Robôs podem adaptar o seu comportamento ao aprender através de contribuição humana como demonstrações e correções. Por exemplo, aqui o robô começa com uma trajetória obtida otimizando alguma função de recompensa pré-especificada tipicamente codificada como uma **trade-off thate** entre recursos importantes v da tarefa, como distâncias até aos obstáculos e eficiência.

Ao observar uma tarefa realizada pelo robô, o humano percebe que obtém um resultado aquém do desejável ou que se esperava observar.

Isto porque o robô segura a caneca a uma distância da mesa muito grande, e que prova não ser muito realista quando em comparação com o humano.

Desta maneira, o humano puxa o braço para baixo e trabalho antigo mostra como o robô consegue interpretar essa introdução humana como evidencia sobre o interesse das pessoas e adapta o seu entendimento de recompensa em dados.

Assim, o robô pode otimizar esta nova função de recompensa para realizar a tarefa de uma maneira que se aproxima do desejo da pessoa.

Isto funciona bastante bem quando os recursos de espaço do robô carregam tudo o que a pessoa possivelmente quer saber ou deseja. Mas e se esse não for o caso, como geralmente não o é no mundo real?

Por exemplo, no caso de querermos que o robô se distancie de nós, eles não têm um recurso para isso. Podes empurrá-lo para longe de ti mas, no máximo, o robô reconhece que não tem as características necessárias para interpretar o ajuste humano.

Este paper serve então para perceber como é que se pode ensinar corretamente um robô acerca das recompensas quando os seus recursos no espaço não são suficientes para capturar o que queremos?

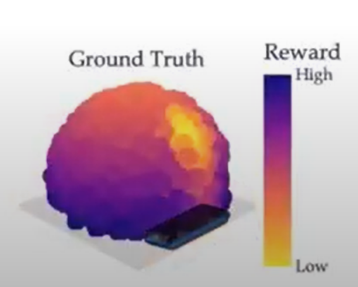
Para responder a esta dúvida, começou-se por olhar a um exemplo onde a pessoa importa-se com a eficiência e em manter a caneca próxima da mesa, porém distante do computador:

Uma imagem com caso

Descrição gerada automaticamente

Fig.1

Espacialmente a recompensa humana vai ser do tipo:



- A recompensa mais baixa está a amarelo e está exatamente acima do computador e bem acima da mesa e torna-se melhor quanto mais próxima da mesa e mais distante do computador

Fig.2

Entretanto, o robô sabe alguns dos recursos, mas não todos eles. Neste caso, ele perde a característica sobre a distância ao computador. Queremos, portanto, que o robô aprenda esta característica que não possui e como esta se combina com os restantes recursos para que a recompensa se conclua.

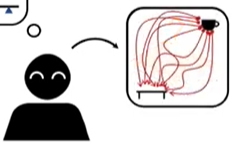
Uma imagem com texto, ClipArt

Descrição gerada automaticamenteAvanços recentes no IRL(deep Inverse Reinforcement Learning) sugerem que podemos fazer ambos os passos num só usando aprendizado profundo. Primeiramente, modela-se o recurso em falta como um mapeamento de rede neural diretamente do estado bruto de alta dimensão do robô, e adicionar às definições de recursos conhecidos. Daí, coleta-se algumas demonstrações da tarefa através do humano e usa-se ambos para extraírem o que falta implicitamente e aprender um estado de recompensa.

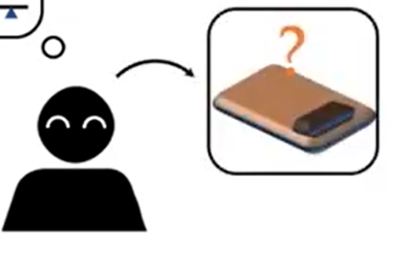
Fig.3

A recompensa final é então dada pela soma das recompensas do estado treinado. Agora que seguimos este processo vemos que o método de entropia máxima da irl faz um trabalho razoável identificando aproximadamente onde o computador está mas não consegue realmente aprender detalhes mais refinados da função de recompensa. Então o que está a acontecer aqui? Afinal, o que está errado?

Bem, as demonstrações que vimos na fig.3 são usadas com o objetivo de ajudar o robô a realizar a tarefa, não são feitas para aprendizagem de recursos assim dizendo. Portanto, com esta nova estrutura de rede neural, o método está meio que na esperança de aprender as características perdidas implicitamente pelas demonstrações.

Mas como tenta tanto tirar recursos salientes(importantes) e aprender uma recompensa por cima, IRL generaliza pobremente por fora das demonstrações espertas. E se ao invés de tentar aprender tudo de uma só vez, focássemos em aprender os recursos perguntando ao humano por alguma introdução(input) por parte deste e apenas se aprendia a recompensa por cima disso usando tarefas de entrada como demonstrações ou correções que se falaram?

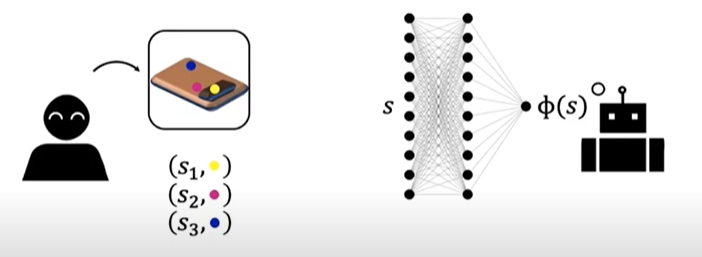
Uma imagem com texto

Descrição gerada automaticamente

Chave de Entendimento:

**Ao invés de aprender sobre os recursos em falta implicitamente, o robô devia perguntar pelos dados que explicitamente ensinam o que está a fazer falta.**

Então, como é que podemos ensinar um recurso?

A primeira ideia natural a surgir seria ainda modelar os recursos da rede neural mas depois perguntar às pessoas por rótulos, diretamente, que representem o valor do recurso em certos estados do ambiente.

Afinal de tudo, o que é mais explícito do que supervisionar diretamente esse recurso?

Depois de coletar dados definidos de rótulos de regressão, podíamos usá-lo para treinar a rede neural segundo uma aprendizagem supervisionada. Infelizmente, para aprender qualquer coisa de útil, o robô iria precisar de um número absurdo de rótulos definidos pela pessoa, o que seria demasiado pesado. Pior ainda, humanos conseguem ser justiçados imprecisamente quando rotulam os estados diretamente. Contudo, acontece que conseguimos fazer muito melhor além disto. Ao invés de termos os estados dos rótulos das pessoas de forma independente, nós podemos juntar um monte de informação se perguntar-mos à pessoa se a sequência de estados com valores de recurso são monotipicamente diminuindo, então chamamos a sequência com estas propriedades um traço de recurso

Robots have features that allow them to understand human desire, such as raising or lowering one of its elements to bring the mug to a level close to people's use.

however, robots have no notion of distance, which is an important feature to take into account in the real world