Controlling Combinatorial Explosion in Inference via Synergy with Nonlinear-Dynamical Attention Allocation

Ben Goertzel[1,2(✉)], Misgana Bayetta Belachew[1,2,4], Matthew Ikle'[3], and Gino Yu[4]

[1] OpenCog Foundation, Hong Kong, China
ben@goertzel.org
[2] Hanson Robotics, Hong Kong, China
[3] Adams State University, Alamosa, USA
[4] School of Design, Hong Kong Poly U, Hong Kong, China

Abstract. One of the core principles of the OpenCog AGI design, "cognitive synergy", is exemplified by the synergy between logical reasoning and attention allocation. This synergy centers on a feedback in which nonlinear-dynamical attention-spreading guides logical inference control, and inference directs attention to surprising new conclusions it has created. In this paper we report computational experiments in which this synergy is demonstrated in practice, in the context of a very simple logical inference problem.

More specifically: First-order probabilistic inference generates conclusions, and its inference steps are pruned via "Short Term importance" (STI) attention values associated to the logical Atoms it manipulates. As inference generates conclusions, information theory is used to assess the surprisingness value of these conclusions, and the "short term importance" attention values of the Atoms representing the conclusions are updated accordingly. The result of this feedback is that meaningful conclusions are drawn after many fewer inference steps than would be the case without the introduction of attention allocation dynamics and feedback therewith.

This simple example demonstrates a cognitive dynamic that is hypothesized to be very broadly valuable for general intelligence.

1 Introduction

One approach to creating AGI systems is the "integrative" strategy, involving combining multiple components embodying different structures or algorithms, and relying on synergistic dynamics between components. One kind of integrative system involves various highly independent software components, each solving a specialized set of problems in a mostly standalone manner, with occasional communication between each other in order to exchange problems and solutions. On the other end of the scale, are systems designed as tightly interconnected components that give rise to complex non-linear dynamical phenomena. Here, we are specifically focused on the latter approach. We will discuss the

B. Steunebrink et al. (Eds.): AGI 2016, LNAI 9782, pp. 334–343, 2016.
DOI: 10.1007/978-3-319-41649-6_34

particulars of one form of cognitive synergy – between probabilistic inference and nonlinear-dynamical attention allocation – within the context of one particular integrative AGI architecture, PrimeAGI (formerly named OpenCogPrime) [9,10], implemented on the OpenCog platform [7].

The specific nature of the synergy explored and demonstrated here is as follows:

- Probabilistic logical reasoning, proceeding via forward chaining and using the PLN (Probabilistic Logic Networks) rule-base, chooses premises for its inferences based on weights called STI (ShortTermImportance) values
- When the inference process discovers something sufficiently surprising (via an information-theoretic measure), it boosts the STI value associated with this discovery
- STI values spread according to a particular set of equations modeled on economics (ECAN, Economic Attention Allocation), so that when an item or fragment of knowledge has a high STI, related entities will also get their STI boosted

Thus, broadly speaking, we have a feedback in which

- importance values prune the combinatorial explosion of inference chaining
- inferentially determined surprisingness guides importance assessments
- importance spreads among related entities

According to this dynamic, entities related to other entities that have been useful for an inference process, will also tend to get brought to the attention to that inference process. This will cause the inference process to focus, much more often than would otherwise be the case, on sets of interrelated knowledge items regarding which there are surprising (interesting) conclusions to be drawn. It is a form of deliberative thinking in which conclusion-drawing and attention-focusing interact synergetically.

This sort of dynamic is very general in nature and, according to the conceptual theory underlying PrimeAGI, is critical to the operation of probabilistic inference based general intelligence. Here we illustrate the synergy via a simple "toy" example, which highlights the nature of the feedback involved very clearly. Our current work involves leveraging the same synergy in more realistic cases, e.g. to help a system maintain focus in the course of inference-guided natural language dialogue.

2 Background: PrimeAGI

Our work here is based upon specific details of the AGI architecture called **PrimeAGI** (formerly known as OpenCogPrime), based on the open-source OpenCog project at http://opencog.org. PrimeAGI is a large and complex system whose detailed description occupies two volumes [9,10].

The concept of cognitive synergy is at the core of the PrimeAGI design, with highly interdependent subsystems responsible for inference regarding patterns obtained from visual, auditory and abstract domains, uncertain reasoning, language comprehension and generation, concept formation, and action planning. The goal of the PrimeAGI project is to engineer systems that exhibit general intelligence equivalent to a human adult, and ultimately beyond.

The dynamics of interaction between processes in PrimeAGI is designed in such a way that knowledge can be converted between different types of memory; and when a learning process that is largely concerned with a particular type of memory encounters a situation where the rate of learning is very slow, it can proceed to convert some of the relevant knowledge into a representation for a different type of memory to overcome the issue, demonstrating **cognitive synergy**. The simple case of synergy between ECAN and PLN explored here is an instance of this broad concept; PLN being concerned mainly with declarative memory and ECAN mainly with attentional memory.

2.1 Memory Types and Cognitive Processes in PrimeAGI

PrimeAGI's memory types are the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience [14], plus attentional memory for allocating system resources generically, and intentional memory for allocating system resources in a goal-directed way. Table 1 overviews these memory types, giving key references and indicating the corresponding cognitive processes, and which of the generic patternist cognitive dynamics each cognitive process corresponds to (pattern creation, association, etc.).

The essence of the PrimeAGI design lies in the way the structures and processes associated with each type of memory are designed to work together in a closely coupled way, the operative hypothesis being that this will yield cooperative intelligence going beyond what could be achieved by an architecture merely containing the same structures and processes in separate "black boxes."

The inter-cognitive-process interactions in OpenCog are designed so that

- conversion between different types of memory is possible, though sometimes computationally costly (e.g. an item of declarative knowledge may with some effort be interpreted procedurally or episodically, etc.)
- when a learning process concerned centrally with one type of memory encounters a situation where it learns very slowly, it can often resolve the issue by converting some of the relevant knowledge into a different type of memory: i.e. **cognitive synergy**

The simple case of ECAN/PLN synergy described here is an instance of this broad concept.

2.2 Probabilistic Logic Networks

PLN serves as the probabilistic reasoning system within OpenCog's more general artificial general intelligence framework. PLN logical inferences take the form of

Table 1. Memory Types and Cognitive Processes in OpenCog Prime. The third column indicates the general cognitive function that each specific cognitive process carries out, according to the patternist theory of cognition.

Memory Type	Specific Cognitive Processes	General Cognitive Functions
Declarative	Probabilistic Logic Networks (PLN) [6]; concept blending [5]	pattern creation
Procedural	MOSES (a novel probabilistic evolutionary program learning algorithm) [13]	pattern creation
Episodic	internal simulation engine [8]	association, pattern creation
Attentional	Economic Attention Networks (ECAN) [11]	association, credit assignment
Intentional	probabilistic goal hierarchy refined by PLN and ECAN, structured according to MicroPsi [2]	credit assignment, pattern creation
Sensory	In OpenCogBot, this will be supplied by the DeSTIN component	association, attention allocation, pattern creation, credit assignment

syllogistic rules, which give patterns for combining statements with matching terms. Related to each rule is a truth-value formula which calculates the truth value resulting from application of the rule. PLN uses forward-chaining and backward-chaining processes to combine the various rules and create inferences.

2.3 Economic Attention Networks

The attention allocation system within OpenCog is handled by the Economic Attention Network (ECAN). ECAN is a graph of untyped nodes and links and links that may be typed either HebbianLink or InverseHebbianLink. Each Atom in an ECAN is weighted with two numbers, called STI (short-term importance) and LTI (long-term importance), while each Hebbian or InverseHebbian link is weighted with a probability value. A system of equations, based upon an economic metaphor of STI and LTI values as artificial currencies, governs importance value updating. These equations serve to spread importance to and from various atoms within the system, based upon the importance of their roles in performing actions related to the system's goals.

An important concept with ECAN is the attentional focus, consisting of those atoms deemed most important for the system to achieve its goals at a particular instant. Through the attentional focus, one key role of ECAN is to guide the forward and backward chaining processes of PLN inference. Quite simply, when PLN's chaining processes need to choose logical terms or relations to include in their inferences, they can show priority to those occurring in the system's attentional focus (due to having been placed their by ECAN). Conversely, when terms or relations have proved useful to PLN, they can have their importance boosted, which will affect ECAN's dynamics. This is a specific example of the cognitive synergy principle at the heart of the PrimeAGI design.

3 Evaluating PLN on a Standard MLN Test Problem

In order to more fully understand the nature of PLN/ECAN synergy, in 2014 we chose to explore it (see [12]) in the context of two test problems standardly used in the context of MLNs (Markov Logic Networks) [4]. These problems are relatively easy for both PLN and MLN, and do not stress either system's capabilities.

The first test case considered there – and the one we will consider here – is a very small-scale logical inference called the *smokes* problem, discussed in its MLN form at [1]. The PLN format of the *smokes* problem used for our experiments is given at https://github.com/opencog/test-datasets/blob/master/pln/tuffy/smokes/smokes.scm.

The conclusions obtained from PLN backward chaining on the *smokes* test case are

```
cancer(Edward) <.62, 1>
cancer(Anna)   <.50, 1>
cancer(Bob)    <.45, 1>
cancer(Frank)  <.45, 1>
```

which is reasonably similar to the output of MLN as reported in [1],

```
0.75 Cancer(Edward)
0.65 Cancer(Anna)
0.50 Cancer(Bob)
0.45 Cancer(Frank)
```

In [12] we explored the possibility of utilizing ECAN to assist PLN on this test problems; however our key conclusion from this work was that ECAN's guidance is not of much use to PLN on the problems as formulated. However, that work did lead us to conceptually interesting conclusions regarding the sorts of circumstances in which ECAN is most likely to help PLN. Specifically, after applying PLN and ECAN to that example, we hypothesized that if one modified the example via adding a substantial amount of irrelevant evidence about other aspects of the people involved, then one would have a case where ECAN could help PLN, because it could help focus attention on the relevant relationships.

This year we have finally followed up on this concept, and have demonstrated that, indeed, if one pollutes the smokes problem by adding a bunch of "noise" relationships with relatively insignificant truth values, then we obtain a case in which ECAN is of considerable use for guiding PLN toward the meaningful information and away from the meaningless, and thus helping PLN to find the meaningful conclusions (the "needles in the haystack") more rapidly. While this problem is very "toy" and simple, the phenomenon it illustrates is quite general and, we believe, of quite general importance.

4 PLN + ECAN on the Noisy Smokes Problem

To create a "noisy" version of the smokes problem, we created a number of "random" smokers and friends and added them to the OpenCog Atomspace along with the Atoms corresponding to the people in the original "smokes" problem and their relationships. We created M random smokers, and N random people associated with each smokers; so $M * N$ additional random people all total. For the experiments reported here, we set $N = 5$. The "smokes" and "friend" relationships linking the random people to others were given truth value strengths of 0.2 for the smoking relationship and 0.85 for the friendship relationship. Of course, these numbers are somewhat arbitrary, but our goal here was to produce a simple illustrative example, not to seriously explore the anthropology of secondhand smoking.

We ran the PLN forward chainer, in a version developed in 2015 that utilizes the OpenCog Unified Rule Engine (URE)[1]. To guide the forward chaining process, we used a heuristic in which the Atom selected as the source of inference is chosen by tournament selection keyed to Atoms' STI values. To measure the surprisingness of a conclusion derived by PLN, we used a heuristic formula based on the Jensen-Shannon Divergence (JSD) between the truth value of an Atom A and the truth value of A^*, the most natural supercategory of A:

$$JSD(A, A^*) * 10^{JSD(A,A^*)}$$

This formula was chosen as a simple rescaling of the JSD, intended to exaggerate the differences between Atoms with low JSD and high JSD. A pending research issue is to choose a heuristic rescaling of the JSD in a more theoretically motivated way; however, this rather extreme scaling function seems to work effectively in practice.

Determining the most natural supercategory is in general a challenging issue, which may be done via using a notion of "coherence" as described in [3]. However, for the simple examples pursued here, there is no big issue. For instance, the supercategory of

```
Evaluationlink
PredicateNode "smokes"
ConceptNode "Bob
```

is simply the SatisfyingSet of the PredicateNode "smokes"; i.e. the degree to which an average Atom that satisfies the argument-type constraints of the

[1] https://github.com/opencog/atomspace/tree/master/opencog/rule-engine.

"smokes" predicate, actually smokes. So then "Bob smokes" is surprising if Bob smokes significantly more often than an average entity.

In the case of this simple toy knowledge base, the only entities involved are people, so this means "Bob smokes" is surprising if Bob smokes significantly more often than the average person the system knows about. In a non-toy Atomspace, we would have other entities besides people represented, and then a coherence criterion would need to be invoked to identify that the relevant supercategory is "People in the SatisfyingSet of the PredicateNode 'smokes'" (including people with a membership degree of zero to this SatisfyingSet) rather than "entities in general in the SatisfyingSet of the PredicateNode 'smokes'".

Similarly, in the context of this problem, the surprisingness of "Bob and Jim are friends" is calculated relative to the generic probability that "Two random people known to the system are friends."

4.1 Tweaks to ECAN

We ended up making two significant change to OpenCog's default ECAN implementation in the course of doing these experiments.

The first change pertained to the size of the system's AttentionalFocus (working memory). Previously the "Attentional Focus" (the working memory of the system, comprised of the Atoms with the highest STI) was defined as the set of Atoms in the Atomspace with STI above a certain fixed threshold, the AttentionalFocusBoundary. In these experiments, we found that this sometimes led to an overly large AttentionalFocus, which resulted in a slow forward chaining process heavily polluted by noise. We decided to cap the size of the AttentionalFocus at a certain maximum value K, currently $K = 30$. This is a somewhat crude measure and better heuristics may be introduced in future. But given that the size of the human working memory seems to be fairly strictly limited, this assumption seems unlikely to be extremely harmful in an AGI context.

The second change pertained to the balance between rent and wages. Previously these values were allowed to remain imbalanced until the central bank's reserve amount deviated quite extremely from the initial default value. However, this appeared to lead to overly erratic behavior. So we modified the code so that rent is updated each cycle, so as to retain balance with wages (i.e. so that, given the particular size of the Atomspace and Attentional Focus at that point in time, rent received will roughly equal wages paid).

4.2 Parameter Setting

The ECAN subsystem's parameters, whose meanings are described in [10], were set as follows:

```
ECAN_MAX_ATOM_STI_RENT      = 3
ECAN_STARTING_ATOM_STI_RENT = 2
ECAN_STARTING_ATOM_STI_WAGE = 30
HEBBIAN_MAX_ALLOCATION_PERCENTAGE = 0.1
```

```
SPREAD_DECIDER_TYPE = Step
ECAN_MAX_SPREAD_PERCENTAGE = 0.5
STARTING_STI_FUNDS    =     500000
```

The PLN forward chainer was set to carry out 5 inference steps in each "cognitive cycle", and one step of basic ECAN operations (importance spreading, importance updating, and HebbianLink formation) was carried out in each cycle as well.

4.3 Experimental Results

Figure 1 shows the number of cycles required to correctly infer the truth value of several relationships in the original smokes problem, depending on the noise parameter M. The results are a bit erratic, but clearly demonstrate that the feedback between ECAN and PLN is working. As the amount of noise Atoms in the Atomspace increases, the amount of time required to draw the correct conclusions does not increase commensurately. Instead, for noise amounts above a very low level, it seems to remain within about the same range as the amount of noise increases.

On the other hand, without ECAN, the PLN forward chainer fails to find the correct conclusions at all with any appreciable amount of noise. In principle it would find the answers eventually, but this would require a huge amount of time to pass. The number of possible inferences is simply be too large, so without some kind of moderately intelligent pruning, PLN spends a long time exploring numerous random possibilities.

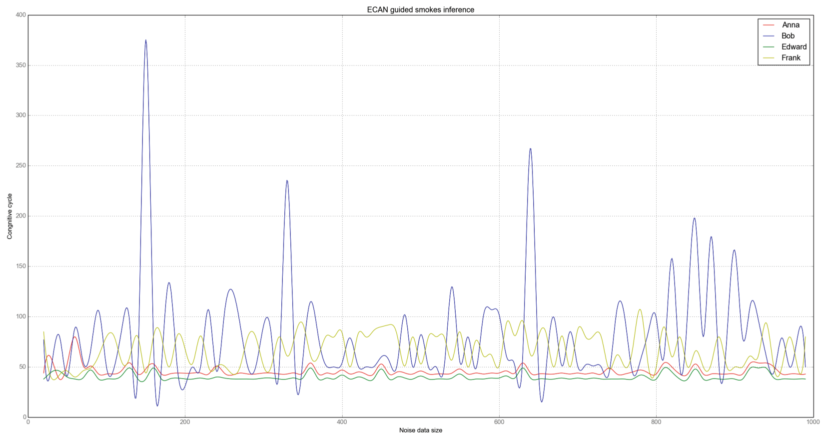

Fig. 1. Number of cognitive cycles required to derive the desired obvious conclusions from the noise-polluted Atomspace. The x-axis measures the amount of noise Atoms added. The graph is averaged over 100 runs. The key point is that the number of cycles does not increase extremely rapidly with the addition of more and more distractions. Red = conclusions regarding Anna; green=Bob, blue=Edward, yellow=Frank. (Color figure online)

5 Conclusion

We began this experimental investigation with the hypothesis that cognitive synergy between PLN and ECAN would be most useful in cases where there is a considerable amount of detailed information in the Atomspace regarding the problem at hand, and part of the problem involves heuristically sifting through this information to find the useful bits. The noisy smokes example was chosen as an initial focus of investigation, and we found that, in this example, ECAN does indeed help PLN to draw reasonable conclusions in a reasonable amount of time, in spite of the presence of a fairly large amount of distracting but uninteresting information.

The theory underlying PrimeAGI contends that this sort of synergy is critical to general intelligence, and will occur in large and complex problems as well as in toy problems like the one considered here. Validating this hypothesis will require additional effort beyond the work reported here, and might conceivably reveal the need for further small tweaks to the ECAN framework.

References

1. Project tuffy. http://hazy.cs.wisc.edu/hazy/tuffy/doc/
2. Bach, J.: Principles of Synthetic Intelligence. Oxford University Press, New York (2009)
3. Ben Goertzel, S.K.: Measuring surprisingness (2014). http://wiki.opencog.org/wikihome/index.php/Measuring_Surprisingness/
4. Niu, F., Re, C., Doan, A., Shavlik, J.: Tuffy: scaling up statistical inference in markov logic networks using an rdbms. In: Jagadish, H.V. (ed.) Proceedings of the 37th International Conference on Very Proceedings of the 37th International Conference on Very Large Data Bases (VLDB 2011), Seattle, Washington, vol. 4, pp. 373–384 (2011)
5. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic Books, New York (2002)
6. Goertzel, B., Ikle, M., Goertzel, I., Heljakka, A.: Probabilistic Logic Networks. Springer, New York (2008)
7. Goertzel, B.: Cognitive synergy: a universal principle of feasible general intelligence? In: Proceedings of ICCI 2009, Hong Kong (2009)
8. Goertzel, B., Pennachin, C., et al.: An integrative methodology for teaching embodied non-linguistic agents, applied to virtual animals in second life. In: Proceedings of the First Conference on AGI. IOS Press (2008)
9. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy. Atlantis Thinking Machines. Springer, Heidelberg (2013)
10. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 2: The CogPrime Architecture for Integrative, Embodied AGI. Atlantis Thinking Machines. Springer, Heidelberg (2013)
11. Goertzel, B., Pitt, J., Ikle, M., Pennachin, C., Liu, R.: Glocal memory: a design principle for artificial brains and minds. Neurocomputing **74**(1–3), 84–94 (2010)

12. Harrigan, C., Goertzel, B., Iklé, M., Belayneh, A., Yu, G.: Guiding probabilistic logical inference with nonlinear dynamical attention allocation. In: Goertzel, B., Orseau, L., Snaider, J. (eds.) AGI 2014. LNCS, vol. 8598, pp. 238–241. Springer, Heidelberg (2014)
13. Looks, M.: Competent Program Evolution. Ph.D. Thesis, Computer Science Department, Washington University (2006)
14. Tulving, E., Craik, R.: The Oxford Handbook of Memory. Oxford U Press, New York (2005)