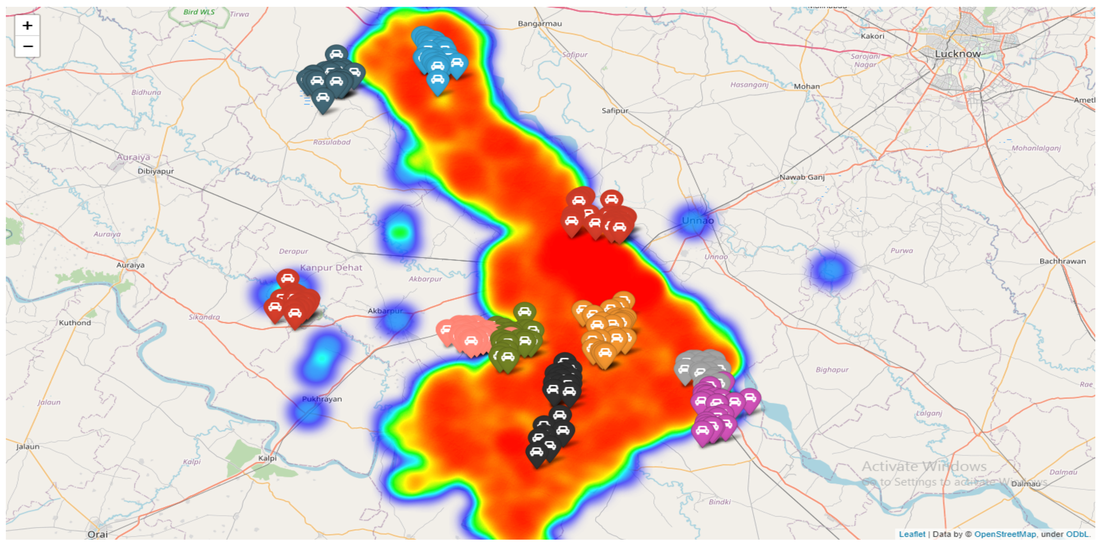
Decentralized Reinforcement Learning for Multi-Agent Patrol Routing

Avijit Roy, Nisheeth Srivastava

Indian Institute of Technology, Kanpur

Introduction

Police patrolling(First-Responder) is an integral part of the safety mechanism in our society. It is a complex process as there are many factors to consider while designing a patrol route. The main concern of a patrol vehicle is to prevent day-to-day crime by covering crime-prone areas and also to attend to any unwanted incidents that occur across the city. Patrol routing is studied in the context of police patrolling. Designing patrol routes optimally is a challenge. Police have to accomplish this with the help of the small resources that they have. With time, technology is evolving and the use of technology is also prevalent in this field. With the invention of the Global Positioning System, it is now possible to accurately locate the crime locations and areas of high crime density. Efficient routes can be constructed by taking into account the high-density crime areas(hot-spots). This problem is known as Multi-Agent Patrol Routing Problem. Reinforcement Learning is a natural fit in solving Multi-Agent Patrol Routing problems.

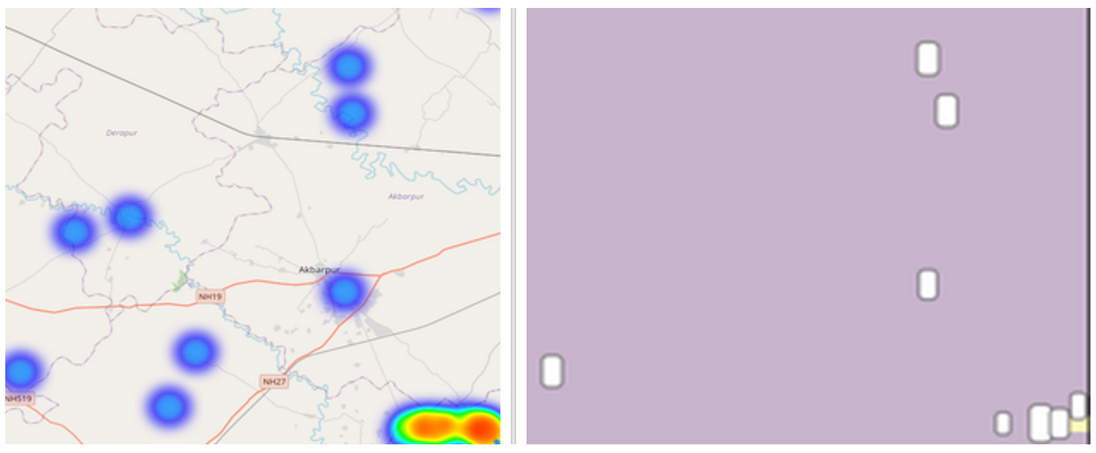


We model the placement of first-responder patrol vehicles on a city map as a multi-agent reinforcement learning problem, where individual agents learn desirable locations for parking based on dynamically updated geo-localized emergency call records. The model is able to outline reasonable patrol locations and routes, adapting to changes in the geographical pattern of call locations, and permits optimization of routes accommodating fuel economy and other cost-based concerns into account in a principled way. We also present an actual patrolling system we have developed around this model, and present simulated results comparing its performance vis-a-vis centroid-based patrol location prediction and judgments made by humans.

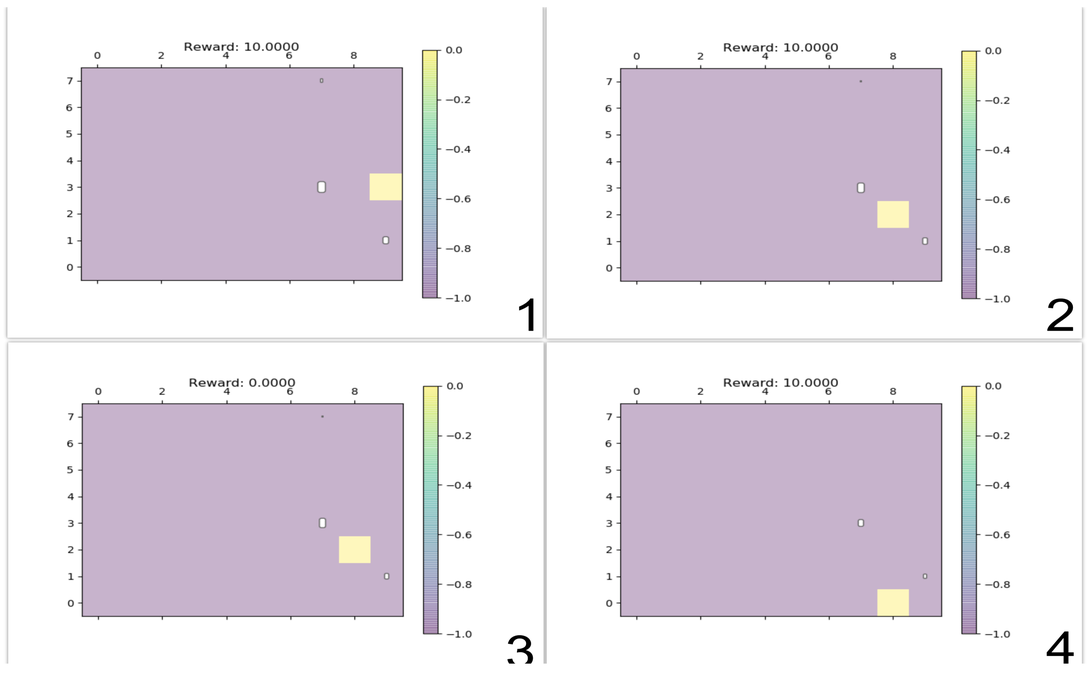
Solution Approach

Balancing exploration of an unknown environment with performing actions assigned high rewards by the system designer. The notion of states, actions, and rewards maps easily to geographical locations, patrol movement, and emergency calls in this problem domain. Traffic congestion adds dynamic uncertainty to the state transition probability, and the varying criticality of calls received from different locations makes it more realistic to treat rewards as unknown. Thus, it seems reasonable to model the multi-agent patrolling problem using reinforcement learning.

We convert the real-world locations to gridworld in the following way. We define a grid-world by taking the minimum and maximum latitude and longitude of the district, defining a rectangular grid-world using this, and then dividing the entire rectangular grid-world into $100m \times 100m$ grid locations. Vehicles move around this gridworld and crimes are mapped to respective grids as shown in the below image.



We have three sets of rewards. For Instant Reward, we generate the rewards at a certain grid location with a probability proportional to the number of calls received from that location in the past. To do this, we take real crime data and map it into the grid-world by assigning the crimes to a grid that it falls into. Instant rewards decay with each step in the simulation. For the Proximity Reward, we calculate the Euclidean distance from each of the other vehicles and calculate the reward. For step penalty, we add a penalty(α) as required for every step other than the same step.



System Description

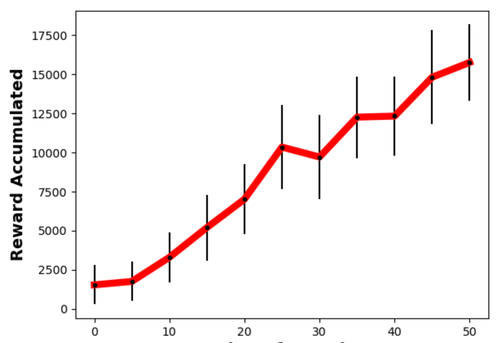
We divide the entire area covered by the EMS into district-wise zones, approximately identified by rectangular bounding boxes. We create routes for different zones separately by running the algorithm once for each zone. A web-based application allows officers to monitor and update vehicle routes. We built the web application on the MERN stack to interact with the model. We use human interaction to improve the routes over time, in the sense that our algorithm replaces its selected patrol point with a human-generated patrol point stored in the database if the human suggestion is within a certain radius of the algorithmically proposed point. We show the location of all the vehicles at the current timestamp in All Vehicles View.

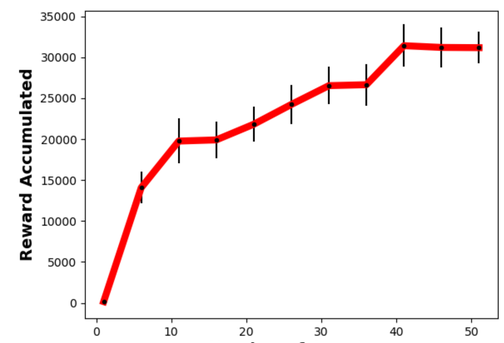
On clicking a vehicle in All Vehicles View, we redirect to Single Vehicle View. In this view, we show the route for 24 hours. We show what the vehicle's position will be at each hour. Users can update each location in the route to a more suitable location. This location will be saved in the NearestLocations collection in the database and will be used next time to generate the route, complementing the algorithm's judgment with human judgment.

Results

In our first experiment, we train the model for a different number of iterations and then run an on-policy iteration to get the total accumulated rewards in that iteration. We see that, as expected, with the increasing number of iterations the agents are learning to accumulate higher rewards together. This in turn means that the agents are learning to adapt the behavior that we intend them to adopt via rewards.

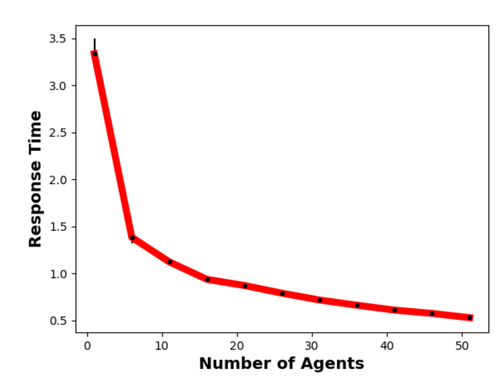
In our second experiment, we measure total accumulated rewards with an increasing number of agents in the simulation. We run the experiment in the same setup for every number of agents. We increase the number of agents and observe the total accumulated rewards. We observe that with the increasing number of agents, the accumulated reward increases. Importantly, the curve shows diminishing returns, reflecting the real-world expectation that it is counter-productive to place too many patrolling units in any given locality.

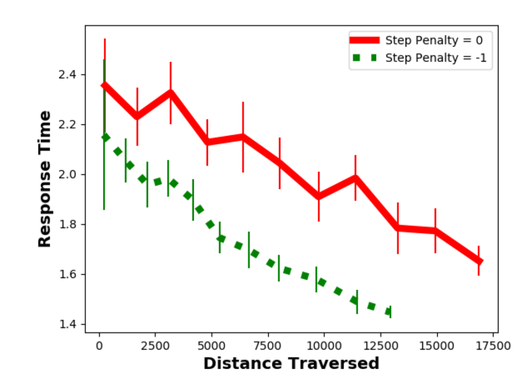




In our third experiment, we measure response time to calls with respect to the number of agents in the system. We use the distance of the nearest agent, when a call occurs, as a proxy for the response time. We observe that the Response-Time decreases drastically initially, then the rate of decrease reduces slowly and the curve flattens and looks reasonably exponential. Taken in conjunction with the results reported in the previous figure, these results clearly indicate the possibility of optimizing the number of actively patrolling agents without compromising very much on response time.

Finally, we measure response time with respect to the distance traversed by all the agents for different values of step penalty. We observe that when there is no step penalty, the agents tend to move more and it decreases with increasing step penalty. By adding step penalty we can see that the fleet as a whole travels less.





To compare the system-generated patrol points with human-generated points, we conducted one experiment as follows. We have taken 4 different zones with different densities of crime. Then we generate 5 types of patrolling points for each of the zones. Now for each patrol point, a user rates the patrolling points out of 100 based on the factors that how much area it covers total distance covered, and response time. The users involved in this experiment were oblivious to the type of patrolling points they are rating.