*SYNOPSIS OF*

# AUTOMATED SPATIO-TEMPORAL ABSTRACTION IN REINFORCEMENT LEARNING

*A THESIS*

*to be submitted by*

**VIMAL MATHEW**

*for the award of the degree*
*of*

**MASTER OF SCIENCE**
*(by Research)*


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY MADRAS**
**CHENNAI**

**May 2007**

# 1 Introduction

Reinforcement learning is a machine learning framework in which an agent manipulates its environment through a series of actions, receiving a scalar feedback signal or reward after each action. The agent learns to select actions so as to maximize the expected total reward.

The nature of the rewards and their distribution across the environment decides the task that the agent learns to perform. Consider a generic agent solving multiple tasks in a common environment. The effectiveness of such an agent increases with its ability to reuse past experience in new tasks. Identifying similar representations between tasks paves the way for a possible transfer of knowledge between such tasks. A sufficiently broad definition of similarity and an ability to effectively store and reuse previous experience can greatly increase the long-term performance of a multipurpose agent.

An automated task-independent abstraction method is introduced that generalizes over tasks sharing the same state-space topology and similar dynamics. New tasks performed on this common environment are able to reuse skills acquired over previous tasks. Pruning techniques are identified for specific situations to identify relevant skills.

The interaction of the agent with the environment is modeled as a dynamical system. Identification of metastable regions in the agent dynamics leads to a partitioning of the state space. Transitions between metastable regions, events of low probability during a random-walk on the state space, are identified as useful skills.

# 2 Motivation

Curiosity drives human-beings to understand the environment around them. In the process of interacting with the environment and exploring it, often accomplishing tasks such as searching for food or shelter, we construct abstract models of the environment. These models are built independent of specific tasks, but mirror the environment. Such an internal representation of the world remains consistent across tasks and enables us to reuse skills across different tasks. The skills that we learn over time enable us to gain greater control over the environment in the long term.

To mirror this approach in the design of software agents, there is a need to design algorithms that can automatically construct task-independent state abstractions and define skills in terms of these abstractions. Such skills will enable an agent to generalize across tasks and reuse experience more effectively.

# 3 Objectives

The Reinforcement Learning framework is used to model a problem-solving software agent able to interact with an environment. Tasks are specified in terms of Markov Decision Processes (MDPs) [15]. Given a collection of tasks that share the same environment and transition dynamics but differ only in the nature of tasks

- Generate a state-abstraction consistent across all such tasks.
- Identify useful skills.
- Recognize relevant skills for a given task.
- Reuse relevant skills to solve new tasks.
- Automate the entire process without using prior domain knowledge.

# 4 Theory

## 4.1 The Laplacian on a manifold

A *Riemannian metric* $g$ on a smooth manifold associates each point $m$ on the manifold with an inner product on the tangent space $T_m M$. A manifold with an associated Riemannian metric is called a *Riemannian manifold* $(M, g)$.

A Riemannian metric $g$ allows the definition of an intrinsic second derivative. For every smooth function

$$f : M \to \mathbb{R}$$

the Hessian $Hessf$ is made up of the derivatives of $f$. Taking the trace of the Hessian with respect to the metric $g$ gives the Laplace-Beltrami operator

$$\Delta f = -div(\nabla f) = -trace_g Hessf$$

The Laplace-Beltrami operator $\Delta$ is a generalization of the Euclidean representation of the Laplace operator to a Riemannian manifold. Further references to the Laplacian on a manifold correspond to the Laplace-Beltrami operator.

Along geodesics, the second derivative coincides with ordinary numerical second derivative. By definition of the trace with respect to $g$, the Laplacian at $m \in M$ can be written as

$$\Delta f(m) = -\sum_{i=1}^{d} \frac{d^2}{dt^2} f(\gamma_i(t))\Big|_{t=0}$$

where the $\gamma_i$ are geodesics through $m$ whose velocities at $m$ form an orthonormal basis of $T_m M$.

The eigen-value problem for the Laplacian on a compact manifold can be written as

$$\Delta f = \lambda f$$

where $f = \phi$ is an *eigenfunction* and the number $\lambda$ is the corresponding eigenvalue.

The set of all eigenvalues of $\Delta$ is called the *spectrum* of $\Delta$

$$Spec(M) = \lambda_k = \{0 \leq \lambda_1 \leq \lambda_2 \leq \ldots\}$$

The eigenvalue $\lambda_0 = 0$ because the Riemannian manifold has no boundary, and corresponds to a constant eigenfunction. If the manifold is connected, the multiplicity of $\lambda_0$ is exactly one. In general, the multiplicity of the zero eigenvalue corresponds to the number of connected components of the manifold.

The Laplacian is a self-adjoint operator on the manifold, and its eigenfunctions form a complete set of real-valued orthonormal basis functions on $L^2(M)$. This also extends for the case of a Laplacian $M$ with a boundary $\delta M$. The *Dirichlet boundary condition* corresponds to functions vanishing on the boundary $\delta M$. Fourier series theory, which corresponds to the domain being an interval in the line $\mathbb{R}$, extends to functions on manifolds with Dirichlet boundaries conditions. Therefore, any function vanishing on $\delta M$ can be written as

$$f = \sum_{i=1}^{\infty} c_i \phi_i$$

where

$$c_i = \int_M f(m)\phi_i(m)dm$$

for each i = $1, 2, \ldots$. If the manifold is considered as a vibrating membrane, the eigenvalues $\lambda_i$ correspond to the frequencies of vibration, also called *harmonics*, the lowest being the *fundamental tone* [2].

## 4.2 Graph-Laplacians

For a finite number of points the manifold degenerates into a weighted graph $G = (V, E, W)$. Each point is represented by a node or vertex $v \in V$. Weighted edges indicate connections between these nodes.

The usage of graph Laplacians has often been justified by their relation to the Laplace-Beltrami operator. Hein et al. [11] provides convergence results on the graph Laplacians. The graph Laplacians converge to the Laplace-Beltrami operator when a uniform measure is defined on the graph. For a graph with a non-uniform measure the random-walk Laplacian $\Delta^{(rw)}$ alone converges to the weighted Laplace-Beltrami operator.

From the definition of the random-walk Laplacian $\Delta^{(rw)} = I - T$ it is clear that it shares the same eigenvectors as the random-walk operator $T$, and the eigenvalues are related by $\lambda_\Delta = 1 - \lambda_T$. The convergence properties of $\Delta^{(rw)}$ indicate its suitability for use for both uniform and non-uniform measures on the graph.

## 4.3 Dynamical systems

A dynamical system is a mathematical formalization defining the time-dependence of a point's position in some state space. The state of a system at any given time can be represented by a single point in an abstract space called a *state space* or *phase space* $\mathcal{M}$. The system evolves with time, modeled in terms of the change in state of a *representative point*. The evolution of such points constitutes the dynamics of the system and follows the *evolution rule* $f^t$. Though the theory of dynamical systems includes continuous-time evolution, we restrict ourselves to discrete-time evolution.

Assume the state space to be a $d$-dimensional manifold $\mathcal{M}$. For a deterministic system, the evolution rule $f^t : \mathcal{M} \to \mathcal{M}$ tells the location of a point $x \in \mathcal{M}$ after a time interval $t$. The pair $(\mathcal{M}, f)$ constitutes a dynamical system [5].

Given an initial point $x_0$, the evolution rule generates a sequence of points $x(t) = f^t(x_0)$, the *trajectory* through the point $x_0 = x(0)$. The subset of points from $\mathcal{M}$ that belongs to the trajectory of a point $x_0$ is called the *orbit* of $x_0$. Trajectories can be broadly classified as

| | | |
|---|---|---|
| **Stationary**: | $f^t(x) = x$ | For all $t$ |
| **Periodic**: | $f^t(x) = f^{t+T_p}(x)$ | For a minimum time period $T_p$ |
| **Aperiodic**: | $f^t(x) \neq f^{t'}(x)$ | For all $t \neq t'$ |

A connected subset of the state space $\mathcal{M}$ that maps onto itself on forward evolution is called an *attractor*. Each attractor in the state space has its own *basis of attraction*, the set of points that fall into the attractor on forward evolution. An attractor can be a fixed point, a periodic orbit, or an aperiodic orbit.

In many systems the evolution of the system with time involves some uncertainty. Such a *stochastic system* can be modeled in terms of the time-evolution of a random variable [1]. The state of the system at any time is given by a random variable $x$. The probability distribution $P_x(s)$ gives the likelihood of the system being in state $s$. With time this probability distribution evolves and is denoted by $P_x(s, t)$. A linear evolution rule for a stochastic system is given in terms of a *Markov chain* for cases where the outcome depends only on the state of the system at the previous time. This transition probability does not depend explicitly on time and can be written as

$$P_x(s(t)|s(t-1))$$

Assuming that there is no loss from the system, the transition probability must satisfy

$$\sum_{s(t)} P_x(s(t)|s(t-1)) = 1$$

For a finite number of states, the probability distribution at any time $t$ can be written as a vector, with each element of the vector corresponding to a state and having value $P_x(s)$. Then the transition probability $P_x(s(t)|s(t-1))$ can be written as a matrix $T_x$. For a Markov chain, the stationary distribution corresponds to a probability distribution over the state space such that $T_x.P_x^s = P_x^s$.

## 4.4 Separation of time-scales for abstraction

Assume the observation of a system over a limited time and with limited time resolution. The processes that occur in this system can be classified [5] broadly into:

- **Fast processes** that are faster than the time resolution of observation.
- **Slow processes** that are slower than the time resolution of observation.
- **Dynamic processes** that occur on the same time scale of observation.

For an ergodic dynamical system, the states of slow processes correspond to an aggregation of states of fast processes, and the dynamics of slow processes are averages over the dynamics of fast processes [1]. The separation of time scales into fast and slow processes can be used to induce spatial and temporal abstractions over the system. As an example, consider a game of hockey played between two teams. As seen by a player in the game, the dynamics of the game is very complex with the state of the system being described by locations of individual players and the strategy followed by each player. Over a longer time frame, the same game can be described in terms of the score of each team. At this time frame, individual player movements may not be relevant. The dynamics of the game can be described in terms of the change of team scores. This description is much simpler, despite complicated microscopic dynamics. At an even longer time scale, a single game can be described in terms of the identity of the winning team. This separation of time scales leads to simple macroscopic descriptions.

## 4.5 Metastability and state abstraction

The existence of attractors causes the dynamics of the agent to be largely confined to *metastable* regions of the state space, with relatively rare transitions between these metastable regions. A typical trajectory may spend relatively long periods of time within a metastable

region before *escaping* from it. Over longer time scales, the dynamics of the system can be described in terms of transitions between metastable regions. The identification of these metastable regions induces a natural abstraction on the state space. States at a macroscopic level correspond to metastable regions of the microscopic dynamics.

The relation of metastability to the spectral characteristics of the transition matrix has been studied for some time. Some of the more recent work include Bovier et al. [3], Hartfiel and Meyer [9], Hassin and Haviv [10], Nadler et al. [13, 14], Schütte [16].

## 4.6 The random-walk as a dynamical system

The evolution rule $f^t$ in a continuous domain is often the solution of a *differential equation of motion*

$$\frac{\partial}{\partial t}x = g(x)$$

Continuous *Brownian motion* can be constructed on any Riemannian manifold and is a diffusion process [8, 18]. This diffusion is described by the heat diffusion equation

$$\frac{\partial}{\partial t}u = -\Delta u$$

The Brownian motion in the continuous case has the following discrete counterpart, the random walk:

$$Pu_n(x) = \sum_y P(x,y)u_n(y) = u_{n+1}(x)$$

where $P$ is the one step transition operator of the walk. A random walk is defined on a graph such that the edge weights define a probability of transition $P$. In terms of a discrete Laplacian operator,

$$\Delta = I - P$$

and the difference operator in time

$$\partial_n u = u_{n+1} - u_n$$

the discrete heat equation can be written as

$$\partial_n u = -\Delta u$$

The transition rule $P$ forms a discrete evolution map such that

$$u_{n+t}(x) = P^t u_n(x)$$

In general, given any random-walk operator $T$ there exists a corresponding random-walk Laplacian $\Delta^{(rw)} = I - D$, and vice versa.

# 5 Spatial abstraction

This chapter models the interaction of an RL agent with an environment in terms of a stochastic dynamical system. A transfer operator is defined that generalizes over a class of RL problems that share the same domain and similar dynamics. Metastable regions are identified by the spectral analysis of this transfer operator to generate a state abstraction.

## 5.1 A Reinforcement Learning Problem and its related random-walk operator

Consider a discrete Markov Decision Process (MDP) $M$ = ($S$, $A$, $P^a_{ss'}$,$R^a_{ss'}$) with of a finite set of discrete states $S$, a finite set of actions $A$, a transition model $P^a_{ss'}$ specifying the distribution over future states $s'$ when an action $a$ is performed in state $s$, and a corresponding reward model $R^a_{ss'}$, specifying a scalar cost or reward [15]. The set of actions possible at a state $s$ is given by $A(s)$. The state-space can be modeled as a weighted graph $\overline{G}$ = ($\overline{V}$, $\overline{W}$). The states $S$ correspond to the vertices $\overline{V}$ of the graph. The edges of the graph are given by $\overline{E}$ = $\{(i,j) : \overline{W}_{ij} > 0\}$. Two states $s$ and $s'$ are considered adjacent if there exists an action with non-zero probability of transition from $s$ to $s'$. This adjacency matrix $A$ provides a binary weight ($\overline{W} = A$) that respects the topology of the state space [12]. The weight matrix on the graph and the adjacency matrix $A$ will be used interchangeably from here onwards.

A *random-walk* operator [12] can be defined on the state space of an MDP as $\mathcal{T} = D^{-1}A$ where $A$ is the adjacency matrix and $D$ is a *valency matrix*, i.e., a diagonal matrix with entries corresponding to the degree of each vertex (i.e., the row sums of $A$).

The graph Laplacians are symmetric matrices that are used to study undirected graphs. In cases where the graphs are directed or have asymmetric weight matrices $T$, the most common methods involve approximating their properties in terms of $(T + T^\star)/2$ or $T.T^\star$ [4, 7]. $T^\star$ represents the time-reversal of $T$ given by $T^\star(x,y) = \frac{\phi(y)T(y,x)}{\phi(x)}$, where $\phi T = T$. These approximations are used to study properties of directed graphs in terms of closely related undirected graphs. For the purpose of solving the Perron cluster eigenproblem, the computationally simpler approximation of $(\mathcal{T} + \mathcal{T}^t)/2$ was empirically found to yield reasonable results.

In case the goal states of the MDP are defined as absorbing states, then these absorbing states are excluded from the graph $\overline{G}$ to get a subgraph $G$. The absorbing states of the MDP are represented by the set $\delta G$ and form the *boundary* of the graph $G$. The sets $G$ and $\delta G$ are disjoint and $\overline{G} = G \cup \delta G$. The graph $G$ has a weight matrix $W$ which is a restriction of $\overline{W}$ to the vertices $G$. This weight matrix $W$ corresponds to the Dirichlet boundary condition and defines a random walk *killed* at exiting the set $G$ [18]. The random-walk

operator created from this weight matrix forms a *transition operator* on the graph G and will be denoted by $T|_G$.

## 5.2 Learning a global random-walk operator across multiple RL tasks

Consider a set of RL tasks that use a common state space $\mathbb{S}$ and share its topology. In general an RL task might be defined only over a subset $S \subseteq \mathbb{S}$. Even for RL tasks defined over $\mathbb{S}$, the goal states might be absorbing and transitions from absorbing states for this task should not be used to estimate the random walk operator for $\mathbb{S}$. The definition of the random-walk operator avoids this error by excluding the absorbing states. Therefore the weighted graph $\mathcal{G}$ and the random walk operator $T|_{\mathcal{G}}$ for the entire state space $\mathbb{S}$ can be estimated in terms of the random walk operators $T|_G$ for individual tasks.

## 5.3 Metastability and the Random-Walk operator

The random-walk operator $T|_{\mathcal{G}}$ (shortened to $T$ from here onwards) provides the probability distribution over the state space after a single step of diffusion, and corresponds to a transfer operator over probabilities. $T$ represents a transfer operator or infinitesimal generator of a diffusion on the graph. The relation between the eigenvalues of such stochastic matrices and metastability has been mentioned earlier.

Recent work in the area of conformational molecular dynamics has lead to robust algorithms to identify metastable states [6, 19]. The random-walk operator constructed here can be considered as a discretized spatial transition operator [16] needed for the PCCA+ algorithm.

## 5.4 The PCCA+ algorithm

The PCCA+ algorithm [6, 19] is a spectral clustering method that identifies a partition on the state-space based on metastability. A set of *almost characteristic functions* $\widetilde{\chi}$ is used to identify individual clusters. PCCA+ provides a geometric interpretation of eigenvector data as a simplex, and uses deviation from simplex structure, measured as a *Min-Chi* value, to identify the number of clusters.

States can be assigned to cluster by choosing the maximum $C_i = \mathrm{argmax}_k \, \widetilde{\chi}_{ik}$. A more robust approach would be to use an *inner simplex algorithm* [20] to partition the states.

Refer the original papers [6, 19, 20] for details regarding the actual implementation of the algorithm.

# 6 Temporal abstraction

The dynamics of the agent can be described as largely confined to metastable abstract states with comparatively fewer transitions between these abstract states. Transitions between abstract states are useful because they define low proabbility transitions and can be used to improve sampling of the state space. The *Options* framework is used to model transitions between abstract states as *subtask options*.

## 6.1 Augmented Options and reward-free MDPs

An *option* [17] in an MDP $\mathcal{M} = (S, A, P^a_{ss'}, R^a_{ss'})$ is defined by the tuple $O = (\mathcal{I}, \pi, \beta)$, where the initiation set $\mathcal{I} \subseteq S$ is the set of states in which the option can be invoked, $\pi$ is the policy to be followed while executing the option, and the termination function $\beta : S \to [0, 1]$ gives the probability of the option terminating in any given state.

For a subtask corresponding to a traversal between two abstract states given by the state spaces $S_1, S_2 \subseteq \mathbb{S}$, an *augmented* option is constructed. Consider an *augmented* option $O_{S_1 \to S_2} = (\mathcal{I}, \pi, \beta, R^a_{ss'})$ defined over a *reward-free MDP* $M = (S, A, P^a_{ss'})$ where $S \supseteq S_1 \cup S_2$.

The rewards related to a *subtask option* can now be associated with the subtask. This *intrinsic* reward can be designed to learn the subtask either by a reward on termination and a *discounted return* formulation or by any other suitable means.

## 6.2 Automatic construction of Subtask Options

A *global* reward-free MDP $\mathbb{M} = (\mathbb{S}, \mathbb{A}, \mathbb{P}^a_{ss'})$ can be defined where $\mathbb{S}, \mathbb{A}$, and $\mathbb{P}^a_{ss'}$ are part of a global environment.

The coupling matrix $\widetilde{W}$ identifies abstract states that are connected to each other. The transitions between two connected abstract states can be stored as a subtask option. For every transition $S_1 \to S_2$, a subtask option $O_{S_1 \to S_2} = (\mathcal{I}, \pi, \beta, R^a_{ss'})$ can be defined with $\mathcal{I} = S_1$ and $\beta = \chi(S_2)$.

The undirected graph approximation for directed graphs can lead to the creation of spurious subtask-options. A generic agent will not be able to learn a policy for such subtasks and therefore they will never be added to the knowledge-base of a generic agent.

### 6.3 Spatio-Temporal abstraction and the Proto-value function framework

The proto-value function framework [12] uses the eigen-vectors of the graph Laplacian as an orthonormal basis for value-functions. These eigen-vectors are the same as that of the random-walk operator and differ only in their eigen-values. Calculating the first $k$-eigenvectors of the graph Laplacian is the most computationally intensive part of the PCCA+ algorithm, a task shared with the proto-value function framework. Therefore if the implementation of PCCA+ is merged with an agent running the RPI algorithm, then the state space abstraction can be obtained with a relatively low increase in computational cost.

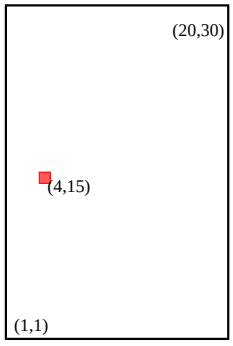
## 7 Design of experiments

The experiments were designed to validate the abstraction algorithm against the following

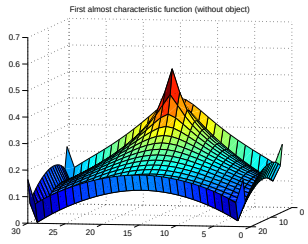
- **Asymmetry**: A few existing state-partitioning algorithms are either biased toward equal-sized partitions, or differ from topological notions of bottlenecks. The first experiment was designed to verify the algorithm for asymmetrically sized partitions on the state space.
- **Global structure**: Some automated state-abstraction algorithms restrict themselves to localized searches for *access states*, and do not identify cases where bottlenecks may be related. The global structure of the state space alone can provide this relation. The second experiment consists of two rooms connected by two-doors placed, with the doors placed far apart. A topological abstraction would identify two rooms. In this abstraction, it would not matter which door caused a transition between the two rooms.
- **Separation of time-scales among state variables**: A few methods find state abstractions by identifying state variables that change at a slower rate than other state variables. The third experiment consists of an object in a room, with a state variable indicating whether or not the agent is holding the object. This experiment validates the proposed abstraction method against such forms of state representation.

## 8 Object in a room

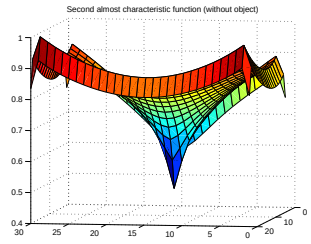
This synopsis restricits itself to a single experiment, identifying state abstractions based on a separation of time-scales among state variables.



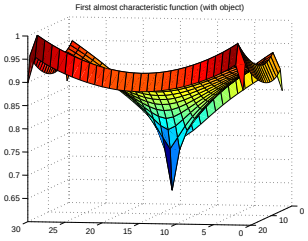

(a) Object in a room



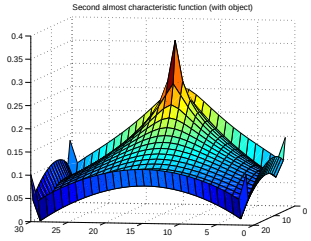

(b) $\widetilde{\chi}_1$(Without object)




(c) $\widetilde{\chi}_2$(Without object)




(d) $\widetilde{\chi}_1$(With object)




(e) $\widetilde{\chi}_2$(With object)

Figure 1: Experimental results

This grid-world (Figure 1a) has an object placed at (4, 15). If an agent reaches this state, it automatically picks up the object. The state space is defined in terms of the tuple $(HoldingObject, x, y)$ where the first feature indicates whether the agent is currently holding the object or not.

The state space consists of two abstract states, one with the agent not holding the object and another with the agent holding it. These two abstract states are connected by a directed transition from states near (4, 15) that are not holding the object to the state (4, 15) and holding the object.

The random walk operator on the directed graph corresponding to this state space is approximated by an undirected graph. This approximation differs from the original dynamics in a probabilistic dropping of the object on transitions out of (4, 15), leading to the neighborhood of $(HoldingObject, 4, 15)$ increasing from $4$ to $8$ ($4$ with $HoldingObject$, and $4$ with $\overline{HoldingObject}$).

The algorithm identified the two clusters corresponding to the two abstract states $HoldingObject$ and $\overline{HoldingObject}$. For states corresponding to $HoldingObject$, $\widetilde{\chi}_1 > \widetilde{\chi}_2$ (Figures 1d and 1e). Similarly, for the abstract state $\overline{HoldingObject}$, $\widetilde{\chi}_2 > \widetilde{\chi}_1$ (Figures 1c and 1b). Therefore the almost characteristic function $\widetilde{\chi}_1$ corresponds to $HoldingObject$ and $\widetilde{\chi}_2$ to $\overline{HoldingObject}$. Three states with transitions to the location (4, 15) were incorrectly classified as belonging to the abstract state $HoldingObject$. The simplex structure correctly classified all states.

The two automatic options constructed correspond to picking up the object *PickUpObj* and dropping the object *DropObj*. The *DropObj* option results due to the undirected graph assumption made earlier. Such an option will never get used by an agent.

The state $(HoldingObject, 4, 15)$ has eight neighbors in the undirected graph approximation. Other states have a maximum of only $4$ neighbors. The comparatively larger neighborhood helps in identifying the state transition as a salient event.

# 9 Conclusions

The thesis introduces a novel approach to automated spatio-temporal abstraction in Reinforcement Learning. The contributions of the thesis are:

- A mathematical framework for state abstraction in RL is provided by using ideas from dynamical systems.
- A task-independent state abstraction is identified that is common to all tasks sharing the same random-walk operator.
- Temporal abstractions corresponding to relatively-rare events are identified over abstract states. These abstractions extend the Intrinsically Motivated RL (IMRL) framework by the ability to automatically determine salient events.
- The Proto-value function framework is related to the IMRL framework through the use of eigenvectors of the graph Laplacian.

# 10 Future work

The global random-walk operator constructed here uses adjacency information alone and disregards actual probabilities of transition. The transition probabilities provide a more accurate reflection of the environment. A study of the trade-offs between suitable approximations and sample complexity might be useful.

The random-walk operator is expected to be symmetric, with approximations required for other cases. There is a need for improved approximations, especially in the case of irreversible dynamics. Alternate methods for state abstraction that do not need symmetric operators might lead to more generic algorithms.

The size of the agent is defined informally, and is determined by the transition probabilities on the state space. A formalized definition of agent-size, along with a way to simulate various sizes will provide a way to construct state space hierarchies.

The abstraction methods presented here are task independent. Useful locations within abstract states could be identified by observing reward structures from the given task. Temporal abstractions restricted to regions within an abstract state could provide a finer granularity on the state space.

The global random-walk operator constructed here depends on the topology of the state space alone. The Bellman operator could be used to obtain a localized distance metric over the state space. Since such a transfer operator would include reward information, the generated spatial abstraction would be reward based.

At a high level, the dynamics of the agent can be described in terms of abstract states and *deterministic* transitions between them. Due to the deterministic nature of policies at this level, methods from deterministic planning domains can be used to identify policies.

# 11 Publication

1. B. Ravindran, Andrew Barto, and Vimal Mathew (2007). Deictic Option Schemas. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1023–1028.

# References


[1] Bar-Yam, Y. (1997). *Dynamics of Complex Systems*. Addison-Wesley.

[2] Berger, M. (2002). *A Panoramic View of Riemannian Geometry*. Springer.

[3] Bovier, A., Eckhoff, M., Gayrard, V., and Klein, M. (2000). Metastability and small eigenvalues in Markov chains. *Journal of Physics A: Mathematical and General*, 33(46):L447–L451.

[4] Chung, F. R. K. (2005). Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9:1–19.

[5] Cvitanović, P., Artuso, R., Mainieri, R., Tanner, G., and Vattay, G. (2005). *Chaos: Classical and Quantum*. Niels Bohr Institute, Copenhagen. `ChaosBook.org`.

[6] Deuflhard, P. and Weber, M. (2005). Robust Perron Cluster Analysis in Conformation Dynamics. In Dellnitz, M., Kirkland, S., Neumann, M., and Schütte, C., editors, *Special Issue on Matrices and Mathematical Biology*, volume 398C of *Linear Algebra and its Applications*, pages 161–184. Elsevier.

[7] Fill, J. (1991). Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the Exclusion process. *The Annals of Applied Probability*, 1(1):62–87.

[8] Grigor'yan, A. (1990). Analytic and geometric background of recurrence and non-explosion of the Brownian motion on Riemannian Manifolds. *Bulletin of American Mathematical Society*, 36:135–249.

[9] Hartfiel, D. J. and Meyer, C. D. (1998). On the structure of stochastic matrices with a subdominant eigenvalue near 1. *Linear Algebra and its Applications*, 272(1–3):193–203.

[10] Hassin, R. and Haviv, M. (1992). Mean passage times and nearly uncoupled Markov chains. *SIAM J. Discret. Math.*, 5(3):386–397.

[11] Hein, M., Audibert, J.-Y., and von Luxburg, U. (2006). Graph Laplacians and their convergence on random neighborhood graphs. Available at arXiv:math.ST/0608522.

[12] Mahadevan, S. and Maggioni, M. (2006). Proto-value Functions: A Laplacian Framework for learning Representation and Control in Markov Decision Processes. Technical Report TR-2006-45, University of Massachusetts, Department of Computer Science.

[13] Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I. (2006a). Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, Cambridge, MA.

[14] Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I. (2006b). Diffusion Maps, Spectral Clustering and reaction coordinates of Dynamical Systems. In *Diffusion Maps and Wavelets*, volume 21 of *Applied and Computational Harmonic Analysis*, pages 113–127. Elsevier.

[15] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

[16] Schütte, C. (1999). *Conformational Dynamics : Modelling, Theory, Algorithm, and Application to Biomolecules*. Habilitation thesis, Dept. of Mathematics and Computer Science, Freie Universität Berlin.

[17] Sutton, R. S., Precup, D., and Singh, S. P. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

[18] Telcs, A. (2006). *The Art of Random Walks*, volume 1885 of *Lecture Notes in Mathematics*. Springer.

[19] Weber, M. (2006). *Meshless Methods in Conformation Dynamics*. PhD thesis, Department of Mathematics and Computer Science, Freie Universität Berlin.

[20] Weber, M., Rungsarityotin, W., and Schliep, A. (2004). Perron Cluster Analysis and Its Connection to Graph Partitioning for Noisy Data. Technical Report ZR-04-39, Zuse Institute Berlin.

# Proposed Contents of the Thesis