# Modeling Human Street-Crossing Behaviour with Bayesian Q-Learning

**Juan Rios (jsrios@wisc.edu)**
Department of Computer Science, 1210 W. Dayton Street
Madison, WI 53706 USA

**Abstract**

Reinforcement learning algorithms are used to model human street-crossing behavior. The premise of [1] observes the behavior of street-crossing pedestrians, and models the decision of crossing or waiting at a crosswalk with oncoming traffic as a logistic function. This paper aims to find if Bayesian Q-Learning as proposed by [2] is more suitable to model such behavior than traditional Q-Learning when the agent is faced with uncertain state observations. The results find that both algorithms can model human behavior. Specifically, both algorithms can predict the critical time gap in which pedestrians switch from waiting to crossing with reasonable accuracy, but the Bayesian Q-learning algorithm better modeled the smoothness in decision making of the pedestrians.

**Keywords:** Bayesian Q-Learning, Reinforcement Learning

## Introduction

The action of street crossing is an affordance that entails crossing a road with a certain distance in time before observed oncoming traffic arrives [1]. Here, the authors videotaped pedestrians crossing a road in a natural setting, where the speed limit is 50 km/hr. 449 observations were made, and their decisions to cross or wait were plotted as a function of the available time gap. The time gap is the amount of time it takes for the next oncoming vehicle to arrive at the crosswalk. A logistic function was fitted to the observed data, and the authors found that in this particular setting a critical time gap of 4.63 seconds, which is the time gap in which a pedestrian transitions in behavior: choosing to cross or choosing to wait.

This paper aims to explore if these street-crossing behaviors can be modeled using agents trained with reinforcement learning algorithms. One such algorithm is traditional Q-learning. To give a brief overview, an agent in an environment can observe a set of possible states $S$ and a set of possible actions $A$ in $s$ where $s$ is a state in the set $S$. The agent interacts with the environment ,by observing $s$ and performing an action $a$, where a is a possible action in $A$. The environment transitions to the next state $s'$, and a reward at $s$' is observed. The goal of the agent is to maximize the total expected reward $E[\sum_i \gamma r_i]$ where $r_i$ is the received reward at step $i$, and $0 < \gamma < 1$ is a discount factor. To achieve this, the agent learns the Q-value $Q(s,a)$ of a state-action pair, which is the value executing $a$ at state $s$. Through this, the agent learns the optimal Q-value at each state and action, and chooses $a$ at $s$ with the highest Q-value [3]. Learning Q-values is done through the bellman equation [4].

One consideration is we assume that the agent can observe the current state $s$, but considering the street-crossing scenario, the pedestrian only has an uncertain estimation of $s$. It is assumed that humans can visually estimate the time gap $t$. Furthermore, it is assumed that through prior experience, these pedestrians have an intuition of the time gap based on the observed position and velocity of the oncoming vehicle.

Rhodes [5] describes the application of Bayesian Inference to model human perception of timed events. Rhodes describes that when participants are asked to reproduce the duration of prior stimuli, participants overestimate short events and underestimate long events. Additionally, prior exposure to similar events affects current estimations of present events. This application of the Bayesian Inference describes the *prior* as knowledge of the world, and the *likelihood* as current sensory information. The *posterior*, the perception of timing of an event, follows a distribution. The implication, as described by Rhodes, is that an ideal bayesian observer, given this *posterior*, chooses an action based on the cost or success of a potential response given the posterior estimate.

In the street-crossing example, we can describe the state $s$ as the state of the vehicle. Which can be expressed as the distance of the vehicle to the crosswalk, and the vehicle's velocity, which is assumed to be constant. This can be summarized as the time gap $t$, the time in seconds that will take the vehicle to arrive at the crosswalk. But the conclusion from [5] is that the pedestrian has uncertainty of the state of the world. One way to approach this, is to have the agent update its *posterior* estimate of the state through observations, and in turn allowing the best estimation of the true state to receive an update of its Q-value given $a$.

A second approach, which is the focal point of this paper, is that the most probable state is not learned, the uncertainty of the observed state is constant. Instead, the agent learns to update the *posterior* of the Q-value of the observed state by interacting with the environment. To achieve this, we apply Bayesian Q-learning to model the street-crossing behavior. It is hypothesized that a Bayesian agent will model the behavior of street-crossing pedestrians better than traditional Q-learning. A summary of Bayesian is summarized in the following section.

### Bayesian Q-learning

The main difference between Bayesian Q-Learning and traditional Q-learning is that while traditional Q-learning

holds point values for the estimate of $Q(s,a)$, Bayesian Q-learning holds a distribution of $Q(s,a)$ [2]. Let $R_{s,a}$ be a random variable that describes the total discounted reward when performing $a$ at $s$. The value of this reward is uncertain, and the Q- value is $E[R_{s,a}]$. The distribution of $R_{s,a}$ is assumed normal, and is described by the mean $\mu_{s,a}$ and precision $\tau_{s,a}$, where the precision is $1/\sigma_{s,a}^2$. Prior distribution over the mean and precision is a normal-gamma distribution. and this distribution is determined by the following hyperparameters: $\mu_0$, $\lambda$, $\alpha$, $\beta$. This collection of hyper parameters will now be described as $\varrho$. given this, the probability of the mean and precision is given as $p(\mu_{s,a}, \tau_{s,a}) \sim NG(\varrho)$. As the agent observes $n$ random and independent samples of the reward, the *posterior* distribution of the mean and variance can be found. This posterior is also a normal-gamma, and is described as $p(\mu_{s,a}, \tau_{s,a} \mid r_1, \ldots, r_n) \sim NG(\varrho')$. These assumptions allow the agent to store the distributions of the Q-value as $\varrho_{s,a}$ [2].

The authors of [2] detail various ways to update the *posterior*. In this paper, *moment updating* is performed, where the update of $\varrho$ *to* $\varrho'$ is a function of the parameters $M_1$, $M_2$, and $n$. The details of computing these momentum parameters and their role in updating $\varrho$ are described in the methodology section, and the explanation of the posterior update is detailed in [2]..

In traditional Q-learning, the notion of exploration versus exploitation is performed explicitly with some probability, letting the agent choose a random action, or choose the action at the current state which has the largest Q-value. In Bayesian Q-learning, the authors of [2] device multiple ways of choosing an action and balancing exploration versus exploitation implicitly. For this paper, the method chosen is termed *myopic-VPI selection*. The premise is that the expected gains of exploration are balanced against the cost of performing a potentially suboptimal action. The first thing to consider is how much can be gained by learning the true value of $\mu_{s,a}$. Specifically, we are interested in the event when the new knowledge changes the agent's policy, and this can happen over two scenarios: (1) an action thought to be the best action is found as being sub-optimal, and (2) an action thought to be sub-optimal is found to be the best action. The detailed explanation, and equations of this action-selection method are described in [2].

## Methodology

Consider the simulated world as shown in Figure 1. The agent starts on one side of the road, and the goal is to cross the street safely. The agent observes the state of the world $s$, where as mentioned earlier, $s$ is the estimated time it takes for the observed vehicle to arrive at the crosswalk. The crosswalk is the square between the agent start and the agent goal. The simulation allows the agent to have a predefined crossing time, which is the time in seconds that it takes for the agent to reach the goal. The range of crossing times range from 2 to 5 seconds, as is observed in the pedestrians in [1]. If the agent reaches the goal, a reward of +1 is observed, if the agent is hit, the reward is -1, and if the agent decides not to cross, the reward is +0.1.

Previously mentioned, the agent observes a state that is not always equal to the true state. recalling the implications found in [5], the observed state is assumed to take a distribution around the true state $s^*$. The observed state distortion used is summarized in Table 1. When the the environment provides the true state $s^*$, the agent has with a probability distribution, observing the observed state $s$ with a value distributed around $s^*$.

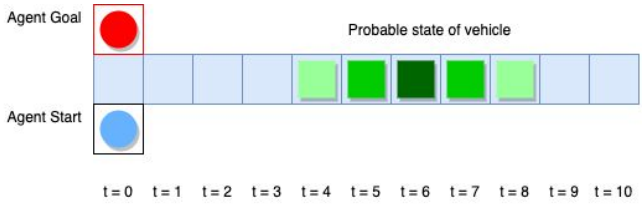



Figure 1: a visualization of the simulated world. The agent starts and observes the state of the world $s$. Based on $s$, the agent chooses to stay or to cross. Although $s$ is noisy, the outcome of the simulation is determined by the true state $s^*$. The goal of the agent is to reach the square designated as the agent goal.

The agents can take 3 forms: (1) Traditional, (2) Bayesian, and (3) Random. The traditional agent maintains a Q-table with point values. The action chosen is based on an epsilon greedy strategy, where with probability $\epsilon$, a random action is chosen, and the best action is chosen with probability 1- $\epsilon$. The value of $\epsilon$ decays at an exponential rate, and is a function of the current training trial and the total number of trials. For early trials, the agent is likely to choose random actions, providing exploration of the state-actions pairs. As the trials progress, the probability of choosing the best action increases, increasing the rate of exploitation. The Q-values are updated with the bellman equation as shown:

$$Q(s,\ a)\ =\ (1\ -\ )Q(s,a)\ +\ (r + \gamma max_{a'}(Q(s',\ a')))$$

Where $s$ is the current observed state, $a$ is the action chosen at $s$, is the learning rate chosen to be 0.01, $s'$ is the observed next state after performing $a$, and $a'$ is the action at $s'$ that yields the highest Q-value for the $s'$, a' pair. $r$ is the reward observed when transitioning from $s$ to $s'$. The value of the true next state $s'^*$ is determined by decrementing $s^*$ by unity. $\gamma$ is chosen to be 0.99

Table 1: distortion of the observed state $s$ from the true state $s^*$.

| value of $s$ | Probability |
|---|---|
| $s^*+2$ | 0.13 |
| $s^*+1$ | 0.17 |
| $s^*$ | 0.40 |
| $s^*-1$ | 0.17 |
| $s^*-2$ | 0.13 |

The Bayesian-learning agent maintains a table with $\varrho$ for each state-action pair. $\varrho$ is described by the parameters $\mu_0$, $\lambda$, $\alpha$, $\beta$, which describe the normal-gamma distribution for the mean and precision parameters $\mu_{s,a}$ , $\tau_s$ of the total discounted reward $R_{s,a}$.

The action chosen is based on *myopic-VPI selection*. the action chosen is one such that the following is maximized:

$$E[Q(s,\ a)]\ +\ V(s,\ a)$$

The *E[Q(s, a)]* is $\mu_0$, since Q(s,a) is $\mu_{s,a}$, which is sampled randomly from the normal-gamma distribution, and *V(s, a)* is computed as:

$$V(s,\ a)\ =\ c\ +E[_{s,a2}]\ -\ \ E[_{s,a1}]\,P(\ _{s,a1}\ <\ E[_{s,a2}]\ )$$

Where *a1* is the best action, and *a2* is the second best action, and $a1 = a$. An action $a$ is the best action when $\mu_{0,s,a} > \mu_{0,s,a'}$ where in this context, *a'* is every other action. In the case that $a \neq a1$ (the current considered $a$ is not the best action), then V(s, a) is:

$$V(s,\ a)\ =\ c\ +E[_{s,a}]\ -\ \ E[_{s,a1}]\,Pr(\ _{s,a}\ >\ E[_{s,a1}]\ )$$

$$c = \frac{\alpha_{s,a}\Gamma(\alpha_{s,a}+\frac{1}{2})\sqrt{\beta_{s,a}}}{(\alpha_{s,a}-\frac{1}{2})\Gamma(\alpha_{s,a})\Gamma(\frac{1}{2})\alpha_{s,a}\sqrt{2\lambda_{s,a}}}\left(1+\frac{E^2[\mu_{s,a}]}{2\alpha_{s,a}}\right)^{-\alpha_{s,a}+\frac{1}{2}}$$

and *Pr( $\mu < X$ )* is*:*

$$Pr(\mu < x) = T((x-\mu_0)\left(\frac{\lambda\alpha}{\beta}\right)^{\frac{1}{2}} : 2\alpha)$$

Where *T(x : d)* is the cumulative t-distribution for $d$ degrees of freedom. Upon observing a reward, the parameters in $\varrho$ are updated through *moment updating*. The hyper parameters in $\varrho$ are updated to represent the posterior distribution. these parameters are $\mu_0$', $\lambda$', $\alpha$', $\beta$' and are updated as follows:

$$' = \ +n/2$$
$$' = \ +n$$
$$_0' = \frac{_0+nM_1}{+n}$$
$$' = \ +n(M_2\ -M_1^2)\ +\frac{n(M_1-_0)^2}{2(+n)}$$

Where $M_1$ and $M_2$ are computed as follows:

$$\begin{aligned} M_1 &= E[r+\gamma R_t] = r+\gamma E[R_t] \\ M_2 &= E[(r+\gamma R_t)^2] = E[r^2+2\gamma r R_t+\gamma^2 R_t^2] \\ &= r^2+2\gamma r E[R_t]+\gamma^2 E[R_t^2] \end{aligned}$$

$$E[R\ ] = _{0,t}'$$
$$E[R^2] = \frac{+1}{-1}+_0$$

$R_t$ is taken to be at next state *s'* where the expected value is the highest value for any *a'* at *s'*. The random agent performs no learning, and always chooses a random action. The training progress is visualized and discussed in the results section.

A summary of the training is provided: The world generates a random state $s$ with values [1, 12], which are the time gaps plotted in [1]. $s$ is generated randomly with a bias to generate smaller valued states, for example states 3 - 8 with greater probability. The reason is that a uniform random generation delays agent learning, since randomly choosing the crossing action will lead to the majority of samples to be successful, so biasing the generation towards a more critical time gap allows the agent to learn the optimal policy quickly. In fact, it was observed that the Q-learning agent is able to perform well after a few training sessions. After, the agent chooses an action $a$ based on $s$. The reward is computed based on the the true next state *s'**, but the Q-values are updated based on the perceived $s$ and *s'*.

A number of agents are trained and each presented with a single, initial generated state, and simulation for each agent continues until the agent chooses to cross. For each agent, the number of actions collected depend on when the agent chooses to cross. For example, if the agent waits for 4 seconds and crosses on the 5th second, then 5 observations are observed. The total number of observations is 311, which match the number reported in [1]. The behavior for both the traditional and bayesian agents are observed, plotted, and logistic regression was used to fit the data.

## Results

The results for the training progress for the 3 agents when there is no noise, $s = s^*$ , are shown in Figure 2. Each agent trains for a total of 120 simulations. After each simulation, the agent is tested for 200 simulations, and during these testing simulations, performance data is collected in the form of successes per test. Success is the number of trials in which the agent did not get hit by the car. Although the desired outcome is for the agent to cross the road, often, the initial time gap observed is small enough that it's in the interest of the agent to not cross at all. The traditional Q-learning agent is able to perform well only after 20 training sessions, the small amount of error can be attributed to the agent choosing random actions that result in being hit,

since the agent will always with some chance choose to explore, even after learning the optimal policy.

The Bayesian agent falls slightly behind in training. One possibility is the bayesian agent has a higher probability of choosing suboptimal actions than the traditional learner for a given state. One downfall of Bayesian Q-learning is the implementation is far more complex than traditional Q-learning, identifying why the algorithm behaves a certain way is much more difficult. Nonetheless, both algorithms manage to achieve reasonable performance before 40 trials, and perform much better than a random picking agent.

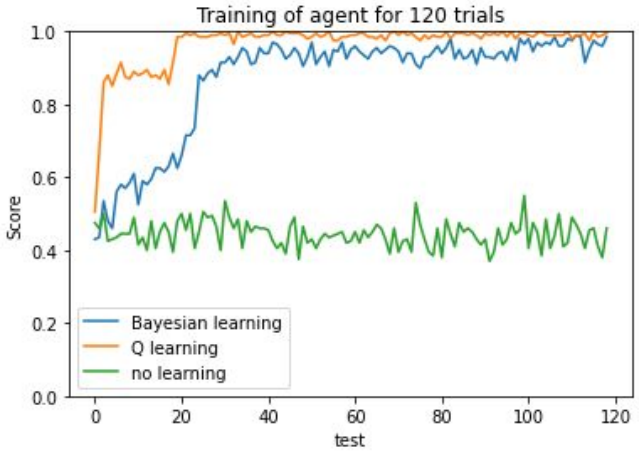



Figure 2: A visualization of the training performance when $s = s^*$ for 3 agents: (1) Bayesian Q-Learning, (2) Traditional Q-learning, and (3) an agent that picks actions randomly.

Figure 3 shows the training performance for the same 3 agents, except that with a probability of 0.4 that $s = s^*$, and with probability 0.6 that $s \neq s^*$. It was hypothesized that the bayesian learner, equipped to handle uncertainty, would outperform the traditional learner. This was not the case, and the Bayesian learner does not perform as well as the traditional agent. One reason given by [2], is that using *moment updating* to update the $\boldsymbol{\varrho}$ parameters results in the estimation of the precision of the mean to increase too fast, resulting in low exploration and the agent reaching a suboptimal policy. [2] suggest that one ad-hoc way to fix this, is to use an exponential forgetting method, where the effect of previously seen examples is reduced by a constant, meaning examples seen long ago have decreasing impact. This quick fix was not implemented and is a possible reason as to why the Bayesian learner does not perform as well. Future work should implement *mixture updating* instead, which combined with *myopic-VPI selection* gives better performance as shown in [2].

Even with noisy state observations, traditional Q-learning manages to maintain good performance. It can be seen that for this simulation, traditional Q-learning is a simple solution for a simple problem. Future work could include exploring the performance of Bayesian versus traditional in a more complex environment, where the state and actions sets $A$ and $S$ are larger.

Future work can also expand the simulation to include a more realistic model to the uncertainty of the states. For example, instead of the agent having constant chance of the true state $s^*$ being distorted *to* $s$. The cause of uncertainty could be more realistic. For example, as discussed earlier in [4], the estimate of a time event is influence by recently

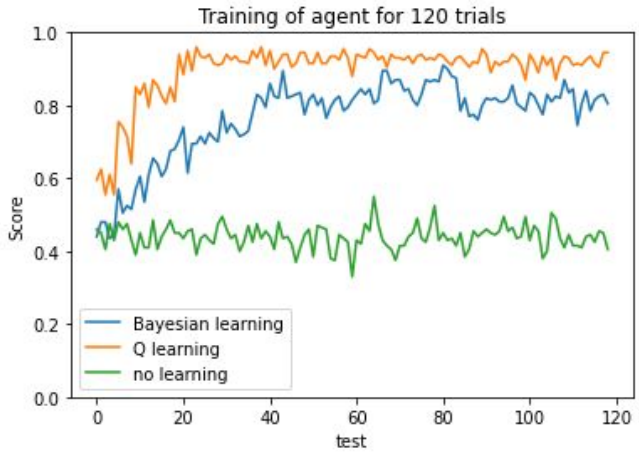



Figure 3: A visualization of the training performance for 3 agents when the chance that $s \neq s^*$ is 0.6.

experienced events. In the context of this simulation, the uncertainty of the agent could be modeled as dependent on the time gap observed for previous vehicles. Additionally, it is not unreasonable to assume that the time gap observed by a pedestrian is constantly being updated as the time gap decreases, and the current model to distort the state is noisier than actuality, specifically at time gaps with lower values.

Although the training performance for the traditional agent is better than the Bayesian agent's performance. This does not yet provide substantial insight on how well these algorithms mimic the behavior of pedestrians observed in [1]. As discussed in the methodology section, the trained agents are exposed to a testing simulation. The observational results of [1] are shown in Figure 4, the visualization for these simulations are shown in Figure 5, and the numeric results are shown in Table 2.

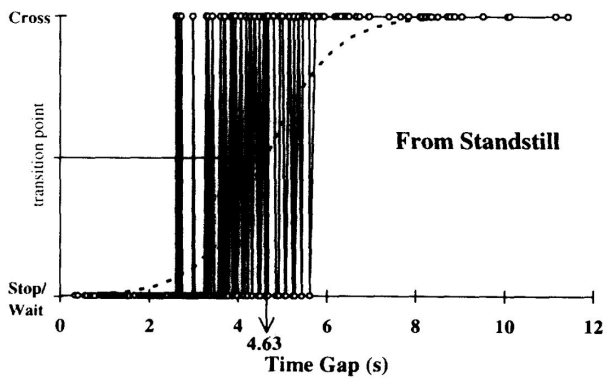
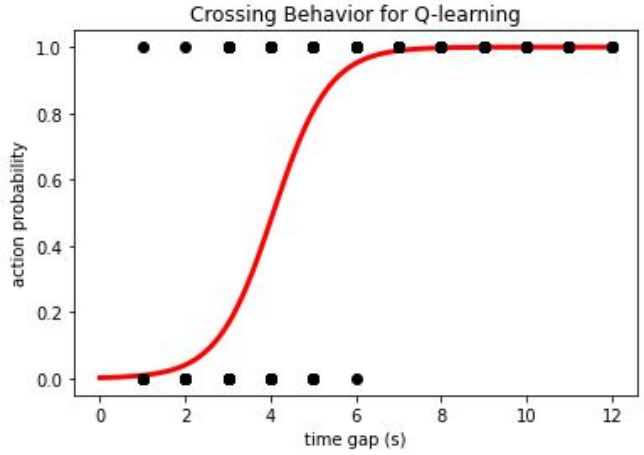
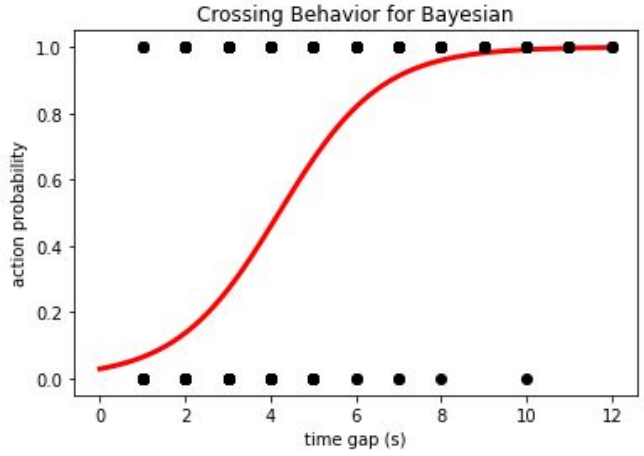



Figure 4: A plot of 311 observations of pedestrian crossing. The time gap characterizes the time where pedestrians transition in between choosing to cross and choosing to wait [1].




(a)




(b)

Figure 5: Fitting of logistical function to the observed behaviors for (a) traditional Q-learning and (b) Bayesian Q-learning. The black dots on the top axis represent crossing observations, and waiting observations on the bottom.

Table 2: critical time gaps and slopes for logistic functions.

| Description | Critical Time Gap | Slope |
|---|---|---|
| Pedestrian | 4.63 | 0.33 |
| Q-learning | 4.05 | 1.53 |
| Bayesian | 4.17 | 0.84 |

The results shown in Table 2 show that the Bayesian learner more accurately represented the results of [1]. The slope is a measurement of the abruptness of the decisions. The Q-learner has a significantly higher slope. Although the critical time gap of the Bayesian-learner is closer to the results, the difference is small. It is likely that under different but equally reasonable assumptions made during the simulation, the Q-learner could show a closer critical time gap. One behavior by the Bayesian-learner is that even with ample time, some agents waited to cross. This behavior is not observed in the traditional Q-learner or the pedestrians, and these observations increase the time gap and decrease the slope. The Q-learner does a better job in showing that no agent waits when the time gap is higher than 6 seconds, much like the pedestrians.

The results indicate that neither algorithm is significantly better than the other at modeling the pedestrian time gaps. Future work could focus on ascertaining that the Bayesian-learner be used to model the abruptness in transition in humans, since the Q-learner has a significantly more abrupt decision rule than the pedestrians.

## Conclusion

The authors of [1] observed the behavior of street crossing pedestrians and used a logistic function to model their behaviors. The street-crossing scenario was modeled using reinforcement learning algorithms. Specifically, traditional Q-learning and Bayesian Q-learning.

In terms of performance, the Q-learning algorithm converged faster than Bayesian-learning for both cases where the states are fully observable or there is agent uncertainty regarding the true state. In terms of modeling human behavior, the Bayesian algorithm better approximated the smoothness in decision-making of the pedestrians, while the Q-learner acted on a stricter decision rule.

Future work can explore the following: (1) Modeling agent estimation of true state using the Bayesian framework shown in [5], (2) implement a more realistic model for adding noise to the true states, (3) implementing *mixture updating* rather than *moment updating* to increase Bayesian-learner performance, and (4) perform more simulations with statistical analysis when comparing traditional and Bayesian learning in modeling human behavior.

The implication of this work is that reinforcement learning algorithms can model human street-crossing behavior. Then these algorithms can be implemented by city planners or civil engineering firms in charge of designing venues that are expected to experience pedestrian traffic and or vehicle traffic. There already exists efforts to model traffic with the aim of reducing congestions and improving mobility [6]. For modeling that requires deep learning, it would be worthwhile to investigate the use of deep bayesian neural networks to learn the posterior distribution of the Q-values at each state-action pair.

## References


[1] Oudejans RR, Michaels CF, van Dort B, Frissen EJP. To cross or not to cross: The effect of locomotion on street-crossing behavior. Ecological Psychology. 1996; 8:259–267.

[2] R. Dearden, N. Friedman, and S. Russell. Bayesian Q-Learning. American Association for Artificial Intelligence, 1998.

[3] C. J. Watkins and P. Dayan. Q-learning. Machine Learning, 8(3):279–292, 1992.

[4] R. E. Bellman. Dynamic Programming. Princeton Univ. Press, 1957.

[5] D. Rhodes, "Bayesian Inference in Human Time Perception", 09-May-2017. [Online]. Available: psyarxiv.com/7fzbk.

[6]W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *ScienceDirect*, 2018.