The Übercruncher: Concept Formation by Analogy Discovery

Marc Pickett I
Cognition Robotics and Learning
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250

Concept formation allows for knowledge transfer, generalization, and compact representation. Concepts are useful for the creation of a generally intelligent autonomous agent. If an autonomous agent is experiencing a changing world, then nearly every experience it has will be unique in that it will have at least slight differences from other experiences. Concepts allow an agent to generalize experiences and other data. In some applications, the concepts that an agent uses are explicitly provided by a human programmer. A problem with this approach is that the agent encounters difficulties when it faces situations that the programmer had not anticipated. For this reason, it would be useful for the agent to automatically form concepts in an unsupervised setting. The agent should be able to depend as little as possible on representations tailored by humans, and therefore it should develop its own representations from raw uninterpreted data.

I propose to develop an algorithm which will use unsupervised analogy discovery to create a conceptual structure (also known as an ontology) that's useful for concisely characterizing data. One purpose of concept formation (and abstraction in general) is to concisely characterize a set of data (Wolff 2003). With this view, one can use *minimum description length* as a guiding principle for concept formation. I propose to use this principle to form an ontology of concepts from a collection of data. The data is a set or a stream of statements, where each statement is an ordered set of symbols. These symbols have no meaning for the program other than they're considered to be ground statements. For example, these symbols can be raw sensor data, or raw descriptions of chess games.

My hypothesis is that this algorithm will serve as a general unsupervised learning system that requires no domain knowledge or human programmed representations aside from encoding the data as a set of ordered sets of symbols. This means that this algorithm should produce useful concepts whether the data is from robot sensors, a description of a board game, or a simulation of automobile traffic. However, since the algorithm will be "bootstrapping" its representations from data, the data should be cheap and plentiful. In particular, there should be sufficient data for any useful patterns to be manifested. Since my method uses the same representation framework for raw data as it does for higher level concepts, the same algorithm can be applied iteratively to form increasingly "meta" concepts.

I hypothesize that this learning method will be able to learn some concepts, specifically concepts dealing with *relationships* between entities, that no existing method I know of can learn. The basic idea for my algorithm is that analogies are found within a set of statements, then the analogies are used to reduce the description length of the statement set by turning the analogy into a new (usually parameterized) concept, then re-encoding some of the statements in terms of this concept. In this paper, I will refer to this algorithm as The Übercruncher since it's an extension of The Cruncher (Pickett & Oates 2005).

Although cognitive scientists have discussed the centrality of analogy in cognition (Hofstadter 2001), and people have done work on structure mapping (Forbus 2001), (Marshall & Hofstadter 1996) and on subgraph isomorphism (Holder, Cook, & Djoko 1994), I know of no implementations that take a large, unsegmented set of statements and use analogy discovery to produce an ontology.

Related Work

This work builds on work on concept formation in the machine learning and artificial intelligence community, work on analogy from the cognitive science community, work on structure mapping, and work on graph isomorphism. A full review of this work is beyond the scope of this research summary, but perhaps the work most relevant to The Übercruncher is The SUBDUE system (Holder, Cook, & Djoko 1994) which compresses graphs by finding common substructures. SUBDUE is similar to The Übercruncher in that the data is not assumed to presegmented, and minimum description length is the guiding principle by which substructures are evaluated. Furthermore, SUBDUE does *induction* in the sense that frequently occurring substructures are replaced by a node that symbolizes the full substructure. However, SUBDUE uses a potentially slow beam search, on which I plan to improve by building a conceptual structure that can be used to speed up learning and classification into a current ontology. Additionally, The Übercruncher represents both concepts and meta-concepts in the same framework so that the same algorithm can be used to find analogies in both data and meta-data. Ignoring differences in representation and search strategy, SUBDUE is essentially

a strictly bottom-up version of The Übercruncher. As discussed in the next section, this approach is prone to time consuming searches and local optima. I plan to improve on this by adding a top-down component and other means for escaping local optima.

Previous Work

This work is a consequence of my research over the past several years on developmental AI. I started this work as an algorithm called "PolicyBlocks" for finding macro-actions in reinforcement learning (Pickett & Barto 2002), then I generalized it to handle attribute-value sets (The Cruncher), and then generalized it to handle relational data (The Übercruncher).

PolicyBlocks finds large and frequent subpolicies within the solutions to several Markov Decision Processes having the same transition probabilities but different reward structures. PolicyBlocks uses the principle of minimum description length and the subpolicies that minimize description length tend to be those that speed learning on future Markov Decision Processes having the same structure.

The first major extension to PolicyBlocks is The Cruncher (Pickett & Oates 2005) (not to be confused with The Übercruncher). The Cruncher extends PolicyBlocks by framing it in terms of ontology formation and minimum description length, and by adding exceptions and the ability to create multiple levels of concepts.

The Cruncher is a simple representation framework and algorithm based on minimum description length for automatically forming an ontology of concepts from attribute-value data sets. Although unsupervised, when The Cruncher is applied to the a Zoo database, it produces a nearly zoologically accurate categorization. Like PolicyBlocks, The Cruncher finds useful macro-actions in Reinforcement Learning. The Cruncher can also learn models from uninterpreted sensor data. In the same paper, we also discuss advantages The Cruncher has over concept lattices and hierarchical clustering.

I've implemented an initial version of The Übercruncher that I'll refer to as The Unterübercruncher. This algorithm shares many features of The Übercruncher including the representation framework and an algorithm for finding analogies and using these analogies to form a multi-tiered ontology. The Übercruncher will improve on The Unterübercruncher chiefly by adding mechanisms for speeding up the search process, which includes allowing for incremental data input, and by providing mechanisms for escaping local optima.

The Unterübercruncher is given a Knowledge Base formulated as a set of statements. It searches for "isomorphisms" in "contiguous" sets of statements, where a pair of statements are connected if they share a common symbol, and a set of statements is contiguous if there's a path within the statement set from every statement in it to every other statement in the set. An isomorphism or analogy is considered useful if it can be used to reduce the number of symbols needed to describe the original Knowledge Base. This measure includes the description of the analogy along with the "calls" to the analogy.

After searching, the best found analogy is used to compress the Knowledge Base, resulting in a shorter description. This entire process (finding analogies and crunching with them) is repeated until no more useful analogies are found. In practice, useful analogies are often found as parts of other analogies, which is what forms a multi-tiered ontology.

Future Work and Conclusion

The work left to do consists of making improvements to The Unterübercruncher, specifically those dealing with efficiency and getting out of local optima, and running The Übercruncher on a variety of domains to demonstrate its versatility. Additionally, I'll need to address how to evaluate the performance of The Übercruncher on these domains.

Preliminary results have given me insight as to some of the shortcomings of the current algorithm (i.e., The Unterübercruncher), namely its relative slowness, and its inability to escape local optima. Needed changes range from lower level "tweaks" (e.g., making changes to the analogy search heuristics, and otherwise addressing the algorithm's "kinks") to more fundamental algorithmic changes (such as making the algorithm incremental and using the ontology to quickly parse (i.e., segment and classify) data).

I propose The Übercruncher as a general unsupervised learning algorithm that is domain independent. I propose to demonstrate the applicability of The Übercruncher by applying The Übercruncher to a wide variety of domains with expectations that it will use the powerful mechanism of analogy to successfully create a useful ontology for each.

References

Forbus, K. 2001. Exploring analogy in the large. *The Analogical Mind: Perspectives from Cognitive Science* 23–58.

Hofstadter, D. R. 2001. Analogy as the core of cognition. *The Analogical Mind: Perspectives from Cognitive Science* 499–538.

Holder, L.; Cook, D.; and Djoko, S. 1994. Substructure discovery in the subdue system. In *Proceedings of the Workshop on Knowledge Discovery in Databases*.

Marshall, J. B. D., and Hofstadter, D. R. 1996. Beyond copycat: Incorporating self-watching into a computer model of high-level perception and analogy-making. In *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*.

Pickett, M., and Barto, A. 2002. Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Pickett, M., and Oates, T. 2005. The cruncher: Automatic concept formation using minimum description length. In *proceedings of the 6th International Symposium on Abstraction, Reformulation and Approximation (SARA 2005)*, Lecture Notes in Artificial Intelligence. Springer Verlag.

Wolff, J. G. 2003. Information compression by multiple alignment, unification and search as a unifying principle in computing and cognition. *Artif. Intell. Rev.* 19(3):193–230.