how could you make recommendations to a driver about unanticipated traffic problems relevant to him/her w/out knowing his/her destination?

predict destination given current path
predict path given destination

how is this related to RL?
because path/policy follow depends on reward/preferences of user

given π & traj can we infer reward model?
→ inverse RL

strong ties to learning from demonstration

see slides 1-12 from Abbeel for introduction

core idea: find R^* that matches expert behavior

$$E\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \mid \pi^*\right] \geq E\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \mid \pi\right] \quad \forall \pi$$

e.g. makes observed policy π^* look best

challenges
- what simple reward model always satisfies the above?
  $R^*(s,a)=0 \ \forall s,a$
- Do we observe π^*? No, typically observe trajectories

today: one very influential result (coming from CMU)
Max Entropy IRL (Ziebart, Maas, Bagnell & Dey)

key idea: ambiguity in reward model, select one with maximum entropy that is consistent w/data observed (we will quantify this more precisely)

background/setting
assume reward is a linear combo of weighted features
(similar to linear value function approx)
state represented by set of features, f_{sj}
trajectory is ζ
feature counts for a trajectory $f_\zeta = \sum_{s_j \in \zeta} f_{sj}$ (states visited in traj ζ)
reward for a trajectory

$$\text{reward}(f_\zeta) = \theta^T f_\zeta = \sum_{s_j \in \zeta} \theta^T f_{sj}$$

input is a set of m trajectories, empirical expected feature count $\tilde{f} = \frac{1}{m}\sum f_\zeta$

feature matching idea (Abbeel & Ng 2004)
find reward such that expected feature counts match the empirical expected feature counts
necessary & sufficient to yield same performance as if agent was solving a MDP w/a reward function linear in features
challenge: still lots of ambiguity

Ziebart et al.: use max entropy
- for deterministic MDPs (given action, single next state)

$$P(\zeta_i|\theta) = \frac{1}{Z(\theta)} e^{\theta^T f_{\zeta_i}} = \frac{1}{Z(\theta)} e^{\sum_{s_j \in \zeta_i} \theta^T f_{s_j}}$$

(ζ_i: traj i; θ: reward param)

→ traj w/equal reward have equal prob
→ traj w/larger rewards have exponentially higher prob

- non-deterministic MDPs
T is transition model
action outcomes γ
outcome sample o: specifies unique s' for each s, a
(could view as random # sample)
MDP deterministic given o
$I_{\zeta \in o}$ = indicator function, = 1 if ζ compatible w/ o

$$P(\zeta|\theta, T) = \sum_{o \in \gamma} P_T(o) \frac{e^{\theta^T f_\zeta}}{Z(\theta, o)} I_{\zeta \in o}$$

generally intractable
approx: assume $Z(\theta, o)$ = constant $\forall o$

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

assuming $Z(\theta, T)$ converges (guaranteed for finite horizon & co horizon discounted settings)
then distrib over paths induces a stochastic π

$$P(a|\theta, s) \propto \sum_t \sum_{\zeta \text{ where } s_t = s, a_t = a} P(\zeta|\theta, t)$$

- learning θ

$$\theta^* = \arg\max_\theta L(\theta) = \arg\max_\theta \sum_{\text{examples } \bar{\zeta}} \log P(\bar{\zeta}|\theta, T)$$

convex for det MDPs

gradient is diff between expected empirical feature counts & current θs expected feature counts

$$\nabla L(\theta) = \bar{f} - \sum_\zeta P(\zeta|\theta, T) f_\zeta$$

$$= \bar{f} - \sum_{s_i} \underbrace{D_{s_i}}_{\text{expected state visitation freq}} f_{s_i}$$

in practice don't have true $\bar{f}$, have sample estimates
see paper for details on how to address

to compute gradient need D_{s_i}
Algorithm 1 in paper
(write on board)
similar to value iteration

driver route modeling

learning preferences for features - can generalize to new states

what if people aren't optimal or aren't using a linear reward model?
what if people also have computational constraints?
(bounded rationality, Russell, Griffiths, etc.)