

Lecture 14: Generalisation and Multitask RL

Diana Borsa

5th March 2020, UCL

This Lecture

- ▶ Last lectures:
 - ▶ Deep RL (function approx, stabilising learning, dealing with non-iid data)
 - ▶ Concept of **learning about multiple things** (multiple predictions and/or control problems) as means to shape the representation.
 - ▶ **Value Improvement Path**
- ▶ This lecture:
 - ▶ Revisit **generalisation, beyond the single task setting**, towards skilful agents (continual/lifelong/multitask learning settings).
 - ▶ **Transfer and re-usability via GVF** (case study: **Successor Features**).
- ▶ Next lectures: More advanced (research) topics including: multi-agent systems, neuroscience, exploration, temporal abstraction.

Motivation

The world we live in requires us to learn many things. These things obey the same physical laws, derive from the same human culture, are preprocessed by the same sensory hardware. . . . Perhaps it is the similarity of the many tasks we learn that enables us to learn so much with so little experience.

Rich Caruana in his *PhD. Thesis*, 1997.[5]

Long-term Goal:

Agents that can learn to achieve **multiple objectives, compose and re-use knowledge** whenever appropriate and meaningfully interact with their environment.

(Quick Recap)

The reward hypothesis (Sutton and Barto 2018)

- ▶ All goals can be represented as **maximisation of a scalar reward**,
 - ▶ All useful knowledge may be encoded as predictions about rewards
 - ▶ For instance in the form of "general" value functions (GVFs),

General value functions (Sutton et al. 2011 [8])

- ▶ A GVF is conditioned on more than just state and actions

$$q_{c,\gamma,\pi}(s, a) = \mathbb{E}[C_{t+1} + \gamma_{t+1}C_{t+2} + \gamma_{t+1}\gamma_{t+2}C_{t+3} + \dots \mid S_t = s, A_{t+i} \sim \pi(S_{t+i})]$$

where $C_t = c(S_t)$ and $\gamma_t = \gamma(S_t)$ where S_t could be the environment state

- ▶ $c : \mathcal{S} \rightarrow \mathbb{R}$ is the **cumulant**
 - ▶ Predict many things, including — but not limited to — reward
- ▶ $\gamma : \mathcal{S} \rightarrow \mathbb{R}$ is the **discount** or termination
 - ▶ Predict for different time horizons γ
- ▶ $\pi : \mathcal{S} \rightarrow \mathcal{A}$ is the **target policy**
 - ▶ Predict under many different (hypothetical) policies π

Examples of GVFs

- ▶ Reward prediction:
 - ▶ $C_t = R_t$, $\gamma = 0$, under the agent's policy π
- ▶ Next state prediction:
 - ▶ $\{C_t^i = S_t^i\}_i$, $\gamma = 0$, under the agent's policy π
 - ▶ $\{C_t^i = \phi^i(S_t)\}_i$, $\gamma = 0$, under the agent's policy π
- ▶ Auxiliary tasks (Pixel/Feature Control)

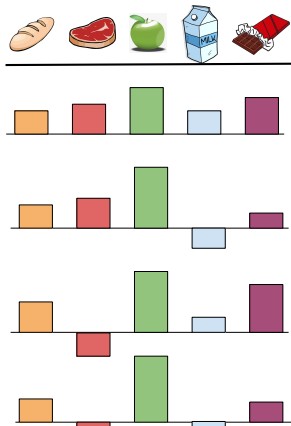
Today: GVFs' benefits beyond feature shaping, as primary **building blocks** of procedural knowledge.

Multi-Cumulant/Multi-Policy Learning in Persistent Environments

Motivating Example



Motivating Example



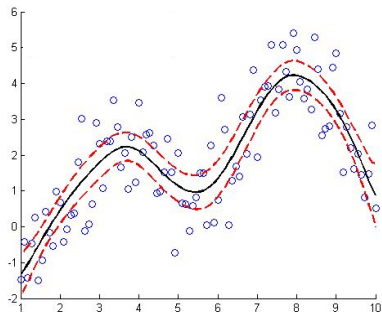
Formally: A family of MDPs

- ▶ Family of MDPs: $M = (\mathcal{S}, \mathcal{A}, p, \mathbf{r}, \gamma)$ that:
 - ▶ Shared dynamics: $(\mathcal{A}, \mathcal{S}, p)$ ("Persistent world")
 - ▶ Structured reward signal:

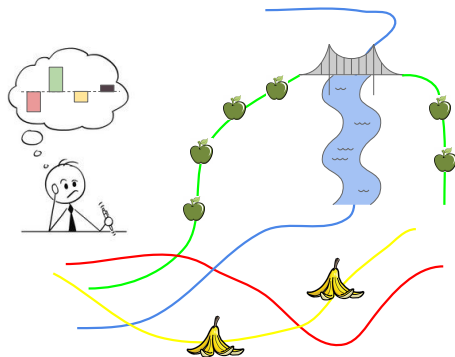
$$\mathbb{E}[R(\mathbf{s}, \mathbf{a}, \mathbf{s}')] = r_{\mathbf{w}} = \phi(\mathbf{s}, \mathbf{a}, \mathbf{s}')^T \mathbf{w}$$

- ▶ **Want:** Generalisation to a different task!
 - ▶ New task is (fully) specified by a different **preference vector \mathbf{w}'**
 - ▶ Given that I've already seen and solved some of these tasks, how can I use the **knowledge gained/built to inform learning** in the new task?

Generalisation in Multitask RL



Parametric Generalisation

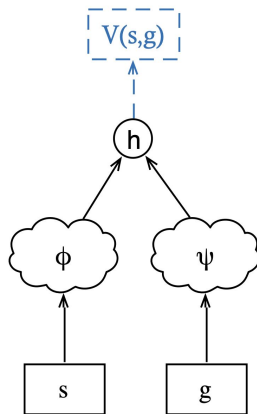
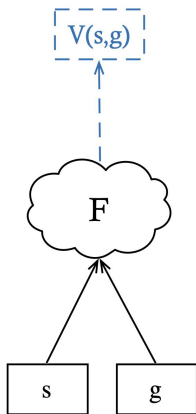


Behaviour Re-Evaluation

Parameteric Generalisation: UVFA [7]

- ▶ **Idea:** Treat the goal/tasks as an input to our approximation .
- ▶ **Aim:** To generalise over the goal/task space. Given a new task, be able to extrapolated to a policy/value function that is appropriate for the new task.
- ▶ Instances:
 - ▶ Goal/Task-conditioned value functions $V(s, g|\theta) \approx V_g^*(s)$
 - ▶ Goal/Task-conditioned policies: $\pi(a, s, g|\theta) \approx \pi_g^*$

Parameteric Generalisation: UVFA [7]

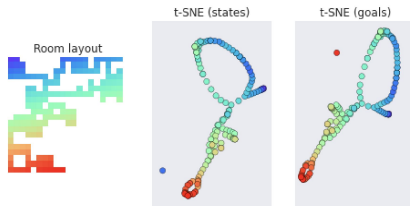


GVF considered:

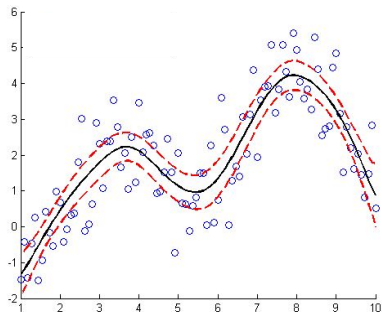
$$Q_{g,\pi}(s,a) := \mathbb{E}_{s'} [R_g(s,a,s') + \gamma_g(s') \cdot V_{g,\pi}(s')]$$

Generalisation across g :

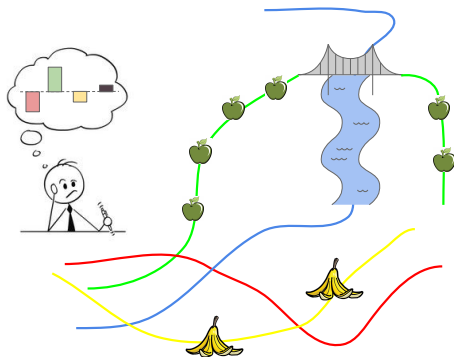
$$(\text{MSE}) \mathbb{E} \left[\left(V_g^*(s) - V(s,g;\theta) \right)^2 \right]$$



Generalisation in Multitask RL



Parametric Generalisation



Behaviour Re-Evaluation

Behaviour Re-Evaluation: General Principle

1. Collection of **multiple behaviours** (previously acquired):

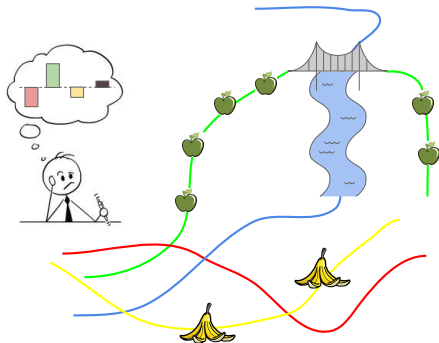
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$$

2. Evaluate behaviours on the **new task** (\mathbf{w}):

$$Q_{\mathbf{w}}^{\pi_i}(s, a), \forall i \leq n$$

3. Policy Improvement:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_{\mathbf{w}}^{\pi_i}(s, a)$$



$$\text{Reward: } r_{\mathbf{w}}(s, a, s') = \phi(s, a, s')^T \mathbf{w}$$

(Reminder) Policy Iteration

Policy Iteration

- ▶ Start with π_0 .
- ▶ Iterate:
 - ▶ Policy Evaluation: $q_k = q_{\pi_k}$
 - ▶ Greedy Improvement: $\pi_{k+1} = \arg \max_a q_{\pi_k}(s, a)$

As $k \rightarrow \infty$, $q_k \rightarrow_{\|\cdot\|_\infty} q^*$. Thus $\pi_k \rightarrow \pi^*$.

Generalising Policy Iteration

1. Collection of **multiple behaviours** :

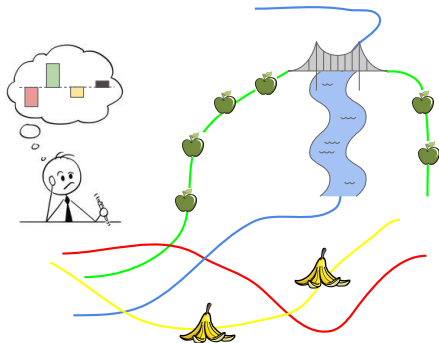
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$$

2. (GPE) **Generalised Policy Evaluation** for r_w :

$$Q_w^{\pi_i}(s, a), \forall i \leq n$$

3. (GPI) **Generalised Policy Improvement**:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_w^{\pi_i}(s, a)$$



$$\text{Reward: } r_w(s, a, s') = \phi(s, a, s')^T w$$

Generalised Policy Improvement (GPI), [2, 3]

Theorem (Generalised Policy Improvement (Barreto et al. 2017,[3]))

Consider a MDP and a set of policies $\Pi = \{\pi_i\}_{i=1,n}$. Let $q_{\pi_i} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ the action-value function evaluating policy π_i , for all $\pi_i \in \Pi$. Define the GPI policy as:

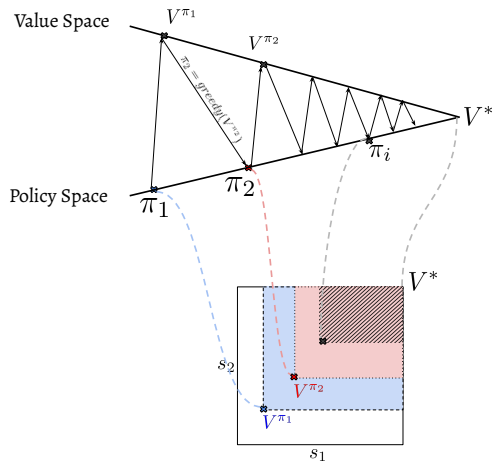
$$\pi_{GPI}(.|s) = \arg \max_a \max_i q_{\pi_i}(s, a)$$

*This policy is guaranteed to be **an improvement over all policies in Π** . That is:*

$$q_{\pi_{GPI}}(s, a) \geq q_{\pi_i}(s, a), \forall i \leq n, a \in \mathcal{A}, s \in \mathcal{S}$$

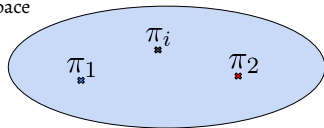
*Moreover, **equality is only satisfied when Π contains already the optimal policy.***

Policy Iteration: Depiction

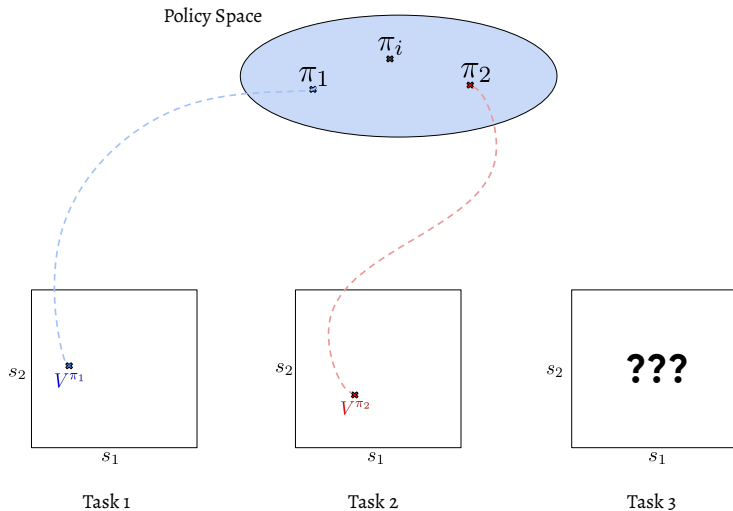


(Generalised) Policy Evaluation + Improvement: Depiction

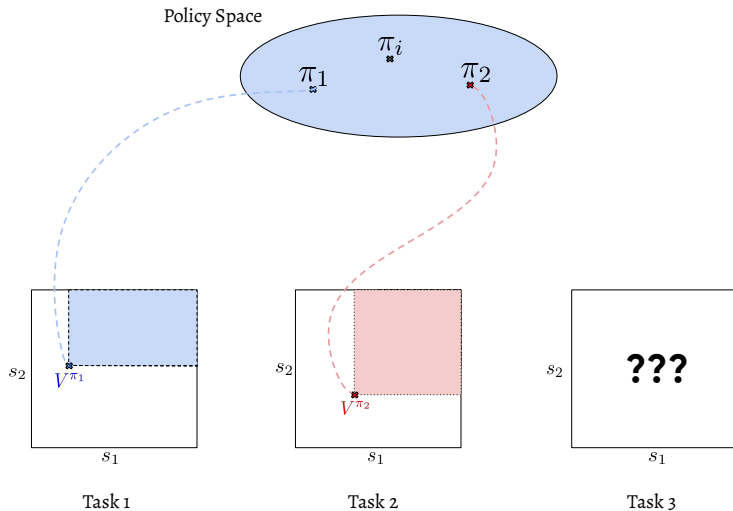
Policy Space



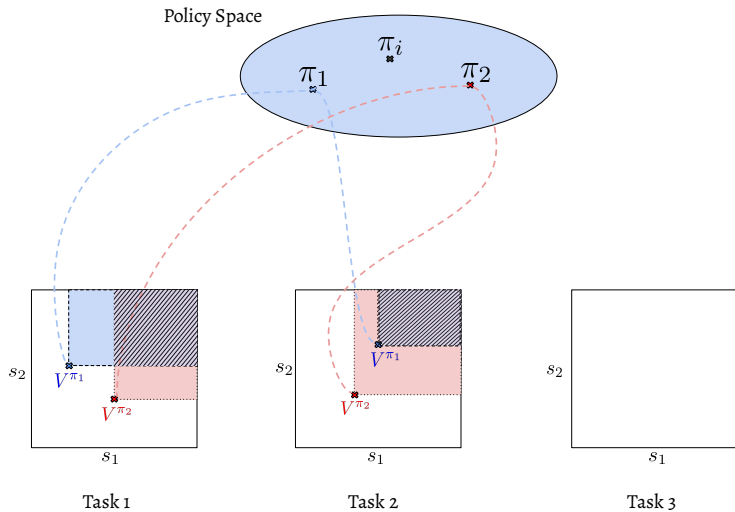
(Generalised) Policy Evaluation + Improvement: Depiction



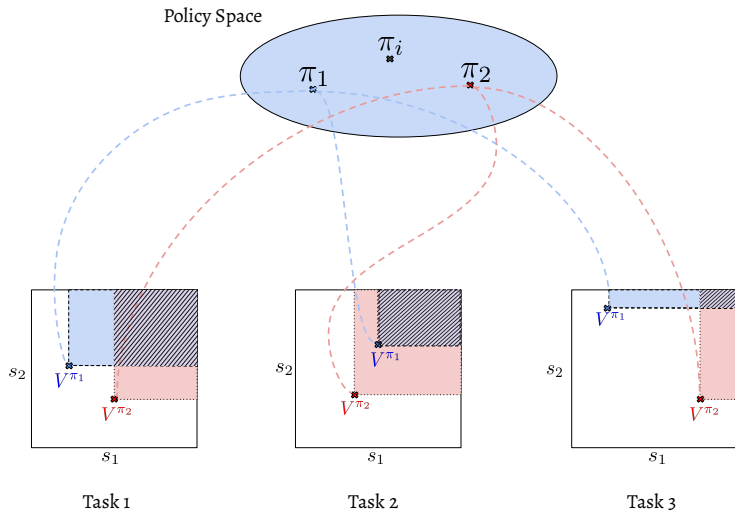
(Generalised) Policy Evaluation + Improvement: Depiction



(Generalised) Policy Evaluation + Improvement: Depiction



(Generalised) Policy Evaluation + Improvement: New task generalisation



Behaviour Re-Evaluation

1. Collection of **multiple behaviours** :

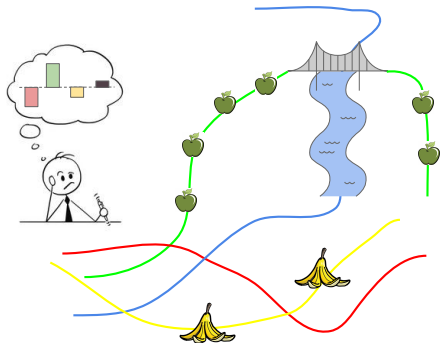
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$$

2. (GPE) **Generalised Policy Evaluation** for r_w :

$$Q_w^{\pi_i}(s, a), \forall i \leq n$$

3. (GPI) **Generalised Policy Improvement**:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_w^{\pi_i}(s, a)$$



$$\text{Reward: } r_w(s, a, s') = \phi(s, a, s')^T w$$

Behaviour Re-Evaluation: Generally Expensive :(

1. Collection of **multiple behaviours** :

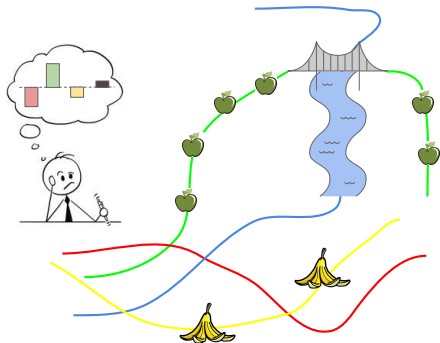
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$$

2. (GPE) **Generalised Policy Evaluation** for r_w :

$$Q_w^{\pi_i}(s, a), \forall i \leq n$$

3. (GPI) **Generalised Policy Improvement**:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_w^{\pi_i}(s, a)$$



$$\text{Reward: } r_w(s, a, s') = \phi(s, a, s')^T w$$

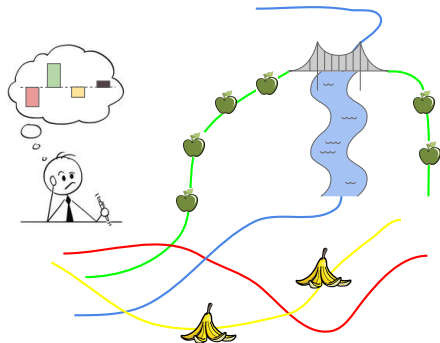
Behaviour Re-Evaluation via Successor Features [3, 6]

- Successor Features (a type of GVF):

$$\psi^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1} | S_t = s, A_t = a \right]$$

- Consider a task $r_{\mathbf{w}}$ in the span of ϕ .
- Instant policy evaluation for any \mathbf{w} :

$$Q_{\mathbf{w}}^\pi(s, a) = \psi^\pi(s, a)^T \mathbf{w}$$



$$\text{Reward: } r_{\mathbf{w}}(s, a, s') = \phi(s, a, s')^T \mathbf{w}$$

(Revised) Behaviour Re-Evaluation under SFs [2, 3]

1. Represent the **multiple behaviours** via SFs:

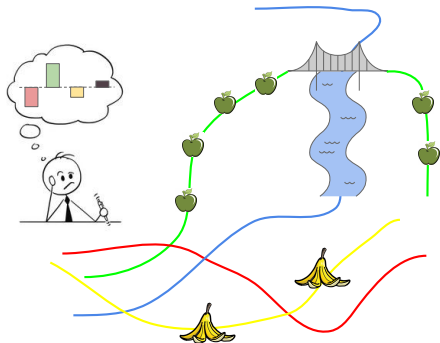
$$\{\psi^{\pi_1}, \psi^{\pi_2}, \dots, \psi^{\pi_n}\}$$

2. Instant **GPE** via recombination for any r_w :

$$Q_w^{\pi_i}(s, a) = \psi^{\pi_i}(s, a)^T \mathbf{w}, \forall i \leq n$$

3. Cheap Improvement: **GPI**

$$\pi(\cdot|s) = \arg \max_a \max_i Q_w^{\pi_i}(s, a)$$



$$\text{Reward: } r_w(s, a, s') = \phi(s, a, s')^T \mathbf{w}$$

What do we get? Safe and Sound transfer of behaviours!

- ▶ Transfer of knowledge to a **new task**:
 - ▶ Specify or **learn/infer** \mathbf{w} s.t. $r_{\mathbf{w}}(s, a) \approx r(s, a)$.
 - ▶ **Behaviour Re-evaluation** via SFs:

$$Q_{\mathbf{w}}^{\pi_i}(s, a) = \psi^{\pi_i}(s, a)^T \mathbf{w}$$

- ▶ Get a **improved policy** via GPI:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_{\mathbf{w}}^{\pi_i}(s, a)$$

(guaranteed to be best than on of the previous behaviours)

- ▶ Revisiting a (old) task:
 - ▶ **No forgetting**: We can continue learning every time we see this task again!
 - ▶ (Bonus) **Backward transfer**: if any of the policies we have learnt since then do well on our current task, we can readily use this knowledge.

Discussion

- ▶ Linearity assumption. How restrictive is this?

$$\phi(s, a, s') = [r(s, a, s'), c_1(s, a, s'), \dots, c_d(s, a, s')]$$

- ▶ Where do $\phi(s, a, s')$ come from? Can we learn them?
- ▶ Two scenarios:
 - ▶ **Scenario 1:** Continual learning/Curriculum setting: r_1, r_2, r_3, \dots .

$$\phi(s, a, s') = [r_1(s, a, s'), r_2(s, a, s'), \dots]$$

Note: by definition r_1, r_2, r_3 are in the span of ϕ .

New task: How does my new reward r_{new} relate (linearly) to previously seen rewards?

- ▶ **Scenario 2:** One big task (e.g. survival) that can be naturally broken down into simpler subtasks. Then ... more to do!

Main Take-away: Generalising Policy Iteration

1. Collection of **multiple behaviours** :

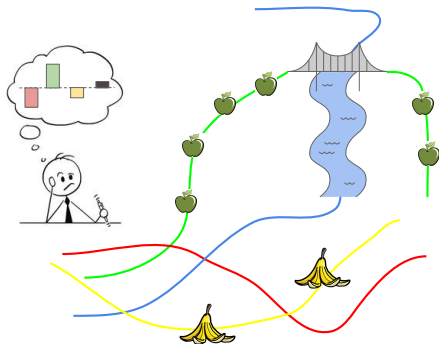
$$\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$$

2. (GPE) **Generalised Policy Evaluation** for $r_{\mathbf{w}}$:

$$Q_{\mathbf{w}}^{\pi_i}(s, a), \forall i \leq n$$

3. (GPI) **Generalised Policy Improvement**:

$$\pi(\cdot|s) = \arg \max_a \max_i Q_{\mathbf{w}}^{\pi_i}(s, a)$$

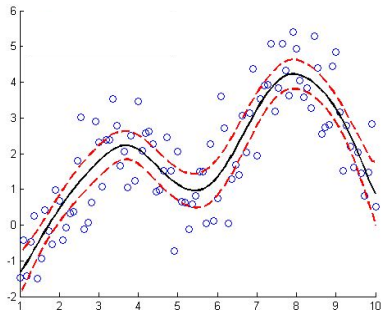


Efficient for rewards:

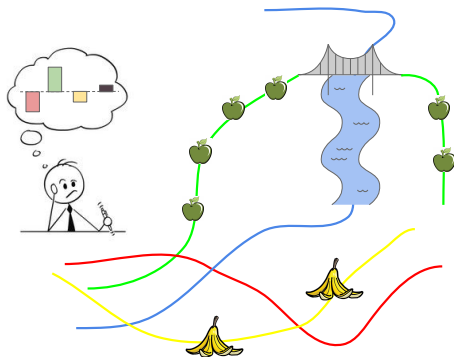
$$r_{\mathbf{w}}(s, a, s') = \phi(s, a, s')^T \mathbf{w}$$

Break

Best of both worlds: Universal Successor Features Approx. [4]



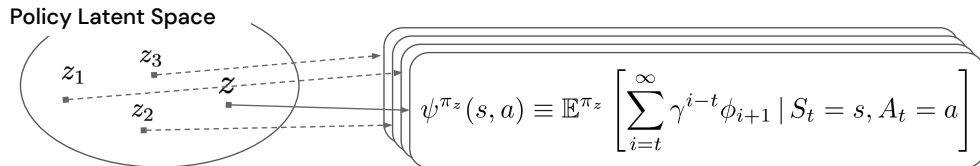
Parametric Generalisation



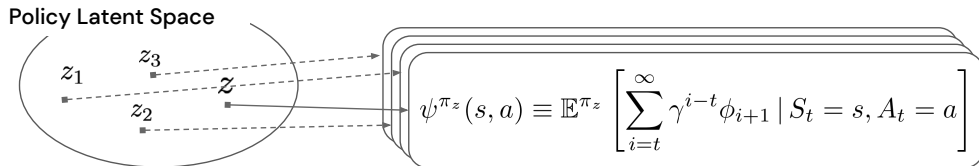
Behaviour Re-Evaluation

Non-examinable

Universal Successor Features



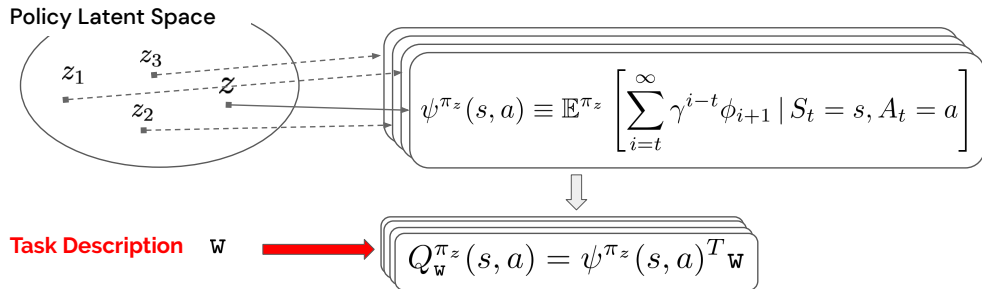
Universal Successor Features



Task Description \mathbf{W}

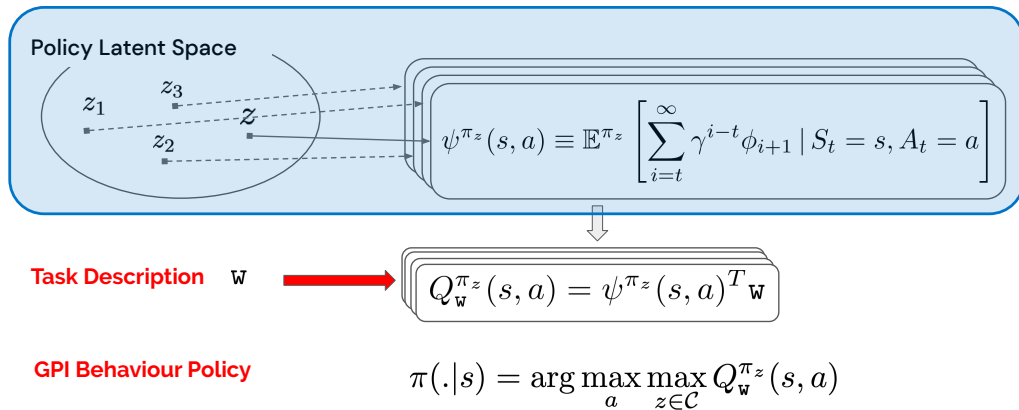
Non-examinable

Universal Successor Features



Non-examinable

Universal Successor Features Approximations [4]



USFAs: Learning

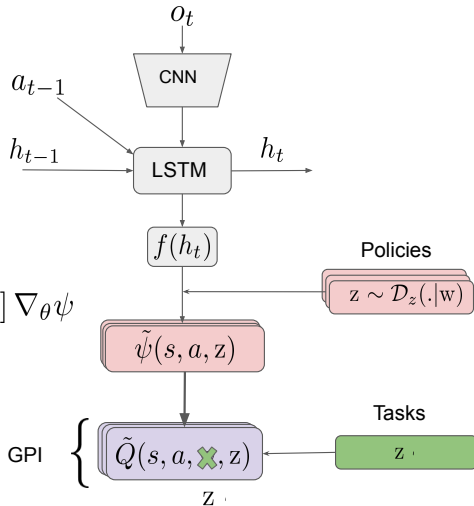
- Sample in **policy** space:

$$\mathbf{z} \sim \mathcal{D}_z(\cdot | \mathbf{w})$$

- Pretend you're in task $w=z$:

$$\theta \stackrel{\alpha}{\leftarrow} [\phi + \gamma\psi(s', \pi_{\mathbf{z}}, \mathbf{z}) - \psi(s, a, \mathbf{z})] \nabla_{\theta} \psi$$

where: $\pi_z = \arg \max_a Q_z^{\pi_z}$



Non-examinable

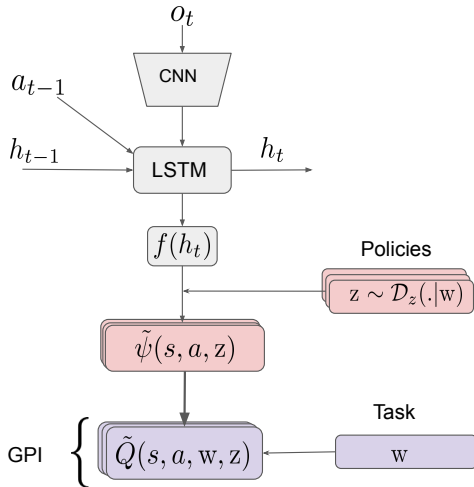
USFAs: Acting

- Sample in policy space:

$$z \sim \mathcal{D}_z(\cdot | w)$$

- Compute GPI policy:

$$\pi(s) = \arg \max_a \max_z \tilde{Q}(s, a, w, z)$$



Non-examinable

USFAs: Generalisation

Theorem (Universal Successor Feature Approximations (Borsa et al. 2019[4]))

Consider a MDP and a set of policies $\Pi = \{\pi_z | z \in \mathcal{C}\}$. Given approximations of the action-value functions $\{Q_w^{\pi_z} = \psi(s, a, z)^T w\}_{z \in \mathcal{C}}$ evaluating optimal policies π_z , for task r_z . And let π be the GPI policy on set \mathcal{C} , then:

$$\|Q_w^* - Q_w^\pi\|_\infty \leq \frac{2}{1-\gamma} \left[\min_{z \in \mathcal{C}}(\delta_d(z)) + \max_{z \in \mathcal{C}}(\|w\| \cdot \delta_\psi(z)) \right]$$

where $\delta_d(z) = \|\phi\|_\infty \|w - z\|$ and $\delta_\psi(z) = \|\psi^{\pi_z} - \psi(s, a, z)\|_\infty$.

USFAs: Generalisation

Theorem (Universal Successor Feature Approximations (Borsa et al. 2019))

Consider a MDP and a set of policies $\Pi = \{\pi_z | z \in \mathcal{C}\}$. Given approximations of the action-value functions $\{Q_w^{\pi_z} = \psi(s, a, z)^T w\}_{z \in \mathcal{C}}$ evaluating optimal policies π_z , for task r_z . And let π be the GPI policy on set \mathcal{C} , then:

$$\|Q_w^* - Q_w^\pi\|_\infty \leq \frac{2}{1-\gamma} \left[\min_{z \in \mathcal{C}}(\delta_d(z)) + \max_{z \in \mathcal{C}}(\|w\| \cdot \delta_\psi(z)) \right]$$

where $\delta_d(z) = \|\phi\|_\infty \|w - z\|$ and $\underbrace{\delta_\psi(z)}_{\text{Parameteric Approximation Error}} = \|\psi^{\pi_z} - \psi(s, a, z)\|_\infty$.

USFAs: Generalisation

Theorem (Universal Successor Feature Approximations (Borsa et al. 2019))

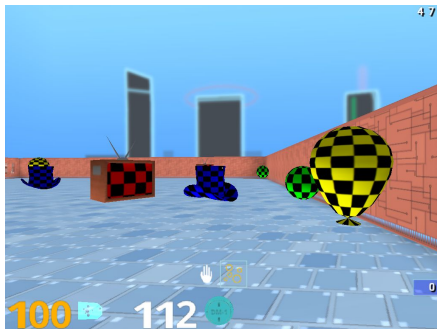
Consider a MDP and a set of policies $\Pi = \{\pi_z | z \in \mathcal{C}\}$. Given approximations of the action-value functions $\{Q_w^{\pi_z} = \psi(s, a, z)^T w\}_{z \in \mathcal{C}}$ evaluating optimal policies π_z , for task r_z . And let π be the GPI policy on set \mathcal{C} , then:





$$\|Q_w^* - Q_w^\pi\|_\infty \leq \frac{2}{1 - \gamma} \left[\min_{z \in \mathcal{C}}(\delta_d(z)) + \max_{z \in \mathcal{C}}(\|w\| \cdot \delta_\psi(z)) \right]$$

where $\underbrace{\delta_d(z) = \|\phi\|_\infty \|w - z\|}_{\text{GPI Zero-shot Performance}}$ and $\delta_\psi(z) = \|\psi^{\pi_z} - \psi(s, a, z)\|_\infty$.

USFAs: Some experimental results [4]

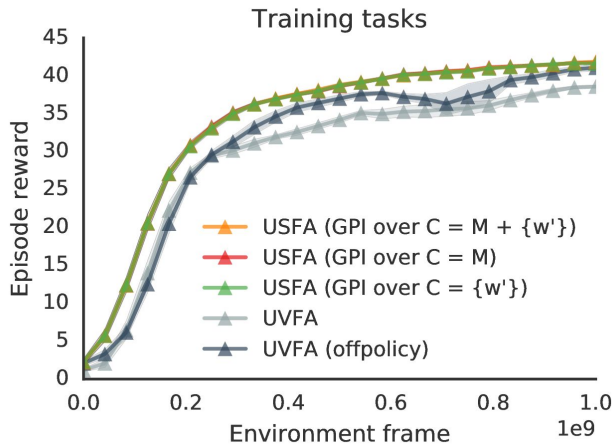
Environment Observation



				
Train tasks	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1
Test tasks	1	1	0	0
	0	1	-1	0
	-1	1	1	-1

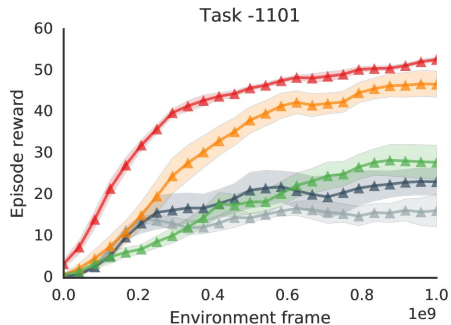
Non-examinable

USFAs: Some experimental results

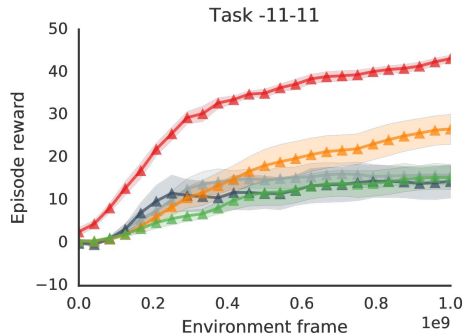


Non-examinable

USFAs: Some experimental results



- USFA (GPI over $C = M + \{w'\}$)
- USFA (GPI over $C = M$)
- USFA (GPI over $C = \{w'\}$)



- UVFA
- UVFA (offpolicy)

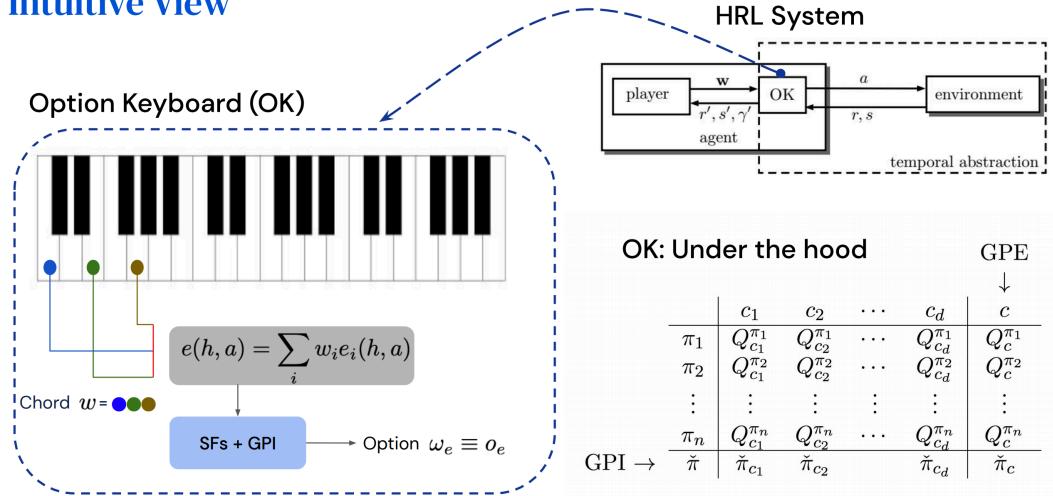
Non-examinable

Beyond zero-shot: Towards complex, composable behaviour....

- ▶ Preferences (and induced rewards) might change over time.
- ▶ Cannot be model by one w across time. Can we learn to change and track preferences/tasks over time?

Option Keyboard: Composing skills [1]

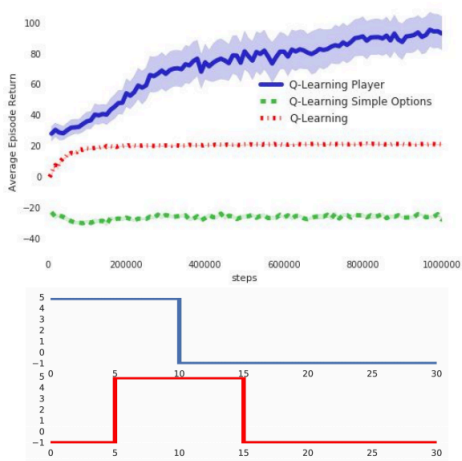
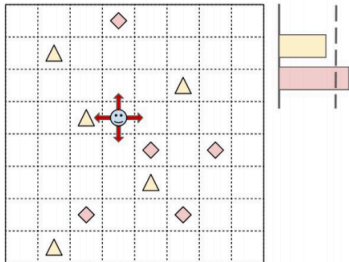
Intuitive View



Non-examinable

Option Keyboard: Composing skills

Task Visualisation



Non-examinable

Final Thoughts

Summary

- ▶ Two types of generalisation one can explore in RL
 1. Parametric FA (most ML)
 2. Behaviour Re-Evaluation (extending Policy Iteration)
- ▶ USFAs shows these are complementary and can be combined!
- ▶ SFs can be a very efficient way of building and transferring behaviours in a persistent environment to combinations of tasks
- ▶ Off-policy and Off-task training on (sampled) fictitious tasks might be a very powerful paradigm of building actionable knowledge (GVFs style)

Remaining Open Questions

- ▶ **Cumulant discovery**: What are the drives/rewarding events/salient features in the environment $\phi(s, a, s')$?
- ▶ **Behaviour basis** Π :
 - ▶ Which behaviours should we be **learning** about?
 - ▶ Which behaviours should be **trust for acting/planning**?
- ▶ Closing the loop on continual/**lifelong RL agents**.
 - ▶ Re-using these partial plans for explicit planning/learning.
 - ▶ Hypothesis-driven exploration.

Generalisation in RL

- ▶ **Warning:** Terms like '*generalisation*'/'*transfer*' are extremely overloaded.
- ▶ Transfer := ability **leverage knowledge** (policies/samples/repres./abstractions) gained from previous tasks **to improve learning** on the current task.
- ▶ Always specify the **dimension** along which one seeks **generalisation**. This can be across:
 - ▶ State-actions space.
 - ▶ **Policy-space.**
 - ▶ **Cumulant/reward space.**
 - ▶ Environments: e.g. transition dynamics change, reward stays the same.
 - ▶ Combination or all of the above!

Questions?

The only stupid question is the one you were afraid to ask and never did.
-Rich Sutton

For questions that arise outside of class, please use Moodle!

References I

- [1] Andre Barreto, Diana Borsa, Shaobo Hou, Gheorghe Comanici, Eser Aygün, Philippe Hamel, Daniel Toyama, Jonathan hunt, Shibl Mourad, David Silver, and Doina Precup. The option keyboard: Combining skills in reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13052–13062. Curran Associates, Inc., 2019.
- [2] André Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin vZídek, and Remi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- [3] André Barreto, Will Dabney, Rémi Munos, Jonathan Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

References II

- [4] Diana Borsa, Andre Barreto, John Quan, Daniel J. Mankowitz, Hado van Hasselt, Remi Munos, David Silver, and Tom Schaul. Universal successor features approximators. In *International Conference on Learning Representations*, 2019.
- [5] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [6] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.
- [7] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal Value Function Approximators. In *International Conference on Machine Learning (ICML)*, pages 1312–1320, 2015.
- [8] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 761–768, 2011.