Modeling Decision Making with Maximum Entropy Inverse Optimal Control
Thesis Proposal

Brian D. Ziebart
Machine Learning Department
Carnegie Mellon University

September 30, 2008

Thesis committee:
J. Andrew Bagnell
Anind K. Dey
Martial Hebert
Dieter Fox, University of Washington

Abstract

Accurately modeling the decision making of humans (and other agents) is an important machine learning problem needed for realizing human-like robots, improving human-computer interactions, and enabling many other artificial intelligence applications. A powerful view for this modeling problem is that humans behave purposefully (i.e., with objectives) and take actions to efficiently fulfill those objectives. A key challenge is the inherent uncertainty in many domains where the factors influencing the human's decisions are not completely observable or they are distorted by noise. Approaches used for generating optimal decisions are ill-suited for modeling observed behavior due to this uncertainty.

To address this uncertainty, we propose a novel probabilistic approach for modeling decision making within the Markov Decision Process framework based on the principle of maximum entropy. Our research agenda includes the mathematical derivation of a model based on this approach, an investigation of the model's theoretical properties, the development of efficient algorithms for inference and learning within the model, and the application of the model to a number of real world decision modeling tasks. We present some preliminary research and empirical evaluations of the approach in the domain of vehicle route preference modeling.

Contents

1 Introduction

Accurate models of human decision making are an important component for realizing an improved symbiosis between humankind and technology across a number of different domains. These models enable intelligent computer interfaces that can anticipate user actions and intentions, ubiquitous computing environments that automatically adapt to the behaviors of their occupants, and robots with human-like behavior that complements our own actions and goals. For simple domains, behavior may be a deterministic function of a small set of variables, and possible to completely specify by hand. However, for sufficiently complex decision making domains, manual specification is too difficult, and instead the model should be automatically constructed from observed behavior.

Figure 1: The road network covering a portion of Pittsburgh.

Fortunately, most human behavior is purposeful – people take actions to efficiently accomplish objectives – rather than completely random. For example, when traversing the road network (Figure 1), drivers are trying to reach some destination, and choosing routes that have a low personalized *cost* (in terms of time, money, stress, etc.). The modeling problem then naturally decomposes into modeling a person's changing "purposes," and the efficient actions taken to achieve those objectives. When "purposes" have domain-level similarity, we'd expect the notion of efficiency to be similar for differing objectives, allowing behavior fulfilling different goals to be useful for modeling common notions of utility and efficiency. Markov decision processes provide a framework for representing these objectives and notions of efficiency.

Dealing with the uncertainty inherent in observed behavior represents a key challenge in the machine learning problem of constructing these models. There are many sources of this uncertainty. The observed agent may base its decision making on additional information that the learner may not be able to observe or that may differ from the observer's information due to noise. Another possibility is that the agent's behavior may be intrinsically random due to the nature of its decision selection process. Additionally, nature often imposes additional randomness on the outcome of observed actions that must be taken into account.

We propose to address uncertainty from both probabilistic transition dynamics and agent behavior variability by employing the principle of maximum entropy within a feature-based Markov decision process framework. In this view of decision making, states and actions are only important insofar as the underlying characteristics (i.e., features) that describe them. Our research agenda consists of three main lines of questions:

- How can maximum entropy-based probabilistic models of decision making be employed for different problem settings (e.g., modeling decisions, decisions in time, groups of decision makers)?
- How efficiently can reasoning (i.e., inference and learning) be performed for these models?
- How can these models and algorithms for reasoning be applied to real problems? And how does our approach empirically compare to previous approaches?

The remainder of this proposal is organized to systematically investigate these questions. We first review the two main building blocks of our work, Markov decision processes and the principle of maximum entropy, and survey existing approaches for modeling decision making. We then present our maximum entropy-based approaches and efficient algorithms for inference and learning within those approaches. We next provide empirical validation of our model for one application domain, and propose additional applications to validate our approach's generality.

2 Background and Survey

Decision making and statistical uncertainty each have a long history of research. We first review the Markov decision process framework for representing decision making problems, and the principle of maximum entropy for resolving ambiguity in probability distributions. We then provide a brief survey of existing techniques for modeling sequences of decisions.

2.1 Markov Decision Processes

A Markov Decision Process (MDP) is a tuple $\mathcal{M} = (S, A, P(s'|s,a), R(s,a))$ of states (S), actions (A), action-dependent state transition probabilities ($P(s'|s,a)$), and reward values ($R(s,a)$). The MDP is "solved" by finding a *policy* ($\pi(s)$) specifying the action for each state that yields the highest expected cumulative reward, $E_{P(s'|s,\pi(s))}[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)]$ over finite or infinite time horizon (T) and optionally with discounted future reward ($\gamma < 1$) (Bellman 1957). Various extensions to MDPs exist: partially observable Markov decision processes (POMDP) for planning with uncertainty in the agent's state (Drake 1962) and semi-Markov decision processes (SMDP) when the reward is a function of the duration that the agent spends in a state (Howard 1963).

2.2 Principle of Maximum Entropy

When given only partial information about a probability distribution, P, many distributions will match that information. The principle of maximum entropy resolves the ambiguity of under-constrained distribution by selecting the distribution that has the least *commitment* to any particular outcome while matching the observational constraints imposed on the distribution (Jaynes 1957). This is equivalent to maximizing the distribution's entropy (Equation 1) while matching any constraints imposed on the distribution.

$$H(P) = -\sum_x P(x) \log P(x) \tag{1}$$

The generalization, maximum relative entropy, is defined with respect to some baseline distribution, Q, and is equivalent to minimizing the Kullback Leibler-divergence (Kullback & Leibler 1951) between P and Q (Equation 2), while matching partial information constraints.

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \tag{2}$$

The uniform distribution is typically employed for Q(x) (yielding Equation 1) because it satisfies exchangability equivalence, which is desirable in the absence of any information. Dudík & Schapire (2006) incorporate regularization with the principle of maximum entropy in heavily constrained distributions to avoid overfitting.

2.3 Inverse Optimal Control and Imitation Learning

Inverse optimal control (Boyd *et al.* 1994, Ng & Russell 2000), originally posed by Kalman, describes the problem of recovering an agent's reward function, R(s,a), given demonstrated sequence(s) of actions, $\{\tilde{\zeta_1} = \{a_1|s_1, a_2|s_2, ...\}, \tilde{\zeta_2}, ...\}$, when the remainder of the MDP, $\mathcal{M}/R$, is known. Vectors of reward factors $\mathbf{f}_{s,a}$ describe each available action, and the reward function is assumed to be a linear function of those factors, $R(s,a) = \theta^\top \mathbf{f}_{s,a}$ parameterized by reward weights, θ. Ng & Russell (2000) formulate inverse optimal control as the recovery of reward weights, θ, that make demonstrated behavior optimal.

Unfortunately this formulation is ill-posed. Demonstrated behavior is optimal for many different reward weights, including degeneracies (e.g., all zeros).

Abbeel & Ng (2004) propose recovering reward weights so that a planner based on those reward weights and the demonstrated trajectories have equal reward (in expectation). This formulation reduces to matching the planner and demonstrated trajectories' expected *feature counts*, $\mathbf{f}_\zeta = \sum_{s,a \in \zeta} \mathbf{f}_{s,a}$.

$$\sum_{\zeta} P_{\text{plan}}(\zeta|\theta)\mathbf{f}_\zeta = \mathbf{f}_{\tilde{\zeta}} \qquad (3)$$

Abbeel & Ng (2004) employ a series of deterministic controls obtained from "solving" the optimal MDP for the distribution over trajectories. When sub-optimal behavior is demonstrated (due to the agent's imperfection or unobserved reward factors), mixtures of policies are required to match feature counts. Many different mixtures will match feature counts and no method is proposed to resolve this ambiguity.

Ratliff, Bagnell, & Zinkevich (2006) resolve this ambiguity by posing inverse optimal control as a maximum margin problem with loss-augmentation. This algorithm operates by raising the costs of actions along the demonstrated trajectory and finding cost weights making the cost-augmented demonstrated trajectory optimal. While the approach yields a unique solution, it suffer from 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.

An imitation learning approach to the problem, which still aims to obtain similar behavior, but without any performance guarantees, relaxes the MDP optimality assumption by employing the MDP "solution" policy's reward, $Q_\theta(a,s) = \max_{\zeta \in \Xi_{s,a}} \theta^\top \mathbf{f}_\zeta$, within a Boltzmann probability distribution.

$$P(\text{action } a|s) = \frac{e^{Q_\theta(a,s)}}{\sum_{\text{action } a'} e^{Q_\theta(a',s)}} \qquad (4)$$

Neu & Szepesvri (2007) employ this distribution within a loss function penalizing the squared difference in probability between the model's action distribution and the demonstrated action distribution. Ramachandran & Amir (2007) utilize it within a Bayesian approach to obtain a posterior distribution over reward values using Markov Chain Monte Carlo simulation. The main weaknesses of the model are that maximum likelihood (and MAP) estimation of parameters is a non-convex optimization, and the learned model lacks performance guarantees with respect to the Markov decision process.

Our proposed approach is both probabilistic and convex. Unlike the mixture of optimal behaviors (Abbeel & Ng 2004), training behavior will always have non-zero probability in our model, and parameter choices are well-defined. Unlike maximum margin planning (Ratliff, Bagnell, & Zinkevich 2006), our method realistically assumes that demonstrated behavior may be sub-optimal (at least for the features observed by the learner). Finally, unlike the Boltzmann Q-value stochastic model (Neu & Szepesvri 2007, Ramachandran & Amir 2007), learning in our model is convex, cannot get "stuck" in local maxima, and provides performance guarantees.

2.4 Probabilistic Graphical Models

A great deal of research within the machine learning community has focused on developing probabilistic graphical models to address uncertainty in data. These models provide a framework for representing independence relationships between variables, learning probabilistic models of data, and inferring the values of latent variables. Two main variants are directed models (i.e., Bayesian networks) and undirected models (i.e., Markov random fields and conditional random fields).

Bayesian networks model the joint distribution of a set of variables by factoring the distribution into a product of conditional probabilities of each variable given its "parent" variables (Pearl 1985). A number of Bayesian network models for planning have been proposed (Attias 2003, Verma & Rao 2006). Unfortunately in many real world decision making problems, decisions are based not only on the current action's features, but the features of all subsequent actions as well. This leads to a very non-compact model that generalizes poorly when predicting withheld data. We investigate these deficiencies empirically in our preliminary experiments.

Markov random fields model the energy between combinations of variables using potential functions. In their generalization, conditional random fields (CRFs) (Lafferty, McCallum, & Pereira 2001), the potential functions can depend on an additional set of variables that are themselves not modeled. In a number of recognition tasks, these additional variables are observations, and the CRF is employed to recognize underlying structured properties from these observations. This approach has been applied to recognition problems for text (Lafferty, McCallum, & Pereira 2001), vision (Kumar & Hebert 2006), and activities (Liao, Fox, & Kautz 2007, Vail, Veloso, & Lafferty 2007). The maximum entropy inverse optimal control model we derive for Markov decision problems with deterministic action outcomes can be interpreted as a chain conditional random field where the entire sequence of decisions is conditioned on the preference variables, and all state and action features. This is significantly different than how conditional random fields have been applied for recognition tasks, where labels for each variable in the sequence are conditioned on local observations from portion of the sequence.

3 Maximum Entropy Inverse Optimal Control

We now investigate addressing decision making uncertainty by employing the principle of maximum entropy to obtain the least *committed* probability distribution over sequences of decisions (i.e., trajectories) possible while providing performance guarantees. We derive probability distributions over decisions based on a standard application of the principle of maximum entropy and propose alternative interpretations of the principle of maximum entropy to obtain additional probability distributions. We propose further extensions to our approach for modeling decision making in time, and modeling the decision making of multiple agents that have similarities.

3.1 Deterministic Dynamics (completed)

We first consider a maximum entropy distribution over trajectories in a Markov decision process with deterministic action outcomes (Ziebart *et al.* 2008a). Following Abbeel & Ng (2004), the distribution is constrained to match feature expectations with the demonstrated trajectories, $\tilde{\zeta}$.

$$\begin{aligned} \max H(P(\zeta)) \\ \sum_{\zeta} P(\zeta)\mathbf{f}_\zeta = \mathbf{f}_{\tilde{\zeta}} \end{aligned} \tag{5}$$

Solving using the method of Lagrange multipliers yields a probability distribution over trajectories and actions.

$$P(\zeta|s,\theta) = \frac{e^{\theta^\top \mathbf{f}_\zeta}}{\sum_{\zeta' \in \Xi_s} e^{\theta^\top \mathbf{f}_{\zeta'}}} = \frac{e^{\theta^\top \mathbf{f}_\zeta}}{Z_s} \tag{6}$$

$$P(a|s,\theta) = \frac{\sum_{\zeta \in \Xi_{s,a}} e^{\theta^\top \mathbf{f}_\zeta}}{\sum_{\zeta \in \Xi_s} e^{\theta^\top \mathbf{f}_\zeta}} = \frac{Z_{s,a}}{Z_s} \tag{7}$$

Where Ξ_s and $\Xi_{s,a}$ are the class of paths starting with state s and with state s and action a. Depending on the setting, these classes may additionally be constrained to terminate at some specific state. We can express these probabilities in terms of the state and state-action partition functions, Z_s and $Z_{s,a}$ (with dependence on θ suppressed).

$$P(a|s,\theta) = \text{softmax}(\log Z_{s,a}) = \text{softmax}(\theta^\top \mathbf{f}_{s,a} + \log Z_{\Phi(s,a)}) \tag{8}$$

The action probability distribution can then be interpreted as a *softmax* function over all available actions (Equation 8)[1].

Choosing reward weights, θ, to maximize the constrained objective function (Equation 5) is equivalent to maximizing the probability of demonstrated trajectories (Equation 6). This function is convex in θ. We present efficient algorithms for optimization in Section 4.2. Numerous variants to Equation 5 exist. For example, if we assume that our features are undesirable, we can constrain feature matching appropriately, $\sum_\zeta P(\zeta)\mathbf{f}_\zeta \leq \mathbf{f}_{\tilde{\zeta}}$. Our reward parameters are then constrained to be negative $\theta \leq 0$, strictly corresponding to *costs*.

[1] Φ provides the deterministic next state given a state and action

3.2 Non-deterministic Dynamics (completed)

In general MDPs, transitions between states are stochastic according to transition probabilities, $P(s'|s,a)$. The maximum entropy action distribution (Equation 5) must take these stochastic transitions into account.

$$P(a|s,\theta) = \text{softmax}\left(\theta^\top \mathbf{f}_{s,a} + \sum_{s'} P(s'|s,a) \log Z_{s'}\right) \tag{9}$$

This stochastic extension to the deterministic *softmax* interpretation (Equation 8) is derived in Appendix A and the result is shown in Equation 9.

In this setting, obtaining the best reward weights, θ, is complicated by the ignorance about what the agent would have done in states that were not visited, but influenced the agent's decision making. To address this uncertainty, we assume expected feature counts with bounded error and the optimization then remains convex with an additional L_1 regularization term (Dudík & Schapire 2006). As the number of demonstrated trajectories increases, the difference between feature count sample means and expected feature counts (based on all possibly dynamic outcomes) due to transition dynamic sampling converges to zero.

3.3 Re-examining "Uniformity" (proposed)

Up until now, our statistical models of decision making have been constructed using a uniform "baseline" distribution over decision sequences[2] (i.e., paths) For arbitrary Markovian baseline distributions, $P_0(\zeta)$, the distribution over paths for deterministic MDPs is:

$$P(\text{path } \zeta|\theta) \propto P_0(\zeta)\, e^{\theta^\top \mathbf{f}_\zeta} = e^{\sum_{s,a\in\zeta}\left(\theta^\top \mathbf{f}_{s,a} - \log P_0(a|s)\right)} \tag{10}$$

One candidate baseline distribution is to choose each *action* uniformly at random from the available actions at each state. This suggests adding a *log action count* feature to each state and learning (rather than assuming) an appropriate feature weight.

Uniform distributions over paths and over actions are both insensitive to the amount of similarity one path shares with other paths. Intuitively, ignoring this dependence is problematic because a single latent feature on a road segment shared by multiple paths can impact the desirability of all those paths. One approach is to use a baseline distribution that better accounts for similarity between paths. We propose employing the maximum entropy distribution over flows (i.e., the expected number of times an action is taken).

$$\max H(\mathbf{D}) \tag{11}$$

$$\sum_{s'} D_{s',a} P(s|s',a) = \sum_{a'} D_{s,a'} \tag{12}$$

[2] This distribution is "uniform" when ignoring stochastic dynamics. See Appendix A.

While this optimization is convex, formulating efficient algorithms for solving it, and evaluating its impact remain as future work.

3.4 Semi-Markovian Modeling (proposed)

In many domains, it's important to model not just *which* actions are taken, but *when* those actions are taken. Thus, the *duration* of time spent in different states is an important consideration for temporally modeling human behavior. When explicitly indexing decisions in time, and not just modeling the time-independent sequences of decisions, Markov models that do not track "duration" as part of the state space have a geometric duration distribution that often does not match reality. This is especially true for domains being modeled at a high frequency, but having relatively sparse "change." Semi-Markovian approaches to MDPs (Sutton, Precup, & Singh 1999) and CRFs (Sarawagi & Cohen 2004) exist. We propose the extension of these state-centric ideas to our feature-centric maximum entropy model.

In this setting, each decision is supplemented with a duration, $\triangle t$. Observed decision making is then a sequence of state-duration-action triples (s, $\triangle t$, a) or, by extracting away states and actions, a sequence of feature vector-duration pairs $(\mathbf{f}_{s,a},\ \triangle t)$[3]. Matching the expected temporal feature counts, $E_\zeta[\triangle t\ \mathbf{f}_{s,a}] = \tilde{E}[\triangle t\ \mathbf{f}_{s,a}]$, can be accomplished by assigning features appropriately to an expanded Markov Decision Process decision space and employing our previously described method. However, simply matching temporal feature counts does not necessarily produce a model with durations similar to demonstrated behavior.

We propose additional constraints for matching higher-order duration statistics. Adding first moment temporal constraints, $E_\zeta[\triangle t\ \mathbf{f}_{s,a}] = \tilde{E}[\triangle t\ \mathbf{f}_{s,a}]$, and second moment temporal constraints, $E_\zeta[(\triangle t)^2\ \mathbf{f}_{s,a}] = \tilde{E}[(\triangle t)^2\ \mathbf{f}_{s,a}]$, to our maximum entropy formulation (Equation 5) corresponds to a feature-weighted, exponentiated squared loss function (i.e., discretized pseudo-Gaussian) that penalizes large deviations from a learned mean. A different penalty is learned for each feature, and each action's duration is a weighted combination of these penalties. This model can be realized by adding a current state-action memory counter to the MDP and making the rewards of transitions out of the state dependent on that memory. We also plan to investigate applying approximations, such as stochastic dynamic programming (Puterman 1994), to expedite inference.

3.5 Similarity-based User Modeling (proposed)

In a number of application domains we are interested in modeling the decision making of groups of agents. Rather than treating each agent as completely independent from all other agents, how can we leverage the similarities between

[3] In addition to action-based features, state-based features and a combination of both are straight-forward extensions.

agents to better predict decision making? For instance, we may know the age and gender of observed decision makers and, if those with similar demographic attributes make similar decisions, we would expect this additional information to be useful.

One approach we propose is to directly supplement the planning space's features with an additional set of features that are also function of our agent's characteristics. Training data from all different agents can be combined into one large pool of data, and our existing approaches for learning will appropriately weight the rewards for these agent characteristic-based features. Choosing appropriate agent characteristic functions that capture similarity remains as future work.

A secondary, more powerful approach we propose is to model the relationship between agents, agent characteristics, reward weights, and demonstrated decisions as a hierarchical model. This would enable, for instance, groups of agents with similar characteristics to be part of two distinct latent "types" of agents. With a small amount of observed decision making data, its "type" could be inferred and the data of similar agents leveraged in prediction. A fully Bayesian treatment of this hierarchical model is likely to require a simulation-based approach, such as Markov Chain Monte Carlo, to obtain a posterior distribution over reward weights.

4 Inference and Learning Algorithms

Armed with the probabilistic models for decision making developed in Section 3, we now investigate whether these models can be efficiently employed for large-scale problems. The number of paths in an MDP grows exponentially with path length, making naïve algorithms that enumerate each potential path intractable for large-scale problems. We propose efficient polynomial time algorithms for inference in decision making domains with thousands of states and for efficient learning from arbitrarily large amounts of training examples.

4.1 Fast Action Distribution Inference (completed)

At first, inference may seem intractable for our path-based maximum entropy inverse optimal control model. Fortunately the numerator and denominator of Equation 7 can each be defined recursively. Returning to partition function notation,

$$Z_s = \sum_{\zeta \in \Xi_s} e^{\theta^\top \mathbf{f}_\zeta} = \sum_{\text{action } a|s} \sum_{\zeta \in \Xi_{s,a}} e^{\theta^\top \mathbf{f}_\zeta} = \sum_a Z_{s,a} \tag{13}$$

$$Z_{s,a} = \sum_{\zeta \in \Xi_{s,a}} e^{\theta^\top \mathbf{f}_\zeta} = e^{\theta^\top \mathbf{f}_{s,a}} \sum_{\zeta \in \Xi_{\Phi(s,a)}} e^{\theta^\top \mathbf{f}_\zeta} = e^{\theta^\top \mathbf{f}_{s,a}} Z_{\Phi(s,a)}$$

The dynamic program employed to compute these values recursively can be viewed as a probabilistic version of value iteration in Markov Decision Processes or a variant of the [forward-]backward algorithm (Baum & Petrie 1966) for Conditional Random Fields. Additional extensions exist for applying similar dynamic programs to the stochastic setting (Appendix A).

4.2 Reward Weight Learning (completed)

Finding the constrained maximum entropy distribution over paths (Equation 5) is equivalent to finding the maximum likelihood estimate of the reward parameters, θ, of Equation 6. This function is convex in θ, so gradient-based methods can be employed to find the global optima.

We use stochastic [exponentiated] gradient ascent. Given a demonstrated trajectory, $\tilde{\zeta}_i$, we have update rule:

$$\begin{aligned}\theta_{t+1} &\leftarrow \theta_t - \eta \nabla P(\tilde{\zeta}_i | \theta_t) \\ &= \theta_t - \eta \left(E_\zeta[\mathbf{f}_\zeta | \theta_t] - \mathbf{f}_{\tilde{\zeta}_i} \right) \end{aligned} \tag{14}$$

Expected feature counts in Equation 14 are computed efficiently using the dynamic partition function recurrence (Equation 13) to first compute action probabilities, and then a second dynamic algorithm to efficiently compute visitation counts.

4.3 Fast MAP Plans for Deterministic Dynamics (completed)

In many application settings, inferring the most likely trajectory (or policy) for many different agents with varying reward weights is useful. For deterministic MDPs, the lowest cost (i.e., highest reward) path is the most probable.

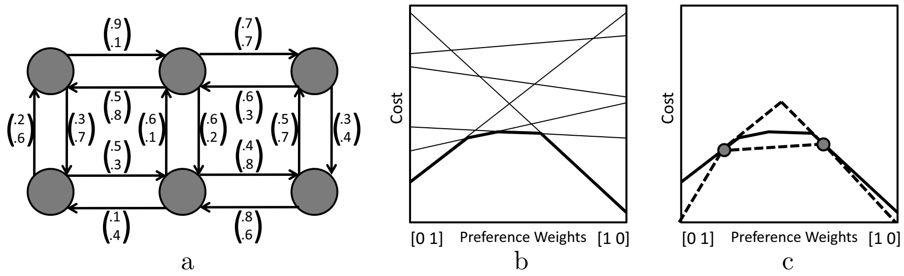

Figure 2: A planning space with two cost factors (a), trajectory costs in preference space (b), and bounded optimal trajectory costs (c).

We employ bounds based on the concavity of optimal path cost over the space of reward weights to more efficiently guide graph-based search (i.e., the A* search algorithm) for some new agent preferences (i.e., negative reward weight) using previously computed optimal policy costs for other preference weights

(Ziebart, Dey, & Bagnell 2008). We additionally combine these preference-based bounds with spatial bounds for memory efficiency.

4.4 Inference with Latent Goals (completed)

In some settings, the agent is trying to reach a particular goal state that is latent to observers. Since the MaxEnt IRL model is probabilistic and conditioned on the goal state for goal-based applications, Bayes rule can be applied quite naturally to obtain a posterior distribution over possible goal states given observations of the agent's actions.

$$\begin{aligned}
P(\text{goal } g|\zeta, \text{origin } o) &\propto P(\zeta|\text{goal } g, \text{origin } o)P(\text{goal } g) \\
&= \frac{\sum_{\zeta' \in \Xi_{\zeta}^{\rightarrow g}} e^{\theta^\top \mathbf{f}_{\zeta'}}}{\sum_{\zeta' \in \Xi_{o}^{\rightarrow g}} e^{\theta^\top \mathbf{f}_{\zeta'}}} P(\text{goal } g) \\
&\propto \frac{Z_{\zeta \rightarrow g}}{Z_{o \rightarrow g}} P(\text{goal } g) \qquad (15)
\end{aligned}$$

However, when there are many potential goal states, computing the likelihood of each independently may not be efficient enough for real-time applications. Fortunately, by applying our dynamic program (Equation 13) starting from the origin state, we can efficiently compute the partition function from the origin to each state, $Z_{o \rightarrow s}$, and from the final state of ζ to each state, $Z_{\zeta \rightarrow s}$. This allows us to efficiently compute the posterior (Equation 15).

4.5 Activity Inference and Learning (proposed)

In some application domains, observations of highly-structured behavior is available, but only at the low level of decisions. For example, we may have the location of a person working within the house, but no labels of the particular tasks that the person is performing. Models that ignore high level tasks will predict poorly because, to a large extent, it is the high level task that determines all low-level behavior.

We propose constructing models incorporating the higher-level decision making (i.e., sequences of sub-goals and activities) using only low-level observations (i.e., actions) by employing an Expectation-Maximization (Dempster, Laird, & Rubin 1977) approach to learn the model. The resulting model can then be employed to predict low-level actions. We expect that by incorporating higher-levels of behavior, even if our model has no semantic meaning attached to that behavior, it will outperform models that strictly model lower-level actions on action prediction tasks.

5 Applications for Modeling Decision Making

We now focus our attention on applications to validate our approach's value and extent of applicability.

Application	**Validation**
Route Preferences (5.1)	**Deterministic Dynamic Model (3.1)**
	Alternate Baseline Distributions (3.3)
	Similarity-based User Modeling (3.5)
	Fast Action Distribution Inference (4.1)
	Reward Weight Learning (4.2)
	Fast MAP Planning (4.3)
	Latent Goal Inference (4.4)
Pedestrian Motion (5.2)	**Deterministic Dynamic Model (3.1)**
	Fast Action Distribution Inference (4.1)
	Reward Weight Learning (4.2)
	Latent Goal Inference (4.4)
	Activity Inference and Learning (4.5)
Family Routing Modeling (5.3)	Deterministic Dynamic Model (3.1)
	Semi-Markov Modeling (3.4)
	Fast Action Distribution Inference (4.1)
	Reward Weight Learning (4.2)
	Activity Inference and Learning (4.5)
Control Improvement (5.4)	Deterministic Dynamic Model (3.1)
	Fast Action Distribution Inference (4.1)
	Reward Weight Learning (4.2)
	Activity Inference and Learning (4.5)

Table 1: Applications and the components of this thesis that they validate. Completed portions are in bold.

An overview of the proposed applications and components of the thesis they validate are shown in Table 1.

5.1 Driver Route Preference Modeling

We first present completed research on modeling the route preferences of drivers in the road network. This application demonstrates our approach's non-temporal model for sequences of decisions. We are able to scale our algorithms to a large planning space, learn from a large pool of training data, and empirically demonstrate an empirical advantage of our approach over a significant amount of previous research in this domain.

There are many criteria for evaluating driving routes. How long will driving the route take? Is it fuel efficient? What are the toll costs? Is it stressful?

How safe is the route? How each driver trades off these various factors in selecting their driving route is a matter of personal preference that is often difficult to manually specify, as the factors themselves are complicated functions of both static characteristics of the road network (e.g., topology, speed limits, toll information), and contextual data (e.g., time of day, day of week, weather, construction, accidents). Knowing these personal preferences for drivers enables personalized route recommendation, warnings for unanticipated hazards during unguided driving, and automated vehicle configuration based on turn predictions.

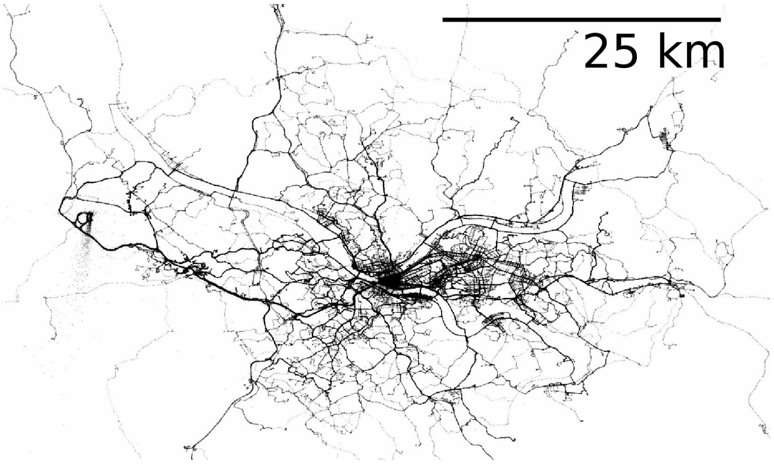

Figure 3: The collected GPS datapoints

We collected over 100,000 miles of GPS trace data (Figure 3) from different groups of drivers (e.g., taxi drivers, retirees). We fit the GPS traces to the road network using a particle filter and applied our model to learn driver preferences as a function of road network features (e.g., segment distance, speed limits, road class, turn type) (Ziebart *et al.* 2008a). Our evaluations for this preliminary research showed significant improvements in most likely path estimation and path density estimation of our model of other Boltzmann Q-value models (Ramachandran & Amir 2007, Neu & Szepesvri 2007) and Maximum Margin Planning (Ratliff, Bagnell, & Zinkevich 2006).

In extensions to this work (Ziebart *et al.* 2008b), we added contextual information (time of day, accidents, construction, congestion) to the model and compared it to other approaches previously to route prediction, turn prediction, and destination prediction. We compared against directed graphical models, which have been employed for transportation routine modeling (Liao *et al.* 2007). Since we are only concerned with single modal transportation, we compare against Markov models of decision at next intersection conditioned on the goal location (Simmons *et al.* 2006) and conditioned on the previous k road segments (Krumm 2008).

	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 2: Evaluation results for optimal estimated travel time route, max margin route, Boltzmann Q-value distributions (Action) and Maximum Entropy

Model	**Dist. Match**	**90% Match**
Markov (1x1)	62.4%	30.1%
Markov (3x3)	62.5%	30.1%
Markov (5x5)	62.5%	29.9%
Markov (10x10)	62.4%	29.6%
Markov (30x30)	62.2%	29.4%
Travel Time	72.5%	44.0%
PROCAB	**82.6%**	**61.0%**

Table 3: Evaluation results for Markov Model with various grid sizes, time-based model, and PROCAB model

We also evaluated our model on the problem of predicting destination given partial trajectory by simply employing Bayes rule and incorporating a prior distribution over destinations. We compare against the Predestination system (Krumm & Horvitz 2006) and a destination-based Markov model (Simmons *et al.* 2006). Predestination discretizes the world into grid cells and probabilistically predicts drivers' destinations based on a statistical model of driver efficiency. It assumes a fixed metric (travel time) and models efficiency (i.e., preference) given that metric, whereas our model, PROCAB, assumes a fixed preference model and learn the driver's metric.

We find both our model and Predestination significantly outperform the Markov model, and our model performs somewhat better given large amount (90%) and small amount (10%–30%) of trip completed.

5.2 Pedestrian Motion Prediction

In our second application domain, we propose modeling the movement of people (i.e., pedestrians) walking in different enviornments. Unlike vehicle route preference modeling, where the road network presents a set of discrete choices, walking movements in an environment are less constrained and are in fact continuous. Dealing with this lack of structure in the decision space is an important component of this application domain. We plan to use this application to validate our approach's generality to grid-based domains, and to model higher level

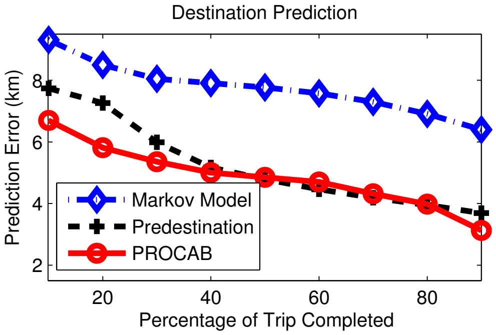

Figure 4: The best Markov Model, Predestination, and PROCAB prediction errors

behavior from only low level data (Section 4.5).

Appropriately planning safe robot or vehicle movements and actions requires accurate predictive models of the future motions of surrounding agents (human or otherwise). Existing models for motion are largely based on current position and velocity. While this works well for short-term predictions, it ignores the longer scale dynamics that govern motion. Namely, that most motion is executed to accomplishing some long-term objective, such as reaching a destination, and the actions that comprise motion are chosen with sensitivity to the surrounding environment.

We have collected and continue to collect people tracking data from within the Intel Research Pittsburgh Laboratory and from pedestrians in urban settings. We will apply our maximum entropy inverse optimal control model of pedestrian motion in both of these environments. Features in these domains correspond to the terrain type (e.g., sidewalk, grass, street) and different types of obstacles and distances to obstacles. For the Intel dataset, we also have longer-term tracking of people within an environment with different sub-goals (e.g., sink, refrigerator, microwave) and higher-level activities (e.g., prepare lunch). We plan to model these higher-level activities as described in Section 4.5.

5.3 Family Routine Modeling

In our next application, we propose modeling the locations and activities of members of families, as collected automatically from GPS traces and by interview. Unlike the previous two applications, in this setting we are modeling multiple agents who have strongly interacting behaviors, and additionally, because of the domain, modeling duration is important. This application will specifically evaluate our proposed approach for modeling temporal duration.

5.4 Control Improvement for the Impaired

In our final application, we propose applying our technique to model the intentions of impaired users of different systems, for the purpose of automatically improving control for those users. We have two domains in mind. The first is joystick-operated control of wheelchairs, and the second is mouse movement for computer interaction. The idea for both is use our approach to model and infer the user's intended goal, and then adjust the control to more easily achieve inferred goals. This application will further demonstrate the potential impact of our approach, and validate its usage for real-time applications.

6 Timeline

We plan the following tentative timeline for the completion of our proposed research.

Sept 2008: Thesis proposal

Oct–Dec 2008: Extend model of decision making:

- Investigate random action-based MaxEnt IRL
- Investigate edge flow-based MaxEnt IRL

Jan–Mar 2009: Develop general purpose software for:

- Specifying the decision making space
- Providing observed decision making data
- Training the model
- Performing inferences in the trained model

Apr–May 2009: Learn high-level behavior from pedestrian data

June–Aug 2009: Extend software for temporal modeling

Sept–Oct 2009: Model family location/activity

Nov–Feb 2010: Control improvement

Jan–Mar 2010: Job search

Mar-May 2010: Write thesis

May 2010: Thesis defense

7 Conclusion

In this proposal, we have presented a research agenda for probabilistically modeling human decision making. Our proposed approach is based on the applying the principle of maximum entropy with the Markov decision process framework. We believe the resulting model to be well-suited for modeling human behavior where relevant information about the decision space may differ between the learner and human, and/or the human's decisions may be influenced by randomness. We have applied this approach to the problem of vehicle route preference modeling with very promising results, and we propose a number of additional applications and necessary theoretic and algorithmic extensions to further evaluate the central thesis of this research.

References

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

Attias, H. 2003. Planning by probabilistic inference. In *Proc. of the 9th Int. Workshop on Artificial Intelligence and Statistics*.

Baum, L., and Petrie, T. 1966. Statistical inference for probabilistic function of finite state Markov chains. *Annals of Mathematical Statistics* 37:1554–1563.

Bellman, R. 1957. A Markovian decision process. *Journal of Mathematics and Mechanics* 6:679–684.

Boyd, S.; El Ghaoui, L.; Feron, E.; and Balakrishnan, V. 1994. Linear matrix inequalities in system and control theory. *SIAM* 15.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1):1–38.

Drake, A. 1962. *Observation of a Markov process through a noisy channel.* Ph.D. Dissertation, Massachusetts Institute of Technology.

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

Howard, R. 1963. Semi-markovian decision processes. In *Proc. 34th Session International Statistical Institute*, 625–652.

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.

Krumm, J. 2008. A markov model for driver route prediction. *Society of Automative Engineers (SAE) World Congress*.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics* 22:49–86.

Kumar, S., and Hebert, M. 2006. Discriminative random fields. *Int. J. Comput. Vision* 68(2):179–201.

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

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

Liao, L.; Fox, D.; and Kautz, H. 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. Rob. Res.* 26(1):119–134.

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.

Pearl, J. 1985. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, 329–334.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

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.

Sarawagi, S., and Cohen, W. W. 2004. Semi-markov conditional random fields for information extraction. In *Proc. NIPS*, 1185–1192.

Simmons, R.; Browning, B.; Zhang, Y.; and Sadekar, V. 2006. Learning to predict driver route and destination intent. *Proc. Intelligent Transportation Systems Conference* 127–132.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1-2):181–211.

Vail, D. L.; Veloso, M. M.; and Lafferty, J. D. 2007. Conditional random fields for activity recognition. In *Proc. AAMAS*, 1–8.

Verma, D., and Rao, R. 2006. Goal-based imitation as probabilistic inference over graphical models. In *Proc. NIPS*, 1393–1400.

Ziebart, B. D.; Maas, A.; Bagnell, J. A.; and Dey, A. K. 2008a. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*.

Ziebart, B. D.; Maas, A.; Dey, A. K.; and Bagnell, J. A. 2008b. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proc. Ubicomp*.

Ziebart, B. D.; Dey, A. K.; and Bagnell, J. A. 2008. Fast planning for dynamic preferences. In *Proc. ICAPS*.

A Stochastic Maximum Entropy Inverse Reinforcement Learning

We set out to employ the principle of maximum entropy to model trajectories in a Markov Decision Process, $\mathcal{M} = (S, A, P(s'|s,a), R(s,a))$ where the next state of an agent given its action is random with known distribution, $P(s'|a,s)$, and the reward function, $R(s,a)$ is unknown.

We maximize the trajectory distribution's entropy conditioned on the MDP's dynamics. This maximization is equivalent to minimizing the Kullback-Leibler divergence between our constrained distribution over trajectories and the dynamics-based distribution over trajectories. The dynamics-based distribution is uniform over action sequences (i.e., paths in the planning space), but the probability of realizing any path may be quite non-uniform due to the MDP's dynamics.

$$\begin{aligned} &\operatorname*{argmin}_{\{P(\zeta)\}} \sum_{\zeta} P(\zeta) \log \frac{P(\zeta)}{\prod_{s_+,a,s\in\zeta} P(s_+|a,s)} \qquad (16) \\ &= \operatorname*{argmax}_{\{P(\zeta\}} -\sum_{\zeta} P(\zeta) \log \prod_{a,s\in\zeta} P(a|s) \end{aligned}$$

$$\begin{aligned} &\sum_{\zeta} P(\zeta)\mathbf{f}_\zeta = \mathbf{f}_{\text{obs}} \\ &\forall_{s\in\mathcal{M}} \sum_a P(a|s) = 1 \\ &\forall_{(a,s)\in\mathcal{M}} P(a|s) \geq 0 \end{aligned}$$

Theorem A. *The maximum entropy distribution over actions (a) for some state (s) satisfying the constrained maximum relative entropy distribution (Equation 1) takes the following form:*

$$P(a|s) = softmax(\theta^\top \boldsymbol{f}_{s,a} + \sum_{s'} P(s'|s,a) \log Z_{s'}) = \frac{Z_{a|s}}{Z_s} \tag{17}$$

$$\begin{aligned} Z_{a|s} &= e^{\theta^\top \boldsymbol{f}_{a|s} + \sum_{s'} P(s'|a,s) \log Z_{s'}} \\ Z_s &= \sum_a Z_{a|s} \end{aligned}$$

Proof of Theorem A. We express the Lagrangian (F) of the optimization (Equation 1) in terms of the probability of a particular first action ($\hat{a}$) of state ($\hat{s}$) to obtain the distribution's general form.

$$\begin{aligned} F = -P(\hat{a}|\hat{s}) \log P(\hat{a}|\hat{s}) - P(\hat{a}|\hat{s}) \sum_{\zeta_{\hat{a}}} P(\zeta_{\hat{a}}) \log \prod_{a,s \in \zeta_{\hat{a}}} P(a|s) \\ +\theta^\top P(\hat{a}|\hat{s}) \left(\sum_{\zeta_{\hat{a}}} P(\zeta_{\hat{a}}) \mathbf{f}_{\zeta_{\hat{a}}} + \mathbf{f}_{\hat{a}|\hat{s}} \right) + C_{\hat{s}} \left(\sum_{a'} P(a'|\hat{s}) - 1 \right) + const \end{aligned}$$

Where $\zeta_{\hat{a}}$ are all possible paths *after* taking action $\hat{a}$ in state $\hat{s}$, and *const* are addition terms independent of $P(\hat{a}|\hat{s})$. Taking the gradient with respect to $P(\hat{a}|\hat{s})$ yields a new constraint:

$$-\log P(\hat{a}|\hat{s}) - \sum_{\zeta_{\hat{a}}} P(\zeta_{\hat{a}}) \log \prod_{a,s \in \zeta_{\hat{a}}} P(a|s) + \theta^\top \left(\sum_{\zeta_{\hat{a}}} P(\zeta_{\hat{a}}) \mathbf{f}_{\zeta_{\hat{a}}} + \mathbf{f}_{\hat{a}|\hat{s}} \right) = C_{\hat{s}} \tag{18}$$

The distribution over actions (Equation 17) ensures $P(a|s)$ is a probability distribution satisfying both probabilistic constraints (Equation 16). We now verify that it satisfies Equation 18. After substituting the action distribution into Equation 18, many terms cancel leaving the following equality to verify.

$$\begin{aligned} &\log Z_{\hat{s}} + \sum_{s_+} P(s_+|\hat{a},\hat{s}) \log Z_{s_+} \qquad (19) \\ &+ \sum_{\zeta_{\hat{a}}} \left[P(\zeta_{\hat{a}}) \sum_{a,s \in \zeta_{\hat{a}}} \left(\sum_{s_+} \left(P(s_+|a,s) \log Z_{s_+} \right) - \log Z_s \right) \right] = C_{\hat{s}} \end{aligned}$$

We can let $C_{\hat{s}} = \log Z_{\hat{s}}$ to normalize distribution, and re-express this equality in terms of expected action visitation frequencies, $D_{s,a}$.

$$\sum_{s_+} P(s_+|\hat{a},\hat{s}) \log Z_{s_+} + \sum_{s,a} \left(D_{s,a} \sum_{s'} \left(P(s'|a,s) \log Z_{s'} \right) - \log Z_s \right) = 0 \quad (20)$$

Using the fact that $\sum_{s,a} D_{s,a} \sum_{s'} P(s'|a,s) = \sum_{a"} D_{s',a"}$, we have equality. θ must then be chosen so that the remaining constraint, $\sum_{\zeta} P(\zeta)\mathbf{f}_{\zeta} = \mathbf{f}_{\text{emp}}$, is also satisfied. □