Maximum Entropy Inverse Reinforcement Learning

Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, and **Anind K. Dey**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bziebart@cs.cmu.edu, amaas@andrew.cmu.edu, dbagnell@ri.cmu.edu, anind@cs.cmu.edu

Abstract

Recent research has shown the benefit of framing problems of imitation learning as solutions to Markov Decision Problems. This approach reduces learning to the problem of recovering a utility function that makes the behavior induced by a near-optimal policy closely mimic demonstrated behavior. In this work, we develop a probabilistic approach based on the principle of maximum entropy. Our approach provides a well-defined, globally normalized distribution over decision sequences, while providing the same performance guarantees as existing methods.

여기선 효용기반!

We develop our technique in the context of modeling real-world navigation and driving behaviors where collected data is inherently noisy and imperfect. Our probabilistic approach enables modeling of route preferences as well as a powerful new approach to inferring destinations and routes based on partial trajectories.

Introduction

In problems of *imitation learning* the goal is to learn to predict the behavior and decisions an agent would choose– e.g., the motions a person would take to grasp an object or the route a driver would take to get from home to work. Capturing purposeful, sequential decision-making behavior can be quite difficult for general-purpose statistical machine learning algorithms; in such problems, algorithms must often reason about consequences of actions far into the future.

A powerful recent idea for approaching problems of imitation learning is to structure the space of learned policies to be solutions of search, planning, or, more generally, Markov Decision Problems (MDP). The key notion, intuitively, is that agents act to optimize an unknown reward function (assumed to be linear in the features) and that we must find reward weights that make their demonstrated behavior appear (near)-optimal. The imitation learning problem then is reduced to recovering a reward function that induces the demonstrated behavior with the search algorithm serving to "stitch-together" long, coherent sequences of decisions that optimize that reward function.

We take a thoroughly probabilistic approach to reasoning about uncertainty in imitation learning. Under the constraint of matching the reward value of demonstrated behavior, we employ the principle of *maximum entropy* to resolve the ambiguity in choosing a distribution over decisions. We provide efficient algorithms for learning and inference for deterministic MDPs. We rely on an additional simplifying assumption to make reasoning about non-deterministic MDPs tractable. The resulting distribution is a probabilistic model that normalizes globally over behaviors and can be understood as an extension to chain conditional random fields that incorporates the dynamics of the planning system and extends to the infinite horizon.

Ambiguity를 줄이기 위해서 principle of maximum entropy.

운전자의 실주행 데이터를 모델링하는데 활용!

Our research effort is motivated by the problem of modeling real-world routing preferences of drivers. We apply our approach to route preference modeling using 100,000 miles of collected GPS data of taxi-cab driving, where the structure of the world (i.e., the road network) is known and the actions available (i.e., traversing a road segment) are characterized by road features (e.g., speed limit, number of lanes). In sharp contrast to many imitation learning techniques, our probabilistic model of purposeful behavior integrates seamlessly with other probabilistic methods including hidden variable techniques. This allows us to extend our route preferences with hidden goals to naturally infer both future routes and destinations based on partial trajectories.

A key concern is that demonstrated behavior is prone to noise and imperfect behavior. The maximum entropy approach provides a principled method of dealing with this uncertainty. We discuss several additional advantages in modeling behavior that this technique has over existing approaches to inverse reinforcement learning including margin methods (Ratliff, Bagnell, & Zinkevich 2006) and those that normalize locally over each state's available actions (Ramachandran & Amir 2007; Neu & Szepesvri 2007).

Background

In the imitation learning setting, an agent's behavior (i.e., its trajectory or path, ζ, of states s_i and actions a_i) in some planning space is observed by a learner trying to model or imitate the agent. The agent is assumed to be attempting to optimize some function that linearly maps the features of each state, $\mathbf{f}_{s_j} \in \Re^k$, to a state *reward value* representing the agent's utility for visiting that state. This function is parameterized by some *reward weights*, θ. The reward value of a trajectory is simply the sum of state rewards, or, equivalently, the reward weight applied to the path *feature* *counts*, $\mathbf{f}_\zeta = \sum_{s_j \in \zeta} \mathbf{f}_{s_j}$, which are the sum of the state features along the path.

$$\text{reward}(\mathbf{f}_\zeta) = \theta^\top \mathbf{f}_\zeta = \sum_{s_j \in \zeta} \theta^\top \mathbf{f}_{s_j}$$

The agent demonstrates single trajectories, $\tilde{\zeta}_i$, and has an expected empirical feature count, $\tilde{\mathbf{f}} = \frac{1}{m} \sum_i \mathbf{f}_{\tilde{\zeta}_i}$, based on many ($m$) demonstrated trajectories.

Recovering the agent's exact reward weights is an ill-posed problem; many reward weights, including degeneracies (e.g., all zeroes), make demonstrated trajectories optimal. Ratliff, Bagnell, & Zinkevich (2006) cast this problem as one of *structured maximum margin prediction* (MMP). They consider a class of loss functions that directly measure disagreement between an agent and a learned policy, and then efficiently learn a reward function based on a convex relaxation of this loss using the structured margin method and requiring only oracle access to an MDP solver. However, this method suffers from some significant drawbacks when no single reward function makes demonstrated behavior both optimal and significantly better than any alternative behavior. This arises quite frequently when, for instance, the behavior demonstrated by the agent is imperfect, or the planning algorithm only captures a part of the relevant state-space and cannot perfectly describe the observed behavior.

MMP의 단점
: 이게 무슨 뜻일까?

Abbeel & Ng (2004) provide an alternate approach based on Inverse Reinforcement Learning (IRL) (Ng & Russell 2000). The authors propose a strategy of matching *feature expectations* (Equation 1) between an observed policy and a learner's behavior; they demonstrate that this matching is both necessary and sufficient to achieve the same performance as the agent if the agent were in fact solving an MDP with a reward function linear in those features.

Feature matching

$$\sum_{\text{Path } \zeta_i} P(\zeta_i) \mathbf{f}_{\zeta_i} = \tilde{\mathbf{f}} \qquad (1)$$

Unfortunately, both the IRL concept and the matching of feature counts are ambiguous. Each policy can be optimal for many reward functions (e.g., all zeros) and many policies lead to the same feature counts. When sub-optimal behavior is demonstrated, mixtures of policies are required to match feature counts, and, similarly, many different mixtures of policies satisfy feature matching. No method is proposed to resolve the ambiguity.

기존의 방법들은
ambiguity 처리에
약하다.

Maximum Entropy IRL

We take a different approach to matching feature counts that allows us to deal with this ambiguity in a principled way, and results in a single stochastic policy. We employ the principle of maximum entropy (Jaynes 1957) to resolve ambiguities in choosing distributions. This principle leads us to the distribution over behaviors constrained to match feature expectations, while being no more committed to any particular path than this constraint requires.

Deterministic Path Distributions

Unlike previous work that reasons about policies, we consider a distribution over the entire class of possible behaviors. This corresponds to paths of (potentially) variable length (Figure 1b) for deterministic MDPs (Figure 1a).

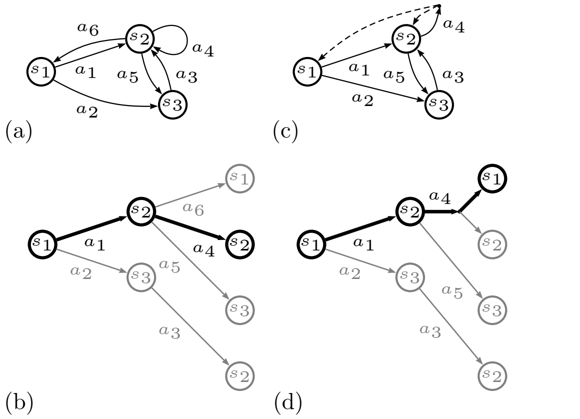

Figure 1: A deterministic MDP (a) and a single path from its path-space (b). A non-deterministic MDP (c) and a single path from its path-space (d).

Similar to distributions of policies, many different distributions of paths match feature counts when any demonstrated behavior is sub-optimal. Any one distribution from among this set may exhibit a preference for some of the paths over others that is not implied by the path features. We employ the principle of maximum entropy, which resolves this ambiguity by choosing the distribution that does not exhibit any additional preferences beyond matching feature expectations (Equation 1). The resulting distribution over paths for deterministic MDPs is parameterized by reward weights θ (Equation 2). Under this model, plans with equivalent rewards have equal probabilities, and plans with higher rewards are exponentially more preferred.

아마 principle of maximum entropy
에서 유도되는 공식이다..?

이건 deterministic version

$$P(\zeta_i|\theta) = \frac{1}{Z(\theta)} e^{\theta^\top \mathbf{f}_{\zeta_i}} = \frac{1}{Z(\theta)} e^{\sum_{s_j \in \zeta_i} \theta^\top \mathbf{f}_{s_j}} \qquad (2)$$

trajectory의 feature visitation

Given parameter weights, the *partition function*, $Z(\theta)$, always converges for finite horizon problems and infinite horizons problems with discounted reward weights. For infinite horizon problems with zero-reward absorbing states, the partition function can fail to converge even when the rewards of all states are negative. However, given demonstrated trajectories that are absorbed in a finite number of steps, the reward weights maximizing entropy must be convergent.

Non-Deterministic Path Distributions

In general MDPs, actions produce non-deterministic transitions between states (Figure 1c) according to the state transition distribution, T. Paths in these MDPs (Figure 1d) are now determined by the action choices of the agent and the random outcomes of the MDP. Our distribution over paths must take this randomness into account.

We use the maximum entropy distribution of paths conditioned on the transition distribution, T, and constrained to match feature expectations (Equation 1). Consider the space of action outcomes, $\mathcal{T}$, and an outcome sample, o, specifying the next state for every action. The MDP is deterministic given o with the previous distribution (Equation 2) over paths compatible with o (i.e., the action outcomes of the path and o match). The indicator function, $I_{\zeta \in o}$ is 1 when ζ is compatible with o and 0 otherwise. Computing this distribution (Equation 3) is generally intractable. However, if we assume that transition randomness has a limited effect on behavior and that the partition function is constant for all $o \in \mathcal{T}$, then we obtain a tractable approximate distribution over paths (Equation 4).

$$P(\zeta|\theta,T) = \sum_{o \in \mathcal{T}} P_T(o) \frac{e^{\theta^\top \mathbf{f}_\zeta}}{Z(\theta,o)} I_{\zeta \in o} \tag{3}$$

일종의 belief state

reward sum

transition model

$$\approx \frac{e^{\theta^\top \mathbf{f}_\zeta}}{Z(\theta,T)} \prod_{s_{t+1},a_t,s_t \in \zeta} \overbrace{P_T(s_{t+1}|a_t,s_t)}^{\text{transition model}} \tag{4}$$

weights to optimize.

Stochastic Policies

This distribution over paths provides a stochastic policy (i.e., a distribution over the available actions of each state) when the partition function of Equation 4 converges. The probability of an action is weighted by the expected exponentiated rewards of all paths that begin with that action.

$$P(\text{action } a|\theta,T) \propto \sum_{\zeta : a \in \zeta_{t=0}} P(\zeta|\theta,T) \tag{5}$$

Learning from Demonstrated Behavior

Maximizing the entropy of the distribution over paths subject to the feature constraints from observed data implies that we maximize the likelihood of the observed data under the maximum entropy (exponential family) distribution derived above (Jaynes 1957).

$P(\zeta|\theta) \simeq \frac{1}{Z} e^{\theta^T f_\zeta} \prod_{s,a \in \zeta} P(s_{t+1}|s_t,a_t)$

$$\theta^* = \operatorname*{argmax}_\theta L(\theta) = \operatorname*{argmax}_\theta \sum_{\text{examples}} \log \underline{P(\tilde{\zeta}|\theta,T)}$$

This function is convex for deterministic MDPs and the optima can be obtained using gradient-based optimization methods. The gradient is the difference between expected empirical feature counts and the learner's expected feature counts, which can be expressed in terms of expected state visitation frequencies, D_{s_i}.

Expert demonstration

expected feature visitation of current reward

$$\nabla L(\theta) = \tilde{\mathbf{f}} - \sum_\zeta P(\zeta|\theta,T)\mathbf{f}_\zeta = \tilde{\mathbf{f}} - \sum_{s_i} \boxed{D_{s_i}\mathbf{f}_{s_i}} \tag{6}$$

At the maxima, the feature expectations match, guaranteeing that the learner performs equivalently to the agent's demonstrated behavior regardless of the actual reward weights the agent is attempting to optimize (Abbeel & Ng 2004).

In practice, we measure empirical, sample-based expectations of the feature values, and not the true values of the agent to be imitated. Assuming the magnitude of the features can be bounded, a standard union and Hoeffding bound argument can provide high-probability bounds on the error in feature expectations as a function of the number of samples– in particular, these bounds have only an $O(\log K)$ dependence on the number of features.[1] Dudík & Schapire (2006) show that the maximum entropy problem that results given bounded uncertainty in feature expectation is a *maximum a posteriori* problem exactly like the one described above, but with an l_1-regularizer added on (with the strength of regularization depending on the uncertainty in that feature expectation). In our experimental section we use the online exponentiated gradient descent algorithm, which is both very efficient and induces an l_1-type regularizing effect on the coefficients. [2]

Efficient State Frequency Calculations

Given the expected state frequencies, the gradient can easily be computed (Equation 6) for optimization. The most straight-forward approach for computing the expected state frequencies is based on enumerating each possible path. Unfortunately, the exponential growth of paths with the MDP's time horizon makes enumeration-based approaches computationally infeasible.

이게 생각보다 매우 중요하다!

Algorithm 1 Expected Edge Frequency Calculation

$\sum_{s_i} D_{s_i} f_{s_i}$를 잘 계산하기!

```
Backward pass
1. Set $Z_{s_{\text{terminal}}} = 1$
2. Recursively compute for $N$ iterations
     $Z_{a_{i,j}} = \sum_k P(s_k|s_i,a_{i,j}) e^{\text{reward}(s_i|\theta)} Z_{s_k}$
        i는 state
        j는 action
        k번째에서 i번째 state로 이동할 확률을 의미한다.
        ⇒ k번째 density × i번째 reward × i에서 k로 갈 확률
     $Z_{s_i} = \sum_{a_{i,j}} Z_{a_{i,j}} + \mathbf{1}_{\{s_i = s_{\text{terminal}}\}}$
        ← 여긴 state에 이동해야요.
Local action probability computation
3. $P(a_{i,j}|s_i) = \frac{Z_{a_{i,j}}}{Z_{s_i}}$
        i번째 state에서 j번째 action의 density / i번째 state의 density ⇒ policy
Forward pass
4. Set $D_{s_i,t} = P(s_i = s_{\text{initial}})$
5. Recursively compute for $t = 1$ to $N$
     $D_{s_k,t+1} = \sum_{s_i} \sum_{a_{i,j}} D_{s_i,t} P(a_{i,j}|s_i) P(s_k|a_{i,j},s_i)$
        recursively update
        transition probability
Summing frequencies
6. $D_{s_i} = \sum_t D_{s_i,t}$
        ← 최종 state density
```

Instead, our algorithm computes the expected state occupancy frequencies efficiently using a technique similar to the

[1] In contrast, margin-based and locally normalizing models rely on techniques that scale linearly in the number of features.

[2] For stochastic MDPs we can achieve better usage of finite data by removing the variance in sample feature expectations due to the uncertainty in the MDP. Space doesn't permit the full exposition of the incomplete (and non-convex) log-likelihood, but the intuitive expectation-maximization algorithm that results fits the maximum-entropy model using initial feature expectations and then improves those estimates by running the resulting policy in the MDP.

forward-backward algorithm for Conditional Random Fields or value iteration in Reinforcement Learning. The algorithm approximates the state frequencies for the infinite time horizon using a large fixed time horizon. It recursively "backs up" from each possible terminal state (Step 1) and computes the probability mass associated with each branch along the way (Step 2) by computing the partition function for Equation 4 at each action and state. These branching values yield local action probabilities (Step 3), from which state frequencies in each timestep can be computed (Steps 4 and 5) and summed for the total state frequency counts (Step 6).

Driver Route Modeling

Our research effort on maximum entropy approaches to IRL was motivated by applications of imitation learning of driver route choices. We are interested in recovering a utility function useful for *predicting driving behavior* as well as for *route recommendation*. To our knowledge, this is the largest-scale IRL problem investigated to date in terms of demonstrated data size.

Route Choice as an MDP

Road networks present a large planning space with known structure. We model this structure for the road network surrounding Pittsburgh, Pennsylvania, as a deterministic MDP with over 300,000 states (i.e., road segments) and 900,000 actions (i.e., transitions at intersections). We assume that drivers who are executing plans within the road network are attempting to reach some goal while efficiently optimizing some trade-off between time, safety, stress, fuel costs, maintenance costs, and other factors. We call this value a *cost* (i.e., a negative reward). We represent the destination within the MDP as an absorbing state where no additional costs are incurred. Different trips have different destinations and slightly different corresponding MDPs. We assume that the reward weight is independent of the goal state and therefore a single reward weight can be learned from many MDPs that differ only in goal state.

Collecting and Processing GPS Data

We collected GPS trace data from 25 Yellow Cab taxi drivers over a 12 week duration at all times of day. This yielded a dataset of over 100,000 miles of travel collected during over 3,000 hours of driving and covering a large area surrounding Pittsburgh. We employed a particle filter to fit the sparse GPS data to the road network and segmented the fitted traces into approximately 13,000 distinct trips using a time-based threshold to determine stopping locations. We discarded roughly 30% of the trips that were too short (fewer than 10 road segments), too cyclic, or too noisy, and split 20% of the remaining trips into a training set and the remaining 80% of the data into a testing set of 7403 examples.

Path Features

Our road network data includes a detailed set of characteristics that describe each road segment. For our experiments, we consider four different dimensions of characteristics: road type, speed, lanes, and transitions. A road segment is categorized in each of these dimensions (i.e., from interstate to local road, high speed to low speed, and one lane to many lanes) and transitions are categorized as straight, left, right, hard left, and hard right. A path is described by how many miles of each road segment categorization it contains and the number of each transition type. Each road segment's contribution to these 22 different counts is represented in the road segment's features.

IRL Models

We apply our Maximum Entropy IRL model (MaxEnt) to the task of learning taxi drivers' collective utility function for the different features describing paths in our road network. We maximize the probability of demonstrated paths within a smaller fixed class of reasonably good paths rather than the class of all possible paths below a fixed length. Our algorithm is efficient (polynomial time) for both classes, but this reduction provides a significant speed up (without introducing optimization non-convexity) and limits consideration of cycles in the road network.

We demonstrate our approach's effectiveness by comparing with two other IRL models. The first is Maximum Margin Planning (MMP) (Ratliff, Bagnell, & Zinkevich 2006), which is a model capable of predicting new paths, but incapable of density estimation (i.e., computing the probability of some demonstrated path). The second model is an action-based distribution model (Action) that has been employed for Bayesian IRL (Ramachandran & Amir 2007) and hybrid IRL (Neu & Szepesvri 2007). The choice of action in any particular state is assumed to be distributed according to the future expected reward of the best policy after taking the action, $Q^*(S,a)$. In our setting, this value is simply the optimal path cost to the goal after taking a particular action.

$$P(\text{action } a|s_i,\theta) \propto e^{Q^*(s_i,a)} \qquad (7)$$

The difference between this action-based model and our model is best illustrated in the following example.

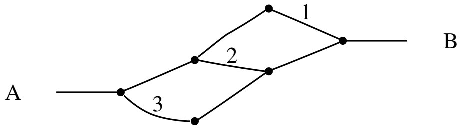

Figure 2: Example of probability distributions over paths.

There are three obvious paths from **A** to **B** in Figure 2. Assuming each path provides the same reward, in the maximum entropy model, each path will have equal probability. In the action-based model, path 3 will have 50% probability while paths 1 and 2 have 25% probability. The distribution will be different for the return trip from **B** to **A**.

More generally, paths in action-based distributions such as this one only compete for probability mass with other paths locally at the action level, and not against other paths that branched earlier. This problem is known as label bias in the Conditional Random Field literature (Lafferty, McCallum, & Pereira 2001). It has undesirable consequences for IRL. For instance, the highest reward policy may not be the most probable policy in the model, and policies with the same expected reward can have different probabilities. Compared to our maximum entropy distribution over paths, this model gives higher probability mass to paths with a smaller branching factor and lower probability mass to those with a higher branching factor.

Comparative Evaluation

We now evaluate each model's ability to model paths in the withheld testing set after being trained on the training set given the path's origin and destination. We use three different metrics. The first compares the model's most likely path estimate with the actual demonstrated path and evaluates the amount of route distance shared. The second shows what percentage of the testing paths match at least 90% (distance) with the model's predicted path. The final metric measures the average log probability of paths in the training set under the given model. For path matching, we evaluate both the most likely path within the action-based model and the lowest cost path using the weights learned from the action-based model. We additionally evaluate a model based on expected travel times that weights the cost of a unit distance of road to be inversely proportional to the speed of the road, and predicts the fastest (i.e., lowest cost) route given these costs.

	Matching	**90% Match**	**Log Prob**
Time-based	72.38%	43.12%	N/A
Max Margin	75.29%	46.56%	N/A
Action	77.30%	50.37%	-7.91
Action (costs)	77.74%	50.75%	N/A
MaxEnt paths	**78.79%**	**52.98%**	**-6.85**

Table 1: Comparison of different models' abilities to match most likely path predictions to withheld paths (average percentage of distance matching and percentage of examples where at least 90% of the paths' distances match) and the probability of withheld paths (average log probability).

The results of this analysis are shown in Table 1. For each of these metrics, our maximum entropy model shows significant ($\alpha < .01$) improvements over the other models.

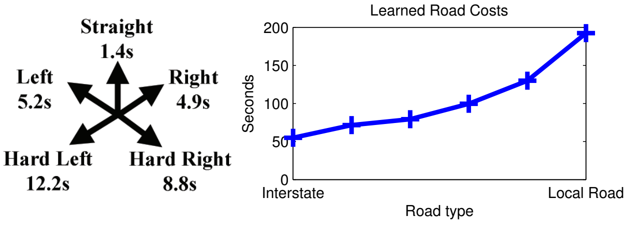

Figure 3: Learned costs of turns (left) and miles of different road types (right) normalized to seconds (with interstate driving fixed to 65 miles per hour).

The learned cost values using our MaxEnt model are shown in Figure 3. Additionally, we learn a fixed per edge cost of 1.4 seconds that helps to penalize paths composed of many short roads.

Applications

Beyond the route recommendation application described above, our approach opens up a range of possibilities for driver prediction. Route recommendation can be easily personalized based on passive observation of a single user. Further, by learning a probability distribution over driver preferences, destinations, and routes the MaxEntIRL model of driver behavior can go beyond route recommendation, to new queries like: "What is the probability the driver will take this street?" This enables a range of new applications, including, e.g., warning drivers about unanticipated traffic problems on their route *without* ever explicitly having to query the user about route or destination; optimizing battery and fuel consumption in a hybrid vehicle; and activating temperature controls at a home prior to the driver's arrival.

So far, we have not described situations where the driver's intended destination is unknown. Fortunately we can reason easily about intended destinations by applying Bayes' theorem to our model of route preference. Consider the case where we want the posterior probability of a set of destinations given a partially traveled path from A to B.

$$\begin{aligned} P(dest|\tilde{\zeta}_{A\to B}) &\propto P(\tilde{\zeta}_{A\to B}|dest)P(dest) \\ &\propto \frac{\sum_{\zeta_{B\to dest}} e^{\theta^\top \mathbf{f}_\zeta}}{\sum_{\zeta_{A\to dest}} e^{\theta^\top \mathbf{f}_\zeta}} P(dest) \end{aligned}$$

These quantities can easily be computed using our inference algorithm (Algorithm 1).

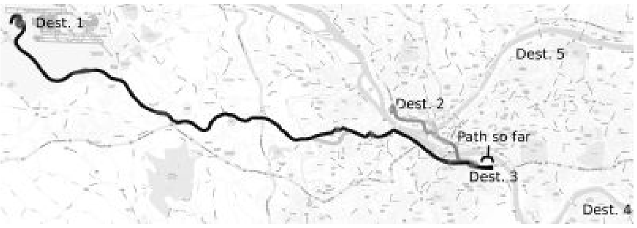

Figure 4: Destination distribution (from 5 destinations) and remaining path distribution given partially traveled path. The partially traveled path is heading westward, which is a very inefficient (i.e., improbable) partial route to any of the eastern destinations (3, 4, 5). The posterior destination probability is split between destinations 1 and 2 primarily based on the prior distribution on destinations.

Figure 4 shows one particular destination prediction problem. We evaluate our model's ability to predict destinations for routes terminating in one of five locations around the city (Figure 4) based on the fraction of total route observed (Figure 5). We use a training set to form a prior over destinations and evaluate our model on a withheld test set. Incorporating additional contextual information into this prior distribution, like time of day, will be beneficial for predicting the destinations of most drivers.

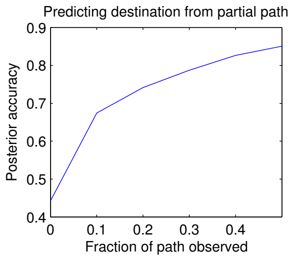

Figure 5: Posterior prediction accuracy over five destinations given partial path.

Related Work

In locally normalizing probabilistic IRL models, probability mass is assigned to each action based on some summary statistic. The value of the optimal policy has been employed (Neu & Szepesvri 2007; Ramachandran & Amir 2007). Beyond this probability mass assignment, paths prefixed with one action do not compete for probability mass with paths prefixed by other actions. The effect, known as label bias in the CRF literature (Lafferty, McCallum, & Pereira 2001), is that paths in portions of the state space with many branches will be biased towards lower probability mass while those with fewer branches will be biased towards higher probability mass. As a consequence, the highest reward behavior in an MDP may not be the most probable, and behaviors that match in expected reward need not match in probability. Our model avoids the label bias problem, giving equivalent probability to behaviors with equivalent expected reward, and larger probability to higher reward behavior. Further, we note that the models suggested lead to potentially difficult non-convex optimization problems with multiple minima.

Route preference modeling has been studied using a few different approaches. Liao *et al.* (2007) model transportation decisions using a directed graphical model. Local action distributions are learned from demonstrated behavior captured in GPS traces. While this model can represent the same distributions as our undirected model, it is much less efficient. Contextual information, like road closures, can influence action probability distribution throughout the entire road network. Consequentially, a different set of action distributions must be learned for every destination and possible context, leading to estimates based on very sparse amounts of data.

Krumm & Horvitz (2006) use route efficiency of partial routes to varying destinations to perform destination prediction of a driver. This same notion of efficiency is captured within our probabilistic model. The TRIP system (Letchner, Krumm, & Horvitz 2006) learns the time inefficiency values drivers are willing to accept for each of their traveled routes and discounts the costs of these previously traveled road segments for each user by the level of accepted inefficiency, implicitly capturing some of their preferences. Our IRL formalization of the problem can be viewed as an extension to this work that not only enables portions of desired routes to have lowered costs, but also increases the costs of undesirable routes. Additionally, approaching the problem in a parametric fashion allows our model to efficiently incorporate contextual information by learning drivers' preferences of those contexts, and to generalize to previously un-encountered road networks.

Conclusions and Future Work

We present a novel approach to inverse reinforcement and imitation learning that cleanly resolves ambiguities in previous approaches, provides a convex, computationally efficient procedure for optimization and maintains important performance guarantees. We applied our method to the problem of modeling route preferences, but we focused primarily on describing and evaluating the differences between our model and other imitation learning models using a small feature space. In future work, we plan to improve our model by incorporating contextual factors (e.g., time of day, weather) into our feature space, and inducing region-based or even specific road based features that can explain, e.g., the avoidance of a particular road only during rush hour, or a steep road during winter weather.

Acknowledgments

The authors thank Eric Oatneal and Jerry Campolongo of Yellow Cab Pittsburgh for their assistance, and the Quality of Life Technology Center/National Science Foundation for supporting this research under Grant No. EEEC-0540865.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*, 1–8.

Dudík, M., and Schapire, R. E. 2006. Maximum entropy distribution estimation with generalized regularization. In *Proc. COLT*, 123–138.

Jaynes, E. T. 1957. Information theory and statistical mechanics. *Physical Review* 106:620–630.

Krumm, J., and Horvitz, E. 2006. Predestination: Inferring destinations from partial trajectories. In *Proc. Ubicomp*, 243–260.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 282–289.

Letchner, J.; Krumm, J.; and Horvitz, E. 2006. Trip router with individualized preferences (trip): Incorporating personalization into route planning. In *Proc. IAAI*, 1795–1800.

Liao, L.; Patterson, D. J.; Fox, D.; and Kautz, H. 2007. Learning and inferring transportation routines. *Artificial Intelligence* 171(5-6):311–331.

Neu, G., and Szepesvri, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. UAI*, 295–302.

Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proc. ICML*, 663–670.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *Proc. IJCAI*, 2586–2591.

Ratliff, N.; Bagnell, J. A.; and Zinkevich, M. 2006. Maximum margin planning. In *Proc. ICML*, 729–736.