ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 531-536

Neural Dynamics for Task-Oriented Grouping of Communicating Agents

Jochen J. Steil

Neuroinformatics Group, University of Bielefeld, Germany
jsteil@techfak.uni-bielefeld.de, www.jsteil.de

Abstract. Many real world problems are given in the form of multiple measurements comprising local descriptions or tasks. We propose that a dynamical organization of a population of communicating agents into groups oriented towards locally similar clusters of subtasks can identify higher level structure and solve such tasks. We assume that an agent may compute the compatibility of its resources with the input descriptions and that it can compare this compatibility with that of other agents. Based on dynamically updated soft assignment variables each agent computes its action preference distribution and communicates it to other agents. Applying theory developed for the competitive-layer model (CLM, Wersing, Steil, Ritter, Neural Computation 13, 357-387, 2001), a recurrent linear threshold network for feature binding and sensory segmentation, we give constructive conditions on the choice of the agents' compatibility functions and dynamical parameters to assure convergence. They guarantee that each agent unambiguously either decides for an action or not to be active at all. We give an approximative stochastic algorithm to sample the decision dynamics and discuss one realized and one proposed example.

1 Introduction

Many of the hard problems in artificial intelligence are of the type to generate or detect global structure of the problem based on local data and descriptions only. Typical examples are found in image processing, where the data are pixel values and the task is to segment and group them to perceptually consistent entities like edges or objects. On a different scale but in the same context it has been proposed that cooperation and self-organization of autonomous agents in solving large-scale optimization problems Parsopoulos and Vrahatis [2002] provide means for emergent computation of otherwise intractable tasks. Thus, in the context of emergent computation, multi-agent systems are found on many levels reaching from the micro-biology up to neural networks, software-agents, or whole Internet-based machines Organic Computing.

In practice many such approaches lack a mathematical treatment of their capabilities to dynamically reach a task oriented and efficient configuration nor can any kind of optimality be achieved. Below we use theory developed for the competitive-layer model (CLM, Wersing, Steil, Ritter, Neural Computation 13, 357-387, 2001), a recurrent linear threshold network for feature binding and sensory segmentation to

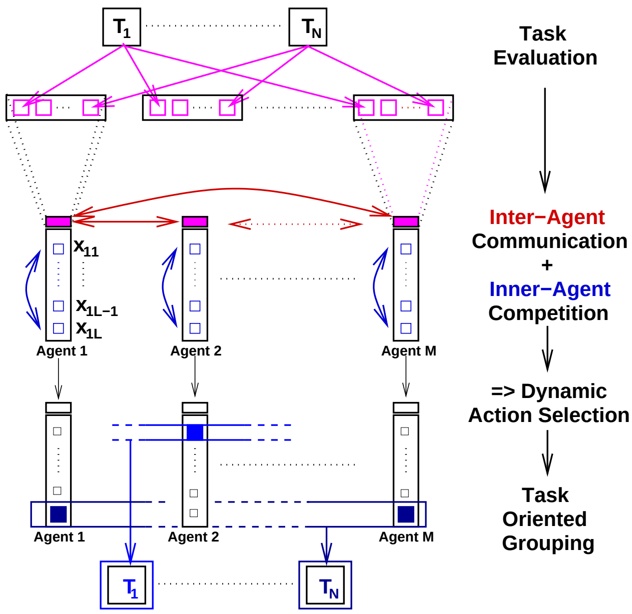

Figure 1: Agents A_r communicate their action preference distribution variables x_{rl} to others to check compatibility for forming a cooperating group.

give constructive conditions on the choice of the agents' compatibility functions and dynamical parameters such that a task oriented grouping of cooperating agent can be guaranteed. It can even be shown that the resulting coalitions of agents are optimal with respect to an energy function incorporating their mutual compatibilities and their grade of specialization with respect to the given tasks.

The architecture is shown in Fig. 1 and consists of three elements: the task evaluation, the organization of inter-agent communication and inner-agent action winner-take-all competition between the actions, and the action selection dynamics.

2 Task evaluation

Our architecture solves task assignment problems using a paradigm of cooperating agents. We assume that each agent has to be assigned to one of a number of N tasks and can chose between L actions, however, with respect to mutual compatibility with other agents and their choices. This framework has been inspired by the architecture of the CLM neural network for perceptual grouping task, where – reinterpreted in the framework presented here – every agents has a fixed assignment to a local feature vector describing the task and the action is to signal presence or absence of a group.

Let N tasks described by feature vectors T_n be given to which R agents A_r have to be assigned. Each agent A_r computes its task evaluation vector

$$t_r = [\mu_r(T_1), ..., \mu_r(T_N)]$$

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 531-536

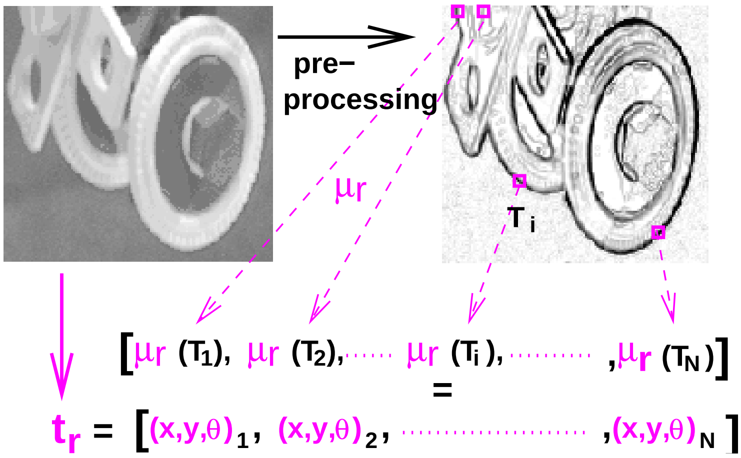

Figure 2: Description of problem is feature vector of positions and filter answers.

which summarizes the agents information about the given tasks. In perceptual grouping for images segmentation typically T_n is a local feature vector for each pixel n. If we further assume that R=N, i.e. there are as many agents as pixels, and $\mu_r = \delta_{rn}$, then each agent is specialized to a certain pixel and all non-local information has to be conveyed by the communication between agents. A corresponding example for contour grouping has been realized and is shown in Fig. 2, where the partial description of the problem is the feature vector of positions and edge filter answers.

We also propose a more general example of CPU assignment, where each agent A_r may represent a CPU with certain resources, to which one or more tasks T_N can be assigned. Then $\mu_r(T_N)$ can encode to which degree the task and the resources match, consider e.g.. needs for memory, cache, processor speed etc. Details on this problems yet have to be specified.

3 Competition and Communication

The main idea of our framework is to organize for each agent A_r a dynamic decision process for one of L actions. The degree of certainty with which A_r decides for action l is expressed by positive decision variables $x_{rl} \geq 0$, $\sum_l x_{rl} = 1$, which together for the action preference distribution of agent A_r. An unambiguous decision requires that $x_{r\hat{l}} = 1$, $x_{rl'} = 0$. Two mechanisms influence the dynamic development of $x_{r\alpha}$: first, an inner-agent competition between possible actions and, second, the evaluation of the compatibility of the current action decision preference of agent A_r with the decision preference of other agents $A_{r'}$ by means of communicating their current states. The latter introduces dynamically propagated knowledge about the overall global state of the agent community.

Inner-agent competition: competition between actions of A_r is organized as inner-agent winner-take-all WTA circuit which uses mutual symmetric inhibitory interactions with strength $I_r^{ll'} = I_r^{l'l} > 0$ between the assignment variables x_{rl} and $x_{rl'}$,

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 531-536

$x_{rl} \geq 0$ and the dynamically enforced (see (2) below) constraint $\sum_l x_{rl} = 1$. The larger $I_r^{ll'}$ the stronger is the mutual exclusion between actions l and l' and the faster is the decision dynamics with respect to these two actions. The self-interaction I_r^{ll} can further encode an a-priori preference distribution for the actions of agent A_r.

Inter-agent communication: mutual influence between agents is evaluated with respect to what degree the decision of agent A_r is compatible to the decisions of agents $A_{r'}$. To this aim, for each agent A_r and each action l we assume a compatibility with agent $A_{r'}$ deciding for the same action with respect to the given tasks,

$$f^l_{rr'} = f^l_{rr'}(t_r, t_{r'})$$

where $f^l_{rr'} > 0$ if the decision of $A_{r'}$ for l supports an decision of A_r for action l and vice versa. We additionally assume a small constant communication cost $-k$ against which the support is weighted.

At each time step all agents may communicate their internal states x_{rl} and collect their total support/inhibition for actions l:

$$F^l_r = \sum_{r'} \left(f^l_{rr'}(t_r, t_{r'}) - k\right) x_{r'l} \tag{1}$$

The function f has to be specified problem specific. For grouping tasks, it typically does not depend on the action l because all actions just signal presence of a group and the segmentation of the input into groups is achieved by coherent activation of groups of the feature dependent agents. For contour grouping as shown in Fig. 2 we have

$$f^l_{rr'} = f_{rr'} = f(t_r, t'_r) = f((x, y, \theta)_r, (x, y, \theta)_{r'})$$

because $\mu_r((x, y, \theta)_n) = \delta_{rn}(x, y, \theta)_n)$ always selects the feature vector of pixel r for the pixel selective agent A_r. For CPU assignment, $f^l_{rr'}$ could, for instance, encode how fast two machines are connected in the network. If fast network is important for the task the weight must be positive if the machines are connected and strongly negative otherwise.

In principle, the problem specific compatibility function f must be specified by the system designer. Methods to generate suitable f from labeled examples for grouping tasks have been studied in Wersing [2001], Weng and Steil [2003], but the derivation similar learning methods for the agent framework is beyond the scope of this paper.

4 Dynamic Action Selection

Next we formulate the **action assignment dynamics**, which incorporates both the local WTA-competition and the collection of support from other agents as

$$\dot{x}_{rl} = J(1 - \sum_l x_{rl}) + \sum_{r'} \left(f^l_{rr'}(t_r, t_{r'}) - k\right) x_{r'l}. \tag{2}$$

The constant J weights the competitive process against the influence of the support. If we perform a synchronous update of all variables, the following specialized version of Theorem 1 in Wersing et al. [2001] shows that a careful choice of J with respect to f results in an unambiguous dynamic action selection for all agents.

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 531-536

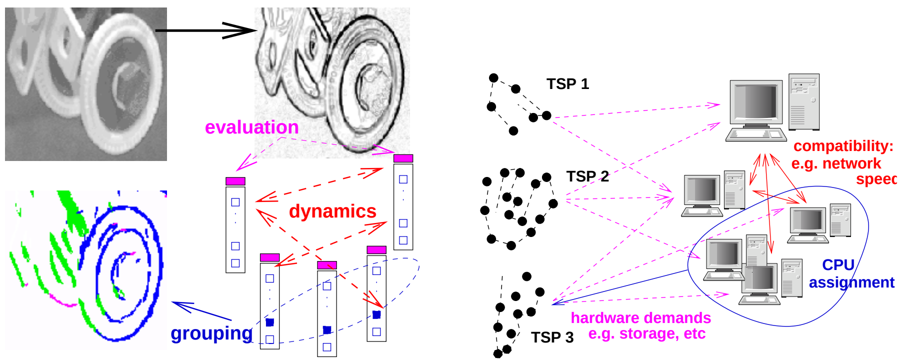

Figure 3: Realized contour segmentation (left) by grouping edges based on symmetry–proximity evaluation of position and orientation. Proposed application to CPU assignment (right) evaluates e.g. compatibility with local network.

Theorem 1 *If $k_{rl} > 0$ with $k_{rl} = I_r^{ll} - f_{rr}^l - \sum_{r' \neq r} \max(0, f_l^{rr'})$, then the dynamics (2) is bounded. If further $f_{rr}^l > 0$ for all r, l, and the vertical interactions satisfy $I_r^{ll} I_r^{l'l'} \leq (I_r^{ll'})^2$ for all l, l' then an attractor of the dynamics (2) has for each agent A_r either*

i) at most one positive decision variable $x_{r\hat{l}}$ with
$x_{r\hat{l}} = \frac{J + F_{r\hat{l}}}{I_r^{\hat{l}\hat{l}}}$, and $x_{rl'} = 0$ for all $l' \neq \hat{l}$, where $\hat{l} = \hat{l}(r)$ is the index of the maximally supported action characterized by $F_{r\hat{l}} > F_{rl'}$ for all $l' \neq \hat{l}$, or

ii) all decision variables x_{rl}, $l = 1, \ldots, L$ for agent A_r vanish.

The theorem guarantees that each attractor of the action selection dynamics leads to an unambiguous decision of all agents for one of the possible actions or to execute non of them ($x_{rl} = 0$) because no action is compatible with the other agents decisions. By choosing suitable $I_{ll'}$ and J we can always ensure convergence to one of these stable states.

5 Simulation and Application

The convergence result of Theorem 1 holds for the system of differential equations 2 when integrated synchronously in continuous time. For a true agent system, however, the updates must be local and asynchronous. As well it is unrealistic to carry out the full sum over the mutual compatibilities to compute the support (1). For practical application, we have to assume a discrete and stochastic order of updates based on partial information obtained by restricted communication only which closely approximates the theoretically stable dynamics.

A suitable approximation for simulation is given by the following algorithm: [1]:
`Initialize all` $x_{rl}(t = 0) \in [1/L - \epsilon, 1/L + \epsilon]$,

[1] The algorithm was proposed by S. Weng in personal communication

```
Initialize J := 1 + max_{r,r',l} f^l_{rr'}
Choose (r, r', l) at random and compute
ξ := (J (1 − Σ_{l'≠l} I_r^{ll'} x_{rl'}) + f^l_{rr'} x_{r'l}) /(1 − f^l_{rr})
update x_{rl} := max(0, ξ) until convergence
```

From the point of view of the agent A_r, this algorithm computes the compatibility (its support F_r^l) based on receiving $x'_r l$ from a single other agent $A_{r'}$ then to update the internal WTA circuit directly. In particular, the sum over all r' in the differential equation 2 is reduced to a single term and subsequently the right hand side is solved for x_{rl} under the assumption $\dot{x}_{rl} = 0$. For carrying out the full support summation, this asynchronous dynamics converges due to a convergence result on asynchronous iteration in neural networks by Feng [1997]. The approximate iterative employment of partial support turned out to work very well in practice in simulations, where we tried to reproduce segmentation results obtained with the original neural CLM dynamics. The outcome for the grouping example discussed above is shown in Fig. 3.

6 Conclusion

We present a flexible framework for the dynamic formation of cooperating groups of agents to solve task assignments problems. It is based on two combined processes, an inner-agent winner-takes-all competition between different actions and an inter-agent communication, which allows the competition to be influenced by the tendency of other agents to cooperate by means of selecting the same action. We reinterprete and generalize the competitive layer model (CLM) to organize these dynamics and apply theoretical results from Wersing et al. [2001] to give conditions for convergence to a unique action selection for each agent. A stochastic asynchronous approximation of this dynamics allows the local and parallel computation necessary for a true agent system. Potential applications are widespread including perceptual grouping tasks (which have been realized and to which also the CLM has successfully been applied) and more general and symbolically specified problems as long as the task compatibility vector μ_r and a suitable agent compatibility function $f^l_{rr'}$ can be specified. Ongoing work aims at a concrete application in the CPU assignment domain.

References

J. Feng. Lyapunov functions for neural nets with nondifferentiable input-output characteristics. *Neural Computation*, 9(1):43–49, 1997.

Organic Computing. Organic computing. URL. http://www.organic-computing.org.

K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2–3):235–306, 2002.

S. Weng and J. Steil. Learning compatibitlity functions for feature binding and sensory segmentation. In *Proc. ICANN*, 2003.

H. Wersing. Learning lateral interactions for feature binding and sensory segmentation. In *NIPS*, pages 1009–1016, 2001.

H. Wersing, J. J. Steil, and H. Ritter. A competitive layer model for feature binding and sensory segmentation. *Neural Computation*, 13(2):357–387, 2001.