Three Hypotheses About the Geometry of Mind

Ben Goertzel [1], Matthew Ikle' [2]

[1] Novamente LLC, Rockville MD
[2] Adams State College, Alamosa CO

Abstract. What set of concepts and formalizations might one use to make a practically useful, theoretically rigorous theory of generally intelligent systems? We present a novel perspective motivated by the OpenCog AGI architecture, but intended to have a much broader scope. Types of memory are viewed as categories, and mappings between memory types as functors. Memory items are modeled using probability distributions, and memory subsystems are conceived as "mindspaces" – geometric spaces corresponding to different memory categories. Two different metrics on mindspaces are considered: one based on algorithmic information theory, and another based on traditional (Fisher information based) "information geometry". Three hypotheses regarding the geometry of mind are then posited: 1) a *syntax-semantics correlation* principle, stating that in a successful AGI system, these two metrics should be roughly correlated; 2) a *cognitive geometrodynamics* principle, stating that on the whole intelligent minds tend to follow geodesics in mindspace; 3) a *cognitive synergy* principle, stating that shorter paths may be found through the composite mindspace formed by considering multiple memory types together, than by following the geodesics in the mindspaces corresponding to individual memory types.

1 Introduction

One of the many factors making AGI research difficult is the lack of a broadly useful, powerful, practical theoretical and mathematical framework. Many theoretical and mathematical tools have played important roles in the creation and analysis of contemporary proto-AGI systems; but by and large these have proved more useful for dealing with *parts* of AGI systems than for treating AGI systems holistically. And the general mathematical theory of AGI [11], though it has inspired some practical work [12] [18], has not yet been connected with complex AGI architectures in any nontrivial way.

This paper gives a rough sketch of a novel theoretical framework that we think may have potential to help with progress toward AGI. While the framework has been developed largely in the context of a quest to understand and improve the dynamics of the OpenCog [10] AGI architecture, it is intended to be much more broadly applicable. The key ingredients of the framework are: modeling multiple memory types as mathematical categories (with functors mapping between them), modeling memory items as probability distributions, and measuring distance between memory items using two metrics, one based on algorithmic information theory and one on classical information geometry. Three core hypotheses are then presented: 1) a **syntax-semantics correlation** principle, stating that in a successful AGI system, these two metrics should be roughly correlated; 2) a **cognitive geometrodynamics** principle, stating that on the whole intelligent minds tend to follow geodesics in mindspace; 3) a **cognitive synergy** principle, stating that shorter paths may be found through the composite mindspace formed by considering multiple memory types together, than by following the geodesics in the mindspaces corresponding to individual memory types.

2 A Simple Formal Model of Intelligent Agents

We utilize here a general formal model of intelligent agents called SRAM (Simple Realistic Agents Model), presented in [9] and inspired by the simpler agents model in [11]. For space reasons we will not explicitly present SRAM here, but only summarize the more relevant points. SRAM begins with a class of active agents which observe and explore their environment and take actions in it, which may affect the environment, and may bring the agent rewards. The agent's "history" consists of the actions, observations and rewards it has experienced. Agents may also have "goals", i.e. functions of their future history that they seek to maximize in order to achieve reward.

SRAM also abstractly models agents' short and long term memories. One assumes the agent's memory consists of separate memory stores corresponding to multiple types of memory, e.g.: procedural (K_{Proc}), declarative (K_{Dec}), episodic (K_{Ep}) and attentional (K_{Att}), which may be formally modeled as follows:

- an injective mapping $\Theta_{Ep} : K_{Ep} \rightarrow \mathcal{H}$ where $\mathcal{H}$ is the space of fuzzy sets of subhistories (subhistories being "episodes" in this formalism)
- a mapping $\Theta_{Proc} : K_{Proc} \times \mathcal{M} \times \mathcal{W} \rightarrow \mathcal{A}$, where $\mathcal{M}$ is the set of memory states, $\mathcal{W}$ is the set of (observation, goal, reward) triples, and $\mathcal{A}$ is the set of actions (this maps each procedure object into a function that enacts actions in the environment or memory, based on the memory state and current world-state)
- a mapping $\Theta_{Dec} : K_{Dec} \rightarrow \mathcal{L}$, where $\mathcal{L}$ is the set of expressions in some formal language (which may for example be a logical language), which possesses words corresponding to the observations, goals, reward values and actions in our agent formalism
- a mapping $\Theta_{Att} : K_{Dec} \cup K_{Proc} \cup K_{Ep} \rightarrow \mathcal{V}$, where $\mathcal{V}$ is a set of "attention values" indicating the importance of a memory to the agent at a given point in time (for instance, OpenCog uses pairs (STI, LTI) indicating short and long term importance values).

The vocabulary of actions contains memory-actions corresponding to the operations of inserting the current observation, goal, reward or action into the episodic and/or declarative memory store. The activity of the agent, at each point in time, consists of enacting one or more of the procedures in the procedural memory store (and a set of simultaneously executed procedures may be formally modeled as a single one). Finally, SRAM's workspace provides a medium for interaction between the different memory types (cf Baars' "global workspace theory" [2]). At each time-step, the agent may carry out an external action a_i on the environment, a memory action m_i on the (long-term) memory, and an action b_i on its **internal workspace**, including insertions/deletions of observations, goals, actions or reward-values from the workspace.

3 Modeling Memory Types Using Category Theory

Next we formalize the different types of memory critical for a human-like integrative AGI system, in a manner that makes it easy to study mappings between different memory types. One way to do this, introduced in [7], is to consider each type of memory as a *category*, in the sense of category theory.

For example, we may model the space of procedures as a *graph*. We assume there exists a set T of "atomic transformations" on the category C_{Proc} of procedures, so that each $t \in T$ maps an input procedure into a unique output procedure. We then consider a labeled digraph whose nodes are objects in C_{Proc} (i.e. procedures), and which has a link labeled t between procedure P_1 and P_2 if t maps P_1 into P_2. Morphisms on program space may then be taken as paths in this digraph, i.e. as composite procedure transformations defined by sequences of atomic procedure transformations. For example, in OpenCog procedures are represented as *ensembles of program trees*, where program trees are defined in the manner suggested in [16]; in this case one can consider tree edit operations as defined in [3] as one's atomic transformations.

The category C_{Dec} of declarative knowledge may be handled somewhat similarly, via assuming the existence of a set of transformations between declarative knowledge items, constructing a labeled digraph induced by these transformations, and defining morphisms as paths in this digraph. For example, if declarative knowledge items are represented as expressions in some logical language, then transformations may be naturally taken to correspond to inference steps in the associated logic system. Morphisms then represent sequences of inference steps that transform one logical expression into another.

Having defined memory types as categories, one can then look at functors between these categories, e.g. transformations that map programs into logical statements and vice versa. For instance a "procedure declaratization" is a functor from K_{Proc} to K_{Dec}; in other words, a pair of mappings (r, s) so that r maps each object in K_{Proc} into some object in K_{Dec}; and s maps each morphism $f_{Proc,i}$ in K_{Proc} into some morphism in K_{Dec}, in a way that obeys $s(f_{Proc,i} \circ f_{Proc,j}) = s(f_{Proc,i}) \circ s(f_{Proc,j})$.

Similarly, we may define a "declaration procedurization" as a functor from K_{Dec} to K_{Proc}. This formalism maps easily into many intuitively simple cases, e.g.: the category *blue* versus the procedure *isBlue* that outputs a number in $[0, 1]$ indicating the degree of blueness of its input; a procedure for multiplying numbers, versus a verbal description of that procedure; a logical description of the proof of a theorem based on some axioms; versus a procedure that produces the theorem given the axioms as inputs.

Episodic-declarative conversion is also important and may be similarly formalized; particular cases of this have received significant attention in the cognitive science literature, referred to by the term "symbol grounding." Conceptually, episode declaratization produces a declaration describing an episode-set (naturally this declaration may be a conjunction of many simple declarations); whereas declaration episodization produces an episode-set defined as the set of episodes whose descriptions include a certain declaration. For example: the predicate $isCat(x)$ could be mapped into the fuzzy set E of episodes containing cats, where the degree of membership of e in E could be measured as the degree to which e contains a cat. In this case, the episode-set would commonly be called the "grounding" of the predicate. Similarly, a relationship such as a certain sense of the preposition "with" could be mapped into the set of episodes containing relationships between physical entities that embody this word-sense.

Given this formalization of mappings between different memory types, we see that a probability distribution over any one memory type, may naturally be mapped into a probability distribution over all the different memory types, using the functors mapping between them. However, additional uncertainty is generally introduced in this mapping, meaning e.g. that a high-confidence distribution over declarations may result in a lower-confidence distribution over procedures.

4 Metrics on Memory Spaces

Bringing together the ideas from the previous sections, we now explain how to use the above ideas to define geometric structures for cognitive space, via defining two metrics on the space of *memory store dynamic states*. Specifically, we define the dynamic state or *d-state* of a memory store (e.g. attentional, procedural, etc.) as the series of states of that memory store (as a whole) during a time-interval. Generally speaking, it is necessary to look at d-states rather than instantaneous memory states because sometimes memory systems may store information using dynamical patterns rather than fixed structures.

It's worth noting that, according to the metrics introduced here, the above-described mappings between memory types are topologically continuous, but involve considerable geometric distortion – so that e.g., two procedures that are nearby in the procedure-based mindspace, may be distant in the declarative-based mindspace. This observation will lead us to the notion of cognitive synergy, below.

4.1 Information Geometry on Memory Spaces

Our first approach involves viewing memory store d-states as probability distributions. A d-state spanning time interval (p, q) may be viewed as a mapping whose input is the state of the world and the other memory stores during a given interval of time (r, s), and whose output is the state of the memory itself during interval (t, u). Various relations between these endpoints may be utilized, achieving different definitions of the mapping e.g. $p = r = t, q = s = u$ (in which case the d-state and its input and output are contemporaneous) or else $p = r, q = s = t$ (in which case the output occurs after the simultaneous d-state and input), etc. In many cases this mapping will be stochastic. If one assumes that the input is an *approximation* of the state of the world and the other memory stores, then the mapping will nearly always be stochastic. So in this way, we may model the total contents of a given memory store at a certain point in time as a probability distribution. And the process of learning is then modeled as one of *coupled changes in multiple memory stores*, in such a way as to enable ongoingly improved achievement of system goals.

Having modeled memory store states as probability distributions, the problem of measuring distance between memory store states is reduced to the problem of measuring distance between probability distributions. But this problem has a well-known solution: the Fisher-Rao metric!

Fisher information is a statistical quantity which has a a variety of applications, ranging beyond statistical data analysis, including physics [5], psychology and AI [1]. Put simply, FI is a formal way of measuring the amount of information that an observable random variable X carries about an unknown parameter θ upon which the probability of X depends. FI forms the basis of the Fisher-Rao metric, which has been proved the only Riemannian metric on the space of probability distributions satisfying certain natural properties regarding invariance with respect to coordinate transformations. Typically θ in the FI is considered to be a real multidimensional vector; however, [4] has presented a FI variant that imposes basically no restrictions on the form of θ, which is what we need here.

Suppose we have a random variable X with a probability function $f(X, \theta)$ that depends on a parameter θ that lives in some space M that is not necessarily a dimensional space. Let $E \subseteq \mathcal{R}$ have a limit point at $t \in \mathcal{R}$, and let $\gamma : E \to M$ be a path. We may then consider a function $G(t) = \ln f(X, \gamma(t))$; and, letting $\gamma(0) = \theta$, we may then define the *generalized Fisher information* as $\mathcal{I}(\theta)_\gamma = \mathcal{I}_X(\theta)_\gamma = E\left[\left(\left(\frac{\partial}{\partial t} \ln f(X; \gamma(t))\right)^2\right) |\theta\right]$.

Next, Dabak [4] has shown that the geodesic between θ and θ' is given by the exponential weighted curve $(\gamma(t))(x) = \frac{f(x,\theta)^{1-t} f(x,\theta')^t}{\int f(y,\theta)^{1-t} f(y,\theta')^t dy}$, under the weak condition that the log-likelihood ratios with respect to $f(X, \theta)$ and $f(X, \theta')$ are finite. It follows that if we use this form of curve, then the generalized Fisher information reduces properly to the Fisher information in the case of dimensional spaces. Also, along this sort of curve, the sum of the Kullback-Leibler distances between θ and θ', known as the J-divergence, equals the integral of the Fisher information along the geodesic connecting θ and θ'.

Finally, another useful step for our purposes is to bring Fisher information together with imprecise and indefinite probabilities as discussed in [6]. For instance an indefinite probability takes the form $((L, U), k, b)$ and represents an envelope of probability distributions, whose means after k more observations lie in (L, U) with probability b. The Fisher-Rao metric between probability distributions is naturally extended to yield a metric between indefinite probability distributions.

4.2 Algorithmic Distance on Memory Spaces

A conceptually quite different way to measure the distance between two d-states, on the other hand, is using algorithmic information theory. Assuming a fixed Universal Turing Machine M, one may define $H(S_1, S_2)$ as the length of the shortest self-delimiting program which, given as input d-state S_1, produces as output d-state S_2. A metric is then obtained via setting $d(S_1, S_2) = (H(S_1, S_2) + H(S_2, S_1)/2$. This tells you the computational cost of transforming S_1 into S_2.

There are variations of this which may also be relevant; for instance [19] defines the generalized complexity criterion $K_{\Phi}(x) = min_{i \in N}\{\Phi(i, \tau_i)|L(p_i)) = x\}$, where L is a programming language, p_i is the i'th program executable by L under an enumeration in order of nonincreasing program length, τ_i is the execution time of the program p_i, $L(x)$ is the result of L executing p_i to obtain output x, and Φ is a function mapping pairs of integers into positive reals, representing the trade-off between program length and memory. Via modulating Φ, one may cause this complexity criterion to weight only program length (like standard algorithmic information theory), only runtime (like the speed prior), or to balance the two against each other in various ways.

Suppose one uses the generalized complexity criterion, but looking only at programs p_i that are given S_1 as input. Then $K_{\Phi}(S_2)$, relative to this list of programs, yields an asymmetric distance $H_{\Phi}(S_1, S_2)$, which may be symmetrized as above to yield $d_{\Phi}(S_1, S_2)$. This gives a more flexible measure of how hard it is to get to one of (S_1, S_2) from the other one, in terms of both memory and processing time.

One may discuss geodesics in this sort of algorithmic metric space, just as in Fisher-Rao space. A geodesic in algorithmic metric space has the property that, between any two points on the path, the *integral of the algorithmic complexity* incurred while following the path is less than or equal to that which would be incurred by following any other path between those two points. The algorithmic metric is not equivalent to the Fisher-Rao metric, a fact that is consistent with Cencov's Theorem because the algorithmic metric is not Riemannian (i.e. it is not locally approximated by a metric defined via any inner product).

5 Three Hypotheses About the Geometry of Mind

Now we present three hypotheses regarding generally intelligent systems, using the conceptual and mathematical machinery we have built.

5.1 Hypothesis 1: Syntax-Semantics Correlation

The informational and algorithmic metrics, as defined above, are not equivalent nor necessarily closely related; however, we hypothesize that on the whole, systems will operate more intelligently if the two metrics are well correlated, implying that geodesics in one space should generally be relatively short paths (even if not geodesics) in another.

This hypothesis is a more general version of the "syntax-semantics correlation" property studied in [15] in the context of automated program learning. There, it is shown empirically that program learning is more effective when programs with similar syntax also have similar behaviors. Here, we are suggesting that an intelligent system will be more effective if memory stores with similar structure and contents lead to similar effects (both externally to the agent, and on other memory systems). Hopefully the basic reason for this is clear. If syntax-semantics correlation holds, then learning based on the internal properties of the memory store, can help figure out things about the external effects of the memory store. On the other hand, if it doesn't hold, then it becomes quite difficult to figure out how to adjust the internals of the memory to achieve desired effects.

The assumption of syntax-semantics correlation has huge implications for the design of learning algorithms associated with memory stores. All of OpenCog's learning algorithms are built on this assumption. For, example OpenCog's MOSES procedure learning component [15] assumes syntax-semantics correlation for individual programs, from which it follows that the property holds also on the level of the whole declarative memory store. And OpenCog's PLN probabilistic inference component [6] uses an inference control mechanism that seeks to guide a new inference via analogy to prior similar inferences, thus embodying an assumption that structurally similar inferences will lead to similar behaviors (conclusions).

5.2 Hypothesis 2: Cognitive Geometrodynamics

In general relativity theory there is the notion of "geometrodynamics," referring to the feedback by which matter curves space, and then space determines the movement of matter (via the rule that matter moves along geodesics in curved spacetime) [17]. One may wonder whether an analogous feedback exists in cognitive geometry. We hypothesize that the answer is yes, to a limited extent. On the one hand, according to the above formalism, the curvature of mindspace is induced by the knowledge in the mind. On the other hand, one may view cognitive activity as approximately following geodesics in mindspace.

Let's say an intelligent system has the goal of producing knowledge meeting certain characteristics (and note that the desired achievement of a practical system objective may be framed in this way, as seeking the true knowledge that the objective has been achieved). The goal then corresponds to some set of d-states for some of the mind's memory stores. A simplified but meaningful view of cognitive dynamics is, then, that the system seeks the shortest path from the current d-state to the region in d-state space comprising goal d-states. For instance, considering the algorithmic metric, this reduces to the statement that at each time point, the system seeks to move itself along a path toward its goal, in a manner that requires the minimum computational cost – i.e. along some algorithmic geodesic. And if there is syntax-semantics correlation, then this movement is also approximately along a Fisher-Rao geodesic.

And as the system progresses from its current state toward its goal-state, it is creating new memories – which then curve mindspace, possibly changing it substantially from the shape it had before the system started moving toward its goal. This is a feedback conceptually analogous to, though in detail very different from, general-relativistic geometrodynamics.

There is some subtlety here related to fuzziness. A system's goals may be achievable to various degrees, so that the goal region may be better modeled as a fuzzy set of lists of regions. Also, the system's current state may be better viewed as a fuzzy set than as a crisp set. This is the case with OpenCog, where uncertain knowledge is labeled with confidence values along with probabilities; in this case the confidence of a logical statement may be viewed as the fuzzy degree with which it belongs to the system's current state. But this doesn't change the overall cognitive-geometrodynamic picture, it just adds a new criterion; one may say that the cognition seeks a geodesic from a high-degree portion of the current-state region to a high-degree portion of the goal region.

5.3 Hypothesis 3: Cognitive Synergy

Cognitive synergy is a conceptual explanation of what makes it possible for certain sorts of integrative, multi-component cognitive systems to achieve powerful general intelligence [8]. The notion pertains to systems that possess knowledge creation (i.e. pattern recognition / formation / learning) mechanisms corresponding to each multiple memory types. For such a system to display cognitive synergy, each of these cognitive processes must have the capability to recognize when it lacks the information to perform effectively on its own; and in this case, to dynamically and interactively draw information from knowledge creation mechanisms dealing with other types of knowledge. Further, this cross-mechanism interaction must have the result of enabling the knowledge creation mechanisms to perform much more effectively in combination than they would if operated non-interactively.

How does cognitive synergy manifest itself in the geometric perspective we've sketched here? Perhaps the most straightforward way to explore it is to construct a composite metric, merging together the individual metrics associated with specific memory spaces.

In general, given N metrics $d_k(x,z), k = 1 \ldots N$ defined on the same finite space M, we can define the "min-combination" metric

$$d_{d_1,\ldots,d_N}(x,z) = min_{y_0=x, y_{n+1}=z, y_i \in M, r(i) \in \{1,\ldots,N\}, i \in \{1,\ldots,n\}, n \in \mathbb{Z}} \sum_{i=0}^{n} d_{r(i)}(y_i, y_{i+1})$$

This metric is conceptually similar to (and mathematically generalizes) min-cost metrics like the Levenshtein distance used to compare strings [14]. To see that it obeys the metric axioms is straightforward; the triangle inequality follows similarly to the case of the Levenshtein metric. In the case where M is infinite, one replaces min with inf (the infimum) and things proceed similarly. The min-combination distance from x to z tells you the length of the shortest path from x to z, using the understanding that for each portion of the path, one can choose any one of the metrics being combined. Here we are concerned with cases such as $d_{syn} = d_{d_{Proc}, d_{Dec}, d_{Ep}, d_{Att}}$.

We can now articulate a geometric version of the principle of cognitive synergy. Basically: cognitive synergy occurs when the synergetic metric yields significantly shorter distances between relevant states and goals than any of the memory-type-specific metrics. Formally, one may say that:

Definition 1. *An intelligent agent A (modeled by SRAM) displays* **cognitive synergy** *to the extent*

$$syn(A) \equiv$$

$$\int \left(d_{synergetic}(x,z) - min \left(d_{Proc}(x,z), d_{Dec}(x,z), d_{Ep}(x,z), d_{Att}(x,z) \right) \right) d\mu(x) d\mu(z)$$

where μ measures the relevance of a state to the system's goal-achieving activity.

6 Next Steps

These ideas may be developed in both practical and theoretical directions. On the practical side, we have already had an interesting preliminary success, reported in [13] where we show that (in some small examples at any rate) replacing OpenCog's traditional algorithm for attentional learning with an explicitly information-geometric algorithm leads to dramatic increases in the intelligence of the attentional component. This work needs to be validated via implementation of a scalable version of the information geometry algorithm in question, and empirical work also needs to be done to validate the (qualitatively fairly clear) syntax-semantics correlation in this case. But tentatively, this seems to be an early example of improvement to an AGI system resulting from modifying its design to more explicitly exploit the mind-geometric principles outlined here.

For an intuition regarding future potential applications of cognitive geometry to OpenCog, the reader is referred to [8] where specific cognitive synergies between the different learning algorithms in the OpenCog AGI architecture (associated with different types of memory) are discussed in an informal way. Each of these synergies may be formalized in the geometric terms outlined here, and doing so is part of our research programme going forward.

On the theoretical side, a mass of open questions looms. The geometry of spaces defined by the min-combination metric is not yet well-understood, and neither is the Fisher-Rao metric over nondimensional spaces or the algorithmic metric (especially in the case of generalized complexity criteria). Also the interpretation of various classes of learning algorithms in terms of cognitive geometrodynamics is a subtle matter, and may prove especially fruitful for algorithms already defined in probabilistic or information-theoretic terms.

References

1. Amari, S.i., Nagaoka, H.: Methods of information geometry. AMS (2000)
2. Baars, B.: In the Theater of Consciousness: The Workspace of the Mind. Oxford University Press (1997)
3. Bille, P.: A survey on tree edit distance and related problems. Theoretical Computer Science 337, 2005 (2005)
4. Dabak, A.: A Geometry for Detection Theory. PhD Thesis, Rice U. (1999)
5. Frieden, R.: Physics from Fisher Information. Cambridge U. Press (1998)
6. Goertzel, B., M. Ikl, I.G., Heljakka, A.: Probabilistic Logic Networks. Springer (2008)
7. Goertzel, B., Pennachin, C., Geisweiller, N.: Building Better Minds: Engineering Beneficial General Intelligence. In preparation (2010)
8. Goertzel, B.: Cognitive synergy: A universal principle of feasible general intelligence? (2009)
9. Goertzel, B.: Toward a formal definition of real-world general intelligence (2010)
10. Goertzel, B.e.a.: Opencogbot: An integrative architecture for embodied agi. Proc. of ICAI-10, Beijing (2010)
11. Hutter, M.: Universal AI. Springer (2005)
12. Hutter, M.: Feature dynamic bayesian networks. In: Proc. of the Second Conf. on AGI. Atlantis Press (2009)
13. Ikle, M., Goertzel, B.: Nonlinear-dynamical attention allocation via information geometry. In: Schmidhuber, J., Thorisson, K. (eds.) Subm.to AGI-11. Springer (2011)
14. Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10 p. 70710. (1966)
15. Looks, M.: Competent Program Evolution. PhD Thesis, Computer Science Department, Washington University (2006)
16. Looks, M., Goertzel, B.: Program representation for general intelligence. Proc. of AGI-09 (2009)
17. Misner, C., Thorne, K., Wheeler, J.: Gravitation. Freeman (1973)
18. Schaul, T., Schmidhuber, J.: Towards practical universal search. In: Proc. of the 3rd Conf. on AGI. Atlantis Press (2010)
19. Yi, S., Glasmachers, T., Schaul, T., Schmidhuber, J.: Frontier search. In: Proc. of the 3rd Conf. on AGI. Atlantis Press (2010)