Scaling up Inverse Reinforcement Learning through Instructed Feature Construction

Tomas Singliar
Dragos D. Margineantu
Boeing Research & Technology
P.O. Box 3707, M/C 7L-44
Seattle, WA 98124-2207
{tomas.singliar,dragos.d.margineantu}@boeing.com

Abstract

Inverse reinforcement learning techniques (IRL) (Ng & Russell, 2000) provide a foundation for detecting abnormal agent behavior and predicting agent intent through estimating its reward function. Unfortunately, IRL algorithms suffer from the large dimensionality of the reward function space. Many applications that can benefit from an IRL-based approach to assessing agent intent involve a domain expert or analyst. This paper proposes a procedure for scaling up IRL by eliciting good IRL basis functions from the domain expert.

1 Introduction

Analyzing large volumes of intelligence data has become an overwhelming task. The mountains of data are a good match for data-hungry machine learning algorithms, presenting automation opportunities. The center of focus in intelligence and surveillance (ISR) is the behavior of people, who are often well approximated by the rational agent assumption. Inverse reinforcement learning (IRL) methods learn the reward function of an agent from past behavior of the same or similar agent and thus provide means for detecting "abnormal" behavior. Once the reward function is approximately known, solving a form of MDP will predict agent's actions. Unfortunately, IRL algorithms suffer from the large dimensionality of the reward function space.

The traditional solution is to approximate the reward or value function by a linear combination of basis functions, which transforms the problem into one of designing a compact set of good basis functions. Basis functions here fill the role that features play in supervised learning, that is, mapping the perceptual space into a simpler decision space. Applying experience from the DARPA Bootstrapped Learning program, whose hypothesis is that it should be easier to learn from a benevolent teacher than from the open world, we design a procedure for eliciting the basis functions from the domain expert – the analyst. Starting with a minimal set of basis functions, the system presents the analyst with anomalies which the analyst then typically explains away by indicating domain features that the algorithm should have taken into account. The value function approximation basis is expanded and the process iterates until only true positives remain.

We applied our proposed technique to Moving Target Indicator (MTI) data. MTI is an application of Doppler radar which allows simultaneous tracking of multiple moving objects, typically vehicles.

2 Approach

We employ the IRL paradigm to characterize the behavior of a class of agents compactly, by means of bounds on the reward function. The principal difficulty of IRL is the dimensionality of the reward function space. Traditionally, the dimensionality is reduced by assuming a linear approximation to the reward function

$$R(s,a) = \sum_{i=1}^{k} w_i \phi_i(s,a)$$

The optimization problems are then greatly simplified and the difficulty shifts into finding good basis functions ϕ_i.

We test our approach on a collection of GMTI (Ground Moving Target Indicator) data. MTI indicator is a technology based on Doppler or synthetic aperture radar which is capable of capturing movement tracks of a large number of vehicles simultaneously. The data is discretized into a grid and mapped onto nodes of a state space graph, where each grid cell corresponds to a node as shown in *Figure 1*. Cells that were never occupied are discarded to reduce the state space. For simplicity and to avoid data sparsity problems, we assume that all observed action sequences were performed by agents that share the reward function (a class of identical agents).

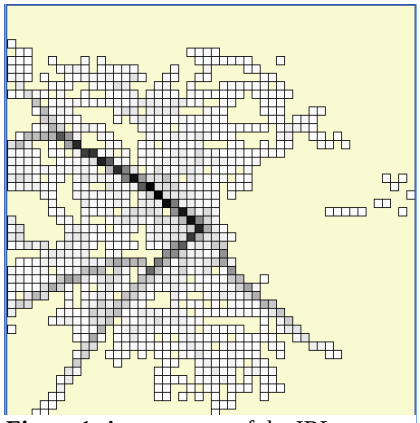

Figure 1. A state space of the IRL problem for the GMTI task, with state visitation frequencies.

Further simplifying assumptions that assure tractability are determinism in action outcomes and perfect rationality. This creates a very simple problem solvable using pure linear programming as opposed to a more complex gradient descent learning procedure that arises when a distribution is placed on behavior that penalizes for deviation from optimal action, e.g. as in (Ziebart et al., 2009).

In order to have a compact yet representative feature set, the best solution is to build the basis function from the domain expert's knowledge. However, domain experts are rarely inclined to build mathematical abstractions of what they view as common sense knowledge. A novel aspect of our work is that we elicit this knowledge in the course of a natural problem-focused "dialog" with the domain expert. The analyst observes the alerts that the system produces and either acts on the true positives, or has the option to explain to the system why the alert is false. When the analyst provides an explanation, a new basis function is created and the learning problem is repeated. The analyst also has the option of excluding the example (action sequence) from learning as "agent in a different class".

The mechanism for detecting anomalies relies on estimating the reward function of a single agent represented by the track. Tracks are processed in an online fashion. If no reward function can make the behavior consistent with the reward bounds derived from previously seen tracks, an alert is created. This will be detected simply by the corresponding linear program being infeasible. Otherwise, the track is added into the repository. An alert can also be created if the reward function satisfies interestingness criteria predefined by the analyst.

Example. In the schematic of *Figure 2*, two vehicles come into the area, briefly stop at an intersection, and continue their travel. The system flags their meeting as intentional, rather than coincidental, because travel incurs a positive cost and shorter routes to their destinations exist. When presented with the alert, the analyst considers the tracks and dismisses the alert, using a menu interface to specify that "it is relevant that the meeting point is at a gas station". This creates a new basis function which takes on a high value at any gas station and will explain away further false alerts of a similar nature. To create a basis function, one of a number of templates is selected and parameterized; e.g. a Gaussian function where the standard deviation is the parameter, and centered on the feature of interest (e.g., the market). Note that this example requires that the analyst previously specified a concept of meeting through a basis function expressing proximity.

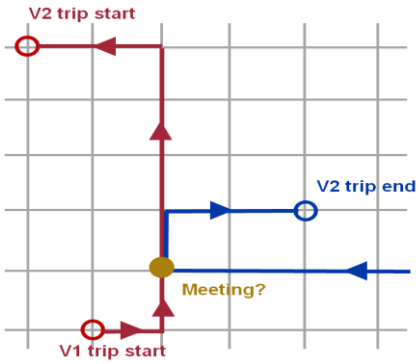

Figure 2. Actions of two vehicles that stop at the same time at a certain location.

Acknowledgements

The research presented in this paper was funded by the DARPA *Bootstrapped Learning* program and by Boeing internal research programs.

References

Abbeel, P. & Ng., A.Y. (2004), Apprenticeship learning via inverse reinforcement learning, *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML 2004.

Baker, C.L., Tenenbaum, J.B., Saxe, R.R. (2007), Goal Inference as Inverse Planning, *Proceedings of the Twenty-Ninth Annual Conference of the Cognitive Science Society*, pp.779-784.

Baker, C.L., Saxe R.R., Tenenbaum, J.B. (2009), Action understanding as inverse planning. *Cognition, 113*, pp. 329-349.

Ng, A.Y. & Russell, S. (2000), Algorithms for inverse reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML 2000.

Ramachandran, D. & Amir, E. (2007), Bayesian Inverse Reinforcement Learning. *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI 2007, pp.2586-2591.

Ziebart, B.D., Mass A., Bagnell, J.A., Dey, A.K. (2008), Maximum entropy inverse reinforcement learning. *Proceedings of AAAI 2008*, pp. 1433-1439.

Ziebart, B.D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert M., Dey, A. K., Srinivasa, S. (2009), Planning-based Prediction for Pedestrians. *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3931-3936.