Basis Function Discovery using Spectral Clustering and Bisimulation Metrics

Gheorghe Comanici and **Doina Precup**
School of Computer Science, McGill University, Montreal, QC, Canada
gcoman@cs.mcgill.ca and dprecup@cs.mcgill.ca

Abstract

We study the problem of automatically generating features for function approximation in reinforcement learning. We build on the work of Mahadevan and his colleagues, who pioneered the use of spectral clustering methods for basis function construction. Their methods work on top of a graph that captures state adjacency. Instead, we use bisimulation metrics in order to provide state distances for spectral clustering. The advantage of these metrics is that they incorporate reward information in a natural way, in addition to the state transition information. We provide theoretical bounds on the quality of the obtained approximation, which justify the importance of incorporating reward information. We also demonstrate empirically that the approximation quality improves when bisimulation metrics are used instead of the state adjacency graph in the basis function construction process.

Introduction

Markov Decision Processes (MDPs) are a powerful framework for modeling sequential decision making in stochastic environments. One of the important challenges in practical applications is finding a suitable way to represent the state space, so that a good behavior can be learned efficiently. In this paper, we focus on a standard approach for learning a good policy, which involves learning first a value function that associates states with expected returns that can be obtained from those states. Sutton and Barto (1998) provides a good overview of many methods that can be used to learn value functions.

In this paper, we focus on the case in which function approximation must be used to represent the value function. Typically, states are mapped into feature vectors, and a set of parameters is learned, enabling the computation of the value for any given state. Having a good set of features is crucial for this type of method. Theoretically, the quality of the approximation depends on the set of features (Tsitsiklis and Van Roy 1997). In practice, the feature set affects not only the quality of the solution obtained, but also the speed of learning. Two types of methods have been proposed in recent years to tackle the problem of finding automatically a good feature set.

The first approach, exemplified by the work of Mahadevan and Maggioni (2005) (and their colleagues) relies only on information about the transitions between states. More specifically, data is used to construct a state connectivity graph. Spectral clustering methods are then used to construct state features. The resulting features capture interesting transition properties of the environment (e.g. different spatial resolution) and are reward-independent. The latter property can be viewed either as an advantage or as a disadvantage. On one hand, reward independence is desirable in order to be able to quickly re-compute values, if the problem changes. On the other hand, if the goal is to compute a good policy for a particular problem, a general feature representation that is insensitive to the task at hand and only captures general dynamics may be detrimental.

The second category of methods aims to construct basis functions that reduce the error in value function estimation (also known as the Bellman error), e.g. (Keller, Mannor, and Precup 2006; Parr et al. 2008). In this case, features are reward-oriented, and are formed with the goal of reducing value function estimation errors. Parr et al. (2008) show that this approach guarantees monotonic improvement as the number of features increases, under mild technical conditions. However, unlike in the case of spectral methods, the resulting features are harder to interpret.

The goal of this paper is to show how one can incorporate rewards in the construction of basis functions, while still using a spectral clustering approach. Specifically, we explore the use of bisimulation metrics (Ferns, Panangaden, and Precup 2004; 2005) in combination with spectral clustering, in order to create good state features for linear function approximation. Bisimulation metrics are used to quantify the similarity between states in a Markov Decision Process. Intuitively, states are close if their immediate rewards are close and they transition with similar probabilities to "close" states. Ferns, Panangaden, and Precup (2004) showed that the difference in values between two states can be bounded above using their bisimulation distance. In this paper, we prove a significant extension of this result, for the case of general linear function approximation. This theoretical result suggests that bisimulation can be used to derive a similarity measure between states for spectral clustering. We il-

lustrate this approach on several problems, showing that it has significantly better results than methods using only features based on the state dynamics without considering reward information.

We start by presenting background on Markov Decision Processes, basis function construction and bisimulation metrics. Then we present the main idea of our approach and the extension of bisimulation metric approximation guarantees to linear function approximation. Finally, we illustrate empirically the utility of bisimulation metrics for feature generation.

Background

We adopt the framework of (finite) discounted Markov Decision Processes, in which the environment is represented as a tuple $\langle S, A, P : S \times A \times S \to [0,1], R : S \times A \to [0,1], \gamma \rangle$, where S is a set of states; A is a set of actions; P is the transition model, with $P^a_{ss'}$ denoting the conditional probability of a transition to state s' given current state s and action a; R is the reward function, with R^a_s denoting the immediate expected reward for state s and action a; and $\gamma \in (0,1)$ is a discount factor. Without loss of generality, we consider $R \in [0,1]$. A policy $\pi : S \times A \to [0,1]$ specifies a way of behaving for the agent.

The model of the environment consists of P and R, which can be represented as matrices $P \in [0,1]^{|S\times A|\times|S|}, P\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the identity vector, and $R \in [0,1]^{|S\times A|}$. In the same manner, policies can also be represented as block-diagonal matrices $\pi \in [0,1]^{|S|\times|S\times A|}, \pi\mathbf{1} = \mathbf{1}$. Given an initial state distribution $d_0 \in \mathbb{R}^{|S|}$, the distribution over state-action pairs at time t is given by $d_0^T \pi (P\pi)^{t-1}$. The value of a policy $V^\pi_{d_0}$ is defined as the expected discounted return:

$$V^{\pi,d_0} = d_0^T \pi R + \gamma d_0^T \pi P \pi R + ... = d_0^T \sum_{i=0}^{\infty} (\gamma \pi P)^i (\pi R)$$

In a finite MDP, often d_0 is assumed to be uniform, in which case the value function is simply given by:

$$V^\pi = \pi R + \gamma \pi P \pi R + \cdots = \sum_{i=0}^{\infty} (\gamma \pi P)^i (\pi R)$$

The well-known Bellman equation for policy evaluation re-expresses the value function as:

$$V^\pi = \pi(R + \gamma P V^\pi)$$

from which $V^\pi = (I - \gamma \pi P)^{-1} \pi R$.

In a finite MDP, there exists a unique, deterministic policy (i.e. $\pi(s,a)$ is either 0 or 1) π^*, whose value function, V^* is optimal for all state-action pairs: $V^* = \max_\pi V^\pi$. This value function satisfies the Bellman optimality equation

$$V^* = \max_{\pi:\text{deterministic}} \pi(R + \gamma P V^*)$$

and is the limit of a recursively defined sequence of iterates:

$$V^{n+1} = \max_{\pi:\text{deterministic}} \pi(R + \gamma P V^n) \quad \text{with } V^0 = \mathbf{0}$$

Well-known incremental sampling algorithms, such as Sarsa and Q-learning, can be used to estimate these values. For a more comprehensive overview see (Puterman 1994; Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998).

Function approximation methods are used in environments that are either continuous or too large for most finite MDP algorithms to be efficient. The value function V^π is approximated using a set F of features. Let $\Phi \in \mathbb{R}^{|S|\times|F|}$ be the feature matrix, mapping states into their feature representation. Any desired value function can then be obtained by using linear approximations: $V \approx \Phi\theta$.

Representation discovery addresses the problem of finding the feature map Φ in the absence of hand-engineered basis functions (Mahadevan 2005; Parr et al. 2008; Keller, Mannor, and Precup 2006). Mahadevan introduced spectral methods that are used to learn simultaneously both the representation and the control policies. Their approach is based on the following derivation (Petrik 2007).

Let π be a policy such that there exists an orthonormal linear map $\Phi \in \mathbb{R}^{|S|\times|S|}$ and a vector λ such that $\pi P = \Phi D_\lambda \Phi^T$, where D_v denotes the diagonal map with vector v as its diagonal. Then,

$$\begin{aligned} V^\pi &= \sum_{i=0}^{\infty} (\gamma \pi P)^i (\pi R) \\ &= \sum_{i=0}^{\infty} \gamma^i (\Phi D_\lambda \Phi^T)^i (\Phi \alpha) \quad \text{for some } \alpha \\ &= \sum_{i=0}^{\infty} \gamma^i \Phi D_\lambda^i \alpha \quad \text{since } \Phi \text{ is orthonormal} \\ &= \Phi \left(\sum_{i=0}^{\infty} \gamma^i D_\lambda^i \alpha \right) = \Phi \left(D^{-1}_{\mathbf{1}-\gamma\lambda} \alpha \right) \end{aligned} \tag{1}$$

Hence, if an orthonormal basis Φ exists, it will provide an ideal representation. Moreover, the basis corresponding to the i^{th} state has weight $\alpha_i/(1-\gamma\lambda_i)$. Still, this representation is only valid for policy π, so for learning optimal control, one would have to find a representation that works for multiple policies. In (Mahadevan 2005) this is done by finding eigenfunctions of *diffusion models* of transitions in the underlying MDP using random policies. The set of feature vectors Φ that will be used in function approximation is a subset of the eigenvectors of the *normalized laplacian* (Chung 1997):

$$L = D_{W\mathbf{1}}^{-\frac{1}{2}} (D_{W\mathbf{1}} - W) D_{W\mathbf{1}}^{-\frac{1}{2}}$$

where $W \in \mathbb{R}^{|S|\times|S|}$ is a symmetric weight adjacency matrix. Note that L has the same eigenvectors as the transition matrix of a random walk determined by W. That is, we construct a graph over the state space and generate a random walk by transitioning with probabilities proportional to the incident weighs. The eigenfunctions that describe the Laplacian will describe the topology of the random graph under W. Geometrically, this provides the smoothest approximation that respects the graph topology (Chung 1997). This approach imposes no restriction on the transition model, but it ignores the reward model, which can prove to be hurtful in some situations (Petrik 2007).

Bisimulation metrics

Bisimulation metrics have been used in the context of reinforcement learning to find a good partition of states. In this case, a large MDP is reduced to a smaller one by clustering states that are close based on the value of the bisimulation metric. If clustering is done by grouping states at distance 0, then the bisimulation property guarantees that behaving optimally in the aggregated MDP (i.e., the MDP over state partitions) will result in optimal behavior in the original MDP as well. Ferns, Panangaden, and Precup (2004) present algorithms for computing the metrics based on finding a fixed point M^* of the following transformation on a metric $M \in \mathbb{R}^{|S|\times|S|}$:

$$F(M)(s,s') = \max_{a\in A}[(1-\gamma)|R^a_s - R^a_{s'}| + \gamma T_K(M)(P^a_{s,\cdot}, P^a_{s',\cdot})]$$

This recursion depends on T_K, the Kantorovich metric over two probability measures. For two vectors $p, q \in [0,1]^n$, $T_K(p,q)$ is obtained by solving the following linear program:

$$T_K(M)(p,q) = \max_{u\in\mathbb{R}^n} u^T(p-q)$$
$$\text{such that} \quad u\mathbf{1}^T - \mathbf{1}u^T \leq M \text{ and } \mathbf{0} \leq u \leq \mathbf{1}$$

Suppose S' is the state space of the aggregate MDP. Let $C : \mathbb{R}^{|S|\times|S'|}$ be the identity map of the aggregation. Then the value function V^*_{agg} of the aggregate MDP satisfies the following :

$$||CV^*_{agg} - V^*||_\infty \leq \frac{1}{(1-\gamma)^2}||\operatorname{diag}(M^*CD^{-1}_{\mathbf{1}^TC}C^T)||_\infty$$

where $||v||_\infty$ stands for the L_∞ norm of a vector v. Note that $M^*CD^{-1}_{\mathbf{1}^TC} \in \mathbb{R}^{|S|\times|S'|}$ computes the normalized distance from a state s to a cluster c. We then apply C^T to obtain the normalized distance from s to the cluster of a state s'. Then we consider only the diagonal entries of this map, and the approximation error is bounded above by its L_∞ norm (or maximum distance) between a state and the states included in the same cluster.

This bound (Ferns, Panangaden, and Precup 2004) guarantees that given some aggregation based on the Kantorovich metric, the approximation will be efficient when the largest distance inside a cluster is small. One would like to generalize the result to function approximation as well: we would like to have good approximation guarantees when the feature set used provides generalization over states that are close according to the bisimulation metric. To do this, we first prove a couple of useful small results.

For a fixed policy π, we denote by $K_\pi(M) \in \mathbb{R}^{|S|\times|S|}$ the map $K_\pi(M)(s,s') = T_K(M)((\pi P)(s), (\pi P)(s'))$. Then we can reformulate bisimulation as:

$$F(M) = \max_{\pi:det.}(1-\gamma)|(\pi R)\mathbf{1}^T - \mathbf{1}(\pi R)^T| + \gamma K_\pi(M)$$

where K_M is a square $|S| \times |S|$ matrix obtained from:

$$K_\pi(M) = \max_{U\in\mathbb{R}^{|S|\times|S^2|}} \operatorname{diag}((I_1(\pi P) - I_2(\pi P))U)$$
$$\text{such that} \quad I_1U - I_2U \leq \operatorname{diag}(I_1MI_2^T)\mathbf{1}^T \text{ and}$$
$$\mathbf{0} \leq U \leq \mathbf{1}$$

where $I_1, I_2 \in \mathbb{R}^{|S^2|\times|S|}$ are identity maps restricted on the first, respectively the second argument.

Lemma 1: Let V^n be the sequence generated by the Bellman operator (i.e. $V^n = \pi(R + \gamma PV^{n-1})$). Then

$$(1-\gamma)(\pi PV^n\mathbf{1}^T - \mathbf{1}(\pi PV^n)^T) \leq K_\pi(F^n(0)).$$

Proof: First, it was proven in (Ferns, Panangaden, and Precup 2004) that under the given circumstances, $\hat{U} = (1-\gamma)V^n\mathbf{1}^T$ is a feasible solution for the Kantorovich LP. For this particular choice of parameters $\hat{U}$,

$$\begin{aligned}&\operatorname{diag}((I_1\pi P - I_2\pi P)\hat{U})\\ &\quad= (1-\gamma)\operatorname{diag}((I_1\pi PV^n - I_2\pi PV^n)\mathbf{1}^T)\\ &\quad= (1-\gamma)(I_1\pi PV^n - I_2\pi PV^n)\end{aligned}$$

where the last equality is a simple linear algebra result which states that for any vector v, $\operatorname{diag}(v\mathbf{1}^T) = v$. Now, we can rearrange the above result in a $|S| \times |S|$ matrix to obtain: $(1-\gamma)(\pi PV^n\mathbf{1}^T - \mathbf{1}(\pi PV^n)^T)$, and the result follows. □

Eigenfunctions that incorporate reward information

Spectral decomposition methods find eigenfunctions (to be used as bases for value function representation); the eigenvalues are used as heuristics to choose only a subset of the basis set (Chung 1997). Recall that

$$V^\pi = \Phi^\pi\left(D^{-1}_{(\mathbf{1}-\gamma\lambda)}\alpha\right) \tag{2}$$

where the importance of the i^{th} basis function is $\alpha_i/(1-\gamma\lambda_i)$. Note the dependence of Φ^π on the policy used to generate the transition model. Since the ultimate goal is to obtain basis functions that are independent of π, many "surrogate" diagonalizable matrices have been proposed. They are usually reflective only of the MDP transition model, rather than the entire MDP model (Mahadevan 2005). The main problem with this approach was illustrated in (Petrik 2007), and it comes from a fault in the heuristic used to select a subset of the basis for approximation. If we only use the eigenvalues of the transition model, the constants α in Equation (2) relative to the reward function are ignored. The quality of the approximation can be affected in these situations. Nonetheless, these methods have the advantage of generalizing over MDPs that only differ in the reward function.

Let π be a fixed policy. Building on (2), we could use the same eigenvalues as heuristics, but with a different set of eigenfunctions:

$$\begin{aligned}V^\pi &= \Phi^\pi\left(D^{-1}_{(\mathbf{1}-\gamma\lambda)}\alpha\right)\\ &= \Phi^\pi D^{-1}_{(\mathbf{1}-\gamma\lambda)}D_\alpha\mathbf{1} = (\Phi^\pi D_\alpha)\left(D^{-1}_{(\mathbf{1}-\gamma\lambda)}\mathbf{1}\right)\end{aligned} \tag{3}$$

Each eigenfuntion is normalized based on the representation α of the reward under the given policy. Then the value function V^π is only represented by eigenfunctions of low order $1/(1-\gamma\lambda_i)$ values. Therefore, if the eigenvalues are to be used as a heuristic in feature extractor selection, one should extract linear state relationships that reflect the interaction

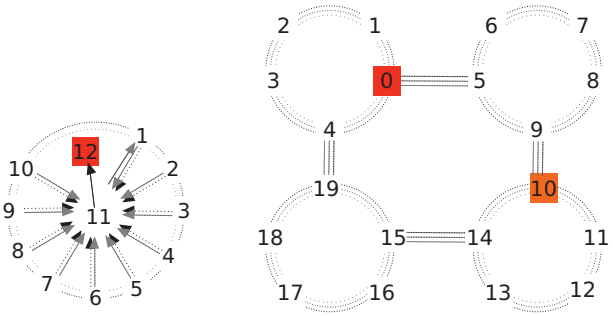

Figure 1: Left: The *Cycle MDP* is controlled by 2 actions: the first one moves uniformly in the cycle with prob. 0.5, and transitions to state 11 with prob. 0.5. The second has prob. 0.3, 0.7 respectively. From state 11 one can use the actions to move deterministically back in the cycle or to the reward state. A reward of 10 is obtained upon entering state 12, where any action transitions back in the cycle. Right: The *Hierarchical MDP* is controlled by 3 actions that either move uniformly at random in each cycle, or jump to an adjacent cycle. Inter-cycle transitions happen with prob. 0.5, 0.7, and 0.3 respectively, based on the action choice. Rewards of 10 and 15 are obtained upon entering states 0 and 10, respectively.

between reward and transition models, similarly to the way in which reward parameters α normalize the eigenfunctions of the transition model. As seen, bisimulation metrics are generated iteratively by combining reward and transition information. We now establish theoretical results that will motivate our feature generation algorithm.

Extending bisimulation bounds for general feature maps

One of the nice properties of the bisimulation metrics introduced in (Ferns, Panangaden, and Precup 2004) is the fact that if one aggregates states faithfully to the bisimulation metric, the resulting MDP has an optimal value function whose approximation error, compared to the true value function, is bounded. Below, we prove an analogous result for the case of function approximation.

Let $\Phi \in \mathbb{R}^{|S|\times|F|}$ be a feature map with the property that $\Phi\mathbf{1} = \mathbf{1}$. This generates an MDP model P_Φ, R_Φ of transitions over features rather than states, but using the same actions. The new problem becomes a smaller MDP $\langle F, A, P_\Phi, R_\Phi, \gamma\rangle$, with

$$P_\Phi = D^{-1}_{\Phi^T\mathbf{1}}\Phi^T P\Phi \quad \text{and} \quad R_\Phi = D^{-1}_{\Phi^T\mathbf{1}}\Phi^T R. \tag{4}$$

We overload the notation and use Φ as the same map from $S \to F$ and from $(S \times A) \to (F \times A)$, depending on the matrix dimensions required. Note that $P\Phi$ determines the probability to transition from a state-action pair to a feature, and the map $D^{-1}_{\Phi^T\mathbf{1}}\Phi^T$ is just a normalized average based on Φ. Also, these are well defined since $R_\Phi \in [0, 1]$ and

$$P_\Phi\mathbf{1} = D^{-1}_{\Phi^T\mathbf{1}}\Phi^T P\Phi\mathbf{1} = D^{-1}_{\Phi^T\mathbf{1}}\Phi^T P\mathbf{1} = D^{-1}_{\Phi^T\mathbf{1}}\Phi^T\mathbf{1} = \mathbf{1}$$

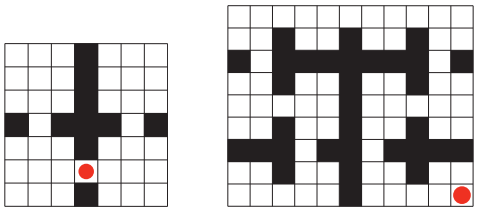

Figure 2: *7x7 and 9x11 Grid Worlds* are controlled by 4 actions representing the four movement directions in a grid. Upon using any action, the corresponding movement is performed with prob. 0.9, and the state does not change with prob. 0.1. If the corresponding action results in collision with wall, the state does not change. Rewards of 10 are obtained upon entering goal states labelled by dots.

One could now solve this new MDP and find V^*_Φ. The quality of the feature selection can be evaluated by comparing ΦV^*_Φ to V^*, similar to the approach used for aggregation methods in (Ferns, Panangaden, and Precup 2004).

Theorem 1: Given an MDP, let $\Phi \in \mathbb{R}^{|S|\times|F|}$ be a set of feature vectors with the property $\Phi\mathbf{1} = \mathbf{1}$. Then the following holds:

$$||\Phi V^*_\Phi - V^*||_\infty \leq \frac{1}{(1-\gamma)^2}||\operatorname{diag}(M^*\Phi D^{-1}_{\mathbf{1}^T\Phi}\Phi^T)||_\infty$$

Proof: First, note the following preliminary properties:

$$\Phi D^{-1}_{\Phi^T\mathbf{1}}(\Phi^T\mathbf{1}) = \Phi\mathbf{1} = \mathbf{1} \tag{5}$$

$$\operatorname{diag}(v\mathbf{1}^T) = \operatorname{diag}(\mathbf{1}v^T) = v \quad \forall v \in \mathbb{R}^n \tag{6}$$

$$\max_{\pi:\text{deterministic}}(\pi\Phi^T v) \leq \Phi^T \max_{\pi:\text{deterministic}} \pi v \quad \forall v \in \mathbb{R}^n \tag{7}$$

The last property is a simple application of the triangle inequality where all values are positive.

Now, let $V^0 = V^0_\Phi = \mathbf{0}$, and generate the sequences $\{V^n\}_{n=1}^\infty$ and $\{V^n_\Phi\}_{n=1}^\infty$ that will converge to the optimal values using the Bellman operator. Then,

$$\begin{aligned}
&|\Phi V^{n+1}_\Phi - V^{n+1}| \\
&= |\Phi \max_{\pi:\text{det}} \pi(R_\Phi + \gamma P_\Phi V^n_\Phi) - \max_{\pi:\text{det}} \pi(R + \gamma PV^n)| \\
&= |\Phi \max_{\pi:\text{det}} \pi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T(R + \gamma P\Phi V^n_\Phi) - \max_{\pi:\text{det}} \pi(R + \gamma PV^n)| \text{ (by (4))} \\
&= |\operatorname{diag}(\Phi \max_{\pi:\text{det}} \pi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T(R + \gamma P\Phi V^n_\Phi)\mathbf{1}^T) \\
&\quad - \operatorname{diag}(\Phi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T\mathbf{1} \max_{\pi:\text{det}}(\pi(R + \gamma PV^n))^T| \text{ (by (5),(6))} \\
&\leq |\operatorname{diag}(\Phi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T \max_{\pi:\text{det}} \pi(R + \gamma P\Phi V^n_\Phi)\mathbf{1}^T) \\
&\quad - \operatorname{diag}(\Phi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T \max_{\pi:\text{det}} \mathbf{1}(R^T + \gamma(V^n)^T P^T)\pi^T)| \text{ (by (7))} \\
&\leq \operatorname{diag}(\Phi D^{-1}_{\Phi^T\mathbf{1}}\Phi^T \\
&\quad \max_{\pi:\text{det}} |\pi(R + \gamma P\Phi V^n_\Phi)\mathbf{1}^T - \mathbf{1}(R^T + \gamma(V^n)^T P^T)\pi^T|)
\end{aligned}$$

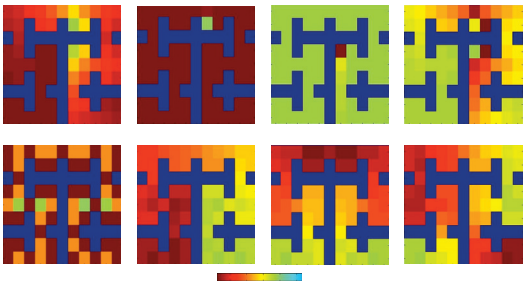

Figure 3: Illustration of the features obtained with introduced modification(top) and without (bottom). In this particular case we show the first 4 features based on the ordering provided by eigenvalues.

Algorithm 1

Given an MDP and a policy π
W stands for the similarity matrix
if desired method is based on bisimulation **then**
  $M^* \leftarrow$ bisimulation metric to some precision
  $W \leftarrow$ inverse exponential of M^*, normalized in $[0, 1]$
else
  desired method is based on state topology
  $\forall s, s' \in S, W(s, s') \leftarrow 1$ if $s \mapsto s'$ or vice-versa has probability > 0, and $W(s, s') \leftarrow 0$ otherwise
end if
$F \leftarrow$ eigenvectors of $D_{W\mathbf{1}}^{-\frac{1}{2}}(D_{W\mathbf{1}} - W)D_{W\mathbf{1}}^{-\frac{1}{2}}$
$\Phi \leftarrow$ the k vectors in F with highest eigenvalues
$\Phi_{ON} \leftarrow$ orthonormal basis of Φ
$V^\pi \leftarrow (I - \gamma P)^{-1}\pi R$, exact value function of π
$\Phi V_\phi \leftarrow V^\pi$'s projection on Φ_{ON}

Next, working on the rightmost factor, we have:

$$\begin{aligned}
&\max_{\pi:\text{det}} |\pi(R + \gamma P \Phi V_\Phi^n)\mathbf{1}^T - \mathbf{1}(R^T + \gamma (V^n)^T P^T)\pi^T| \leq \\
&\leq \max_{\pi:\text{det}} \left(|\pi R \mathbf{1}^T - \mathbf{1}(\pi R)^T| + \gamma |\pi P V^n \mathbf{1}^T - \mathbf{1}(V^n)^T (\pi P)^T|\right) \\
&\quad + \gamma \max_{\pi:\text{det}} |\pi P(\phi V_\Phi^n - V^n)\mathbf{1}^T| \\
&\leq \max_{\pi:\text{det}} (1-\gamma)^{-1}((1-\gamma)|\pi R \mathbf{1}^T - \mathbf{1}(\pi R)^T| \\
&\qquad + \gamma |(\pi P)(1-\gamma)V^n \mathbf{1}^T - \mathbf{1}(1-\gamma)(V^n)^T(\pi P)^T|) \\
&\quad + \gamma \max |\Phi V_\Phi^n - V^n| \mathbf{1}\mathbf{1}^T \\
&\leq (1-\gamma)^{-1} M_n + \gamma ||\Phi V_\Phi^n - V^n||_\infty \mathbf{1}\mathbf{1}^T
\end{aligned}$$

Note that the last derivation is a result of Lemma 1. Putting it all together we get:

$$\begin{aligned}
&|\Phi V_\Phi^{n+1} - V^{n+1}| \\
&\leq \text{diag}(\Phi D_{\Phi^T \mathbf{1}}^{-1} \Phi^T ((1-\gamma)^{-1} M_n + \\
&\qquad + \gamma \max ||\Phi V_\Phi^n - V^n||_\infty \mathbf{1}\mathbf{1}^T)) \\
&\leq (1-\gamma)^{-1} \text{diag}(\Phi D_{\Phi^T \mathbf{1}}^{-1} \Phi^T M_n) + \gamma ||\Phi V_\Phi^n - V^n||_\infty \mathbf{1}
\end{aligned}$$

We obtain the result of the statement by recursion and by taking the limits of the inequality. □

Empirical Results

From a practical point of view, the result above suggests that selecting features that respect the bisimulation metric guarantees that the error in the approximation is not large. To illustrate this idea, we modify the spectral decomposition methods presented in (Mahadevan 2005) to incorporate the bisimulation metric.

We start by defining a similarity matrix W_B that reflects the bisimulation metric M^*. We first apply to each entry of M^* the inverse exponential map, $x \mapsto e^{-x}$, and then normalize the entries to the interval $[0, 1]$, by applying the map $x \mapsto (x - \min_x)/(\max_x - \min_x)$. W_B is then contrasted to other similarity matrices that have previously been studied, known as accessibility matrices: W_A, desscribed in Algorithm 1.

Next, the normalized Laplacian is computed for both weight matrices, and the feature vectors will be selected from its set of eigenvectors, Φ_K and Φ_A, respectively:

$$L = D_{W\mathbf{1}}^{-\frac{1}{2}}(D_{W\mathbf{1}} - W)D_{W\mathbf{1}}^{-\frac{1}{2}}$$

Since most of the time these sets of eigenvectors are linearly independent, they will both allow one to represent the exact value function for a policy on the underlying MDP. Still, for control purposes, one seeks to use only a limited number of feature vectors, much smaller than the number of states, chosed with heuristics based on based on (1) and (3). Algorithm 1 outlines this approach.

Experimental setup

The environments used in the experiments are presented in Figures 1 and 2. A set of 300 policies were randomly generated for these MDPs and Algorithm 1 was used to evaluate them when different number of features were used for approximation.

Figure 4 presents a summary of the results obtained. As it can be seen, using bisimulation can provide considerable improvement in terms of the approximation power. Using feature sets that ignore reward information results in an error that becomes negligible when the number of features is as large as the state space. With the exception of the 9x11 Grid World, negligible error was obtained quite early when using bisimulation. Last but not least, we studied the behavior of the newly introduced methods when the precision of the bisimulation metric is reduced in order to improve computation time. In the case of the smaller MDP based on cycles, precision was not a factor, as the same results were obtained for metrics computed within 10^{-1} to 10^{-7} precision. This was not the case with the larger grid MDPs. In the 7x7 grid, which was specifically designed so that the topology is very indicative of the value function, bisimulation only has an advantage when computed with high precision. However, in the larger 9x11 MDP, where the transition structure is not sufficiently correlated to the value function, even a rougher approximation for bisimulation provides improvements.

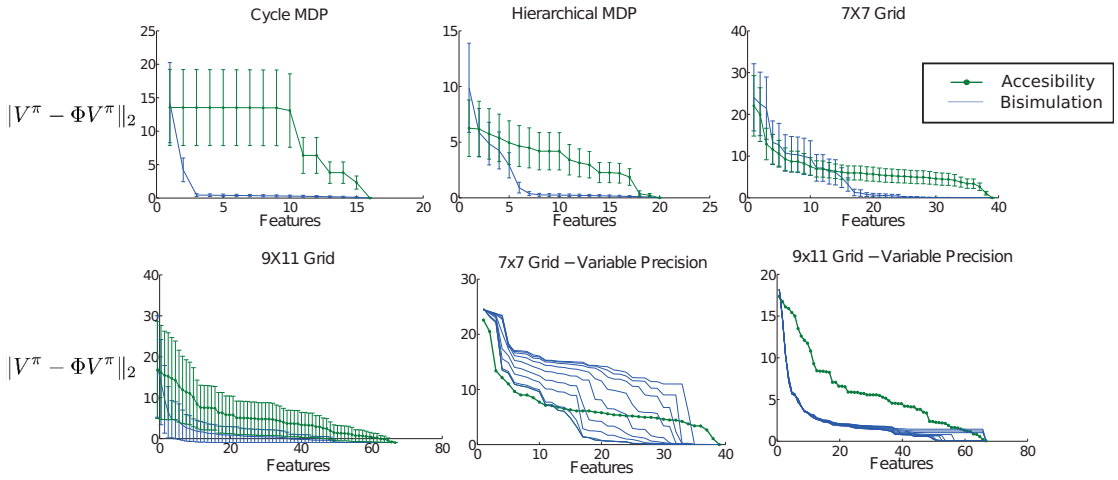

Figure 4: Empirical results are shown as comparisons between the two methods described. The best approximations possible are plotted as functions of the number of features.This is done on the MDPs described in Figures 1, 1, and 2. For 300 randomly generated policies, Algorithm 1 was used to compute the best approximation to the value function using both bisimulation and the accessibility matrix for state similarity. The graphs represent average L_2-error in approximation. The last two graphs were generated by running the same algorithm at different numerical precision(between 10^{-7} and 10^{-1}) of the bisimulation metric.

Conclusion and future work

We presented an approach to automatic feature construction in MDPs based on using bisimulation methods and spectral clustering. The most important aspect of this work is that we obtain features that are *reward-sensitive*, which is empirically necessary, according to our experiments. Even when the precision of the metric is reduced to make computation faster, the features we obtain still allow for a very good approximation of the value function.

The use of bisimulation allows us to obtain solid theoretical guarantees on the approximation error. However, the cost of computing or even approximating bisimulation metrics is often prohibitive. The results presented here were meant as a proof-of-concept to illustrate the utility of bisimulation metrics for feature construction. We are currently exploring the use of other reward-based feature construction methods, with smaller computational costs. More empirical validation of our approach is also necessary.

Acknowledgements

This work was founded in part by FQRNT and ONR. We would like to thank Pablo Samuel Castro for useful discussions and support. We also want to thank the anonymous reviewers for their useful comments.

References

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Bellman, MA.

Chung, F. 1997. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for Finite Markov Decision Processes. In *Conference on Uncertainty in Artificial Intelligence*.

Ferns, N.; Panangaden, P.; and Precup, D. 2005. Metrics for Markov Decision Processes with Infinite State Spaces. In *Conference on Uncertainty in Artificial Intelligence*.

Keller, P. W.; Mannor, S.; and Precup, D. 2006. Automatic Basis Function Construction for Approximate Dynamic Programming and Reinforcement Learning. In *International Conference on Machine Learning*, 449–456. New York, New York, USA: ACM Press.

Mahadevan, S., and Maggioni, M. 2005. Proto-Value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decission Processes. *Machine Learning* 8:2169–2231.

Mahadevan, S. 2005. Proto-Value Functions: Developmental Reinforcement Learning. In *International Conference on Machine Learning*, 553–560.

Parr, R.; Painter-Wakefiled, H.; Li, L.; and Littman, M. L. 2008. Analyzing Feature Generation for Value Function Approximation. In *International Conference on Machine Learning*, 737–744.

Petrik, M. 2007. An Analysis of Laplacian Methods for Value Function Approximation in MDPs. In *International Joint Conference on Artificial Intelligence*, 2574–2579.

Puterman, M. L. 1994. *Markon Decission Processes: Discrete and Stochastic Dynamic Programming*. Wiley.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. MIT Press.

Tsitsiklis, J. N., and Van Roy, B. 1997. An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control* 42(5):674–690.