A Study in Pattern Assimilation for Adaptation and Control

Steven Hulet[1] Brandon Rohrer[2] Sean Warnick[1]

[1]Information Dynamics and Intelligent Systems Group, Brigham Young University, Provo, UT 84602, USA
[2]Intelligent Systems and Robotics Center, Sandia National Laboratories, Albuquerque, NM 87185-1010, USA

***Abstract*— Typical adaptation and control system design exploit knowledge of the system to be controlled. Nevertheless, studies in human movement, language, and other areas suggest nature develops adaptation mechanisms that do not appear to have access to such information. This study explores the possibility of such mechanisms in both natural language and simple learning environments.**

I. Motivation

Typical control theoretic formulations of adaptation tend to employ knowledge of a particular class of models of the system, or process to be controlled. This class of models is usually parameterized in such a way that an estimate of which model in the class best represents observed behavior can be systematically updated as new interactions with the system reveal more observations of its behavior. Adaptation then occurs as the response to an updated model is likewise updated to better meet the stated objectives [2].

Criticism of the above approach has included the observation that many adaptive systems we observe in nature do not seem to have access to a priori knowledge of a model class of sufficient complexity to explain the eventual success of the control system [3]. For example, as an infant learns to walk, we have not found evidence that the child retains a model of the at-least six degree of freedom inertial mechanism each one of her limbs represent (not to mention torso motion used in the crawling stages of movement and later assisting balance, etc.), and that the learning process is one where the child is essentially identifying the parameters of the compound model of the human motor system. Instead, we find evidence of a different adaptation process at work.

For example, experiments with patients recovering from stroke suggest that healthy human movement is the product of smoothly sequenced submovements, and that complexity in movement results from a richer repertoire of submovements. These observations began when [8] noted that the earliest movements made by recovering patients were "fragmented," where the fragments were each highly stereotyped. [12] showed that patient recovery appeared to be a process of blending these submovements more and more successfully, regardless of age, impairment severity, or time since stroke. It would seem that whatever library of submovement fragments exists after stroke, the recovery process assimilates them into an expanding library of increasingly complex submovements from which the patient can maneuver.

This work is supported by grants from Sandia National Laboratories, the Office of Research and Creative Activities at BYU, and the BYU Bookstore. Please direct all correspondence to Steven Hulet at srh62@cs.byu.edu.

The adaptation process observed in stroke patients, however, is supported by other human-motor studies that identify the presence of distinct submovements including slow finger movements [14], eye saccades [4], tracing constant curvature paths [1], cyclical movements [15], [5], [6], ballistic movements [11], and movements requiring high accuracy [10]. Especially noteworthy, however, is the observation that developing infants may exhibit a similar learning process of expanding a library of motion primitives through a relatively simple process of trial-and-error concatenation of existing submovements [7].

Human language patterns exhibit a similar hierarchical structure. Natural language not only appears to be composed of various substructures, such as the obvious decomposition of paragraphs into sentences, words, and letters, but it would seem that meaning is achieved through the concatenation of various language subpatterns such as phrases, idioms, and analogies that themselves operate as a single language fragment. A key question is whether human communication is composed through a process of pattern assimilation, or whether a more complex model of language is somehow retained and parametrically identified through experience.

Although both human motion and language acquisition may be explained through the suggestion that the relevant model class is "hard-wired" in the structure of our brains, and subsequent learning is, in fact, a parameter identification process, studies suggest pattern-based approaches may actually be responsible for these phenomenon. Can pattern-based approaches yield the behavioral complexity, robustness to environmental changes and uncertainty, or learning performance observed in nature? This paper takes a step towards answering that question by hypothesizing an algorithmically simple memory-driven pattern-based learning mechanism and reporting its performance, both in explaining the presence of synonyms as connectors between distinct sub-patterns in natural language, and in the adaptation of a simple pointer-robot simulation.

II. Pattern Management Processes

The definition of a pattern management process begins with the idea of primitive patterns or fragments. These initial fragments form the basis from which more complex maneuvers are composed. To understand the nature of the assumptions implicit in approaching a control or estimation problem in this way, we begin by considering vector spaces of actions, $\mathcal{U}$, states, $\mathcal{X}$, and observations, $\mathcal{Y}$, related by a nonlinear system of the form

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), u(t), t), \\ y(t) &= g(x(t), u(t), t), \end{aligned} \tag{1}$$

with $u(t) \in \mathcal{U}$, $x(t) \in \mathcal{X}$, and $y(t) \in \mathcal{Y}$. We make the usual technical assumptions to guarantee the global existence and uniqueness of solutions, say through the satisfaction of an appropriate Lipschitz condition, and note that such systems are extremely general and capable of describing a wide variety of phenomena.

A pattern-based approach to the control of the system (1) is a mechanism that can generate a sequence of actions yielding a desired observation, without knowledge of f or g. With these restrictions, the only information left to the mechanism is knowledge of how input patterns seem to generate output patterns, hence the mechanism must adapt through experience to develop an increasingly complex repertoire of action-consequence associations.

As an input stream is parsed, it is broken into smaller substrings of events that are then compared against an existing library of previously encountered substrings, and added to the library if no match is found. We start with an empty library, $L = \{\}$, which will contain ordered pairs (E_i, c_i) where c_i is the number of times event substring E_i has been encountered, and an empty buffer set $B = \{\}$, which holds event substrings which have only been encountered once. As the learner observes a stream of events, $[e_i, e_{i+1}, \ldots, e_{i+n}]$, it analyzes them in blocks of m events, beginning again with each element in turn, where the maximal substring length m is necessary for computational tractability. By remembering patterns, the learner remembers each event in the context in which it occurs. Since longer substrings contain more context, we first check to see if $E_i^m = [e_i, e_{i+1}, \ldots, e_{i+m}]$ has been seen before. If E_i^m is found in B then it has been encountered before. If E_i^m is found in L, then it's count is incremented and analysis continues with the next event, e_{i+1}, or E_{i+1}^m. If E_i^m is found in B but not L, it is added to L with a count of two. If E_i^m is not found in B, then it has never been seen before. It is added to B, and analysis continues with E_i^{m-1}, the substring of events beginning with the same event e_i but one event less in length.

When a match is not immediately found, the unmatched pattern must be remembered so as to allow the possibility of a future pattern matching against it. This can result, however, in both the patterns $[e_i, e_{i+1}, e_{i+2}]$ and $[e_i, e_{i+1}]$ being generated from the same sequence, giving the false impression that the combination $[e_i, e_{i+1}]$ occurs more frequently than it actually does. Hence the need for B, to hold all of the generated one-time patterns, and L, to hold the confirmed reoccurring patterns. Since pattern generation for a given E_i halts as soon as a match is found, the integrity of L is preserved. Using a buffer set has the added advantage of greatly reducing the cardinality of L, since many patterns only appear once.

By comparing a substring with similar substrings in the library, predictions can be made about future events and similarities between existing events can be discovered. We will demonstrate this with two examples.

III. Implications for Natural Language

Natural language is an ideal field for testing such a memory-driven pattern-based application because of the complexities involved in automatically extracting meaning from text and the large amount of readily available text, naturally divided into atomic elements. Analysis of natural language also affords the opportunity to learn an underlying structure without the added complexity of control inputs. We undertake the synonym detection problem, which is to, given a target word, automatically return other words with similar meaning. This is relevant to many areas of natural language processing because the meaning of a sentence or document can easily be missed if none of the topic words are repeated, but a variety of synonyms used instead.

We first build a library by reading untagged text, using words as events. To determine word similarity, the user first enters a query. The library is then searched for every pattern containing the query; these patterns are referred to as query patterns. For each query pattern the word immediately preceding the query (the pre-word) and the word immediately following the query (the post-word) are located. Next, the library is again searched, this time for every pattern containing the pre-word followed sometime thereafter by the post-word. These are synonym patterns. All phrases found occurring between the pre-word and the post-word in synonym patterns are candidate synonyms. This poses a minimum requirement of contextual similarity– the two words closest to the query must also be the closest to any candidate synonym. Figure 1 shows an example of how the query "great" can generate the candidate synonym "very large" through the comparison of two patterns.

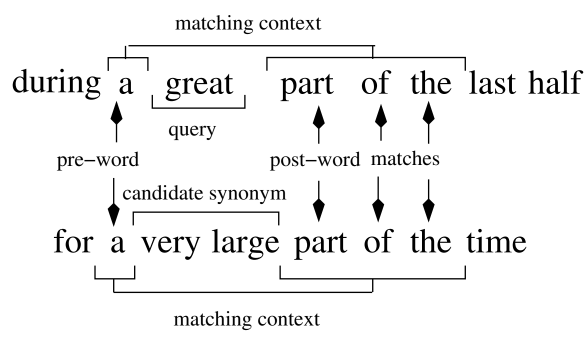

Fig. 1. The similarity of two phrases can be judged by the amount of matching context.

The weighting heuristic seeks to give candidate synonyms found in contexts very similar to that of the query greater weight; the more similar the contexts, the more similar the words. The similarity of context is judged by the number of additional words, or matches, common between the query pattern and the synonym pattern, beginning with the pre-word and post-word and counting outward (see figure 1). Candidate synonyms are sorted by the number of matches; in the event of a tie, the candidate synonyms are sorted by the number of different contexts in which they appear. Two words found in a large number of identical contexts are usually more similar than two words which appear in only one common context.

We tested our learner after reading just over 25 million words from a diverse selection of texts. Table 2 shows the first ten results returned on some sample queries, sorted by the above method. The results indicate that the learner is able to draw conclusions about strongly related words simply by observing how the language is used. The learner successfully generates groups of similar words using no predefined lookup tables, part-of-speech tags, or other previous knowledge of the language structure. The results are returned in an order intuitive to humans and suggest an accurate relative degree of similarity between words. Antonyms, while clearly not synonyms, are strongly related word pairs, and appear in similar contexts. Note that "small" ranks highly as similar to "large." By allowing candidate synonyms to be of any length, we are also able to capture synonyms requiring more than one word to express.

These results support the idea that similar words appear in similar contexts [9]. Consistently, the greater the contextual match up between the query phrase and the synonym phrase, the greater the similarity of the phrases. As more text is read the accuracy of the results is expected to improve even further.

Once the library has been constructed, it can also be used for a number of other natural language analysis tasks. Phrase completion is easily performed by checking a seed phrase against the library, and determining what words or phrases usually follow. This process can be iterated upon, resulting in novel text generation. Checking a user's text against the library provides a useful spellchecking tool, which, since it is entirely context based, has no difficulty detecting real-word spelling and other grammatical errors.

IV. Adaptation For Control

This same learner can easily be applied to robotic movement control with only minor modifications. One difference is the addition of control inputs, which allows the learner to interact with its environment. We created a virtual pointer robot with one degree of freedom [13]. It has two possible actions, turn 10^o clockwise (R) and 10^o counterclockwise (L), and its world state corresponds to an angle degree measured in 10^o bins $(1 \quad 36)$. The robot is entirely independent of the world, capable only of executing a turning action and observing the world state, and begins its existence with no previous knowledge of how its environment functions. Event sequences now take the form of alternating actions and observations. We allowed the robot to interact with six different environments:

SEVEN	LARGE	SUGAR
seven	large	sugar
five	great	flour
two	small	fruit
four	considerable	butter
three	certain	salt
ten	very large	water
twelve	good	mace
fifteen	very small	meat
twenty	vast	cream
fifty	larger	brandy
FEET	FATHER	ROAD
feet	father	road
face	mother	river
heart	wife	street
head	son	table
side	head	fire
house	life	hill
lips	voice	head
work	face	house
hands	heart	room
back	name	lake

Fig. 2. Returned synonyms on sample queries. Only the first 10 results are shown.

- **Simple system**: Measurement states 136 and command events R and L as described previously.
- **Hard stop**: Same as the simple system, but with a "hard stop" inserted at 0^o, prohibiting continuous rotational movement.
- **Sensory state scramble**: Same as the simple system, but after 5000 trials, the numerical labels for sensory states are renamed 136 in random order, making all prior learning inapplicable and misleading.
- **Random error**: Same as the simple system, but with up to 5^o of random error added to each command event, resulting in movements of between 5^o and 15^o. With measurement resolution limited to 10^o, the error will express itself as measurement states being either skipped or unchanged when a command is issued.
- **Random delays**: Same as the simple system, but each command event has a 50% chance of being delayed and executed at the instant the next command is issued. As a result, when a command is issued, zero, one, or two command events may actually take place.

In each case, the learning agent generated random command events and attempted to predict the results before executing the command. Predictions were generated by

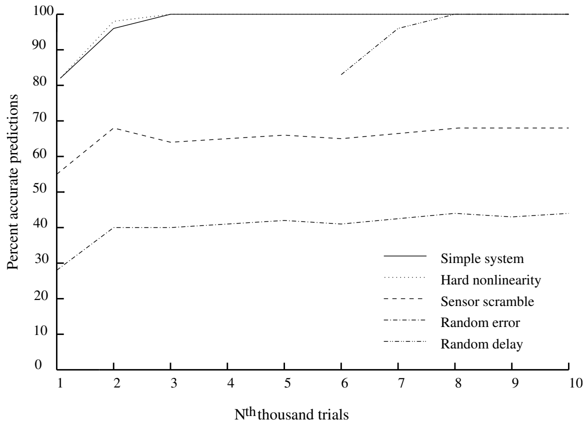

Fig. 3. Simulation performance for the pointer robot in six environments.

searching through previously observed patterns for instances containing a portion of the most recent event history. Patterns that matched a longer portion of the event history were favored heavily. Patterns that were observed recently or that had been observed many times were also favored. Once a pattern was selected a prediction was obtained by reading what happened next when the situation had been encountered previously. In each condition, the learning agent began with a clean slate; that is, there were no previously observed experiences upon which to build. As a result, lack of prior experience made it impossible for the agent to offer a prediction in some cases. These were counted as unsuccessful predictions.

The results of the simulations are shown in figure 3. As shown in the plot, the learning agent achieved 100% accuracy in the simple system after 2000 trials. The learning agent showed similar performance in the presence of a hard stop. In both these conditions, the performance of the system is deterministic, allowing correct predictions at every time step.

Scrambling the sensory state labels changed the system fundamentally, making the probability of encountering a previously observed pattern small. Learning essentially began from scratch, and the initial learning transient was repeated after scrambling. This resulted in a marked decrease in performance initially, but the agent quickly recovered and predicted the last 1000 trials perfectly.

The introduction of random noise into the movement amplitude made perfect prediction impossible. The noise amplitude exactly corresponded to the resolution of the position measurement, 10^o. As a result, knowledge of the current position allowed prediction the subsequent position with an accuracy of only 50%. With a longer event history, it was possible to increase the accuracy, but only to a certain extent. The learning agent began with a prediction accuracy slightly higher than 50%, and gradually it increased to near 70%.

Random time delays introduced the possibility that zero, one, or two command events might be executed at once. With a 50% probability of delay, at any given time step there was a 25% chance that no command would be executed, a 25% chance that two commands would be executed simultaneously, a 25% chance that the previous command alone would be executed, and only a 25% chance that the current command alone would be executed. As a result, even once the behavior of this simple system is learned, only a 25% success rate can be expected with no knowledge of prior events. However, with a complete knowledge of prior events, it was possible to infer whether the prior command event had been executed, allowing a prediction accuracy of 50%. The learning agent began with prediction accuracy slightly higher than 25%, and that accuracy climbed to just over 45% after 10,000 trials.

V. Conclusion

A simple pattern-based adaptation mechanism was used for both synonymity detection in natural language and basic control tasks in a virtual pointer-robot simulation. Results suggest that under certain conditions, memory-based methods may perform well without prior knowledge of their environment.

References

[1] W. Abend, E. Bizzi, and P. Morasso. Human arm trajectory formation. *Brain*, 105:331–348, 1982.
[2] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1995.
[3] R. A. Brooks. Intelligence without representation. Number 47 in Artificial Intelligence, pages 139–159. 1991.
[4] H. Collewijn, C. J. Erkelens, and R. M. Steinman. Binocular coordination of human horizontal saccadic eye movements. *J Physiol*, 404:157–182, 1988.
[5] E. R. F. W. Crossman and P. J. Goodeve. Feedback control of hand-movements and fitts law. *Quarterly Journal of Experimental Psychology*, 35A:251–278, 1983.
[6] J. A. Doeringer. An investigation into the discrete nature of human arm movements. *Unpublished doctoral dissertation, Massachusetts Institute of Technology*, 1999.
[7] C. von. Hofsten. Structuring of early reaching movements: A longitudinal study. *Journal of Motor Behavior*, 23(4):280–292, 1991.
[8] H. I. Krebs, M. L. Aisen, B. T. Volpe, and N. Hogan. Quantization of continuous arm movements in humans with brain injury. *Proceedings of the National Academy of Sciences USA*, 96:4645–9, 1999.
[9] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.
[10] T. E. Milner. A model for the generation of movements requiring endpoint precision. *Neuroscience*, 49:365–374, 1992.
[11] P. Morasso and F. A. Mussa-Ivaldi. Trajectory formation and handwriting: A computational model. *Biological Cybernetics*, 45:131–142, 1982.
[12] B. Rohrer, S. Fasoli, H. I. Krebs, B. Volpe, W.R. Frontera, J. Stein, and N. Hogan. Submovements grow larger, fewer, and more blended during stroke recovery. *Motor Control*, 8(4):472–483, 2004.
[13] B. Rohrer and S. Hulet. A discrete-time model of human motor learning.
[14] A. B. Vallbo and J. Wessberg. Organization of motor output in slow finger movements in man. *Journal of Physiology*, 469:673–691, 1993.
[15] R. S. Woodworth. The accuracy of voluntary movement. *Psychology Review Monogr Suppl.*, 1899.