
16.412J Problem Set #1

Part B: Your Cognitive Robot

It seems inevitable that mechanical agents will continue to precede human explorers in novel and/or hostile environments, particularly as the hardware and software capabilities of these agents improve. However, time delays in communication currently impede high-fidelity telepresence in remote locations such as the Martian surface. Such fidelity can be largely improved to the extent that lower-level behaviors of human explorers can be captured and integrated into autonomous agents, so that only higher-level behaviors—with time scales comparable to communication intervals—require active human involvement.

Programming these lower-level behaviors directly has proven costly and error-prone. Learning systems present an alternative, more flexible, and potentially more cost-effective approach to achieving integrated low-level autonomous capabilities. By observing human low-level behaviors under similar conditions, they can gradually anticipate certain classes of human actions under most circumstances, providing substantial savings in communication. This approach benefits from continuous, gradual improvement: initial behaviors can be developed based on simulations in local analog environments, then continually refined based on human input in the actual remote environment. The mechanical agent grows increasingly capable over time, always providing the ability to test the next stage of development before committing it to autonomous control. Confidence is improved by the fact that human operators and the remote agent work from precisely the same input data, to the extent feasible. Moreover, with sufficient robustness, any undesirable actions taken in exceptional circumstances can be reversed and corrected by human operators.

I envision a cognitive robot capable of learning by observation some of the principal but relatively unsophisticated tasks associated with remote exploration, such as navigation, interest feature identification, and diagnostic feature inspection. Ideally, such an agent could be trained in inexpensive Earth-based simulations and then deployed to remote locations like the Martian surface, where its performance would improve over time.

Part A: Topics of Fascination

An agent like the one described above will draw upon three core areas of research:

1. Knowledge Representation. Clearly the agent will require mechanisms for representing the behavior rules it has learned, but the main applications of well-developed representational schemes will lie in effectively managing the input data from its environment. A hierarchical ontology, likely containing several specialized elements, will be required to abstract visual and other sensory stimuli to a level providing appropriate inputs to decision algorithms. Due to the complexity of these elements of the system, I plan to avoid the complexities of real-world knowledge representation and develop decision algorithms based on data from "toy" examples.

2. Statistical Learning. The agent must be able to observe the behaviors of a human operator in an environment and develop decision strategies based on patterns extracted from this sample data. The agent must also be able to dynamically incorporate new sample data during its operations and incorporate new inferences into its strategic model. The complexity of the model should be optimally adjusted to the volume of training data available, and should be independently controllable by the method based on resource constraints.

3. Probabilistic Reasoning. While arguably a component of statistical learning, I regard the agent's actual decision procedures as a somewhat distinct element of the system, which may be handled independently from the formulation of those procedures based on learning. The execution of these decision procedures will almost certainly require probabilistic inference, although the specific methods involved will depend upon the strategies produced by the learning framework.

Part C: Making Your Cognitive Robot

I regard statistical learning as the most fundamental element of the system described above, and most worthy of further investigation: by comparison, knowledge representations are more specific and complex, and probabilistic reasoning methods are better understood. This agent would ideally be able to perform uninformed pattern recognition—that is, without the aid of pre-defined model hypotheses—on behaviors exhibited by humans based on compatible input data streams. While the system may rely initially upon more fundamental models of various phenomena, behaviors specific to the exploration activities should be entirely learned, so that they can be dynamically modified with ease. For feasibility, the system should be able to identify and adapt its effective prioritization of available indicators in evolving its decision strategies. The method(s) used should be dynamically extensible as the training data set is enlarged.

Part D: Researching a Critical Reasoning Method

The primary motivations for algorithm selection will come from flexibility—the ability to recognize general patterns and estimate general functions—and feasibility—the ability to perform classification and decision-making using a compact strategic representation. Hence the general thrust of this effort will lie in searching for powerful generalizations of more elementary schemes (e.g., parameter decision trees constructed by ID3). This intuition motivates discussion of the following three papers from the recent literature:

1. Weiss SM, Indurkha N. 1995. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research* 3:383-403.

The authors initially show how to generalize traditional discrete decision trees used for classification to regression trees used for functional estimation. Like decision trees, regression trees perform partitioning based on a disjunctive normal form strategy, which has advantages for clarity of knowledge organization and traceability to features. However, for pattern recognition applications, they propose a more general rule-based approach to regression, which eliminates the DNF constraint and can potentially find much more compact representations. This can be important for large spaces, and can potentially find rules with substantially clearer interpretations. Through a number of real-world examples, they show that rule-based regression algorithms using numerical optimization techniques can significantly outperform tree-based methods both in performance and speed. They also show how this approach can be effectively combined with partitioning and

nearest-neighbor methods (e.g., bounding pseudo-classes on the basis of a fixed neighborhood population) in order to improve performance still further. They also pursue sample storage compression enhancements with some success.

The methods presented here move in the desired direction in terms of generalizing the approach to classification and estimation, and provide concrete algorithms and examples demonstrating their effectiveness in certain situations not unlike exploration behavior. However, the fundamental methodology still relies upon complete storage of training samples and a somewhat discretized pseudo-classification approach to pattern decomposition. That is, rule-based regression may not be general enough to provide the kind of dynamic adaptability and scaling to training data desirable in our agent. The other papers in this collection introduce and develop a more general approach based on "support vectors," which may provide a better high-level framework for our applications.

2. Boser BE, Guyon IM, Vapnik VN. 1992. A training algorithm for optimal margin classifiers. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, Pittsburgh, PA. 144-152.

These authors take a step beyond both tree-based and rule-based parametric decision methods by devising a method to effectively reparametrize the space of observations according to the most useful global indicators. That is, they construct a dynamically generated basis for the input space using "support vectors" optimally chosen to maximize the resolution of the boundary between decision classes. This provides significant improvement on traditional regression-based approaches, which tend to smooth over any atypical patterns as represented in the original input basis. Their training algorithm also grows dynamically with new input data, while incorporating many other linear and nonlinear methods as special cases. Moreover, the authors show how to construct a dual space representation (of reduced dimension) for the actual decision kernel, which allows the underlying quadratic optimization problem to be solved efficiently using standard numerical techniques. The authors demonstrate empirical performance on many classical pattern recognition problems (such as handwritten digit recognition) significantly exceeding other leading algorithms, in some cases even those with pre-defined task-specific models for those problems.

This approach demonstrates both generality for uninformed pattern recognition and dynamic adaptation and scaling to enlarging training data sets that we desire for our agent. By comparison to the Weiss et al. paper, however, it still does not demonstrate applications to functional estimation as well as pattern recognition, which will also be a key element of the agent's exploratory behavior. For this further development, we turn to the last paper in this collection.

3. Vapnik V, Golowich S, Smola A. 1997. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: *Neural Information Processing Systems, Vol. 9*. MIT Press, Cambridge, MA.

In this paper, the authors extend the support vector method to three critical classes of applications: function approximation, regression estimation (the principal subject of Weiss et al.), and signal processing. The primary content of the paper consists of explicit mathematical algorithms for carrying out each of these tasks in a generalized fashion, but relatively simple example applications are demonstrated for each one, with sufficient realism to provide indicators about performance. The authors show in general how the reduced effective dimensionality of the support vector basis translates into lower complexity in all these areas; that is, complexity is driven by the desired complexity of the result rather than the complexity of the initial parametrization of the problem. Performance bounds are generally impressive, demonstrating feasibility for a wide range of problems.

Ultimately, it seems likely that our agent may be built upon a support vector machine architecture, which may or may not rely on some initial modeling knowledge, depending upon the simplicity of the "toy" environment chosen for development and testing. For explicitly discrete classes of problems, rule-based methods may yet be of some utility, since they generally afford simpler symbolic representations.

Part E: A Simple Project for Your Cognitive Robot

One potentially feasible project for developing this framework might pursue a specific two-dimensional visual task, using human training data captured by a PC-based input device (such as a mouse). To truly demonstrate the success of the architecture, the task should be sufficiently complex as to defy simple rule-based description, as might be encoded directly in software by a programmer well-versed in the activity. A plausible analogy with exploration tasks (navigation and/or feature recognition), such as locating a target, would be preferable though not essential. It might be beneficial for pre-defined agent models and sample training data to be produced by different people, so as to enforce the uninformed pattern recognition aspect of the problem.