# SPATIAL INTENTION RECOGNITION USING OPTIMAL MARGIN CLASSIFIERS

Thomas Coffee

Shuonan Dong

Shen Qu

## Introduction

The high costs of human spaceflight operations favor large investments to optimize astronauts' time usage during extravehicular activity. These have included extensive expenditures in training, tool development, and spacecraft design for serviceability. However, astronauts' space suits themselves still encumber more than aid, a focus of several current research programs. Potential improvements face tight integration between suits and astronaut activities, resulting in many mechanical and computational challenges.

One major area of work aims to alleviate the difficulties of conducting precise or prolonged movements within a pressurized garment. Powered prosthetic assistance may provide a solution to this problem, but creates key operational challenges. Standard digital or verbal user command interfaces may prove incompatible with such devices, limited by low bandwidth and nonintuitive control structures. Tactile control using, for example, hand or finger gestures seems far more suitable for controlling mechanical effectors, providing high speed and intuitive spatial relationships between command signals and desired actions.

Flexibility and robustness in controllers like these will likely require personalized command recognition tailored to individual astronauts. The need for speed and natural facility will make this capability even more indispensable than in, say, speech recognition. Command recognition systems should dynamically adjust their interpretation rules as training data is accumulated, improving their precision and following long-term trends as astronauts develop their working behaviors throughout a career's worth of extravehicular activity.

In this project, we propose a relatively simple gesture-based spatial command recognition system as an analog to more advanced systems suitable for augmenting extravehicular activities with robotic assistance. We aim initially to achieve discrete pattern recognition, with a possible extension to continuous parameter spaces, which may ultimately find favor in many spatial applications.

## Problem Statement

We propose a software agent capable of identifying spatially motivated commands among a finite set indicated by short two-dimensional gestures within the continuous movement stream of a pointing device such as a computer mouse. The agent will construct optimized interpretation rules based on training data sets corresponding to single human users over a period of time, with identifying rules adjusted dynamically during further use. The system may be extended to allow command spaces parameterized by continuous variables. It may also allow users to refine agent interpretations post facto by providing optional explicit clarification after initial training is completed.

## Background Material

Here we provide some background from the most relevant recent literature on statistical learning. This description is adapted from earlier review by the first author. We aim to go beyond elementary classification methods such as binary decision trees to achieve more powerful yet still feasible pattern recognition (discrete case) and functional estimation (continuous case) capabilities.

Weiss et al. (1995) 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 (DNF) 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. 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.

By contrast, Boser et al. (1992) take a step beyond both tree-based and rule-based parametric decision methods by devising a method to effectively reparameterize 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 Weiss et al., however, it still does not demonstrate applications to functional estimation as well as pattern recognition, which will be essential for operating with continuous-parameter command spaces.

Vapnik et al. (1997) extend the support vector method to three major classes of applications including regression estimation. 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. Given its well-demonstrated utility in both discrete recognition and continuous estimation, we currently plan to build our agent upon a support vector machine architecture.

## Technical Approach

Two-dimensional cursor input streams will be captured during one or more continuous sessions per individual user using a standard PC pointing device (such as a mouse). Visual feedback may or may not be provided to the user indicating the movement history of the pointing device, though more likely not given the speed of application at which we are aiming. The user will also be provided a means to indicate interpretations to be assigned to command gestures during training; the desired precision of these indications (with respect to gesture endpoints) has yet to be determined.

Temporal information will be preserved to maximize the interpretive capability of the agent; if extensions for explicit refinement post facto are pursued, information on session ordering will also be maintained. Input data streams will be sampled at lower than maximum resolution in order to improve time and storage requirements for processing (we intend to archive all input data for later analysis even though not strictly necessary for applying the algorithms discussed above).

Extending the system to support continuous-parameter command spaces (if feasible) will most likely require some degree post facto refinement to maintain convergence on user intentions. In this case, the user will have a consistently available mechanism for making adjustments to interpreted commands based on visual feedback, most likely through a separate pointing device or interface. The details of these potential extensions have yet to be determined.

Pattern recognition and functional estimation software will be based on the support vector algorithms described in the papers cited above. The software will be implemented in *Mathematica* 5.1, using the *GUIKit* package built on *J/Link* to construct a user interface running on a local Java Virtual Machine.

## Iterative Development

We propose two iterations of spiral development in order to gain useful insights for final product development and maximize the chances of success.

**First Iteration.** We plan to develop a simplified version of the system incorporating minimal functionality in all major components. This version will perform a binary discrete pattern recognition (between a gesture pattern and its complementary input space) based on high-quality prefabricated training data. Here we intend to verify the basic soundness of the algorithm and implementation strategies.

**Second Iteration.** We plan to incorporate multiple patterns and multiple individuals generating real training data sets. Depending upon time available and challenges encountered, we may pursue extensions to continuous-parameter command spaces and post facto user refinement of interpretation rules.

## Project Plan

We plan to develop the core elements of our agent and test and analyze its capabilities as a tightly integrated team, though individual software implementation and testing tasks may be partitioned among the group as needed. The table below outlines our anticipated development schedule:

| Date of Completion | Activity |
|---|---|
| Sunday 4/17 | Develop software architecture:<br>• identify major components and interfaces<br>• detail implementation strategy for major components<br>• assign responsibilities for component development |
| Sunday 4/24 | Develop and test software interfaces:<br>• establish data protocols between major components<br>• demonstrate compatibility via fabricated data sets |
| Sunday 5/1 | Complete first iteration:<br>• demonstrate binary discrete pattern recognition<br>• verify algorithm performance by comparison to predictions |
| Sunday 5/8 | Complete second iteration:<br>• demonstrate command recognition for multiple individuals<br>• develop agent extensions as time allows<br>• analyze algorithm performance and applicability |
| Sunday 5/11 | Document final results:<br>• organize results, analysis, and remaining extensions<br>• prepare final report and presentation materials |

## References

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

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.

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.