Representation Learning on Graphs: A Reinforcement Learning Application

Sephora Madjiheurem and Laura Toni
{sephora.madjiheurem.17, l.toni}@ucl.ac.uk
Department of Electrical and Electronic Engineering
University College London, London, UK

Abstract

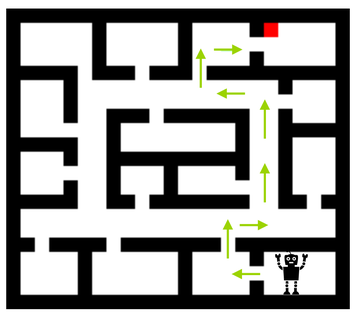
In this work, we study **value function approximation** in reinforcement learning problems with **high dimensional state or action spaces**. We first show that the **state value function is not necessarily a smooth function** in the search state space and we highlight the importance of features learning for an improved value function approximation. Then, we adopt different representation learning algorithms on graphs to **learn the basis functions that best represent the value function.** We empirically show that node2vec, an algorithm for scalable feature learning in networks, and Variational Graph Auto-Encoder constantly outperform the commonly used smooth proto-value functions in low-dimensional feature space

Background

Reinforcement Learning (RL)

- An agent takes *actions* in an *environment*
- Sequential *rewards* are observed
- Goal: find a policy that **maximises the cumulative reward** over time
- Approach: optimise the *value function*:

$$V^*(s) = \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s,a,s') V^*(s') \right)$$



Value Function Approximation

- Challenge: learning V(s) becomes intractable in high-dimensional search space ➜ A solution is to **approximate V(s)**
- Linear VF approximation aims at at approximating the value function V(s) with a linear combination of *basis functions* $\boldsymbol{\phi}(s)$:

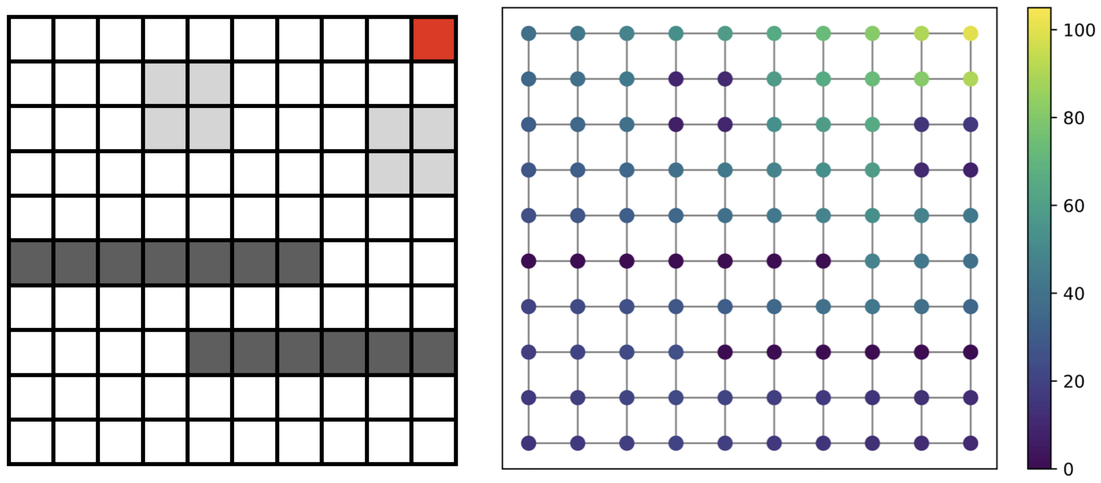
$$\tilde{V}(s|\theta) \;=\; \sum_{i=1}^{d} \theta_i \phi_i(s) \;\approx\; V(s)$$

- Challenge: identify the right set of basis functions
  ➜ Design the problem as a graph to exploit sparse representations

Graph-Based VF Approximation: State-of-the-art

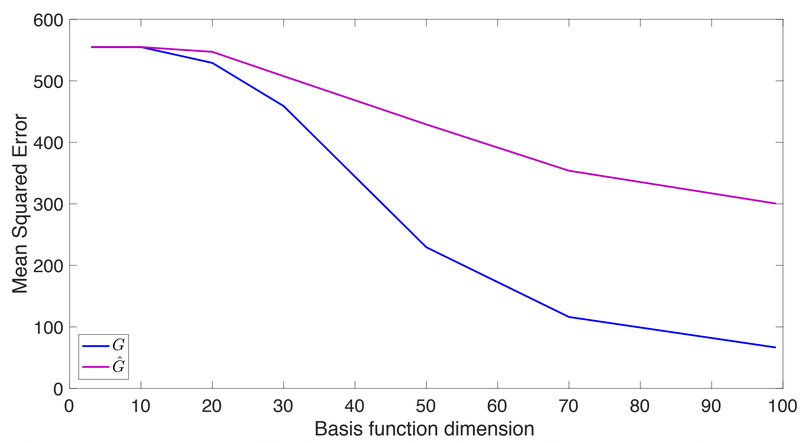
Proto-Value Functions (PVFs)

- Represent the environment as a **graph of states** where the nodes represent the states and the edges the transitions
- **Assumption: the value function is a smooth function**



Left: Maze environment: white = accessible room, dark grey = strict walls, light grey = difficult access rooms, red = goal room (+100 reward). **Right:** Optimal value function computed with value iteration.

- The basis functions are chosen to **preserve the smoothness** of the value function (the graph Laplacian's smoothest eigenvectors)
- Limitation: the value function is **not necessarily a smooth function**



➜ Need to automatically learn the basis functions from the geometry of the graph

Representation Learning on Graph

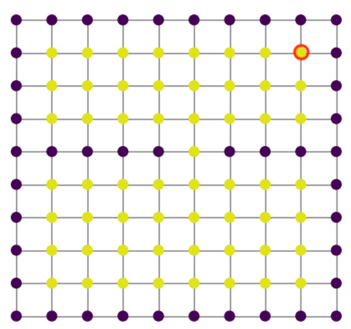
- Idea: **learn** continuous feature representations from the MDP induced state graph instead of constructing them.
- These representations should be **compact** and embed the **structural geometry** of the state space.
  - ☑ Node2vec
  - ☑ Struc2Vec
  - ☑ GraphWave
  - ☑ Variational Graph Auto-Encoder

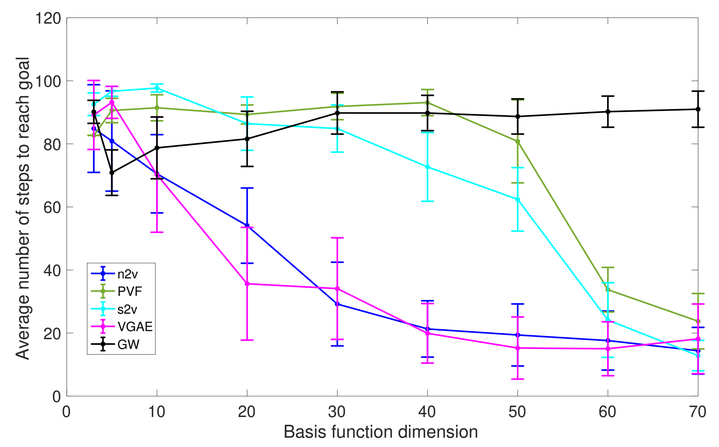
1. Sample Collection
Execute realisation of episodes and collect sample transitions in set D.

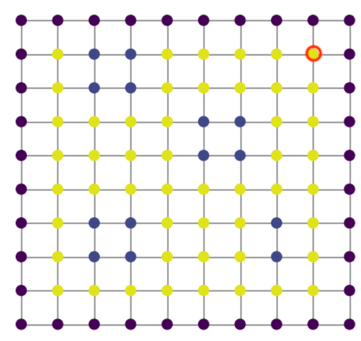
2. Representation Learning
Build the graph G from samples in D. Run your chosen representation learning algorithm on G to learn the basis functions $\phi_i(s)$.

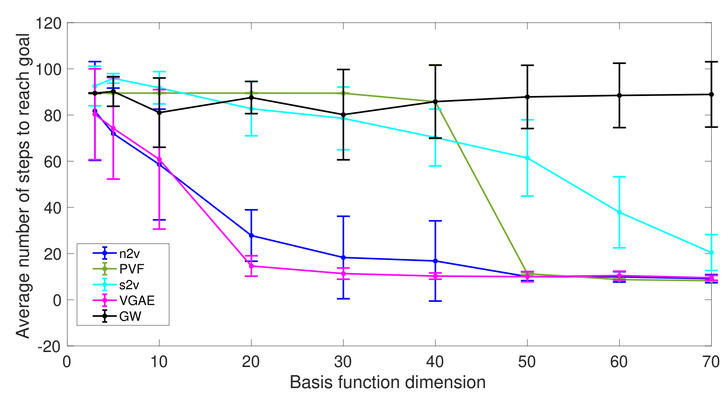
3. Control Learning
Using a parameter estimation algorithm such as LSPI, find a policy π that maximizes the value function within the linear span of the basis functions $\phi_i(s)$.

Results









Top: Two-room environment. Bottom and obstacle-room (bottom): yellow = accessible room, purple = strict walls, blue = difficult access rooms, red = goal room (+100 reward).

Conclusion

✓The the value function is not **smooth** on estimated unweighted graphs

✓Learning basis functions that **embed the geometry of the state graph** leads to improved performances

✓Such embedding models need to capture **the structural equivalence** of the nodes while preserving the l**ocal properties** of the graph.

✓**Node2vec**, and **Variational Graph Auto-Encoder** outperform the commonly used smooth proto-value functions in low-dimensional feature space