LEARNING DRIVER PREFERENCES FOR FREEWAY MERGING USING MULTITASK IRL

Sanath Bhat
THINC Lab
Department of Computer Science, UGA

OUTLINE

- Introduction
  - Problem Description and Motivation
  - Background
  - Contributions
  - Related Work
- Data and Setup
  - NGSIM
  - Metadata Description
  - MDP Definition
  - Trajectory Extraction

OUTLINE (Contd…)

- Task Separation
  - Split-Merge Clustering
  - Results
- Likelihood Weighting Multitask IRL
  - Hierarchical Bayesian Multitask Model
  - Choice of priors
  - Likelihood Weighting Multitask IRL Algorithm
  - Results
- Conclusion and Future Work

INTRODUCTION

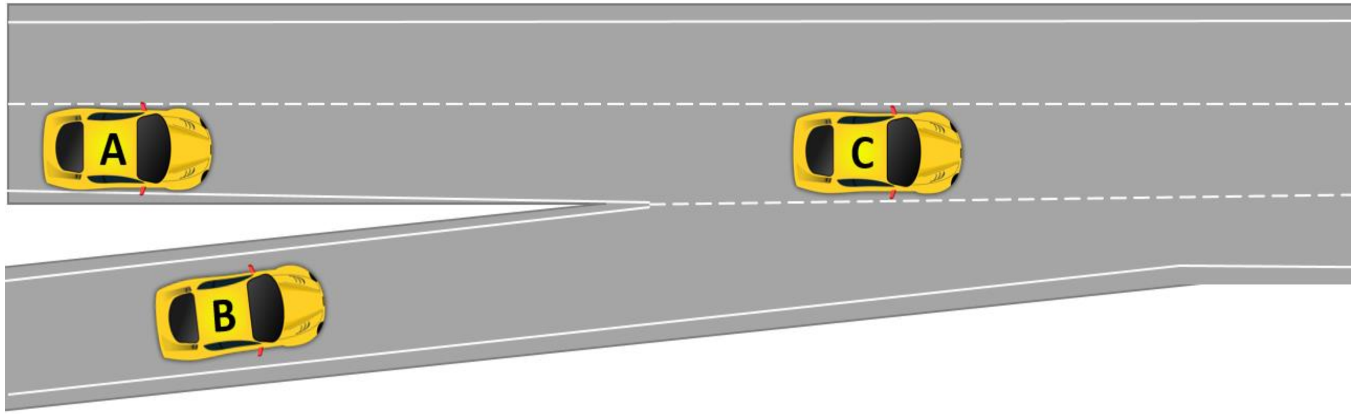
AUTONOMOUS CARS

THINC lab

PROBLEM DESCRIPTION AND MOTIVATION

- Sub-problems of autonomous car-decision making
  - Pathing
  - Collision detection
  - Following relevant signage
  - Parking
  - Passing
  - Lane-Merging

PROBLEM DESCRIPTION AND MOTIVATION



- Car A accelerates
- Car A brakes
- Car A ??

BACKGROUND : MARKOV DECISION PROCESS AND RELATED CONCEPTS

- MDP - 5-tuple $\{S, A, T, R, \gamma\}$:
  - State set, S : $\{S_0, S_1, \ldots S_{|S|}\}$
  - Action set, A : $\{A_0, A_1, \ldots A_{|A|}\}$
  - Transition Function, $T(s' \mid s, a)$: $S \times A \times S \rightarrow [0, 1]$, $\sum_{s'} T(s' \mid s, a) = 1$ for each (s, a)
  - Reward Function, $R(s, a)$: $S \times A \rightarrow \mathbb{R}$
  - Discount factor, $\gamma \in [0,1]$
- Other related:
  - Policy, $\pi(s) : S \rightarrow A$ OR $\underline{\pi(s, a) : S \times A \rightarrow [0, 1]}$, $\sum_a \pi(a \mid s) = 1$ for each s
  - Value Function, $V(s) : S \times A \rightarrow \mathbb{R}$
  - Q-function, $Q(s, a)$: $S \times A \rightarrow \mathbb{R}$

BACKGROUND : MARKOV DECISION PROCESS & RELATED CONCEPTS (Contd…)

- Value function

$$V(s) = \max_{a \epsilon A}\left(R(s,a) + \gamma \sum_{s'} T\,(s,a,s') * V(s')\right)$$

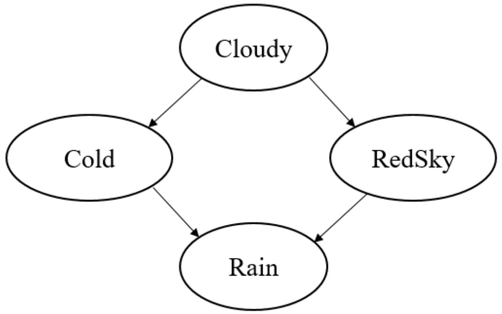
- Value Iteration, Policy Iteration

- Q-function using converged Value function

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} T\,(s,a,s') * V(s')$$

BACKGROUND : LIKELIHOOD WEIGHTING

P(Cloudy)	0.33



Cloudy	P(Cold)
true	0.4
false	0.1

Cloudy	P(RedSky)
true	0.1
false	0.05

Cold	RedSky	P(Rain)
true	true	0.6
true	false	0.15
false	true	0.2
false	false	0.0

P(RedSky | Cloudy = true, Rain = false) = ?

1. Sample weight w = 1.
2. Cloudy = true is evidence => w = w x P(Cloudy = true) = 0.33
3. Cold ~ P(Cold | Cloudy = true). Suppose we get true.
4. RedSky ~ P(RedSky | Cloudy = true). Suppose we get false.
5. Rain=true is evidence => w = w x P(Rain = false | Cold =true, RedSky = false) = 0.33 x 0.85 = 0.2805
6. Tabulate as W[RedSky = false] += 0.2805

function LIKELIHOOD-WEIGHTING(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
    N, the total number of samples to be generated
  local variables: **W**, a vector of weighted counts for each value of X, initially zero

  for j = 1 to N **do**
    x, $w \leftarrow$ WEIGHTED-SAMPLE(bn, **e**)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in **x**
  return NORMALIZE(**W**)

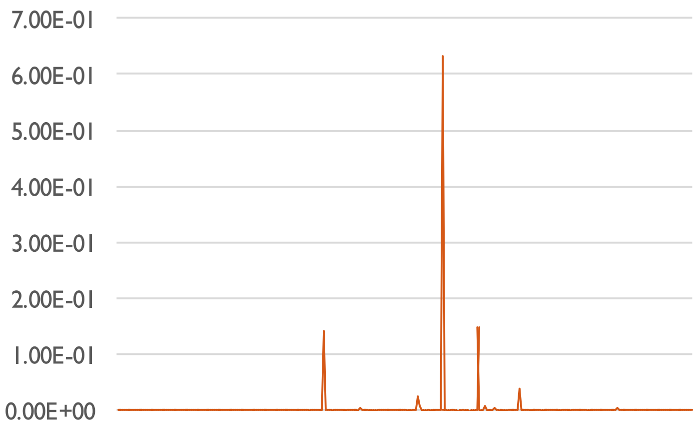
---

function WEIGHTED-SAMPLE(bn, **e**) **returns** an event and a weight

  $w \leftarrow 1$; $\mathbf{x} \leftarrow$ an event with n elements initialized from **e**
  foreach variable X_i **in** $X_1, \ldots, X_n$ **do**
    if X_i is an evidence variable with value x_i in **e**
      then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return x, w

BACKGROUND : LIKELIHOOD WEIGHTING (Contd…)

- Problems with Likelihood Weighting
  - Fares poorly with large number of evidence variables
  - Worsens if evidence occurs late in the network



BACKGROUND : HIERARCHICAL BAYESIAN MODELING

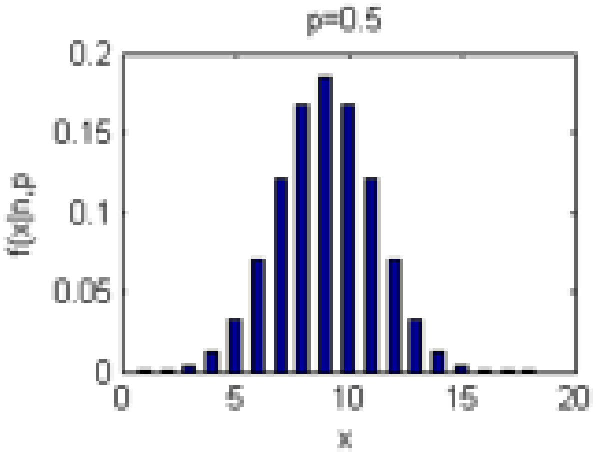
Bayesian Modeling the distribution of the number of heads when n coins are tossed

Problem: Determine p given various x obtained by using some coin

Model

- $X \sim Bin(n, p) = Bin(20, 0.5)$

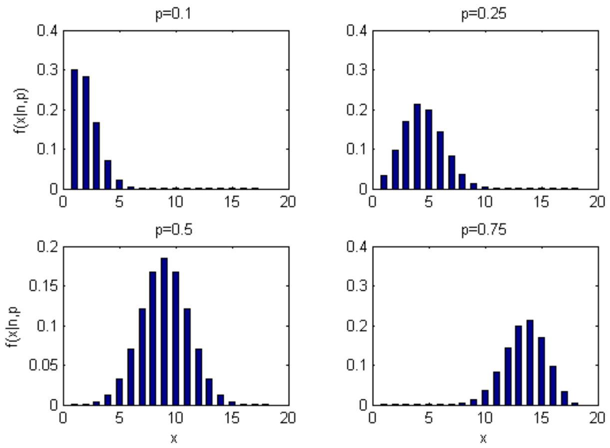
  i.e. $p=0.5$ (a fixed value)



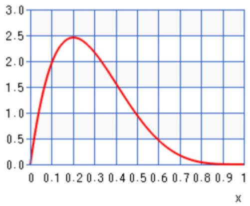
BACKGROUND : HIERARCHICAL BAYESIAN MODELING

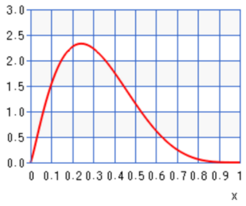
Hierarchical Bayesian Modeling the distribution of the number of heads when n coins are tossed

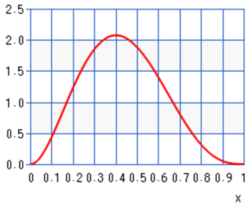
- X ~ Bin(n, p) = Bin(20, p)



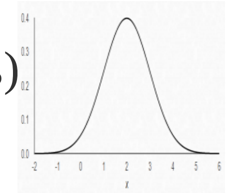
p ~ Beta(α, β) (a parameter)

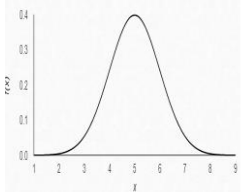






α ~ $\mathcal{N}(2, 1)$, β ~ $\mathcal{N}(5, 2)$ (hyperparameters)





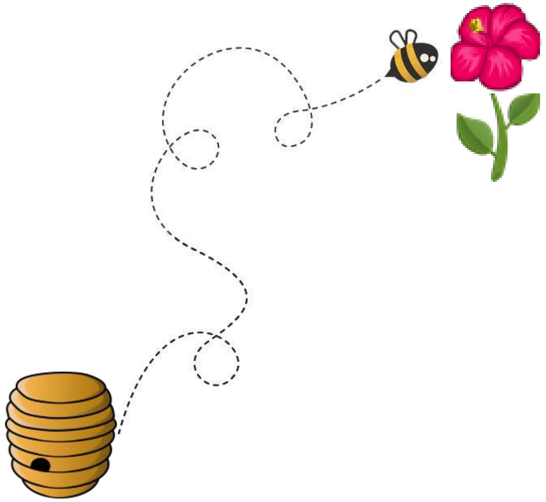
BACKGROUND : HIERARCHICAL BAYESIAN MODELING

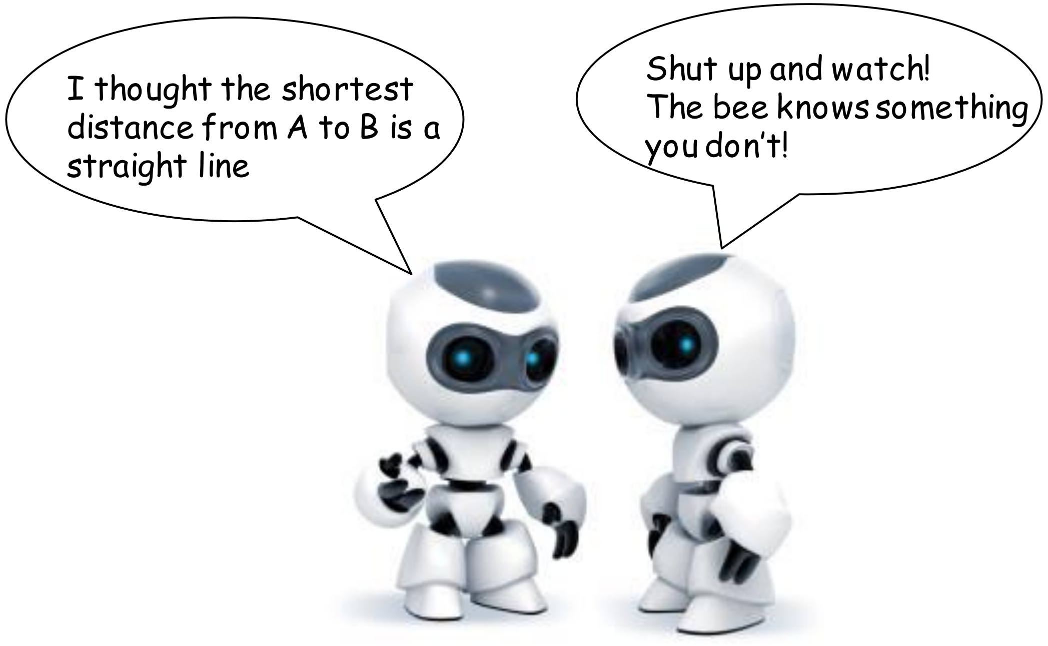
- Hierarchical Model
  - Level 1-Evidence: $X \mid p \sim \text{Bin}(100, p)$
  - Level 2-Prior: $p \mid \gamma \sim \text{Beta}(\alpha, \beta)$ …$\gamma = (\alpha, \beta)$
  - Level 3-Hyperprior: $\gamma \sim \mathcal{N}(2, 1) \times \mathcal{N}(5, 2)$
  - (Can have more levels! E.g.. Prior over Gaussian parameters of hyperpriors)

- Joint posterior given by

$$P(\gamma, p \mid X) \propto P(X \mid p, \gamma)\, P(p, \gamma)\, P(\gamma)$$

BACKGROUND : INVERSE REINFORCEMENT LEARNING





BACKGROUND : INVERSE REINFORCEMENT LEARNING

- Noteworthy mentions
  - Maximum Entropy IRL, Ziebart et al.
  - Bayesian IRL – Ramachandran et al.
  - Non-parametric Bayesian IRL for multiple reward functions, Choi et al.
  - Bayesian Multitask IRL, Dimitrakakis et al.

CONTRIBUTIONS

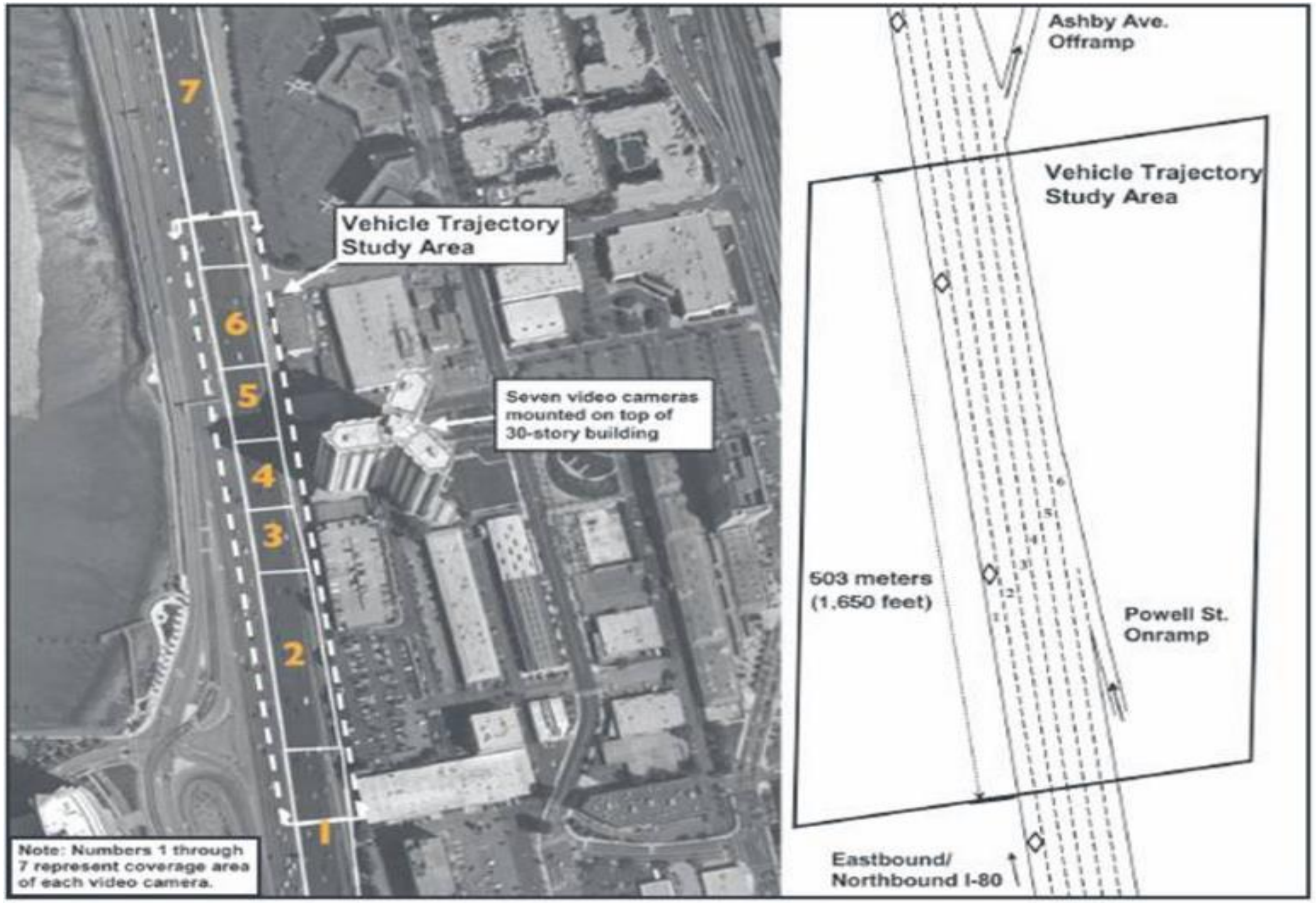
- Establish that a driver has different motivations when passing through different sections of a **merging zone**
- Hypothesize the existence of an '**average behavior**' motivated by **multiple reward functions** for all drivers passing through the merging zone
- To split the merging task into multiple tasks each with its own reward functions and policies differing from that of others.
- To determine the **individual reward functions and policies** of the average driver in each task.

DATA AND SETUP

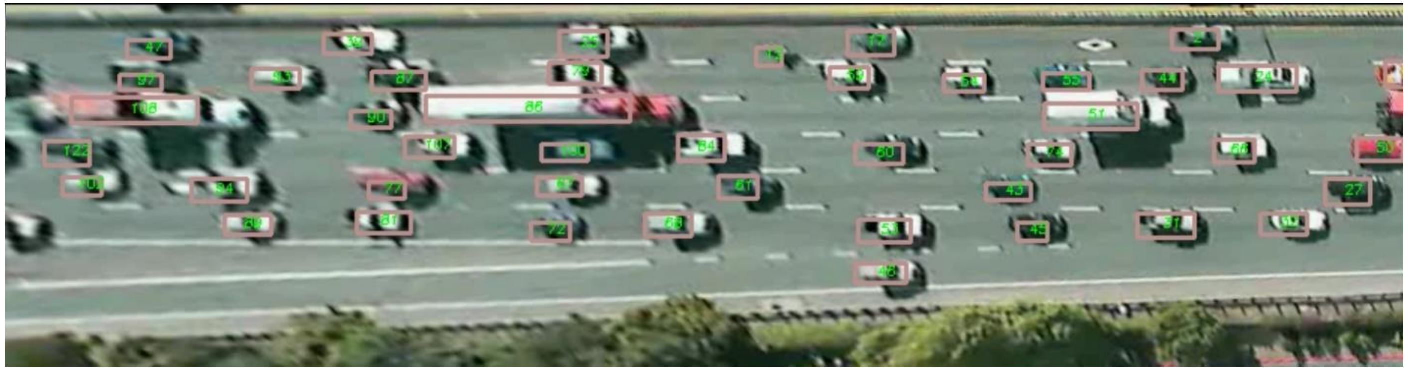
THE NGSIM PROGRAM

- Developed a core of open behavioral algorithms to support traffic simulation and microscopic modeling of vehicles and their interactions
- I80 Dataset – one of the datasets collected under this program. Dataset contains vehicle trajectory data.
  - Where: Eastbound I-80 in the SFO Bay Area in Emeryville, CA
  - When: April 13, 2005
  - Span: Over a length of 1650 feet including an acceleration ramp
  - How: Using 7 synchronized cameras mounted on a 30 story building
  - Duration: Three 15 minute intervals
    - 4.00 p.m. - 4.15 p.m.
    - 5.00 p.m. - 5.15 p.m.
    - 5.15 p.m. - 5.30 p.m.
  - Type of Data: Video + Transcribed csv format

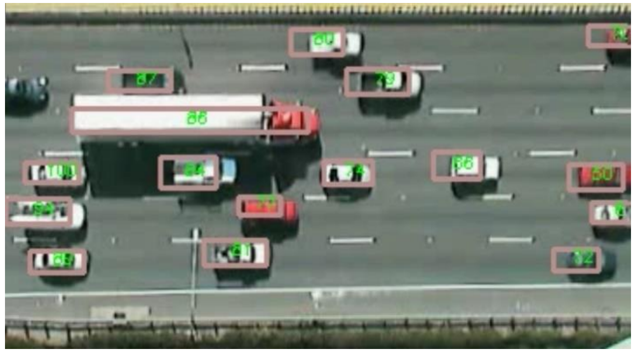
THE NGSIM PROGRAM



THE NGSIM PROGRAM



0 ~250 ~714



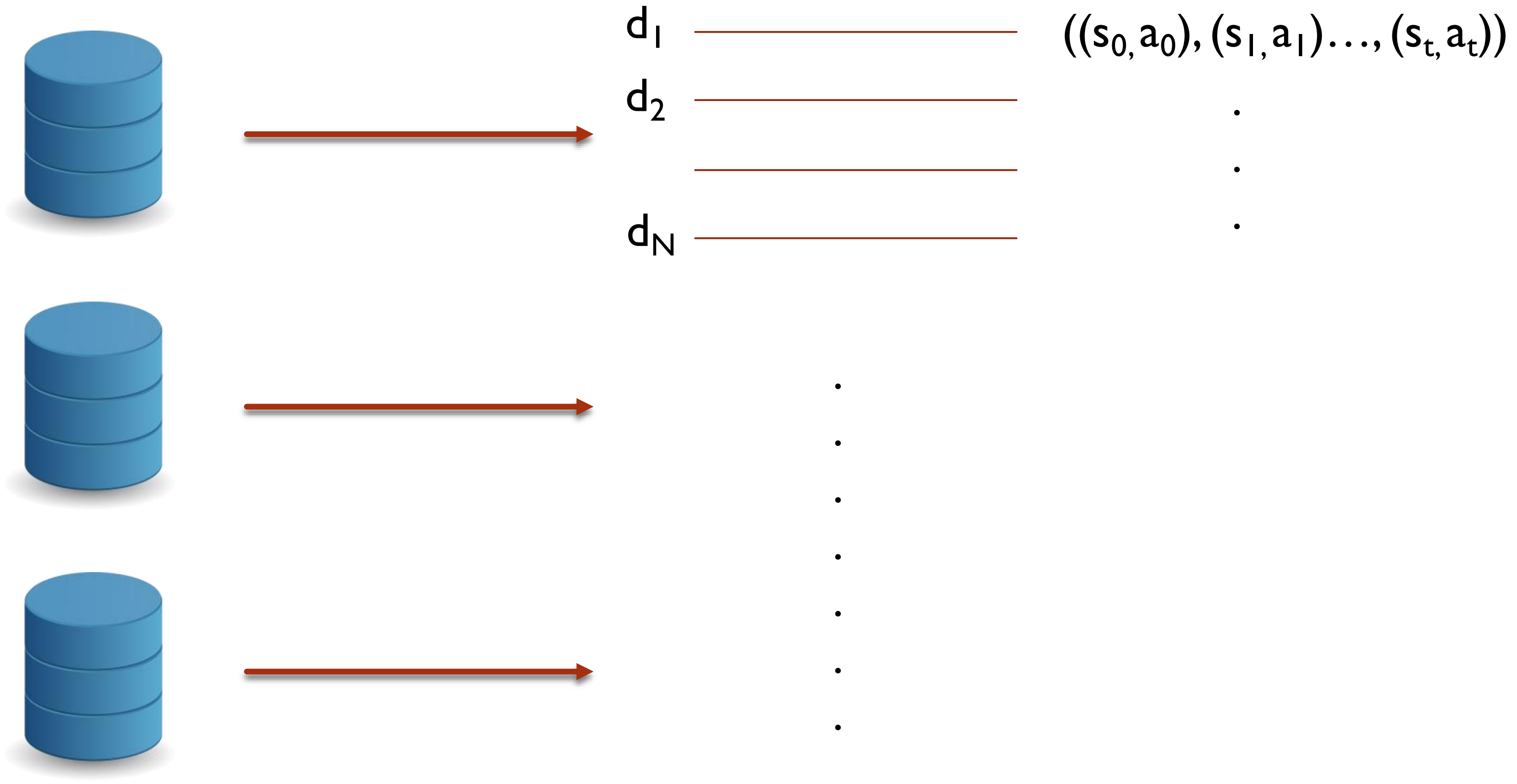
~715 ~900 ~1090

METADATA DESCRIPTION : NGSIM DATASET

- A collection of 3 CSV files each representing the 3 time intervals
- Each row has instantaneous information about a vehicle in one frame represented in 18 columns
- The most important columns that are of our concern are as follows

Column Field	Description	Column Field	Description
Vehicle ID	unique ID of current vehicle	Vehicle acceleration	instantaneous acceleration(feet/s^2)
Frame ID	frame numbers increment by one every 100ms	Lane identification	current lane # of the vehicle
Local Y	the longitudinal coordinate of the front center of the vehicle	Leading Vehicle ID	ID of the leading vehicle
Vehicle length	measured in feet	Following Vehicle ID	ID of the following vehicle
Vehicle velocity	instantaneous velocity (feet/sec)	Spacing	the distance to the front bumper of the preceding vehicle

TRAJECTORY DATA EXTRACTION : OVERVIEW



TRAJECTORY DATA EXTRACTION : MDP MODEL

- MDP model of car A
- State Features:
- Spacing (x_{AC}): Distance between the front bumper of car A and the rear bumper of the leading car i.e. car C.

$$x_{AC} = \text{Spacing in dataset} - \text{length of leading car}$$

- Relative velocity (v_{AC}): Augment the original dataset with another column - instantaneous velocity of the leading car. Then,

$$v_{AC} = \text{velocity(of car A)} - \text{velocity of leading car(augmented column)}$$

TRAJECTORY DATA EXTRACTION : MDP MODEL

- MDP model of car A
- State features:
  - Spacing (x_{AC}): Distance between the front bumper of car A and the rear bumper of the leading car i.e. car C.

    x_{AC} = Spacing in dataset – length of leading car
  - Relative velocity (v_{AC}): Augment the original dataset with another column - instantaneous velocity of the leading car. Then,

    v_{AC} = velocity(of car A) – velocity of leading car(augmented column)
- Action features:
  - Acceleration(a)

x_{AC} (feet)	v_{AC} (feet/second)	a (feet/s^2)
< 0	> 45	< -9
0 – 14	35 – 45	-9 – -4.8
14 – 28	25 – 35	-4.8 – -0.6
28 – 42	15 – 25	-0.6 – 0.6
42 – 56	5 – 15	0.6 – 4.8
56 – 70	-5 – 5	4.8 – 9
70 – 84	-15 – -5	> 9
84 – 98	-25 – -15	
98 – 112	-35 – -25	
112 – 126	-45 – -35	
126 – 140	< -45	
140 – 168		
168 – 196		

TRAJECTORY DATA EXTRACTION : MDP MODEL

x_{AC} \ v_{AC}	> 45	35 – 45	25 – 35	15 – 25	5 – 15	-5 – 5	-15 – -5	-25 – -15	-35 – -25	-45 – -35	< -45
< 0	0	1	2	3	4	5	6	7	8	9	10
0 – 14	11	12	13	14	15	16	17	18	19	20	21
14 – 28	22	23	24	25	26	27	28	29	30	31	32
28 – 42	33	34	35	36	37	38	39	40	41	42	43
42 – 56	44	45	46	47	48	49	50	51	52	53	54
56 – 70	55	56	57	58	59	60	61	62	63	64	65
70 – 84	66	67	68	69	70	71	72	73	74	75	76
84 – 98	77	78	79	80	81	82	83	84	85	86	87
98 – 112	88	89	90	91	92	93	94	95	96	97	98
112 – 126	99	100	101	102	103	104	105	106	107	108	109
126 – 140	110	111	112	113	114	115	116	117	118	119	120
140 – 168	121	122	123	124	125	126	127	128	129	130	131
168 – 196	132	133	134	135	136	137	138	139	140	141	142
> 196	143	144	145	146	147	148	149	150	151	152	153

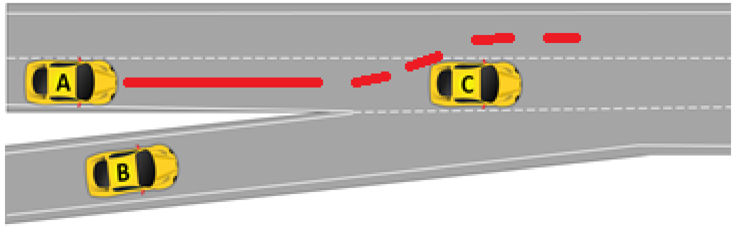
State Space

a (feet/s^2)	Actions
< -9	0
-9 – -4.8	11
-4.8 – -0.6	22
-0.6 – 0.6	33
0.6 – 4.8	44
4.8 – 9	55
> 9	66

Action Space

TRAJECTORY DATA EXTRACTION : TRAJECTORY EXTRACTION

- Extract lane #6 vehicles – About 450-550 vehicles in each time period
- Map features to position, state, action triplets
- Note: Trajectories may be 'broken' or incomplete



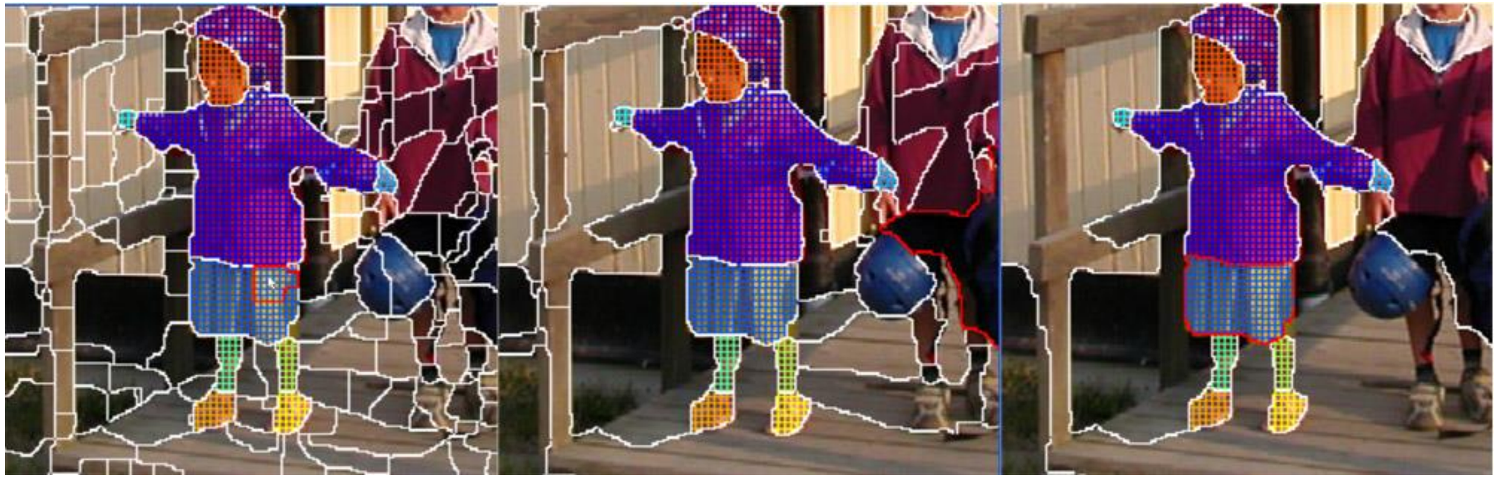
TRAJECTORY DATA EXTRACTION : TRANSITION FUNCTION

- Based on sampling next state distribution for each action for a given state
- Frequency of (s, a , s') triplet given (s, a) is proportional to T(s'| s, a)
- Motion model based update of car A's features based on action a.
- Car C assumed to be driving with constant velocity

TASK SEPARATION

SPLIT-MERGE CLUSTERING : OVERVIEW

- To determine the number of different tasks being performed by the driver of car A, i.e., number of sections to divide the road into.
- Each section's behavior is 'homogenous' and heterogenous w.r.t. adjacent sections
- Analogous to segmentation in Image Processing



SPLIT-MERGE CLUSTERING : OVERVIEW (contd…)

- Pixel intensity >> Section behavior →

0	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
2	0.039801	0.084577	0.164179	0.338308458	0.21393	0.084577114	0.074626866
3	0.006897	0.027586	0.117241	0.503448276	0.117241	0.110344828	0.117241379
4	0.094118	0.035294	0.082353	0.517647059	0.058824	0.070588235	0.141176471
5	0.117647	0.044118	0.102941	0.573529412	0.117647	0.029411765	0.014705882
…	…	…	…	…	…	…	…
153	…	…	…	…	…	…	…

- Measure of similarity

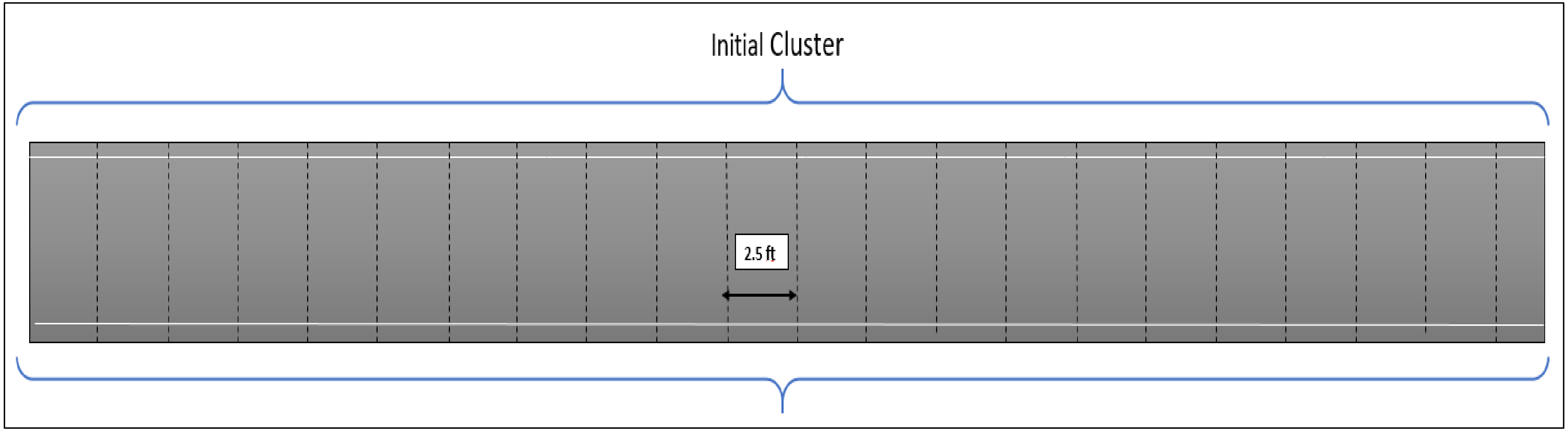
Average Hellinger distance

$$\text{Hellinger Distance : } H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2}$$

$$\text{Average Hellinger Distance : } H_{avg}(P,Q) = \frac{\sum_{S}\frac{1}{\sqrt{2}}\sqrt{\sum_{a=1}^{|A|}\left(\sqrt{p_{sa}}-\sqrt{q_{sa}}\right)^2}}{|S|}$$

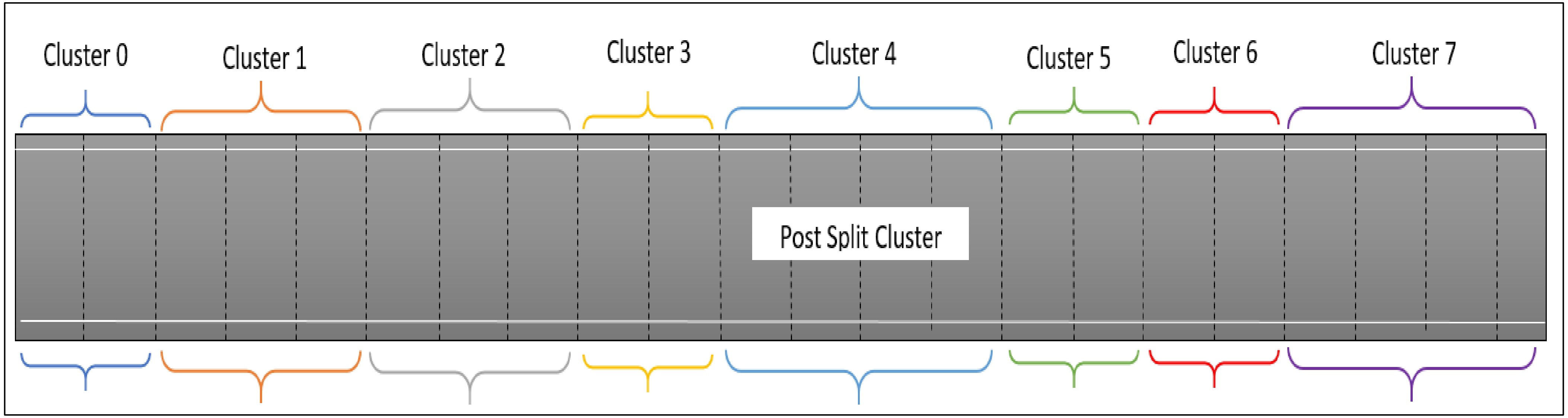
SPLIT-MERGE CLUSTERING : INITIAL CLUSTER

- Create initial cluster



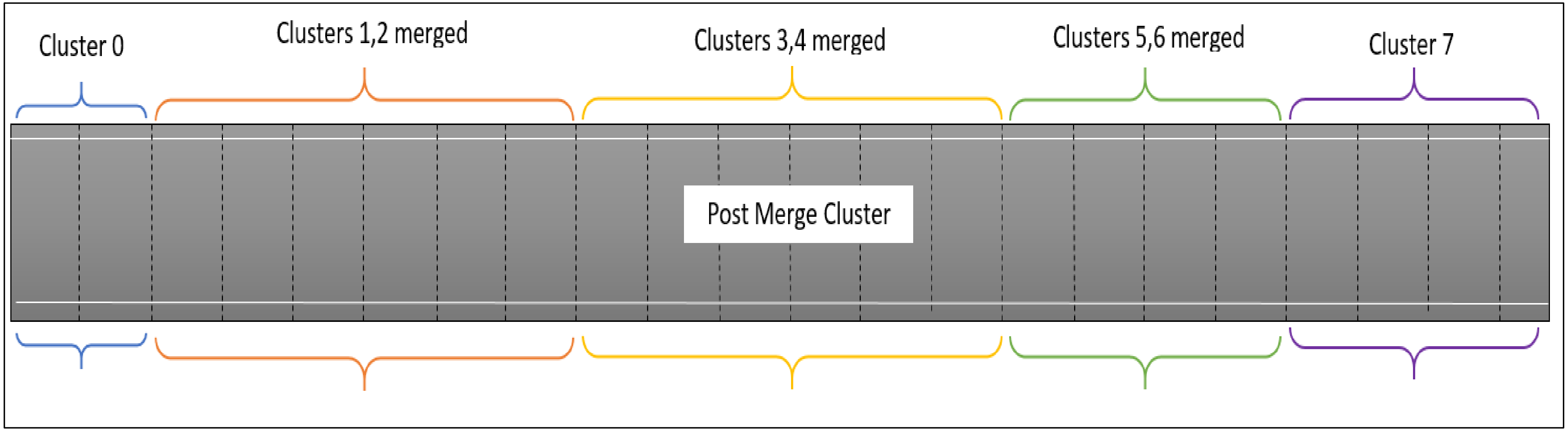
SPLIT-MERGE CLUSTERING : POST-SPLIT CLUSTERS

- Split with threshold ϵ, recursively, until you can



SPLIT-MERGE CLUSTERING : POST-MERGE CLUSTERS

- Merge iteratively with threshold $\delta > \epsilon$, until no consecutive sections can merge



- Repeat the split-merge process until convergence

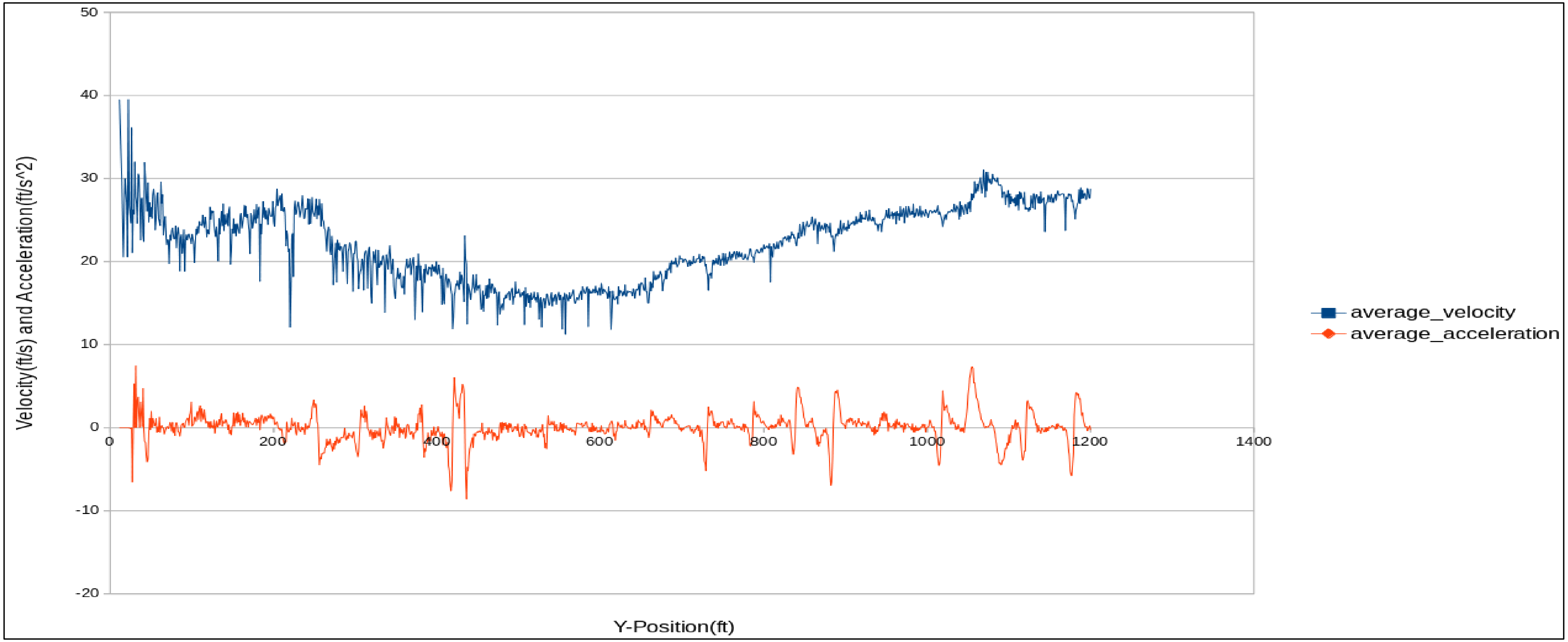
TASK SEPARATION : RESULTS

Task section#	Bounds(y-position on freeway in feet)
0	0 – 67.5
1	67.5 – 437.5
2	437.5 – 440
3	440 – 830
4	830 – 840
5	840 – 1182.5
6	1182.5 – 1190
7	1190 – 1200

Task section#	Bounds(y-position on freeway in feet)
0	0 – 440
1	440 – 840
2	840 – 1200

THINC lab

TASK SEPARATION : RESULTS COMPARISON



LIKELIHOOD WEIGHTING MULTITASK IRL

HIERARCHICAL BAYESIAN MULTITASK MODEL: BASICS AND NOTATIONS

- Trajectory: $(s^t, a^t) \equiv \{(s_1, a_1), (s_2, a_2) \ldots (s_t, a_t)\}$
  - Markovian assumption leads to
    - $s_{t+1} \sim T(S|\, s_t , a_t)$
    - $a_t \sim \pi(A|\, s_t)$
- Tasks m = 3
- Number of trajectories in task m : N_m
- n^{th} trajectory of task m: $d_{m,n} \equiv (s_{m,n}^{T_{m,n}}, a_{m,n}^{T_{m,n}})$ where $T_{m,n}$ denotes the length of n^{th} trajectory of task m
- Reward function of m^{th} task : ρ_m
- Policy of m^{th} task : π_m
- Controlled Markov Process: ν = MDP/ρ
- Problem: Find reward functions ρ_1, ρ_2, ρ_3 and policies π_1, π_2, π_3 for each of the 3 tasks

HIERARCHICAL BAYESIAN MULTITASK MODEL: BASICS AND NOTATIONS

- The likelihood of a trajectory (s^t, a^t) given a MDP μ, and policy π
- $P((s^t, a^t) \mid \mu, \pi) = P((s_1, a_1), (s_2, a_2)\ldots (s_t, a_t) \mid \mu, \pi)$

$$= T_\mu(s_1|s_0, a_0) * \pi(a_1|s_1) * T_\mu(s_2|s_1,a_1) * \pi(a_2|s_2) \ldots * T_\mu(s_t|s_{t-1},a_{t-1}) * \pi(a_t|s_t)$$

$$= \prod_{i=1}^{t} T_\mu(s_i|s_{i-1}, a_{i-1})\, \pi(a_i|s_i)$$

where $T_\mu(s_1|s_0, a_0) = T_\mu(s_1)$ which is the initial state distribution

HIERARCHICAL BAYESIAN MULTITASK MODEL: BASICS AND NOTATIONS

- Optimal Q-function for MDP μ : $Q^*_\mu(s, a)$
  - Can be computed using value iteration
- Need for a stochastic policy
- Stochastic policy needs to be parameterized so that prior can be assumed over it
- Soft-max function with inverse temperature parameter for policy

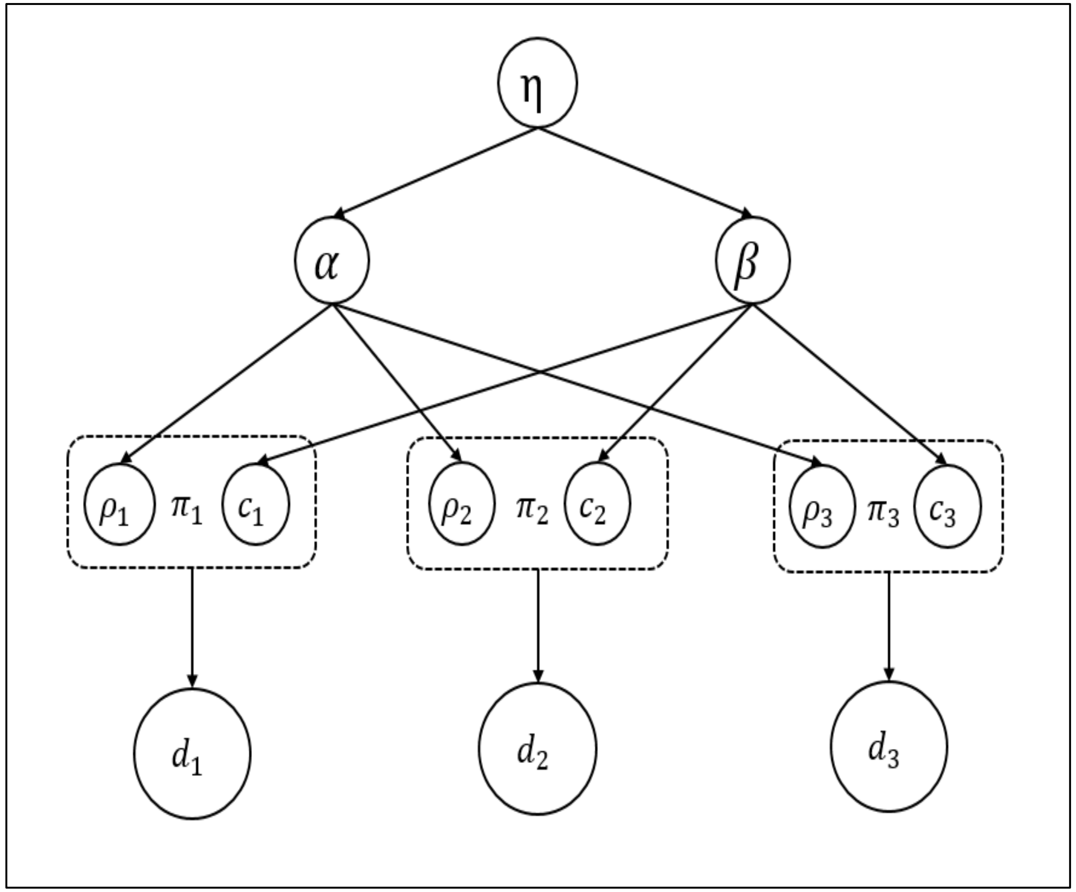
$$\pi(a_i | s_i, \mu, c) = \text{Softmax}(a_i | s_i, \mu, c) \qquad \ldots\ c \in \mathbb{R}^1$$

$$= \frac{e^{cQ^*_\mu(s_i, a_i)}}{\sum_{a \epsilon A} e^{cQ^*_\mu(s_i, a)}}$$

- **μ and ρ can be interchangeable** since CMP *ν* is constant

HIERARCHICAL BAYESIAN MULTITASK MODEL : MODEL DESCRIPTION

- Space of reward functions for each task : $\mathcal{R}$.
  - Prior over $\mathcal{R}$ with parameter α : $\psi(\rho) \equiv P(\mathcal{R} \mid \alpha)$
- Space of policies : $\mathcal{P}$
  - Policy space parameterized by c given a reward function ρ, $\pi \equiv f(c, \rho) \in \mathcal{P}$
  - Prior over c with parameter β : $\xi(c) \equiv P(\mathbb{R}^1 \mid \beta)$
- Space of α : $\mathcal{A}$ and Space of β : $\mathcal{B}$
  - Product space over the reward and c parameter priors : $\mathcal{A} \times \mathcal{B}$
  - Product hyperprior over $\mathcal{A} \times \mathcal{B}$: $\eta \equiv (\eta_\alpha, \eta_\beta)$
    - $\eta(\alpha, \beta) = \eta_\alpha(\alpha) \cdot \eta_\beta(\beta)$
    - $\eta_\alpha(\alpha) \equiv P(\alpha \mid \eta_\alpha)$
    - $\eta_\beta(\beta) \equiv P(\beta \mid \eta_\beta$



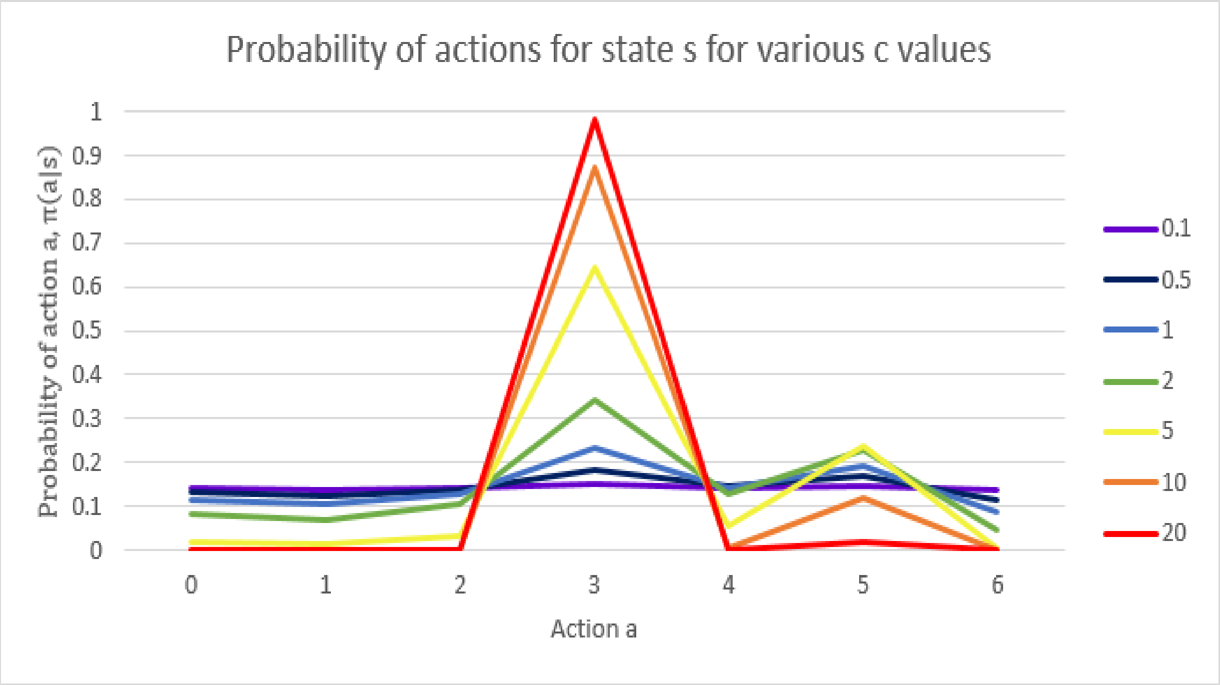
CHOICE OF PRIORS : OVERVIEW

- Choice of priors for each of the 4 variables - ρ, c, α, β
  - Prior over c : Soft-max Prior
  - Prior over the Soft-max c-prior : Soft-max hyperprior
  - Prior over ρ : Reward prior
  - Prior over the Reward prior : Reward hyperprior

CHOICE OF PRIORS : SOFT-MAX PRIOR

- Effect of c parameter on a policy given the reward ρ

a	0	1	2	3	4	5	6
Q*(s,a)	30.3	30.2	30.4	31	30.5	30.8	30



  - Higher c value => peaky stochastic policies
  - Lower the c value => flatter stochastic policies
  - Too high c values => for action a corresponding to highest Q-value $Pr(a) \rightarrow 1$
- **Choice of Prior : Exp(c | β)**

  If, $\beta = 0.2$, Expected value of this distribution $= 1/\beta = 5$

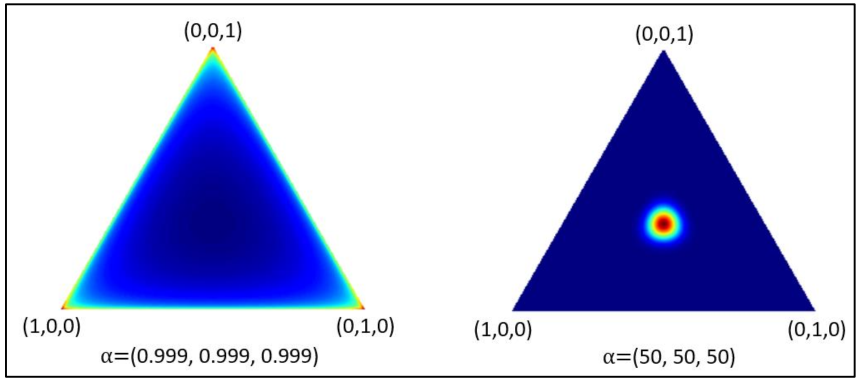
CHOICE OF PRIORS : SOFT-MAX HYPERPRIOR

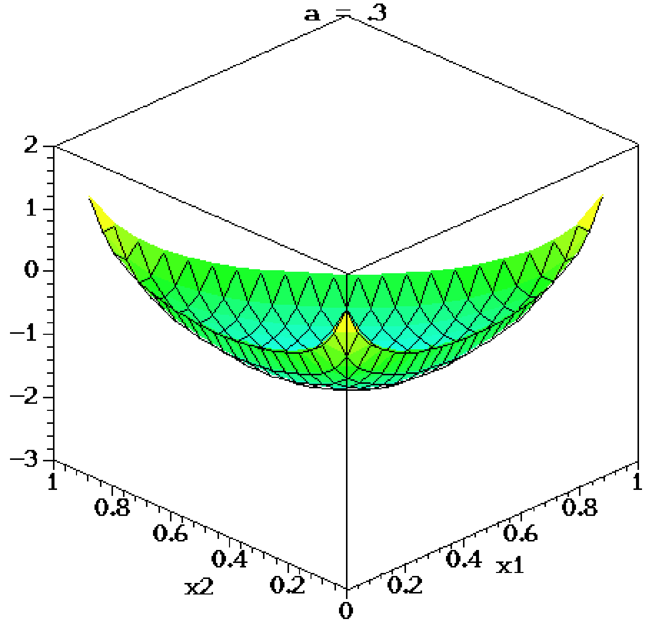
- Instead of $\beta = 0.2$, we use another Exponential distribution with expected value = 0.2 => $\eta_\beta = 5$

- **Choice of prior : Exp (β | 5)**

CHOICE OF PRIORS : REWARD PRIOR

- Dirichlet Distribution : $Dir(\alpha) = Dir(\alpha_1, \alpha_2 \ldots, \alpha_k)$. Assume $\alpha_1 = \alpha_2 = \ldots, = \alpha_k = \alpha$





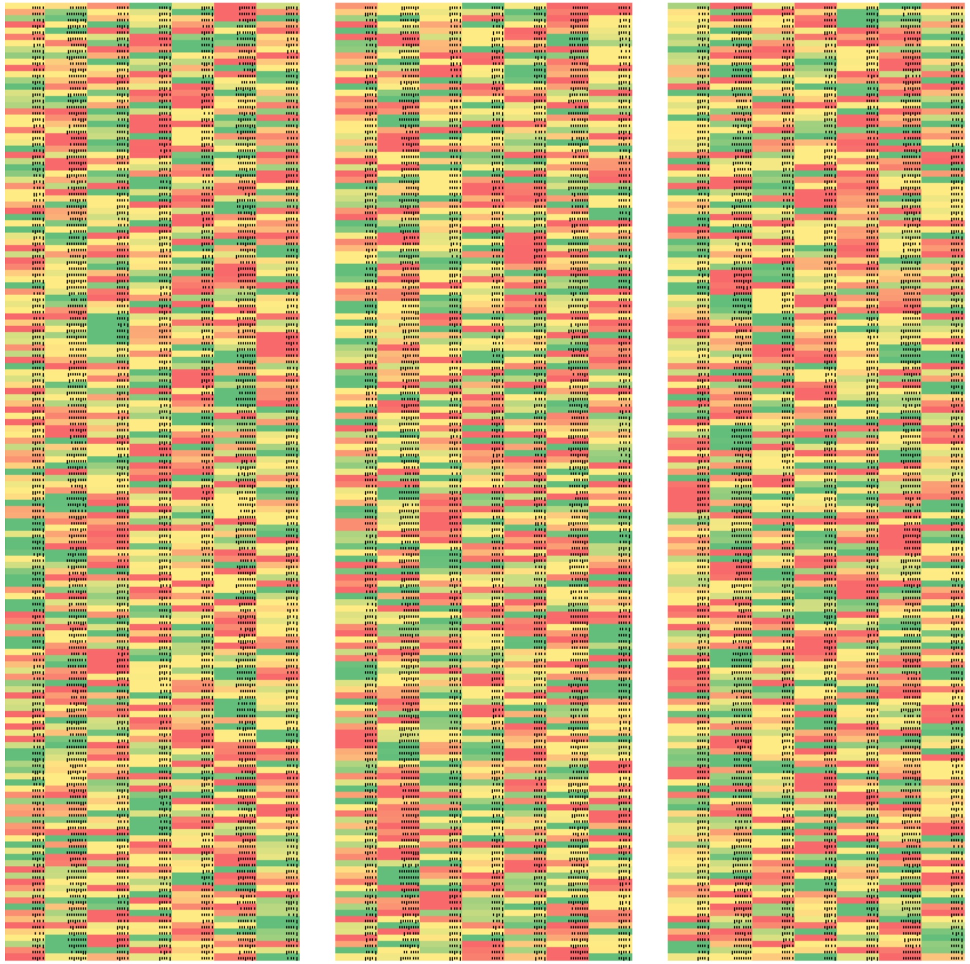
Pic courtesy: Wikipedia

- Every point $x = (x_1, x_2 \ldots, x_k)$ on the support is constrained by $\sum_{i=1}^{k} x_i = 1$
- Rewards are parsimonious, i.e. for each state, high for few actions, low for most
- One alpha per state => $|S|$ α parameters
- **Choice of Prior : Product-Dirichlet(ρ | $\overline{\alpha}$) where $\overline{\alpha} \in \mathbb{R}^{|S|}$**

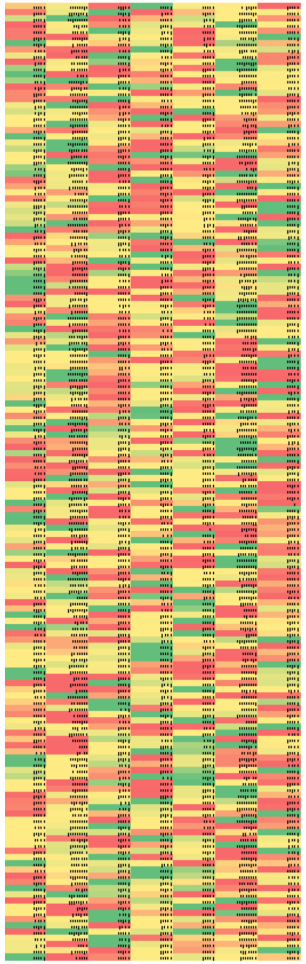
CHOICE OF PRIORS : REWARD HYPERPRIOR

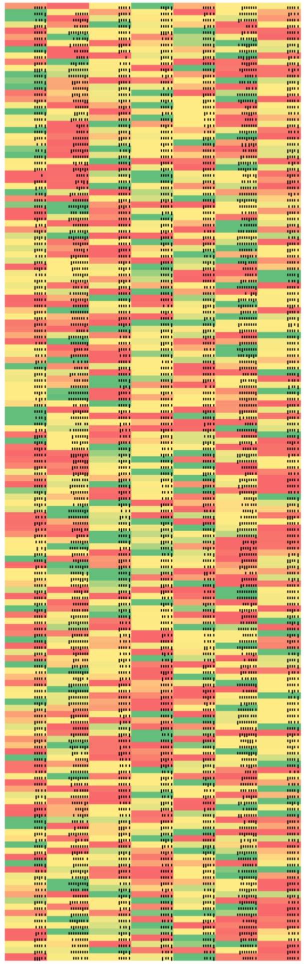
- We need low α values for good Dirichlet priors
- A sharp Exponential Distribution, say, with expected value = 0.1 => $\eta_\alpha = 5$
- **Choice of Prior : Exp(α | 10)**

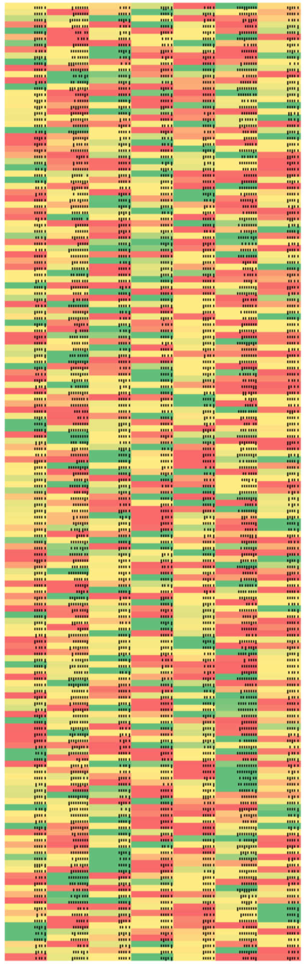
RESULTS : EXPECTED REWARDS



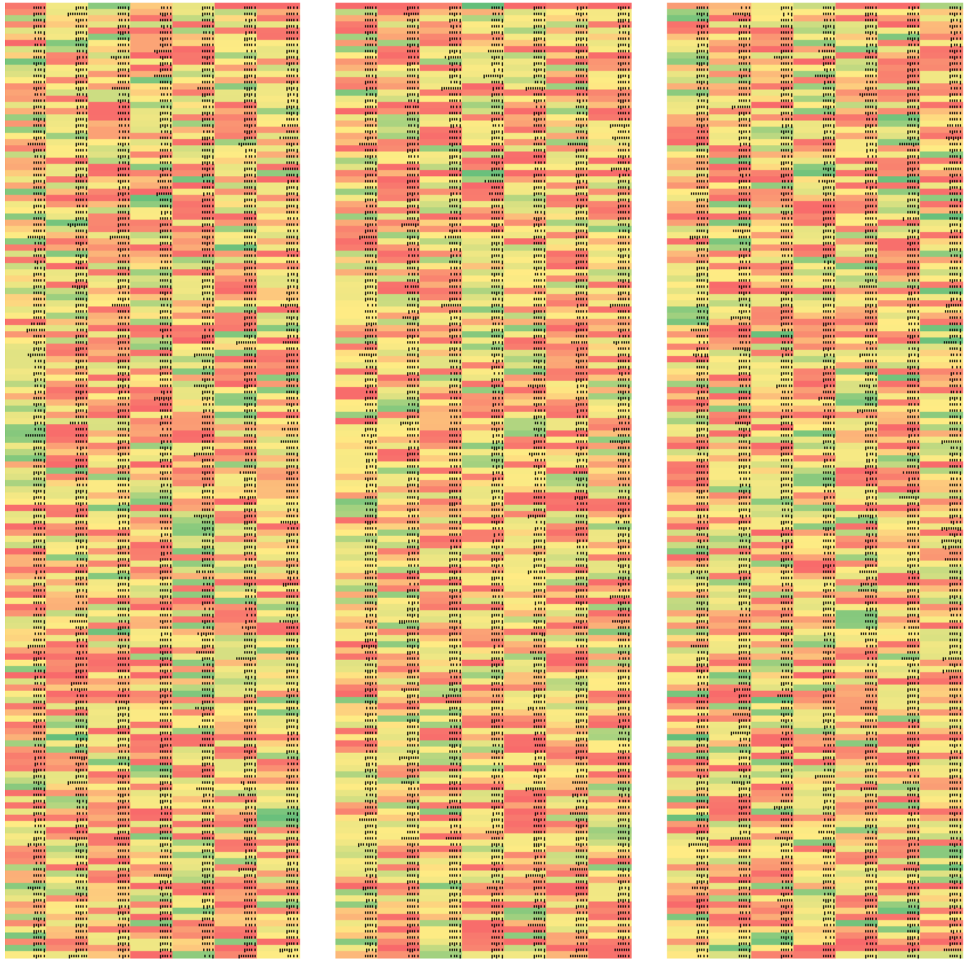
RESULTS : EXPECTED REWARDS



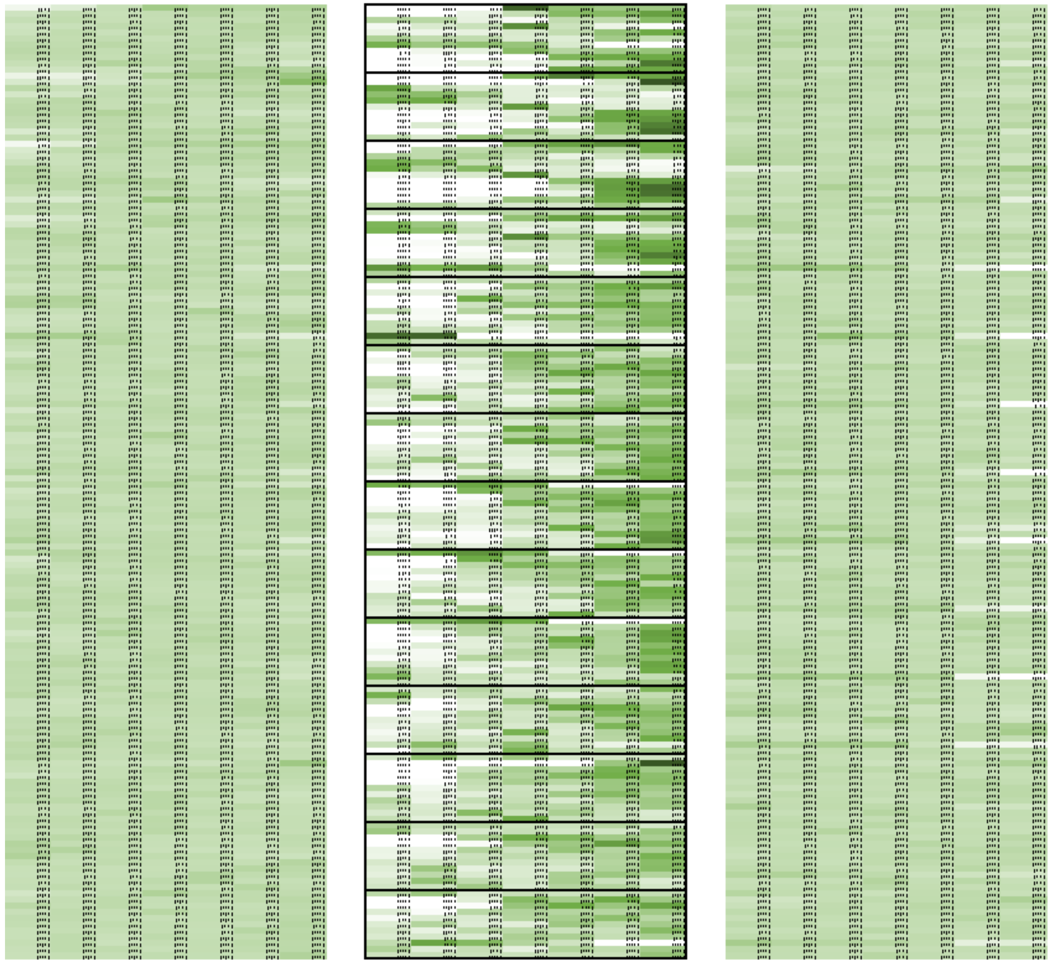




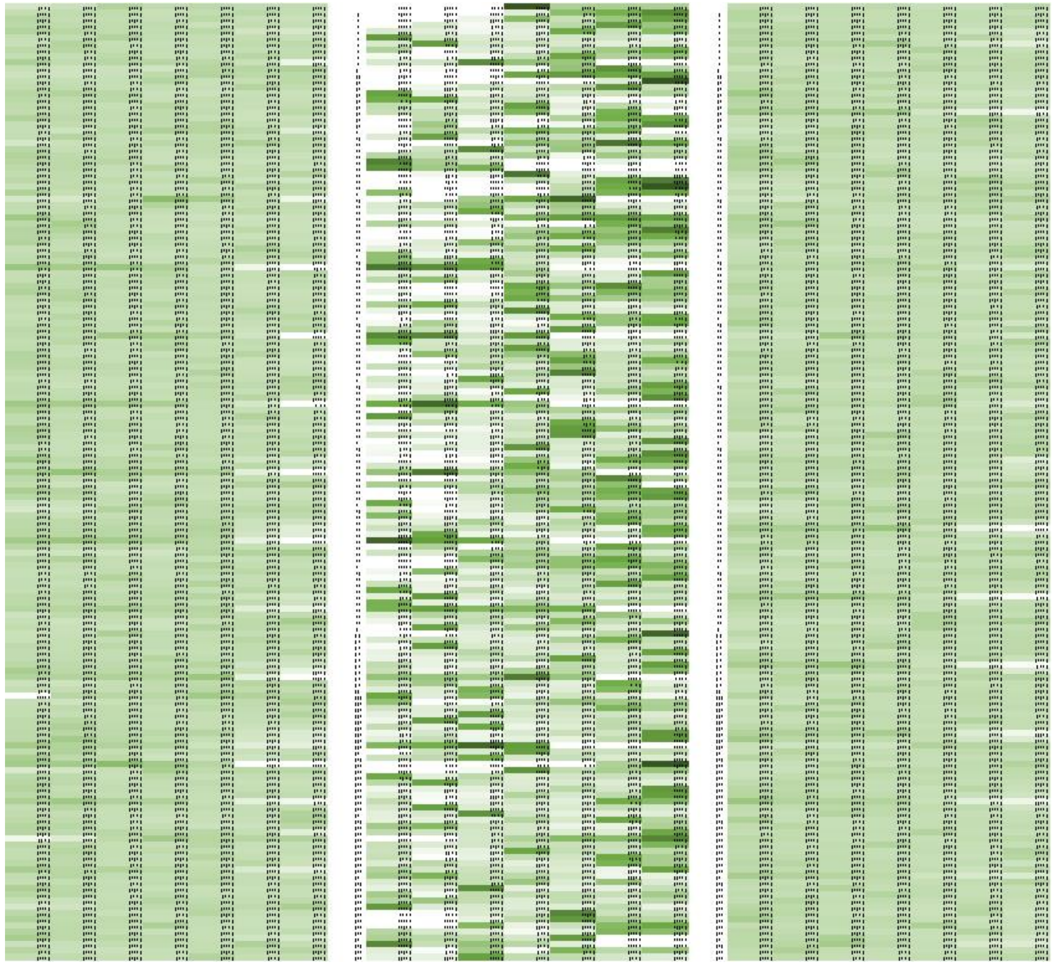
RESULTS : EXPECTED REWARDS



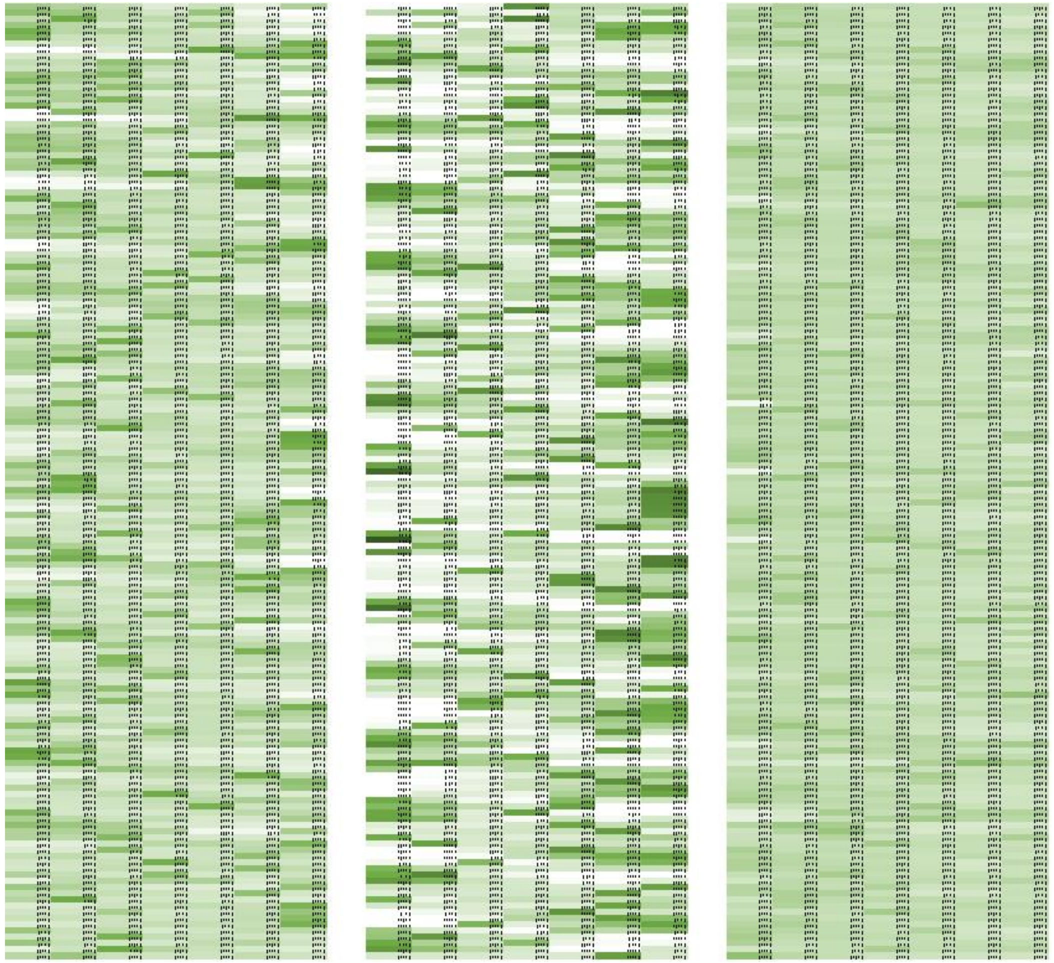
RESULTS : EXPECTED POLICIES



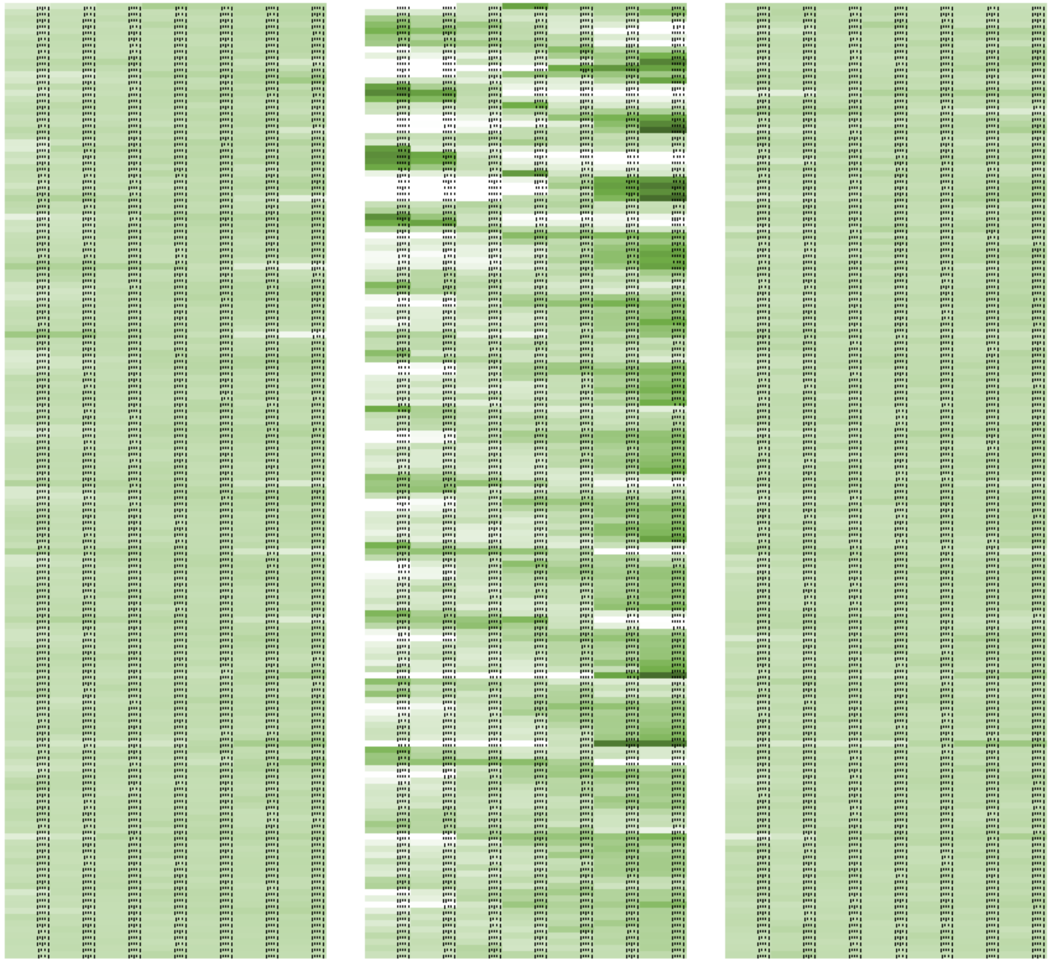
RESULTS : EXPECTED POLICIES



RESULTS : EXPECTED POLICIES



RESULTS : EXPECTED POLICIES



CONCLUSION AND FUTURE WORK

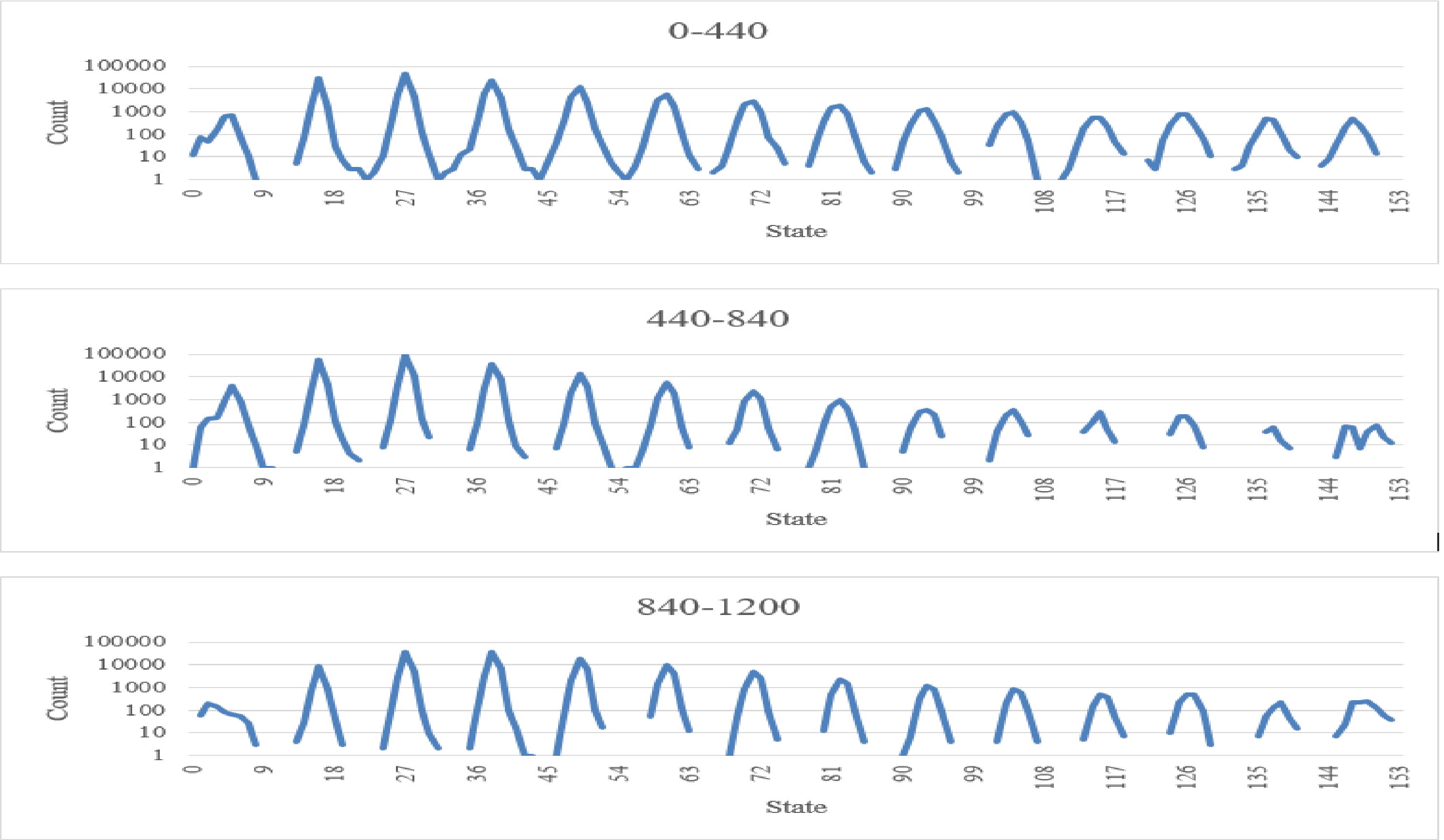
CONCLUSION

- High variance in expected rewards from different runs
- No discernible pattern suggesting difference in reward functions for the tasks

- Policies follow a trend
- Task 2 policy favors braking/acceleration actions more than constant velocity
- Peakier policies for task 2 compared to task 1 and task 3

- Suspected main cause: Not enough information about all states in the trajectories!

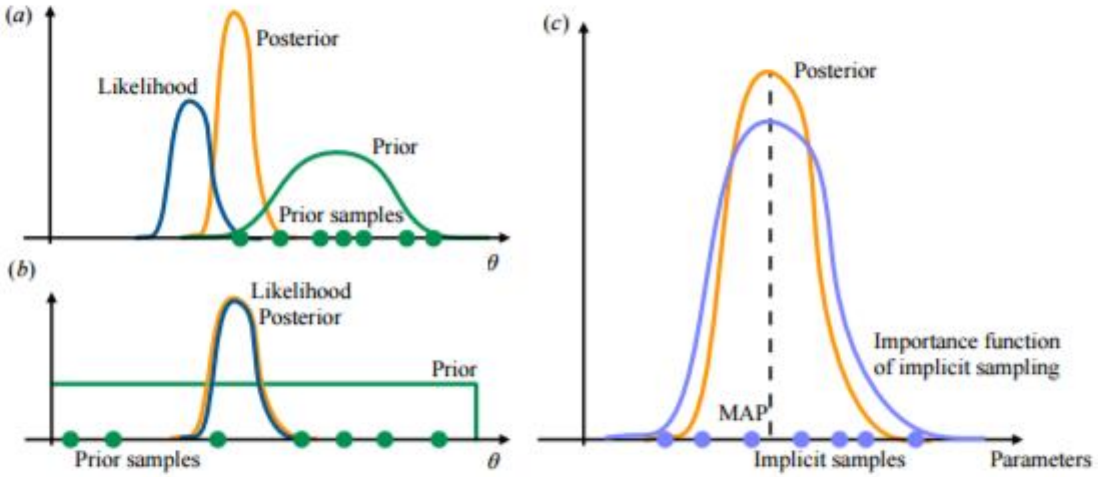
CONCLUSION



State counts in all trajectories

FUTURE WORK

- Implicit sampling
  - Generating implicit samples that receive large posterior probability instead of generating a large number of prior samples which rarely fall in the high posterior region.



FUTURE WORK

- Eliminate c parameter, reduce network complexity
  - Attempt to learn rewards of the expert than trying to best it
  - Computing stochastic policies as

$$\pi^*(a \mid s) \propto \sum_{s'} T\,(s, a, s') * V(s')$$

$$\pi^*(a \mid s) = \frac{\sum_{s'} T\,(s, a, s') * V(s')}{\sum_{a'} \sum_{s'} T\,(s, a', s') * V(s')}$$

FUTURE WORK

- Better state space
  - Non-uniform discretization
  - Highlight minute differences between nearby states which may have been fused inadvertently
  - Reduce states which are missing in trajectories by merging them together

REFERENCES

- [1] Jochem, T., Pomerleau, D., Kumar, B. and Armstrong, J., 1995, September. PANS: A portable navigation platform. In *Intelligent Vehicles' 95 Symposium., Proceedings of the* (pp. 107-112). IEEE.
- [2] Das, I., 2016, August. Inverse reinforcement learning of risk-sensitive utility. (Masters' Thesis, University of Georgia Athens)
- [3] Russell, S.J., Norvig, P. & Davis, E., 2010. *Artificial intelligence: a modern approach* 3rd ed.(Vol 2), Boston: Pearson.
- [4] Russell, S., 1998, July. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 101-103). ACM.
- [5] Ng, A.Y. and Russell, S.J., 2000, June. Algorithms for inverse reinforcement learning. In *Icml* (pp. 663-670).
- [6] Ziebart, B.D., Maas, A.L., Bagnell, J.A. and Dey, A.K., 2008, July. Maximum Entropy Inverse Reinforcement Learning. In *AAAI* (Vol. 8, pp. 1433-1438).
- [7] Ramachandran, D. and Amir, E., 2007. Bayesian inverse reinforcement learning. *Urbana*, *51*(61801), pp.1-4.
- [8] Rothkopf, C. and Dimitrakakis, C., 2011. Preference elicitation and inverse reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, pp.34-48.

REFERENCES (Contd…)

- [9] Dimitrakakis, C. and Rothkopf, C.A., 2011, September. Bayesian multitask inverse reinforcement learning. In *European Workshop on Reinforcement Learning* (pp. 273-284). Springer Berlin Heidelberg.
- [10] Babes, M., Marivate, V., Subramanian, K. and Littman, M.L., 2011. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 897-904).
- [11] Choi, J. and Kim, K.E., 2012. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems* (pp. 305-313).
- [12] Wilson, A., Fern, A., Ray, S. and Tadepalli, P., 2007, June. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *Proceedings of the 24th international conference on Machine learning* (pp. 1015-1022). ACM.
- [13] US Department of Transportation, 2006, December. Fact Sheet-Interstate 80 Freeway Dataset, *Federal HighWay Administration (FHWA)*.
- [14] Geweke, J., 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pp.1317-1339.
- [15] Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic robotics*. MIT press.
- [16] Morzfeld, M., Tu, X., Wilkening, J. and Chorin, A., 2015. Parameter estimation by implicit sampling. *Communications in Applied Mathematics and Computational Science*, *10*(2), pp.205-225.