Sketch of an AGI architecture with illustration

András Lőrincz and **Zoltán R. Bárdosi** and **Dániel Takács**
Eötvös Loránd University
Pázmány Péter s. 1/C, Budapest, Hungary 1117

Abstract

Here we present a framework for AGI inspired by knowledge about the only working prototype: the brain. We consider the neurobiological findings as directives. The main algorithmic modules are defined and solutions for each subtasks are given together with the available mathematical (hard) constraints. The main themes are compressed sensing, factor learning, independent process analysis and low dimensional embedding for optimal state representation to be used by a particular RL system that can be integrated with a robust controller. However, the blending of the suggested partial solutions is not a straightforward task. Nevertheless we start to combine these modules and illustrate their working on a simulated problem. We will discuss the steps needed to complete the integration.

Introduction

In recent years, attempts have been made to understand the algorithmic principles of general intelligence. The first issue of the Cognitive Computation Journal reviews questions, including, e.g., if human like intelligence is achievable [McC09], 'howto' do the reverse engineering of the vertebrate brain [Gur09] among others that are relevant for artificial general intelligence (AGI) and neuroscience. Some of the AGI works started early and built upon theories of cognition, like SOAR (for a review, see [Lai09]) and Clarion (see, e.g., [Sun07] and references therein). Recent works consider universal searches [Hut09] and [Sch09] and build universal algorithms.

Another approach tries to limit the number of available algorithms and considers constraints [Lőr09b]: *directive soft constraints* come from neuroscience and cognition. These constraints are soft, because the interpretation of the findings is debated in many cases. Another type of constraints comes from mathematics. These are *hard constraints*. However, care should be taken as they may also have their own pitfalls hidden deeply in the conditions of the different theorems applied. The central question to these types of constraints is what algorithms can limit combinatorial explosion. In this paper we elaborate on the control issues of this approach.

The main contributions of the paper is the illustration of (i) how high-dimensional sensory information can be connected to autoregressive exogenous (ARX) processes via low-dimensional embedding, (ii) how one can estimate the inverse dynamics from the ARX model, and (iii) how one can connect the low-dimensional map to the event-learning framework [STL03] of reinforcement learning.

In the next section we review the basic components of the architecture. Then we highlight the working of some of the components through an illustrative example. Short discussion and conclusions about some experimental possibilities close the paper.

Background

We extend the architecture detailed in [Lőr09b]. We also consider missing components. We start from the observation that natural data – in many cases, but not always – exhibit heavy-tailed distributions. Such distributions may satisfy the conditions of Compressive Sensing (CS), a highly advantageous feature if part of the information is missing (see Compressive Sensing Resources `http://dsp.rice.edu/cs`).

Since heavy-tailed distribution is not warranted, the processing of sensory information in the architecture utilizes 'cross entropy' global probabilistic search [Rub97] to optimize overcomplete sparse representation using L_0 norm [LPS08]. Recent results on CS indicate (see, [DTDS07, NV07] and references therein) and numerical studies reinforce [PL09a] that certain versions of orthogonal matching pursuit complement and speed-up the cross entropy method.

One can alleviate the problem of combinatorial explosion by separating information of different kinds into (quasi-) independent lower-dimensional subspaces. As an example, facial speech and expressions can be represented in lower dimensions, respectively, by applying bilinear factorization [CDB02]. Since sparse codes and independent component analysis are related to each other [OF97], one option is that multi-linear extensions of Independent Process Analysis (IPA) [SL09a], similar to the multi-linear extension of Independent Component Analysis [VT05] may solve the separation problem. It is reassuring soft information from neuroscience that

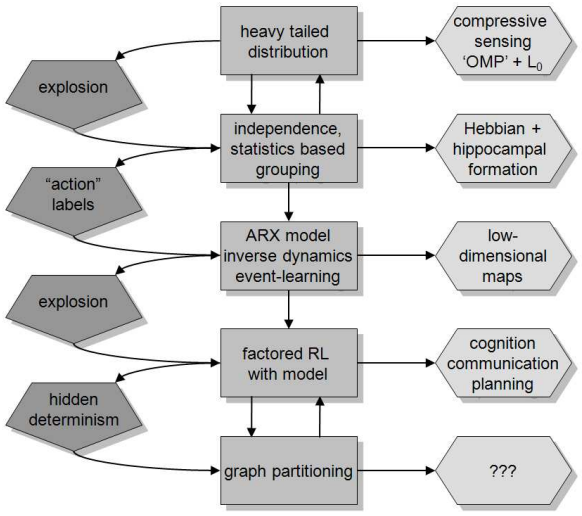


Figure 1: Squares: algorithmic components. Pentagons: emergent problems that need to be solved to proceed further. Hexagons: reassuring soft information from neuroscience and cognition, 'OMP': some version of orthogonal matching pursuit, L_0: probabilistic sparsification using L_0 norm.

the constraint of *Hebbian learning* on IPA gives rise to an architecture that closely resembles the entorhinal-hippocampal loop [Lőr09a, LS09a] that plays a central role in the encoding of episodic memory, or *events* in the brain.

There are a number of missing parts before this portion of the AGI architecture could be built, such as (i) the method to fuse information from different information sources (or modalities), (ii) the method to learn and embed the independent factors into different low-dimensional networks, and (iii) the method to extend the linear recurrent IPA network to non-linear ones (for a comprehensive lists on recurrent networks, see `http://www.idsia.ch/~juergen/rnn.html` and [LH09]).

From the point of view of controlling, IPA separates the relevant sensory information, such as position and direction [LS09a]. Furthermore, one can learn the effect of control – in an exogenous autoregressive (ARX) process – by optimal experimental design both for the observed [PL09b] and for the hidden variables [SL09a]. Inversion of the learned relations can be used as estimates of the inverse dynamics: it can derive control values for a desired state given the actual (experienced) one. This feature is advantageous for our architecture that applies robust controllers [LHS01].

Approximate inverse dynamics is a necessary presumption of *event-learning* [STL03] that works with a background controller and optimizes policy with regard to desired (planned) states. We note that event learning admits robust controllers.

Combinatorial explosion, however, spoils event-learning, unless factors and thus factored RL methods are available. It has been shown that (i) factored methods combined with sampling converge in polynomial time [SL08a], (ii) the exploration–exploitation dilemma can be overcome for optimistic initial models even for factored RL [SL09b], and (iii) the RL optimization remains polynomial.

We list two problems that need to be solved in this architecture:

Factor learning. It is unclear how to learn the factors in general. One suggestion considers factors as the primitives of symbols and the '*symbol learning problem*' as structure-noise separation of graphs [Lőr09b]. The corresponding graph partitioning problem is polynomial even for extreme graphs [Tao06]

Control of an under-controlled plant. The robust controller assumes an over-controlled plant (i.e., the number of controls is larger than the number of freedom). This condition may not be satisfied in general.

In what follows, we illustrate how to solve the last point. We shall start from high-dimensional space, will embed it into a low-dimensional one, but the control will remain undercomplete. We will use optimal control methods to overcome this obstacle.

Architecture for the under-controlled case

We present a simplified version of the architecture described in [Lőr09b]. We treat the following stages:

Sample selection: This stage selects samples under random control. Selected states are not too close to each other.

Low-dimensional embedding: In our illustration, the dimension of the low dimensional manifold is known, so we do not have to estimate it. Selected samples are embedded into the low dimensional space.

Identification of the ARX process: Out-of-sample estimations in the low-dimensional space are used for the identification of the ARX process.

LQR solution to the inverse dynamics: We have an under-controlled situation. We use a linear-quadratic regulator (LQR) to overcome this problem.

Exploring the space: Optimistic initial model (OIM) is used for exploring the space and for learning the values of different initial and final state-pairs, or *events*.

Now, we introduce the illustration.

Illustration with a pendulum

In the early phase of learning, the pendulum was subject to random control:

$$\ddot{\psi} = -\frac{g}{l} sin(\psi) - \frac{\gamma\dot{\psi}}{ml^2} + \frac{f}{ml^2}$$

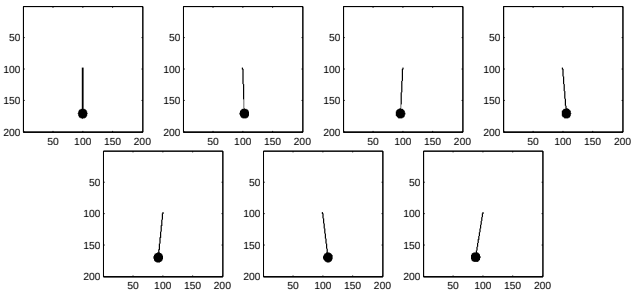


Figure 2: Selected sensors.

where ψ is the angle of the pendulum, and parameters were set as $g = 9.81\frac{m}{s^2}$, $m = 1kg$, $l = 100m$, and $\gamma = 500$. We limited the value of control f either to $100N$, or to $200N$ in our experiments.

Sample selection

We used 200×200 images of a pendulum (Fig. 2), so the input dimension was 40,000. In order to lower the dimension, we choose only a few samples (our sensors) from the input space. For similarity estimation, we used a Gaussian kernel: $K(x, y) = \exp\left(-\frac{||x-y||^2}{\sigma^2}\right)$ with $\sigma = 0,75$. For a set of sensors $S = \{s_1, \dots s_{|S|}\}$ the sensor space is determined by this kernel: for any input x, the sensed values form a vector in $R^{|S|}$

$$\phi_S(x) = (K(s_1, x), \dots, K(s_{|S|}, x))^T.$$

During sample generation we used $f^{rnd}_{max} = 100N$ and limited by time

Algorithm 1 Time limited sample generation

Given: duration of training T, initial angle ψ_0
$img_0 \leftarrow genPendImage(a_0)$
for $t = 0 \dots T-1$ **do**
  $f_t \leftarrow genRandControl()$
  $\ddot{\psi}_{t+1} \leftarrow simulatePendulum(f_t)$
  $img_{t+1} \leftarrow genPendImage(\ddot{\psi}_{t+1})$
end for

Algorithm 2 Similarity based sensor selection

$S \leftarrow \{img_0\}$
for $t = 1 \dots T$ **do**
  if $\forall s \in S : K(s, img_t) < 0.3$ **then**
    $S \leftarrow S \cup \{img_t\}$
  end if
end for

After time limited sensor selection, low dimensional embedding takes place.

Low-dimensional embedding and out-of-sample estimation

We used different input sets, including the set of simple views of the pendulum. We have tried a number of embedding algorithms and found that ISOMAP [BST+02] suits our input sets the best. On the selected data we applied the ISOMAP embedding algorithm with 3 nearest neighbors and generated a 1-dimensional embedding.

For inputs, which were not included into set S we approximated the corresponding 1-dimensional value by means of the partial least square (PLS) regression method (for a review and the extensions of the method, see [RK06]). The procedure is summarized below:

Algorithm 3 Embedding and out-of-sample estimation

{Data to be embedded:}
$A \leftarrow \emptyset$
for all $s \in S$ **do**
  $A \leftarrow A \cup \{\phi_S(s)\}$
end for

{Embedding:}
$G \leftarrow ISOMAP(A, 1D, 3\text{-NN})$
{Then $G \subset \mathbb{R}, G = \{g_a | a \in A\}$}

{Out-of-state estimations}
for $t = 0 \dots T$ **do**
  $\phi_t \leftarrow \phi_S(img_t)$
  $N \leftarrow SelectNearest(A, \phi_t, 3)$
  $\hat{x}_t \leftarrow PLS(N, \{g_a \in G | a \in N\}, \phi_t)$
end for

Results are shown in Fig. 3

Identification of the ARX process

The following equation of motion has been assumed

$$x_{t+1} = A_1 x_t + A_2 x_{t-1} + B u_t + \varepsilon_t \qquad (1)$$

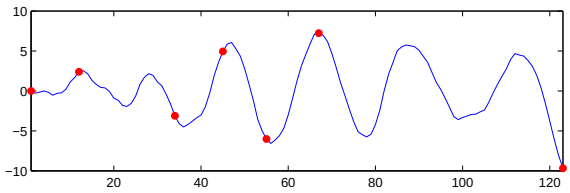
where $A_1, A_2, B \in \mathbb{R}$, ε is a normally distributed stochastic variable with 0 mean and covariance σ. Note, however, that this assumption may not hold; it is subject to our estimation errors, including the error of the PLS estimation. We have estimated parameters B, A_1 and A_2 with a least squares estimation derived from the cost function:

$$J(A_1, A_2, B) = \sum_{t=1}^{T} (\hat{x}_{t+1} - A_1 \hat{x}_t - A_2 s_{t-1} - B u_t)^2$$

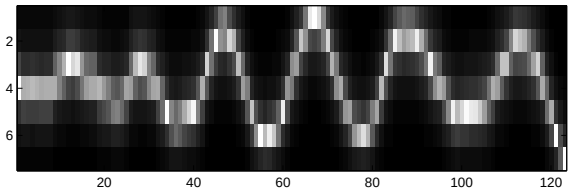
where $\hat{x}$ denotes the PLS estimated coordinates. We also tried an on-line Bayesian method that gave similar results.

LQR solution to the inverse dynamics

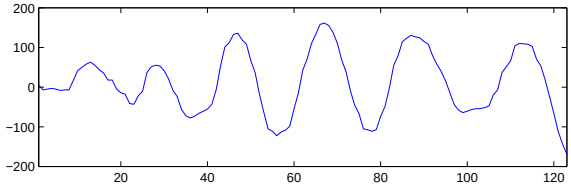
We define the control problem by means of *event learning* RL algorithm [STL03] that provides the actual and



(a) Motion of the pendulum as a function of time in angle space between $\pm 10\,^\circ$. Motion is subject to random control. Red dots: positions of selected sensors



(b) Responses in sensor space. The outputs of the sensors are shown vs. time. Sensors are ordered according to their 1-dimensional embeddings.



(c) PLS position estimation in the embedded 1-dimensional space during the motion of the pendulum

Figure 3: Motion of the pendulum in real space, in sensor space and in embedded 1-dimensional space.

desired states to a backing controller. The controller tries to satisfy 'desires'. For given experienced state and desired state pair the controller provides a control value or a control series. Then, event learning learns the limitations of the backing controller and optimizes the RL policy in the event space accordingly.

In the present scenario, we have a 1-dimensional plant of second order, so the state of the plant is determined by its position and its speed, or alternatively, by its present and previous positions. The same holds for the desired state, which thus has two degrees of freedom. For our 1D control problem, 2 time steps are needed to reach the desired state, if it is possible at all. If it is, then there are many solutions, out of which an optimal control can help to choose: one assumes costs associated with the trajectory and the value of the control. Denoting the state by $q_t = (x_t, x_{t-1})^T$, we can convert Eq. 1 into the following form $q_{t+1} = Fq_t + Cv_t + \epsilon_t$, where

$$F = \begin{bmatrix} A_1 & A_2 \\ 0 & I \end{bmatrix} \tag{2}$$

$C = (B, 0)^T$, $v_t = (u_t, 0)^T$, and $\epsilon_t = (\varepsilon_t, 0)^T$.

For N-step experienced state q_i and desired state q_i^d $(i = 1, \dots N)$ one may use quadratic cost functions both for the $q_i - q_i^d$ differences and for the control values using the cost function

$$\begin{aligned} J(U) &= \sum_{i=0}^{N} (q_i - q_i^d)^T Q_i (q_i - q_i^d) \\ &\quad + \sum_{i=0}^{N-1} v_i^T R_i v_i \end{aligned} \tag{3}$$

where $Q_i = Q_i^T \geq 0$ and similarly, $R_i = R_i^T \geq 0$. Then,

$$\begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} = H \begin{bmatrix} v_0 \\ \vdots \\ v_{N-1} \end{bmatrix} + \begin{bmatrix} I \\ \vdots \\ F^N \end{bmatrix} q_0$$

where

$$H = \begin{bmatrix} 0 & \dots & & \\ C & 0 & \dots & \\ FC & C & 0 & \dots \\ \vdots & \vdots & & \\ F^{N-1}C & F^{N-2}C & \dots & C \end{bmatrix}$$

that we can write in the following short form: $\mathbf{q} = H\mathbf{v} + Gq_0$, where $\mathbf{q} = (q_0, \dots, q_N)^T$, $\mathbf{v} = (v_0, \dots, v_N)^T$, and $G = (I, \dots F^N)^T$. Now, the cost function can be rewritten into the following quadratic form

$$\begin{aligned} J(U) &= ||diag(Q_0^{\frac{1}{2}}, \dots, Q_N^{\frac{1}{2}})(H\mathbf{v} + Gq_0 - \mathbf{q})||^2 \\ &+ ||diag(R_0^{\frac{1}{2}}, \dots, R_{N-1}^{\frac{1}{2}})\mathbf{v}||^2 \end{aligned} \tag{4}$$

that can be solved by simple routines. In our case, $Q_0, \dots Q_{N-2} = 0$, $Q_{N-1}, Q_N = 1$, and $R_0 \dots R_{N-1} = R$.

Now, we turn to the optimization of the RL problem.

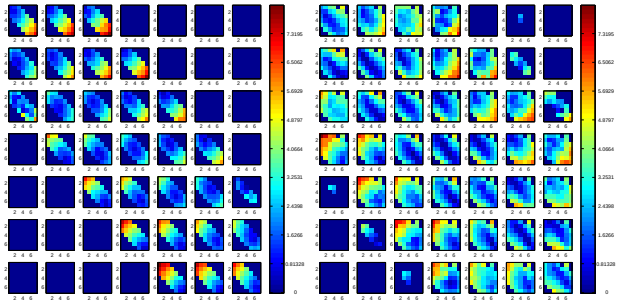
Exploring the space: experienced transitions The estimated value of an event $E_{i,j}^\pi$ in event learning is the estimated long-term cumulated discounted reward of event $e = (q_i, q_j^+)$ in a Markov decision process under fixed policy $\pi = \pi(q, q^+)$, with actual state q and desired state q^+. The plus sign indicates that the second argument of event e is the successor state of the state represented by the first argument. We note that the OIM policy [SL08b] is not fixed, but it converges to the optimal policy. Here we use OIM for the exploration of the space. OIM, however, could be used for diverse tasks, e.g., for inverting the pendulum, like in [STL03].

We used $f_{max}^{OIM} = 200N$ in this case to allow OIM to connect a larger number of states. Experimental results are shown in Figs. 4 and 5.

Discussion and conclusions

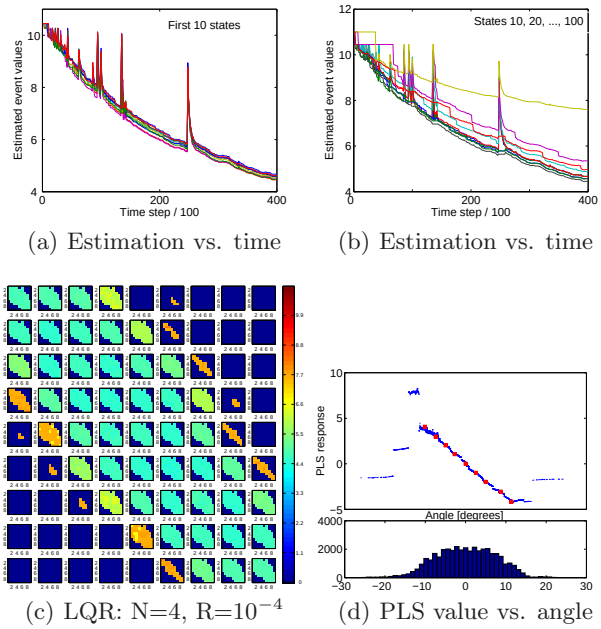
The presented toy problem and its solution highlight the following issues relevant to AGI

- Factors like position, speed, etc. can be learned by neural mechanisms [LS09b]. On the other hand, we do not know how to learn factors in general. Multilinear IPA might work for some cases.



(a) LQR: N=2, R=0.01 (b) LQR: N=4, R=10^{-4}

Figure 4: Color coded error between desired and experienced states using OIM. Each block represents one of the $7 \times 7 = 49$ actual states. The horizontal and vertical indices of the subfigures represent the number of the actual position and the number of the previous position, respectively. Each square within the each block represents a desired state and is also indexed according to the actual and the previous positions. The warmer the color the larger the error. Transitions corresponding to large blue areas have not occurred during the course of the experiment. The larger the number of time steps used by LQR and the smaller the cost of the control, the more states can be reached. OIM addresses the exploration-exploitation dilemma by estimating the transition probabilities and thus enabling planning.



(a) Estimation vs. time (b) Estimation vs. time

(c) LQR: N=4, R=10^{-4} (d) PLS value vs. angle

Figure 5: Convergence of event values $E_{i,j}$ (see text). (a): Convergence of the OIM estimation for the first 10 events. (b): Convergence of every $10^t h$ event from the first 100 events. (c): Estimated values of all found events 40,000 steps. The warmer the color, the larger the OIM estimated value. Transitions corresponding to large blue areas have not occurred until the experiment was finished. (d): Out-of-sample estimation vs. pendulum angle and histogram of positions in arbitrary units after 40,000 steps. Spikes and sudden drops correspond to the discoveries of new states and to visits to known states, respectively.

- We suspect factors correspond to a highly simplified directed bipartite graph derived from a highly complex graph connecting the products of observed actual and desired states to the next state. Weights of this graph represent the transition probabilities. According to the conjecture [Lőr09b], one can use the separated structured part of the graph. The vertices of this graph make the factors. The rest of the original graph is made of noisy Erdős-Rényi-like blocks and the structured part saves the transition probabilities. It has been shown that such compression is possible even for extreme graphs [Tao06].
- We used low-dimensional embedding – suggested by neuronal control systems, like the 'gelatinous medium' surrounding the limb [GG93] – in order to transform high-dimensional information for the sake of lower dimensional and possibly robust control.
- Optimal control can serve the optimization of under-controlled plants. Beyond the presented LQR method, which assumes no noise, extensions to observations spoiled by Gaussian noise are available [SL04]. Further extensions to stochastic optimal control are desired.
- ARX based *optimal control*, such as LQR control, has feedforward characteristics. On the other hand, *robust control* works by error correction. *Experienced events* that emerge through the application of these controllers or their combinations require further optimization using *RL based optimal control*.

From the point of view of neuroscience, the roles of the two control related networks, the cerebellum and the basal ganglia are controversial [Bas06, YBK09]. We suspect that this controversy arises from the fact that optimal control appears in two ways: once for feedforward control of plants, and once for optimal decision making. We conjecture that one might gain more insight into the functioning of these neural architectures by comparing their roles in over-controlled and under-controlled control tasks.

Acknowledgements

We are most grateful to Zoltán Szabó for helpful discussions and to Gábor Szirtes for careful reading of the manuscript. This research has been partially supported by the EC NEST 'Percept' grant under contract 043261. Opinions and errors in this manuscript are the author's responsibility, they do not necessarily reflect the opinions of the EC or other project members.

References

[Bas06] A. J. Bastian. Learning to predict the future: the cerebellum adapts feedforward movement control. *Current Opinion in Neurobiology*, 16:645–649, 2006.

[BST+02] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford. The ISOMAP algorithm and topological stability. *Science*, 290:2319–2323, 2002.

[CDB02] E. S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *Proc. Pacific Conf. Comp. Graphics Appl.*, 2002. `http://cims.nyu.edu/~bregler/pubs.html`.

[DTDS07] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Technical Report, 2007. `http://stat.stanford.edu/~idrori/StOMP.pdf`.

[GG93] M. S. Graziano and C. G. Gross. A bimodal map of space: somatosensory receptive fields in the macaque putamen with corresponding visual receptive fields. *Experimental Brain Research*, 97:96–109, 1993.

[Gur09] K. N. Gurney. Reverse engineering the vertebrate brain. *Cogn. Comput.*, 1:29–41, 2009.

[Hut09] M. Hutter. Feature reinforcement learning: Part I. Unstructured MDPs. *J. of AGI*, 1:3–24, 2009.

[Lai09] J. Laird. Soar cognitive architecture tutorial. In *2nd Conf. on Artificial General Intelligence*, AGI-2009, 2009. `http://agi-conf.org/2009/slides/`.

[LH09] M. Lukosevicius and J. Herbert. Reservoir computing approaches to recurrent neural network training. *Comp. Sci. Rev.*, 3:127–149, 2009.

[LHS01] A. Lőrincz, Gy. Hévízi, and Cs. Szepesvári. Ockham's razor modeling of the matrisome channels of the basal ganglia thalamocortical loop. *Int. J. of Neural Syst.*, 11:125–143, 2001.

[Lőr09a] A. Lőrincz. Hebbian constraint on the resolution of the homunculus fallacy leads to a network that searches for hidden cause-effect relationships. In *AGI-2009*, volume 8 of *Adv. in Intell. Syst. Res. (ISSN: 1951-6851)*, pages 126–131. Atlantis Press, 2009.

[Lőr09b] A. Lőrincz. Learning and representation: From compressive sampling to the 'symbol learning problem'. In *Handbook of Large-Scale Random Networks*, pages 445–488. Springer, Germany, 2009.

[LPS08] A. Lőrincz, Zs. Palotai, and G. Szirtes. Spike-based cross-entropy method for reconstruction. *Neurocomputing*, 71:3635–3639, 2008.

[LS09a] A. Lőrincz and G. Szirtes. Here and now: how time segments may become events in the hippocampus. *Neural Networks*, 22:738–747, 2009.

[LS09b] A. Lőrincz and G. Szirtes. Representation theory meets anatomy: Factor learning in the hippocampal formation. In *Connectionist Models of Behaviour and Cognition II*, volume 18, pages 253–264, Singapore, 2009. World Scientific.

[McC09] J. L. McClelland. Is a machine realization of truly human-like intelligence achievable? *Cogn. Comput.*, 1:17–21, 2009.

[NV07] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. 2007. http://arxiv.org/abs/0712.1360.

[OF97] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[PL09a] Zs. Palotai and A. Lőrincz. OMP speeds up CE in L_0 norm optimizations. unpublished, 2009.

[PL09b] B. Póczos and A. Lőrincz. Identification of recurrent neural networks by Bayesian interrogation techniques. *J. of Mach. Learn. Res.*, 10:515–554, 2009.

[RK06] R. Rosipal and N. Kramer. *Subspace, Latent Structure and Feature Selection Techniques*, chapter Overview and Recent Advances in Partial Least Squares, pages 34–51. Springer, 2006.

[Rub97] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112, 1997.

[Sch09] J. Schmidhuber. Ultimate cognition á la Gödel. *Cogn. Comput.*, 1:177–193, 2009.

[SL04] I. Szita and A. Lőrincz. Kalman filter control embedded into the reinforcement learning framework. *Neural Comp.*, 16:491–499, 2004.

[SL08a] I. Szita and A. Lőrincz. Factored value iteration converges. *Acta Cyb.*, 18:615–635, 2008.

[SL08b] I. Szita and A. Lőrincz. The many faces of optimism: a unifying approach. In *ICML 2008*, pages 1048–1055, Helsinki, 2008. Omnipress.

[SL09a] Z. Szabó and A. Lőrincz. Controlled complete ARMA independent process analysis. In *Int. Joint Conf. on Neural Networks (IJCNN 2009)*, pages 3038–3045, 14-19 June 2009. ISSN: 1098-7576.

[SL09b] I. Szita and A. Lőrincz. Optimistic initialization and greediness lead to polynomial time learning in factored MDPs. In *ICML 2009*, Montreal, 2009. Omnipress.

[STL03] I. Szita, B. Takács, and A. Lőrincz. Epsilon-MDPs: Learning in varying environments. *J. of Mach. Learn. Res.*, 3:145–174, 2003.

[Sun07] R. Sun. The importance of cognitive architectures: An analysis based on CLARION. *J. of Exp. and Theor. Artif. Intell.*, 19:159–193, 2007.

[Tao06] T. Tao. Szemerédi's regularity lemma revisited. *Contrib. Discrete Math.*, 1:8–28, 2006.

[VT05] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In *Proc. Comp. Vision and Pattern Recog.*, 2005.

[YBK09] K. Yarrow, P. Brown, and J. W. Krakauer. Inside the brain of an elite athlete. *Nature Reviews Neuroscience*, 10:585–596, 2009.