SAND2004-4518
Unlimited Release
Printed September 2004

A Robotic Framework for Semantic Concept Learning

Kevin M. Squire
Stephen Levinson

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Patrick G. Xavier

Intelligent Systems and Robotics Center
Sandia National Laboratories
P. O. Box 5800, M/S 1004
Albuquerque, NM 87185-1004

Abstract

This report describes work carried out under a Sandia National Laboratories Excellence in Engineering Fellowship in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. Our research group (at UIUC) is developing a intelligent robot, and attempting to teach it language. While there are many aspects of this research, for the purposes of this report the most important are the following ideas. Language is primarily based on semantics, not syntax. To truly learn meaning, the language engine must be part of an embodied intelligent system, one capable of using associative learning to form concepts from the perception of experiences in the world, and further capable of manipulating those concepts symbolically. In the work described here, we explore the use of hidden Markov models (HMMs) in this capacity. HMMs are capable of automatically learning and extracting the underlying structure of continuous-valued inputs and representing that structure in the states of the model. These states can then be treated as symbolic representations of the inputs. We describe a composite model consisting of a cascade of HMMs that can be embedded in a small mobile robot and used to learn correlations among sensory inputs to create symbolic concepts. These symbols can then be manipulated linguistically and used for decision making.

This is the project final report for the University Collaboration LDRD project, "A Robotic Framework for Semantic Concept Learning".

1 Introduction

This report describes the study of cognitive development using a constructive approach. The basis of our work can be summarized as follows. We believe that human intelligence, and hence language, is primarily semantic. We believe that the mind forms semantic concepts through the correlation of events and cues close together in time and/or space. We further believe that an integrated sensory-motor system is necessary to ground these concepts and allow the mind to form a semantic representation of reality—there is no such thing as a disembodied mind.

Starting with these ideas, we are developing a robotic platform in ongoing work at the University of Illinois at Urbana-Champaign, complete with basic sensory-motor and computing capabilities. The sensory-motor components are functionally equivalent to their human or animal counterparts, and include binaural hearing, stereo vision, tactile sense, and basic proprioceptive control. On top of these components, our group is implementing various processing and learning models, with the intention of creating and aiding semantic understanding and intelligent behavior. Our goal is to produce a robot that will learn to understand and carry out simple tasks in response to natural language requests.

This technical report describes work on a semantic learning model completed under a Sandia National Laboratories Excellence in Engineering Fellowship. The rest of the report is organized as follows. Section 2 gives a brief overview of our robotic framework. Section 3 introduces hidden Markov models (HMMs) and recursive maximum-likelihood estimation (RMLE), both of which we use as part of our semantic learning model, described in Section 4. We offer some further results and conclusions in Section 5.

2 Overview: A Robotic Framework for Studying Cognition

We use the cognitive cycle depicted in Fig. 1 to guide the design of our robotic system. This simple diagram shows the flow of cognition among four systems: a sensory system, an associative memory, a working memory, and a vocalization and motor system. The diagram is reminiscent of ones used by psychologists to describe the human memory system (see e.g., [1], p. 66), with some some additional emphasis on the associative nature of memory and on the embodiment and interaction of the system with the environment. These emphasized areas are key requirements for embodied learning. Below we describe two different views of this cycle: the somatic system view (the body), and the noetic system view (the mind).

2.1 Somatic System

The somatic system is the physical, "body" component of the mind-body system. It is comprised of the physical components necessary for cognition: the senses, muscular (motor) system, nervous system, and the brain.

To do the most human-like cognitive studies, we would like to work with a robot which is as anthropomorphic as possible. For our work, we chose Arrick Robotics' Trilobot [2] (see Fig. 2). The robot's anthropomorphic capabilities are rich enough to suit our purposes. In particular, the robot can move freely via wheels, can move its head, and use its arm to manipulate common objects, allowing relatively complex behaviors. A speaker is

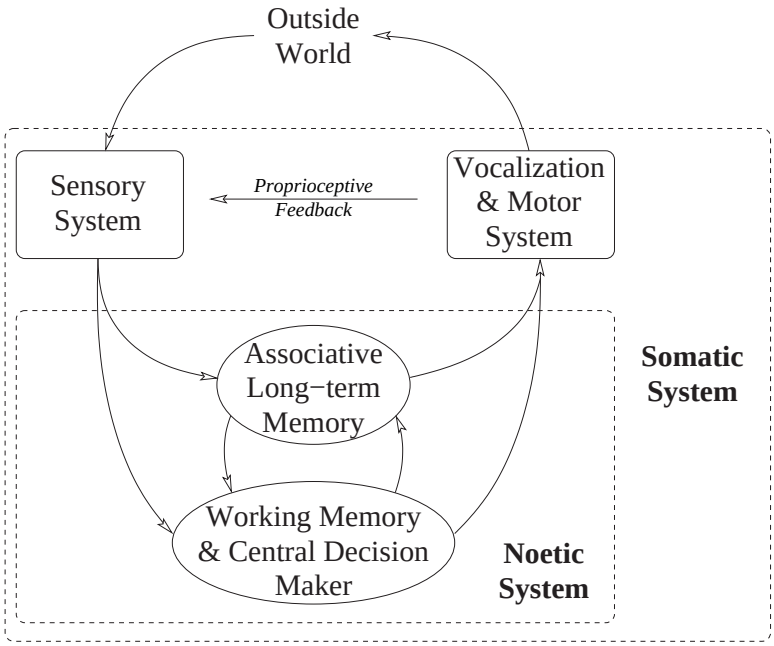


Figure 1: Cognitive Cycle. This figure shows the flow of cognition among the senses, long term memory, working memory, the motor system, and the environment.

available on-board for the production of sounds and, with additional processing, speech. We have added cameras and microphones to the robot to give it stereo vision and hearing capabilities, and have implemented, in software, some basic audio and visual processing and feature extractors to mimic aspects of these systems [3–8]. The robot also has a number of touch and other sensors available. We have incorporated a computer on-board which collects input from the cameras, microphones, and sensors, and sends control commands to the robot. The computer can also handle limited processing of the data, but a wireless transmitter is available to transmit the data to other workstations, where most processing occurs. This distributed system of computers houses the "brain" of our robot. To facilitate the communications necessary for this system, we did extensive design and coding of a distributed communications and processing framework early in this research. See [3] for details.

2.2 Noetic System

The noetic system in Fig. 1 represents the "mind" aspect in the mind-body paradigm. The main goal of our research is to implement functional equivalents for high-level cognitive functions in this area. We can characterize this goal by looking at three different aspects of the mind: memory, learning, and behavior.

Memory is often described hierarchically, dividing first into short-term memory and long-term memory. Long-term memory is further divided into procedural, semantic, and episodic memories [1,9,10]. While they are all connected and interrelated, our interest here is on semantic memory—our knowledge and understanding of the world. As indicated in the introduction, we believe that memory is primarily associative.

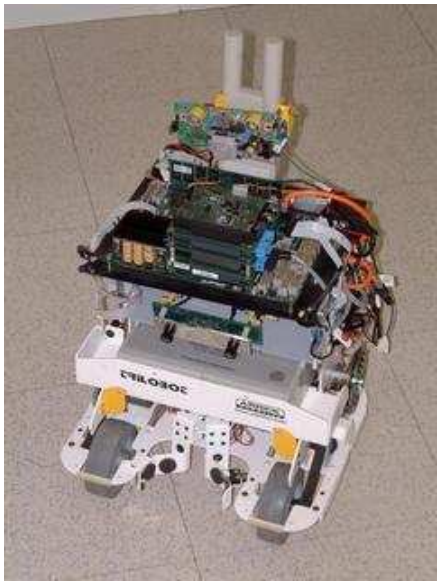


Figure 2: Illy, one of two Arrick Robotics Trilobots we use for our cognition and language acquisition research. The base unit for the robots was heavily augmented, to include stereo cameras and microphones, an on-board computer and wireless ethernet.

Learning can be described informally as a transition from one mental state to another where information is gained [11]. If memory is primarily assocative, then learning must principally involve the formation of associations. According to D. Shanks, in associative learning, "the environment provides a relationship among contingent events, allowing [a] person to predict one [event] in the presence of others." [11] Possible events include both environmental cues and the subject's own behavior. The relationship between or among events can be *causal* or *structural*. In causal relationships, one event occurs, followed by another, perhaps after a brief time interval. For example, there is a consistent causal relationship between touching a hot burner and feeling pain. Structural relationships relate features or properties of an object or event with other features which frequently co-occur. For example, after both seeing and smelling a fire, the presense of one of these events generally indicates the presence of the other. A less obvious example of a structural relationship is the association of a word with a particular object or event, a key focus of our research. This type of association allows the formation of symbolic concepts, permitting symbolic manipulation.

If memory contains our knowledge about the world, and learning modifies that knowledge, behavior puts that knowledge into use. While behavioral expression is an integral component of our long term research goals, it is not as immediately important to the research described herein.

The work described in the next few sections describes our composite HMM for associative learning. It consists of a cascade of hidden Markov models (HMMs), with models lower in the cascade responsible for learning low-level sensory-motor concepts, and models higher in the cascade responsible for learning higher concepts. The next section gives a formal description of HMMs and briefly describes an on-line learning algorithm for training them.

Section 4 then describes our hierarchical model for associative learning using HMMs.

3 Hidden Markov Models and Recursive Maximum Likelihood Estimation

A hidden Markov model (HMM) is a discrete time stochastic process with two components, $\{X_n, Y_n\}$, where (i)$\{X_n\}$ is a finite-state Markov chain, and (ii) given $\{X_n\}$, $\{Y_n\}$ is a sequence of conditionally-independent random variables. The conditional distribution of Y_k depends on $\{X_n\}$ only through X_k. The name HMM arises from the assumption that $\{X_n\}$ is not observable, and so its statistics can only be ascertained from $\{Y_n\}$.

HMMs have many interesting features that we believe can be easily exploited for learning associations. As noted previously, concepts can be formed from the correlation in time among events. HMMs by construction have a notion of sequence, and have proven quite effective at learning time series and spatial models in areas such as speech processing [12] and computational biology [13–15]. This characteristic of HMMs provides a useful starting point for learning time correlation.

Another property of HMMs useful for learning concepts is their ability to discover structure in input data. Cave and Neuwirth [16] demonstrated this capability by training a low order ergodic HMM on text. They found that the states of the model represented broad categories of letters, discovering some of the underlying structure of the text. Poritz [17] developed a similar model for speech data, and Ljolje and Levinson [18] created a speech recognizer based on this type of model. Our hierarchical model exploits this natural capability of HMMs to discover structure in order to learn higher level concepts.

Finally, in addition to their familiar role as recognizers, HMMs can be used in a generative capacity. In particular, when placed in a hierarchy, we can drive the various HMMs to produce sequences of states and corresponding output, roughly simulating thoughts and actions.

Some characteristics of HMMs are not as useful for our work, however. Two of the most common methods used for HMM parameter estimation, the Baum-Welch method and methods based on the Viterbi algorithm (for both, see e.g., [12]), both require off-line processing of large amounts of data. For our goal of learning concepts in real time using a robot, these methods are not very useful. We would much prefer an iterative or on-line training procedure.

There are generally two approaches researchers have used to implement on-line stochastic training procedures for HMMs. The first minimizes the prediction error of the model via recursive methods [19–21]. The other approach maximizes the Kullback-Leibler information between the estimated model and true model, or equivalently, maximizes the likelihood of the estimated model for an observation sequence [21–26]. This is the approach we have chosen. Our recursive maximum-likelihood estimation (RMLE) algorithm is based mostly on [26]. Using their general derivation for HMMs with continuous observation densities, we have derived formulas specific to multidimensional Gaussian observations, and extended their work and proofs to HMMs with discrete observations. We describe that work briefly below. For details, see [3].

3.1 HMM Signal Model

An HMM is a discrete time stochastic process with two components, $\{X_n, Y_n\}$ defined on probability space $(\Omega, \mathcal{F}, P)$.[1] Let $\{X_n\}_{n=1}^{\infty}$ be a discrete time first order Markov chain with state space $R = \{1, \ldots, r\}$, r a fixed known constant. The model starts in a particular state $i = 1, \ldots, r$ with probability $\pi_i = P(X_1 = i)$. Define $\pi \in \Pi$ by $\pi = \{\pi_i\}$, where Π is the set of length-r stochastic vectors. For $i, j = 1, \ldots, r$, the transition probabilities of the Markov chain are given by

$$a_{ij} = P(X_n = j | X_{n-1} = i). \tag{1}$$

Let $A = \{a_{ij}\}$. Then $A \in \mathcal{A}$, where $\mathcal{A}$ is the set of all $r \times r$ stochastic matrices.

In a hidden Markov model, $\{X_n\}$ is not visible, and its statistics can only be ascertained from a corresponding observable stochastic process, $\{Y_n\}$. The process $\{Y_n\}$ is a probabilistic function of $\{X_n\}$, i.e., given X_n, Y_n takes values from some space E according to a conditional probability distribution. The corresponding conditional density of Y_n is generally assumed to belong to a parametric family of densities $\{b(\cdot; \theta) : \theta \in \Theta\}$, where the density parameter θ is a function of X_n, and Θ is the set of valid parameters for the particular conditional density assumed by the model. The conditional density of Y_n given $X_n = j$ can be written $b(\cdot; \theta_j)$.

Example 1 *(Gaussian observation density): Suppose the observation density for each state in an HMM is described by a univariate Gaussian distribution. Then parameter set* $\Theta = \{(\mu, \sigma) \in \mathbb{R} \times (0, \infty)\}$, $\theta_j \in \Theta$, *and* $\{Y_n\} = \{y_n\}$ *is a sequence of continuously valued conditionally independent outputs on* $\mathbb{R}$, *each with probability distribution*

$$b\left(y_n; \mu_j, \sigma_j\right) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(y_n - \mu_j)^2}{2\sigma_j^2}\right] \tag{2}$$

for $X_n = j$.

Example 2 *(Discrete observation density): Suppose observations* Y_n *are drawn from a discrete set of symbols* $V = \{v_k\}$, $k = 1, \ldots, s$. *Then* $\Theta = \{(b_1, \ldots, b_s) \in [0, 1]^s | \sum_{k=1}^{s} b_k = 1\}$ *is the set of length-s stochastic vectors,* $\theta_j \in \Theta$, *and* $\{Y_n\} = \{y_n\}$ *is a sequence of discrete symbols, each* y_n *having probability*

$$b(y_n; \theta_j) = b_{jk}|_{y_n = v_k} \tag{3}$$

for $X_n = j$.

Here, for simplicity, Y_n are assumed to be scalar valued, although the formulation here easily generalizes to vector-values.

Define the HMM parameter space as $\Phi = \Pi \times \mathcal{A} \times \Theta$. The model $\varphi \in \Phi$ is then defined as

$$\varphi = \{\pi_1, \ldots, \pi_r, a_{11}, a_{12}, \ldots, a_{rr}, \theta_1, \ldots, \theta_r\}. \tag{4}$$

The model parameters for a particular model are accessed via coordinate projections, e.g., $a_{ij}(\varphi) = a_{ij}$. In some cases (e.g., when considering the RMLE algorithm below), we will not

[1] $(\Omega, \mathcal{F}, P)$ is needed for the formal definition of an HMM but is not necessary to grasp the rest of this paper. Ω is a set of events of interest and $\mathcal{F}$ is a σ-field describing the class of subsets of Ω over which a probability measure P is defined. See, e.g., Chapter 1 of [27] for a more complete decription.

be concerned with estimating π. In that case, $\Phi = \mathcal{A} \times \Theta$, and φ changes accordingly. Note that the literature occasionally describes other model parameterizations (see, e.g., [20, 22]).

Let p be the length of φ. When estimating model parameters, let $\varphi^* \in \Phi$ be the fixed set of "true" parameters of the model we are trying to estimate.

For a vector or matrix v, v' represents its transpose. Define the r-dimensional column vector $b(y_n; \varphi)$ and $r \times r$ matrix $B(y_n; \varphi)$ by

$$b(y_n; \varphi) = [b_1(y_n; \theta_1(\varphi)), ..., b_r(y_n; \theta_r(\varphi))]' \tag{5}$$

and

$$B(y_n; \varphi) = \text{diag}[b_1(y_n; \theta_1(\varphi)), ..., b_r(y_n; \theta_r(\varphi))]. \tag{6}$$

$b(y_n; \varphi)$ and $B(y_n; \varphi)$ give the observation density evaluated at y_n for each state (in model φ), as a vector and diagonal matrix, respectively.

Using the definitions above, it can be shown (see, e.g., [28]) that the likelihood of the sequence of observations $(y_1, \ldots, y_n)$ (also the joint distribution of $(Y_1, \ldots, Y_n)$) for model φ is given by

$$p_n(y_1, \ldots, y_n; \varphi) = \pi(\varphi)' B(y_1; \varphi) \prod_{k=2}^{n} A(\varphi) B(y_k; \varphi) \mathbf{1}_r. \tag{7}$$

3.2 Recursive Maximum Likelihood Estimation of HMM Parameters

Maximum-likelihood estimation (MLE) is formally defined as follows. For observation sequence $(y_1, \ldots, y_n)$, find

$$\hat{\varphi} = \arg\max_{\varphi \in \Phi} p_n(y_1, \ldots, y_n; \varphi), \tag{8}$$

where $\hat{\varphi}$ is the most likely estimate of the true underlying parameters φ^*. The recursive maximum-likelihood estimation (RMLE) algorithm defined here is a online iterative solution to this problem.

The derivation of the RMLE algorithm for HMMs proceeds as follows. We first show how to calculate the likelihood $p_n(y_1, \ldots, y_n; \varphi)$ for a given HMM model recursively, using prediction (or forward) filters. We note that maximizing $\log p_n(y_1, \ldots, y_n; \varphi)$ is equivalent to and generally easier than maximizing $p_n(y_1, \ldots, y_n; \varphi)$ [29], and that $\log p_n(y_1, \ldots, y_n; \varphi)$ can also be calculated recursively. We can then search for the maximum of $\log p_n(y_1, \ldots, y_n; \varphi)$ using the derivative of the update of this recursion.

For the results of this section to hold, it is necessary to assume various conditions on periodicity, continuity, and ergodicity for the model. For simplicity, we will assume all necessary conditions hold. Please see [3, 26] for details. This derivation was largely taken from [26].

Define the prediction filter as

$$u_n(\varphi) = [u_{n1}(\varphi), \ldots, u_{nr}(\varphi)]' \tag{9}$$

where

$$u_{ni}(\varphi) = P(X_n = i | y_{n-1}, \ldots, y_1), \tag{10}$$

is the probability of being in state i at time n given all previous observations. Using this filter, the likelihood $p_n(y_1, \ldots, y_n; \varphi)$ can be written as

$$p_n(y_1, \ldots, y_n; \varphi) = \prod_{k=1}^{n} b(y_k; \varphi)' u_k(\varphi). \tag{11}$$

(See [3], Appendix E.)

The value of $u_n(\varphi)$ can be calculated recursively as

$$u_{n+1}(\varphi) = \frac{A(\varphi)' B(y_n; \varphi) u_n(\varphi)}{b(y_n; \varphi)' u_n(\varphi)} \tag{12}$$

when initialized by $u_1(\varphi) = \pi(\varphi)$.

Let $w_n^{(l)}(\varphi) = (\partial/\partial\varphi_l) u_n(\varphi)$ be the partial derivative of $u_n(\varphi)$ with respect to (w.r.t.) the lth component of φ. Each $w_n^{(l)}(\varphi)$ is an r-length column vector, and

$$w_n(\varphi) = (w_n^{(1)}(\varphi), w_n^{(2)}(\varphi), \ldots, w_n^{(p)}(\varphi)) \tag{13}$$

is an $r \times p$ matrix. Taking the derivative of $u_{n+1}(\varphi)$ from Equation 12,

$$\begin{aligned} w_{n+1}^{(l)}(\varphi) &= \frac{\partial u_{n+1}(\varphi)}{\partial \varphi_l} \\ &= R_1(y_n, u_n(\varphi), \varphi) w_n^{(l)}(\varphi) + R_2^{(l)}(y_n, u_n(\varphi), \varphi) \end{aligned} \tag{14}$$

where

$$R_1(y_n, u_n(\varphi), \varphi) = A(\varphi)' \left[I - \frac{B(y_n; \varphi) u_n(\varphi) \mathbf{1}_r'}{b(y_n; \varphi)' u_n(\varphi)} \right] \frac{B(y_n; \varphi)}{b(y_n; \varphi)' u_n(\varphi)} \tag{15}$$

$$\begin{aligned} R_2^{(l)}(y_n, u_n(\varphi), \varphi) = {} & A(\varphi)' \left[I - \frac{B(y_n; \varphi) u_n(\varphi) \mathbf{1}_r'}{b(y_n; \varphi)' u_n(\varphi)} \right] \frac{[\partial B(y_n; \varphi)/\partial \varphi_l] u_n(\varphi)}{b(y_n; \varphi)' u_n(\varphi)} \\ & + \frac{[\partial A(\varphi)'/\partial \varphi_l] B(y_n; \varphi) u_n(\varphi)}{b(y_n; \varphi)' u_n(\varphi)}. \end{aligned} \tag{16}$$

Using these equations, we can recursively calculate $w_n(\varphi)$ at every iteration.

For a set of observations $(y_1, ..., y_n)$, we would like to find the maximum of $p_n(y_1, \ldots, y_n; \varphi)$. Equivalently, we can maximize $\log p_n(y_1, \ldots, y_n; \varphi)$. Define the log-likelihood of observations $(y_1, \ldots, y_n)$ as

$$\ell_n(\varphi) = \frac{1}{n+1} \log p_n(y_1, ..., y_n; \varphi). \tag{17}$$

Using Equation 11, we can rewrite this as

$$\ell_n(\varphi) = \frac{1}{n+1} \sum_{k=1}^{n} \log[b(y_k; \varphi)' u_k(\varphi)]. \tag{18}$$

To estimate the set of optimal parameters φ^*, we want to find the maximum of $\ell_n(\varphi)$, which we will attempt via recursive stochastic approximation. For each parameter l in φ, at

each time n, we take $(\partial/\partial\varphi_l)$ of the most recent term inside the summation in Equation 18, to form an "incremental score vector"

$$S(\tilde{Y}_n;\varphi) = \left(S^{(1)}(\tilde{Y}_n;\varphi), ..., S^{(p)}(\tilde{Y}_n;\varphi)\right)' \tag{19}$$

with

$$\begin{aligned} S^{(l)}(\tilde{Y}_n;\varphi) &= \frac{\partial}{\partial\varphi_l}\log[b(y_n;\varphi)'u_n(\varphi)] \\ &= \frac{b(y;\varphi)'[(\partial/\partial\varphi_l)u_n(\varphi)] + [(\partial/\partial\varphi_l)b(y_n;\varphi)]'u_n(\varphi)}{b(y_n;\varphi)'u_n(\varphi)} \\ &= \frac{b(y;\varphi)'w_n(\varphi) + [(\partial/\partial\varphi_l)b(y_n;\varphi)]'u_n(\varphi)}{b(y_n;\varphi)'u_n(\varphi)} \end{aligned} \tag{20}$$

where

$$\tilde{Y}_n \triangleq (Y_n, u_n(\varphi), w_n(\varphi)). \tag{21}$$

The RMLE algorithm takes the form

$$\varphi_{n+1} = \Pi_G\left(\varphi_n + \epsilon_n S(\tilde{Y}_n;\varphi_n)\right) \tag{22}$$

where ϵ_n is a sequence of step sizes satisfying $\epsilon_n \geq 0$, $\epsilon_n \to 0$ and $\sum_n \epsilon_n = \infty$, G is a compact and convex set (here, $G \subseteq \Phi$, the set of all valid parameter sets φ), and Π_G is a projection onto set G. The purpose of the projection is to ensure valid probability distributions and maintain all necessary conditions. Note that Equation 22 is a gradient update rule, with constraints.

Krishnamurthy and Yin have proved convergence of this learning method for HMMs with continuous observation densities in [26]. In [3] we give an argument for extending this proof to discrete observations.

Equations 16 and 20 can both be simplified for each type of parameter in φ. In [3] we give derivations of both equations for

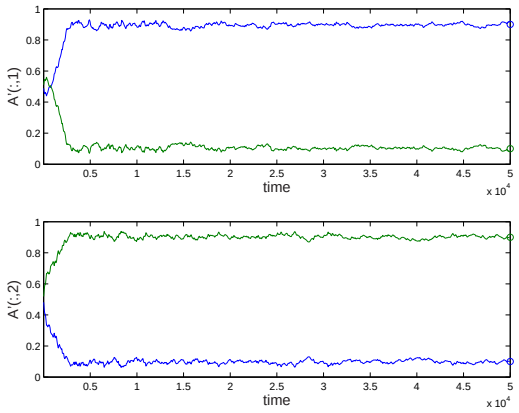
1. transition probabilities $a_{ij}(\varphi)$,
2. observation probabilities $b_{jk}(\varphi)$ when assuming discrete observations,
3. mean vector $\mu_j(\varphi)$ and covariance matrix $\Sigma_j(\varphi)$, when assuming continuous observations taken from multidimensional Gaussian distributions, and
4. $R_j(\varphi)$ for $\Sigma_j(\varphi) = R_j'(\varphi)R_j(\varphi)$, where $R_j(\varphi)$ is the upper-triangular matrix of the Cholesky decomposition of $\Sigma_j(\varphi)$ in a multidimensional Gaussian distribution.

3.3 Numerical Simulations

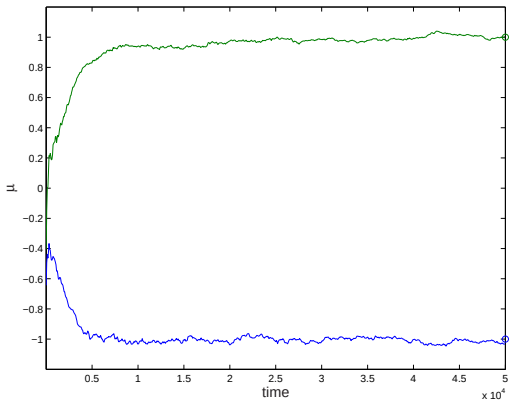
The model and algorithm presented above were implemented in Matlab and tested via Monte Carlo simulation, for various model types and parameters. Fig. 3 shows one example of such a run.

For this example, we generated data from a simple two state model, with transition matrix

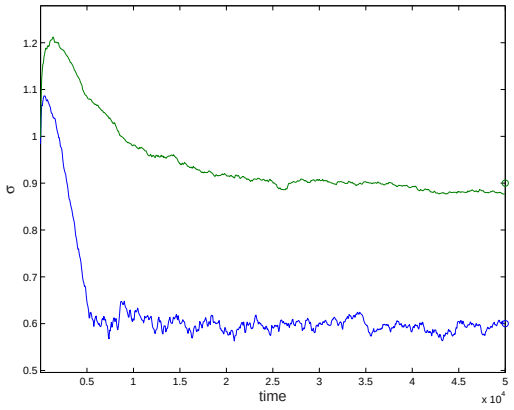
$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$



(a) Transition probability estimates.



(b) Gaussian mean estimates.



(c) Gaussian standard deviation estimates.

Figure 3: Training example. ($\varepsilon = \frac{.1}{n^{0.5}}$)

and observations generated from Gaussians with parameters

$$\mu = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}, \quad \sigma = \begin{bmatrix} 0.6 \\ 0.9 \end{bmatrix}.$$

For training, we set $\varepsilon_n = \frac{0.1}{n^{.5}}$, and initialized the training model to

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \quad \mu = \begin{bmatrix} -0.75 \\ -0.50 \end{bmatrix}, \quad \sigma = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}.$$

The parameters converged within 40,000 iterations.

Note, again, that this is an online training algorithm, so the model is always updating model parameters as observations occur. The number of iterations for convergence is highly variable and depends on a large number of factors, including the number of parameters in the model, training rate parameters, and the characteristics of the data used for training. However, we may be able to generally improve the efficiency of the training algorithm, using ideas discussed in the next section.

More extensive testing and analysis with multiple initial parameter values are presented in [3].

3.4 Model Averaging and Tracking

When analyzing stochastic approximation algorithms, one goal is to improve asymptotic efficiency. One way to do this is to use averaging. Krishnamurthy and Yin [26] suggest the averaging in both the iterates (i.e., φ_n) and the observations (as measured by $S(\tilde{Y}, \varphi)$). This averaging takes the form

$$\varphi_{n+1} = \Pi_G(\bar{\varphi}_n + \varepsilon_n n \bar{S}_n) \tag{23}$$

$$\bar{\varphi}_{n+1} = \bar{\varphi}_n - \frac{1}{n+1}\bar{\varphi}_n + \frac{1}{n+1}\varphi_{n+1} \tag{24}$$

$$\bar{S}_{n+1} = \bar{S}_n - \frac{1}{n+1}\bar{S}_n + \frac{1}{n+1}S_{n+1}, \tag{25}$$

with $\varepsilon_n = 1/n^\gamma$, $0.5 \leq \gamma \leq 1$. In [26], Krishnamurthy and Yin provide convergence, asymptotic optimality, and asymptotic normality proofs for the modified algorithm. These formulas can also be modified to work with a "fixed history" by replacing n in Equations 23-25 with a fixed constant k, or, alternatively, $\min(n, k)$. Various sources [23, 26] also suggest the use of fixed ε for use in tracking. Analysis of the RMLE algorithm for tracking slowly varying HMM parameters also appears in [26]. We provide numerical simulations and analysis of all of these variations in [3].

3.5 Learning a Model of Unknown Order[2]

For HMMs, it is generally assumed that the number of states needed to represent an underlying process is known. When working with a real system, however, knowing the optimal number of states may be difficult or impossible. A recent tutorial paper on hidden Markov

[2]This work was completed after the end of funding from the Sandia National Laboratories Fellowship, but is included in the interest of completeness.

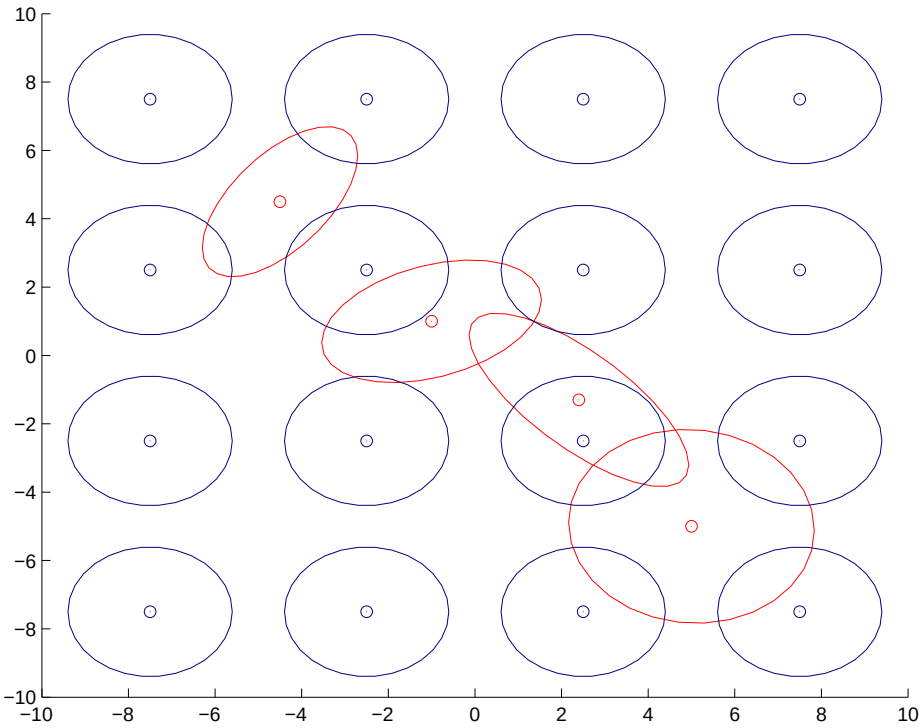


Figure 4: Initialization of an HMM with two-dimensional Gaussian observation densities. Each density is indicated on the graph by its mean and a contour line containing 80% of the density. Each density is also shaded according to the stationary probability of its state, with more likely states shaded darker. Densities of the model to be learned are drawn in red.

processes by Ephraim and Mehrev [30] summarizes the state of the art of order estimation in hidden Markov models. In almost all cases this involves the comparison of a large number of learned models with different state orders.

We propose an *ad hoc* approach for learning the underlying state order of a set of observations. Our proposal is for HMMs with Gaussian observation densities, although it should be valid for other densities.

Our method works as follows. First, we initialize a model with a large number of states, with the observation densities initially covering the region of space occupied by the observations. In our example, we assume that our observations will be contained in the region $\{(x, y) : x, y \in (-10, 10)\}$, and we choose to start with 16 states with Gaussian densities equally spaced throughout this region. Fig. 4 shows this setup,where the densities for each state are drawn in blue. The densities of the states in the model to be learned are drawn in red.

Note that there is no indication on this graph of transition probabilities. However, in this figure and in the graphs in Fig. 5, the density associated with each state is colored according to that state's stationary probability, derived from the stationary distribution of the transition probability matrix A. Initially, all transition probabilities are equal, so the stationary distribution (and therefore the distribution coloring) is uniform. Darker coloring

of mean and contour lines indicates higher stationary probability for a particular state.

The parameters of the source model in this experiment are

$$A = \begin{bmatrix} .7 & .1 & .1 & .1 \\ .1 & .7 & .1 & .1 \\ .1 & .1 & .7 & .1 \\ .1 & .1 & .1 & .7 \end{bmatrix}, \mu = \begin{bmatrix} (-4.5, 4.5) \\ (-1, 1) \\ (2.4, -1.3) \\ (5, -5) \end{bmatrix},$$

$$\Sigma_1 = \begin{bmatrix} 1.0 & 0.75 \\ 0.75 & 1.5 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 2.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 2.0 & -1.5 \\ -1.5 & 2.0 \end{bmatrix} \quad \Sigma_4 = \begin{bmatrix} 2.5 & -0.1 \\ -0.1 & 2.5 \end{bmatrix}.$$

Fig. 5 documents the progression of the training. As can be seen, by 200,000 iterations, the active states in the model have generally converged to the original model, and states far away from sample data have remained inactive. More study is needed to analyze this procedure.

3.6 Discussion

This section explored the use of the recursive maximum-likelihood estimation (RMLE) algorithm for on-line training of hidden Markov models (HMMs). We have successfully trained models using the algorithm, exploring how various combinations of training parameters affect learning. We have also successfully demonstrated that a large model with states whose distributions cover a section of space can correctly learn the structure of that space. The next section will describe the use of HMMs in a cascade structure for learning semantic concepts.

4 An Associative Memory Model for Semantic Learning

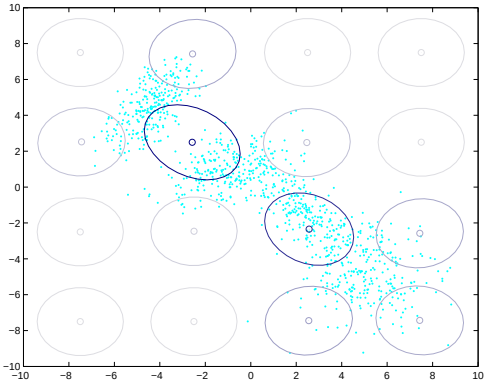
Let us restate our basic assumptions: first, language is primarily semantic, that is, it is concerned mostly with our knowledge of the world; second, this understanding is gained by recognizing and learning relationships between or among events and cues in the environment; and third, that this learning requires the learner to be embodied and situated in the environment. In this section, we will develop a basic model for learning semantic associations from environmental cues.

Our focus is on semantic knowledge gained primarily through repeated stimulation from the environment, and so, for now, we are ignoring one-shot or fast-map learning [31–34].

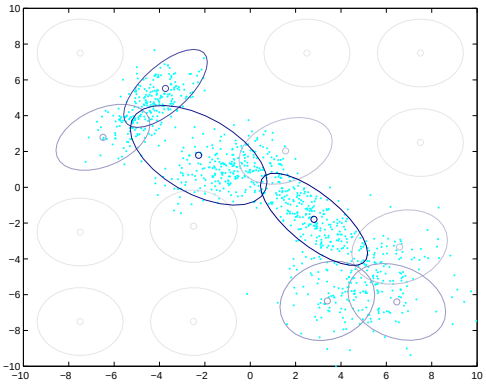
We also note that, because of our focus on learning, our work is similar to research in multimodal learning. As such, our approach may be applicable to research in that area.

4.1 General Associative Memory Model for Semantic Learning

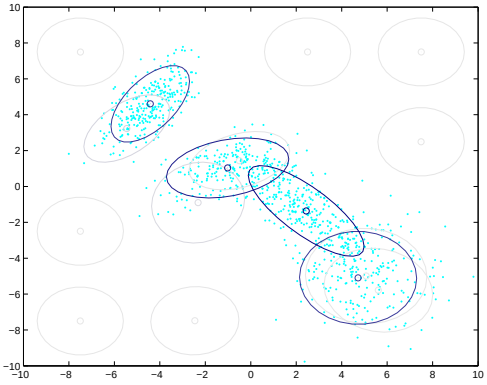
Semantics is meaning. It is our knowledge of the world and how it works. Evolutionarily and developmentally, we have first learned our knowledge of the world through the correlation of sensory-motor events and cues. Some examples pointed out in Section 2.2 include learning



(a) 1000 iterations



(b) 33,000 iterations



(c) 208,000 iterations

Figure 5: Learning an HMM with two-dimensional Gaussian observations, using a model with a large number of states. The model was run with history $k = 1000$, and constant learning rate $\varepsilon = .001$.

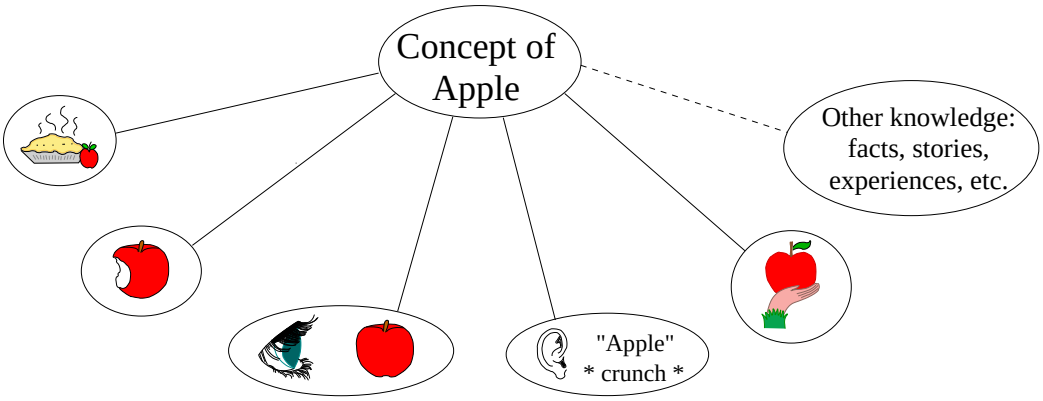


Figure 6: The apple concept is associated with the different ways we sense apples, as well as with other related knowledge.

what happens when one touches a hot burner, learning to associate the sight and smell of fire, and learning to associate a word with an event or some other co-occuring cue.

If we refer to learning simply as association, this has a high degree of agreement with behaviorist theories, particularly with regard to learning the relationship between cues or events and one's own actions. When talking about animal learning, behaviorism is often the best explanation, and it can describe much of human behavior as well. How do human and animal behavior differ then? One important difference is that humans can communicate meaning linguistically, using symbols representing concepts.[3] The question becomes, can we mimic this behavior, that is, can we build a system which can learn meaning in a behaviorist manner (i.e., via association), and in addition, can create symbols that can be manipulated and communicated? We think so.

According to [37], "concepts are the most fundamental constucts in theories of mind." While there is some debate about the definition of concepts, or even whether they exist [37], a concept is generally defined in terms of the features that are associated with it, as well as the rules that relate these features ([10], pp. 409). Fig. 6 shows an example, where the concept of "apple" is associated with its smell, taste, sight, sounds, and feel of an apple, as well as other related knowledge.

One feature to note about Fig. 6 is the fact that the concept is represented as a discrete unit. It does not simply exist as a set of weights connecting two sensory modalities. This formulation differs from that of many of the models often used to associate different information streams, where associative relationships are related directly (e.g., Hopfield networks and related work [38,39], some Bayesian Networks [40], and fused or coupled HMMs [41,42]). This difference is important because it allows the concept to be manipulated as a symbol.

Fig. 7 gives a more abstract illustration of concept connections.

Taking the models one at a time, the visual model independently learns visual concepts of the objects or other cues in its environment. These concepts could include such things as colors, shapes, textures, or types of motion, although each of these may be put into a separate model. The audio model learns concepts from audio cues, including speech. At the

[3] As an aside, chimps, dogs, bees, and some other animals may be able to communicate and/or understand symbols to a limited extent. See, e.g., [34–36].

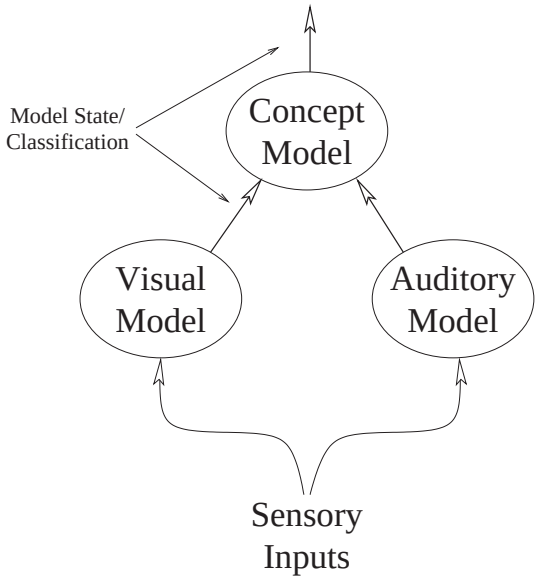


Figure 7: Visual/auditory concept hierarchy

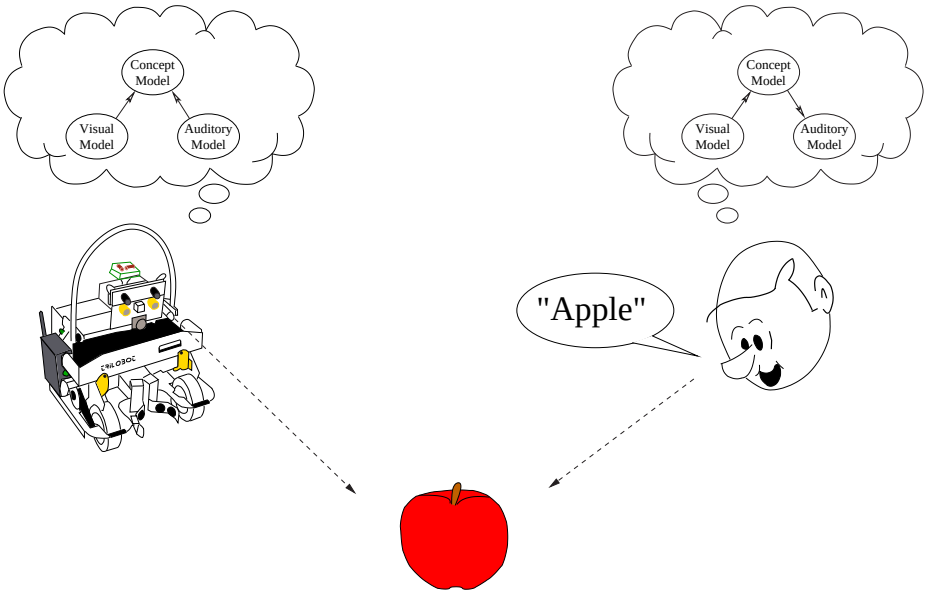


Figure 8: Associative learning of the word “apple”.

lowest level, this might include environmental sounds and phonemes. The concept model learns frequently co-occuring states or classifications of the lower models. Learning in all models is unsupervised, although depending on the model and learning method chosen, models may be initialized with a bias to learn better and/or faster. The model can, of course, scale up to include more types of sensory models.

Note that Fig. 7 is greatly simplified, and as shown, would be able to learn the correlation between the sight and sound of, say running water, but perhaps not word associations. For implementation on a robot, this problem is dealt with in [3].

One necessary condition for effective communication is that the two people (or in our case, the person and the robot) communicating share a similar set of concepts. Thus, the learning of concepts can be described as an attempt to learn a model of another person's knowledge. Fig. 8 shows this idea graphically. The figure shows an interaction between two subjects, a person and a robot, each with his own cognitive model of the world. The immediate goal of the robot is to learn the cognitive model the person is using to understand the immediate environment.

As hinted by the scenario in Fig. 8, we envision this model as one part of a more complex model, designed around the cognitive cycle described by Fig. 1. For reasons highlighted in Section 3.2, we have chosen to use HMMs for the individual models. The next section describes this specialization of the model.

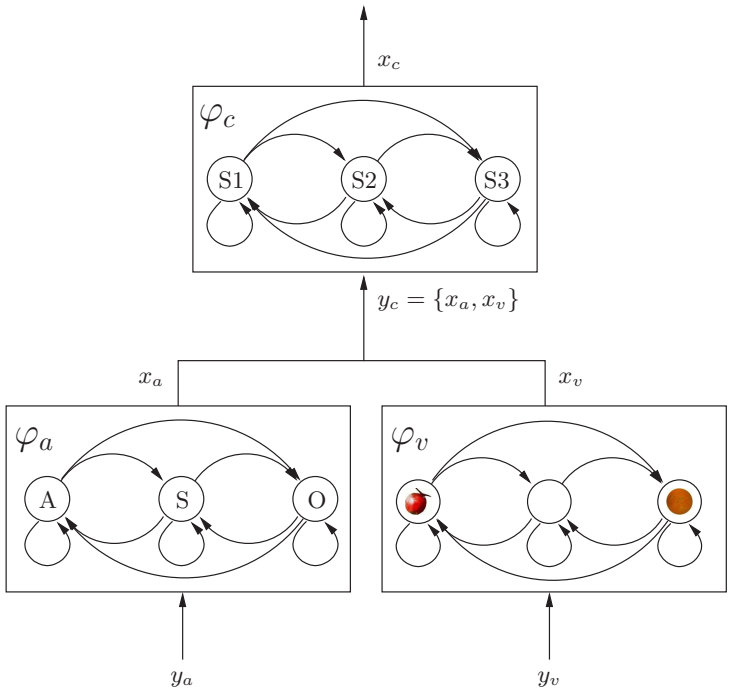


Figure 9: A composite HMM model for associative learning.

4.2 Composite HMM-based Associative Memory

Starting with Fig. 7 and using HMMs, we come up with the composite HMM model shown in Fig. 9, where φ_a corresponds to the auditory model, φ_v corresponds to the visual model, and φ_c corresponds to the concept model. The inputs to the visual and auditory models are feature vectors representing corresponding visual and audio inputs. The output of these models is the discrete state classifications, which are concatenated and used as the input to the concept model.

Since this model is meant to be embedded in a robot, we require an online learning algorithm. In our case, we have chosen to use the RMLE algorithm described in Section 3. In [3], we discuss convergence of the full composite model using the RMLE.

Using the scenario presented in Fig. 8, we have run a Monte Carlo simulation of the composite model in Matlab. In the simulation, we start with a composite HMM representing the knowledge of the human, and another composite HMM representing the knowledge of the robot. The goal is for the robot model to learn the parameters of the human model. The simulation runs as follows:

1. First, a simulated visual stimulus is presented to both the robot model and the human model.

2. The human model recognizes the visual stimulus (via his visual model), "thinks" about it (the concept model recognizes the input), and randomly "speaks" a sound corresponding to the concept (the concept model drives an audio model to produce output).

3. The robot model uses its visual model to classify the visual stimulus according to its current "understanding" (i.e., its current model parameterization), and also updates that "understanding" to closer match the observation (i.e., the parameters of the observation probability distributions for the model states are updated using the RMLE algorithm). It does the same with available audio input. Using the state classifications from the auditory and visual models, the robot model classifies the concept with the concept model and updates its parameters using the RMLE algorithm.

When the simulation is run, the robot model learns a parameterization similar to (though not identical to) the human model. The discrepancy occurs because, as formulated, the simulation is asymmetric. When the human concept model drives the auditory model to produce output, it does not use the transition probabilities of the auditory model itself. This means that the auditory transition probabilities in the robot model learn something other than the transition probabilities of the auditory HMM in the human model. (In fact, as currently implemented, the auditory transition probabilities more closely reflect the human concept transition probabilities.) The concept HMMs transition and observation probabilities in the robot model differ slightly from the human model for related reasons. One possible workaround is to learn the robot model in the current setup, and then use this model to teach a second robot model. This second robot model should be identical to the first. We have not yet tried this experiment. What the results have shown is that the composite model is able to learn associations between two input modalities.

Fig. 10 shows our vision of the full composite HMM as will be implemented in our robot. The lowest level of the cascade consists of a number of HMMs from multiple modalities

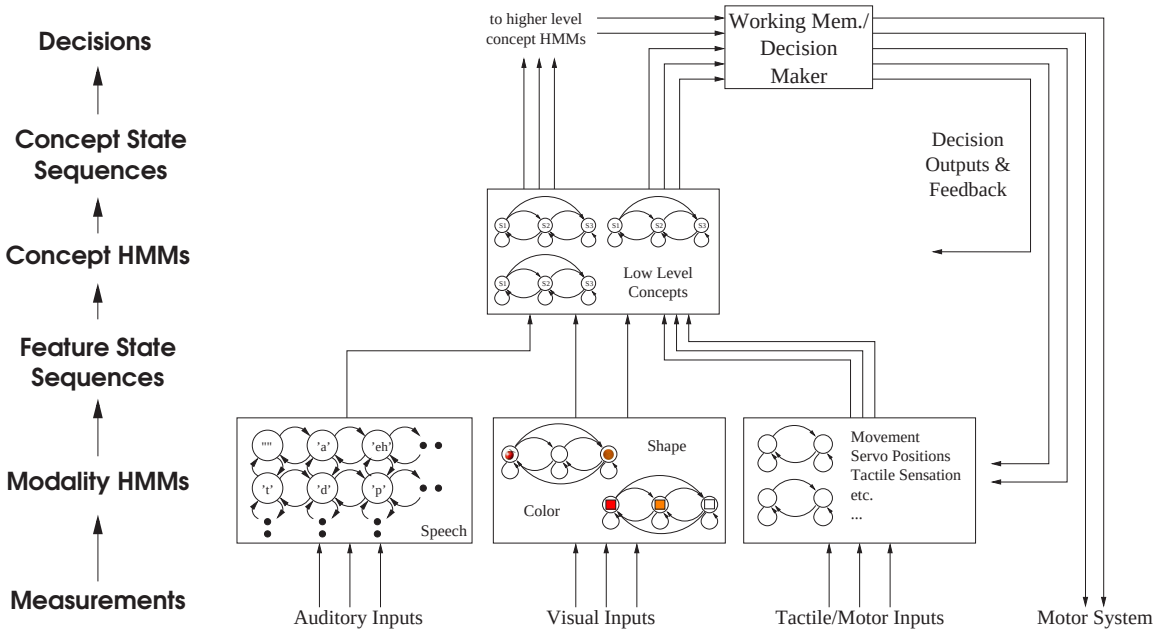


Figure 10: Vision of the entire associative learning system.

(i.e., various HMMs for auditory, visual, proprioceptive and kinesthetic senses). The state outputs of these HMMs will then be used as the inputs to a subsequent layer of HMMs for concept learning. This idea can be extended to more layers for learning higher level concepts. We believe this particular approach to be novel for concept learning, and moreover capable of modeling some of the deep structure of language and reality. See [3] for more details on our implementation.

5 Conclusion

In this report we have described the background and recent progress of our work on a Composite HMM architecture for associative semantic learning. In the past year, we have implemented and extensively studied the RMLE algorithm for training hidden Markov models, and successfully embedded HMMs in a composite model for associative learning of concepts. We are in the progress of running experiments on our robot using this framework. By constructing such a real-world device capable of learning and potentially manipulating symbolic concepts linguistically, we believe this work will help advance the understanding of human cognition and language acquisition.

References

[1] M. H. Ashcraft, *Human Memory and Cognition*. New York: Harper Collins, 1989.

[2] Arrick Robotics, http://www.robotics.com/.

[3] K. Squire, *A Robotic Framework for Semantic Learning*. Ph.d. dissertation, University of Illinois at Urbana-Champaign, 2004. (forthcoming).

[4] W. Zhu and S. E. Levinson, "Edge orientation-based multiview object recognition," in *Proc. IEEE Int'l Conf. on Pattern Recognition*, vol. 1, (Barcelona, Spain), pp. 936–939, 2000.

[5] W. Zhu, S. Wang, R. S. Lin, and S. E. Levinson, "Tracking of object with SVM regression," in *Proc. IEEE Int. Conf. on Comput. Vision & Pattern Recognition*, vol. 2, (Hawaii), pp. 240–245, 2001.

[6] R. S. Lin, "Learning vision-based robot navigation," M.S. thesis, University of Illinois at Urbana-Champaign, 2004.

[7] D. Li and S. E. Levinson, "A robust linear phase unwrapping method for dual-channel sound source localization," in *Int. Conf. on Robot. Automat.*, (Washington D.C.), May 2002.

[8] D. Li and S. E. Levinson, "A Bayes-rule based hierarchical system for binaural sound source localization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, (Hong Kong), Apr. 2003.

[9] D. L. Schacter and E. Tulving, eds., *Memory Systems 1994*. Cambridge, MA: The MIT Press, 1994.

[10] R. L. Solso, *Cognitive Psychology*. Boston: Allyn and Bacon, 4th ed., 1995.

[11] D. R. Shanks, *The Psychology of Associative Learning*. Cambridge: Cambridge University Press, 1995.

[12] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall PTR, 1993.

[13] P. Boufounos, S. El-Difrawy, and D. Ehrlich, "Hidden Markov models for DNA sequencing," in *Workshop on Genomic Signal Processing and Statistics (GENSIPS 2002)*, Oct. 2002.

[14] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling," *J. of Molecular Biology*, vol. 235, pp. 1501–1531, 1994.

[15] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman, "A generalized hidden Markov model for the recognition of human genes in DNA," in *Proc. 4th Int. Conf. Intell. Syst. Molecular Bio.*, pp. 134–142, 1996.

[16] R. L. Cave and L. P. Neuwirth, "Hidden Markov models for English," in *Proc. of the Symposium on the Applications of Hidden Markov Models to Text and Speech*, (Princeton, NJ), pp. 16–56, IDA-CRD, Oct. 1980.

[17] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1291–1294, 1982.

[18] A. Ljolje and S. E. Levinson, "Development of an acoustic-phonetic hidden Markov model for continuous speech recognition," *IEEE Trans. Signal Processing*, vol. 29, no. 1, pp. 29–39, 1991.

[19] A. Arapostathis and S. I. Marcus, "Analysis of an identification algorithm arising in the adaptive estimation of Markov chains," *Math Control Signals Systems*, vol. 3, no. 1, pp. 1–29, 1990.

[20] I. B. Collings, V. Krishnamurthy, and J. B. Moore, "On-line identification of hidden Markov models via recursive prediction error techniques," *IEEE Trans. Signal Processing*, vol. 42, pp. 3535–3539, Dec. 1994.

[21] F. LeGland and L. Mével, "Recursive estimation in hidden Markov models," in *Proc. 36th IEEE Conf. Decision Contr.*, (San Diego, CA), Dec. 1997.

[22] U. Holst and G. Lindgren, "Recursive estimation in mixture models with Markov regime," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1683–1690, Nov. 1991.

[23] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden Markov model parameters based on the Kullback-Leiber information measure," *IEEE Trans. Signal Processing*, vol. 41, pp. 2557–2573, Aug. 1993.

[24] T. Rydén, "On recursive estimation for hidden Markov models," *Stochastic Processes and their Applications*, vol. 66, pp. 79–96, 1997.

[25] F. LeGland and L. Mével, "Recursive identification of HMM's with observations in a finite set," in *Proc. 34th IEEE Conf. Decision Contr.*, (New Orleans), pp. 216–221, Dec. 1995.

[26] V. Krishnamurthy and G. G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Trans. Inform. Theory*, vol. 48, pp. 458–476, Feb. 2002.

[27] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2002.

[28] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal*, vol. 62, pp. 1035–1074, Apr. 1983.

[29] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1994.

[30] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1518–1569, June 2002.

[31] S. Carey and E. Bartlett, "Acquiring a single new word," *Papers and Reports on Child Language Development*, vol. 15, pp. 17–29, 1978.

[32] L. Markson and P. Bloom, "Evidence against a dedicated system for word learning in children," *Nature*, vol. 385, pp. 813–815, Feb. 1997.

[33] K. Yip and G. J. Sussman, "Sparse representations for fast, one-shot learning," *Proc. Nat. Conf. Artif. Intell.*, 1997.

[34] J. Kaminski, J. Call, and J. Fischer, "Word learning in a domestic dog: Evidence for 'fast mapping'," *Science*, vol. 304, pp. 1682–1683, June 2004.

[35] E. S. Savage-Rumbaugh, S. G. Shankar, and T. J. Taylor, *Apes, Language, and the Human Mind*. New York: Oxford University Press, 1998.

[36] J. C. Nieh, "Stingless-bee communication," *American Scientist*, vol. 87, pp. 428–435, Sept. 1999.

[37] S. Laurence and E. Margolis, "Concepts and cognitive science," in *Concepts: Core Readings* (S. Laurence and E. Margolis, eds.), pp. 3–81, Cambridge, MA: The MIT Press, 1999.

[38] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[39] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West Publishing Company, 1992.

[40] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.

[41] H. Pan, *A Bayesian Fusion Approach and Its Application to Integrating Audio and Visual Signals in HCI*. Ph.d. dissertation, University of Illinois at Urbana-Champaign, 2001.

[42] S. M. Chu, *Multimodal Fusion with Applications to Audio-Visual Speech Recognition*. Ph.d. dissertation, University of Illinois at Urbana-Champaign, 2003.