization accuracy by combining individual classifiers*.

The mapping will find a reliable labeling for the visual input data (more generally for data from any other modality) with a possibility to also incorporate fuzziness of this

mapping. For some concepts finding an unambiguous mapping is very easy, for others it is much more difficult or impossible (such as the love has no dominating colour, but sky is usualy blue). Since the *mapping is established only among the clusters where it makes sense*, dealing with *a lot of redundant information is avoided*. Similar idea is used in classification algorithms which use sparse matrices.

The proposed architecture also treasures from the fact, that *processing more smaller problems paralelly is both computationally effective and reaches better performance than processing them altogether*. The lower computational demands can be easily seen e.g. for visual processing where the GMM is used. The estimated parameters for the model are means and covariance matrices. The number of estimated parameters is therefore quadratically dependent on the number of dimensions. Therefore when dividing the initial problem to N equally sized subproblems (e.g. language data and visual data processed separatelly), the number of estimated parameters will be $N * D/N + N * (D/N)^2 = D + D^2/N$ compared to $D + D^2$. Also all operations required for estimating these parameters will benefit from this fact (e.g. finding inverse of matrices).

Furthermore when we divide the input data so each subproblem has *lower number of clusters to be found*, it is higher probable that those will be found correctly (e.g. processing separatelly different visual features such as colour, texture, size, orientation or shape) because the *complexity of the problem decreased* and restrict the space of possible variations, so the initialization-dependent unsupervised clustering algorithms perform better. This is not always possible, for example for variable sentences and language data we do not know which data will belong to which cluster before the clusterization is done.

Chapter 10

Future research

The presented research can be extended from many aspects: whole architecture design and learning mechanisms, processing of unimodal information and experimental design and used datasets.

Processing of unimodal information

From the processing of unimodal information aspects, *state-of-the-art vision and speech algorithms* can be applied to improve performance of clusterization of individual modalities. Since this work focuses mainly on symbol grounding problem and finding the mapping between language and vision; processing of the individual modalities is not optimized.

Particularly, in vision, the methods used in the thesis are not able, in the presented design, to recognize partly ocluded objects and have problems with objects seen from different perspective. Even though methods for automatically finding the number of clusters were proposed, there is still a big place for improvements in this area. Automatic detection of number of clusters is very important and its combination with online learning should be used. The ability of clusterization to deal with unequal number of datapoints in clusters is another important parameter which should be taken into account in the future research.

In a speech recognition processing part, Hidden markov Models were used to model separate words. Hidden Markov Models are used also in the publicly accessible software for language processing such as Sphinx 4, which I tested in the experiment with iCub robot. I plan to use these software more extensively in a future research. Anyway, when dealing with unsupervised learning for learning larger number of words, the problems with clusterization occur. Therefore more sophisticated clusterization methods should be used (e.g. spectral clustering). Also step-wise learning of individual words during online learning should help to easier separate the input data. In future, I would like to focus on processing only real speech signals instead of using artificially generated langauge data, which were used in a major part of the thesis. As well, as for vision, automatic detection of the number of clusters (separate words) in the data should be implemented.

The sentence processing could be improved by gradual evolution of the form of transmission and emission matrices describing the sentence so they can adapt to the current situation. Starting from simpler to more complex sentences (this corresponds to teaching kids who are firstly taught by simple mainly one-word sentences which are gradually getting more and more complex). As well processing sentences with more complex grammar including verbs, adjectives and prepositions will be necessary. The last thing is to implement continual processing of language data instead of batch learning.

Architecture design and learning

From the architecture design aspect, the architecture should be extended so also information from *other modalities* can be incorporated (e.g. touch sensors or movement of the robot are combined with a language information). This will enable finding *mapping between language and motoric actions*. Further, *more modalities can be combined together* to further increase the accuracy of their individual recognition (e.g. we have in hand partly ocluded object having most probably red colour, we hear something like "box" or "sock" and when we touch it, it is soft and having a fabric texture, so we decide it is most probably "red sock").

When focusing on the used learning, I would like to refocus from a batch learning to more natural *online learning* or its combination. This should be combined with the fact that data from different modalities can be learned in different peace because not always corresponding information from both modalities is available.

Another mechanism, which I would like to enable in the future version of the architecture is *asking for the missing knowledge*. This means that we can ask for the name of an object which we do not know how to label ("What is this?") or the opposite way around we can ask to be shown an object/situation for which we only know the name ("Show me a cube"). This behavior could appear in a case when our knowledge is very fuzzy or when this information is totally missing.

Experimental design and used datasets

To test the ability of the architecture to generalize the acquired data, *more complex scenarios* should be used. This include learning *more different objects in visual domain* and their corresponding names in language domain or trying to detect more objects in the scene described by more complex language information provided together with a visual input.

When focusing on the scenarios with more objects in a scene, we can also *vary their position* or focus on the *interaction among these objects*. The first mentioned aspect will correspond to preposition in a sentence (on, under, at) or spatial information (left, right) and the interaction of objects can be described and mapped to the verbs provided in the sentence ("pull", "push", "tear apart", "jump", "walk" etc.). To be able to map these time-varying visual information together with language data, also tracking algorithms will have to be incorporated into the architecture.

Furthermore, the proposed architecture could *generalize learned words describing situations or objects* and create model of sufixes and prefixes of words. This means, that the robot can create new words for weird new objects and situations based on the similarity to previously observed data (e.g. after learning words "heat", "play","replay" he will be able to create a new word "reheat"). Then, he can start to consistently call the observed object by the newly learned word.

Bibliography

[1] W. V. Quine, "On the reasons for indeterminacy of translation," *The Journal of Philosophy*, vol. 67, no. 6, pp. 178–183, 1970.

[2] K. R. Coventry and S. C. Garrod, *Saying, seeing and acting: The psychological semantics of spatial prepositions*. Psychology Press, 2004.

[3] A. M. Glenberg and M. P. Kaschak, "Grounding language in action," *Psychonomic bulletin and review*, vol. 9.3, pp. 558–565, 2002.

[4] A. Cangelosi, A. Greco, and S. Harnad, "From robotic toil to symbolic theft: grounding transfer from entry-level to higher-level categories," *Connection Science*, vol. 12.2, pp. 143–162, 2000.

[5] M. Vavrečka and I. Farkaš, "A multimodal connectionist architecture for unsupervised grounding of spatial language," *Cognitive Computation*, vol. 6.1, pp. 101–112, 2014.

[6] D. K. Roy, "Learning visually grounded words and syntax for a scene description task," *Computer Speech and Language*, vol. 16.3, pp. 353–385, 2002.

[7] P. Vogt, "Language evolution and robotics: Issues on symbol grounding," *Artificial cognition systems*, vol. 176, 2006.

[8] V. Tikhanoff, A. Cangelosi, and G. Metta, "Integration of speech and action in humanoid robots: icub simulation experiments," *IEEE Transactions on Autonomous Mental Development*, vol. 3.1, pp. 17–29, 2011.

[9] F. Stramandinoli, D. Marocco, and A. Cangelosi, "The grounding of higher order concepts in action and language: a cognitive robotics model," *Neural Networks*, vol. 32, pp. 165–173, 2012.

[10] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *Adaptive Behavior*, vol. 13, no. 1, pp. 33–52, 2005.

[11] N. Mavridis, "A review of verbal and non-verbal human-robot interactive communication," *Elsevier Journal of Robotics and Autonomous Systems*, vol. 63, pp. 22–35, January 2015.

[12] M. Fleischman and D. Roy, "Grounded language modeling for automatic speech recognition of sports video," in *ACL*, 2008.

[13] P. Gorniak and D. Roy, "Speaking with your sidekick: Understanding situated speech in computer role playing games," in *AIIDE*, pp. 57–62, 2005.

[14] E. Goldberg, N. Driedger, and R. Kittredge, "Using natural-language processing to produce weather forecasts," *IEEE Expert*, vol. 9(2), pp. 45–53, 1994.

[15] J. Kennedy, P. Baxter, and T. Belpaeme, "Comparing robot embodiments in a guided discovery learning interaction with children," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 293–308, 2015.

[16] M. Taddeo and L. Floridi, "Solving the symbol grounding problem: a critical review of fifteen years of research," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 17.4, pp. 419–445, 2005.

[17] K. Procházková, S. Daniš, and P. Svoboda, "Specific heat study of PrNi4Si," *Acta Physica Polonica-Series A: General Physics*, vol. 113, no. 1, pp. 299–302, 2008.

[18] K. Štěpánová, M. Vavrečka, and L. Lhotská, "Changes in strategy during the mental rotation task and its correlation with EEG," *Clinical Neurophysiology*, vol. 123, no. 3, p. e10, 2012.

[19] P. Bukovský, K. Štěpánová, M. Vavrečka, and L. Lhotská, "Differences in eeg between gifted and average gifted adolescents: Mental rotation task," in *Computational Intelligence for Multimedia Understanding (IWCIM), 2015 International Workshop on*, pp. 1–5, IEEE, 2015.

[20] M. Hoffmann, K. Štěpánová, and M. Reinstein, "The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1790–1798, 2014.

[21] K. Štěpánová, M. Vavrečka, and L. Lhotská, "Hierarchický pravděpodobnostní model osvojování jazyka (in czech)," in *Kognice a umělý život XIV (KUZ XIV)*, 2014.

[22] K. Štěpánová, "iCub robot - language acquisition through visual grounding." `https://www.youtube.com/watch?v=L5i6CinZsUE`, May 2016.

[23] K. Štěpánová and M. Vavrečka, "Estimating number of components in gaussian mixture model using combination of greedy and merging algorithm," *Pattern Analysis and Applications*, pp. 1–12, 2016.

[24] "The american heritage® dictionary of the english language, fourth edition." `http://dictionary.reference.com/browse/scrotum`, Nov 2012.

[25] P. Thagard, "Cognitive science," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Mit Press, fall 2012 ed., 2012.

[26] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt & Company, June 1983.

[27] R. Sun, E. Merrill, and T. Peterson, "From implicit skills to explicit knowledge: A bottom-up model of skill learning," *Cognitive Science*, vol. 25, pp. 203–244, 1999.

[28] N. Chater and C. D. Manning, "Probabilistic models of language processing and acquisition," *Trends in Cognitive Sciences*, vol. 10, pp. 335–344, 2006.

[29] D. Marr and T. Poggio, *From Understanding Computation to Understanding: Neural Circuitry*. Artificial intelligence memo, Defense Technical Information Center, 1976.

[30] R. Sun, L. A. Coward, and M. J. Zenzen, "On levels of cognitive modeling," *Philosophical Psychology*, vol. 18, pp. 613–637, 2005.

[31] A. Newell, *Unified Theories of Cognition*. The William James Lectures, Harvard University Press, 1994.

[32] M. L. Minsky, *Semantic Information Processing*. The MIT Press, 1969.

[33] L. Perlovsky, *Neural Networks and Intellect: using model-based concepts*. New York, NY: Oxford University Press, 2001.

[34] R. Sun, "Desiderata for cognitive architectures," *Philosophical Psychology*, vol. 17, pp. 341–373, 2004.

[35] G. F. Ashby and S. Helie, "A tutorial on computational cognitive neuroscience: Modeling the neurodynamics of cognition," *Journal of Mathematical Psychology*, vol. 55, pp. 273–289, Aug. 2011.

[36] G. Gigerenzer, Z. Swijtink, T. Porter, L. Daston, J. Beatty, and L. Kruger, *The empire of chance*. Cambridge, UK: Cambridge University Press, 1989.

[37] M. Lee, "How cognitive modeling can benefit from hierarchical bayesian models," *Journal of Mathematical Psychology*, vol. 55, pp. 1–7, 2011.

[38] J. K. Kruschke, "Bayesian data analysis," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 1, pp. 658–676, 2010.

[39] N. Chater, J. B. Tenenbaum, and A. Yuille, "Probabilistic models of cognition: Conceptual foundations," *Trends in Cognitive Sciences*, vol. 10, no. 7, pp. 287 – 291, 2006.

[40] T. L. Griffiths, C. Kemp, and J. B. Tenenbaum, "Bayesian models of cognition," in *The Cambridge handbook of computational cognitive modeling* (R. Sun, ed.), Cambridge University Press, 2008.

[41] A. Sanborn, T. Griffiths, and S. R.M., "Uncovering mental representations with Markov Chain Monte Carlo," *Cognitive psychology*, vol. 60, pp. 63–106, 2010.

[42] M. Lee, "Three case studies in the bayesian analysis of cognitive models," *Psychonomic Bulletin & Review*, vol. 15, pp. 1–15, 2008.

[43] A. C. Courville, N. D. Daw, and D. S. Touretzky, "Bayesian theories of conditioning in a changing world," *Trends in Cognitive Sciences*, vol. 10, pp. 294–300, 2006.

[44] A. Yuille and D. Kersten, "Vision as Bayesian inference: analysis by synthesis?," *Trends in Cognitive Sciences*, vol. 10, pp. 301–308, 2006.

[45] K. P. Kording and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nautre*, vol. 427, pp. 244–247, 2004.

[46] K. P. Kording and D. M. Wolpert, "Bayesian decision theory in sensorimotor control," *Trends in Cognitive Sciences*, vol. 10 (7), pp. 319–326, 2006.

[47] M. Steyvers, T. L. Griffiths, and S. Dennis, "Probabilistic inference in human semantic memory," *Trends in Cognitive Sciences*, vol. 10, pp. 327–334, 2006.

[48] F. Xu and J. Tenenbaum, "Word learning as bayesian inference," *Psychological Review*, vol. 114, no. 2, pp. 245–272, 2007.

[49] C. L. Baker, J. B. Tenenbaum, and R. R. Saxe, "Goal inference as inverse planning," in *Proceedings of the Twenty-Ninth Annual Conference of the Cognitive Science Society*, 2007.

[50] N. Nilsson, *Learning Machines*. McGraw-Hill, New York, 1965.

[51] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New Yotk: John Willey & Sons, 2001.

[52] L. Perlovsky, *Emotional cognitive neural algorithms with engineering applications. Dynamic logic: From vague to crisp*. Springer, 2011.

[53] P. Winston, *Artificial Intelligence*. Addison-Wesley. Reading, MA, 2nd ed. ed., 1984.

[54] A. Segre, "Applications of machine learning," *IEEE Expert*, vol. 7, no. 3, pp. 31–34, 1992.

[55] R. Bellman, *Adaptive control processes: a guided tour*. Rand Corporation Research studies, Princeton University Press, 1961.

[56] A. Glennerster, "Computational theories of vision," *Current Biology*, vol. 12, no. 20, pp. R682–R685, 2002.

[57] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information a statistically optimal fashion," *Nature*, vol. 415, pp. 429–433, 2002.

[58] D. C. Knill and A. Pouget, "The bayesian brain: the role of uncertainty in neural coding and computation," *Trends in Neurosciences*, vol. 27 (12), pp. 712–719, 2004.

[59] D. C. Knill and W. Richards, *Perception as Bayesian inference*. Cambridge University Press, 1996.

[60] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes," *Nature Neuroscience*, vol. 9 (11), pp. 1432–1438, 2006.

[61] T. Bayes, "An essay towards solving a problem in the doctrine of chances," *Phil. Trans. of the Royal Soc. of London*, vol. 53, pp. 370–418, 1763.

[62] P. Laplace, *Théorie analytique des probabilités*. Paris: Courcier, 1812.

[63] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[64] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[65] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[66] G. Murray, "Contribution to discussion of paper by A. P. Dempster, N. M. Laird and D. B. Rubin," *J. Roy. Statist. Soc. Ser. B*, vol. 39, pp. 27–28, 1977.

[67] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM review*, vol. 26, no. 2, pp. 195–239, 1984.

[68] X.-L. Meng and D. B. Rubin, "On the global and componentwise rates of convergence of the EM algorithm," *Linear Algebra and its Applications*, vol. 199, pp. 413–425, 1994.

[69] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for gaussian mixtures," *Neural computation*, vol. 8, no. 1, pp. 129–151, 1996.

[70] J. Ma, L. Xu, and M. I. Jordan, "Asymptotic convergence rate of the EM algorithm for gaussian mixtures," *Neural Computation*, vol. 12, no. 12, pp. 2881–2907, 2000.

[71] K. Lange, "A quasi-newton acceleration of the EM algorithm," *Statistica sinica*, pp. 1–18, 1995.

[72] M. Jamshidian and R. I. Jennrich, "Acceleration of the EM algorithm by using quasi-newton methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, no. 3, pp. 569–587, 1997.

[73] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *PROCEEDINGS OF THE IEEE*, pp. 257–286, 1989.

[74] J. Subrahmonia, K. Nathan, and M. P. Perrone, "Writer dependent recognition of on-line unconstrained handwriting," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, ICASSP '96, Atlanta, Georgia, USA, May 7-10, 1996*, pp. 3478–3481, 1996.

[75] G. Rigoll, A. Kosmala, and S. Eickeler, "High performance real-time gesture recognition using hidden Markov models," *Gesture and Sign Language in Human-Computer Interaction*, pp. 69–80, 1998.

[76] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13.2, pp. 260–269, 1967.

[77] J. Fan, *On Markov and Hidden Markov Models with Applications to Trajectories*. PhD thesis, University of Pittsburgh, 2014.

[78] T. Jebara, Y. Song, and K. Thadani, "Spectral clustering and embedding with hidden Markov models," *Machine Learning: ECML 2007*, pp. 164–175, 2007.

[79] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *The Journal of Machine Learning Research*, vol. 5, pp. 819–844, 2004.

[80] P. Smyth, "Clustering sequences with hidden Markov models," in *Advances in Neural Information Processing Systems*, pp. 648–654, MIT Press, 1997.

[81] M. Butler, *Hidden Markov model clustering of acoustic data*. PhD thesis, University of Edinburgh, 2003.

[82] E. Coviello, A. B. Chan, and G. R. Lanckriet, "Clustering hidden Markov models with variational HEM," *The Journal of Machine Learning Research*, vol. 15.1, pp. 697–747, 2014.

[83] A. Panuccio, M. Bicego, and V. Murino, "A hidden markov model-based approach to sequential data clustering," *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 734–743, 2002.

[84] D. García-García, E. P. Hernández, and F. Díaz-de María, "A new distance measure for model-based sequence clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 1325–1331, 2008.

[85] M. Bicego, V. Murino, and A. Figueiredo, "Similarity-based clustering of sequences using hidden markov models," *Machine Learning and Data Mining in Pattern Recognition*, pp. 86–95, 2003.

[86] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. I–375–I–381, 2003.

[87] F. Porikli, "Clustering variable length sequences by eigenvector decomposition using hmm," *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 352–360, 2004.

[88] D. García García, *Similarity measures for clustering sequences and sets of data*. PhD thesis, University of Pittsburgh, 2011.

[89] R. M. Shiffrin, M. D. Lee, W. Kim, and E.-J. Wagenmakers, "A survey of model evaluation approaches with a tutorial on hierarchical bayesian methods.," *Cognitive Science*, vol. 32, no. 8, pp. 1248–1284, 2008.

[90] G. M. Allenby, P. E. Rossi, and R. E. McCulloch, "Hierarchical bayes models: a practitioners guide," tech. rep., Available at SSRN 655541, 2005.

[91] D. Green and J. Swets, *Signal detection theory and psychophysics*. Wiley, 1966.

[92] J. G. Snodgrass and J. Corwin, "Pragmatics of measuring recognition memory: Applications to dementia and amnesia," *Journal of Experimental Psychology: General*, vol. 17, pp. 34–50, 1988.

[93] R. M. Nosofsky, "Attention, similarity, and the identification-categorization relationship," *Journal of Experimental Psychology: General*, vol. 115, pp. 39–57, 1986.

[94] R. Ratcliff and G. McKoon, "The diffusion decision model: Theory and data for two-choice decision tasks.," *Neural Computation*, vol. 20, no. 4, pp. 873–922, 2008.

[95] L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *Journal of Mathematical Psychology*, vol. 55, no. 1, pp. 25–35, 2010.

[96] T. Lodewyckx, F. Tuerlinckx, P. Kuppens, N. Allen, and L. Sheeber, "A hierarchical state space approach to affective dynamics," *Journal of mathematical psychology*, vol. 55, pp. 68–83, 2011.

[97] J. Pooley, M. Lee, and W. Schankle, "Understanding memory impairment with memory models and hierarchical bayesian analysis," *Journal of mathematical psychology*, vol. 55, pp. 47–56, 2011.

[98] R. Ratcliff and F. Tuerlinckx, "Estimating parameters of the diffusion model: approaches to dealing with contaminant reaction times and parameter variability," *Psychonomic Bulletin & Review*, vol. 9, pp. 438–481, 2002.

[99] T. L. Griffiths, M. Steyvers, and J. Tenenbaum, "Topics in semantic representation," *Psychological Review*, vol. 114, no. 2, pp. 211–244, 2007.

[100] M. Bar, "A cortical mechanism for triggering top-down facilitation in visual object recognition," *Journal of cognitive neuroscience*, vol. 15(4), pp. 600–609, 2003.

[101] D. L. Schacter, I. G. Dobbins, and D. M. Schnyer, "Specificity of priming: a cognitive neuroscience perspective," *Nature Reviews Neuroscience*, vol. 5(11), pp. 853–862, 2004.

[102] D. L. Schacter and D. R. Addis, "The cognitive neuroscience of constructive memory: remembering the past and imagining the future," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362(1481), pp. 773–786, 2007.

[103] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Computer Journal*, vol. 41, pp. 578–588, 1998.

[104] L. I. Perlovsky, "Vague-to-crisp? neural mechanism of perception," *IEEE Trans. on neural networks*, vol. 20, pp. 1363–1367, 2009.

[105] L. I. Perlovsky, "Neural networks, fuzzy models and dynamic logic," *Aspects of Automatic Text Analysis*, vol. 209, pp. 363–386, 2007.

[106] L. I. Perlovsky and M. M. McManus, "Maximum likelihood neural networks for sensor fusion and adaptive classification," *Neural Netw.*, vol. 4, pp. 89–102, Jan. 1991.

[107] L. I. Perlovsky, "Neural network with fuzzy dynamic logic," in *Proc. of Int. Joint Conference on Neural Network*, pp. 3046–3051, 2005.

[108] R. Deming and L. I. Perlovsky, "Multi-target/multi-sensor tracking from optical data using modeling field theory." World Congress on Computational Intelligence (WCCI).

[109] A. Cangelosi, V. Tikhanoff, J. F. Fontanari, and E. Hourdakis, "Integrating language and cognition: A cognitive robotics approach," *Comp. Intell. Mag.*, vol. 2, pp. 65–70, Aug. 2007.

[110] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. on Pattern Anal. and Machine Intell*, vol. 24, pp. 381–396, 2000.

[111] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Split and merge EM algorithm for improving gaussian mixture density estimates," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, pp. 274–283, 1998.

[112] Y. Li and L. L., "A novel split and merge EM algorithm for gaussian mixture model," in *ICNC '09. Fifth International Conference on Natural Computation*, pp. 479–483, 2009.

[113] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient greedy learning of gaussian mixture models," *Neural Computation*, vol. 15, pp. 469–485, 2003.

[114] N. Vlassis and A. Likas, "A greedy EM algorithm for gaussian mixture learning," *Neural Processing Letters*, vol. 15, pp. 77–87, 2002.

[115] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, December 1974.

[116] G. E. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. 461—-464, 1978.

[117] G. Bouchard and G. Celeux, "Selection of generative models in classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28(4), pp. 544–554, 2006.

[118] G. Celeux and G. Soromenho, "An entropy criterion for assessing the number of clusters in a mixture models," *Journal of Classification*, vol. 13, pp. 195–212, 1994.

[119] H. Tenmoto, M. Kudo, and M. Shimbo, "MDL-based selection of the number of components in mixture models for pattern classification," *Adv. Pattern Recognit.*, vol. 1451, pp. 831–836, 1998.

[120] Y. Lee, K. Y. Lee, and J. Lee, "The estimating optimal number of gaussian mixtures based on incremental k-means for speaker identification," *International Journal of InformationTechnology*, vol. 12, 2006.

[121] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, pp. 451–461, Feb. 2003.

[122] J. Grim, J. Novovicova, P. Pudil, P. Somol, and F. Ferri, "Initialization normal mixutres of densities," *Applied Stat.*, vol. 36(3), pp. 318–324, 1987.

[123] P. Smyth, "Model selection for probabilistic clustering using crossvalidated likelihood," *Statistics and Computing*, vol. 9, p. 63, 2000.

[124] G. McLachlan, "On bootstraping the likelihood ratio test statistic for the number of components in a normal mixture," *Applied Stat.*, vol. 36, pp. 318–324, 1987.

[125] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1344–1348, 2005.

[126] D. Ververidis and C. Kotropoulos, "Gaussian mixture modeling by exploiting the mahalanobis distance," *Trans. Sig. Proc.*, vol. 56, pp. 2797–2811, July 2008.

[127] H. Bozdogan and S. Sclove, "Multi-sample cluster analysis using Akaike's information criterion," *Annals of the Institute of Statistical Mathemathics*, vol. 36, pp. 163–180, 1984.

[128] S. Roberts, D. Husmaier, I. Rezek, and W. Penny, "Bayesian approaches to gaussian mixture modelling," *IEEE Trans.Pattern Analysis and Machine Intell.*, vol. 20, no. 11, pp. 1133–1142, 1998.

[129] J. Banfield and A. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, vol. 49, pp. 803–821, 1992.

[130] H. Bozdogan, "Choosing the number of component clusters in the mixture model using a new informational complexity criterion of the inverse-fisher information matrix," *Information and Classification*, pp. 40–54, 1993.

[131] M. Windham and A. Cutler, "Information ratios for validating mixture analysis," *J. Am. Statistical Assoc.*, vol. 87, pp. 1188–1192, 1992.

[132] C. Biernacki and G. Celeux, "An improvement of the NEC criterion for assessing the number of clusters in a mixture model," *Patt. Recognition Lett.*, vol. 20, no. 3, pp. 267–272, 1999.

[133] J. J. Oliver, R. A. Baxter, and C. S. Wallace, "Unsupervised learning using MML," in *Machine learning: Proceedings of the thirteenth international conference (ICML 96)* (M. K. Publishers, ed.), pp. 364–372, 1996.

[134] J. Rissanen, *Stochastic Complexity in Statistical Inquiry Theory*. Series in Computer Science, World Scientific Publishing Company Incorporated, 1989.

[135] C. Biernacki and G. Govaert, "Using the classification likelihood to choose the number of clusters," *Computing Science and Statistics*, pp. 451–457, 1997.

[136] R. Yang, Z. and M. Zwolinski, "A mutual information theory for adaptive mixture models," *IEEE Trans. on Pattern Anal. and Machine Intell*, vol. 23, no. 4, pp. 1–8, 2001.

[137] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1154–1166, 2004.

[138] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems.," *Pattern Recognition*, vol. 39, no. 5, pp. 761–775, 2006.

[139] M. Lichman, "UCI machine learning repository," 2013.

[140] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural science*. McGraw Hill Professional, 2013.

[141] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, p. 1627– 1645, 2009.

[142] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1794–1801, 2009.

[143] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, pp. 119–130, 1988.

[144] B. W. Mel, "Seemore: combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition," *Neural computation*, pp. 777–804, 1997.

[145] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, pp. 1019–1025, 1999.

[146] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1019–1025, 2005.

[147] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 854–869, 2007.

[148] G. Wallis and E. Rolls, "A model of invariant object recognition in the visual system," *Prog. Neurobiol.*, p. 167–194, 1997.

[149] M. Riesenhuber and T. Poggio, "Models of object recognition," *Nature neuroscience*, p. 1199, 2000.

[150] M. Riesenhuber and T. Poggio, "Computational models of object recognition in cortex: A review," *Prog. Neurobiol.*, p. 167–194, 2000.

[151] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, pp. 91–110, 2004.

[152] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 433–449, 1999.

[153] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," *Advances in Neural Information Processing Systems*, pp. 244–252, 2010.

[154] C. Matuszek, N. FitzGerald, L. Zettlemoyer, L. Bo, and D. Fox, "A joint model of language and perception for grounded attribute learning," in *ICML*, 2012.

[155] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in *27th International Conference on Machine Learning (ICML-10)*, pp. 1215–1222, 2010.

[156] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *26th Annual International Conference on Machine Learning*, pp. 609–616, 2009.

[157] A. S. Ogale, C. Fermuller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 988–992, 2005.

[158] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[159] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[160] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1.4, pp. 541–551, 1989.

[161] A. Vrečko, D. Skočaj, N. Hawes, and A. Leonardis, "A computer vision integration model for a multi-modal cognitive system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pp. 3140–3147, 2009.

[162] D. Mumford, "Pattern theory: the mathematics of perception," in *in ICM*, Higher Education Press, 2002.

[163] A. Yuille and D. Kersten, "Vision as bayesian inference: analysis by synthesis?," *Trends in cognitive sciences*, vol. 10.7, pp. 301–308, 2006.

[164] D. Kersten and A. Yuille, "Bayesian models of object perception," *Current opinion in neurobiology*, vol. 13.2, pp. 150–158, 2003.

[165] F. Han and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of computer vision*, vol. 63.2, pp. 113–140, 2005.

[166] Y. Chen, L. Zhu, A. Yuille, and H. Zhang, "Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation, and recognition using knowledge propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31.10, pp. 1747–1761, 2009.

[167] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31.1, pp. 59–73, 2009.

[168] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester, "Object detection with grammar models," *Advances in Neural Information Processing Systems*, pp. 442–450, 2011.

[169] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19.7, pp. 696–710, 1997.

[170] P. W. Battaglia, J. Hamrick, and T. J.B., "Simulation as an engine of physical scene understanding," in *Proceedings of the National Academy of Sciences*, vol. 110 (45), pp. 18327–18332, 2013.

[171] A. Dekaban and D. Sadowsky, "Changes in brain weights during the span of human life: relation of brain weights to body heights and body weights," *Ann. Neurology*, vol. 4, pp. 345–356, 1978.

[172] P. K. Kuhl, "Early language acquisition: cracking the speech code," *Nature Reviews Neuroscience*, vol. 5, pp. 831–843, 2004.

[173] J. C. Lilly, "Critical brain size and language," *Perspectives in biology and medicine*, vol. 6.2, pp. 246–255, 1963.

[174] K. R. Gibson and A. C. Petersen, *Brain maturation and cognitive development: comparative and cross-cultural perspectives*. Transaction Publishers, 1991.

[175] E. Lenneberg, *Biological Foundations of Language*. Wiley, 1967.

[176] R. Joseph, "Origins of thought: Consciousness, language, egocentric speech and the multiplicity of mind," *Journal of Cosmology*, vol. 14, pp. 4577–4600, 2011.

[177] M. A. L. Ralph, "Neurocognitive insights on conceptual knowledge and its breakdown," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 369, no. 1634, p. 20120392, 2014.

[178] F. Pulvermüller, "Words in the brain's language," *Behavioral and brain sciences*, vol. 22, no. 02, pp. 253–279, 1999.

[179] J. R. Binder, R. H. Desai, W. W. Graves, and L. L. Conant, "Where is the semantic system? a critical review and meta-analysis of 120 functional neuroimaging studies," *Cerebral Cortex*, vol. 19, no. 12, pp. 2767–2796, 2009.

[180] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[181] N. Chater and M. H. Christiansen, "Computational models of psycholinguistics," in *The Cambridge Handbook of Psycholinguistics* (R. Sun, ed.), Cambridge University Press, 2008.

[182] K. I. Forster, "Accessing the mental lexicon," *New approaches to language mechanisms*, vol. 30, pp. 231–256, 1976.

[183] J. Morton, "Interaction of information in word recognition," *Psychological review*, vol. 76.2, p. 165, 1969.

[184] R. C. Berwick and A. S. Weinberg, *The grammatical basis of linguistic performance*. Cambridge, MA: MIT Press, 1984.

[185] M. W. Crocker and F. Keller, "Probabilistic grammars as models of gradience in language processing," *Gradience in grammar: Generative perspectives*, pp. 227–245, 2006.

[186] H. Kurtzman, *Studies in Syntactic Ambiguity Resolution*. Linguistics Club Bloomington, Ind: IU Linguistics Club, Indiana University Linguistics Club, 1985.

[187] V. H. Yngve, "A model and an hypothesis for language structure," in *Proceedings of the American philosophical society*, pp. 444–466, 1960.

[188] S. Vasishth and R. L. Lewis, "Symbolic models of human sentence processing," in *Encyclopedia of language and linguistics 5* (K. Brown, ed.), pp. 410–419, Cambridge University Press, 2006.

[189] R. M. Kaplan and J. Bresnan, "Lexical-functional grammar: A formal system for grammatical representation," *Formal Issues in Lexical-Functional Grammar*, pp. 29–130, 1982.

[190] G. Fanselow, M. Schlesewsky, D. Cavar, and R. Kliegl, "Optimal parsing: Syntactic parsing preferences and optimality theory," *Rutgers Optimality Archive*, p. 367, 1999.

[191] L. Frazier, *On comprehending sentences: Syntactic parsing strategies*. Indiana University Linguistics Club, 1979.

[192] E. Gibson, "The dependency locality theory: A distance-based theory of linguistic complexity," *Image, language, brain*, pp. 95–126, 2000.

[193] M. A. Just and P. A. Carpenter, *The psychology of reading and language comprehension*. Allyn and Bacon, 1987.

[194] J. A. Van Dyke and R. L. Lewis, "Distinguishing effects of structure and decay on attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities," *Journal of Memory and Language*, vol. 49.3, pp. 285–316, 2003.

[195] N. Chomsky, *Aspects of the Theory of Syntax*. MIT press, 1969.

[196] M. H. Christiansen and N. Chater, "Connectionist natural language processing: The state of the art," *Cognitive science*, vol. 23.4, pp. 417–437, 1999.

[197] D. L. Rohde and D. C. Plaut, "Higher-level cognitive functions and connectionist modeling. connectionist models of language processing.," *Cognitive studies*, vol. 10.1, pp. 10–28, 2003.

[198] J. L. McClelland and J. L. Elman, "The trace model of speech perception," *Cognitive psychology*, vol. 18.1, pp. 1–86, 1986.

[199] M. G. Gaskell, M. Hare, and W. D. Marslen-Wilson, "A connectionist model of phonological representation in speech perception," *Cognitive Science*, vol. 19.4, pp. 407–439, 1995.

[200] D. Rumelhart and J. McClelland, "On learning the past tenses of english verbs," in *Parallel distributed processin* (M. Rumelhart and the PDP Research Group, eds.), Cambridge, MA: MIT Press, 1986.

[201] S. Pinker and A. Prince, "On language and connectionism: Analysis of a parallel distributed processing model of language acquisition," *Cognition*, vol. 28.1, pp. 73–193, 1988.

[202] M. H. Christiansen, J. Allen, and M. S. Seidenberg, "Learning to segment speech using multiple cues: A connectionist model," *Language and cognitive processes*, vol. 13.2-3, pp. 221–268, 1998.

[203] M. Takáč, L. Benušková, and A. Knott, "Mapping sensorimotor sequences to word sequences: A connectionist model of language acquisition and sentence generation," *Cognition*, vol. 125.2, pp. 288–308, 2012.

[204] T. J. Sejnowski and C. R. Rosenberg, "Parallel networks that learn to pronounce english text," *Complex systems*, vol. 1.1, pp. 145–168, 1987.

[205] M. S. Seidenberg and J. L. McClelland, "A distributed, developmental model of word recognition and naming," *Psychological review*, vol. 96.4, p. 523, 1989.

[206] S. J. Hanson and J. Kegl, "Parsnip: A connectionist network that learns natural language grammar from exposure to natural language sentences," in *Ninth Annual Conference of the Cognitive Science Society*, pp. 106–119, 1987.

[207] A. Stolcke and J. Segal, "Precise n-gram probabilities from stochastic context-free grammars," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 74–79, 1994.

[208] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine learning*, vol. 7.2-3, pp. 195–225, 1991.

[209] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48.1, pp. 71–99, 1993.

[210] M. Zorzi, G. Houghton, and B. Butterworth, "Two routes or one in reading aloud? a connectionist dual-process model," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 24.4, p. 1131, 1998.

[211] J. Allen and M. S. Seidenberg, "The emergence of grammaticality in connectionist networks," *The emergence of language*, pp. 115–151, 1999.

[212] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychological review*, p. 1036, 2004.

[213] B. H. Juang and L. R. Rabiner, "Hidden markov models for speech recognition," *Technometrics*, vol. 33.3, pp. 251–272, 1991.

[214] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, ..., and E. Thayer, "The 1996 hub-4 sphinx-3 system," in *Proc. DARPA Speech recognition workshop*, pp. 85–89, 1997.

[215] T. Araki, T. Nakamura, T. Nagai, S. Nagasaka, T. Taniguchi, and N. Iwahashi, "Online learning of concepts and words using multimodal LDA and hierarchical Pitman-Yor language model," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1623–1630, IEEE, 2012.

[216] D. Norris, A. Cutler, J. M. McQueen, and S. Butterfield, "Phonological and conceptual activation in speech comprehension," *Cognitive Psychology*, vol. 53.2, pp. 146–193, 2006.

[217] G. E. Legge, T. S. Klitz, and B. S. Tjan, "Mr. chips: an ideal-observer model of reading," *Psychological review*, vol. 104.3, p. 524, 1997.

[218] J. H. Hulstijn, "Theoretical and empirical issues in the study of implicit and explicit second-language learning: Introduction," *Studies in second language acquisition*, vol. 27.02, pp. 129–140, 2005.

[219] D. C. Mitchell, F. Cuetos, M. M. Corley, and M. Brysbaert, "Exposure-based models of human parsing: Evidence for the use of coarse-grained (nonlexical) statistical records," *Journal of Psycholinguistic Research*, vol. 24.6, pp. 469–488, 1995.

[220] T. Desmet, C. De Baecke, D. Drieghe, M. Brysbaert, and W. Vonk, "Relative clause attachment in dutch: On-line comprehension corresponds to corpus frequencies when lexical variables are taken into account," *Language and Cognitive Processes*, vol. 21.4, pp. 453–485, 2006.

[221] E. Charniak, "Statistical parsing with a context-free grammar and word statistics," *AAAI/IAAI*, pp. 598–603, 1997.

[222] M. Collins, "Head-driven statistical models for natural language parsing," *Computational linguistics*, vol. 29.4, pp. 589–637, 2003.

[223] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[224] D. Jurafsky, "Probabilistic modeling in psycholinguistics: Linguistic comprehension and production," *Probabilistic linguistics*, vol. 21, pp. 251–272, 2003.

[225] C. Connine, F. Ferreira, C. Jones, C. Clifton, and L. Frazier, "Verb frame preferences: Descriptive norms," *Journal of Psycholinguistic Research*, vol. 13.4, pp. 307–319, 1984.

[226] R. Lado, *Linguistics Across Cultures: Applied Linguistics for Language Teachers.* . Ann Arbor: University of Michigan Press., 1957.

[227] B. F. Skinner, *Verbal Behavior.* Acton, MA: Copley Publishing Group, 1957.

[228] U. Weinreich, *Language in Contact: Findings and Problems.* New York, 1953.

[229] S. Pinker, *The language instinct: The new science of language and mind.* Penguin UK, 1994.

[230] J. Piaget, *The origins of intelligence in children*, vol. 8. New York: International Universities Press, 1952.

[231] A. Gopnik, "How babies think," *Scientific American*, vol. 303.1, pp. 76–81, 2010.

[232] J. Bruner, *Child's Talk: Learning to Use Language.* Oxford: Oxford University Press, 1983.

[233] L. S. Vygotsky, *Mind in society: The development of higher psychological processes.* Cambridge, Harvard University Press, 1978.

[234] E. Moerk, "Corrections in first language acquisition: Theoretical controversies and factual evidence," *International Journal of Psycholinguistics*, vol. 10, pp. 33–58, 1994.

[235] S. C. Hayes, D. Barnes-Holmes, and B. Roche, "Relational frame theory: A précis," in *Relational Frame Theory*, pp. 141–154, Springer US, 2001.

[236] F. Tonneau, "Relational frame theory: A post-skinnerian account of human language and cognition," *Advances in child development and behavior*, vol. 28, pp. 101–138, 2002.

[237] B. MacWhinney, "The competition model," *Mechanisms of language acquisition*, pp. 249–308, 1987.

[238] M. Tomasello, *Constructing a Language: A Usage-Based Theory of Language Acquisition.* Cambridge, MA: Harvard University Press, 2003.

[239] A. Prince and P. Smolensky, "Optimality theory: Constraint interaction in generative grammar," tech. rep., Rutgers University and University of Colorado at Boulder, 1993.

[240] D. Archangeli, "Optimality theory," in *Encyclopedia of Cognitive Science*, Springer London, 1997.

[241] P. K. Kuhl, "Human adults and human infants show a 'perceptual magnet effect' for the prototypes of speech categories, monkeys do not," *Perception and psychophysics*, vol. 50.2, pp. 93–107, 1991.

[242] P. K. Kuhl, "A new view of language acquisition," in *Proceedings of the National Academy of Sciences*, pp. 11850–11857, 2000.

[243] A. S. Hsu and N. Chater, "Probabilistic language acquisition: Theoretical, computational, and experimental analysis," *Cognition*, 2010.

[244] A. S. Hsu, N. Chater, and P. M. Vitányi, "The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis," *Cognition*, vol. 120.3, pp. 380–390, 2011.

[245] M. Dowman, "Addressing the learneability of verb subcategorizations with bayesian inference," in *Proceedings of the 22nd annual conference of the Cognitive Science Society*, 2000.

[246] M. Dowman, "Using minimum description length to make grammatical generalizations." Talk given at Machine Learning and Cognitive Science of Language Acquisition.

[247] S. Foraker, T. Regier, N. Khetarpal, A. Perfors, and J. Tenenbaum, "Indirect evidence and the poverty of the stimulus: The case of anaphoric one," *Cognitive Science*, vol. 33.2, pp. 287–300, 2009.

[248] P. Grünwald, "A minimum description length approach to grammar inference," in *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, Springer Lecture Notes in Artificial Intelligence*, p. 203–216, Springer London, 1997.

[249] A. Perfors, J. Tenenbaum, and T. Regier, "Poverty of the stimulus? a rational approach," in *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, p. 663–668, 2006.

[250] T. Regier and S. Gahl, "Learning the unlearnable: The role of missing evidence," *Cognition*, vol. 93.2, pp. 147–155, 2004.

[251] J. J. Horning, "A procedure for grammatical inference," in *IFIP Congress*, pp. 519–523, 1971.

[252] N. Chater and P. Vitányi, "Simplicity: A unifying principle in cognitive science?," *Trends in cognitive sciences*, vol. 7.1, pp. 19–22, 2003.

[253] D. Klein and C. D. Manning, "Corpus-based induction of syntactic structure: Models of dependency and constituency," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 478, 2004.

[254] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42.1, pp. 335–346, 1990.

[255] A. Cangelosi, A. Greco, and S. Harnad, "Symbol grounding and the symbolic theft hypothesis," in *Simulating the evolution of language. Springer London* (K. Brown, ed.), pp. 191–210, Springer London, 2002.

[256] P. Vogt, "The physical symbol grounding problem," *Cognitive Systems Research*, vol. 3.3, pp. 429–457, 2002.

[257] M. J. Mayo, "Symbol grounding and its implications for artificial intelligence," in *Proceedings of the 26th Australasian computer science conference-Volume 16*, pp. 55–60, Australian Computer Society, Inc., 2003.

[258] R. Sun, "Symbol grounding: a new look at an old idea," *Philosophical Psychology*, vol. 13.2, pp. 149–172, 2000.

[259] P. Davidsson, "Toward a general solution to the symbol grounding problem: Combining machine learning and computer vision," in *AAAI Fall Symposium Series, Machine Learning in Computer Vision: What, Why and How*, pp. 157–161, AAAI Press, 1993.

[260] M. T. Rosenstein and P. R. Cohen, "Symbol grounding with delay coordinates," in *Working Notes of the AAAI Workshop on The Grounding of Word Meaning*, 1998.

[261] L. Steels and P. Vogt, "Grounding adaptive language games in robotic agents," in *Proceedings of the fourth european conference on artificial life*, 1997.

[262] A. Billard and K. Dautenhahn, "Experiments in learning by imitation-grounding and use of communication in robotic agents," *Adaptive Behavior*, vol. 7.3-4, pp. 415–438, 1999.

[263] P. Varshavskaya, "Behavior-based early language development on a humanoid robot," in *Proceedings of the 2nd International Conference on Epigenetics Robotics (Edinburgh, Scotland)*, p. 149–158, 2002.

[264] S. Coradeschi, A. Loutfi, and B. Wrede, "A short review of symbol grounding in robotic and intelligent systems," *KI-Künstliche Intelligenz*, vol. 27, no. 2, pp. 129–136, 2013.

[265] D. Roy, "Grounding words in perception and action: computational insights," *Trends in cognitive sciences*, vol. 9.8, pp. 389–396, 2005.

[266] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, ..., and A. Zeschel, "Integration of action and language knowledge: A roadmap for developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 2.3, pp. 167–195, 2010.

[267] C. E. Snow, "Mothers' speech research: From input to interaction," in *Talking to children: Language input and acquisition*, pp. 31–49, Cambridge University Press, 1977.

[268] R. Tincoff and P. W. Jusczyk, "Some beginnings of word comprehension in 6-month-olds," *Psychological Science*, vol. 10(2), pp. 172–175, 1999.

[269] L. Fenson, P. S. Dale, J. S. Reznick, E. Bates, D. J. Thal, S. J. Pethick, ..., and J. Stiles, "Variability in early communicative development," in *Monographs of the society for research in child development 59*, pp. i–185, Wiley, 1994.

[270] D. Swingley and R. N. Aslin, "Spoken word recognition and lexical representation in very young children," *Cognition*, vol. 76, pp. 147–166, 2000.

[271] A. Woodward and K. Hoyne, "Infants' learning about words and sounds in relation to objects," *Child development*, vol. 70(1), pp. 65–77, 1999.

[272] S. M. Pruden, K. Hirsh-Pasek, R. M. Golinkoff, and E. A. Hennon, "The birth of words: Ten-month-olds learn words through perceptual salience," *Child development*, vol. 77.2, pp. 266–280, 2006.

[273] P. Bloom, *How children learn the meanings of words: Learning, development and conceptual change*. Cambridge, MA: MIT Press, 2000.

[274] L. L. Namy and S. R. Waxman, "Words and gestures: Infants' interpretations of different forms of symbolic reference," *Child Development*, vol. 69(2), pp. 295–308, 1998.

[275] L. L. Namy and S. R. Waxman, "Children's noun learning: How general learning processes make specialized learning mechanisms," in *The emergence of language*, pp. 227–305, Taylor and Francis, 1999.

[276] D. A. Baldwin, "Infants' contribution to the achievement of joint reference," *Child Development*, vol. 62(5), pp. 875–890, 1991.

[277] D. A. Baldwin, "Infants' ability to consult the speaker for clues to word reference," *Journal of Child Language*, vol. 20(2), pp. 395–418, 1993.

[278] D. A. Baldwin, E. M. Markman, B. Bill, R. N. Desjardins, J. M. Irwin, and G. Tidball, "Infants' reliance on a social criterion for establishing wordobject relations," *Child Development*, vol. 67.6, pp. 3135–3153, 1996.

[279] M. Tomasello, "Do young children have adult syntactic competence?," *Cognition*, vol. 74.3, pp. 209–253, 2000.

[280] M. Tomasello, "The item-based nature of children's early syntactic development," *Trends in cognitive sciences*, vol. 4.4, pp. 156–163, 2000.

[281] S. Pinker, *Learnability and cognition*. Cambridge, MA: MIT Press, 1989.

[282] J. M. Siskind, "A computational study of cross-situational techniques for learning word-to-meaning mappings," *Cognition*, vol. 61.1, pp. 39–91, 1996.

[283] L. Smith and C. Yu, "Infants rapidly learn word-referent mappings via cross-situational statistics," *Cognition*, vol. 106(3), pp. 1558–1568, 2000.

[284] J. F. Fontanari, V. Tikhanoff, A. Cangelosi, R. Ilin, and L. I. Perlovsky, "Cross-situational learning of object–word mapping using neural modeling fields," *Neural Networks*, vol. 22(5), pp. 579–585, 2009.

[285] B. Roy, N. Mavridis, S. Tellex, A. Salata, J. Guinness, M. Levit, and P. Gorniak, "The human speechome project," in *Proceedings of the 28th Annual Cognitive Science Conference*, pp. 192–196, 2006.

[286] D. Roy, "New horizons in the study of child language acquisition," in *INTERSPEECH-2009, Brighton, United Kingdom*, 2009.

[287] L. W. Barsalou, "Perceptions of perceptual symbols," *Behavioral and brain sciences*, vol. 22.04, pp. 637–660, 1999.

[288] D. Roy, "Grounded speech communication," in *Proceedings of the International Conference on Spoken Language Processing*, 2000.

[289] D. Roy, "Learning visually grounded words and syntax of natural spoken language," *Evolution of communication*, vol. 4.1, pp. 33–56, 2000.

[290] D. Roy, "Grounded spoken language acquisition: Experiments in word learning," *IEEE Transactions on Multimedia*, vol. 5.2, pp. 197–209, 2003.

[291] D. Roy, "Grounding words in perception and action: computational insights," *Trends in cognitive sciences*, vol. 9.8, pp. 389–396, 2005.

[292] A. Cangelosi, T. Riga, B. Giolito, and D. Marocco, "Language emergence and grounding in sensorimotor agents and robots," in *First International Workshop on Emergence and Evolution of Linguistic Communication*, pp. 3–8, 2004.

[293] A. Cangelosi, "Grounding language in action and perception: From cognitive agents to humanoid robots," *Physics of life reviews*, vol. 7.2, pp. 139–151, 2010.

[294] N. Mavridis and D. Roy, "Grounded situation models for robots: Where words and percepts meet," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[295] Y. Yang, A. Guha, C. Fermuller, and Y. Aloimonos, "A cognitive system for understanding human manipulation actions," *Advances in Cognitive Sysytems*, vol. 3, pp. 67–86, 2014.

[296] A. S. Ogale, C. Fermüller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 988–992, 2005.

[297] D. Joyce, L. Richards, A. Cangelosi, and K. Coventry, "On the foundations of perceptual symbol systems: Specifying embodied representations via connectionism," in *Proceedings of the 5th Intl. Conference on Cognitive Modeling*, 2003.

[298] M. Dyer, "Grounding language in perception," in *ARtifficial Intelligence and Neural Networks: Steps Toward Principled Integration*, San Diego, CA: Academic Press, 1994.

[299] J. B. Tenenbaum and F. Xu, "Word learning as bayesian inference," in *Proceedings of the 22nd annual conference of the cognitive science society*, 2000.

[300] T. L. Griffiths, N. Chater, C. Kemp, A. Perfors, and J. B. Tenenbaum, "Probabilistic models of cognition: Exploring representations and inductive biases," *Psychological Review*, vol. 14(8), pp. 357–364, 2010.

[301] L. Steels, "Evolving grounded communication for robots," *Trends in cognitive sciences*, vol. 7.7, pp. 308–312, 2003.

[302] D. K. Roy and A. P. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive science*, vol. 26.1, pp. 113–146, 2002.

[303] A. Cangelosi and D. Parisi, *Simulating the evolution of language Vol.1*. London: Springer, 2002.

[304] A. Cangelosi and et al., "The italk project: Integration and transfer of action and language knowledge in robots," in *Proceedings of Third ACM/IEEE International Conference on Human Robot Interaction*, vol. 12, 2008.

[305] K. Pastra, "The poeticon and poeticon++ projects," in *International Conference on Cognitive Systems*, 2012.

[306] V. Tikhanoff, J. F. Fontanari, A. Cangelosi, and L. I. Perlovsky, "Language and cognition integration through modeling field theory: category formation for symbol grounding," in *Artificial Neural Networks–ICANN 2006*, pp. 376–385, Springer Berlin Heidelberg, 2006.

[307] C. Crangle and P. Suppes, "Language and learning for robots," research report, Center for the Study of Language and Information, 1994.

[308] P. McGuire, J. Fritsch, J. J. Steil, F. Röthling, G. A. Fink, S. Wachsmuth, ..., and H. Ritter, "Multimodal humanmachine communication for instructing robot grasping tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1082–1088, 2002.

[309] D. Sofge, D. Perzanowski, M. Bugajska, W. Adams, and A. Schultz, "An agent-driven human-centric interface for autonomous mobile robots," in *Proceedings SCI*, pp. 1082–1088, 2003.

[310] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, ..., and D. Mulanda, "Humanoid robots as cooperative partners for people," *Int. Journal of Humanoid Robots*, vol. 1.2, p. 134, 2004.

[311] C. Breazeal, *Designing Sociable Robots*. The MIT Press, 2004.

[312] D. Bailey, N. Chang, J. Feldman, and S. Narayanan, "Extending embodied lexical development," in *Proceedings of the Twentieth Conference of the Cognitive Science Society)*, pp. 84–89, 1998.

[313] A. S. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," in *Dynamical vision*, pp. 115–126, Springer Berlin Heidelberg, 2007.

[314] J. M. Siskind, "Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic," *J. Artif. Intell. Res.(JAIR)*, vol. 15, pp. 31–90, 2001.

[315] L. Fernandino, J. R. Binder, R. H. Desai, S. L. Pendl, C. J. Humphries, W. L. Gross, L. L. Conant, and M. S. Seidenberg, "Concept representation reflects multimodal abstraction: A framework for embodied semantics," *Cerebral Cortex*, p. bhv020, 2015.

[316] D. K. Roy and A. P. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive science*, vol. 26, no. 1, pp. 113–146, 2002.

[317] W. Yi and D. Ballard, "Recognizing behavior in hand-eye coordination patterns," *International Journal of Humanoid Robotics*, vol. 6, p. 337–359, 2009.

[318] J. B. Tenenbaum, T. L. Griffiths, and C. Kemp, "Theory-based bayesian models of inductive learning and reasoning," *Trends in cognitive sciences*, vol. 10.7, pp. 309–318, 2006.

[319] M. C. Frank, N. D. Goodman, and J. B. Tenenbaum, "Using speakers' referential intentions to model early cross-situational word learning," *Psychological Science*, vol. 20(5), pp. 578–585, 2009.

[320] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, p. 859–865, 2011.

[321] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *AAAI*, 2013.

[322] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1729–1736, 2011.

[323] L. S. Zettlemoyer and M. Collins, "Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, p. 658–666, 2005.

[324] T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, and M. Steedman, "Lexical generalization in ccg grammar induction for semantic parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1512–1523, 2011.

[325] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, ..., and T. Berg, "Babytalk: Understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35.12, pp. 2891–2903, 2013.

[326] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, "Collective generation of natural image descriptions," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.*, pp. 359–368, 2012.

[327] K. Barnard, P. Duygulu, N. d. Freitas, D. Forsyth, D. Blei, and M. Jordan, "Matching words and pictures," *JMLR*, p. 1107–1135, 2003.

[328] H. Yu and J. M. Siskind, "Grounded language learning from video described with sentences," in *ACL*, 2013.

[329] H. Reckman, J. Orkin, and D. K. Roy, "Learning meanings of words and constructions, grounded in a virtual game.," in *KONVENS*, pp. 67–75, 2010.

[330] P. Gorniak and D. Roy, "Grounded semantic composition for visual scenes," *Journal of Artificial Intelligence Research*, vol. 21, pp. 429–470, 2004.

[331] M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," *Trends in neurosciences*, pp. 414–417, 1983.

[332] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[333] A. F. Canales, D. M. Gomez, and C. R. Maffet, "A critical assessment of the consciousness by synchrony hypothesis," *Biological research*, vol. 40, no. 4, pp. 517–519, 2007.

[334] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[335] B.-H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T technical journal*, vol. 64, no. 2, pp. 391–408, 1985.

[336] C. Büchel, C. Price, and K. Friston, "A multimodal language region in the ventral visual pathway," *Nature*, vol. 394, no. 6690, pp. 274–277, 1998.

[337] G. Spitsyna, J. E. Warren, S. K. Scott, F. E. Turkheimer, and R. J. Wise, "Converging language streams in the human temporal lobe," *The Journal of neuroscience*, vol. 26, no. 28, pp. 7328–7336, 2006.

[338] P. Li and B. MacWhinney, "Patpho: A phonological pattern generator for neural networks," *Behavior Research Methods, Instruments and Computers*, vol. 34.3, pp. 408–415, 2002.

[339] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator," in *Proceedings of 8th workshop on performance metrics for intelligent systems, ACM*, pp. 57–61, 2008.

[340] G. Metta and et al., "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of 8th workshop on performance metrics for intelligent systems, ACM*, pp. 50–56, 2008.

[341] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–38, 2006.

[342] P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf, "The cmu sphinx-4 speech recognition system," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, vol. 1, pp. 2–5, Citeseer, 2003.

[343] J. F. Werker and P. J. McLeod, "Infant preference for both male and female infant-directed talk: a developmental study of attentional and affective responsiveness.," *Canadian Journal of Psychology/Revue canadienne de psychologie*, vol. 43, no. 2, p. 230, 1989.

[344] M. R. Brent and J. M. Siskind, "The role of exposure to isolated words in early vocabulary development," *Cognition*, vol. 81, no. 2, pp. B33–B44, 2001.

[345] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Networks*, vol. 15, no. 8, pp. 1041–1058, 2002.

[346] F. Belmonte Klein, "GWR and GNG Classifier - File Exchange - MATLAB Central." `http://uk.mathworks.com/matlabcentral/fileexchange/57798-gwr-and-gng-classifier`, 2016.

[347] C. E. Rasmussen, "The infinite gaussian mixture model," in *NIPS*, vol. 12, pp. 554–560, 1999.

[348] S. J. Gershman and D. M. Blei, "A tutorial on bayesian nonparametric models," *Journal of Mathematical Psychology*, vol. 56(1), pp. 1–12, 2012.

[349] S. Richardson and P. J. Green, "On bayesian analysis of mixtures with an unknown number of components," *J.R.Statist.Soc. B*, pp. 731–792, 1997.

[350] K. Smith, A. D. Smith, R. A. Blythe, and P. Vogt, "Cross-situational learning: a mathematical approach," *Lecture Notes in Computer Science*, vol. 4211, pp. 31–44, 2006.

Appendix

Appendix A

List of Author's Publications

PUBLICATIONS OF THE AUTHOR RELEVANT TO THE THESIS

Publications in journals with impact factor

1. **K. Štěpánová (80 % contribution)** and M. Vavrečka, "Estimating number of components in gaussian mixture model using combination of greedy and merging algorithm," *Pattern Analysis and Applications*, pp. 1–12, 2016.
Impact factor (2015): 1.104

2. M. Hoffmann, **K. Štěpánová (equal contribution)**, and M. Reinstein, "The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1790–1798, 2014.
Impact factor (2015): 1.618
Nmb of citations: 5

Other publications

3. **K. Štěpánová (70 % contribution)**, M. Vavrečka, L. Lhotská, "Hierarchical probabilistic model of language acquisition (in Czech)," in *Kognice a umělý život XIV* (KUZ XIV), 2014.

4. **K. Štěpánová (70% contribution)**, M. Vavrečka, L. Lhotská, "Initialization methods for neural modeling fields that yield to finding the optimal number of clusters (in Czech)", In: *Kognice a umělý život XII (KUZ XII)*, Praha, s. 214-219, 2012.

REMAINING PUBLICATIONS OF THE AUTHOR

Publications in journals with impact factor

1. **K. Procházková (equal contribution)**, S. Daniš, and P. Svoboda, "Specific heat study of PrNi4Si," *Acta Physica Polonica-Series A*, vol. 113, no. 1, pp. 299–302, 2008.
Impact factor (2015): 0.43

2. **K. Štěpánová (80 % contribution)**, M. Vavrečka, L. Lhotská, "Changes in Strategy During the Mental Rotation Task," **(abstract)** In: *Clinical Neurophysiology*, vol.123, no.3, p. e10, 2012.
Impact factor (2015): 3.426

3. **K. Štěpánová (80 % contribution)**, M. Vavrečka, L. Lhotská, "Relation Between Human Brain Activity During Mental Rotation and Mathematical Abilities," **(abstract)**, In: *Activitas Nervosa Superior Rediviva*, Bratislava, pp. 148, 2011.
Impact factor (2011): 0.5

Other Publications

4. P. Bukovský, **K. Štěpánová (equal contribution)**, M. Vavrečka, "Differences in EEG Between Gifted and Average Gifted Adolescents: Mental Rotation Task," In *International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM)*, Prague, Czech Republic, 2015.

5. **K. Štěpánová (equal contribution)**, P. Volf, J. Durdiaková, M. Vavrečka, L. Lhotská, "EEG Signal and Behavioral Data Analysis of Mentaly Gifted and Average Adolescents Performing Mental Rotation Task," In: *Proceedings of BioDat 2014 - Conference on Advanced Methods of Biological Data and Signal Processing*, Prague, 2014.

6. **K. Štěpánová (equal contribution)**, M. Vavrečka, J. Durdiaková, L. Lhotská, "Differences of EEG Signal between Gifted and Average Adolescents," In: *61. společný sjezd České a Slovenské společnosti pro klinickou neurofyziologii*, 2014.

7. **K. Štěpánová (40 % contribution)**, E. Bakštein, M. Vavrečka, D. Novák, "Methods for Students' Motivation During the Biomedical Engineering Study," In: *Trends in biomedical engineering*, Košice, Technical University of Kosice, Slovakia, 2013, pp. 35-38, 2013.

8. **K. Štěpánová (50 % contribution)**, M. Macaš, "Visualizing Correlations Between EEG Features by Two Different Methods," In: *BioDat 2012 - Conference on Advanced Methods of Biological Data and Signal Processing*, 2012.

9. **K. Štěpánová (80 % contribution)**, M. Vavrečka, L. Lhotská, "EEG signal during the mental rotation (adolescents with IQ>130) (in Czech)," In: *Česká a slovenská neurologie a neurochirurgie*. 2012.

10. **K. Štěpánová (60 % contribution)**, M. Macaš, L. Lhotská, "Correlation-Based Neural Gas for Visualizing Correlations between EEG Features," In *7th Int Conf on Soft Comp Models in Industrial and Environm Applications/5th Computational Intelligence in Security for Information Syst/3rd Int Conf on EUropean Transnational Educ*, Ostrava, Czech Republic, 2012.

11. **K. Štěpánová (70 % contribution)**, T. Strašrybka, L. Lhotská, "PSYCHEEg: Matlab toolbox for psychological experiments using Electroencephalographic measurement", *International Symposium on Computers in Education (SIIE)*, Andorra la Vella, Andorra, 2012.

12. M. Vavrečka, **K. Štěpánová (20 % contribution)**, L. Lhotská, "Elimination of interindividual differences in the area of EEG processing (in Czech)," In: *Proceedings of Trendy v biomedicínském inženýrství*, Ostrava, s.184-186, 2011.