Occam Learning through Pattern Discovery: Computational Mechanics in AI Systems

Dr. James A. Crowder, Dr. John N. Carbone
Raytheon Intelligence and Information Systems
JACrowder@raytheon.com, John_N_Carbone@raytheon.com

Abstract - *The push for real-time autonomous AI systems has been sought for decades. The DoD has spent considerable R&D budgets looking for systems that can operate with no or little supervision. These systems must process incredible amounts of heterogeneous information looking for information. In order to achieve these goals, we must affect real learning, or "learning with experience," in autonomous AI systems [10]. The goal of having machines that learn with experience is one of the most intriguing problems in computer science and computer engineering. As the types of problems we would like AI systems to solve get more complex and more diverse, it is becoming a necessary task as well. Unfortunately, by its nature, learning is somewhat fuzzy, and random in nature, for information comes at us in stochastic fashion [22]. In fact, the overall goal is to learn things we do not yet know, and in doing so find patterns that we can learn. This constitutes not patter matching, or pattern recognition, but is, in fact, pattern discovery. Nonetheless, we would like a mathematical framework for machine learning to aid in our understanding and improve our ability to make progress toward autonomous AI systems.*

1 Introduction

Multi-layer neural networks have been successfully applied to complex pattern recognition and functional approximation problems. The notion of pattern recognition is readily known and understood [3]. However, here we introduce a different concept in AI machine learning, the concepts of *Pattern Discovery*, and a different way of looking at learning for autonomous systems, whose input will come in very stochastic and unpredicted ways. Here we look at the concepts of finding causal structure in stochastic data that come out of computational physics.

The term Pattern Discovery is intended to be in contrast to the well understood concepts of both *Pattern Recognition* and *Pattern Learning*. In Patter Recognition the aim is to analyze input data and assign it to one or more pre-determined categories or patterns. In most Pattern Learning systems, the goal is to determine which of the several pre-determined categories available to the algorithms corresponds to the correct one [25]. Obviously the two are closely correlated and in both cases, the categories or representations have been handed to the algorithms, due to choices and decisions which are external or outside the pattern recognition and learning algorithms or procedures.

For our use of Pattern Discovery, the goal is to avoid the necessity for *a priori* knowledge about what structures, or patterns, may be relevant [20]. This is not a new problem, and is as old as the first attempt to process information. The classical approach, based on statistical mechanics, is to derive patterns (or macroscopic properties) from raw data (or microscopic components). Here we take the inverse approach, extending the concept of extracting "geometry" or causal structures, from a time or frequency series of data or information. Here we build upon the concept of "Occam Learning" [7] to construct the simplest model capable of capturing causal structures, or "patterns" in the data which constitutes a representation of the causal structure of the hidden process(es) which generated the behavior observed or captured. We assert that this representation is the maximally efficient model of the observed data-generating process, based on the learning principles laid out in the Occam Learning Process. The underlying computational mechanics concepts have been used to analyze dynamical systems, evolving spatial computation, and stochastic resonance, among others. However, the combination of computational mechanics coupled with the Occam AI learning constructs provides a unique method for Pattern Discover in large, heterogeneous data sets. These methods, combined with extensive AI cognitive processes, e.g., the Artificial

Cognitive Neural Framework (ACNF) [1, 4, 5, 6], move us toward providing real-time, completely autonomous AI systems.

Presented here is a mathematical framework for discovering, describing, and quantifying new patterns, based in computational mechanics and using tools from statistical physics. It construct optimal, minimal models of stochastic processes and their underlying causal structures that drive an "Occam Learning" model of intrinsic computational information transformation. Here we summarize the mathematical foundation of computational mechanics, especially those constructs in optimality and uniqueness to drive the Occam learning algorithms. We describe the principles and motivations underlying the computational mechanics, emphasizing its underlying connections to the minimum description length principles underlying Occam Learning, and its implications to Probably, Approximately Correct (PAC) machine learning concepts [2].

We will first examine the concepts and issues involving Pattern Discover and how they are addressed with computational mechanics. Then we provide discussion of the mathematical structures, based in computational mechanics that provide the foundation for Pattern Discovery, with particular attention to optimality and uniqueness theorems. These uniqueness theorems are then utilized within the Occam Learning framework to provide AI learning algorithms to learn and extend these unique pattern structures. Differences between this work and other work in computational mechanics is that we utilize Renyi's Entropy theory vs. Shannon's, utilizing Renyi's mutual information theory in our computations involving stochastic processes [6, 11].

2 Pattern Discovery Concepts

Any approach we take to handling Pattern Discovery should meet a number of criteria:

- ***Predictive*** – the models the algorithms produce should allow the system to predict the original process or system that produced the data, and provide a compressed description of it (learned pattern).
- ***Computational*** – should have stored in system memories how that process or system stores, transmits, and transforms information (what was the causal structure that produced the information?).
- ***Calculable*** – either analytically or by systematic approximation.
- ***Causal*** – system should understand how instances of the discovered patterns are actually produced.
- ***Naturally Stochastic*** – the learned patterns models should not just be tolerant of noise, but should be explicitly formulated in terms of stochastic ensembles.

For our uses, the key idea of utilizing computational mechanics is the supposition that the information required to drive the Occam Learning Pattern Discovery is actuall in the data, or information picked up by the system's sensors, provided there is enough information. The real problem dealing with real-time information coming from a number of possibly heterogeneous sensors is to determine which data sets (or partitions) of data sets should be treated as equivalent, and how the data should be partitioned. We assert that correct mapping of data to partitions should leave the system with discovered patterns that provide the system's cognitive framework with the same degree of knowledge about the future of the data (similar prediction accuracy and consistency), has a proper degree of plausibility, but is also vague enough to account for future growth of the learned pattern definition [17, 18]. So the question is: how to create the partitions? For this we look to genetic algorithms. We create generations of partitions and then use these to create Occam patterns, or memories, from the populations, based on the partition constraints (based on computational mechanics). These are evaluated, based on Entropy calculations. Those partitions from the population that produce the best fit and utilized (with mutation and crossover) are used to create a new generation of partition population. The process continues until an optimal partition is created. Those partitions that produce patterns similar to those already in the system's memories are sent to algorithms that evaluate the patterns for memory extensions or reinterpretations. Those that are not already part of the system's memories are used to create new memories.

For this discussion, we refer to $\boldsymbol{H[X]}$ as the entropy of discrete random variable $\boldsymbol{X}$, interpreted as the uncertainty in $\boldsymbol{X}$. $\boldsymbol{H[X|Y]}$ is the entropy of $\boldsymbol{X}$ conditional on $\boldsymbol{Y}$, and $\boldsymbol{I[X|Y]}$ is the mutual information between $\boldsymbol{X}$ and $\boldsymbol{Y}$, as measured by Renyi's Entropy and Mutual Information computations. Also, we restrict ourselves to discrete-valued, discrete-time stochastic processes (analogous to sensor data being collected by an autonomous system). Such processes are sequences of random variables, $\boldsymbol{S_i}$, and the values are taken from a countable set $\boldsymbol{A}$. This is reasonable since we are talking about a system with multiple sensors, each taking in data over a specified period of time, each with a countable number of data samples.

Our goal is to discover a pattern, or Occam memory that will predict all or part of the future of process $\boldsymbol{S}, \vec{S}$, using some function and some part of $\overleftarrow{S}$. We begin by taking the set $\overleftarrow{S}$ of past data points and partitioning it into mutually exclusive and jointly comprehensive subsets, as shown in Figure 1. That is, we make a class $\mathfrak{R}$ of subsets.

Patterns in Data Ensembles: In order to discuss Pattern Discovery in data ensembles, we must have a way to discuss the uncertainty of Occam memories to predict future information states.

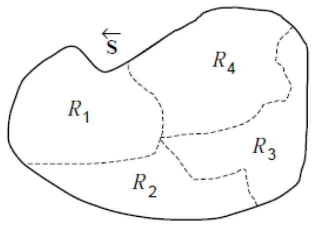

Figure 1 - $\mathfrak{R}$ Partitions of the set $\overleftarrow{S}$

We cannot use:

$H\left[\vec{S}\right]$ Since this is infinite. Instead we use:

$H\left[\overrightarrow{S}^{L}\right]$ where the uncertainty of the next ***L*** data is treated as a function of ***L***. Therefore, $\mathfrak{R}$ captures or discovers a pattern, ***iff*** there exists an L such that:

$$H\left[\overrightarrow{S}^{L} \mid \mathfrak{R}\right] < LH[S]$$

$\mathfrak{R}$ discovers a pattern when it tells us something about how the distinguishable parts of the process affect each other, or how $\mathfrak{R}$ exhibits its independence, based on the entropy calculations discussed earlier. The smaller that

$$H\left[\overrightarrow{S}^{L} \mid \mathfrak{R}\right]$$

is, the stronger the pattern discovered by $\mathfrak{R}$. The causal state, as determined by the Occam memory of a captured pattern, together with the next observed process, determine a new causal state (and may cause a redefinition of the Occam memory). Thus, there is a natural relation of succession among the causal states of a captured pattern or causal process. This leads us to the definition of a captured or discovered pattern, which leads to an Occam Memory within the AI system. Each discovered pattern, or Occam Memory will have the following properties:

- Occam Memories are deterministic
- All Occam Memory causal states are independent
- All Occam memories are reconstructed from information fragements
- All Occam memory causal states are maximally prescient
- All Occam memory causal states are minimal for all prescient rival memories
- All Occam memory causal states are unique
- All Occam memories are minimally stochastic for all prescient rival memories.
- The excess entropy, ***E***, of an Occam Memory is the Mutual Information between the memory's semi-infinite past and its semi-infinite future.

3 Computational Mechanics and Occam Learning

"*Entities should not be multiplied unnecessarily*."
William of Occam (1320 AD)

This maxim from William of Occam, called "***Occam's Razor***," is often sited to justify one hypothesis over others, and is taken to mean "*prefer simpler explanations*." However, what reason might we have to believe that simpler explanations lead us to a hypothesis with fewer errors?

One might simply reason this from the observation that there are far fewer simple explanations than complex ones. However, it may be no more complex than the reasoning that simple explanations are less likely to fit data, just by chance. Another way to view this is that by favoring smaller hypotheses over larger, we are less likely to run across bad hypotheses, which one of the fundamental axioms behind Occam Learning. Another axiom of Occam Learning is:

"***Learning is Data Compression***"

The more the data is compressed, i.e., the more complex the learning algorithm, the more likely something subtle that is important is missed or eliminated. Reasoning from this perspective, we define an "Occam Learning Algorithm" to be one that produces hypotheses, or "Pattern Discoveries, that are simple in structure, and grow slowly as more data are analyzed. In fact, analysis has shown [4, 6, 7] that if we have a small hypothesis space, then by taking a polynomial number of data samples, we can achieve "Uniform Convergence," i.e., the chance that any bad hypothesis with error > c, that is still consistent with the data, can be forced below some arbitrary number δ [23]. In the converse, is it impossible to get uniform convergence with a large hypothesis spaces, given a polynomial number of data samples, the answer is, sometimes [7].

Since learning is very stochastic in nature, particularly for real-time systems with heterogeneous data inputs, and given that it is impossible to know how many data points for a given unknown pattern may exist, we employ Occam Learning to provide Pattern Discovery. What we desire, then, is a mathematical framework and foundation for AI system learning, based in computational mechanics and Occam Learning principles to provide the AI system with autonomous understanding, reasoning, and decision.

Towards this end, a memory computational framework that encompasses the computational theory of machine learning discussed here, the goals of which are:

- To provide computational mechanics mathematical models that capture key aspects of Occam Learning.
- To provide the system self-analytical metrics for its algorithms:
  - When will they succeed?
  - How long will they take?
- To develop algorithms that provably meet desired criteria;
- To provide the system self-guidance about which algorithms to use when.
- To allow the system to analyze the inherent ease or difficulty of learning problems.

Figure 2 illustrates the Occam Learning, computational mechanics framework.

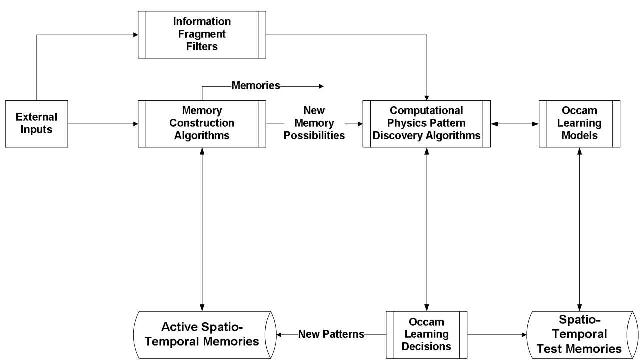

Figure 2 – The Occam Learning, Computational Mechanics Learning Framework

This Occam Learning system is one aspect of the "Learning Algorithms" depicted in the overall Artificial Cognitive Neural Framework depicted in Figure 3 [11]. The three main subsystems within the architecture are [10]:

- The Mediator: this gathers information and facilitates communication between software agents within the system [8]. The Mediator takes information from perceptrons and from coalitions of perceptrons and updates the short-term, long-term, and episodic memories.
- the Memory System: The information available in memory, i.e., what the system has learned, is continually broadcast to the conscious perceptrons that form the cognitive center of the system.
- Cognitive System: this is responsible for the cognitive functionality of perception [12], consciousness, emotions, information processing, etc.

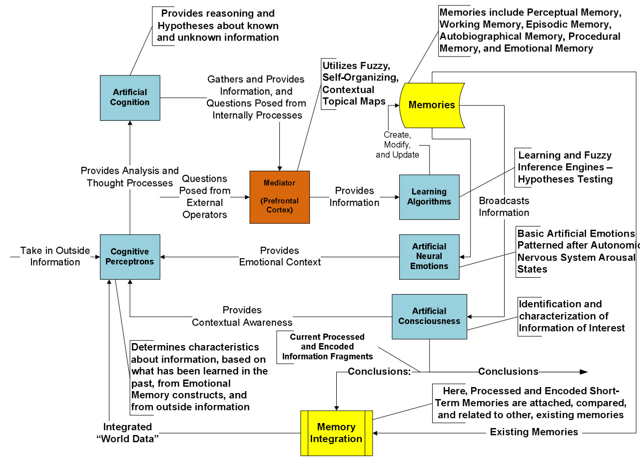

Figure 3 – The Artificial Cognitive Neural Framework

The Learning Center, or Learning Algorithms as it is depicted in Figure 3, provides a number of learning systems, of which Occal Learning is the most basic and fundamental. Other learning structures, such as the Probably, Approximately Correct (PAC) learning algorithms take their cues from the Occam Learning structures (patterns), and use combinations of these to infer higher-order learning within the AI system. These higher-order learning algorithms allow the system to "infer" new and more complex concepts, that turn into new, more complex memories, than it previously stored or had available. In this way, the system "evolves." The ACNF provides:

- An architectural framework for "conscious" software agents.
- A plug-in framework for the domain-independent portions of the "consciousness" mechanisms.
- An easily self-customizable framework for the domain-specific portions of the "consciousness" mechanisms.
- The cognitive mechanisms for behaviors and emotions required for the conscious software agents.

The ACNF contains several different memory systems (including emotional memories) each with a specific purpose. The Occam and PAC memory systems reside in both the Perceptual and Emotional memory systems [9]. The memory systems and their descriptions are [13, 14, 15, 16]:

- **Perceptual Memory**: this memory enables identification, recognition, and characterization, including emotions.
- **Working Memory**: this contains preconscious buffers as a temporary workspace for the internal AI system activities.

- **Episodic Memory**: this is a content-addressable associative memory with a rapid decay (very short-term memory) [2].
- **Autobiographical Memory**: this is the long-term bi-directional associative memory for facts and data.
- **Procedural Memory**: this is the long-term memory for learned skills within the system.
- **Emotional Memory**: both long-term (spatio-temporal) and implicit (inference) emotional memories.

4 Conclusions and Discussion

Here we have laid the foundations for learning structures that will be required for real-time autonomous AI systems. We have provided a mathematical basis for these learning algorithms, based in computational mechanics. The Occam Learning system is but one of many learning constructs that must be present for an AI system to actually act autonomously and to make sense of a complex world it will find itself a part of. The Occam Learning Computational Framework provides the ability for simple Pattern Discovery that feeds more complex memory and inference systems within the ACNF to allow the autonomous system to think, reason, and evolve. We have but scratched the surface in providing constructs and methodologies required for a self-aware, thinking, reasoning, and fully autonomous real-time AI system [19].

5 References

1. Crowder, J., Barth, T., and Rouch, R., "*Learning Algorithms for Stochastically Driven Fuzzy, Genetic Neural Networks*." NSA Technical Paper, Denver (1999) ENIGMA_1999_002.
2. Crowder, J., Barth, T., and Rouch, R., "*Neural Associative Memory for Storing Complex Memory Patterns*." NSA Technical Paper, Denver (1999) ENIGMA_1999_003.
3. Crowder, J., Barth, T., and Rouch, R., "*Evolutionary Neural Infrastructure with Genetic Memory Algorithms: ENIGMA Theory Development*." NSA Technical Paper, Denver (1999) ENIGMA_1999_004.
4. Crowder, J. A., "*Integrating an Expert System into a Neural Network with Genetic Programming for Process Planning*." NSA Technical Paper TIT_01_01_013_2001_001 (2001).
5. Crowder, J. A. "*ELYSE, The Evolving, Life-like Yielding, Symbiotic Environment.*" NSA Technical Paper Technical Paper TIT_01_01_013_2001_002 (2001).
6. Crowder, J. A., "*Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System*." NSA Technical Paper CON_0013_2002_003 (2002).
7. Crowder, J. A., "*Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems*." NSA Technical Paper CON_0013_2003_009 (2003).
8. Crowder, J. A., "*Operative Information Software Agents (OISA) for Intelligence Processing*," AIAA Aerospace@Infotech 2010 Conference (May, 2010).
9. Crowder, J. A., Friess, S., "*Artificial Neural Emotions and Emotional Memory*." International Conference on Artificial Intelligence, ICAI'10 (July, 2010).
10. Crowder, J. A., "*Flexible Object Architectures for Hybrid Neural Processing Systems*," International Conference on Artificial Intelligence, ICAI'10 (July, 2010).
11. Crowder, J. A., Carbone, J. N., "*The Great Migration: Information to Knowledge using Cognition-Based Frameworks*." Springer Science, New York (2011).
12. Marsella, S., and Gratch J., "*A Step Towards Irrationality: Using Emotion to Change Belief.*" 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
13. Newell, A., "*Unified Theories of Cognition*." Cambridge MA: Harvard University Press (2003).
14. Damasio A (1994) "*Descartes' error: Emotion, reason, and the human brain*." New York: Gosset/Putnam.
15. Dolan RJ, Vuilleumier P (2003), "*Amygdala automaticity in emotional processing*." Ann N Y Acad Sci 985:348-355.
16. Dudai Y (2004) "*The neurobiology of consolidations, or, how stable is the engram*?" Annu Rev Psychol 55:51-86.
17. Eichenbaum H (2002) "*The cognitive neuroscience of memory*." New York: Oxford University Press.
18. Kosko, G., "*Fuzzy Cognitive Maps*," International Journal of Man-Machine Studies, 24: 65-75.
19. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
20. J. Rowley and R. Hartley, "*Organizing knowledge: An introduction to managing access to information*." Ashgate Pub Co, 2008.
21. L. Zadeh, "*A note on web intelligence, world knowledge and fuzzy logic*," Data & Knowledge Engineering, vol. 50, pp. 291-304, 2004.
22. S. S. Liao, J. W. He, and T. H. Tang, "*A framework for context information management*," Journal of Information Science, vol. 30, pp. 528-539, 2004.
23. M. Pinto, "*A grounded theory on abstracts quality: Weighting variables and attributes*," Scientometrics, vol. 69, pp. 213-226, 2006.