

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/383696770>

Using Nonlinear Dynamical Attention Allocation to Focus Probabilistic Logical Inference Upon Relevant Information

Preprint · September 2024

DOI: 10.13140/RG.2.2.16246.61767

CITATIONS

0

READS

57

6 authors, including:



Matthew Ikle
SingularityNET

47 PUBLICATIONS 198 CITATIONS

[SEE PROFILE](#)



Gino Yu
The Hong Kong Polytechnic University

40 PUBLICATIONS 320 CITATIONS

[SEE PROFILE](#)

Using Nonlinear Dynamical Attention Allocation to Focus Probabilistic Logical Inference Upon Relevant Information

Matthew Ikle^{1,2} and Misgana Bayetta^{2,3} and Ben Goertzel⁴ and Amen Belayneh^{2,5} and Cosmo Harrigan^{4,5} and Gino Yu⁵

¹ Adams State University, ² School of Design, Hong Kong Poly U, ³ iCog Labs, ⁴ Novamente LLC, ⁵ OpenCog Foundation

Abstract. In a previous article, the authors described a series of experiments combining probabilistic logical inference with an artificial economics based attention allocation system. In those experiments, the authors compared their results with those from two standard examples chosen from the Markov Logic Networks literature. The examples were insufficient to determine the full usefulness of the integrated system, as the information provided in the examples was precisely the information required for inference. Due to limitations of the test suite, any attention allocation system would be unable to provide additional direction.

In this current followup article, the authors describe a new series of experiments and tests intended to demonstrate the effective utilization of attention allocation for inference control. The authors conclude by demonstrating the success, on the first of these experiments, of the cognitive synergy provided via an integration of attention control and probabilistic inference mechanisms.

1 Introduction

Building upon prior work by the authors, involving the integration of probabilistic reasoning and resource allocation, we describe an updated scheme for these two systems for achieving the goal we describe as “cognitive synergy”: the principle of “proactive and mutually-assistive feedback between different cognitive processes associated with different types of memory” [9]. We describe the initial investigations integrating the probabilistic inference system PLN and the nonlinear dynamical attention allocation system ECAN, within the context of the OpenCogPrime (OCP) Artificial General Intelligence framework, in [1]. There we used two test problems, chosen from the Markov Logic Networks (MLN) literature, and compared the results against standard MLN methods.

From that initial experiment, we realized that the test examples were too limited in scope to test real-world reasoning systems. While the results of standalone PLN and PLN in combination with ECAN were comparable to those from the MLN literature, it was clear that no cognitive synergy between PLN and ECAN was ever called upon in the example tests.

The tests simply failed to stress any of the systems with real-world knowledge bases full of misleading, confounding, and extraneous data. Instead, the standard tests contained precisely the information required to determine the desired inference chain. The real world, on the other hand, bombards us with thousands (or millions or even thousands of millions) of pieces of useless, uncertain, or confusing information. To demonstrate the role of attention allocation requires demonstrating a process of sifting and winnowing through this massive amount of extraneous information in order to find precisely those relatively few pieces of relevant information needed for inferential processing.

In this article, we first describe a series of test experiments designed to culminate in providing the sort of “stress-test” required for judging the performance of probabilistic reasoning systems in the “real” world. We demonstrate the tight-integration of the PLN and ECAN systems and display the cognitive synergy arising through this integration as judged from the first test in our test series. We conclude by describing our future plans for implementing deeper PLN/ECAN integration and for inference control to succeed on our second test.

2 OpenCogPrime

As an AGI system designed with tightly coupled components that give rise to complex non-linear dynamics, cognitive synergy is at the core of the design of OpenCogPrime. The system is comprised of 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.

OCP is a large and complex system whose detailed description occupies two volumes [7]. After providing a basic overview of the architecture, we will focus on describing the role of cognitive synergy in two particular aspects of the architecture: the declarative learning component based on probabilistic logic and the attention allocation component based on artificial economics. Although this covers only a relatively small part of a larger complex system, it serves to provide constructive examples of cognitive synergy creating emergent dynamics in a specific AGI design.

Memory Types in OCP The memory types in OCP are the declarative, procedural, sensory, and episodic memory types that are widely discussed in cognitive neuroscience [13], as well as attentional memory for allocating system resources in a generic manner, and intentional memory for allocating system resources in a goal-directed fashion. These memory types are described in more detail in Table 1, where details regarding the corresponding cognitive processes are specified, and references to the associated cognitive functions are included.

The dynamic of closely coupled structures and processes that are associated with each type of memory working together is at the core of the OCP design. The hypothesis is that this will produce a cooperative intelligence that exceeds what could be realized by an architecture where the same structures and processes were present in distinctly separate components.

Memory Type	Specific Cognitive Processes	General Cognitive Functions
Declarative	Probabilistic Logic Networks (PLN) [5]; concept blending [4]	pattern creation
Procedural	MOSES (a novel probabilistic evolutionary program learning algorithm) [10]	pattern creation
Episodic	internal simulation engine [9]	association, pattern creation
Attentional	Economic Attention Networks (ECAN) [8]	association, credit assignment
Intentional	probabilistic goal hierarchy refined by PLN and ECAN, structured according to MicroPsi [3]	credit assignment, pattern creation
Sensory	DeSTIN deep learning architecture	association, attention allocation, pattern creation, credit assignment

Table 1. Memory Types and Cognitive Processes in OpenCogPrime. The third column indicates the general cognitive function that each specific cognitive process carries out.

The dynamics of interaction between processes in OCP is designed in such a way that

- knowledge can be converted between different types of memory (for example, an item of declarative knowledge may, with some computational cost, be interpreted procedurally or episodically)
- 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 described here is an instance of this broad concept.

3 Probabilistic Logic Networks

PLN serves as the probabilistic reasoning system within OpenCog’s more general artificial general intelligence framework. Designed upon solid theoretical and mathematical foundations, PLN nevertheless allows for pragmatic use of heuristic approaches as deemed necessary or convenient.

Architecturally, PLN borrows heavily upon other approaches to uncertain inference including Bayesian probability theory, fuzzy logic, Pei Wang’s Non-Axiomatic Reasoning System (NARS)[15], algorithmic information theory, and Walley’s theory of imprecise probabilities[14].

Declarative knowledge representation within PLN is handled by a weighted labelled hypergraph called the Atomspace, which consists of multiple types of nodes and links, generally weighted with probabilistic truth values and attention values.

Within PLN, a distinction is made between rules and formulas. PLN logical inferences take the form of syllogistic rules, which give patterns for combining statements with matching terms. Related to each rule is a 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.

4 Economic Attention Networks

While PLN can serve as a standalone system, it was primarily designed with the goal of interacting with other cognitive processes within OpenCog, including attentional memory for system resource allocation. The attention allocation system within OpenCog is handled by the Economic Attention Network (ECAN).

The ECAN is a graph of untyped nodes and links that may or may not be typed HebbianLink. Each Atom in an ECAN is weighted with two numbers, called STI (short-term importance) and LTI (long-term importance), while each HebbianLink 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.

5 Evaluating PLN & ECAN

Towards the end of establishing the role of PLN/ECAN cognitive synergy we focused on:

1. crafting a set of two test examples:
 - (a) the first would contain enough extraneous confounding data to require ECAN to determine upon which of the many pieces of information available, the reasoning system should choose to reason;
 - (b) the second would, in addition, contain extraneous sequences of inference rules;
2. designing appropriate measures of Atom and Rule usefulness;
3. building a testing bench for standalone parameter optimization of the attention allocation system.

5.1 The Test Suite

For the purposes of the experiments described here, we will focus on the simpler problem of adding only extraneous data in the form of test 1a above. While we will later describe our plans for handling the more difficult case of choosing patterns of relevant inference rules (test 1b above,) we have not yet implemented those plans.

5.2 Surprisingness

To determine which Atoms have been deemed most useful for fulfilling a PLN task we must first define a measure of usefulness. For this purpose, we decided upon a measure we call “surprisingness,” intended to capture the notion of an Atom’s or pattern’s deviation from its expected value. More precisely, suppose we have for a particular predicate P , a population pop over which P is evaluated, and a particular instance x of the population,

EvaluationLink $< s_{pop} >$
PredicateNode P
ConceptNode pop

and

EvaluationLink $< s >$
PredicateNode P
ConceptNode x

we then define $\text{surprisingness}(x, pop) = |s - s_{pop}|$.

5.3 The ECAN test bench

The ECAN equations contain numerous parameters. To facilitate and simplify our experiments we designed an “ECAN test bench” in which we could more easily study and optimize parameter settings. The test bench allowed us to set up a series of controlled experiments to test ECAN’s responses to artificially controlled stimuli. We were then able to adjust parameter settings to ensure proper behavior on the part of the ECAN system prior to integration with PLN. Particular desired behaviors we wished to observe include:

- Atoms in the attentional focus should “pull” Atoms strongly connected to it via HebbianLinks into the attentional focus; and conversely,
- Atoms very weakly (equivalently, inversely) Hebbian-linked to Atoms in the attentional focus should tend to “pull” the corresponding Atoms out of the attentional focus.

<pre>// Predicate definitions *Friends(person, person) Smokes(person) Cancer(person) // Rule definitions 0.5 !Smokes(a1) v Cancer(a1) 0.4 !Friends(a1,a2) v !Smokes(a1) v Smokes(a2) 0.4 !Friends(a1,a2) v !Smokes(a2) v Smokes(a1)</pre>	<pre>Friends(Anna, Bob) Friends(Anna, Edward) Friends(Anna, Frank) Friends(Edward, Frank) Friends(Gary, Helen) !Friends(Gary, Frank) Smokes(Anna) Smokes(Edward)</pre>	<pre>Cancer(x)</pre>
(a) MLN program	(b) Evidence	(c) Query

Fig. 1. The *smokes* test problem, in the format commonly used for MLN

5.4 The test problem

We began with one of the original test problems from the MLN literature: the *smokes* problem which is shown in its original MLN format below:

It is important to point out here that the * in front of the Friends predicate implies that Friends satisfy the closed world assumption so all ground atoms of this predicate that are not listed are presumed false. The problem is to determine the probabilities, from the given information and inference rules, that each of the individuals incurs cancer.

We should make a few observations here about the form of the query and data for the test example.

- Gary and Helen are disconnected from the rest of the friends graph;
- Neither Gary nor Helen smoke;
- The rules can be transformed into
 - If X smokes then X has cancer (with some probability);
 - If X and Y are friends, and X smokes, then Y smokes (again with some probability value attached).

While this problem appears to contain uncertain truth values, embedded via the rule probabilities, the query itself does not require much real-world reasoning to solve. All that is needed is to manipulate the given probability values. Given the disconnected nature of the graph, no probability of smoking and hence of cancer is transmitted to either Gary or Helen. The graph also contains a number of crisp truth values: pairs of individuals are either friends or not friends, and a person either smokes or does not smoke.

In contrast, the reasoning behind integrating PLN and ECAN is that attention allocation should guide the probabilistic inference system itself. The integrated system should choose atoms based upon their usefulness in answering the query posed and should treat random bits of added information as less useful. This contrasts with the original set of experiments in which only useful pieces of information were provided to the system and so no focusing of attention was therefore necessary or helpful.

To build out our modified query, we thus added the following information to the original graph

- All original “non-smoker” nodes (i.e. Bob, Frank, Gary, and Helen) were supplied with random but small strength and confidence values for the *smokes* predicate;
- Extraneous nodes, with random strength and confidence values for the *smokes* predicate;
- Extraneous links, also with random strength and confidence values for the *friends* links;

We now fed the new queries into two inference systems:

- PLN with no attention allocation;
- PLN guided by the ECAN attention allocation system.

Without the added extraneous random information, both of these problems are easy for MLN, standalone PLN, and PLN+ECAN to solve, as documented in [1]. In our extended test, as expected, the standalone PLN system struggled to solve even the simplest *smokes* test example, but ECAN helped focus the system upon the relevant information in the query. The Tuffy MLN framework has no similar ability to sift through extraneous data. In order to compare PLN+ECAN directly with MLN we will first need to supply extraneous inferential rules as well as extraneous random data. Since the PLN+ECAN code base is still in progress, this step will need to wait until the code is mature enough.

In a real-world application, the weights of the “rules” used in inferences like this (which are really just *ImplicationLinks* like any others) would be set via “direct evaluation” on a relevant dataset, or via inference from other logical relationships learned from other sources; and the rules themselves might be mined from data via OpenCog’s frequent pattern mining algorithms [12] or via OpenCog’s MOSES probabilistic evolutionary learning algorithm [10]. For the examples given here, though, we have followed the practice of most MLN testing and simply set the rules and weights externally to the AI system. This means that for the purposes of this paper, we are studying only the probabilistic inference process of CogPrime, in isolation from its pattern mining aspects.

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 [2],

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

Modifying the rule probabilities will change the probabilities somewhat, but in a fairly smooth way in this case.

6 Conclusion and Future Directions

We have described an example scenario requiring the choice of selected Atoms upon which PLN should allocate reasoning efforts. We have demonstrated that ECAN can help focus PLN probabilistic reasoning upon appropriate sets of Atoms. While we have demonstrated the cognitive synergy arising from this integration of ECAN and PLN, we still have much work to complete our research in these integration efforts. Effective resource allocation needs to be demonstrated not only across Atom choice, but in the much larger context of rule selection and forward and backward chaining.

Our current work focuses on demonstrating that this use of economic attention allocation (ECAN) to help guide probabilistic inference (PLN) can show significant benefit. This provides one elementary example of the cognitive synergy principle that lies at the core of the OpenCogPrime AGI architecture.

References

- 1.
2. Project tuffy, <http://hazy.cs.wisc.edu/hazy/tuffy/doc/>
3. Bach, J.: Principles of Synthetic Intelligence. Oxford University Press (2009)
4. Fauconnier, G., Turner, M.: The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Basic (2002)
5. Goertzel, B., Ikle, M., Goertzel, I., Heljakka, A.: Probabilistic Logic Networks. Springer (2008)
6. Goertzel, B.: Opencogprime: A cognitive synergy based architecture for artificial general intelligence. In: Cognitive Informatics, 2009. ICCI'09. 8th IEEE International Conference on. pp. 60–68. IEEE (2009)
7. Goertzel, B., Pennachin, C., Geisweiller, N.: Engineering General Intelligence, Part 1. Atlantis Press (2014)
8. Goertzel, B., Pitt, J., Ikle, M., Pennachin, C., Liu, R.: Glocal memory: a design principle for artificial brains and minds. Neurocomputing (2010)
9. Goertzel, B., Pitt, J., Wigmore, J., Geisweiller, N., Cai, Z., Lian, R., Huang, D., Yu, G.: Cognitive synergy between procedural and declarative learning in the control of animated and robotic agents using the opencogprime agi architecture. In: AAAI (2011)
10. Looks, M.: Competent Program Evolution. Ph.D. thesis, Washington University at Saint Louis (2006)
11. McCallum, A., K., N., J., R., K., S.: Automating the construction of internet portals with machine learning. Information Retrieval 3(2), 127–163 (2000)
12. O'Neill, J., Goertzel, B., Ke, S., Lian, R., Sadeghi, K., Shiu, S., Wang, D., Yu, G.: Pattern mining for general intelligence: The fishgram algorithm for frequent and interesting subhypergraph mining. In: J. Bach, B. Goertzel, M.I. (ed.) Artificial General Intelligence. pp. 189–198. Lecture Notes in Artificial Intelligence, Springer (2012)
13. Tulving, E., Craik, R.: The Oxford Handbook of Memory. Oxford U. Press (2005)
14. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London, New York (1991)
15. Wang, P.: Non-Axiomatic Reasoning System: Exploring the essence of intelligence. Ph.D. thesis, Indiana University (1995)