In chapter 2 of Reinforcement Learning, we learned several methods to balance exploration and exploitation. And we plot graph and draw lines to compare them. I find that the y-axis, which is the average reward, is scalable data. We use some actual digits to define rewards. However, the rewards of some human actions are hard to define. How can we use reinforcement learning to deal with tasks that their rewards cannot be easily defined?

I’d like to use an example to share my idea. Suppose in a game we want agents to transport goods to safe zone. The agents can drive cars, but enzymes are more likely to find the agents. Enzymes have a lower probability to find the agents if they don’t take vehicles. We want use reinforcement learning to find best approach. I think we can take use of supervising learning first. We input a large amount of data and let agents to generate possible rewards for different actions. Then the agents use the generated reward model to do reinforcement learning. The data can be the time for different ways, the condition of vehicles and condition of road so on. Given those data, agents can estimate the effect of different actions and build the probability model. With the reward model, agents explore or exploit to decide whether to take vehicles, when and where to take vehicles.

Sometimes we can only access a small amount of data. The combination of supervising learning and reinforcement learning may not achieve goals with imitated data. Here we can use some success case as the model and let agents to analysis the model. Then we make agents try to find a reward function to fulfill our goals (or statement). Agents give larger rewards to those actions that like success case. With

These estimate rewards, agents can do reinforcement learning.

Above all are my ideas to solve the actions without a defendable reward.