University of BRISTOL

Machine Learning

Reinforcement Learning and Decisions

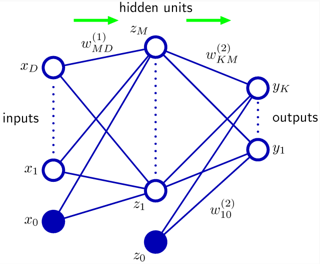
Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

November 27, 2018

http://www.carlhenrik.com

Introduction

Neural Network



$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j}^{M} w_{kj}^{(2)} h\left(\sum_{j}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

Composite Functions

Why are composite functions attractive?

$$\mathbf{y} = g(\mathbf{x}) = f_K(f_{K-1}(f_{K-2}(\dots f_1(\mathbf{x})\dots)))$$

Linear Algebra

- Kernel of a function

$$\text{Kern}(f_k) = \left\{(\mathsf{x}, \mathsf{x}') | f_k(\mathsf{x}) = f_k(\mathsf{x}')\right\}$$

Linear Algebra

- Kernel of a function

$$\text{Kern}(f_k) = \left\{(\mathbf{x}, \mathbf{x}') | f_k(\mathbf{x}) = f_k(\mathbf{x}')\right\}$$

- Image of a function

$$\text{Im}(f_k(\mathbf{x})) = \{\mathbf{y} \in Y | \mathbf{y} = f_k(\mathbf{x}), \mathbf{x} \in X\}$$

Linear Algebra

- Rank-Nullity Theorem

$$\dim(\mathrm{Im}(f)) = \dim(V) - \dim(\mathrm{Kern}(f))$$

Linear Algebra

- Rank-Nullity Theorem

$$\dim(\mathrm{Im}(f)) = \dim(V) - \dim(\mathrm{Kern}(f))$$

- Kernel of function

$$\mathrm{Kern}(f_1) \subseteq \mathrm{Kern}(f_{k-1} \circ \ldots \circ f_2 \circ f_1) \subseteq \mathrm{Kern}(f_k \circ f_{k-1} \circ \ldots \circ f_2 \circ f_1)$$

Linear Algebra

- Rank-Nullity Theorem

$$\dim(\mathrm{Im}(f)) = \dim(V) - \dim(\mathrm{Kern}(f))$$

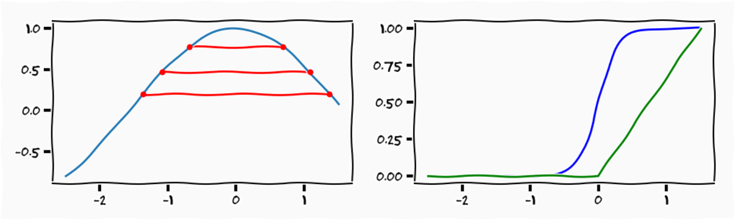
- Kernel of function

$$\mathrm{Kern}(f_1) \subseteq \mathrm{Kern}(f_{k-1} \circ \ldots \circ f_2 \circ f_1) \subseteq \mathrm{Kern}(f_k \circ f_{k-1} \circ \ldots \circ f_2 \circ f_1)$$

- Image of a function

$$\mathrm{Im}(f_k \circ f_{k-1} \circ \ldots \circ f_2 \circ f_1) \subseteq \mathrm{Im}(f_k \circ f_{k-1} \circ \ldots \circ f_2) \subseteq \ldots \subseteq \mathrm{Im}(f_k)$$

Composition functions

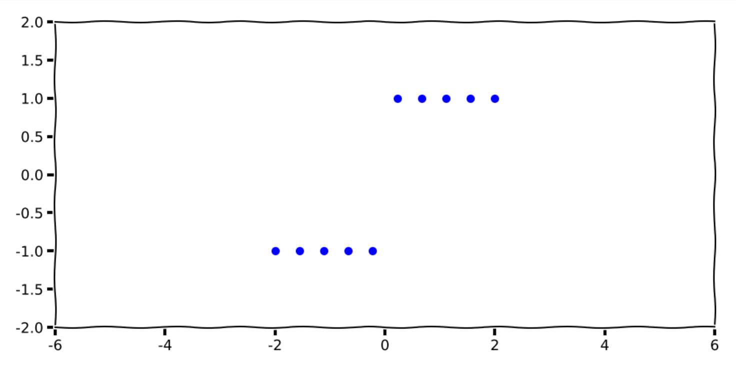


$$y = f_k(f_{k-1}(\dots f_0(x))) = f_k \circ f_{k-1} \circ \dots \circ f_1(x)$$

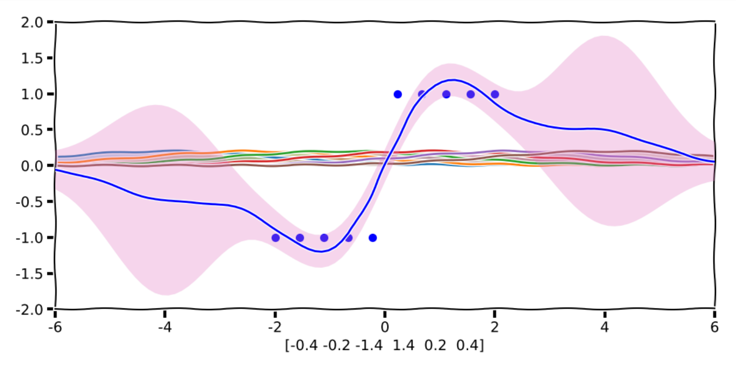
$$\mathrm{Kern}(f_1) \subseteq \mathrm{Kern}(f_{k-1} \circ \dots \circ f_2 \circ f_1) \subseteq \mathrm{Kern}(f_k \circ f_{k-1} \circ \dots \circ f_2 \circ f_1)$$

$$\mathrm{Im}(f_k \circ f_{k-1} \circ \dots \circ f_2 \circ f_1) \subseteq \mathrm{Im}(f_k \circ f_{k-1} \circ \dots \circ f_2) \subseteq \dots \subseteq \mathrm{Im}(f_k)$$

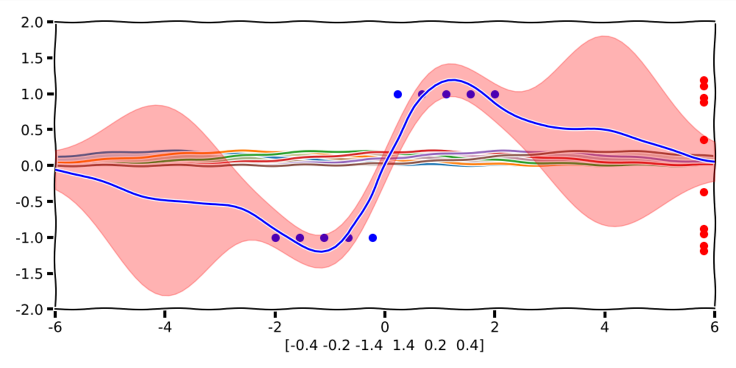
Composite Functions



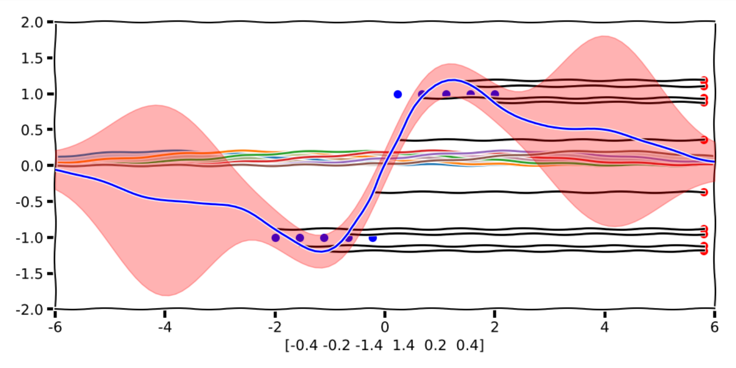
Composite Functions



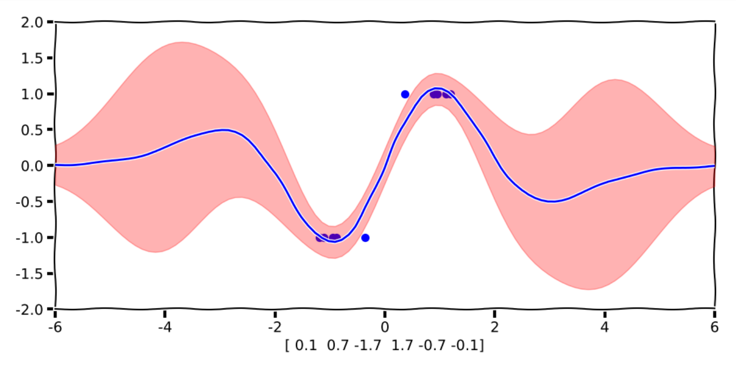
Composite Functions



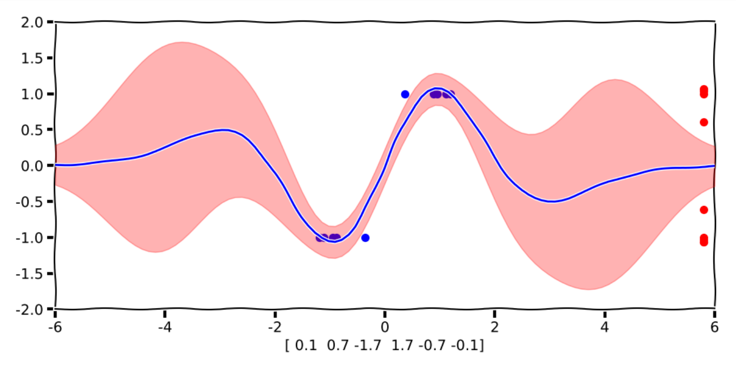
Composite Functions



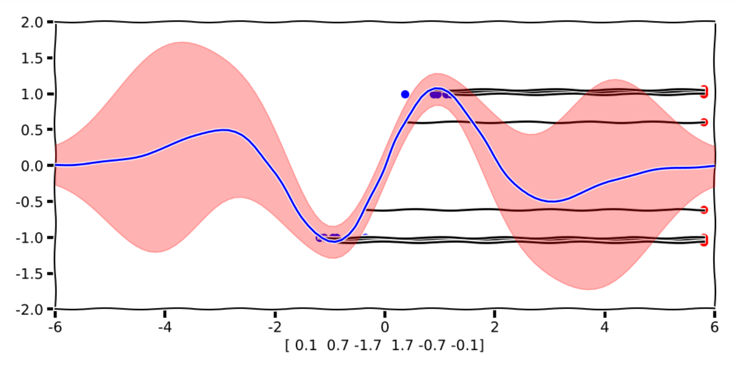
Composite Functions



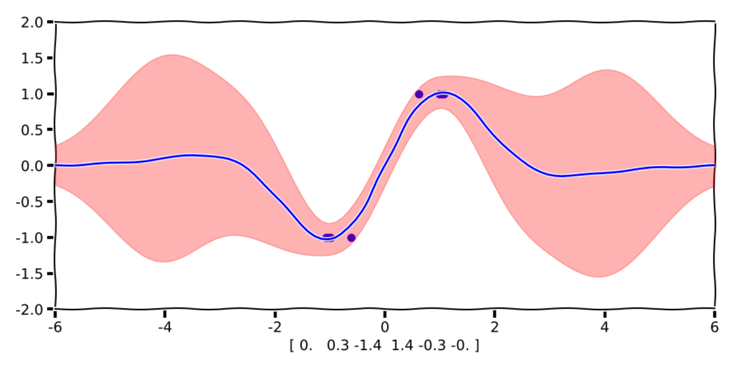
Composite Functions



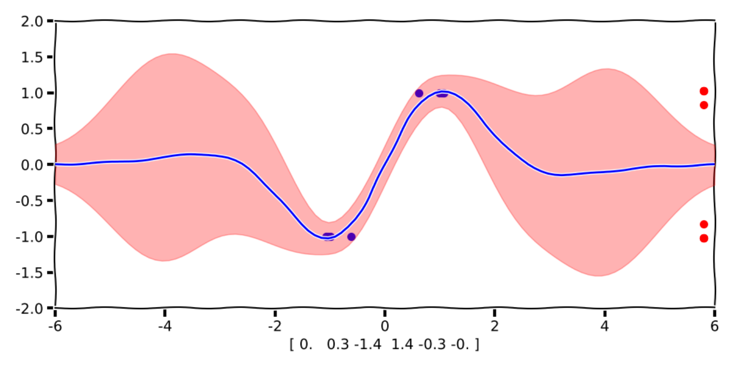
Composite Functions



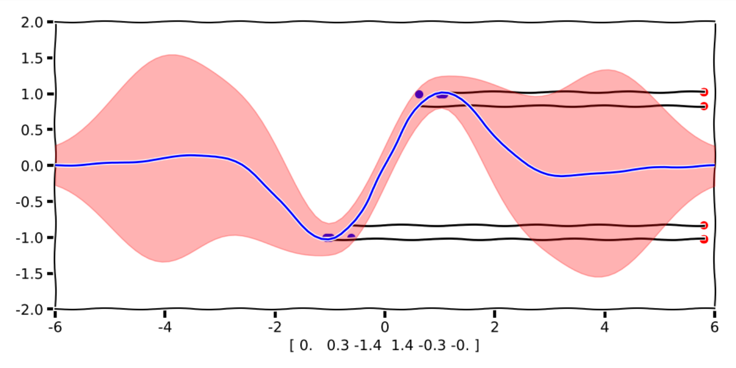
Composite Functions



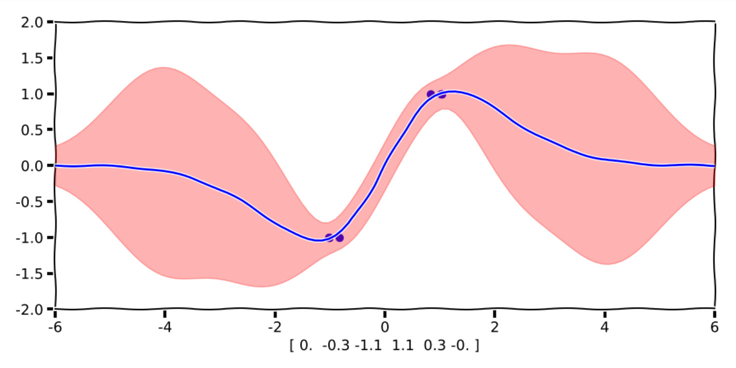
Composite Functions



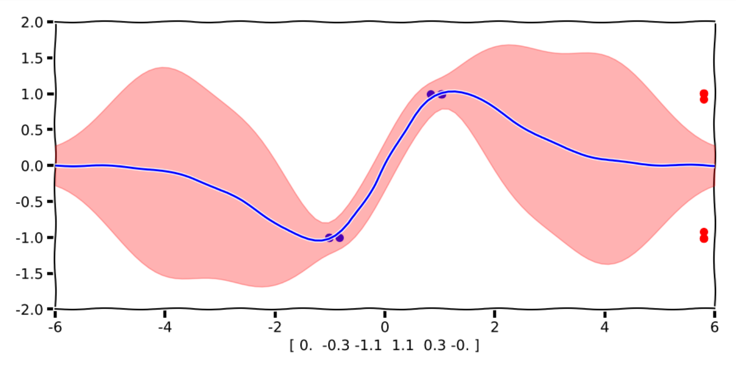
Composite Functions



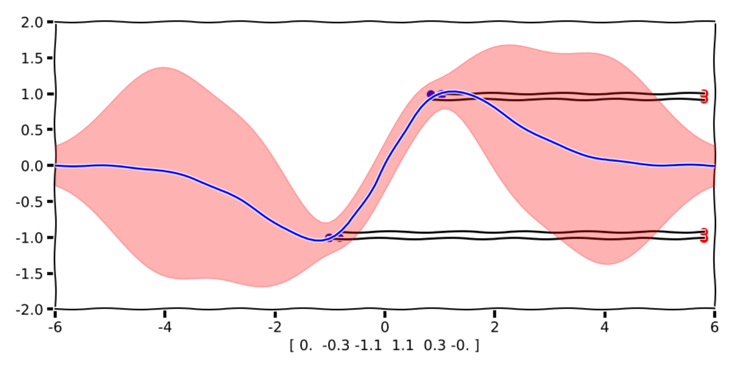
Composite Functions



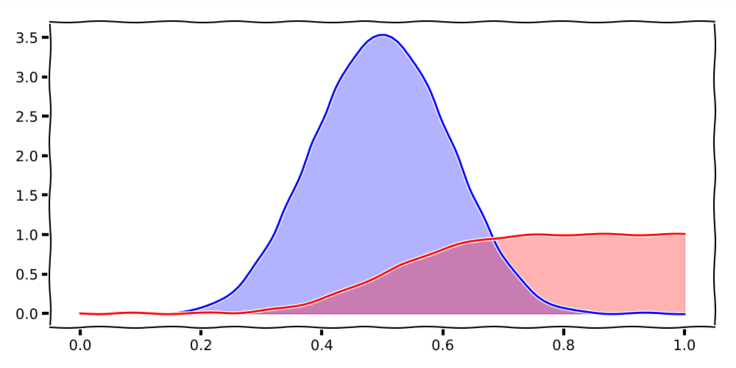
Composite Functions



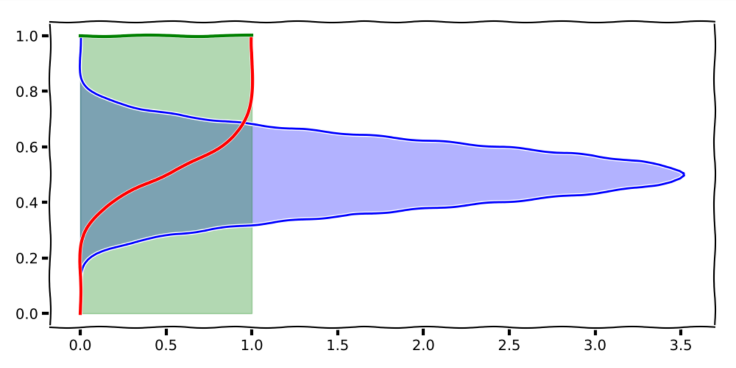
Composite Functions



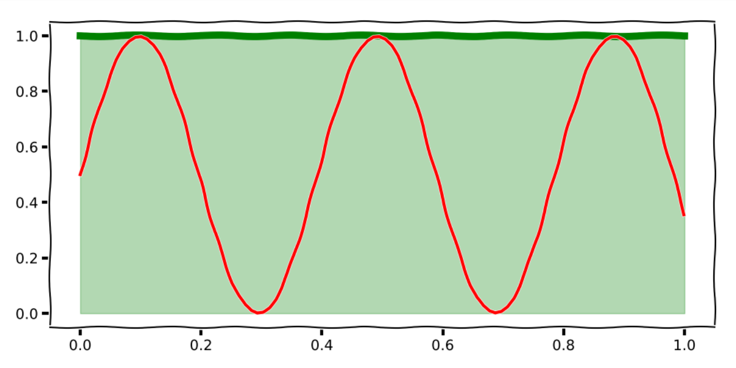
Sampling



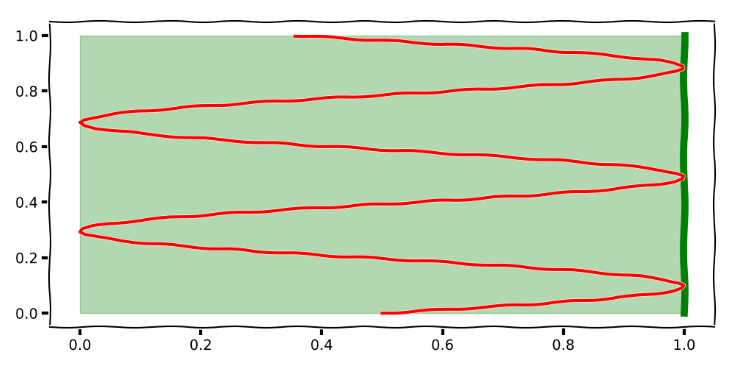
Sampling



Change of Variables



Change of Variables



Reinforcement Learning

Machine Learning

Supervised Learning predict output from input

$$\mathcal{D} = \{y_i, x_i\}_{i=1}^{N}$$

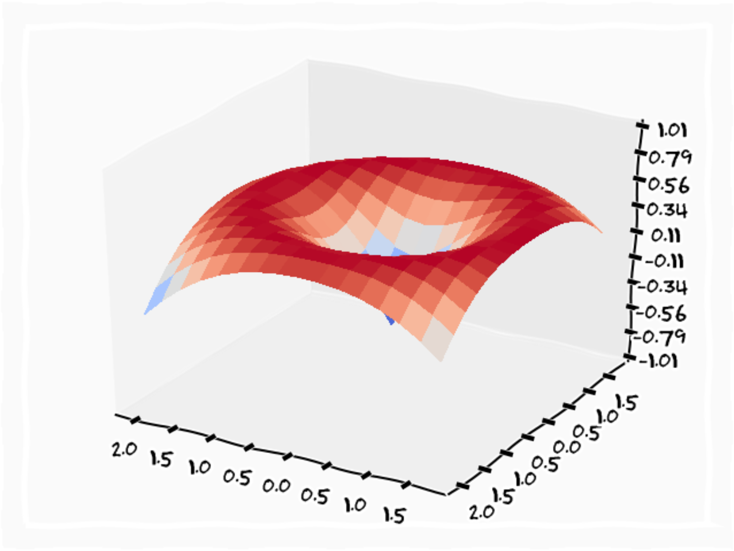
$$p(\mathbf{y}|\mathbf{x}, \theta)$$

Unsupervised Learning model the data

$$\mathcal{D} = \{y_i\}_{i=1}^{N}$$

$$p(\mathbf{y}|\theta)$$

Structure



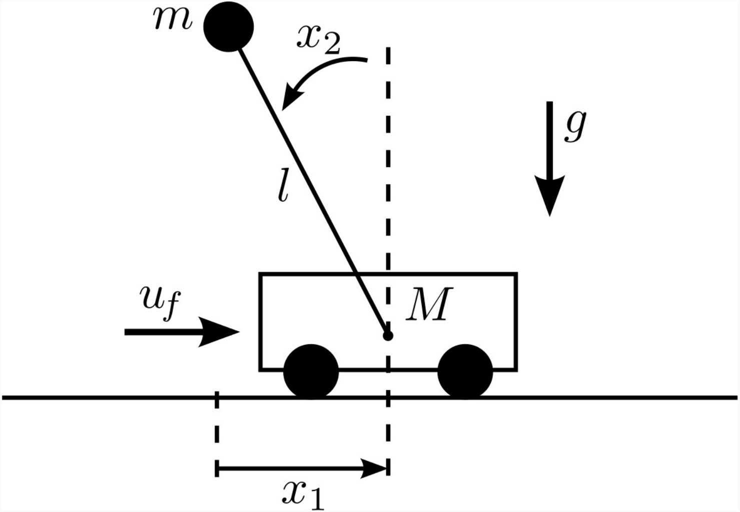
1.01
0.79
0.56
0.34
0.11
-0.11
-0.34
-0.56
-0.79
-1.01
2.0
1.5
1.0
0.5
0.0
0.5
1.0
1.5
2.0
1.5
1.0
0.5
0.0
0.5
1.0
1.5

Reinforcement

Reinforcement Learning

Can we learn without specifying how the task should be achieved by providing, rewards (positive) and punishment (negative)?

Inverted Pendlum[1]

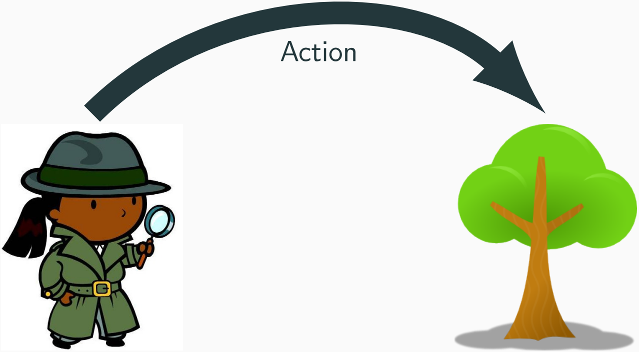


[1] https://www.youtube.com/watch?v=XiigTGKZfks

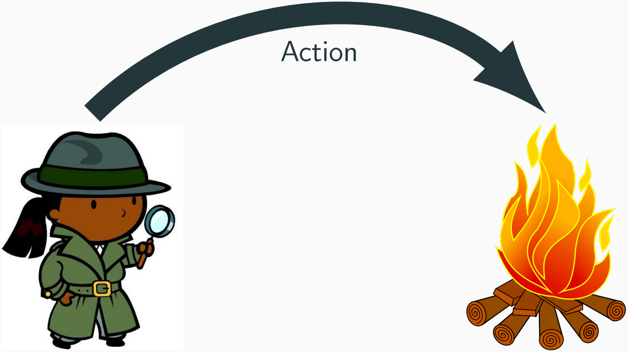
Formalism

Formalism

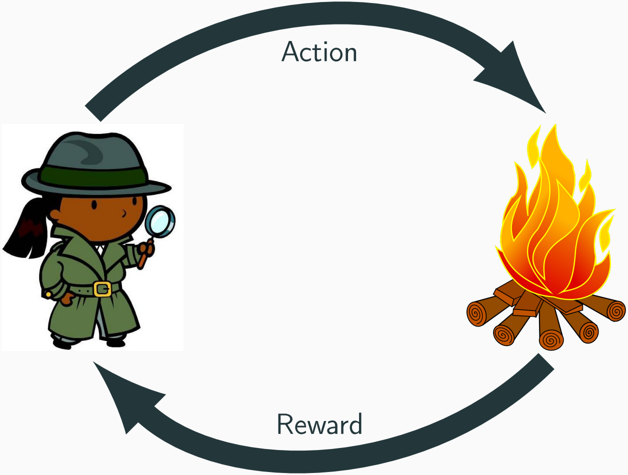
Formalism



Formalism



Formalism



Formalism [1]

$\mathcal{S}$ a discrete set of environment states

$\mathcal{A}$ a discrete set of agent actions

r a scalar set of of reinforcement signals, real line, or $\{0, 1\}$

I input function, how the agent views the state of the environment

π policy, mapping from state to action that maximises a long-run measurment of reinforcement

Setting

- The world is non-determinstic
  - we can be in the same state and do the same action and different things happens
- The world is stationary
  - the probabilities of the state transitions do not change
- Input function
  - if the agent can see the state of the world we call this fully observable
  - if the agent can only see part of the state, its partially observable

Example

Environment you are in state 65 you have 4 possible actions

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 7 units, you are now in state 15 you have 2 possible actions

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 7 units, you are now in state 15 you have 2 possible actions

Agent I take action 1

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 7 units, you are now in state 15 you have 2 possible actions

Agent I take action 1

Environment you recieved reinforcement of -4 units, you are now in state 65 you have 4 possible actions

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 7 units, you are now in state 15 you have 2 possible actions

Agent I take action 1

Environment you recieved reinforcement of -4 units, you are now in state 65 you have 4 possible actions

Agent I take action 2

Example

Environment you are in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 7 units, you are now in state 15 you have 2 possible actions

Agent I take action 1

Environment you recieved reinforcement of -4 units, you are now in state 65 you have 4 possible actions

Agent I take action 2

Environment you recieved reinforcement of 5 units, you are now in state 44 you have 5 possible actions

Optimal Behaviour

- Finite time horizon

$$E\left[\sum_{t=0}^{h} r_t\right]$$

Optimal Behaviour

- Finite time horizon

$$E\left[\sum_{t=0}^{h} r_t\right]$$

- Average reward

$$\lim_{h\to\infty} E\left[\frac{1}{h}\sum_{t=0}^{h} r_t\right]$$

Optimal Behaviour

- Finite time horizon

$$E\left[\sum_{t=0}^{h} r_t\right]$$

- Average reward

$$\lim_{h\to\infty} E\left[\frac{1}{h}\sum_{t=0}^{h} r_t\right]$$

- Infinite horizon (discounted reward)

$$E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

$$0 \leq \gamma \leq 1$$

Markov Decision Process

Fully observable system

- Transition matrix

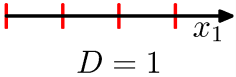
$$T(s, a, s') = p(s(t+1) = s' | s(t) = s, a(t) = a)$$

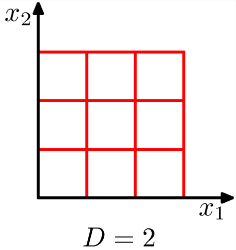
- Reward matrix

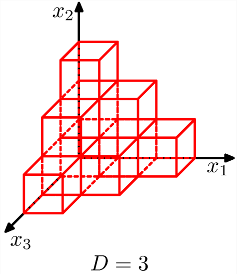
$$R(s, a, s') = E(s(t) = s, a(t) = a, s(t+1) = s')$$

- For this set-up the optimal policy can be computed using Dynamic Programming

Markov Decision Process





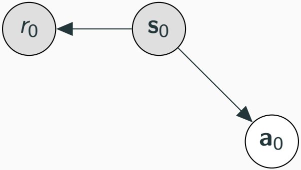


We cannot enumerate all the states and actions

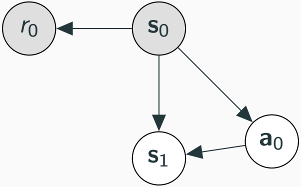
Reinforcement Learning

Reinforcement Learning

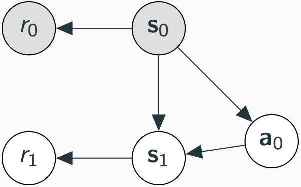
Reinforcement Learning



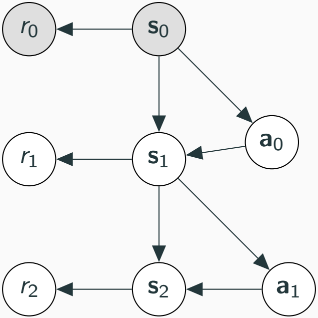
Reinforcement Learning

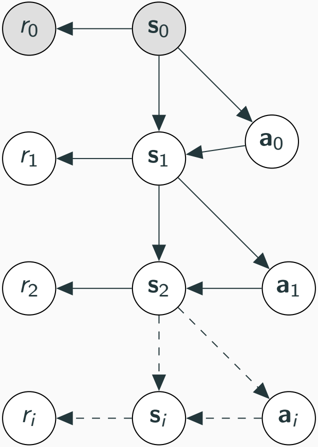


Reinforcement Learning

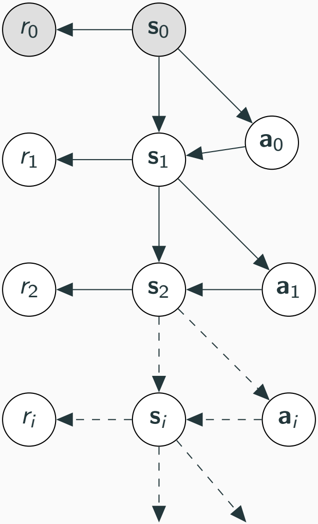


Reinforcement Learning

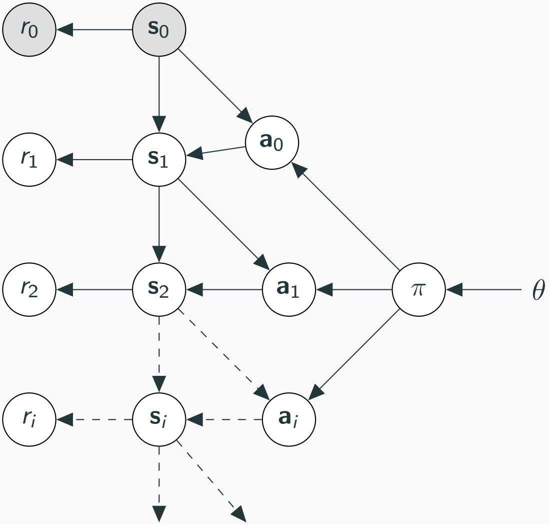



r_0
$\mathbf{s}_0$
r_1
$\mathbf{s}_1$
$\mathbf{a}_0$
r_2
$\mathbf{s}_2$
$\mathbf{a}_1$
r_i
$\mathbf{s}_i$
$\mathbf{a}_i$

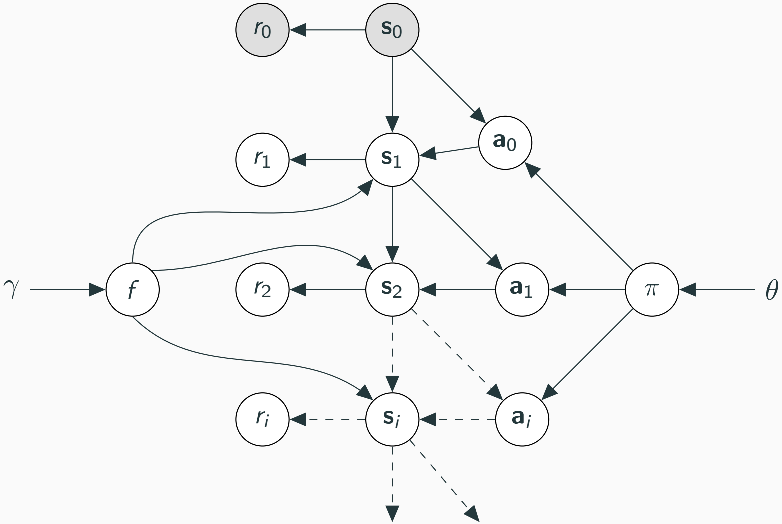
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

- Dynamic model

$$p(\mathbf{s}_t|\mathbf{a}_{t-1}, \mathbf{s}_{t-1}, f)$$

- Policy

$$p(\mathbf{a}_t|\mathbf{s}_t, \pi)$$

- Reward

$$p(r_t|\mathbf{s}_t)$$

$$p(\mathbf{y}_t|\mathbf{s}_t)$$

- We might not be able to observe the state

Reinforcment Learning

$$
\begin{aligned}
p(\mathbf{s}_{0,\ldots,T}, \mathbf{a}_{0,\ldots,T-1}, r_{0,\ldots,T}, f, \pi, \gamma, \theta | \mathbf{s}_0) = \\
& p(\mathbf{s}_T | a_{T-1}, s_{T-1}, f) p(a_{T-1} | s_{T-1}, \pi) \ldots \\
& \qquad \vdots \\
& p(\mathbf{s}_2 | a_1, s_1, f) p(a_1 | s_1, \pi) \\
& p(\mathbf{s}_1 | a_0, s_0, f) p(a_0 | s_0, \pi) \\
& p(f | \gamma) p(\pi | \theta) p(\gamma) p(\theta)
\end{aligned}
$$

- if we want to learn dynamics and policy we need to marginalise out f, π, θ and γ
- very very hard problem as uncertainty propagated a long way

Model Free

$$Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

- Learning a model is very hard due to uncertainty propagation
- Directly learn a function from state action sets to reward

- We can use any availible machine learning method to learn these functions
- Gaussian processes
- Linear regression
- Composite functions (neural networks)
- The tricky thing is how to get the data?

Bandits

Bandit problems

- You are in a room with k slot machines
- Each have a different (unknown) probability of pay-off
- You are permitted h different executions
- Whats the optimal strategy?

Exploration vs. Exploration

- When thinking of a strategy to come up with a policy we need to balance
  - exploring the environment
  - exploiting what we know

Exploration vs. Exploration

- When thinking of a strategy to come up with a policy we need to balance
    - exploring the environment
    - exploiting what we know
- Re-inforcement learning can be thought of in terms of bandit problems
    - sequential bandits
    - delayed reward
    - betting in poker (I think)

Exploration vs. Exploration

- When thinking of a strategy to come up with a policy we need to balance
  - exploring the environment
  - exploiting what we know
- Re-inforcement learning can be thought of in terms of bandit problems
  - sequential bandits
  - delayed reward
  - betting in poker (I think)
- Designing this strategy is the main challenge

Exploration vs. Exploration

- When thinking of a strategy to come up with a policy we need to balance
  - exploring the environment
  - exploiting what we know
- Re-inforcement learning can be thought of in terms of bandit problems
  - sequential bandits
  - delayed reward
  - betting in poker (I think)
- Designing this strategy is the main challenge
- Is Reinforcement Learning Bayesian Optimisation?

Deep Mind

https://www.youtube.com/watch?v=faDKMMwOS2Q

Models vs. Decisions

https://youtu.be/QHcAlAprFxA

Decisions

Models[2]

- So far we have just described how to model data
    1. make assumptions
    2. combine assumptions with data
    3. derive posterior
- Everything has been stochastic as we propagate uncertainty

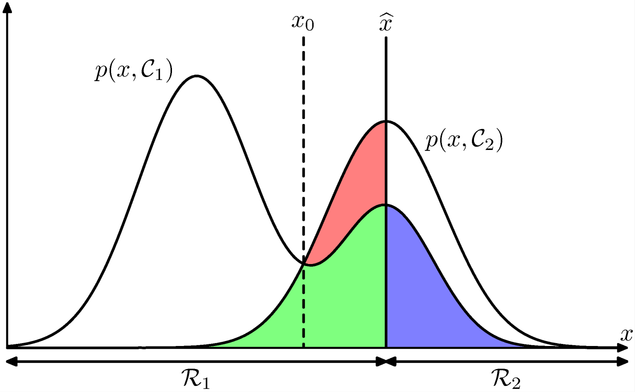
[2] http://inverseprobability.com/2017/11/15/decision-making

Models[2]

- So far we have just described how to model data
  1. make assumptions
  2. combine assumptions with data
  3. derive posterior
- Everything has been stochastic as we propagate uncertainty
- *you can't marginlise over menu items in a resturant*

[2] http://inverseprobability.com/2017/11/15/decision-making

Decisions



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, C_2) + p(\mathbf{x} \in \mathcal{R}_2, C_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, C_2) \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, C_1) \mathrm{d}\mathbf{x}
\end{aligned}
$$

Loss Functions

	Cancer	$\neg$ Cancer
Cancer	0	100
$\neg$ Cancer	1	0

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \mathrm{d}\mathbf{x}$$

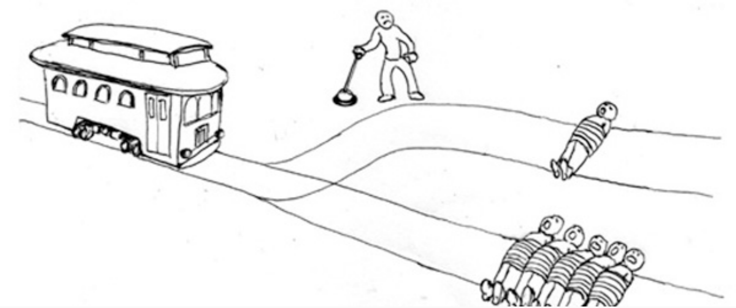
Utalitarian theory

Utilitarianism is an ethical theory which states that the best action is the one that maximizes utility. "Utility" is defined in various ways, usually in terms of the well-being of sentient entities. Jeremy Bentham, the founder of utilitarianism, described utility as the sum of all pleasure that results from an action, minus the suffering of anyone involved in the action.
– Wikipedia

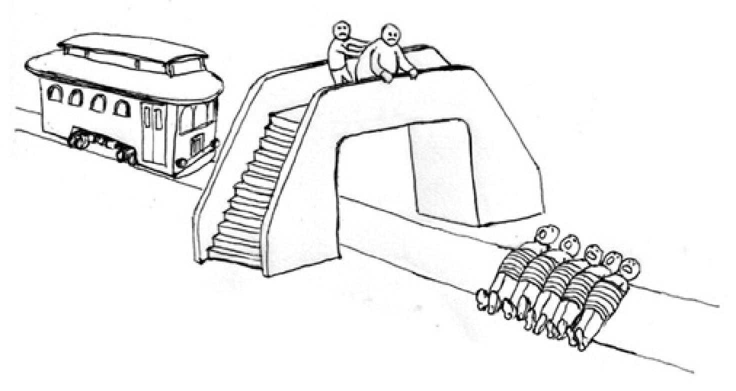
Utalitarian theory

- We can make formulate decisions as a mathematical principle
- Very attractie as it allows decisions to be made explicitly it allows for accountability
- Does it work?

Trolley



Trolley



Numbers

	Cancer (1%)	$\neg$ Cancer (99%)
Positive	80%	9.6%
Negative	20%	90.4%

- *What is the probability that you have cancer given a positive test?*

Numbers

$$p(\text{cancer=true}|\text{test=pos}) = \frac{p(\text{test=pos}|\text{cancer=true})p(\text{cancer=true})}{p(\text{test=pos})}$$

Numbers

$$p(\text{cancer}=\text{true}|\text{test}=\text{pos}) = \frac{p(\text{test}=\text{pos}|\text{cancer}=\text{true})p(\text{cancer}=\text{true})}{p(\text{test}=\text{pos})}$$

$$p(\text{test}=\text{pos}) = \int p(\text{test}=\text{pos}|\text{cancer})p(\text{cancer}) =$$

$$p(\text{test}=\text{pos}|\text{cancer}=\text{false})p(\text{cancer}=\text{false})+$$

$$p(\text{test}=\text{pos}|\text{cancer}=\text{true})p(\text{cancer}=\text{true})$$

Numbers

$$p(\text{cancer=true}|\text{test=pos}) = \frac{p(\text{test=pos}|\text{cancer=true})p(\text{cancer=true})}{p(\text{test=pos})}$$

$$p(\text{test=pos}) = \int p(\text{test=pos}|\text{cancer})p(\text{cancer}) =$$

$$p(\text{test=pos}|\text{cancer=false})p(\text{cancer=false})+$$

$$p(\text{test=pos}|\text{cancer=true})p(\text{cancer=true})$$

$$p(\text{cancer=true}|\text{test=pos}) = \frac{0.8 \cdot 0.01}{0.096 \cdot 0.99 + 0.8 \cdot 0.01} = 0.078$$

Numbers

$$p(\text{cancer=true}|\text{test=pos}) = \frac{p(\text{test=pos}|\text{cancer=true})p(\text{cancer=true})}{p(\text{test=pos})}$$

$$p(\text{test=pos}) = \int p(\text{test=pos}|\text{cancer})p(\text{cancer}) =$$

$$p(\text{test=pos}|\text{cancer=false})p(\text{cancer=false})+$$

$$p(\text{test=pos}|\text{cancer=true})p(\text{cancer=true})$$

$$p(\text{cancer=true}|\text{test=pos}) = \frac{0.8 \cdot 0.01}{0.096 \cdot 0.99 + 0.8 \cdot 0.01} = 0.078$$

- Only 15% of (medical) doctor answers this correctly

Self driving cars

Uncertainty [2]

- We might argue that there is such a thing as a single consistent utility function

Uncertainty [2]

- We might argue that there is such a thing as a single consistent utility function
  - Most likely we will then over simplify things

Uncertainty [2]

- We might argue that there is such a thing as a single consistent utility function
  - Most likely we will then over simplify things
- Different people have different utility functions

Uncertainty [2]

- We might argue that there is such a thing as a single consistent utility function
  - Most likely we will then over simplify things
- Different people have different utility functions
  - what makes me happy does not make Donald Trump happy

Uncertainty [2]

- We might argue that there is such a thing as a single consistent utility function
  - Most likely we will then over simplify things
- Different people have different utility functions
  - what makes me happy does not make Donald Trump happy
- We need to treat utility functions with uncertainty

Final note

$$p(\text{decision}) = \int p(\text{decision}|\text{utility function})p(\text{utility function})$$

- You know how to do this, you know the theory of this, you can move this up to any level

Summary

Summary

- Reinforcement learning
  - its nothing different at all, same ideas of modelling
  - how can we learn when we do not know how to do something
  - exploration exploitation
  - if we knew all states, all rewards, all actions it can be done exact

Summary

- Reinforcement learning
  - its nothing different at all, same ideas of modelling
  - how can we learn when we do not know how to do something
  - exploration exploitation
  - if we knew all states, all rewards, all actions it can be done exact
- Decisions
  - we need to make decisions
  - add additional information such as utility
  - one single utility functions is a too simplistic assumption
  - add uncertainty and marginalise

eof

References

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore.
Reinforcement learning: A survey.
J. Artif. Intell. Res., 4:237–285, 1996.