# Regularized Reinforcement Learning with Performance Guarantees

Mahdi Milani Fard

Doctor of Philosophy

School of Computer Science
Faculty of Science

McGill University

Montreal,Quebec

2014-03-01

A thesis submitted to McGill University
in partial fulfillment of the requirements
of the degree of Doctor of Philosophy

## DEDICATION

To my wife, parents and supporting friends.

# ACKNOWLEDGEMENTS

I would like to thank all members of McGill's Reasoning and Learning lab who provided me with useful thoughts and ideas throughout my graduate studies. I am particularly thankful to my supervisor, Joelle Pineau, for her relentless help and support over this time. I would also like to thank Doina Precup for her invaluable contributions to my research and studies at McGill University.

Special thanks goes to Yuri Grinberg, Amir-massoud Farahmand and Csaba Szepesvári for their contributions to my research and publications. I am also thankful to Pierre-Luc Bacon for his help in writing the French abstract, and Angus Leigh for being an awesome lab-mate.

## ABSTRACT

Reinforcement learning covers a broad category of control problems in which the learning agent interacts with the environment in order to learn to maximize the collected utility. Such exploratory interaction is often costly, encouraging sample-efficient algorithms to be used in the process. This thesis explores two avenues that can help improve the sample complexity of such algorithms, one through prior domain knowledge on the dynamics or utilities, and the other by leveraging sparsity structures in the collected observations.

We take advantage of domain knowledge in the form of a prior distribution to develop PAC-Bayesian regularized model-selection algorithms for the batch reinforcement learning problem, providing performance guarantees that hold regardless of the correctness of the prior distribution. We show how PAC-Bayesian policy evaluation can leverage prior distributions when they are informative and, unlike standard Bayesian approaches, ignore them when they are misleading.

In the absence of prior knowledge, we explore regularization of model-selection through random compressed sensing when generating features for the policy evaluation problem. In commonly occurring sparse observation spaces, such compression can help control the estimation error by substantially reducing the dimensionality of the regression space, at the cost of a small induced bias.

Our proposed methods can provably outperform the alternatives in sample or time complexity, showcasing how informed or agnostic regularization can further impact the effectiveness of reinforcement learning algorithms.

# ABRÉGÉ

L'apprentissage par renforcement couvre un grand nombre de problèmes de contrôle pour lesquels l'agent apprenant interagit avec l'environment afin d'apprendre à maximiser l'utilité collectée. La phase d'interaction exploratoire est souvent coûteuse, encourageant l'utilisation d'algorithmes faisant un usage efficace des échantillons. Cette thèse explore deux avenues de recherche qui peuvent aider à améliorer l'efficacité d'échantillonnage de ces algorithmes: d'une part en utilisant une connaissance apriori des dynamiques ou utilités, et d'une autre en exploitant les structures creuses ("sparse structures") dans les observations collectées.

Nous tirons profit de la connaissance du domaine sous forme de distribution apriori afin de développer des algorithmes régularisés de sélection de modèle de type PAC-Bayésien pour le problème d'apprentissage par renforcement par lots. Nous obtenons ainsi des garanties de performance applicables indépendamment du choix de distribution apriori. Nous démontrons comment les politiques d'évaluation PAC-Bayésiennes peuvent faire usage des distributions apriori lorsque celles-ci sont informatives, et dans le cas contraire, arrivent à les ignorer.

En l'absence de connaissance apriori, nous explorons la régularisation de la sélection de modèle par l'entremise de l'acquisition comprimée lorsque des caractéristiques doivent être générées pour le problème d'évaluation de politique. Pour les espaces d'observations creux les plus fréquents, une telle compression peut aider à contrôler l'erreur d'estimation en réduisant significativement la dimension de l'espace de régression sans induire un biais trop important.

Les méthodes que nous proposons peuvent surpasser de manière prouvable les alternatives quant à leur efficacité d'échantillonnage et complexité en temps, démontrant du même coup comment la régularisation informée ou agnostique peut avoir un impact sur l'efficacité des algorithmes d'apprentissage par renforcement.

TABLE OF CONTENTS

## LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1
# Introduction, Motivation and Rationale

Planning and decision making have been well studied in the Artificial Intelligence (AI) community. Reinforcement Learning (RL) is a branch of AI that tries to develop a computational approach to solving the problem of learning through interaction. RL is the process of learning what to do—how to map situations to actions—so as to maximize a numerical reward signal (Sutton and Barto, 1998). The agent has to interact with the environment (often times intelligently) to gather information and draw conclusions regarding which action is the optimal one in each situation.

Many methods have been developed to solve the RL problem with different types of environment and different types of agents. These methods are usually based on probabilistic models of the environment along with a Markovian property on the system state. It is relatively easy to find an optimal policy for a given RL problem when the dynamics of the environment are known, using methods such as value iteration or policy iteration (Sutton and Barto, 1998).

## 1.1 Planning with Uncertainty

When the dynamics are not known, one has to interact with the environment and collect data to build approximate models of the problem. We will thus be working with noisy models, based on a finite number of interactions with the environment. This leads to a type of uncertainty that is usually referred to as *extrinsic uncertainty* (Mannor et al., 2007), referring to the variance in our estimates of the model parameters. This is particularly problematic, as the performance of an optimal action based on an imperfect model might be significantly worse than best one, causing the agent to end up with a sub-optimal behaviour.

To reduce the extrinsic variance and solve the problem of over-fitting, similar to many other fields in machine learning, we can apply model regularization before any decision making. This is often achieved by introducing a bias into the model-selection mechanism, hence the *bias–variance tradeoff* (Hastie et al., 2009). These are heuristics that bias us towards particular models when we do not have enough data to arrive at a clear conclusion. They can be based on specific domain knowledge, or based on structural assumption such as sparsity or smoothness.

Domain-specific bias can be provided in form of a Bayesian prior distribution. Bayesian inference is then applied either in the model-based setting (Strens, 2000) or model-free setting (Engel et al., 2003, 2005). Bayesian methods, unlike the uninformed Probably Approximately Correct (PAC) methods, are greedy with respect to the policy inferred by the prior. As a result of this greediness, Bayesian methods can converge to suboptimal solutions (Kolter and Ng, 2009b).

General domain-independent assumptions have also been used for model regularization in machine learning. Sparsity and smoothness are the most commonly used assumptions in many machine learning areas and have been applied to the RL setting in different forms. One can use specific domain-independent Bayesian priors that impose smoothness on the model parameters or utility of actions. This requires similar states and actions to have similar dynamics or utilities. Such analysis translates into an $\ell^2$ regularization on the parameter space of a function approximator, and is shown to provide performance guarantees even in continuous state-spaces (Farahmand et al., 2009b,a). Sparsity in RL model-selection, on the other hand, is either imposed by $\ell^1$ regularization (Kolter and Ng, 2009a) or explicit feature selection or feature extraction methods (Menache et al., 2005; Keller et al., 2006; Parr et al., 2007; Geramifard et al., 2011).

## 1.2 Research Focus

Our research focuses on the RL regularization problem in two ends of the spectrum: domain-specific guided regularization using Bayesian prior distributions, and agnostic feature selection based on domain-independent sparsity assumption in the feature space. In both cases, we investigate methods for which we can get probabilistic performance guarantees with only mild assumptions on the problem domain.

We incorporate domain-specific information through prior distributions. Our use of prior is, however, fundamentally different from that of the traditional Bayesian analysis, in that we do not assume the prior to be capturing the true probabilistic nature of the problem. Instead, we treat the prior as a heuristic that guides our search for models that better match the observed data. This is done through a PAC-Bayesian methodology, first introduced for supervised learning (McAllester, 1999a). We apply our methods to both model-free and model-based RL, and propose algorithms based on the PAC-Bayesian analysis in discrete and continuous state-spaces. Our empirical results in transfer learning scenarios confirm that the PAC-Bayesian methodology is able to leverage prior distributions when they are informative and, unlike standard Bayesian RL approaches, ignore them when they are misleading.

For the cases where domain-specific information is not available, we develop methods that exploit general sparsity assumption on the state-space. We tap into the methodology of compressed sensing (Donoho, 2006) and apply it for feature extraction in the RL problem. Ghavamzadeh et al. (2010) have recently applied random projections to derive RL algorithms for feature selection in high-dimensional state-spaces. This is, however, only a first step of such applications and is the subject of active research. In that regard, we investigate the combination of compressed sensing methods such as random projections with conventional RL feature extraction methods based on Bellman residuals (Parr et al., 2007). We aim to provide performance guarantees on such methods with minimal assumptions on the problem domain. Our results could be further used to derive online feature generation algorithms that adaptively increase the complexity of the model as more data becomes available.

## 1.3 Thesis Statement

This work extends and analyses regularization methods in reinforcement learning, when either domain-specific information is available through Bayesian priors, or where minimal domain-independent sparsity assumptions are used. We apply PAC-Bayesian analysis for the case of informed regularization, and use ideas from compressed sensing for agnostic regularization.

**Impact and applications:** We expect the result of our research to have considerable impact in sample efficacy both in batch and online reinforcement learning, specially in cases where general reinforcement learning models are not restrictive enough to converge to good solutions with limited training data. Regularization can be used in these cases to control for the complexity of the model accordingly.

## 1.4 Statement of Contributions

The following is a detailed list of novel theoretical and empirical contributions included in this thesis:

- A PAC-Bayesian bound for model-based RL is provided in Theorems 6.2. An empirical analysis of this bound is provided in Section 6.6.1.
- A PAC-Bayesian bounds for model-free RL in finite-state problems is provided in Theorem 6.3, with corresponding empirical analysis included in Section 6.6.2.
- A PAC-Bayesian bound for continuous-state RL problem with value function approximation is provided in Theorem 6.5. An empirical analysis is included in Section 6.6.3.
- A worst-case analysis of compressed least-squares regression in sparse spaces is provided in Theorem 8.2. The corresponding empirical evaluations are presented in Section 8.5.
- An analysis of compressed least-squares regression with non-i.i.d. data in sparse spaces is presented in Theorem 8.3, with application to feature generation in RL presented in Theorem 9.1 and Lemma 9.3. The empirical analysis is provided in Section 9.5.

## 1.5 Statement of Authorship

This thesis is based on several results published in peer reviewed conferences. The author of this thesis is the main author in charge of most of the theoretical and empirical analysis included in all the following publications. Co-authors provided supervision on the research, contributed to the idea and helped with small parts of the theoretical analysis.

- Theorems 6.2 and 6.3 with corresponding empirical analysis, extend the work of Fard and Pineau (2010).
- Theorem 6.5 and its corresponding empirical analysis, extend the work of Fard, Pineau, and Szepesvári (2011)
- Theorem 8.2 and the corresponding empirical evaluations, extend the work of Fard, Grinberg, Pineau, and Precup (2012).
- Theorems 8.3 and 9.1, Lemma 9.3, and their empirical analysis extend the work of Fard, Grinberg, Farahmand, Pineau, and Precup (2013).

## 1.6 Overview

This thesis is organized in 3 parts. Part I, goes over the notation and background of RL. In Part II, we cover the technical background pertaining to PAC and PAC-Bayesian analysis, and then present our PAC-Bayesian methods for RL. Part III contains a review of the compressed sensing literature and the application of compressed regression to feature generation for the RL problem. We finish by an overall conclusion and discussion in Chapter 10. All basic notations used in the thesis are defined in Appendix A.

# Part I

# Reinforcement Learning Background and Notation

# CHAPTER 2
# Markov Decision Process and Sequential Decision Making

This chapter is a review of background material on decision theory in reinforcement learning (RL). We look into the main concepts behind sequential decision making and the mathematical formulations used in RL. We also go through some of the key algorithmic methods for learning and decision making in RL.

## 2.1 Sequential Decision Making

Decision theory was originally introduced in the context of game theory with early applications in economics (Neumann and Morgenstern, 1944). In its modern form, it can be thought of as the study of probabilistic processes along with utility theory. The assumption is that there is an agent acting in an environment, performing actions, and receiving signals. Utility theory assumes that part of this signal from the environment is a numerical value that specifies how good the outcome was. This could be the health measure of a patient, or the profit of a trading agent.

The system is assumed to evolve in a probabilistic manner. In the most general case, the future of the system is a probabilistic function of the entire history of the agent's interactions with its environment. However, often times we deal with systems that are known to have the *Markovian property*. This property assumes that the system has a state-space. At each point in time, the system is in one particular state (which might or might not be known to the agent) and the future of the system solely depends on the current state and the future actions.

The sequential decision process deals with the problem of finding strategies for the acting agent that will maximize the long-term future utility. This could be the average or sum of future utilities at each time step.

Depending on the type of signal received by the agent and the probabilistic nature of the process, we can divide sequential decision-making problems into several classes. Among these, we look at the Markov Decision Process, which assumes the agent knows the state of the system.

## 2.2 Markov Decision Processes

A Markov Decision Process (MDP) is a model of system dynamics in sequential decision problems that involves probabilistic uncertainty about future states of the system (Bellman, 1957). The system is assumed to be in a state at any given time. The agent observes the state and performs an action accordingly. The system will then make a transition to the next state and the agent will receive some reward.

The MDP model allows us to find the optimal action at each state such as to maximize the long-term future reward.

Formally, an MDP is defined by the 5-tuple $(\mathcal{S}, \mathcal{A}, T_0, T, R, \gamma)$:

- **States**: $\mathcal{S}$ is the set of states. The state usually captures the complete configuration of the system. Once the state of the system is known, the future of the system is independent from all the history. This means that the state of the system is a sufficient statistic of the history of the system.

In a robotic application, for instance, the state could be the position of the robot. Thus once we know where the robot is, it is not important which path it took to get there.

- **Actions**: $\mathcal{A} : S$ is the set of actions allowed by the RL agent. For example, a robot can move in several directions, each of which constitutes an action.
- **Starting distribution**: $T_0 \in \mathcal{M}(\mathcal{S})$ defines the distribution of the state at the beginning of the agent's interaction with the environment.
- **Transition Probabilities**: $T : S \times A \to \mathcal{M}(\mathcal{S})$ defines the transition probabilities of the system. This function specifies how likely it is to end up at any state, given the current state and a specific action performed by the agent. As stated before, transition probabilities are specified based on the Markovian assumption. That is, if the state of the system at time $t$ is denoted by $S_t$ and the action at that time is $A_t$, then we have:

$$\mathbb{P}(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = \mathbb{P}(S_{t+1}|S_t, A_t). \tag{2.1}$$

  We focus on *homogeneous* processes in which the system dynamics are independent of the time. Thus the transition function is stationary with respect to time:

$$T(s, a) \overset{\text{def}}{=} \mathbb{P}(\cdot|S_t = s, A_t = a). \tag{2.2}$$

- **Rewards**: $R : \mathcal{S} \times \mathcal{A} \to \mathcal{M}([0, R_{\max}])$ is the probabilistic reward model. Depending on the current state of the system and the action taken, the agent will receive a reward drawn from this model. We focus on homogeneous processes in which, again, the reward distribution does not change over time. If the reward at time $t$ is denoted by $R_t$ then we have:

$$R_t \sim R(S_t, A_t). \tag{2.3}$$

Sometimes the reward is taken to be deterministic. We will be using the general probabilistic model throughout this thesis. The mean of this distribution will be denoted by $\bar{R}(.,.)$.

- **Discount Factor**: $\gamma \in [0, 1)$ is the discount rate used to calculate the long-term return. It is a way of trading off between short-term and long-term rewards. It can also be thought of as the probability that the process dies at each step.

The agent starts in an initial state $S_0 \sim T_0$. At each time step $t$, an action $A_t \in \mathcal{A}$ is taken by the agent. The environment then makes a transition to $S_{t+1} \sim T(S_t, A_t)$ and the agent receives an immediate reward $R_t \sim R(S_t, A_t)$.

The goal of the agent is to maximize the discounted sum or rewards, which is usually referred to as the *return* (denoted by $D$):

$$D = \sum_t \gamma^t R_t. \tag{2.4}$$

In the finite horizon case, this sum is taken up to the horizon limit and the discount factor can be set to 1. However, in the infinite horizon case the discount factor should be less than 1 so that the return has finite value. The return on the process depends on both the stochastic transitions and rewards, as well as the actions taken by the agent.

## 2.3 Policy and Value Function

Policy is a way of defining the agent's behaviour with respect to the changes in the environment. A (probabilistic) policy on an MDP is a mapping from the state-space to a distribution over the action space:

$$\pi : S \to \mathcal{M}(\mathcal{A}). \tag{2.5}$$

A deterministic policy is a policy that defines a single action per state. We overload the notation $\pi(S) \in \mathcal{A}$ for deterministic policies.

The agent interacts with the environment and takes actions according to the policy. The value function of the policy is defined to be the expectation of the return given that the agent acts according to that policy:

$$V^\pi(s) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{A_t \sim \pi(S_t) \\ S_{t+1} \sim T(S_t, A_t) \\ R_t \sim R(S_t, A_t)}} \left[ \sum_{t=0}^{\infty} \gamma^t R_t \middle| S_0 = s \right]. \tag{2.6}$$

Note that since $R_t \in [0, R_{\max}]$, we must have $V^\pi(s) \leq V_{\max} \overset{\text{def}}{=} R_{\max}/(1-\gamma)$. Using the linearity of the expectation, we can write the above expression in a recursive form, known as the *Bellman equation* (Bellman, 1957):

$$\forall s \in \mathcal{S} : V^\pi(s) = \mathop{\mathbb{E}}_{A \sim \pi(s)} \left[ \bar{R}(s, A) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,A)} [V^\pi(S')] \right]. \tag{2.7}$$

One can use sums (if states and/or actions are finite) or integrals (for infinite set of states and actions) to write the Bellman equation.

The right hand side of 2.6 can be written down in terms of the *Bellman operator*. The operator $\mathbb{T}^\pi$ applied to a function $V(\cdot)$ is defined as:

$$(\mathbb{T}^\pi V)(s) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{A\sim\pi(s)}\left[\bar{R}(s,A) + \gamma \mathop{\mathbb{E}}_{S'\sim T(s,A)}[V(S')]\right]. \tag{2.8}$$

The Bellman equation can then be written in terms of the Bellman operator as:

$$\forall s \in \mathcal{S} : V^\pi(s) = (\mathbb{T}^\pi V^\pi)(s). \tag{2.9}$$

Given an estimate of the value function $\hat{V}^\pi$ one can calculate the amount of deviation from the Bellman equation. This is often referred to as the Bellman error:

$$e_{\hat{V}^\pi}(s) \overset{\text{def}}{=} (\mathbb{T}^\pi \hat{V}^\pi)(s) - \hat{V}^\pi(s). \tag{2.10}$$

The value function has been used as the primary measure of performance in much of the RL literature. There are, however, some ideas that take the risk or the variance of the return into account as a measure of optimality (Heger, 1994; Sato and Kobayashi, 2000). But the more common criteria is to assume that the agent is trying find a policy that maximizes the value function. Such a policy is often referred to as an optimal policy.

We can also define the value function over the state–action pairs. This is usually referred to as the action-value function, $Q$-function, or $Q$-value, of that pair. By definition:

$$Q^\pi(s,a) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{t\geq 1: A_t\sim\pi(S_t)\\ S_{t+1}\sim T(S_t,A_t)\\ R_t\sim R(S_t,A_t)}}\left[\sum_{t=0}^{\infty}\gamma^t R_t \middle| S_0 = s, A_0 = a\right]. \tag{2.11}$$

That is, the $Q$-value is the expectation of the return, given that the agent takes action $a$ first and then follows policy $\pi$. It is often used to compare the optimality of actions given a fixed policy. The $Q$-function also satisfies the Bellman equation:

$$\forall s \in \mathcal{S}, a \in \mathcal{A} : Q^\pi(s,a) = \bar{R}(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,a)} \left[ \mathop{\mathbb{E}}_{A' \sim \pi(S')} \left[ Q^\pi(S', A') \right] \right]. \tag{2.12}$$

Similar to the value function, we can define a Bellman operator for the action-value function. Overloading the notation, the action-value Bellman operator $\mathbb{T}^\pi$ applied on a function $Q(.,.)$ is defined as:

$$(\mathbb{T}^\pi Q)(s,a) \overset{\text{def}}{=} \bar{R}(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,a)} \left[ \mathop{\mathbb{E}}_{A' \sim \pi(S')} \left[ Q(S', A') \right] \right], \tag{2.13}$$

and the action-value Bellman equation will be:

$$\forall s \in \mathcal{S}, a \in \mathcal{A} : Q^\pi(s,a) = (\mathbb{T}^\pi Q^\pi)(s,a). \tag{2.14}$$

The action value Bellman error is defined similarly:

$$e_{\hat{Q}^\pi}(s,a) \overset{\text{def}}{=} (\mathbb{T}^\pi \hat{Q}^\pi)(s,a) - \hat{Q}^\pi(s,a). \tag{2.15}$$

## 2.4 Policy Optimality

An optimal policy, denoted by $\pi^*$, is defined to be a policy that maximizes the value function at the initial state. If $\Pi$ is the set of all policies, we have:

$$\mathop{\mathbb{E}}_{S_0 \sim T_0} \left[ V^{\pi^*}(S_0) \right] = \sup_{\pi \in \Pi} \mathop{\mathbb{E}}_{S_0 \sim T_0} \left[ V^\pi(S_0) \right]. \tag{2.16}$$

It has been shown (Bellman, 1957) that for any MDP, there exists an optimal deterministic policy that is no worse than any other policy for that MDP. The optimal value function $V^*$ satisfies the *Bellman optimality equation* (Bellman, 1957):

$$V^*(s) = \sup_{a \in \mathcal{A}} \left[ \bar{R}(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,a)} \left[ V^*(S') \right] \right]. \tag{2.17}$$

One can rewrite the above in terms of the *Bellman optimality operator* $\mathbb{T}^*$ applied to a function $V(\cdot)$ is defined as:

$$(\mathbb{T}^* V)(s) \overset{\text{def}}{=} \sup_{a \in \mathcal{A}} \left[ \bar{R}(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,a)} \left[ V(S') \right] \right]. \tag{2.18}$$

Using this, the Bellman optimality equation can be written as:

$$\forall s \in \mathcal{S} : V^*(s) = (\mathbb{T}^* V^*)(s). \tag{2.19}$$

We can similarly define the Bellman optimality operator and the corresponding Bellman equations for the action-value function. Overloading the notation, the Bellman operator $\mathbb{T}^*$ applied to an action-value function $Q$ is defined as:

$$(\mathbb{T}^* Q)(s,a) = \bar{R}(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim T(s,a)} \left[ \sup_{a' \in \mathcal{A}} Q(S', a') \right], \tag{2.20}$$

and the Bellman optimality equation for optimal action-value function $Q^*$ becomes:

$$\forall s \in \mathcal{S}, a \in a : Q^*(s,a) = (\mathbb{T}^* Q^*)(s,a). \tag{2.21}$$

We define the *regret* of a policy $\pi$ on state $s$ to be the difference of its value and the optimal value function at that state: $V^*(s) - V^\pi(s)$. Similarly, for a state–action pair, the regret is defined to be the difference of the action value of a policy and the optimal action value function: $Q^*(s,a) - Q^\pi(s,a)$. Note that the regret might be defined differently in other related works (Szepesvari, 2010).

## 2.5 Representation of the Value Function

With small and finite state-spaces, one can use a tabular representation of the value function (i.e. representing each state-value by a number). With large or infinite state-spaces, one has to use function approximation to represent the value function. In this work we only work with parametric function approximations. The value function is thus defined in terms of a parameter vector $\theta$.

The state of the MDP is also often represented in terms of *features* when working with a continues-state-space. We thus represent the state features by a vector of values: $\mathbf{x}_t$ represents the vector of features associated with state of state at time $t$. For continues MDPs, we use $\mathbf{x}_t$ to directly represent the state of the MDP. We use $S_t$ and $\mathbf{x}_t$ interchangeably.

Linear value function approximation is among the most commonly used methods in RL. With parameter vector $\theta$, the value of state at time $t$ is approximated by:

$$V_\theta(\mathbf{x}_t) = \theta^T \mathbf{x}_t. \tag{2.22}$$

By designing a good feature vector and tuning the value of $\theta$, one can create a good estimator of the value function. In the next chapter, we discuss some of the algorithms developed to find a good set of features and parameters.

## 2.6 Dynamic Programming with Exact Models

There are many methods developed to find the value of a policy. These methods are referred to as *policy evaluation* algorithms. If the state-space is small and the model of the MDP is fully known, the simplest way to find the value function is by solving the Bellman equation using matrix inversion (Sutton and Barto, 1998). This method, however, becomes impractical with larger state-spaces due to computational cost and numerical instability of direct matrix inversion.

### 2.6.1 Value Iteration

There are dynamic programming methods (Bellman, 1957) to find the value function by iteratively applying the Bellman operator to the current estimate of the value function. We start with some initial estimate $V_0^\pi$ of the value function and then keep updating it until convergence is observed:

$$V_{t+1}^\pi \leftarrow \mathbb{T}^\pi V_t^\pi, \tag{2.23}$$

where $V_t^\pi$ is our estimate of the value function at time $t$. The series of value estimates converge to the correct value, since the Bellman operator is a $\gamma$-contraction with respect to the infinity norm (Williams and Baird, 1993):

$$\|\mathbb{T}^\pi V - \mathbb{T}^\pi V'\|_\infty \leq \gamma \|V - V'\|_\infty \,, \tag{2.24}$$

which entails:

$$\|V_k^\pi - V^\pi\|_\infty \leq \gamma^k \|V_0^\pi - V^\pi\|_\infty \,, \tag{2.25}$$

Similar contraction properties also hold for the action-value function:

$$\left\|\mathbb{T}^{\pi} Q - \mathbb{T}^{\pi} Q'\right\|_{\infty} \leq \gamma \left\|Q - Q'\right\|_{\infty}, \tag{2.26}$$

With value function approximation, direct Bellman backups are not always possible since the function might not be representable in the function space spanned by the approximation. We thus need to apply a projection operator after applying the Bellman operator:

$$V^{\pi}_{t+1} \leftarrow \Pi^{\mathcal{F}}_{\nu(\pi)} \mathbb{T}^{\pi} V^{\pi}_{t}, \tag{2.27}$$

where $\mathcal{F}$ is the function approximation space and $\nu(\pi)$ is the stationary distribution of states under policy $\pi$.

When aiming to find the optimal value function, one can use the optimal Bellman operator in the above equations and obtain a fixed-point using dynamic programming. This algorithm is referred to as *Value Iteration*:

$$V^{*}_{t+1} \leftarrow \mathbb{T}^{*} V^{*}_{t}. \tag{2.28}$$

The above sequence can be shown to converge to $V^*$ with a similar $\gamma$-contraction in the error (Sutton and Barto, 1998). When using function approximation, one has to use a projection into the approximation hypothesis space at each iteration.

### 2.6.2 Policy Iteration

Instead of directly working with the value function, one can try to find an optimal policy for the MDP. *Policy Iteration* (PI) is an RL algorithm that aims to gradually improve the behaviour policy with each iteration (Sutton and Barto, 1998). The algorithm starts with an initial policy $\pi_0$ and updates the policy to be greedy with respect to the estimated value of the previous iteration:

$$\pi_{k+1} \leftarrow \text{Greedy with respect to } Q^{\pi_k} \tag{2.29}$$

It can be shown that with exact backups, the policy is improved at every step and that its value converges to the optimal value function (Sutton and Barto, 1998).

# CHAPTER 3
# Reinforcement Learning Algorithms

In the general RL problem, unlike the cases discussed in the previous chapter, we do not have exact knowledge of the system dynamics or the reward model. Therefore, the agent *learns* to act optimally by interacting with the environment and correcting the estimates and the policy, based on the observed trajectories. There is a large literature on computing of the optimal value and an optimal policy for an unknown MDP. This chapter covers some of these algorithms most relevant to the topics of this thesis.

## 3.1 Categories of RL Problems and Algorithms

When interacting with an unknown MDP, there are two approaches to solving for an optimal policy. One can interact with the MDP and use the observations to estimate the transition and reward model (often referred to *model-based RL*) or otherwise directly estimate the value of a given policy or the optimal value function (*model-free RL*).

RL problems can be broadly categorized into two types. In the *batch RL problem*, we are given a fixed set of trajectories sampled on an unknown MDP under some predefined behaviour policy. In the *online RL problem*, we get to choose the behaviour policy as trajectories are collected, hence acting "online". The online RL problem has to deal with exploration (choosing the correct policy that allows us to better know the underlying model). The batch problem, on the other hand, has to deal with *off-policy* evaluation, where one would need to evaluate a policy given trajectories collected on another policy (Szepesvari, 2010). In this thesis, we focus mostly on the batch learning problem.

There are two broad categories of RL algorithm and methods. *Online RL algorithms* gradually change the estimates of the model parameters or values, using constant time update rules at each iteration. *Batch RL algorithms*, on the other hand, apply batch updates to the estimates, using many collected trajectories at once (Szepesvari, 2010).

Note that batch algorithms can be applied in online problems (using mini-batch techniques), and online algorithms can be applied in batch problems (for smaller computational or memory complexity).

Most of the contributions of this thesis are on batch algorithms, but we briefly discuss the online adaptations when possible.

## 3.2 Value Prediction

When the MDP transition and reward models are fully known, one can directly calculate and the value function using dynamic programming. Though when the model of MDP is not known, we have to interact with the MDP to gather trajectories and estimate the value function based on finite amount of observations.

### 3.2.1 Temporal Differences with Fixed Point Solutions

With unknown models, the simplest way to approximate the value of a state under a given policy, is to sample several trajectories starting with that state and averaging the returns (an instance of the *Monte-Carlo* method). However, such estimator has a large variance, is not data-efficient and requires a reset operator, often not available for the problem at hand.

An alternative is to use *Temporal Difference* (TD) errors as a proxy to the Bellman error. The TD-error of an estimated value function $\hat{V}_t$ after observing the $t$-th transition and reward is defined as:

$$\delta_t = R_t + \gamma \hat{V}_t(S_{t+1}) - \hat{V}_t(S_t). \tag{3.1}$$

It is easy to show that the expectation of the temporal difference at $S_t$ equals the Bellman error at that point (Sutton and Barto, 1998):

$$\mathop{\mathbb{E}}_{\substack{A_t \sim \pi(S_t) \\ S_{t+1} \sim T(S_t, A_t) \\ R_t \sim R(S_t, A_t)}} [\delta_t] = e_{\hat{V}_t}(S_t). \tag{3.2}$$

TD-errors are thus proxies to estimating the Bellman error.

One can thus try to minimize the overall TD-errors across observed trajectories. *TD-learning* algorithms are methods that gradually change the estimated value function to reduce the observed TD-error.

In a small and finite state-space, one can represent the value function with a tabular representation. The tabular TD(0) algorithm is then a simple update to one of the elements of the table at each time point:

$$\hat{V}_{t+1}(s) = \hat{V}_t(s) + \alpha_t \delta_t \mathbb{I}_{\{S_t = s\}}, \tag{3.3}$$

were $\alpha_t$ is the step-size for the update at time $t$. The above is shown to converge to the fixed point of the Bellman operator and hence the correct value function.

With value function approximation, one can similarly update the parameter of the value function with an increment:

$$\begin{aligned} \delta_t &= R_t + \gamma V_{\theta_t}(\mathbf{x}_{t+1}) - V_{\theta_t}(\mathbf{x}_t), && (3.4) \\ \theta_{t+1} &= \theta_t + \alpha_t \delta_t \nabla_\theta V_{\theta_t}(\mathbf{x}_t). && (3.5) \end{aligned}$$

TD($\lambda$) (Sutton and Barto, 1998) is a generalization of TD(0) that uses *eligibility traces* to make increments to the value function. The choice of $\lambda \in [0, 1]$ provides an interpolation between TD(0) and the Monte-Carlo update. One has to be careful using TD($\lambda$) with function approximations when off-policy samples are used. Unlike the tabular case, convergence is no longer guaranteed (Bertsekas and Tsitsiklis, 1996).

### 3.2.2 Gradient Temporal Difference Methods

One can avoid the divergence issues of TD($\lambda$) by directly minimizing the *projected Bellman loss* function:

$$J^{\pi}_{\mathrm{PB}}(\theta) = \left\| V_\theta - \Pi^{\mathcal{F}}_{\nu} \mathbb{T}^{\pi} V_\theta \right\|^2_{\nu}, \qquad (3.6)$$

were $\nu$ can be different from the stationary distribution of $\pi$.

For linear value function approximation, algorithms GTD2 (Gradient Temporal Difference learning, version 2) and TDC (Temporal Difference learning with Corrections) use stochastic gradient descent to optimize the above loss function (Sutton et al., 2009). They both iteratively update two sequences of parameters to obtain an unbiased estimate of the gradient of the objective 3.6.

GTD2 uses the Least-Mean Square (LMS) rule to update one of the sequences and TDC uses the technique of two-timescale stochastic approximation (Borkar, 1997). These methods have been extended to non-linear function approximation(Maei et al., 2009) and have also been modified to use eligibility traces (Maei and Sutton, 2010).

### 3.2.3 Batch Methods

One can get better sample efficiency and stronger convergence guarantees with batch offline methods. These methods aim to minimize a loss function over the entire set of sampled trajectories instead of using gradual updates to the estimated values.

Bellman Residual Minimization (BRM) aims to minimize the overall Bellman error (often weighted by the stationary distribution) (Sutton and Barto, 1998):

$$J^{\pi}_{\text{BR}}(\theta) = \left\| Q_\theta - \mathbb{T}^\pi Q_\theta \right\|^2_{\nu(\pi)} \,, \tag{3.7}$$

were $\nu(\pi)$ is the stationary distribution imposed by the policy over state–action pairs. The problem with this loss function is that its direct empirical estimate is biased:

$$\begin{aligned} &\mathop{\mathbb{E}}_{\substack{S'\sim T(s,a)\\ R\sim R(s,a)\\ A'\sim\pi(S_t)}} \left[ \{\hat{Q}(s,a) - (R + \gamma \hat{Q}(S',A'))\}^2 \right] \\ &\qquad = \{\hat{Q}(s,a) - (\mathbb{T}^\pi \hat{Q})(s,a)\}^2 + \mathop{\mathbb{V}\mathrm{ar}}_{\substack{S'\sim T(s,a)\\ R\sim R(s,a)\\ A'\sim\pi(S_t)}} \left[ R + \gamma \hat{Q}(S',A') \right]. \end{aligned} \tag{3.8}$$

To get an unbiased estimator, one might need to use the *double-sampling* method (Sutton and Barto, 1998), or use an auxiliary function to cancel the variance term (Antos et al., 2008).

An alternative is to minimize the empirical estimate to the projected Bellman loss defined in Equation 3.6. With linear function approximation, the minimization turns into a least-squares problem, referred to as Least-Squares Temporal Difference (LSTD) algorithm (Boyan, 2002; Lagoudakis and Parr, 2003).

Define the empirical operators:

$$\mathbb{S}_n V \overset{\text{def}}{=} [V(S_1), V(S_2), \ldots, V(S_n)]^T, \tag{3.9}$$

$$\mathbb{T}_n V \overset{\text{def}}{=} [R_1 + \gamma V(S_2), R_2 + \gamma V(S_3), \ldots, R_n + \gamma V(S_{n+1})]^T. \tag{3.10}$$

Let $\Pi^{\mathcal{F}}$ be the projection onto the space of parametrized value functions with respect to the $\ell^2$ norm $\|.\|$. LSTD minimizes the empirical version of the projected Bellman loss:

$$\hat{J}_{\text{PB}}(\theta) = \left\| \mathbb{S}_n V_\theta - \Pi^{\mathcal{F}} \mathbb{T}_n V_\theta \right\|^2. \tag{3.11}$$

This is an empirical approximation to the projected Bellman loss defined in Equation 3.6. For linear function approximation, the solution can be obtained by solving a set of linear equations:

$$\sum_{t=1}^{n} \mathbf{x}_t \delta_t = \mathbf{0}, \tag{3.12}$$

were $\mathbf{x}_t$ is the feature vector of the state at time $t$.

The sample complexity of the LSTD has been analysed with respect to the empirical and stationary norms (with fast-mixing MDPs). The bounds show that LSTD is sample-efficient with a typical bias–variance trade-off with respect to the complexity of the function approximator (Lazaric et al., 2010). The computational cost, on the other hand, can become prohibitive with large numbers of samples and features.

### 3.2.4 Regularized Least-Squares Methods

Direct regularization has been used for general complexity reduction in RL without domain-specific knowledge. One can add regularization terms to the Bellman loss functions in Equations 3.7 and 3.6 to control for the complexity of the value function fit. $\ell^2$ regularization can been used to enforce smoothness on the value function. Sample complexity bounds show how to achieve optimal rates of convergence (Farahmand et al., 2009b,a).

$\ell^1$ regularization have also been studied in use with the LSTD algorithm. The problem with $\ell^1$ regularization is that the resulting objective function is not convex and thus non-trivial to optimize. Kolter and Ng (2009a) applied an adaption of Least Angle Regression (LARS) method to find the solution to the regularized LSTD. The solution is sparse and one can thus build the LSTD solution only on the active features. Therefore their method can achieve a potentially large computation gain by inverting a much smaller matrix. The sample complexity is also reduced as one only needs to estimate the weights for the active features.

## 3.3 Control

In the online RL scenario, one needs to *control* the behaviour policy and gradually improve it to form an optimal policy. Having observed a history of transitions, one can either act greedily according to an estimated model (often referred to as *exploitation*) or otherwise try actions with more uncertainty about them to gain information and later improve the estimates (referred to as *exploration*). The exploration–exploitation trade-off is among the main problems in online RL and have been studies in various scenarios (Szepesvari, 2010).

This section covers the main categories of control algorithms used in online RL scenario. We include the algorithms mostly discussed in the literature and more relevant to the topics covered in this thesis.

### 3.3.1 Q-Learning

The Value Iteration algorithm described in Section 2.6.1, requires full knowledge of the underlying MDP. When faced with an unknown MDP, one can approximate the application of the Bellman operator with a TD error update rule. This algorithm is referred to as Q-learning (Watkins, 1989) and can be implemented in various ways.

Q-learning in finite MDPs is an online constant time update rule. After observing the tuple $(S_t, A_t, R_t, S_{t+1})$, we update the estimate of the action value function as follows:

$$\delta_t^* = R_t + \gamma \max_{a' \in \mathcal{A}} Q_t(S_{t+1}, a') - Q_t(S_t, A_t), \tag{3.13}$$

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t \delta_t^* \mathbb{I}_{\{S_t = s, A_t = a\}}. \tag{3.14}$$

One can show that the expectation of the TD-error $\delta_t$ defined above is the optimal Bellman error on the observed state–action pair:

$$\mathop{\mathbb{E}}_{\substack{S_{t+1}\sim T(s,a)\\ R_t\sim R(s,a)}}\left[\delta_t^*|S_t = s, A_t = a\right] = (\mathbb{T}^* Q_t)(s,a) - Q_t(s,a). \tag{3.15}$$

The policy that is followed by the Q-learning algorithm is often an $\epsilon$-*greedy* or a *Boltzmann action-selection* method to allow for some exploration and achieve consistency (Szepesvari, 2010). More systematic action selection methods are discussed in the following sections.

With large or continuous state-spaces, one has to use function approximation to represent the value function. With parametrized value function approximation, the update rule for Q-learning will be as follows (Szepesvari, 2010):

$$\delta_t^* = R_t + \gamma \max_{a'\in\mathcal{A}} Q_{\theta_t}(S_{t+1}, a') - Q_{\theta_t}(S_t, A_t), \tag{3.16}$$

$$\theta_{t+1} = \theta_t + \alpha_t \delta_t^* \nabla_\theta Q_{\theta_t}(S_t, A_t). \tag{3.17}$$

Melo et al. (2008) have shown that the above update rules converge under certain conditions for the sampling distribution. There are also gradient type algorithms similar to the stochastic gradient TD algorithms that have recently been introduced for Q-learning (Maei et al., 2010).

Fitted Q-iteration is another algorithm that approximates (fitted) value iteration. It follows by generating a batch of trajectories on a policy and then approximating the optimal Bellman operation by using generic function approximation methods. Tree-based approximation (Ernst et al., 2005) and kernel averaging (Ormoneit and Sen, 2002) are among the methods used for fitted Q-iteration.

### 3.3.2 Actor Critic Methods

With a known MDP model, the Policy Iteration algorithm can be used to gradually update a policy until an optimal policy is found. When the transition and reward models are not known, one has to use approximate policy evaluation methods when evaluating intermediate policies. Such algorithms are referred to as Generalized Policy Iteration (GPI) (Szepesvari, 2010).

A more general framework, referred to as *actor–critic methods*, separate the notion of an actor (the behaviour policy) and the critic (the policy evaluation routines). The actor of these models are often stochastic policies and the critics can be based on any of the policy evaluation methods discussed in the previous sections.

#### Critic Algorithms

With finite states and actions one can use TD-type methods as critics. SARSA is a critic method based on the TD-learning algorithms. It is essentially TD(0) applied to state–action value functions:

$$\delta_t = R_t + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t), \tag{3.18}$$

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t \delta_t \mathbb{I}_{\{S_t=s, A_t=a\}}. \tag{3.19}$$

The convergence guarantees of SARSA are similar to that of TD-algorithms. There have been extensions to use eligibility traces with SARSA($\lambda$) (Rummery and

Niranjan, 1994) and also extensions using stochastic gradient techniques (Maei and Sutton, 2010).

Batch method have also been used as critics. LSTD-Q and LSTD-Q($\lambda$) are generalizations of LSTD for state–action value functions. These methods can use features of the states and are thus useful with large or continuous state and action spaces.

**Actor Algorithms**

The actor part of an actor–critic method can be designed in many different ways. The choice of the behaviour policy can affect the convergence properties of the algorithm.

The actor can be defined to be greedy with respect to the evaluation by the critic. This can be done either in (mini-)batch mode or in online mode. In batch mode, the actor fixes the behaviour policy at each iteration and collects a number of samples. The critic then uses a batch policy evaluation method to evaluate the policy. Least-Squares Policy Iteration (LSPI), is one such algorithm that uses LSTD-Q(0) as critic (Lagoudakis and Parr, 2003).

Greedy online methods interleave the update on the actor and critic parts instead of using mini-batches. SARSA(0) is among such algorithm with asymptotic consistency guarantees under *infinite exploration* (Singh et al., 2000).

$\epsilon$-greedy and Boltzmann-type stochastic greedy policies are more commonly used as critics as they provide better convergence. They effectively result in a smoother update to the behaviour policy and are thus less likely to get stuck in a local optima (Szepesvari, 2010).

*Policy gradient* methods is another class of critics that use stochastic gradient ascent on the value of policies in a parametrized policy class. Common choices for the policy class are *Gibbs* policies for finite action spaces and *Gaussian* policies for continuous action spaces in $\mathbb{R}^n$ (Szepesvari, 2010).

The performance of a policy is often measured by the expected return of the policy starting from some initial distribution on the state. There are many methods to estimating the gradient of performance with respect to policy parameters. One can use finite differences, for instance, to approximate the gradient of the expected return. A more common approach is to use *likelihood ratio* policy gradient (in methods such as REINFORCE (Glynn, 1987)) or importance sampling techniques (Jie and Abbeel, 2010). These methods use importance weighing techniques to estimate the gradient of the expected return using the gradient of the log action probabilities.

### 3.3.3 Systematic Exploration with Regret Bounds

While approximations of Value Iteration and Policy Iteration have been used successfully in RL control problem, optimal exploration is still an open problem. Most of the results with Q-learning and actor–critic type algorithms can only provide convergence guarantees in the limit of infinite samples and under strict sampling conditions.

Systematic exploration methods focus on fining solutions with finite sample guarantees on a measure of optimality on the resulting policy. A more thorough specification of the target and the analysis of such algorithms are provided in Chapter 4.

## 3.4 Feature Section and Extraction in RL

When dealing with an unknown state-space with a large number of features, even linear function approximation might result in high variance in the value function estimation. This is specially the case, when the number of possible features is larger than the number of trajectories provided to the RL algorithm (in the online case this also happens in the earlier learning stages). General forms of regularization in function approximation can be used to remedy the problem. In particular, we can build the value function based on a subset of features. One thus needs to choose the correct feature set such that the model based on that would be as close to the true value function as possible.

Feature extraction methods have specially been used in the RL literature to achieve concise value models that match the observed transitions. Menache et al. (2005) introduced two algorithms to adapt basis functions as features for linear function approximation in RL. Keller et al. (2006) applied neighbourhood component analysis as a dimensionality reduction technique to construct a low dimensional state-space based on the temporal difference error. In their work, they iteratively add feature that would help predict the Bellman error. Parr et al. (2007) later showed that any feature extraction method based on a *Bellman-error-based* approach with certain conditions will provably tighten approximation error bounds.

Online feature extraction methods have also been studied in the RL literature. Geramifard et al. (2011) have recently introduced the *incremental Feature Dependency Discovery* (iFDD) as a fast online algorithm to extract non-linear binary feature for linear function approximation. In their work, one keeps a list of candidate features (non-linear combination of two active features), and among these adds the features that correlates the most with the temporal-difference error.

Agnostic feature selection methods have only recently been studied in the RL domain with almost no empirical analysis. Ghavamzadeh et al. (2010) used random projections to reduce the dimensionality of the state-space in cases where the trajectory batch is smaller than the number of features. They provide performance bounds for both LSTD and LSPI with random projections. An important contribution of this thesis (see Chapter 9) connects agnostic feature selection methods from the compressed sensing literature to Bellman-error-based methods from the RL domain.

## 3.5 Bayesian RL

Compared to pure frequentist methods, Bayesian analysis provides better sample efficiency and faster convergence rates when we have access to an informative prior distribution over the parameters of the solution. Bayesian methods have been applied to both model-based and model-free RL. For the model-based approaches, one is given a prior distribution over the transition and reward models. This is mostly studied in finite state-spaces, where the transition model is defined by a set of multinomial distributions. Starting from state $s$, for any action $a$, the probability of arriving at any next state is defined by a multinomial distribution. One can then use Dirichlet distributions as the conjugate prior over the parameters of these multinomial distributions (Strens, 2000; Duff, 2002; Wang et al., 2005). In the batch RL setting, one can marginalize the prior over the data to get a posterior. This is a relatively easy task, as the update can be done by incrementing the counts of the Dirichlet distributions associated with the observed transitions.

In the online setting, unlike batch RL, the learning method has some control over how the observed trajectories are collected, and hence has to deal with the exploration–exploitation trade-off. Bayesian RL aims to solve this problem by planing in the state-space extended with the Dirichlet counts. Assuming that the prior distribution incorporates the true probabilistic nature of the RL domain, the optimal policy suggested by the partially observable RL problem on the extended state-space is the optimal plan that correctly balances exploration and exploitation (Duff, 2002).

Bayesian analysis has also been applied to model-free RL, where one is given a prior over the value function. This is often achieved by means of Gaussian processes. GPTD (Engel et al., 2003) and GPSARSA (Engel et al., 2005) are examples of such methods, extending the TD and SARSA methods discussed in previous sections. The use of Gaussian processes often results in computationally intractable planning problems due to the large kernel matrices that grow with the sample size. Therefore approximations have been used to achieve more practical inference algorithms by pruning points from the sample (Engel et al., 2003).

Bayesian methods, although elegant and concrete, have often been criticized not only for their computational cost, but also for their strong assumptions on the correctness of the prior distribution. There are usually no theoretical guarantees when performing Bayesian inference with priors that do not admit the correct posterior, meaning that they can converge to suboptimal solutions (Kolter and Ng, 2009b). In Chapter 6, we argue here that a more adaptive method can provide stronger guarantees on the performance and at the same time be more data efficient if an informative prior is taken into account.

# Part II

# PAC-Bayesian Reinforcement Learning

# CHAPTER 4
# PAC Learning

Learning can be thought of as the process of acquiring the skill to perform a task in the absence of explicit programming (Valiant, 1984). Since the invention of computing machines, the idea of *machine learning* has been receiving attention from different scientific communities. Just as the computational phenomenon had to be described in precise models for its theory to be developed, machine learning also needs concrete mathematical models to be crafted as a theory. Such theory can help explain the human's learning experience and let us build algorithms and machines that can learn. The theory should also shed light on the limits of what can be learned, just as computability does on what can be computed (Valiant, 1984).

Concrete notions of learnability in computational learning theory dates back to the well known work of Valiant on the *theory of the learnable* (Valiant, 1984). His work led to a well studied branch of machine learning, known as Probably Approximately Correct (PAC) learning. Under the theory of PAC learnability, different classes of learning problems are studied to find out whether or not they can be learned (efficiently) with arbitrary accuracy with high probability. Blumer et al. (1989) extended Viliant's definitions to a more general class of learning problems, and showed that a necessary and sufficient condition of PAC learnability is the finiteness of Vapnik–Chervonenkis (VC) dimension. They provided upper and lower bounds on the sample complexity of learning problems as functions of the VC dimension of the hypothesis class.

## 4.1 Theory of the PAC Learnable for Finite Hypothesis Spaces

A theory of learning is a starting point to understand human's learning behavior and is essential to building devices that can learn automatically from data. Valiant was the first to address this by introducing a theory of the learnable (Valiant, 1984). He defines a learning machine to consist of a *learning protocol* along with a *deduction procedure*. The learning protocol specifies how the information is collected form outside, and the deduction mechanism is the means by which a recognition algorithm of the target concept is deduced.

Valiant was the first to introduce PAC learning as a formal model of machine learning. His work was on the special case of Boolean input spaces. We will start by the original definitions of PAC on Boolean functions and then study the natural extensions to real-valued state-spaces. The basic PAC model is the learning task aiming to find a specific Boolean function, called a *target concept* $c : \{0,1\}^m \to \{0,1\}$ within a *concept class* C. We have access to a training set $\mathcal{D}$ that includes instances in the form $(X, c(X))$, with $X$ drawn from an unknown arbitrary fixed distribution $D$ over $\{0,1\}^m$. The algorithm then chooses a *hypothesis* $h : \{0,1\}^m \to \{0,1\}$ from a *hypothesis space* $H$ as the output of the learning process. The *true error rate* of a hypothesis $h$ under the distribution $D$ is the probability that an example $X$ drawn from $D$ results in different output from $h$ and $c$:

$$J_D(h) \overset{\text{def}}{=} \underset{X\sim D}{\mathbb{P}\mathrm{r}} \{h(X) \neq c(X)\}\,. \tag{4.1}$$

We can now define PAC learnable concept classes (Valiant, 1984):

**Definition 1.** *A concept class $\mathcal{C}$ of $m$ Boolean variables is **PAC learnable by learning algorithm** $A$ using a hypothesis space $H$, if for all target concepts $c \in C$, all probability distributions $D$ over $\{0,1\}^m$, and all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, the learning algorithm $A$ will produce with probability no less than $1 - \delta$, a hypothesis $h \in H$ such that $J_D(h) < \epsilon$, requiring a running time polynomial in $m$, $1/\epsilon$ and $1/\delta$.*

**Definition 2.** *A concept class $\mathcal{C}$ of $m$ Boolean variables is **PAC learnable** if there exists a learning algorithm $A$ and a hypothesis space $H$ such that $\mathcal{C}$ is PAC learnable with $A$ using $H$.*

Such learning algorithm is often referred to as a uniform learning algorithm, as the guarantee is independent from $D$. The requirement of polynomial running time is sometimes referred to as *efficient PAC learnability*. However, we assume the efficiency to be part of the definition of PAC learning throughout this thesis.

Valiant proved PAC learnability for certain classes of Boolean expressions including *k-CNFs* and *monotone DNFs* (Valiant, 1984). Haussler (1988) extended these results to any finite hypothesis space. The results show that with enough number of samples, any hypothesis that is consistent with the observations is going to be $\epsilon$-optimal with high probability:

**Theorem 4.1 (Haussler, 1988).** *For given $\epsilon$ and $\delta$ of the PAC model, the minimum number of samples needed for learning an $\epsilon$-optimal solution with probability no less than $1 - \delta$ is upper bounded by:*

$$\frac{1}{\epsilon}(\ln |H| + ln(\frac{1}{\delta})). \tag{4.2}$$

The learning algorithm can be any algorithm that finds a hypothesis consistent with all the observations (referred to as a *consistent learner*). Therefore, if a polynomial consistent learner exists, and if $\ln|H|$ is polynomial in the number of variables $m$, then this class is (efficiently) PAC learnable.

There is a simple proof based on the fact that with enough number of samples, one would get a counter example to all the hypotheses with error rates more than $\epsilon$ with probability no less than $1-\delta$.

## 4.2 PAC Learnability and the VC-Dimension

The bounds based on the size of the hypothesis class are often loose and can only be used for finite concept classes. Blumer et al. (1989) extended the notion of learnability to general concept classes by connecting the sample complexity with Vapnik–Chervonenkis (VC) dimension. They showed that the necessary and sufficient condition of uniform learnability is the finiteness of the VC dimension.

The VC dimension, first introduced by Vapnik and Chervonenkis (1971), is a measure of complexity of a hypothesis class. It is also referred to as the *capacity* of a class of functions (Vapnik, 1982).

**Definition 3.** *A set $\mathcal{S}$ of examples is said to be shattered by a hypothesis class $H$, if for any subset $\mathcal{S}' \subseteq \mathcal{S}$, there exists a hypothesis $h \in H$, such that: $h(X) = 1 \Leftrightarrow X \in \mathcal{S}'$.*

The above definition indicates that for any labelling of $\mathcal{S}$, there exists a hypothesis in $H$ consistent with such labelling. Using this, we can define the Vapnik–Chervonenkis (VC) dimension (Vapnik and Chervonenkis, 1971):

**Definition 4.** *Let $X$ be an instance (input) space and $H$ a hypothesis space ($h : X \to \{0,1\}$ for $h \in H$). The Vapnik–Chervonenkis dimension of $H$, denoted by $d_{\mathrm{VC}}^{H}$, is the size of the largest finite subset of $X$ that can be shattered by $H$, or equals $\infty$ if the largest subset does not exist.*

The VC dimension is known for many of the commonly used hypothesis spaces, including linear and polynomial classifiers. For instance, the class of linear classifiers in $d$ dimensions has a VC dimension of $d+1$.

Blumer et al. (1989) give upper and lower bounds on the sample complexity of learning problems with finite $VC$ dimension: Any consistent learner with polynomially bounded sample size (linear in the VC dimension) will have an error rate less than $\epsilon$ with probability no less than $1-\delta$.

**Theorem 4.2 (Blumer et al., 1989).** *A concept class $\mathcal{C}$ is uniformly PAC learnable on the same hypothesis class $H = C$, if and only if $d_{\mathrm{VC}}^{H} < \infty$:*

*(a) For sample sizes greater than*

$$\max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8d_{\mathrm{VC}}^{H}}{\epsilon}\log\frac{13}{\epsilon}\right), \tag{4.3}$$

*any consistent hypothesis in $H$ has error rate less than $\epsilon$ with probability no less than $1-\delta$.*

*(b) for $0 < \epsilon < 1/2$ and sample size less than*

$$\max\left(\frac{1-\epsilon}{\epsilon}\ln\frac{1}{\delta}, d_{\mathrm{VC}}^{H}(1 - 2(\epsilon(1-\delta)+\delta))\right), \tag{4.4}$$

*no learning algorithm $A : X^m \to H$ can guarantee an error rate less than $\epsilon$ with probability no less than $1-\delta$.*

This theorem shows that depending on whether or not the VC dimension of a concept class $\mathcal{C}$ is finite, either $\mathcal{C}$ is uniformly learnable with sample size around $O(d/\epsilon \log(1/\delta))$ or $\mathcal{C}$ is not uniformly learnable at all. If $\mathcal{C}$ is learnable, then the necessary and sufficient sample sizes for PAC learnability are linear in the VC dimension.

VC bounds are often tighter than the bounds based on the size of the hypothesis space in the case of finite spaces. Intuitively, the VC dimension is a measure of complexity of the hypothesis space, often times correlated with the number of free parameters of the hypothesis. The above theorem has been applied to many classes of learning algorithms to bound their sample complexity and error rate. Other bounds have also been introduced that do not need the consistent learners (Angluin and Laird, 1988). This is specifically useful when $H$ is different from $\mathcal{C}$ and does not include any consistent classifier (e.g. when samples are drawn from a hypothesis in $H$ with some noise).

## 4.3 PAC Bounds for Reinforcement Learning

Efficient learnability can also be considered in control problems when the learning algorithm has to interact with the environment in an RL setting. Efficient (PAC) learnability in reinforcement learning was first addressed by Fiechter (1994). Before his work, all other learning methods on observable Markovian environments only provided asymptotic convergence guarantees with no bound on the sample complexity of the learning problem. Guarantees on the error (regret) rates are important if one wants to achieve efficiency in solving the RL problem.

Fiechter's model of reinforcement learning is slightly different from the MDP model we defined in the previous chapters. He assumes the existence of a "reset" operation that would interrupt the current sequence of transitions and take the agent to a fixed initial state $s_0$. He defines PAC learnability in such environments to be the task of finding a policy (in time polynomial in $|\mathcal{S}|$, $|\mathcal{A}|$, $1/\epsilon$, $1/\delta$, $1/(1-\gamma)$ and $R_{max}$) whose value on the initial state $s_0$ is no more than $\epsilon$ away from the optimal value, with probability no less than $1-\delta$. If $\pi$ is the output of the algorithm, we must have with respect to randomness of the sample:

$$\mathbb{P}\mathrm{r}\left\{V^*(s_0) - V^\pi(s_0) \leq \epsilon\right\} \geq 1-\delta. \tag{4.5}$$

Fiechter (1994) loosely bounds the sampling complexity of the RL problem by a polynomial (pointing out that this is only a proof of the existence of a PAC learner). Kearns and Singh (2002) introduce the first practical PAC learner for the RL problem on general finite MDPs with the Explicit Explore or Exploid ($\mathrm{E}^3$) algorithm. Their model does not require a reset operation. Here, the PAC learning task is to explore the MDP for polynomial time and then halt in state $s$, having found a policy $\pi$ for which the value function is close enough to the optimal value with high probability:

$$\mathbb{P}\mathrm{r}\left\{V^*(s) - V^\pi(s) \leq \epsilon\right\} \geq 1-\delta. \tag{4.6}$$

The learnability guarantee for $\mathrm{E}^3$ are based on the fact that when we keep empirical estimates of the transition probabilities and the reward values, if these estimates are close enough to the true models, our estimate of the value function based on empirical models is also close to the true value function. Thus acting according to the optimal policy on the estimated model gives us near optimal performance on the true model. $E^3$ manages to use this by keeping track of “known” and “unknown” states. For the known states, we have observed enough samples so that the empirical transition and reward models are close approximations of the true ones, giving us near optimal policies on the known part of the state-space. The algorithm decides to either *exploit* (stay in the known part of the space and act near optimally) or *explore* (try to reach unknown states and improve the accuracy of the empirical estimates). By balancing this process, $E^3$ manages to PAC learn the RL problem in this setting.

The R-MAX algorithm, introduced by Brafman and Tennenholtz (2003), approaches the same problem by making the choice between exploration and exploitation implicit. R-MAX aims to solve the more general problem of learning in *stochastic games*, rather than MDPs. The main idea of R-MAX is that it assigns the maximum possible value to the states that are “unknown”, which in turn encourages the algorithm to visit those states if possible. Once the algorithm visits an unknown state enough times, then it becomes known and the empirical value (which is guaranteed to be close to the true value) is assigned to the state. This trick, sometimes referred to as being *optimistic in the face of uncertainty*, has been used in later works (Wiering and Schmidhuber, 1998; Strehl and Littman, 2004) as a means to encourage implicit exploration of unknown states. As the number of samples needed to mark a state as known is polynomial, there could be at most a polynomial number of exploration rounds, making the sample complexity polynomial [1] .

Instead of having known/unknown states, one can keep track of confidence bounds over the estimates of the transition and reward models. Wiering and Schmidhuber (1998) introduced the Model-Based Interval Estimation (MBIE) algorithm that uses the same idea of acting optimistically in the face of uncertainty. However, instead of assigning the maximum possible value to the unknown states, it keeps track of confidence intervals over the transitions and rewards, and then uses the most optimistic values within the confidence bounds (i.e. the one resulting in the highest return) as the estimate. This result in the following optimistic, optimal Bellman equation:

$$\tilde{V}^*(s) = \sup_{a \in \mathcal{A}} \left( \sup_{\tilde{R} \in \mathcal{CI}_{R(s,a)}} \tilde{R}(s,a) + \gamma \sup_{\tilde{T} \in \mathcal{CI}_{T(s,a)}} \mathop{\mathbb{E}}_{s' \sim \tilde{T}(s,a)} \left[ \tilde{V}^*(s') \right] \right), \tag{4.7}$$

where $\mathcal{CI}_T$ and $\mathcal{CI}_R$ are the transition and reward models consistent with the frequentist confidence intervals. For finite states and actions, the solution to the above equation can be found efficiently by solving a set of linear equations.

Strehl and Littman (2004) showed that MBIE empirically outperforms R-MAX (shows faster convergence to higher returns). They later provided theoretical proof (Strehl and Littman, 2005) that MBIE has polynomial sample complexity as is a PAC learning algorithm for the RL domain. They assume the PAC learning

[1] R-MAX uses a slightly different notion of PAC learnability, which make it depend on the $\epsilon$-*return mixing time* of the stochastic process.

to be the bounding of the number of suboptimal actions, rather than the time it takes to learn a near optimal solution. The following theorem shows the sample complexity for MBIE algorithm:

**Theorem 4.3 (Strehl and Littman, 2005).** *Let $\pi_t$ be the MBIE's policy at time $t$. With probability no less than $1-\delta$, the condition $V^{\pi_t}(s_t) \geq V^*(s_t) - \epsilon$ is true for all except for at most $O\left(\frac{|\mathcal{S}|^2|\mathcal{A}|R_{max}^5 \ln^3 \frac{|\mathcal{S}|.|\mathcal{A}|.R_{max}}{(1-\gamma)\epsilon\delta}}{(1-\gamma)^6\epsilon^3}\right)$ timesteps.*

The proof of MBIE's efficiency parallels the proof of R-MAX (there are notions of known/unknown states in the proof, but they do not appear in the algorithm itself).

The much simpler algorithm, Model Based Interval Estimation with Exploration Bonus (MBIE-EB), adds a simple bonus term to the value function to encourage exploration (Strehl and Littman, 2008). It still uses the idea of acting optimistically under uncertainly, but uses a bonus term rather that searching for the most optimistic models in the confidence intervals. The optimistic Bellman equation for MBIE-EB is as follows:

$$\tilde{V}^*(s) = \sup_{a\in\mathcal{A}} \left(\hat{R}(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim\hat{T}(s,a)} \left[\tilde{V}^*(s')\right] + \frac{\beta}{1+n(s,a)}\right), \tag{4.8}$$

where $\beta$ is a parameter of the algorithm, $\hat{R}$ and $\hat{T}$ are frequentist estimates, and $n(s,a)$ is the number of times action $a$ has been taken in state $s$. It can be shown that with a particular choice of $\beta$, the sample complexity of the MBIE-EB is asymptotically similar to MBIE (Strehl and Littman, 2008).

There are more recent PAC learners for the RL problems in different settings. Extensions to *factored MDPs* (Kearns and Koller, 1999), *metric MDPs* (Kakade et al., 2003) and *continuous MDPs* (Brunskill et al., 2008) have been developed. The study of these methods requires formalisms and discussions that are beyond the scope of this thesis.

## 4.4 Concentration Bounds for Fast-Mixing Markov Chains

In this section, we study the mixing condition on the Markov chain induced by a given fixed policy in an MDP. Specifically we assume that the Markov chain *uniformly quickly forgets its past*. For such chains, we present here an extension of Bernstein's inequality based on Samson (2000).

A *time-homogeneous Markov chain* is a sequence, $X_{1:n} \overset{\text{def}}{=} X_1, X_2, \ldots, X_n$, $X_i \in \mathcal{X}$, for some measurable space $\mathcal{X}$, with transition kernel $P(X) \in \mathcal{M}(\mathcal{X})$:

$$X_{i+1} \sim P(X_i). \tag{4.9}$$

Consider the concentration of the average of the Markov process:

$$(X_1, f(X_1)), \ldots, (X_n, f(X_n)), \tag{4.10}$$

where $f : \mathcal{X} \to [0, b]$ is a fixed measurable function. To arrive at a concentration inequality, we need a characterization of how fast $X_i$ forgets its past. Let $P^i(X) \in \mathcal{M}(\mathcal{X})$ be the $i$-step transition kernel:

$$X_{i+k} \sim P^i(X_k). \tag{4.11}$$

$P^i(X)$ can of course be defined in terms of $P(X)$. Define upper-triangular matrix $\mathbf{\Gamma}_n = (\gamma_{ij}) \in \mathbb{R}^{n \times n}$ as:

$$\gamma_{ij}^2 = \begin{cases} \sup_{(X,X') \in \mathcal{X}^2} \|P^{j-i}(X) - P^{j-i}(X')\|_{\mathrm{TV}} & \text{if } i < j \\ 1 & \text{if } i = j \\ 0 & \text{if } i > j \end{cases} \tag{4.12}$$

The operator norm of this matrix $\|\mathbf{\Gamma}_n\|$ w.r.t. the Euclidean distance, is a measure of dependence for the random sequence $X_1, X_2, \ldots, X_n$. For example with independent $X_i$'s, we have $\mathbf{\Gamma}_n = \mathbf{I}$ and $\|\mathbf{\Gamma}_n\| = 1$. In general $\|\mathbf{\Gamma}_n\|$, which appears in our concentration inequalities for dependent sequences, can grow with $n$.

We say that a time-homogeneous Markov chain *uniformly quickly forgets its past* if:

$$\tau = \sup_{n \geq 1} \|\mathbf{\Gamma}_n\|^2 < +\infty. \tag{4.13}$$

We refer to $\tau$ as the *forgetting time* of the chain. Conditions under which a Markov chain uniformly quickly forgets its past are of major interest (Farahmand and Szepesvári, 2011). Samson (2000) gives examples of such chains having *majorization condition*, referred to as *uniformly ergodic* (Meyn and Tweedie, 2009). Note that there are other cases when $\|\mathbf{\Gamma}_n\|$ is known to be independent of $n$. Most notable, this holds when the Markov chain is contracting.

The following result from Farahmand and Szepesvári (2011) is a trivial corollary of Theorem 2 of Samson (2000). Samson's result is stated for empirical processes and can be considered as a generalization of Talagrand's inequality to dependent random variables.

**Theorem 4.4 (Farahmand and Szepesvári, 2011).** *Let $f$ be a measurable function on $\mathcal{X}$ whose values lie in $[0,b]$, $X_{1:n} \sim C_n \in \mathcal{M}(\mathcal{X}^n)$ be a homogeneous Markov chain, taking values in $\mathcal{X}$ with forgetting time $\tau$. Let $Z = \frac{1}{n}\sum_{i=1}^{n} f(X_i)$. For all $\epsilon \geq 0$:*

$$\underset{X_{1:n}\sim C_n}{\mathbb{P}\mathrm{r}} \left\{ Z - \underset{X_{1:n}\sim C_n}{\mathbb{E}} [Z] \geq \epsilon \right\} \leq \exp\left( - \frac{\epsilon^2 \, n}{2b\tau(\underset{X_{1:n}\sim C_n}{\mathbb{E}} [Z] + \epsilon)} \right),$$

$$\underset{X_{1:n}\sim C_n}{\mathbb{P}\mathrm{r}} \left\{ \underset{X_{1:n}\sim C_n}{\mathbb{E}} [Z] - Z \geq \epsilon \right\} \leq \exp\left( - \frac{\epsilon^2 \, n}{2b\tau \underset{X_{1:n}\sim C_n}{\mathbb{E}} [Z]} \right).$$

These bounds are similar to the concentration bounds with i.i.d. data, but the "effective" sample size here is $n/\tau$. We use these bounds when analysing Markov chains in later chapters.

# CHAPTER 5
# PAC-Bayesian Analysis

Since the introduction of PAC learning theorems, many commonly known classes of learning problems have been studied for sample complexity and efficient learning algorithms. This ranges form special classes of Boolean expressions (Valiant, 1984) to linear classifiers and neural networks (Bartlett and Maass, 1995). There is also extensive literature on PAC learning in control problems in Markovian Decision Processes (Fiechter, 1994; Kearns and Singh, 2002; Brafman and Tennenholtz, 2003; Wiering and Schmidhuber, 1998; Strehl and Littman, 2005). The main drawback with most of these studies is that the bounds suggested by PAC theorems are often loose and impractical, leading to highly conservative learning algorithms.

The Bayesian approach to statistical learning, on the other hand, often times provides tighter bounds and faster convergence for learning algorithms. This is because one can incorporate domain knowledge into the learning algorithm (using a prior distribution) to constrain and direct the learning process. This, of course, comes at the expense of imposing the assumption of the correctness of the prior.

A general argument by the frequentist is that the loose bounds of PAC learning is the direct result of the inherent complexity of the learning problem when the assumption of the correctness of a prior is not enforced. Specifically, PAC learning does not impose any assumptions on the distribution of the data other than the i.i.d. assumption [1] , whereas the Bayesian viewpoint assumes the data is drawn from a distribution consistent with the prior. Therefore, bounds and correctness results of the Bayesian viewpoint will not hold if the prior is not correct.

The PAC-Bayesian approach, first introduced by McAllester (1999a) (extending the work of Shawe-Taylor and Williamson (1997)), combines the distribution-free correctness of PAC theorems with faster convergence results of informed Bayesian learning. This is achieved by removing the assumption of the correctness of the prior and, instead, measuring the consistency of the prior with the data. This is particularly useful as we can achieve uniform learning while having the speedup and efficiency of the Bayesian approach. Empirical analysis shows that model selection algorithms using these bounds are competitive with some of the most popular learning algorithms such as AdaBoost and Support Vector Machines (Germain et al., 2009).

The first attempt to provide uniform (distribution-independent) guarantees under prior injection, was the introduction of the Bayesian Luckiness framework by Shawe-Taylor and Williamson (1997). They place a prior on the hypothesis space and then estimate the volume of this space that is consistent with the data [2] . To do so, for each possible hypothesis $h$, defined on a geometric parameter space by

[1] Independent and identically distributed (i.i.d.) training dataset, sampled according to an unknown distribution.

[2] They only consider uniform priors in their original work.

the parameters $w$, they calculate the size of the ball around $w$ that is consistent with $h$ on all possible labellings of a given set of samples. Now given the target labellings on that sample set, they choose a $w$ as the output that is consistent with the labels and has the largest consistent ball around it. The size of the consistent ball around a parameter value is connected to the margin of the classifiers. Therefore, the mentioned algorithm would be an extension to the VC dimension framework in which large margins are equivalent to lower VC classes. This can also be explained in the *structural risk minimization* (SRM) framework that try to balance the correctness with the complexity of the classifier. In essence, the injection of the prior into the hypothesis space makes the large margin classifiers more appealing, leading to bounds on the generalization error that are tighter when the margins are large.

This chapter surveys recent PAC-Bayesian results and compares them to previous bounds and methods in the original PAC learning theory. We then study how PAC-Bayesian analysis extends to margin classifiers and general loss functions.

## 5.1 PAC-Bayes Bounds for Stochastic Classifiers

The first PAC-Bayesian results started with a preliminary model selection theorem, often times referred to as the "folk theorem" or the "Occam's razor". Here we use the form introduced by McAllester (1999a). Let $l(c, x) \to [0, 1]$ be a loss function that measures the correctness of the classifier $c$ with respect the target concept $c^*$ on a data point $x$. For instance for 0-1 loss we have:

$$l(c,x) = \begin{cases} 0 & c(x) = c^*(x) \\ 1 & c(x) \neq c^*(x) \end{cases} \tag{5.1}$$

Let $l_D(c)$ (true loss) be the expected loss of $c$ when the samples are drawn from i.i.d. form $D$: $l_D(c) = \underset{X\sim D}{\mathbb{E}} l(c,X)$. The *empirical loss* can be defined as $l_{U_\mathcal{D}}(c)$, where $U_\mathcal{D}$ is the uniform distribution on the sample set $\mathcal{D}$. The empirical loss is thus the empirical average of $l(c,.)$ on the sample set $\mathcal{D}$: $l_{U_\mathcal{D}}(c) = \sum_{x\in\mathcal{D}} l(c,x)/|\mathcal{D}|$. McAllester's version of the folk theorem is as follows:

**Theorem 5.1 (McAllester, 1999a).** *For any (prior) probability distribution $\rho_0$ assigning non-zero probability to each concept in a countable concept class $\mathcal{C}$, any choice of distribution $D$ on the sample space, any loss function $l$ mapping to $[0,1]$, and for any $0 < \delta < 1$, while sampling $n$ instances from $D$:*

$$\underset{\mathcal{D}\sim D^n}{\mathbb{P}\mathrm{r}} \left\{ \forall c \in \mathcal{C} : l_D(c) \leq l_{U_\mathcal{D}}(c) + \sqrt{\frac{\ln\frac{1}{\rho_0(c)} + \ln\frac{1}{\delta}}{2n}} \right\} \geq 1-\delta. \tag{5.2}$$

The above theorem suggests an algorithm that would select a concept $c$ that minimizes the bound. This is a form of *Structural Risk Minimization* (SRM) when the prior $\rho_0$ encodes the complexity of the classifier.

Notice that this bound holds simultaneously for all concepts. One can use this observation to extend this result to bound the loss function of a stochastic (Gibbs) classifier. A Gibbs classifier is defined by a probability distribution $\rho$ over the set of concepts. Denoted by $c^\rho_{\text{Gibbs}}$, the Gibbs classifier stochastically chooses a base classifier $c$ according to $\rho$ and uses that to classify the query. The loss functions can therefore be defined as the expectation of the loss over $\rho$: $l_D(\rho) = \mathop{\mathbb{E}}_{c\sim\rho} l_D(c)$. The PAC-Bayesian generalization of Theorem 5.1 for all posterior distributions $\rho$ can bound the loss of the Gibbs classifier:

**Theorem 5.2 (McAllester, 1999b).** *For any (prior) probability distribution $\rho_0$ over the concept class $\mathcal{C}$, any choice of distribution $D$ on the sample space, any loss function $l$ mapping to $[0,1]$, and for any $0 < \delta < 1$, while sampling $n$ instances from $D$:*

$$\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\forall\rho \in \mathcal{M}(\mathcal{C}) : l_D(\rho) \le l_{U_\mathcal{D}}(\rho) + \sqrt{\frac{\mathrm{KL}(\rho\|\rho_0) + \ln\frac{1}{\delta} + \ln n + 2}{2n-1}}\right\} \ge 1-\delta, \quad (5.3)$$

Here $\mathrm{KL}(\rho\|\rho_0)$ is the Kullback-Leibler (KL) divergence, which measures the difference between two distributions:

$$\mathrm{KL}(\rho\|\rho_0) \overset{\mathrm{def}}{=} \mathop{\mathbb{E}}_{c\sim\rho}\left[\ln\frac{d\rho(c)}{d\rho_0(c)}\right], \quad (5.4)$$

where $\frac{d\rho(c)}{d\rho_0(c)}$ is the Radon–Nikodym derivative of $\rho$ with respect to $\rho_0$. The Radon–Nikodym derivative can be substituted with the ratio of Probability Density Functions (PDF) if they exist.

Theorem 5.2 bounds the true loss of any Gibbs classifier around its empirical loss. The bound suggests an algorithm that, given a prior $\rho_0$, would select a distribution $\rho$ over the concept class that minimizes the bound. This can be thought of as the *PAC-Bayesian posterior* distribution (as opposed to the Bayesian posterior). McAllester (1999b) finds the value of the bound for such (optimal) posterior, for a given prior, and suggests a tractable posterior with a near optimal bound.

## 5.2 PAC-Bayes Bounds for Averaging Classifiers and Margins

The original PAC-Bayes bounds by McAllester are useful for Gibbs classifiers. Most of the commonly used classifiers, however, are not stochastic. *Averaging classifiers*, on the other hand, are more popular in learning algorithms (Germain et al., 2009). An averaging classifier (or a voting classifier) is defined as:

$$c^{\rho}_{\text{Avg}}(x) \overset{\text{def}}{=} \text{sign}(\bar{c}^{\rho}(x)) \overset{\text{def}}{=} \text{sign}(\mathop{\mathbb{E}}_{c\sim\rho} c(x)), \tag{5.5}$$

where $\bar{c}^{\rho}$ is the weighted average vote. The voting classifier takes the popular vote by taking the sign of $\bar{c}^{\rho}$.

Langford and Seeger (2001) used the PAC-Bayes bounds for Gibbs classifiers to bound the error rate of averaging classifiers (later improved in Langford et al. (2001)):

**Theorem 5.3 (Langford et al., 2001).** *For any (prior) probability distribution $\rho_0$ over the concept class $\mathcal{C}$, any choice of distribution $D$ on the sample space, and for any $0 < \delta < 1$, while sampling $n$ instances from $D$:*

$$\mathop{\mathbb{P}\text{r}}_{\mathcal{D}\sim D^n}\Bigg\{\forall\rho\in\mathcal{M}(\mathcal{C}),\theta>0: \tag{5.6}$$
$$l^0_D(c^{\rho}_{\text{Avg}}) \le l^{\theta}_{U_{\mathcal{D}}}(c^{\rho}_{\text{Avg}}) + O\left(\sqrt{\frac{\theta^{-2}\text{KL}(\rho\|\rho_0)\ln n + \ln n + \ln\delta^{-1}}{n}}\right)\Bigg\} \ge 1-\delta,$$

*where we define the* margin loss*:*

$$l^{\gamma}_D(c^{\rho}_{\text{Avg}}) = \mathop{\mathbb{P}\text{r}}_{X\sim D}\left\{c^*(X)\bar{c}^{\rho}(X) \le \gamma\right\}. \tag{5.7}$$

Here $l^0_D(c^{\rho}_{\text{Avg}})$ is the true error rate (the expected rate that the average vote $\bar{c}^{\rho}$ does not have the same sign as the target concept) and $l^{\theta}_{U_{\mathcal{D}}}(c^{\rho}_{\text{Avg}})$ is the empirical percentage of cases that violate the $\theta$ *margin*. The more classifiers agree on the correct class for $x$, the larger $c^*(x)\bar{c}^\rho(x)$ will get. Intuitively, it means that if most of the classifiers agree on the sample set (the margin is large and samples are far from the decision boundary), we are expected to get tighter bounds for the true error rate.

For any given $\rho$, the above bound could be minimized by some choice of $\theta$. So for any given prior $\rho_0$, one can think of the optimal $\rho$ and $\theta$ that minimize this bound. However, finding such optimal selections could be intractable in general.

Even when classifiers are not *averagers*, one can often construct an averaging classifier that is equivalent to the simple deterministic one. That is, for any classifier $c$ in a concept class $\mathcal{C}$, one might be able to find a posterior $\rho$ for which $c^\rho_{\mathrm{Avg}}(\cdot) = c(\cdot)$. Having such posterior $\rho$, one can use the bounds on the averaging classifier to get bounds on the original classifier $c$.

Herbrich and Graepel (2002) were the first to use the above observation to bound the true error rate of linear classifiers with margins, using appropriate choices for the $\rho_0$ and $\rho$. Their bound was used on linear Support Vector Machines (SVM). They show that an SVM is equivalent to an averaging classifier for which they can get PAC-Bayesian bounds on the true error rate. Their bound, however, did not scale well to high dimensions.

Going in the same direction, Langford and Shawe-Taylor (2002) provided tighter and more practical bounds on the error rate based on the empirical margin loss for any type of classifier with margin. These results are much tighter than the original margin bounds in the SRM literature and scale to high dimensions. They show that for any linear classifier $c$ with the weight vector $w$ in the feature space having margin $\gamma$, one can construct a posterior $\rho$ so that an averaging classifier with respect to $\rho$ is equivalent to $c$ (i.e. $c(\cdot) = c^{\rho}_{\text{Avg}}(\cdot)$). With careful choice of $\rho_0$ and $\rho$ for the PAC-Bayesian theorem, they bound the true error rate of $c^{\rho}_{\text{Avg}}$ (same as the true error rate for $c$) around the empirical margin loss. The resulting bound is given by the following theorem:

**Theorem 5.4 (Langford and Shawe-Taylor, 2002).** *For all averaging classifiers $c_w$ with normalized weights $w$, any choice of distribution $D$ on the sample space, while sampling $n$ instances from $D$:*

$$\Pr_{\mathcal{D}\sim D^m}\left\{\forall\epsilon, w, \gamma\colon \mathrm{KL}\Big(l^{\gamma}_{U_{\mathcal{D}}}(c_w) + \epsilon\Big\|l^0_D(c_w) - \epsilon\Big) \leq \frac{-\ln \bar{F}\left(\frac{2\bar{F}^{-1}(\epsilon)}{\gamma}\right) + 2\ln\frac{m+1}{\delta}}{m}\right\} \geq 1-\delta, \tag{5.8}$$

*where* $\mathrm{KL}(q\|p) = q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$ *for scalars $p$ and $q$, and we define* $\bar{F}(x) = 1 - \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}dx$.

One can optimize the above bound by choosing an appropriate value for $\gamma$. Langford (2005) introduced a similar bound and suggested techniques to optimize it with respect to the choice of margin and posterior parameters.

## 5.3 Tighter Bounds for General Loss Functions

The connection between the error rate of stochastic classifiers and the averaging ones usually come in the form of the following upper bound (Langford and Shawe-Taylor, 2002):

$$l^0_D(c^{\rho}_{\text{Avg}}) \leq 2l^0_D(c^{\rho}_{\text{Gibbs}}) \tag{5.9}$$

For instance for a uniform $\rho$, for the averaging classifier to err, at least half of the base classifiers should err, resulting in the error rate of no less than 1/2 for the Gibbs classifier.

This bound is not tight for classifiers with smaller margins and small variance in the margin size across the Gibbs samples (Germain et al., 2006). In the extreme case, if all the base classifiers have the same error rate, then the averaging classifier should have the same error rate as the Gibbs classifier. This bound is thus loose for classifiers produced by bagging (Breiman, 1996) or boosting (Freund and Schapire, 1995).

One can get tighter bounds for such classifiers by taking into account higher moments of the error rates. For example, if the variance on the error rate is small, then the bound on the error rate of the averaging classifier should be tighter. Germain et al. (2006) suggested a method to use the Taylor expansion of loss functions to get tighter bounds on the error rate of any convex combination of classifiers. Let $W_\rho(x)$ be the fraction of binary classifiers under $\rho$ that err for $x$:

$$W_\rho(x) = \underset{c \sim \rho}{\mathbb{P}\mathrm{r}} \left\{ c(x) \neq c^*(x) \right\} \tag{5.10}$$

Loss functions for averaging classifiers can usually be expressed in terms of $W_\rho$. For instance for 0-1 loss, one can use the indicator that $W_\rho$ is bigger than 1/2. The

0-1 loss can be thought of as the limit for sigmoid or tanh loss:

$$\begin{aligned} l_D^0(c_{\text{Avg}}^\rho) &= \Pr_{X\sim D}\{W_\rho(X) > 1/2\} && (5.11)\\ &= \mathop{\mathbb{E}}_{X\sim D} \mathbb{I}_{\{W_\rho(X)>1/2\}} && (5.12)\\ &= \mathop{\mathbb{E}}_{X\sim D} \lim_{\beta\to\infty} \frac{1}{2}(1+\tanh(\beta[2W_\rho(X)-1])). && (5.13)\end{aligned}$$

For many loss fucntions $\zeta$ (including tanh, sigmoid and exponential) we can expand the function using the Taylor series around $W_\rho(x) = 1/2$:

$$\begin{aligned} \zeta_D(c_{\text{Avg}}^\rho) &= \mathop{\mathbb{E}}_{X\sim D} \zeta_\rho(X) && (5.14)\\ &= \mathop{\mathbb{E}}_{X\sim D} \frac{1}{2}(1+f(2W_\rho(X)-1)) && (5.15)\\ &= \mathop{\mathbb{E}}_{X\sim D} \frac{1}{2} + \frac{1}{2}\sum_{k=1}^{\infty} g(k)(2W_\rho(X)-1)^k. && (5.16)\end{aligned}$$

Germain et al. (2006) connect the above expansion to the moments of the margin for the Gibbs classifier. They construct a binary classifier based on the Taylor expansion and then use the PAC-Bayesian bound of Theorem 5.2 to get a tighter bound on the loss of the averaging classifier. Defining $K$ and $\bar{K}$ to be:

$$\begin{aligned} K &= \sum_{k=1}^{\infty} |g(k)|, && (5.17)\\ \bar{K} &= \frac{1}{K}\sum_{k=1}^{\infty} |g(k)|\, k. && (5.18)\end{aligned}$$

We have the following bound:

**Theorem 5.5 (Germain et al., 2006).** *For any choice of (prior) distribution $\rho_0$ over the hypothesis space $\mathcal{C}$, any loss function $\zeta$ defined by Equation 5.14, $K$ and $\bar{K}$ defined by Equation 5.17 and Equation 5.18, and any choice of $\delta \in (0, 1]$, while* *sampling $n$ instances from sampling distribution $D$:*

$$\underset{\mathcal{D}\sim D^n}{\mathbb{P}\mathrm{r}}\left\{\forall Q\in\mathcal{M}(\mathcal{C}):\mathrm{KL}\left(\frac{1}{K}\left[\zeta_{U_{\mathcal{D}}}(c^{\rho}_{\mathrm{Avg}})-\frac{1}{2}\right]+\frac{1}{2}\,\middle\|\,\frac{1}{K}\left[\zeta_{D}(c^{\rho}_{\mathrm{Avg}})-\frac{1}{2}\right]+\frac{1}{2}\right)\right.$$
$$\left.\le\frac{1}{n}\left[\bar{K}\ .\ \mathrm{KL}(\rho\|\rho_0)+\ln\frac{n+1}{\delta}\right]\right\}\ge 1-\delta. \tag{5.19}$$

The value for $K$ is small (single digit) for most of the loss functions used in machine learning algorithms. Therefore, the above bound is tight on averaging classifiers that are based on low-margin “weak learners”, such as Adaboost, whereas the bounds based only on the first moment of the margin are impractical and loose (Germain et al., 2006).

The distributions $\rho_0$ and $\rho$ can be chosen based on the problem at hand. Germain et al. (2006) used a uniform prior $\rho_0$ on a discretized hypothesis space of linear classifiers, and used it for boosting with sigmoid and exponential loss. The posterior is then the one specified by the boosting algorithm. They show that their bound closely follow the test set error rates.

## 5.4 PAC-Bayes and Informed Priors

The bound on the error rate of margin classifiers provided by Langford and Shawe-Taylor (2002) use an isotropic Gaussian $\mathcal{N}(\mathbf{0}, I)$ around the origin as the prior over the normalized hypothesis parameters. Intuitively this prior enforces a complexity penalty for the set of hypothesis with large parameter values, encouraging classification models that are “smoother”. The posterior would be the same as the prior with the mean moved towards the parameter of the hypothesis in question.

This Gaussian prior, however, is not generally informative. Ambroladze et al. (2006) proposed an algorithm that would learn an informative prior based on a small portion of the training set and then compute the bound on the reminder of the examples relative to that prior. They first move the mean of the Gaussian towards an empirical estimate of the target parameters based on a small portion of the training data. They use this estimate instead of the isotropic Gaussian to get tighter bounds on the error rate.

## 5.5 Applications of PAC-Bayesian Analysis

PAC-Bayesian analysis has been mostly applied to supervised learning problems. The early analysis started with Gibbs classifiers (McAllester, 1999b). Later work have focused on deterministic margin bounds (Langford and Shawe-Taylor, 2002) and extensions to structured-prediction problems (London et al., 2013). More recently PAC-Bayesian framework have been used to analyse regularization techniques such as *dropouts* in neural networks (McAllester, 2013).

The application of PAC-Bayesian framework with sequential data in control problems have been limited to the analysis with martingales and bounds in finite action bandits (Seldin et al., 2011b,a, 2012). Chapter 6 includes our novel contributions in the application of PAC-Bayesian analysis to more general RL frameworks.

# CHAPTER 6
# PAC-Bayesian Reinforcement Learning

In this chapter, we derive PAC-Bayesian bounds on the approximation error in the value function of stochastic policies for reinforcement learning on MDPs. We start by a bound on model-based RL where a prior distribution is given on the space of possible models in a finite state-space. We then introduce bounds with model-free RL, where a prior is given on the space of value functions. In both cases, the bound depends both on an empirical estimate and a measure of distance between the stochastic policy and the one imposed by the prior distribution. We present empirical results where model-selection is performed based on these bounds, and show that PAC-Bayesian bounds follow Bayesian policies when the prior is informative and mimic the PAC policies when the prior is not consistent with the data. This allows us to adaptively balance between the distribution-free correctness of PAC and the data-efficiency of Bayesian inference.

## 6.1 Model-Based PAC-Bayesian RL in Finite MDPs

In model-based RL, one aims to estimate the transition and reward functions and then act optimally according to the estimated models. In finite MDPs, there are a finite number of parameters to estimate: The transition probabilities (parameters of multinomial distribution) and the average rewards. PAC methods use the empirical average for their estimated model along with frequentist bounds. Bayesian methods use the Bayesian posterior to estimate the model. This section provides a bound that suggests an adaptive method to choose a stochastic estimate between these two extremes, which is both data-efficient and has guaranteed performance.

In this section, we assume that the reward model is known and we only need to estimate the transition parameters. In practice, estimates of the reward model converge much faster to the correct values, resulting in small errors in the value prediction. Nevertheless, once can extend the same analysis using bounds and prior distributions over the reward parameters. We avoid the analysis of the reward estimation to simplify the derivations.

Assuming that the reward model is known, one can build empirical models of the transition dynamics by gathering sample transitions, denoted by $\mathcal{D}$, and taking the empirical average. For finite MDPs, we assume state $s$ is an integer between 1 and $|\mathcal{S}|$, and action $a$ is an integer between 1 and $|\mathcal{A}|$. Let this empirical average model be a multinomial $\hat{T}(s,a)$ approximation $T(s,a)$. We use the tensor notation $\hat{\mathbf{T}}_{s,a,s'}$ and $\mathbf{T}_{s,a,s'}$ to represent the parameters of these multinomial:

$$\mathbf{T}_{s,a,s'} \overset{\text{def}}{=} \Pr_{S' \sim T(s,a)} \left\{ S' = s' \right\}, \tag{6.1}$$

$$\hat{\mathbf{T}}_{s,a,s'} \overset{\text{def}}{=} n_{s,a,s'}/n_{s,a}, \tag{6.2}$$

where $n_{s,a,s'}$ and $n_{s,a}$ are the number of corresponding transitions and samples. Note that $\hat{\mathbf{T}}_{s,a}$ and $\mathbf{T}_{s,a}$ are vectors of parameters for the corresponding multinomial distributions.

Trivially, $\mathbb{E}\,\hat{\mathbf{T}} = \mathbf{T}$. The empirical value function, denoted by $\hat{Q}$, is defined to be the value function on an MDP with the empirical transition model. As one observes more and more sample trajectories on the MDP, the empirical model gets increasingly more accurate, and so will the empirical value function. Different forms of the following lemma, connecting the error rates on $\hat{\mathbf{T}}$ and $\hat{Q}$, are used in many of the PAC-MDP results:

**Lemma 6.1 (Strehl and Littman, 2005).** *There is a constant $k \geq (1-\gamma)^2/\gamma$, such that if:*

$$\forall s, a : \|\hat{\mathbf{T}}_{s,a} - \mathbf{T}_{s,a}\|_1 \leq k\epsilon, \tag{6.3}$$

*then the worst case error on the state–action value function is bounded by $\epsilon$ for all policies:*

$$\forall \pi : \|\hat{Q}^\pi - Q^\pi\|_\infty \leq \epsilon. \tag{6.4}$$

As a consequence of the above lemma, one can act near-optimally in the part of the MDP for which we have gathered enough samples to have a good empirical estimate of the transition model. PAC-MDP methods explicitly (Kearns and Singh, 2002) or implicitly (Brafman and Tennenholtz, 2003) use that fact to exploit the knowledge on the model as long as they are in the "known" part of the state-space. The downside of these methods is that without further assumptions on the model, it will take a large number of sample transitions to get a good empirical estimate of the transition model.

The Bayesian approach to modeling the transition dynamics, on the other hand, starts with a prior distribution over the transition probability and then marginalizes this prior over the data to get a posterior distribution. This is usually done by assuming independent Dirichlet distributions over the transition probabilities, with some initial count vector $\alpha$, and then adding up the observed counts to this initial vector to get the conjugate posterior (Duff, 2002). The initial $\alpha$-vector encodes the prior knowledge on the transition probabilities, and larger initial values further bias the empirical observation towards the initial belief.

If a strong prior is close to the true values, the Bayesian posterior will be more accurate than the empirical point estimate. However, a strong prior peaked on the wrong values will bias the Bayesian model away from the correct probabilities. Therefore, the Bayesian posterior might not provide the optimal estimate of the model parameters. A "good posterior" distribution might be somewhere between the empirical point estimate and the Bayesian posterior.

To develop a PAC-Bayesian bound we need to develop a notion of stochastic policy similar to the Gibbs classifier. In this section, we define a *Gibbs policy* to follow this process: a deterministic policy is sampled from a given distribution, and then that policy is used to interact with the environment. This is different form of stochastic policy than is usually seen in the RL literature. When a complete policy is sampled, the actions on different states might be correlated, and the action selection is the same when observing repeated states in a trajectory.

The following theorem is our first PAC-Bayesian bound on the prediction error of the value function when using a Gibbs policy based on an arbitrary posterior distribution $\rho$ over the transition parameters. We assume that we are given prior knowledge in form of a prior distribution $\rho_0$ on the transition parameters. The theorem provides a simultaneous bound over all posterior distributions and thus can be used for model-selection based on the posterior distribution.

**Theorem 6.2.** *Let $\pi^*_{\tilde{\mathbf{T}}}$ be a deterministic optimal policy with respect to the MDP with transition tensor $\tilde{\mathbf{T}}$, and define $\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}} \overset{\text{def}}{=} \|\hat{Q}^{\pi^*_{\tilde{\mathbf{T}}}} - Q^{\pi^*_{\tilde{\mathbf{T}}}}\|^2_\infty$. Let $k$ be the constant in Lemma 6.1. For any prior distribution $\rho_0$ on the transition tensor space $\mathcal{T}$, any i.i.d. sampling distribution over transitions $\mathcal{D} \sim D^n$, for any $0 < \delta < 1$:*

$$\Pr_{\mathcal{D}\sim D^n}\left\{\forall \rho \in \mathcal{M}(\mathcal{T}) : \mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho} \Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}} \leq \frac{\mathrm{KL}(\rho\|\rho_0) - \ln\delta + |\mathcal{S}|\ln 2 + \ln|\mathcal{S}| + \ln n_{\min}}{(n_{\min}-1)k^2/2}\right\} \geq 1-\delta, \tag{6.5}$$

*where $n_{\min} = \min_{s,a} n_{s,a}$.*

The above theorem (proved in Section 6.5) provides a high-probability lower bound on the expectation of the true value function when the policy is taken to be optimal according to the sampled model from the posterior. Simplifying the bound by taking the square root of the sides, we have that for all state-action pairs $(s,a)$:

$$\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho} Q^{\pi^*_{\mathbf{T}'}}(s,a) \geq \mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho} \hat{Q}^{\pi^*_{\mathbf{T}'}}(s,a) - \tilde{O}\left(\sqrt{\frac{\mathrm{KL}(\rho\|\rho_0) + |\mathcal{S}|}{n_{\min}}}\right). \tag{6.6}$$

This lower bound suggests a stochastic model-selection method in which one searches in the space of posteriors to maximize the bound. Notice that there are two elements to the above bound. One is the PAC part of the bound that suggests the selection of models with high empirical value functions for their optimal policy. There is also a penalty term (or a regularization term) that penalizes distributions that are far from the prior (the Bayesian side of the bound).

## 6.2 Margin for Deterministic Policies in Finite MDPs

One could apply Theorem 6.2 with any choice $\rho$. Generally, this will result in a bound on the value of a stochastic policy. However, if the optimal policy is the same for all of the possible samples from the posterior, then we will get a bound for that particular deterministic policy.

We define the *support* of deterministic policy $\pi$, denoted by $\mathcal{T}_\pi$, to be the set of transition models for which $\pi$ is an optimal policy. Putting all the posterior probability on $\mathcal{T}_\pi$ will result in a tighter bound for the value of the policy $\pi$. The tightest bound occurs when $\rho$ is a scaled version of $\rho_0$ summing to 1 over $\mathcal{T}_\pi$, that is when we have:

$$\rho(\tilde{\mathbf{T}}) = \begin{cases} \frac{\rho_0(\tilde{\mathbf{T}})}{\rho_0(\mathcal{T}_\pi)} & \tilde{\mathbf{T}} \in \mathcal{T}_\pi \\ 0 & \tilde{\mathbf{T}} \notin \mathcal{T}_\pi \end{cases} \tag{6.7}$$

In that case, the KL divergence is $\mathrm{KL}(\rho \| \rho_0) = -\ln \rho_0(\mathcal{T}_\pi)$, and the high-probability bound will be:

$$Q^\pi \geq \hat{Q}^\pi - \tilde{O}\left(\sqrt{\frac{|\mathcal{S}| - \ln \rho_0(\mathcal{T}_\pi)}{n_{\min}}}\right). \tag{6.8}$$

Intuitively, we get tighter bounds for policies that have larger empirical values and higher prior probabilities supporting them.

Finding $\rho_0(\mathcal{T}_\pi)$ might not be computationally tractable. Therefore, we define a notion of *margin* for transition functions and policies and use it to get tractable bounds. The margin of a transition tensor $\tilde{\mathbf{T}}$, denoted by $\theta_{\mathbf{T}'}$, is the maximum distance we can move away from $\tilde{\mathbf{T}}$ such that the optimal policy does not change (assuming the deterministic optimal policy is unique):

$$\forall \tilde{\mathbf{T}}' : \Big\{ \forall s \in \mathcal{S}, a \in \mathcal{A} : \|\tilde{\mathbf{T}}(s,a,.) - \tilde{\mathbf{T}}'(s,a,.)\|_1 \le \theta_{\tilde{\mathbf{T}}} \Big\} \Rightarrow \pi^*_{\tilde{\mathbf{T}}'} = \pi^*_{\tilde{\mathbf{T}}}. \tag{6.9}$$

The margin defines a hypercube around $\tilde{\mathbf{T}}$ for which the optimal policy does not change. In cases where the support set of a policy is difficult to find, one can use this hypercube to get a reasonable bound for the true value function of the corresponding policy. In that case, we would define the posterior to be the scaled prior defined only on the margin hypercube. The idea behind this method is similar to that of the Luckiness framework (Shawe-Taylor and Williamson, 1997) and large-margin classifiers (Herbrich and Graepel, 2002; Langford and Shawe-Taylor, 2002). This shows that the idea of maximizing margins can be applied to control problems as well as classification and regression tasks.

To find the margin of any given $\tilde{\mathbf{T}}$, if we know the value of the second best policy, we can calculate its regret according to $\tilde{\mathbf{T}}$. This will be the smallest regret $\eta_{\min}$ (regret gap). Using Lemma 6.1, we can conclude that if for all $s$ and $a$, $\|\tilde{\mathbf{T}}(s,a,.) - \tilde{\mathbf{T}}'(s,a,.)\|_1 \le k\eta_{\min}/2$, then the value of the best and second best policies can change by at most $\eta_{\min}/2$, and thus the optimal policy will not change. Therefore, $\theta_{\tilde{\mathbf{T}}} \ge k\eta_{\min}/2$. One can then define the posterior on the transitions inside the margin to get a bound for the value function.

## 6.3 Model-Free PAC-Bayesian RL in Finite MDPs

In this section we introduce a PAC-Bayesian bound for model-free reinforcement learning on discrete state-spaces. This time we assume that we are given a prior distribution on the space of value functions, rather than on transition models.

This prior encodes an initial belief about the optimal value function for a given RL domain. Such method could be useful, for example, in the context of transfer learning, where one has learned a value function in one environment and then uses that as the prior belief on a similar domain.

Recall the optimal Bellman operator $\mathbb{T}^*$ applied to the action-value $Q$ functions. In most cases, we do not have access to the Bellman optimality operator. When we only have access to a sample set $\mathcal{D}$ collected on the RL domain, we can define the empirical Bellman optimality operator $\hat{\mathbb{T}}^*$ to be:

$$(\hat{\mathbb{T}}^* Q)(s,a) = \frac{1}{n_{s,a}} \sum_{(s,a,s',r)\in\mathcal{D}} \left( r + \gamma \max_{a'} Q(s',a') \right), \tag{6.10}$$

Note that $\mathop{\mathbb{E}}_{\mathcal{D}\sim D^n}\left[\hat{\mathbb{T}}^* Q\right] = \mathbb{T}^* Q$ (since $\hat{\mathbb{T}}^*$ implicitly applies a projection by averaging). We consider only $Q$ functions that are bounded by the maximum possible value $V_{\max}$. Therefore all the $\mathbb{T}^* Q$ and $\hat{\mathbb{T}}^* Q$ values we could observe are bounded by $V_{\max}$. If the sampling of transitions (or equivalently sampling of state–action pairs) are i.i.d., one can use Hoeffding's inequality to bound the difference between the empirical and true Bellman operators for any fixed state–action pair:

$$\Pr_{\mathcal{D}\sim D}\left\{ \left| \hat{\mathbb{T}} Q(s,a) - \mathbb{T} Q(s,a) \right| > \epsilon \right\} \le 2e^{-2n_{s,a}\epsilon^2/V_{\max}^2}. \tag{6.11}$$

Q-learning (Sutton and Barto, 1998) and its derivations with function approximation (Sutton et al., 2000), and also batch methods such as LSTD (Boyan, 2002), often aim to minimize the empirical (projected) TD error. We argue that it might be better to choose a loss function that is not only capturing the empirical Bellman error, but also penalizes the structural risk of the model. We aim to minimize a PAC-Bayesian upper bound on the prediction error of the action-value function, which in turns controls the decision-theoretic "complexity" of the chosen posterior model.

The following theorem (proved in Section 6.5) is our first PAC-Bayesian bound for model-free batch RL on finite state-spaces:

**Theorem 6.3.** *Let* $\Delta_Q^{\mathcal{D}} = \left( \|Q - Q^*\|_\infty - \frac{\|Q - \hat{\mathbb{T}}^* Q\|_\infty}{1-\gamma} \right)_+^2$. *For any prior distributions* $\rho_0$ *over a measurable space of value functions* $\mathcal{Q}$ *with range in* $[0, V_{\max}]$, *any i.i.d. sampling distribution over transitions* $\mathcal{D} \sim D^n$*:*

$$\Pr_{\mathcal{D}\sim D^n} \left\{ \forall \rho : \mathop{\mathbb{E}}_{Q\sim\rho} \Delta_Q^{\mathcal{D}} \leq \frac{\mathrm{KL}(\rho\|\rho_0) - \ln\delta + \ln|\mathcal{S}| + \ln|\mathcal{A}| + \ln(2n_{\min})}{2(n_{\min}-1)(1-\gamma)^2/V_{\max}^2} \right\} > 1 - \delta. \quad (6.12)$$

We can simplify the bound of Theorem 6.3 by taking the square root of the sides and rearranging the terms, and get a high-probability upper bound on the expected worst-case value prediction error:

$$\mathop{\mathbb{E}}_{Q\sim\rho} \|Q - Q^*\|_\infty \leq \frac{\mathop{\mathbb{E}}_{Q\sim\rho} \left\|Q - \hat{\mathbb{T}}^* Q\right\|_\infty}{1-\gamma} + \tilde{O}\left( \sqrt{\frac{\mathrm{KL}(\rho\|\rho_0)}{n_{\min}}} \right). \quad (6.13)$$

This suggests a model-selection method in which one would search for a posterior $\rho$ to minimize the above bound. The PAC side of the bound guides this model-selection method to look for posteriors with smaller empirical TD error. The Bayesian part, on the other hand, penalizes the selection of posteriors that are far from the prior distribution.

One can use general forms of priors that would impose smoothness or sparsity for this model-selection technique. In that sense, this method would act as a regularization technique that penalizes complex and irregular value functions.

The idea of regularization in RL with function approximation is not new to this work (Kolter and Ng, 2009b; Farahmand et al., 2009b,a). This bound, however, is more general, as it could incorporate not only smoothness constraints, but also other forms of prior knowledge into the learning process.

## 6.4 Model-Free PAC-Bayesian RL in Continuous Space MDPs

The PAC-Bayesian bounds presented the previous sections are on finite state-spaces and require an i.i.d. sampling strategy that visits all states and applies all actions similarly often (the bounds depend on the minimum of the number of visits $n_{\min}$). These bounds, however, are worst-case simultaneous bounds on all states and actions, and can be used in off-policy setting. Specifically, the bounds are on the optimal value function or the value of near-optimal policies, using a dataset collected from any i.i.d. distribution.

In this section we introduce a PAC-Bayesian bound in continuous state and action MDPs, using trajectories with dependent states and actions. This generalization, of course, comes with some assumptions and limitations. In this section, we only seek to evaluate a fixed policy (value prediction), rather than finding optimal values or policies. We also require the MDP to *mix fast* under the given policy and have a stationary distribution. Unlike the worst-case bounds of previous sections, the bound we obtain here is with respect to the $\ell^2$ norm under stationary distribution of the policy.

Consider an MDP with (possibly infinite) state-space $\mathcal{S}$, and a fixed policy $\pi$ whose stationary distribution $\nu(\pi)$ over the states exists. Fixing the policy, we collect a trajectory of states, actions and rewards on the MDP: $S_{1:n+1}$, $A_{1:n}$, $R_{1:n}$.

We define the following:

$$\hat{J}^{\pi}_{\mathrm{TD}}(V) \overset{\mathrm{def}}{=} \frac{1}{n}\sum_{i=1}^{n}(R_i + \gamma V(S_{i+1}) - V(S_i))^2, \tag{6.14}$$

$$J^{\pi}_{\mathrm{TD}}(V) \overset{\mathrm{def}}{=} \mathop{\mathbb{E}}_{\substack{S\sim\nu(\pi)\\ A\sim\pi(S)\\ R\sim R(S,A)\\ S'\sim T(S,A)}} \left[(R + \gamma V(S') - V(S))^2\right], \tag{6.15}$$

$$\Lambda^{\pi}_{\mathrm{TD}}(V) \overset{\mathrm{def}}{=} \mathop{\mathbb{E}}_{S\sim\nu(\pi)} \mathop{\mathbb{V}\mathrm{ar}}_{\substack{A\sim\pi(S)\\ R\sim R(S,A)\\ S'\sim T(S,A)}} \left[R + \gamma V(S')\right]. \tag{6.16}$$

$\hat{J}^{\pi}_{\mathrm{TD}}(V)$ is the average squared TD-error. $J^{\pi}_{\mathrm{TD}}(V)$ and $\Lambda^{\pi}_{\mathrm{TD}}(V)$ are the second norm and the variance of the TD-error under the stationary distribution.

To arrive at a bound linking $\hat{J}^{\pi}_{\mathrm{TD}}(V)$ and $J^{\pi}_{\mathrm{TD}}(V)$, we need to use the theory of fast-mixing Markov chains, discussed in Section 4.4. Consider the Markov chain $X_{1:n}$ consisting of collected transitions:

$$X_i \overset{\mathrm{def}}{=} (S_i, A_i, R_i, S_{i+1}). \tag{6.17}$$

We consider the chains that uniformly quickly forget their past (with a finite forgetting time $\tau$). For such chains, we can link the empirical squared TD-error to its expectation under the stationary distribution:

**Lemma 6.4.** *Let $\mathcal{D} = X_{i:1}$ be the Markov chain of transitions under a fixed policy $\pi$, defined in Equation 6.17, with distribution $C^{\pi}_{n}$ and forgetting time $\tau$, and assume $\pi$ has a stationary distribution $\nu(\pi)$. For any measurable function $V$ bounded by $V_{\max}$, and any $0 < \delta < 1$:*

$$\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim C^{\pi}_{n}} \left\{ J^{\pi}_{\mathrm{TD}}(V) - \hat{J}^{\pi}_{\mathrm{TD}}(V) \leq V^2_{\max}\sqrt{\frac{2\tau}{n}\ln\frac{1}{\delta}} \right\} \geq 1 - \delta. \tag{6.18}$$

*Proof of Lemma 6.4.* From the definition of stationary distribution and factorizing the expectations, we have that:

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim C_n^\pi}\left[\hat{J}^\pi_{\mathrm{TD}}(V)\right] = J^\pi_{\mathrm{TD}}(V). \tag{6.19}$$

Each term in $J^\pi_{\mathrm{TD}}$ is bounded by $V_{\max}^2$. combining the above with the concentration inequality of Theorem 4.4, we obtain the bound of the Lemma. □

One can use the bound of Lemma 6.4 along with the change of measure inequality and variance decomposition techniques to obtain the following PAC-Bayesian theorem on the prediction error a Gibbs value estimator. The proof is provided in Section 6.5.

**Theorem 6.5.** *Fix a measurable set $\mathcal{V}$ of real-valued, measurable functions with domain $\mathcal{X}$, which are bounded by $0$ and $V_{\max}$. Assume the conditions and notations of Lemma 6.4. Fix any prior measure $\rho_0$ over $\mathcal{V}$. Then, for all $0 < \delta < 1$:*

$$\mathop{\mathbb{P}\mathrm{r}}_{X_{1:n}\sim C_n^\pi}\Bigg\{\forall \rho \in \mathcal{M}(\mathcal{V}) : \mathop{\mathbb{E}}_{V\sim\rho}\left[\|V - V^\pi\|^2_{\nu(\pi)}\right] \leq \\ \frac{1}{(1-\gamma)^2}\left(\mathop{\mathbb{E}}_{V\sim\rho}\left[J^\pi_{\mathrm{TD}}(V)\right] + \sqrt{\frac{\mathrm{KL}(\rho\|\rho_0) + \ln\frac{n}{\delta}}{(n-1)/2\tau V_{\max}^4}} - \mathop{\mathbb{E}}_{V\sim\rho}\left[\Lambda^\pi_{\mathrm{TD}}(V)\right]\right)\Bigg\} > 1-\delta. \tag{6.20}$$

Theorem 6.5 bounds the expected error of approximating $V^\pi$ with a value function drawn randomly from some distribution $\rho$. Note that in this theorem, $\rho_0$ must be a fixed distribution, chosen *a priori* (i.e. prior distribution), but $\rho$ can be chosen in a data dependent manner (i.e., it can be a "posterior" distribution).

Notice that there are three elements to the above bound (right hand side). The first term is the empirical component of the bound, which enforces the selection of solutions with smaller empirical Bellman residuals. The second term is the Bayesian component of the bound, which penalizes distributions that are far from the prior. The third term corrects for the variance in the return at each state.

If we can empirically estimate the right hand side of the above inequality, then we can use the bound in an algorithm. For example, we can derive a PAC-Bayesian model-selection algorithm that searches in the space of posteriors $\mu$ so as to minimize the upper bound.

### 6.4.1 Linearly parametrized classes of functions

In this section, we study the application of Theorem 6.5 to the class of linearly parametrized functions with bounded parameters,

$$\mathcal{V}_{\text{Lin}} = \left\{ V_\theta(s) \overset{\text{def}}{=} \theta^T \varphi(s) : \theta \in \mathbb{R}^D, \|\theta\| \leq V_{\max}/F_{\max} \right\}, \tag{6.21}$$

where $\varphi(\cdot) : \mathcal{S} \to \mathbb{R}^D$ is the feature vector of a state, and $F_{\max}$ is the finite bound over the feature norm:

$$F_{\max} \overset{\text{def}}{=} \sup_{s \in \mathcal{S}} \|\varphi(s)\| < \infty. \tag{6.22}$$

In this case, the prior and posterior measures $\rho_0$ and $\rho$ can be put on the ball $\{\theta : \|\theta\| \leq V_{\max}/F_{\max}\}$. A natural choice for the prior would be one that puts higher probability for smaller $\ell^2$ norms , thus encouraging smoothness in the value function, or one that penalizes the $\ell^1$ or $\ell^0$ norms to enforce sparsity in the parameters as a means of feature selection.

Let us now turn to the estimation of the variance term. Assume that the policy is deterministic. With that assumption, the reward for each transition will be independent of the next state. We thus get:

$$\underset{\substack{R\sim R(S,\pi(S))\\ S'\sim T(S,\pi(S))}}{\mathbb{V}\mathrm{ar}}\left[R+\gamma V(S')\right] \;=\; \underset{R\sim R(S,\pi(S))}{\mathbb{V}\mathrm{ar}}\left[R\right]+\gamma^2 \underset{S'\sim T(S,\pi(S))}{\mathbb{V}\mathrm{ar}}\left[V(S')\right]. \tag{6.23}$$

Now with linear value functions $V(s) = V_\theta(s) = \theta^\top \varphi(s)$, we have:

$$\underset{S'\sim T(S,\pi(S))}{\mathbb{V}\mathrm{ar}}\left[V(S')\right] \;=\; \theta^T \underset{S'\sim T(S,\pi(S))}{\mathbb{C}\mathrm{ov}}\left[\varphi(S')\right]\theta. \tag{6.24}$$

Defining:

$$\sigma_R^2 \;=\; \underset{S\sim\nu(\pi)}{\mathbb{E}}\left[\underset{R\sim R(S,\pi(S))}{\mathbb{V}\mathrm{ar}}\left[R\right]\right] \tag{6.25}$$

$$\Sigma_\varphi \;=\; \underset{S\sim\nu(\pi)}{\mathbb{E}}\left[\underset{S'\sim T(S,\pi(S))}{\mathbb{C}\mathrm{ov}}\left[\varphi(S')\right]\right], \tag{6.26}$$

we get the expected variance term:

$$\Lambda^\pi_{\mathrm{TD}}(V) = \sigma_R^2 + \gamma^2\theta^\top\Sigma_\varphi\theta\,. \tag{6.27}$$

### 6.4.2 Estimating the constants

In some cases the terms $\sigma_R^2$ and $\Sigma_\varphi$ are known (e.g., $\sigma_R^2 = 0$ when the rewards are a deterministic function of the start state and action, and $\Sigma_\varphi = 0$ when the dynamics are deterministic). An alternative is to estimate these terms empirically. This can be done by, e.g., double sampling of next states (assuming one has access to a generative model, or if one can reset the state[1] ). If such estimates are generated based on finite sample sets, then we will need to add extra deviation terms to the bound of Theorem 6.5.

For simplicity, we assume that these terms are either known or can be estimated on a separate dataset of many transitions. Examples of such cases are studied in the empirical results.

The estimation of the forgetting constant $\tau$ is the subject of active research (Farahmand and Szepesvári, 2011). There are explicit upper bounds for special cases. For example, $\tau = 1$ for i.i.d. samples. The forgetting constant can also be estimated from data if we have access to a generative model.

## 6.5 Proofs of PAC-Bayesian Bounds

The proof of the stated PAC-Bayesian bounds all rely on the following lemma, *the change of measure inequality*, originated by McAllester (1999b), and extended to general measures by Banerjee (2006):

**Lemma 6.6 (Banerjee, 2006).** *For any measurable function $f(h)$ on $\mathcal{H}$, and for any distributions $\rho_0$, and $\rho_1$ over $\mathcal{H}$:*

$$\mathop{\mathbb{E}}_{h\sim\rho_1} f(h) \le \mathrm{KL}(\rho_1\|\rho_0) + \ln \mathop{\mathbb{E}}_{h\sim\rho_0} e^{f(h)}. \tag{6.28}$$

[1] Alternately, one can co-estimate the mean and variance terms (Sutton et al., 2009), keeping a current guess of them and updating both estimates as new transitions are observed.

### 6.5.1 Proof of Theorem 6.2

We use the following intermediate lemma for the proof:

**Lemma 6.7.** *Using the notation of Theorem 6.2:*

$$\Pr_{\mathcal{D}\sim D^n}\left\{\mathbb{E}_{\tilde{\mathbf{T}}\sim\rho_0}\left[e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}\right] \geq \frac{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}}{\delta}\right\} \leq \delta. \tag{6.29}$$

*Proof of Lemma 6.7.* For a fix $\tilde{\mathbf{T}}$, let $a_s = \pi^*_{\tilde{\mathbf{T}}}(s)$. We have:

$$\begin{aligned}
\Pr_{\mathcal{D}\sim D^n}\left\{\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}} \geq \epsilon\right\} &\leq \sum_{s=1}^{\mathcal{S}} \Pr_{\mathcal{D}\sim D^n}\left\{\|\tilde{\mathbf{T}}(s,a_s,.) - \mathbf{T}(s,a_s,.)\|_1 > k\sqrt{\epsilon}\right\} && (6.30)\\
&\leq \sum_{s=1}^{\mathcal{S}}\left(2^{|\mathcal{S}|}e^{-\frac{1}{2}n_{s,a_s}k^2\epsilon}\right) && (6.31)\\
&\leq |\mathcal{S}|2^{|\mathcal{S}|}e^{-\frac{1}{2}n_{\min}k^2\epsilon}. && (6.32)
\end{aligned}$$

Line 6.30 is by Lemma 6.1. Line 6.31 is a concentration inequality for multinomials (Weissman et al., 2003). Using the above, we have thus for all $\tilde{\mathbf{T}}$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}\sim D^n}\left[e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}\right] &= 1 + \int_1^\infty \Pr_{\mathcal{D}\sim D^n}\left\{e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}} \geq t\right\}dt && (6.33)\\
&= 1 + \int_0^\infty \Pr_{\mathcal{D}\sim D^n}\left\{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}} \geq t\right\}e^t dt && (6.34)\\
&= 1 + \int_0^\infty \Pr_{\mathcal{D}\sim D^n}\left\{\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}} \geq \frac{2t}{(n_{\min}-1)k^2}\right\}e^t dt && (6.35)\\
&\leq 1 + \int_0^\infty |\mathcal{S}|2^{|\mathcal{S}|}e^{-\frac{1}{2}n_{\min}k^2\frac{2t}{(n_{\min}-1)k^2}}e^t dt && (6.36)\\
&= 1 + |\mathcal{S}|2^{|\mathcal{S}|}(n_{\min}-1) && (6.37)\\
&\leq |\mathcal{S}|2^{|\mathcal{S}|}n_{\min}. && (6.38)
\end{aligned}$$

Line 6.34 changes the variable of integration, and in Line 6.36 we use the inequality of Equation 6.32.

To prove Lemma 6.7 we apply Markov's inequality, and then swap the expectations (since $\rho_0$ is a fixed "prior" and hence independent from $\mathcal{D}$):

$$\begin{aligned}\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho_0}\left[e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}\right]\geq\frac{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}}{\delta}\right\} &\leq \frac{\mathop{\mathbb{E}}_{\mathcal{D}\sim D^n}\left[\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho_0}\left[e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}\right]\right]}{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}/\delta} &(6.39)\\ &\leq \frac{\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho_0}\left[\mathop{\mathbb{E}}_{\mathcal{D}\sim D^n}\left[e^{\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}\right]\right]}{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}/\delta} &(6.40)\\ &\leq \delta. &(6.41)\end{aligned}$$

This concludes the proof of the lemma. □

*Proof of Theorem 6.2.* Lemma 6.7 and Lemma 6.6 together imply Therorem 6.2 by using:

$$f_{\mathcal{D}}(\tilde{\mathbf{T}})=\frac{1}{2}(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}, \tag{6.42}$$

on the hypothesis space $\mathcal{T}$:

$$\begin{aligned}&\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\exists\rho\in\mathcal{M}(\mathcal{T}):\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho}\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}\geq\frac{\mathrm{KL}(\rho\|\rho_0)-\ln\delta+|\mathcal{S}|\ln 2+\ln|\mathcal{S}|+\ln n_{\min}}{(n_{\min}-1)k^2/2}\right\}\\ &=\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\exists\rho\in\mathcal{M}(\mathcal{T}):\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho}\frac{(n_{\min}-1)k^2\Delta^{\mathcal{D}}_{\tilde{\mathbf{T}}}}{2}\geq\mathrm{KL}(\rho\|\rho_0)+\ln\frac{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}}{\delta}\right\} &(6.43)\\ &=\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\exists\rho\in\mathcal{M}(\mathcal{T}):\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho}f_{\mathcal{D}}(\tilde{\mathbf{T}})-\mathrm{KL}(\rho\|\rho_0)\geq\ln\frac{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}}{\delta}\right\} &(6.44)\\ &\leq\mathop{\mathbb{P}\mathrm{r}}_{\mathcal{D}\sim D^n}\left\{\ln\mathop{\mathbb{E}}_{\tilde{\mathbf{T}}\sim\rho_0}e^{f_{\mathcal{D}}(\tilde{\mathbf{T}})}\geq\ln\frac{|\mathcal{S}|2^{|\mathcal{S}|}n_{\min}}{\delta}\right\}\leq\delta &(6.45)\end{aligned}$$

□

### 6.5.2 Proof of Theorem 6.3

We first visit the following simple lemma linking the Bellman error and error in the value function:

**Lemma 6.8 (Williams and Baird, 1993).** *For any $Q$, and the optimal action value function $Q^*$:*

$$\|Q - Q^*\|_\infty \leq \frac{1}{1-\gamma} \|Q - \mathbb{T}^* Q\|_\infty \tag{6.46}$$

*Proof of Lemma 6.8.* Since $\mathbb{T}^*$ is a contraction with respect to the infinity norm and $Q^*$ is its fixed point, we have:

$$\begin{aligned} \|Q - Q^*\|_\infty &= \|Q - \mathbb{T}^* Q + \mathbb{T}^* Q - \mathbb{T}^* Q^*\|_\infty & (6.47) \\ &\leq \|Q - \mathbb{T}^* Q\|_\infty + \|\mathbb{T}^* Q - \mathbb{T}^* Q^*\|_\infty & (6.48) \\ &\leq \|Q - \mathbb{T}^* Q\|_\infty + \gamma \|Q - Q^*\|_\infty, & (6.49) \end{aligned}$$

which proves the lemma by rearrangement. □

**Lemma 6.9.** *Using the notation of Theorem 6.3, for any fixed prior $\rho_0$ over the space of value functions:*

$$\Pr_{\mathcal{D} \sim D^n} \left\{ \mathop{\mathbb{E}}_{Q \sim \rho_0} \left[ e^{2(n_{\min}-1)(1-\gamma)^2 \Delta_Q^{\mathcal{D}} / V_{\max}^2} \right] \leq \frac{2|\mathcal{S}||\mathcal{A}| n_{\min}}{\delta} \right\} \leq 1 - \delta. \tag{6.50}$$

*Proof of Lemma 6.9.* Fix an action value function $Q$. We have for any $\epsilon > 0$:

$$\begin{aligned}
\Pr_{\mathcal{D}\sim D^n}\left\{\Delta_Q^{\mathcal{D}} \geq \epsilon\right\} &= \Pr_{\mathcal{D}\sim D^n}\left\{\|Q - Q^*\|_\infty \geq \sqrt{\epsilon} + \left\|Q - \hat{\mathbb{T}}^* Q\right\|_\infty /(1-\gamma)\right\} && (6.51)\\
&\leq \Pr_{\mathcal{D}\sim D^n}\left\{\|Q - \mathbb{T}^* Q\|_\infty \geq (1-\gamma)\left(\sqrt{\epsilon} + \left\|Q - \hat{\mathbb{T}}^* Q\right\|_\infty /(1-\gamma)\right)\right\} && (6.52)\\
&\leq \sum_{s,a} \Pr_{\mathcal{D}\sim D^n}\left\{|Q(s,a) - \mathbb{T}^* Q(s,a)| \geq (1-\gamma)\sqrt{\epsilon} + \left\|Q - \hat{\mathbb{T}}^* Q\right\|_\infty\right\} && (6.53)\\
&\leq \sum_{s,a} \Pr_{\mathcal{D}\sim D^n}\left\{\left|Q(s,a) - \hat{\mathbb{T}}^* Q(s,a)\right| + \left|\hat{\mathbb{T}}^* Q(s,a) - \mathbb{T}^* Q(s,a)\right|\right. \\
&\qquad\qquad \left.\geq (1-\gamma)\sqrt{\epsilon} + \left\|Q - \hat{\mathbb{T}}^* Q\right\|_\infty\right\} && (6.54)\\
&\leq \sum_{s,a} \Pr_{\mathcal{D}\sim D^n}\left\{\left|\hat{\mathbb{T}}^* Q(s,a) - \mathbb{T}^* Q(s,a)\right| \geq (1-\gamma)\sqrt{\epsilon}\right\} && (6.55)\\
&\leq \sum_{s,a} 2e^{-2n_{s,a}(1-\gamma)^2\epsilon/V_{\max}^2} && (6.56)\\
&\leq 2|\mathcal{S}||\mathcal{A}|e^{-2n_{\min}(1-\gamma)^2\epsilon/V_{\max}^2}. && (6.57)
\end{aligned}$$

Line 6.52 follows from Lemma 6.8. Line 6.53 is by the union bound. Line 6.55 is by the definition of infinity norm. Last derivation is by Hoeffding inequality of Line 6.11. The above derivation bounds the CDF of $\Delta_Q^{\mathcal{D}}$. Thus for any $Q$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}\sim D^n}\left[e^{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2}\right] &= 1 + \int_1^\infty \Pr_{\mathcal{D}\sim D^n}\left\{e^{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2} \geq t\right\} dt \\
&= 1 + \int_0^\infty \Pr_{\mathcal{D}\sim D^n}\left\{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2 \geq t\right\} e^t dt \\
&= 1 + \int_0^\infty \Pr_{\mathcal{D}\sim D^n}\left\{\Delta_Q^{\mathcal{D}} \geq \frac{V_{\max}^2 t}{2(n_{\min}-1)(1-\gamma)^2}\right\} e^t dt \\
&\leq 1 + \int_0^\infty 2|\mathcal{S}||\mathcal{A}|e^{-n_{\min}t/(n_{\min}-1)} e^t dt && (6.58)\\
&= 1 + 2|\mathcal{S}||\mathcal{A}|(n_{\min}-1) && (6.59)\\
&\leq 2|\mathcal{S}||\mathcal{A}|n_{\min}. && (6.60)
\end{aligned}$$

Using Markov's inequality, swapping the order of expectations (since $\rho_0$ is a "prior" independent of $\mathcal{D}$), and using the above derivation we get:

$$\Pr_{\mathcal{D}\sim D^n}\left\{\mathop{\mathbb{E}}_{Q\sim\rho_0}\left[e^{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2}\right] \geq \frac{2|\mathcal{S}||\mathcal{A}|n_{\min}}{\delta}\right\} \tag{6.61}$$

$$\leq \frac{\mathop{\mathbb{E}}_{\mathcal{D}\sim D^n}\left[\mathop{\mathbb{E}}_{Q\sim\rho_0}\left[e^{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2}\right]\right]}{2|\mathcal{S}||\mathcal{A}|n_{\min}/\delta} \tag{6.62}$$

$$\leq \frac{\mathop{\mathbb{E}}_{Q\sim\rho_0}\left[\mathop{\mathbb{E}}_{\mathcal{D}\sim D^n}\left[e^{2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2}\right]\right]}{2|\mathcal{S}||\mathcal{A}|n_{\min}/\delta} \leq \delta \tag{6.63}$$

This concludes the proof of Lemma 6.9. □

*Proof of Theorem 6.3.* Lemma 6.9 and Lemma 6.6 together imply Therorem 6.2 by using:

$$\Pr_{\mathcal{D}\sim D^n}\left\{\forall\rho : \mathop{\mathbb{E}}_{Q\sim\rho}\Delta_Q^{\mathcal{D}} \leq \frac{\mathrm{KL}(\rho\|\rho_0) - \ln\delta + \ln|\mathcal{S}| + \ln|\mathcal{A}| + \ln(2n_{\min})}{2(n_{\min}-1)(1-\gamma)^2/V_{\max}^2}\right\} > 1-\delta \tag{6.64}$$

$$f_{\mathcal{D}}(Q) = 2(n_{\min}-1)(1-\gamma)^2\Delta_Q^{\mathcal{D}}/V_{\max}^2, \tag{6.65}$$

on the hypothesis space $\mathcal{T}$:

$$\Pr_{\mathcal{D}\sim D^n}\left\{\exists\rho\in\mathcal{M}(\mathcal{Q}) : \mathop{\mathbb{E}}_{Q\sim\rho}\Delta_Q^{\mathcal{D}} \geq \frac{\mathrm{KL}(\rho\|\rho_0) - \ln\delta + \ln|\mathcal{S}| + \ln|\mathcal{A}| + \ln(2n_{\min})}{2(n_{\min}-1)(1-\gamma)^2/V_{\max}^2}\right\}$$

$$= \Pr_{\mathcal{D}\sim D^n}\left\{\exists\rho\in\mathcal{M}(\mathcal{Q}) : \mathop{\mathbb{E}}_{Q\sim\rho} f_{\mathcal{D}}(Q) - \mathrm{KL}(\rho\|\rho_0) \geq \ln\frac{2|\mathcal{S}||\mathcal{A}|n_{\min}}{\delta}\right\} \tag{6.66}$$

$$\leq \Pr_{\mathcal{D}\sim D^n}\left\{\ln\mathop{\mathbb{E}}_{Q\sim\rho_0} e^{f_{\mathcal{D}}(Q)} \geq \ln\frac{2|\mathcal{S}||\mathcal{A}|n_{\min}}{\delta}\right\} \leq \delta, \tag{6.67}$$

witch concludes the proof of Theorem 6.3. □

### 6.5.3 Proof of Theorem 6.5

The proof follows a similar path as the proofs for the other PAC-Bayesian bounds, but uses concentration inequalities for fast-mixing chains instead of using i.i.d. sample sets. We start by the following lemma:

**Lemma 6.10.** *Let* $\Delta_V^{\mathcal{D}} \overset{\text{def}}{=} (J_{\text{TD}}^{\pi}(V) - \hat{J}_{\text{TD}}^{\pi}(V))_+^2$*. Using the notation of Theorem 6.5, for any fixed prior* $\rho_0$ *over the space of value functions:*

$$\Pr_{\mathcal{D}\sim C_n^{\pi}} \left\{ \mathbb{E}_{V\sim\rho_0} \left[ e^{(n-1)\Delta_Q^{\mathcal{D}}/2\tau V_{\max}^4} \right] \leq \frac{n}{\delta} \right\} > 1 - \delta. \tag{6.68}$$

*Proof of Lemma 6.10.* For any fixed $V$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}\sim C_n^{\pi}} \left[ e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4} \right] &= 1 + \int_1^{\infty} \Pr_{\mathcal{D}\sim C_n^{\pi}} \left\{ e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4} \geq t \right\} dt \\
&= 1 + \int_0^{\infty} \Pr_{\mathcal{D}\sim C_n^{\pi}} \left\{ (n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4 \geq t \right\} e^t dt \\
&= 1 + \int_0^{\infty} \Pr_{\mathcal{D}\sim C_n^{\pi}} \left\{ \Delta_V^{\mathcal{D}} \geq \frac{2\tau V_{\max}^4 t}{n-1} \right\} e^t dt \\
&\leq 1 + \int_0^{\infty} e^{-nt/(n-1)} e^t dt && (6.69) \\
&= n. && (6.70)
\end{aligned}$$

Line 6.69 is due to Lemma 6.4. Using Markov's inequality, swapping the expectation and using the above derivation, we have:

$$\begin{aligned}
\Pr_{\mathcal{D}\sim C_n^{\pi}} \left\{ \mathbb{E}_{V\sim\rho_0} \left[ e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4} \right] \geq \frac{n}{\delta} \right\} &\leq \frac{\delta}{n} \mathbb{E}_{\mathcal{D}\sim C_n^{\pi}} \left[ \mathbb{E}_{V\sim\rho_0} \left[ e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4} \right] \right] && (6.71) \\
&\leq \frac{\delta}{n} \mathbb{E}_{V\sim\rho_0} \left[ \mathbb{E}_{\mathcal{D}\sim C_n^{\pi}} \left[ e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4} \right] \right] && (6.72) \\
&\leq \delta, && (6.73)
\end{aligned}$$

which concludes the proof of the lemma. □

The following intermediate lemma uses the change of measure technique to provide a uniform bound over all distributions.

**Lemma 6.11.** *Using the notation of Theorem 6.5, for any fixed prior $\rho_0$ over the space of value functions:*

$$\Pr_{\mathcal{D}\sim C_n^\pi}\left\{\forall\rho\in\mathcal{M}(\mathcal{V}):\mathop{\mathbb{E}}_{V\sim\rho}\left[J_{\mathrm{TD}}^\pi(V)\right]\le\mathop{\mathbb{E}}_{V\sim\rho}\left[\hat{J}_{\mathrm{TD}}^\pi(V)\right]+\sqrt{\frac{\mathrm{KL}(\rho\|\rho_0)+\ln\frac{n}{\delta}}{(n-1)/2\tau V_{\max}^4}}\right\}>1-\delta. \tag{6.74}$$

*Proof of Lemma 6.11.* Using the definition of $\Delta_V^{\mathcal{D}}$ from Lemma 6.10, we define:

$$f_{\mathcal{D}}(V)=e^{(n-1)\Delta_V^{\mathcal{D}}/2\tau V_{\max}^4}. \tag{6.75}$$

It follows immediately from an application of Lemma 6.6 (change of measure inequality) and using Lemma 6.10:

$$\Pr_{\mathcal{D}\sim C_n^\pi}\left\{\forall\rho\in\mathcal{M}(\mathcal{V}):\mathop{\mathbb{E}}_{V\sim\rho}\left[\Delta_V^{\mathcal{D}}\right]\le\frac{\mathrm{KL}(\rho\|\rho_0)+\ln\frac{n}{\delta}}{(n-1)/2\tau V_{\max}^4}\right\}>1-\delta, \tag{6.76}$$

which concludes the proof. □

*Proof of Theorem 6.5.* For any fixed $V$ we have:

$$\begin{aligned}\|V-V^\pi\|_{\nu(\pi)}^2 &\le \frac{\|V-\mathbb{T}^\pi V\|_{\nu(\pi)}^2}{(1-\gamma)^2} && (6.77)\\ &\le \frac{J_{\mathrm{TD}}^\pi(V)-\Lambda_{\mathrm{TD}}^\pi(V)}{(1-\gamma)^2}. && (6.78)\end{aligned}$$

First line is by $\gamma$-contraction of the Bellman operator on $\ell^2$ norm under the sampling distribution (Bertsekas and Tsitsiklis, 1996), and the second line is by variance decomposition (Antos et al., 2008). Taking expectation on the sampling of $V\sim\rho$, and applying Lemma 6.11 completes the proof. □

## 6.6 Empirical Evaluations

To illustrate the model-selection techniques based on the bounds in the work, we consider model-based and model-free experiments in finite and continuous state-spaces, comparing the results of a frequentist model-section to the results of Bayesian and PAC-Bayesian model-selection.

### 6.6.1 Results on Model-Based PAC-Bayesian RL

The model-based domain is a chain model in which states are ordered by their index. The last state has a reward of 1 and all other states have reward 0. There are two types of actions. One is a stochastic "forward" operation which moves us to the next state in the chain with probability 0.5 and otherwise makes a random transition. The second type is a stochastic "reset" which moves the system to the first state in the chain with probability 0.5 and makes a random transition otherwise. In this domain, we have at each state two actions that do stochastic reset and one action that is a stochastic forward. There are 10 states and $\gamma = 0.9$.

When there are only a few number of sample transitions for each state–action pair, there is a high chance that the frequentist estimate confuses a reset action with a forward. Therefore, we expect a good model-based prior to be useful in this case. We use independent Dirichlets as our prior. We experiment with priors for which the Dirichlet $\alpha$-vector sums up to 10. We define our good prior to have $\alpha$-vectors proportional to the true transition probabilities. A misleading prior is one for which the vector is proportional to a transition model when the actions are switched between forward and reset. A weighted sum between the good and bad priors creates a range of priors that gradually change from being informative to misleading.

We compare the expected regret of three different methods. The empirical method uses the optimal policy with respect to the empirical models. The Bayesian method samples a transition model from the Bayesian Dirichlet posteriors (when the observed counts are added to the prior $\alpha$-vectors) and then uses the optimal policy with respect to the sampled model. The PAC-Bayesian method uses $n_{s,a,s'} + \lambda\alpha^{\text{prior}}_{s,a,s'}$ as the $\alpha$-vector of the posterior, and finds the value of $\lambda \in [0, 1]$, using linear search within values with distance 0.1, that maximizes the lower bound of Theorem 6.2 using $\delta = 0.05$. It then samples from that distribution and uses the optimal policy with respect to the sampled model. The running time for a single run of this method is a few seconds.

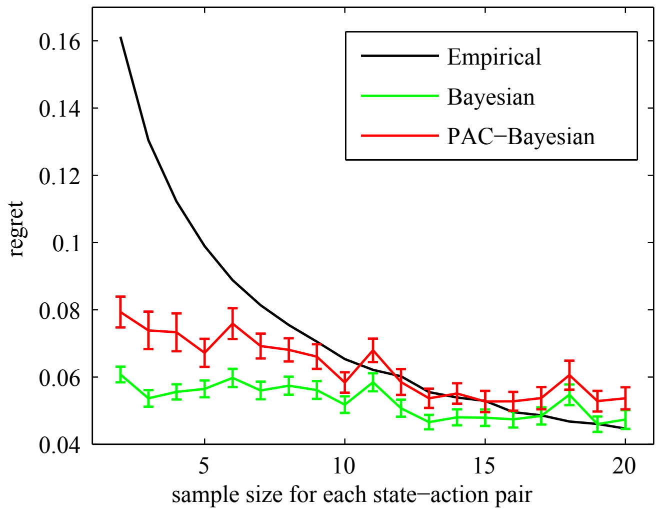


Figure 6–1: Average regrets of different model-based methods with informative prior. Error bars are 1 standard deviation of the mean.

Figure 6–1 shows the comparison between the maximum regret in these methods for different sample sizes when the prior is informative. This is averaged over 50 runs for the Bayesian and PAC-Bayesian methods and 10000 runs for the empirical method (to get a tight estimate of the average). The number of sampled transitions is the same for all state–action pairs. As expected, the Bayesian method outperforms the empirical one for small sample sizes. We can see that the PAC-Bayesian method is closely following the Bayesian one in this case.

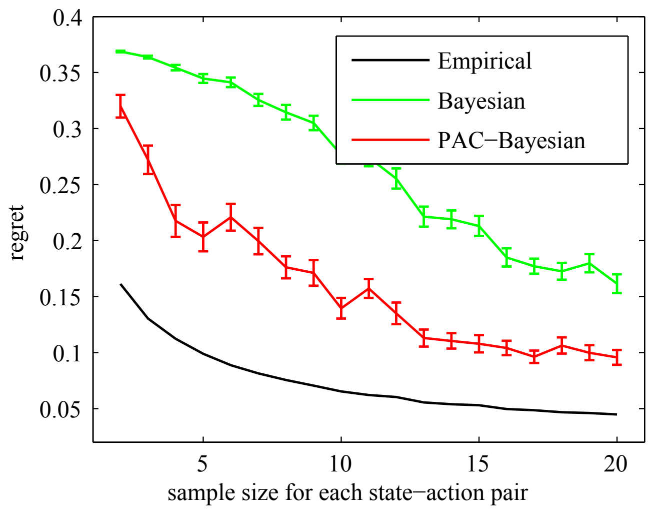


Figure 6–2: Average regrets of different model-based methods with a misleading prior. Error bars are 1 standard deviation of the mean.

With a misleading prior, however, as we can see in Figure 6–2, the empirical method outperforms the Bayesian one. This time, the regret rate of the PAC-Bayesian method follows that of the empirical method.

Figure 6–3 shows how the PAC-Bayesian method switches between following the empirical estimate and the Bayesian posterior as the prior gradually changes

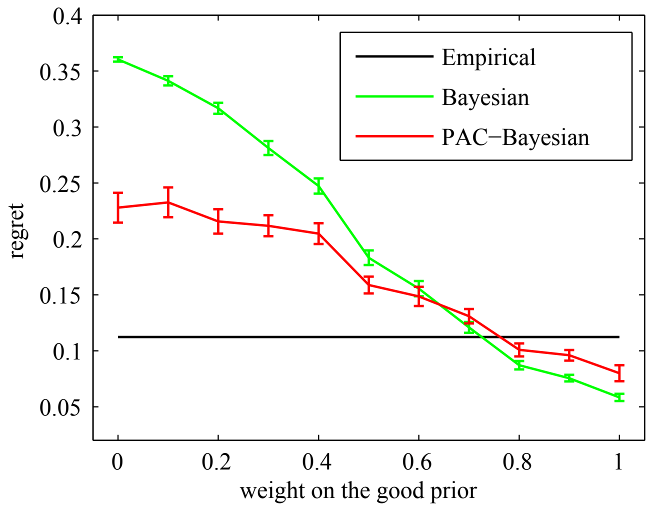
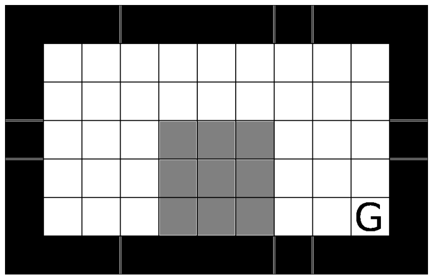


Figure 6–3: Average regrets of different model-based methods by ranging prior from misleading to informative. Error bars are 1 standard deviation of the mean.

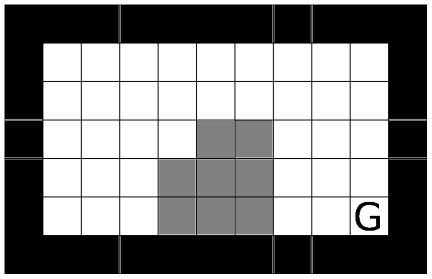
from being misleading to informative (four sample transitions per state action pair). This shows that the bound of Theorem 6.2 is helpful as a model selection technique when the prior cannot be necessarily trusted.

### 6.6.2 Results on Finite-State Model-Free PAC-Bayesian RL

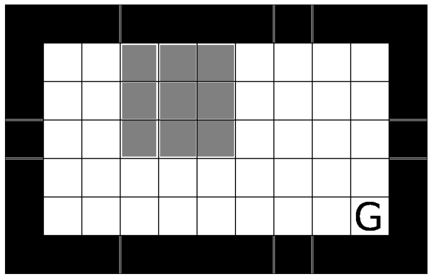
The next experiment is to test the model-free bound of Theorem 6.3. The domain of the experiment is a "puddle world" (Boyan and Moore, 1995). An agent moves around in a grid world of size $5 \times 9$ containing puddles with reward $-1$, an absorbing goal state with reward $+1$, and reward 0 for the remaining states. There are stochastic actions along each of the four cardinal directions that move in the correct direction with probability 0.7 and move in a random direction otherwise. If the agent moves towards the boundary then it stays in its current position.



(a) Initially leaned domain.



(b) Target domain with similar map.



(c) Target domain with unrelated map.

Figure 6–4: Maps of puddle world RL domains. Shaded boxes represent puddles.

We first learn the true value function of a known prior map of the world (Figure 6–4a). We then use that value function as the prior for our model-selection technique on two other environments. One of them is a similar environment where the shape of the puddle is slightly changed (Figure 6–4b). We expect the prior to be informative and useful in this case. The other environment is, however, largely different from the first map (Figure 6–4c). We thus expect the prior to be misleading.

We use independent Gaussians (one over the value of each state–action pair) as the prior, with the initial map's $Q$-values for the mean $\mu_0$, and $\sigma_0^2 = 0.01$ for the variance. The posterior is chosen to be the product of Gaussians with mean

$$\left(\frac{\lambda\mu_0}{\sigma_0^2}+\frac{n\hat{Q}(.,.)}{\hat{\sigma}^2}\right)\Bigg/\left(\frac{\lambda}{\sigma_0^2}+\frac{n}{\hat{\sigma}^2}\right), \tag{6.79}$$

and variance

$$\left(\frac{\lambda}{\sigma_0^2}+\frac{n}{\hat{\sigma}^2}\right)^{-1}, \tag{6.80}$$

where $\hat{\sigma}^2$ is the empirical variance over the TD errors. We sample from this posterior and act according to its corresponding greedy policy.

For $\lambda = 1$, this is an approximate Bayesian posterior for the mean of a Gaussian with known variance. For $\lambda = 0$, the prior is completely ignored (hence we sample from a frequentist confidence interval). We will, however, find the $\lambda \in [0, 1]$ that minimizes the PAC-Bayesian bound of Theorem 6.3 with $\delta = 0.05$. For each method, we report the *maximum regret* of the chosen policy $\pi$ over all state–action pairs: $\|Q^\pi - Q^*\|_\infty$.

Table 6–1: Maximum regret $\|Q^\pi - Q^*\|_\infty$ of different model-selection methods in the puddle world transfer learning problem.

| | Similar Map | Unrelated Map |
|---|---|---|
| Purely Empirical | $0.21 \pm 0.03$ | $\mathbf{0.19 \pm 0.03}$ |
| Bayesian | $\mathbf{0.10 \pm 0.01}$ | $1.16 \pm 0.09$ |
| PAC-Bayesian | $\mathbf{0.12 \pm 0.01}$ | $\mathbf{0.22 \pm 0.04}$ |
| PAC-Bayesian $\lambda$ | $0.58 \pm 0.01$ | $0.03 \pm 0.03$ |

Table 6–1 shows the average over 100 runs of the maximum regret for these methods and the average of the selected $\lambda$, with equal sample size of 20 per state–action pair. Again, it can be seen that the PAC-Bayesian method makes use of the

prior (with higher values of $\lambda$) when the prior is informative, and otherwise follows the empirical estimate (smaller values of $\lambda$). It adaptively balances the usage of the prior based on its consistency over the observed data.

### 6.6.3 Results on Continuous-Domain PAC-Bayesian RL

In this section, we investigate how the bound of Theorem 6.5 can be used in model selection for transfer learning in the RL setting with continuous state-spaces. One experiment is presented on the well-known *Mountain–Car* problem, and the other focuses on a generative model of epileptic seizures built from real-world data.

#### Case Study: Mountain–Car

We design a transfer learning experiment on the Mountain–Car domain (Sutton and Barto, 1998), where the goal is to drive an underpowered car beyond a certain altitude up a mountain. We refer the reader to the reference for details of the domain. We learn the optimal policy (name it $\pi_0^*$) on the original Mountain–Car problem ($\gamma = 0.9$, $R = 1$ passed the goal threshold and 0 otherwise). Note that the reward and the dynamics are deterministic, therefore $\Lambda_{\text{TD}}^{\pi}(V) = \mathbf{0}$ for all $V$. The task is to learn the value function on the original domain, and use that knowledge in similar (though not identical) environments, as a means to accelerate the learning process in those new environments.

We estimate the value of $\pi_0^*$ on the original domain with tile coding (4 tiles of size $8 \times 8$). Let $\theta_0$ be the LSTD solution on a very large sample set in the original domain. To transfer the domain knowledge from this problem, we construct a prior distribution to be a multivariate Gaussians with mean $\theta_0$ and covariance $\sigma_0^2\mathbf{I}$, having $\sigma_0^2 = 0.01$. Note that we are violating the boundedness of $\|\theta\|$ discussed before, but this can be avoided by truncating and scaling the distributions over an $\ell^2$-bounded space. The difference is negligible in practice since the variances are small.

In a new environment, we collect a set of trajectories (100 trajectories of length 5), and search in the space of $\lambda$-parametrized posterior measures, defined as follows: measure $\rho_\lambda$ is a multivariate Gaussians with mean

$$\left(\frac{\lambda\theta_0}{\sigma_0^2} + \frac{\hat{\theta}}{\hat{\sigma}^2}\right) \Bigg/ \left(\frac{\lambda}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2}\right), \tag{6.81}$$

and variance

$$\left(\frac{\lambda}{\sigma_0^2} + \frac{1}{\hat{\sigma}^2}\right)^{-1} \mathbf{I}, \tag{6.82}$$

where $\hat{\theta}$ is the LSTD solution based on the sample set on the new environment, and $\hat{\sigma}^2$ (variance of the empirical estimate) is set to 0.01 based on the number of sampled transitions. The search for the best $\lambda$-parameterized posterior is driven by our proposed PAC-Bayesian upper bound on the approximation error in Theorem 6.5. When $\lambda = 0$, $\rho_\lambda$ will be a purely empirical estimate, whereas when $\lambda = 1$, we get the Bayesian posterior for the mean of a Gaussian with known variance (standard Bayesian inference with empirical priors).

Note that because the Mountain–Car is a deterministic domain, the variance term of Theorem 6.5 is 0. As we use trajectories with known length, we can also upper bound the forgetting constants in the theorem and evaluate the bound completely empirically based on the observed sample set.

We test this model-selection method on two new environments. The first is a mountain domain very similar to the original problem, where we double the effect of the acceleration of the car. The true value function of this domain is close the original domain, and so we expect the prior to be informative (and thus $\lambda$ to be close to 1). In the second domain, we change the reward function such that it decreases, inversely proportional to the car's altitude: $\bar{R}(S) = 1 - h(S)$, where $h(S) \in [0, 1]$ is the normalized altitude at state $S$. The value function of $\pi$ under this reward function is largely different from that of the original one, which means that the prior distribution is misleading, and the empirical estimate should be more reliable (and $\lambda$ close to 0).

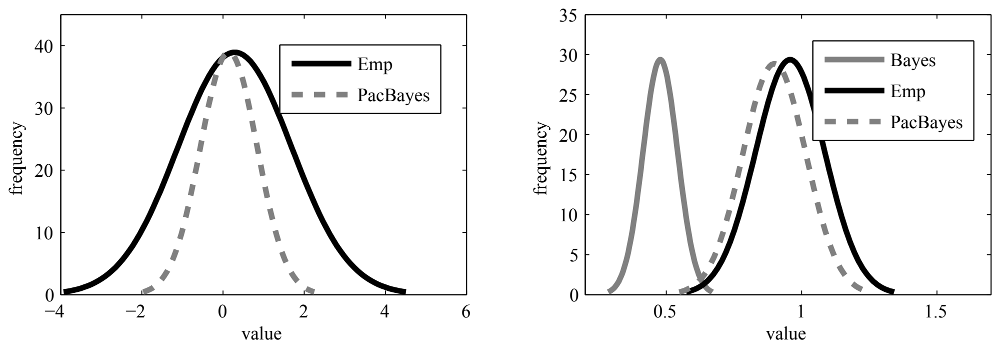
Table 6–2: Error in the estimated value function $\left\|V - V^{\pi_0^*}\right\|^2_{\rho^{\pi_0^*}}$ on the Mountain–Car domain. The last row shows the value of the $\lambda$ parameter selected by the PAC-Bayesian method.

| | Similar Environment | Different Environment |
|---|---|---|
| Purely Empirical | $2.35 \pm 0.12$ | $\mathbf{0.03 \pm 0.01}$ |
| Bayesian | $\mathbf{1.03 \pm 0.09}$ | $2.38 \pm 0.05$ |
| PAC-Bayesian | $\mathbf{1.03 \pm 0.09}$ | $\mathbf{0.07 \pm 0.01}$ |
| PAC-Bayesian $\lambda$ | 1 | $0.06 \pm 0.01$ |

Table 6–2 reports the average true error of approximating $V^{\pi_0^*}$ using different methods over 100 runs (purely empirical method is when $\lambda = 0$, Bayesian is when $\lambda = 1$). For the similar environment, the PAC-Bayesian bound of Theorem 6.5 is minimized consistently with $\lambda = 1$, indicating that the method is fully using the Bayesian prior. The error is thus decreased to less than a half of that of the

empirical estimate. For the environment with largely different reward function, standard Bayesian inference results in poor value prediction, whereas the PAC-Bayesian method is selecting small values of $\lambda$ and is mostly ignoring the prior.

To further investigate how the value function estimate changes with these different methods, we consider an estimate of the value for the state when the car is at the bottom of the hill. This point estimate is constructed from the PAC-Bayesian estimate using the value function obtained by using only the mean of $\rho_\lambda$. To get a sense of the dependence of this estimate on the randomness of the sample the estimate is constructed over 100 runs. We also predict these values using a Bayesian estimate and a purely empirical estimate.



(a) Similar Target Environment. (b) Unrelated Target Environment.

Figure 6–5: Distribution of the estimated value function.

Figure 6–5a shows a normal fit to the histogram of the resulting estimates for the similar environment, for the purely empirical and the PAC-Bayesian estimates. As it can be seen, the distribution of PAC-Bayesian estimates (which coincides with the Bayesian posterior as the best $\lambda$ is consistently 1 in this case) is centered around the correct value, but is more peaked than the empirical distribution. This shows that the method is using the prior to converge faster to the correct value.

Figure 6–5b compares the distribution of the estimated values for the highly different environment. We can see that, as expected, the Bayesian estimate is heavily biased due to the use of a misleading prior. The PAC-Bayes estimate is only slightly biased away from the empirical one with the same variance on the value. Again, this confirms that PAC-Bayes model-selection is largely ignoring the prior when the prior is misleading.

**Case Study: Epilepsy Domain**

We also evaluate our method on a more complex domain. The goal of the RL agent here is to apply direct electrical neuro-stimulation such as to suppress epileptiform behaviour. We use a generative model constructed from real-world data collected on slices of rat brain tissues (Bush et al., 2009); the model is available in the RL-Glue framework. Observations are generated over a 4-dimensional real-valued state-space. The action choice corresponds to selecting the frequency at which neurostimulation is applied. The reward is $-1$ for steps when a seizure is occurring, $-1/40$ for each stimulation pulse, and 0 otherwise. We use $\gamma = 0.8$. The specifics of this reward model is due to the medical side-effects of over-stimulation (Bush et al., 2009).

We first apply the best clinical fixed rate policy (stimulation is applied at a consistent 1Hz) to collect a large sample set (Bush et al., 2009). We then use LSTD to learn a linear value function over the original feature space. Similar to the Mountain–Car experiment, we construct a prior (with a similar mean and variance structure), and use it for knowledge transfer in two new cases. This time, we keep the dynamics and reward function intact and instead change the policy. The first modified policy we consider applies stimulation at a fixed rate of 2Hz; this is expected to have a similar (but slightly worse) value function as the original (1Hz) policy. The other policy we consider applies no stimulation; this is expected to have a very different value function as the seizures are not suppressed.

Table 6–3: The error of value-function estimates $\|V - V^{\pi}\|^2_{\rho^{\pi}}$ on the Epilepsy domain.

| | 2 Hz Stimulation | No Stimulation |
|---|---|---|
| Empirical | $0.0044 \pm 0.0007$ | $0.54 \pm 0.06$ |
| Bayesian | $0.0013 \pm 0.0001$ | $0.86 \pm 0.07$ |
| PAC-Bayesian | $0.0022 \pm 0.0004$ | $0.69 \pm 0.08$ |
| PAC-Bayesian $\lambda$ | $0.62 \pm 0.05$ | $0.30 \pm 0.05$ |

We wish to estimate the parameters $\theta$ of a quadratic value function approximator based on the continuous state features. We sample 10,000 on-policy trajectories of length 1 and use them with the PAC-Bayesian model-selection mechanism described previously (with similar $\lambda$-parametrized posterior family on the $\theta$ parameters) to get estimates of the value function. Table 6–3 summarizes the performance of different methods on the evaluation of the new policies (averaged over 50 runs).

The results are not as polarized as those of the Mountain–Car experiment, partly because the domain is noisier, and because the prior is neither exclusively informative or misleading. Nonetheless, we observe that the PAC-Bayesian method is using the prior more ($\lambda$ averaging around 0.62) in the case of the 2Hz policy, which is consistent with clinical evidence showing that 1Hz and 2Hz stimulations have a similar suppression effects (Bush et al., 2009). The prior is considered less ($\lambda$ averaging around 0.30) for the no-stimulation policy which has substantially (though not entirely) different effects.

## 6.7 Related Work

Model-selection based on error bounds has been studied previously with regularization techniques (Farahmand et al., 2009b,a; Farahmand and Szepesvári, 2011). These bounds are generally tighter for point estimates, as compared to the distributions used in this work. However, our method is more general as it could incorporate arbitrary domain knowledge into the learning algorithm with any type of prior distribution. It can of course use sparsity or smoothness priors, which correspond to well-known regularization methods.

Online PAC RL with Bayesian priors has been addressed with the BOSS algorithm (Asmuth et al., 2009). Unlike our analysis, the correctness of their regret bound is based on the assumption that the true model of the system is sampled from the prior distribution. The correctness of these bounds are thus contingent on the correctness of the prior.

Recently, there has been applications of PAC-Bayesian method to the more restrictive bandit problems, using the policy reweighing technique and Azuma-Hoeffding inequalities (Seldin et al., 2011b,a, 2012). The key difference between their work and our analysis is that their method can be used in online settings to solve the exploration–exploitation problem, while our bounds hold for the more general RL problem with state transitions.

Seldin et al. (2011a) propose a PAC RL algorithm based on a PAC-Bayesian bound with an “oracle prior” placed over the correct action values. However, they point out that their analysis is a first step in the application of PAC-Bayesian techniques in online RL. In fact, there are PAC methods that provide tighter regret bounds than the one proposed by their PAC-Bayesian analysis (Auer and Ortner, 2010). Following the methods discussed in this work, tightening the bounds and extensions to contextual bandits and more general RL settings remain interesting open problems.

## 6.8 Discussion and Future Work

In this chapter, we introduced the first set of PAC-Bayesian bounds for the general batch RL setting. We demonstrate how such bounds can be used to estimate the optimal value function and find a near-optimal policy in finite-state-spaces, given prior distributions over the model parameters or the optimal value function. We also introduced a PAC-Bayesian bound for value function approximation of a fixed policy with continuous states and actions.

Our empirical results show that PAC-Bayesian model-selection uses prior distributions when they are informative, and ignores them when they are misleading. This approach can be used in a number of applications, including transfer learning, as explored in the empirical results.

In using the model-based bound of Theorem 6.2, we expect the running time of searching in the space of parametrized posteriors to increase rapidly with the size of the state-space. A more scalable version would sample models around the posteriors, solve each model, and then use importance sampling to estimate the value of the bound for each possible posterior. This problem does not exist with the finite-state model-free approach of Theorem 6.3, as we do not need to solve the MDP for each sampled model.

Theorem 6.5 gives an upper bound for the estimation error of distributions as opposed to point estimates. While stochastic value functions might be useful in some domains (especially when the use of stochastic policies is not problematic), one might also be interested in bounding the error of some fixed value function. We can derive such bounds by using Jensen's inequality and Fubini's theorem, bounding the error of the point estimate at the mean of the posterior (using $\|\mathbb{E}_{V\sim\mu} V - V^{\pi}\|_{\nu(\pi)}^2 \leq \mathbb{E}_{V\sim\rho} \|V - V^{\pi}\|_{\nu(\pi)}^2$). PAC-Bayesian analysis for the approximation of the optimal value function in continuous domains (i.e. bounding $\mathbb{E}_{V\sim\rho} \|V - V^{*}\|_{\nu}^2$) is also an interesting avenue of future work.

Our PAC-Bayesian analysis presented in this chapter did not address the exploration–exploitation problem in online RL, where the algorithm has to decide on the exploration policy while the training trajectories are being collected. While our worst case analysis in finite state spaces can technically be applied in online RL, it is not going to be any faster than the pure frequentist exploration. Our continuous bound cannot be used in online RL either, as we only analyze the error in our estimate on fixed measures (the distribution of visited states change during the exploration with non-stationary policies).

Extensions of our results to online RL, even with more restrictive models such as contextual bandits instead of MDPs, would open the door to the systematic use of prior information when dealing with the exploration–exploitation problem. With that in mind, we expect that a mix of our PAC-Bayesian analysis in continuous state spaces with those of Seldin et al. (2011b,a) would provide us with extensions for online RL in contextual bandits and more general RL settings. Specifically, the weighted sampling strategy (Sutton and Barto, 1998), also used in the analysis of non-stochastic bandits (Auer et al., 2003), can help derive off-policy bounds and online RL algorithms.

# Part III

# Reinforcement Learning and Compressed Sensing

# CHAPTER 7
# Compressed Sensing and Random Projections

Compressed Sensing (CS) is a topic of active research in the signal processing community and has recently gained attention in machine learning in different domains (Hegde et al., 2007; Wright et al., 2008; Mairal et al., 2008; Maillard and Munos, 2009; Scheinberg and Rish, 2010). The original goal of CS was to compress a signal to lower dimensions (sample input at lower rates), such that the original can be almost perfectly reconstructed from the compressed version (Donoho, 2006; Candès et al., 2006). This compression is usually done via a carefully chosen linear projection from the original space to a lower dimensional space.

When working with large dimensional datasets, one can make use of the regularities present in the data to derive a more compact representation. Such regularities could be in the sparsity of the data (only a few non-zero elements in the feature vector of each instance), or in the data coming from a smooth low-dimensional manifold embedded in the high-dimensional observed feature space. We could also be working with data coming from a finite subset of points in the larger space of features.

Compressed Signal Processing (CSP) introduces a set of methods to deal with high dimensional datasets with such regularities. In this chapter, we review different problems that have been solved with a compressed version of the data. These include signal recovery, detection, classification and estimation. In Chapters 8 and

9, we extend some of these results to compressed regression and their use in RL algorithms with high-dimensional feature spaces.

## 7.1 Concentration of Measure and Stable Embeddings

The analysis of CS starts with the notion of *stable embeddings* in vector spaces. We call $f$ an $\epsilon$-stable embedding of $(\mathcal{U}, \mathcal{V})$ if:

$$\forall \mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}:\ 1 - \epsilon \leq \frac{\|f(\mathbf{u}) - f(\mathbf{v})\|^2}{\|\mathbf{u} - \mathbf{v}\|^2} \leq 1 + \epsilon. \tag{7.1}$$

Such embeddings preserve the pairwise distances within an $\epsilon$ approximation.

Stable embedding in a sparse space, with few non-zero elements for each instance, is closely related to the Restricted Isometry Property (RIP). Define $\mathbb{R}_k^D$ to be the set of all $k$-sparse vectors in $\mathbb{R}^D$:

$$\mathbb{R}_k^D \overset{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^D s.t.\ \|\mathbf{x}\|_0 \leq k\}. \tag{7.2}$$

We say that a projection matrix $\mathbf{\Phi}^{d \times D}$ satisfies RIP of order $k$, if there exists $\epsilon \in (0, 1)$ such that:

$$\forall \mathbf{x} \in \mathbb{R}_k^D:\ 1 - \epsilon \leq \frac{\|\mathbf{\Phi}\mathbf{x}\|^2}{\|\mathbf{x}\|^2} \leq 1 + \epsilon. \tag{7.3}$$

A projection satisfying RIP of order $2k$ is an $\epsilon$-stable embedding of $(\mathbb{R}_k^D, \mathbb{R}_k^D)$, or of $(\mathbb{R}_{2k}^D, \{0\})$.

The existence of stable embedding and their construction is at the center of CSP. The famous Johnson–Lindenstrauss (JL) Lemma, states the existence of compressive stable embeddings for finite sets:

**Lemma 7.1 (Johnson and Lindenstrauss, 1984).** *Let $\mathcal{X} \subset \mathbb{R}^D$ be a finite set of size $p$, and let $\epsilon \in (0, 1)$. There exists a Lipschitz mapping $f : \mathbb{R}^D \to \mathbb{R}^d$, with*

$d = O(\epsilon^{-2} \ln p)$ *such that:*

$$\forall x, x' \in \mathcal{X}: \; 1 - \epsilon \leq \frac{\|f(x) - f(x')\|^2}{\|x - x'\|^2} \leq 1 + \epsilon. \tag{7.4}$$

The lemma states the existence of a stable embedding of $(\mathcal{X}, \mathcal{X})$, for a finite set $\mathcal{X}$. In particular, we can show that random projections are, with high probability, stable embeddings of both finite and sparse spaces.

Several forms of randomly generated projections matrices have been studies in the CS literature. The most common is to sample $\mathbf{\Phi}^{d \times D}$ using i.i.d. Gaussian distributions:

$$\mathbf{\Phi}_{i,j} \sim \mathcal{N}(0, 1/d). \tag{7.5}$$

Independent Bernoulli sampling is another option:

$$\mathbf{\Phi}_{i,j} = \begin{cases} +1/\sqrt{d} & \text{with probability } 1/2, \\ -1/\sqrt{d} & \text{with probability } 1/2. \end{cases} \tag{7.6}$$

Sparser projections have also been used as compressive embeddings:

$$\mathbf{\Phi}_{i,j} = \begin{cases} +3/\sqrt{d} & \text{with probability } 1/6, \\ 0 & \text{with probability } 2/3, \\ -3/\sqrt{d} & \text{with probability } 1/6. \end{cases} \tag{7.7}$$

All the above mentioned random projections follow a form of concentration of measure inequality. Let $\mu_{\mathbf{\Phi}}$ be any of the above distributions on $\mathbf{\Phi}^{D \times d}$. We have for any fixed $\mathbf{x} \in \mathbb{R}^D$:

$$\underset{\mathbf{\Phi} \sim \mu_{\mathbf{\Phi}}}{\mathbb{E}} \left[ \|\mathbf{\Phi x}\|^2 \right] = \|\mathbf{x}\|^2 . \tag{7.8}$$

The random variable $\|\mathbf{\Phi x}\|^2$ is also strongly concentrated about its expected value. For any fixed $\mathbf{x} \in \mathbb{R}^D$ and any $\epsilon \in (0,1)$:

$$\Pr_{\mathbf{\Phi} \sim \mu_{\mathbf{\Phi}}} \left\{ \left| \|\mathbf{\Phi x}\|^2 - \|\mathbf{x}\|^2 \right| \geq \epsilon \|\mathbf{x}\|^2 \right\} \leq 2e^{-\frac{d}{2}(\epsilon^2/2 - \epsilon^3/3)}. \tag{7.9}$$

The concentration of measure inequality for the case of i.i.d. Gaussian sampling of $\mathbf{\Phi}$ can be proved using tail bounds of Gamma random variables, and the other cases are proved by relations to the moments of Gaussian random variables (Achlioptas, 2001).

Using the condition of Equation 7.9, one can prove that random projections of size $O((\ln p - \ln \delta)/\epsilon^2)$ satisfy the JL Lemma with probability greater than $1 - \delta$ (Johnson and Lindenstrauss, 1984; Davenport et al., 2010). One can also use the same concentration bound to show that projections of size $O((k \ln \frac{n}{k} - ln\delta)/\epsilon^2)$ satisfy RIP of order $k$ (Baraniuk et al., 2008). We can see that projections of sizes much smaller than the nominal dimensionality of the space are enough to guarantee a stable embedding both in finite and sparse spaces.

## 7.2 Compressed Recovery

The use of random projections in signal processing was originally motivated by the need to sample and store very efficiently large datasets such as images and video (Candès et al., 2006; Candès and Wakin, 2008). The basic idea is that if the signal is generated as a linear combination of a small set of functions (chosen from a much larger set), then it can be reconstructed very well from a small, fixed number of *randomized* measurements. A solid theoretical foundation has been established for compressed sensing methods, showing that as the number of random measurements increases, the error in the reconstruction decreases at a nearly-optimal rate (Donoho, 2006).

Given the compressed signal $\mathbf{y} \overset{\text{def}}{=} \mathbf{\Phi x}$, the typical reconstruction algorithm returns the vector $\tilde{\mathbf{x}}$ with the smallest $\ell^1$ norm, whose embedding is $\mathbf{y}$ (Chen et al., 1998):

$$\underset{\tilde{\mathbf{x}} \in \mathbb{R}^D}{\operatorname{argmin}} \|\tilde{\mathbf{x}}\|_1 \text{ subject to } \mathbf{\Phi}\tilde{\mathbf{x}} = \mathbf{y}. \tag{7.10}$$

One can show that if the original signal $\mathbf{x}$ is (near) $k$-sparse, accurate reconstruction is possible if the projection satisfies RIP or order $2k$ (Candes, 2008). We can thus consider random projection as compression algorithms, with an $\ell^1$ optimization method used for the decompression.

Signal recovery is also possible when the original data lies on a smooth manifold. The use of random projections helps us avoid the typical non-linear transformation for signal compression, though the recovery of the original signal requires a search on the smooth manifold and might result in costly non-linear optimization problems (Baraniuk and Wakin, 2009).

## 7.3 Compressed Detection and Classification

Compressed projections resulting in stable embeddings can also be used for detection and classification tasks. Consider the signal detection problem of choosing between the two hypothesises:

$$\mathcal{H}_0 \quad : \quad \mathbf{x} = \mathbf{n}, \tag{7.11}$$

$$\mathcal{H}_1 \quad : \quad \mathbf{x} = \mathbf{s} + \mathbf{n}, \tag{7.12}$$

where $\mathbf{s} \in \mathbb{R}^D$ is a known signal, and $\mathbf{n} \in \mathbb{R}^D$ is a vector of i.i.d. Gaussian noise: $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. One can create a test by thresholding the sufficient statistics $\mathbf{x}^T\mathbf{s}$ to distinguish between the two hypothesis, controlling the false positive and detection rates.

Now consider the case where one has to detect the signal when the observation is compressed:

$$\tilde{\mathcal{H}}_0 \quad : \quad \mathbf{y} \overset{\text{def}}{=} \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{n}, \tag{7.13}$$

$$\tilde{\mathcal{H}}_1 \quad : \quad \mathbf{y} \overset{\text{def}}{=} \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}(\mathbf{s} + \mathbf{n}). \tag{7.14}$$

We can again construct a sufficient statistics $\mathbf{y}^T(\mathbf{\Phi}\mathbf{s})$, inner product of the projections, and create a detector by thresholding the statistic. Davenport et al. (2010) show that if the projection satisfies the concentration inequality of Equation 7.9, then the compressed detector has a detection rate similar to the detector in the original space.

Compressed classification is a generalization of the above technique when the signal can be any of a finite set of known values. Define the set of original and compressed hypothesises:

$$\mathcal{H}_i \quad : \quad \mathbf{x} = \mathbf{s}_i + \mathbf{n}, \tag{7.15}$$

$$\tilde{\mathcal{H}}_i \quad : \quad \mathbf{y} \overset{\text{def}}{=} \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}(\mathbf{s}_i + \mathbf{n}), \tag{7.16}$$

for a set of $p$ known signal $\{\mathbf{s}_i\}_{i=1}^p$. Consider a classification algorithm in the original observation space that returns the class $i$ minimizing the statistic $\left\|\mathbf{x}^T - \mathbf{s}_i\right\|$. The corresponding compressed classifier will return $i$ minimizing $\left\|\mathbf{y}^T - \mathbf{\Phi}\mathbf{s}_i\right\|$.

Since random projection are stable embeddings of $(\{\mathbf{x}\}, \{\mathbf{s}_i\}_{i=1}^p)$, the statistics and thus the detection rates are similar for the original and compressed classifiers (Davenport et al., 2010).

## 7.4 Compressed Estimation

Estimation is another task that can be performed with a compressed observation. Specifically, a linear function in a sparse or finite space can be closely approximated by a linear function of the compressed signal. If we observe $\mathbf{y} \overset{\text{def}}{=} \mathbf{\Phi x}$ and want to estimate a linear function $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$, we can use $g(\mathbf{y}) = \mathbf{y}^T(\mathbf{\Phi w})$ in the compressed space (inner product in between the compressed weight vector and the compressed signal). The difference between $f$ and $g$ can be controlled if the projection provides a stable embedding.

The following theorem by Davenport et al. (2006) (later extended to general stable embeddings in Davenport et al. (2010)) provides a bound on the error due to the compression when working with finite sets of input signals.

**Theorem 7.2 (Davenport et al., 2006).** *Let $\mathbf{\Phi}^{d\times D} \sim \mu_{\mathbf{\Phi}}$ be a random projection that satisfies the concentration of measure inequality of Equation 7.9. Let $\mathcal{X} \subset \mathbb{R}^D$ be a finite set of points. Then for any fixed $\mathbf{w} \in \mathbb{R}^D$ and $\epsilon > 0$:*

$$\Pr_{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}}\left\{\forall\mathbf{x}\in\mathcal{X} : \left|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right| \le \epsilon\,\|\mathbf{x}\|\,\|\mathbf{w}\|\right\} \ge 1-(4|\mathcal{X}|+2)e^{-d\epsilon^2/48}. \quad (7.17)$$

This is a linear estimation version of the JL Lemma, as it states that the inner product is preserved between any fixed weight vector and any set of fixed signals. Davenport et al. (2010) provide a generalization of the above theorem when working with stable embeddings with any pair of weight and signal spaces.

In the next chapter we adapt their analysis to sparse vector spaces and use the resulting bound to analyse the error of regression in the compressed space. We further extend these results to the analysis of linear regression with non i.i.d. data collected from a Markovian process.

# CHAPTER 8
# Compressed Regression In Sparse Spaces

Modern machine learning methods have to deal with overwhelmingly large datasets, e.g. for text, sound, image and video processing, as well as for time series prediction and analysis. Much of this data contains very high numbers of features or attributes, sometimes exceeding the number of labelled instances available for training. Even though learning from such data may seem hopeless, in reality, the data often contains structures that can facilitate the development of learning algorithms. In this chapter, we focus on a very common type of structure, in which the instances are sparse, in the sense that a very small percentage of the features in each instance is non-zero (under a known or unknown basis). For example, a text may be encoded as a very large feature vector (millions of dimensions) with each feature being 1 if a corresponding dictionary word is present in the text, and zero otherwise. Hence, in each document, a very small number of features will be non-zero.

Several algorithms, discussed in detail later in the chapter, have been designed to deal with this setting. Here, we focus on a class of methods for learning in large and sparse feature sets using methods originated in compressed sensing and signal processing. Compressed sampling has been studied in the context of supervised regression from two points of view. One idea is to use random projections to compress the dataset, by combining training instances using random projections

(Zhou et al., 2007). Such methods are useful, for instance, when the training set is too large or one has to handle privacy issues.

Another idea is to project each input vector into a lower dimensional space, and then train a predictor in the new compressed space (compression on the feature space). As is typical of dimensionality reduction techniques, this will reduce the variance of most predictors at the expense of introducing some bias. Random projections on the feature space, along with least-squares predictors are studied by Maillard and Munos (2009), and their analysis shows a bias–variance trade-off with respect to on-sample error bounds, which is further extended to bounds on the sampling measure, assuming an i.i.d. sampling strategy.

We start this chapter by providing a bias–variance analysis of ordinary least-squares (OLS) regression in compressed spaces, referred to as Compressed OLS (COLS), when random projections are applied on sparse input feature vectors. Section 8.2 provides an analysis of the worst-case prediction error when samples are i.i.d. and a Gaussian noise model is assumed. In Section 8.3, we provide a more generic *on-measure* bound for data coming from fast mixing chains with dependent noise terms. We use this bound in the next chapter to introduce and analyse a feature generation algorithm for the value estimation problem in RL.

The bounds introduced in this chapter can be used to select the optimal size of the projection, such as to minimize the sum of expected approximation (bias) and estimation (variance) errors. It also provides the means to compare the error of linear predictors in the original and compressed spaces.

## 8.1 Inner Product in Compressed Spaces

Chapter 7 covered various methods and techniques developed for signal processing given a compressed observation of the input signal. This section introduces an extension of the finite-set bound of Theorem 7.2 to $k$-sparse signals. A set $\mathcal{X}_k^D$ is said to be a $k$-sparse space of dimension $D$, if all of its elements have $\ell^0$ norm of at most $k$ under a fixed basis matrix $\mathbf{\Psi}^{D\times D}$:

$$\forall \mathbf{x} \in \mathcal{X}_k^D : \exists \mathbf{z} \in \mathbb{R}_k^D \text{ s.t. } \mathbf{x} = \mathbf{\Psi}\mathbf{z}. \tag{8.1}$$

The following theorem (proved in Section 8.4) provides the error rate of linear estimation when feature signals are in $k$-sparse space:

**Theorem 8.1.** *Let $\mathbf{\Phi}^{d\times D} \sim \mu_{\mathbf{\Phi}}$ be a random projection that satisfies the concentration of measure inequality of Equation 7.9. Let $\mathcal{X}_k^D$ be the $k$-sparse space of dimension $D$, as defined in Equation 8.1. Then for any fixed $\mathbf{w} \in \mathbb{R}^D$ and $0 < \xi < 1$ we have:*

$$\Pr_{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}} \left\{ \forall \mathbf{x} \in \mathcal{X}_k^D : \left| (\mathbf{\Phi}\mathbf{x})^T(\mathbf{\Phi}\mathbf{w}) - \mathbf{x}^T\mathbf{w} \right| \leq \epsilon_{prj} \|\mathbf{x}\| \, \|\mathbf{w}\| \right\} \geq 1 - \xi, \tag{8.2}$$

*where:*

$$\epsilon_{prj} = \sqrt{\frac{48k}{d} \ln \frac{4D+2}{\xi}}. \tag{8.3}$$

Theorem 8.1 states that the inner product between a fixed weight vector and points inside a sparse space is preserved after a stable random projection. It shows that if a function is linear in a sparse space, it is almost linear in an exponentially smaller projected space of size $\tilde{O}(k \ln D)$ with a controllable constant induced bias.

The preservation of linear estimation suggests algorithms that avoid prediction in the observation space and instead work in the compressed space to reduce the variance of the predictor. We study these algorithms in the next sections.

## 8.2 Compressed Linear Regression

We start by analysing the worst case prediction error of the OLS solution. We then proceed to the bias–variance analysis of OLS in the projected space.

In a linear prediction task, we seek to predict a target function $f(\cdot)$ that is assumed to be a (near-) linear function of the input signal $\mathbf{x} \in \mathcal{X}$:

$$f(\mathbf{x}) \overset{\text{def}}{=} \mathbf{x}^T\mathbf{w} + b_f(\mathbf{x}), \text{ where } |b_f(\mathbf{x})| \leq \epsilon_f \|\mathbf{x}\| \|\mathbf{w}\|, \tag{8.4}$$

for some $\epsilon_f > 0$ and some fixed weight vector $\mathbf{w}$. The dependence of the bias term on $\|\mathbf{x}\|$ and $\|\mathbf{w}\|$ allows for scaling of the input and output and retaining the relative relationship with the bias.

Assume we are given a training set of $n$ input–output pairs, consisting of a full-rank input matrix $\mathbf{X}^{n\times D}$, along with noisy observations of $f$:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{b}_f + \eta, \tag{8.5}$$

where $(\mathbf{b}_f)_i = b_f(\mathbf{x}_i)$, and we assume a homoscedastic noise term $\eta$ to be a vector of i.i.d. random variables distributed as $\mathcal{N}(0, \sigma_\eta^2)$.

Given the above, we seek to find a predictor that for any query $\mathbf{x} \in \mathcal{X}$ predicts the target signal $f(\mathbf{x})$. The OLS predictor can be defined using the Moore–Penrose pseudoinverse as follows:

$$\hat{\mathbf{w}} \overset{\text{def}}{=} \mathbf{X}^\dagger\mathbf{y}, \tag{8.6}$$

and the OLS prediction is calculated as follows:

$$\hat{f}(\mathbf{x}) \overset{\text{def}}{=} \mathbf{x}^T\hat{\mathbf{w}} \tag{8.7}$$

If the dimension of the input space is large and we do not have many sample points, then the error of the OLS estimator can be large. One way to deal with this problem is to use feature selection and feature compression to decrease the dimension size and get more accurate estimates of the weights in the new feature space.

Random projections, of course, can be used here for dimensionality reduction. In compressed regression, one first projects the inputs into a lower dimensional space using random projections, then uses the OLS estimator on the compressed input signals as the predictor of the target function. For a random projection $\mathbf{\Phi}^{d\times D}$, we define the COLS solution to be:

$$\hat{\mathbf{w}}_{\mathbf{\Phi}} \overset{\text{def}}{=} (\mathbf{X}\mathbf{\Phi}^T)^{\dagger}\mathbf{y}, \tag{8.8}$$

and the COLS prediction is calculated as follows:

$$\begin{aligned} \hat{f}_{\mathbf{\Phi}}(\mathbf{x}) &\overset{\text{def}}{=} (\mathbf{\Phi}\mathbf{x})^T\hat{\mathbf{w}}_{\mathbf{\Phi}} && (8.9)\\ &= \mathbf{x}^T(\mathbf{\Phi}^T\hat{\mathbf{w}}_{\mathbf{\Phi}}). && (8.10)\end{aligned}$$

Note that after calculating the COLS solution $\hat{\mathbf{w}}_{\mathbf{\Phi}}$, one can calculate the term $\hat{\mathbf{w}} = \mathbf{\Phi}^T\hat{\mathbf{w}}_{\mathbf{\Phi}}$, discard the projection matrix, and use $\mathbf{x}^T\hat{\mathbf{w}}$ as the prediction. $\hat{\mathbf{w}}$ is the corresponding weight vector in the original observation space.

The following theorem (proved in Section 8.4) bounds the error of the compressed predictor for an arbitrary query point.

**Theorem 8.2.** *Let* $\mathbf{\Phi}^{d \times D} \sim \mu_{\mathbf{\Phi}}$ *be a random projection that satisfies the concentration of measure inequality of Equation 7.9. Let the input space* $\mathcal{X}_k^D$ *be a* $k$*-sparse space of dimension* $D$*, as defined in Equation 8.1. Let* $\hat{f}_{\mathbf{\Phi}}(\mathbf{x})$ *be the COLS estimator as defined in Equation 8.9 on a training set following Equation 8.5 with bias bounded by* $\epsilon_f$ *and noise variance* $\sigma_\eta^2$*. For any* $0 < \xi_{prj} < 1$ *and* $0 < \xi_{var} \leq \sqrt{2/e\pi}$*:*

$$\Pr_{\mathbf{\Phi} \sim \mu_{\mathbf{\Phi}}} \Bigg\{ \forall \mathbf{x} \in \mathcal{X}_k^D, \forall \mathbf{X}^T \in (\mathcal{X}_k^D)^n : \Pr_{\eta \sim \mathcal{N}(\mathbf{0}, \sigma_\eta^2 \mathbf{I})} \left\{ \left| f(\mathbf{x}) - \hat{f}_{\mathbf{\Phi}}(\mathbf{x}) \right| \leq \epsilon_{bias} + \epsilon_{var} \right\} \geq 1 - \xi_{var} \Bigg\} \geq 1 - \xi_{prj}, \tag{8.11}$$

*where:*

$$\epsilon_{bias} = (\epsilon_f + \epsilon_{prj}) \Big( \|\mathbf{w}\| \, \|\mathbf{x}\| + \|\mathbf{w}\| \, \|\mathbf{X}\|_F \, \left\| (\mathbf{\Phi}\mathbf{x})^T (\mathbf{X}\mathbf{\Phi}^T)^\dagger \right\| \Big), \tag{8.12}$$

$$\epsilon_{var} = \left\| (\mathbf{\Phi}\mathbf{x})^T (\mathbf{X}\mathbf{\Phi}^T)^\dagger \right\| \sigma_\eta \sqrt{\ln \frac{2}{\pi \xi_{var}^2}}, \tag{8.13}$$

$$\epsilon_{prj} = \sqrt{\frac{48k}{d} \ln \frac{4D+2}{\xi_{prj}}}. \tag{8.14}$$

This theorem can be somewhat simplified by using the properties of random projections. Note that because we use random projections satisfying Equation 7.9, the projection preserves the norm of points in $\mathcal{X}_k^D$, thus $\|\mathbf{\Phi}\mathbf{x}\| \simeq \|\mathbf{x}\|$. Furthermore, the norm of $\mathbf{\Phi}$ can be bounded using the bound discussed by Candès et al.

(2006); we have with probability $1 - \delta_{\mathbf{\Phi}}$:

$$\begin{aligned} \|\mathbf{\Phi}\| &\leq \sqrt{D/d} + \sqrt{(2\log(2/\delta_{\mathbf{\Phi}}))/d} + 1, & (8.15) \\ \left\|\mathbf{\Phi}^{\dagger}\right\| &\leq \left[\sqrt{D/d} - \sqrt{(2\log(2/\delta_{\mathbf{\Phi}}))/d} - 1\right]^{-1}. & (8.16) \end{aligned}$$

Similarly, when $n \gg d$, and the observed features are not heavily correlated, we expect $\left\|(\mathbf{X}\mathbf{\Phi}^T)^{\dagger}\right\|$ to be of order $\tilde{O}(\sqrt{d/\left\|\mathbf{X}\right\|_F})$. Assume $\epsilon_f = 0$ and that $\|\mathbf{x}\| = 1$ for all points in the input space. Ignoring the terms constant in $d$, we can rewrite the bound on the error up to logarithmic terms as:

$$\tilde{O}\left(\sqrt{k\log(D/k)}\frac{1}{\sqrt{d}}\right) + \tilde{O}\left(\frac{\sigma_\eta}{\sqrt{n}}\sqrt{d}\right).$$

The first term is a part of the bias due to the projection (excess approximation error). The second term is the variance term that shrinks with larger training sets (estimation error). We clearly observe the trade-off with respect to the compressed dimension $d$. With the assumptions discussed above, the optimal projection size is thus of order $\tilde{O}(\sqrt{kn})$, which resembles the suggested size of $\tilde{O}(\sqrt{n})$ for on-measure error minimization discussed in Maillard and Munos (2009).

Theorem 8.2 provides a simultaneous worst-case bound on the entire input space with no distributional assumptions on the input. Not surprisingly, this type of bound is often loose when $\mathbf{x}$ and $\mathbf{X}$ are sampled from similar distributions and the training set is independently sampled. There is an unavoidable $\tilde{O}(\sqrt{k})$ additive constant in the bound that results form a worst-case analysis.

In the fixed design case, where $\mathbf{x}$ is sampled uniformly from a fixed training set $\mathbf{X}$, one can get much tighter bounds. Such bounds however, fail to provide off-sample guarantees and might require very strong distributional assumptions. Maillard and Munos (2009, 2012) provide discussions on how random projections can be applied in a random-design settings using JL type bounding on elements. Detailed analysis of the random design setting is provided by Hsu et al. (2012).

## 8.3 Compressed Regression with Correlated Training Data

The worst case bound of Theorem 8.2 is theoretically interesting, but loose when the distribution of the data and the query point are the same. The i.i.d. assumption on the data might be too strong when analysing time dependent datasets, such as the ones encountered in RL problems. In this section, we look at the case where the datapoints in $\mathbf{X}$ are coming from a fast-mixing Markov chain. In the next chapter, we use this analysis to bound the error of a value function approximator that uses compressed regression at its core.

This time, we assume a bias-free response model with target $f(\mathbf{x}) \overset{\text{def}}{=} \mathbf{x}^T\mathbf{w}$:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \eta. \tag{8.17}$$

The sequence of $(\mathbf{x}_i, \eta_i)$ is assumed to come from a Markov chain $C_n$ with forgetting constant $\tau$. We assume the noise term $\eta$ is a bounded martingale difference sequence:

$$\forall i : |\eta_i| \leq \eta_{\max} \text{ and } \mathbb{E}\left[\eta_i | \mathbf{x}_{1:i}, \eta_{1:i-1}\right] = 0. \tag{8.18}$$

A biased target with sub-Gaussian noise might also be analysed with the same methodology presented here, though we assume a bounded noise and zero bias to simplify the derivations and proofs.

The following theorem (proved in Section 8.4) provides a bound on the error of COLS when the training data is coming from a fast-mixing Markov chain:

**Theorem 8.3.** *Let $\mathbf{\Phi}^{d\times D} \sim \mu_{\mathbf{\Phi}}$ with $D > d \geq 12$ be a random projection that satisfies the concentration of measure inequality of Equation 7.9. Let the input space $\mathcal{X}_k^D$ be a k-sparse space of dimension D, as defined in Equation 8.1. Let $\hat{f}_{\mathbf{\Phi}}(\mathbf{x})$ be the COLS estimator as defined in Equation 8.9 on a training set following Equation 8.17. Assume $\{(\mathbf{x}_i, \eta_i)\}_{i=1}^n$ is generated from a Markov chain $C_n$ with forgetting time $\tau$, stationary distribution $\rho(\mathbf{x})$ and noise model of Equation 8.18. For any $0 < \xi < 1/4$, if $n \geq 4\,\frac{\sup_{\mathbf{x}\in\mathcal{X}_k^D}\|\mathbf{x}\|^2}{\inf_{\mathbf{x}\in\mathcal{X}_k^D}\|\mathbf{x}\|^2}\ln\frac{8}{\xi}$:*

$$\Pr_{\substack{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}} \\ \{(\mathbf{x}_i,\eta_i)\}_{i=1}^n\sim C_n}} \left\{ \left\| \hat{f}_{\mathbf{\Phi}}(\mathbf{x}) - f(\mathbf{x}) \right\|_\rho \leq \epsilon_{bias} + \epsilon_{var} \right\} \geq 1 - \xi_{prj}, \tag{8.19}$$

*defining:*

$$\begin{aligned} \epsilon_{bias} &= 10\,\alpha\epsilon_{prj}\,\|\mathbf{w}\|\,\|\mathbf{x}\|_\rho\sqrt{\frac{1}{d\nu}}, & (8.20)\\ \epsilon_{var} &= 2\,\alpha\epsilon_{prj}\,\|\mathbf{w}\|\sqrt{\frac{d\tau}{n\nu}\log\frac{9d}{\xi}} + \sqrt{8}\,\alpha\eta_{\max}\sqrt{\frac{\kappa d}{n\nu}\log\frac{8d}{\xi}} & (8.21)\\ \epsilon_{prj} &= \sqrt{\frac{48k}{d}\ln\frac{16D+8}{\xi}}, & (8.22)\\ \alpha &= \max\left(1, \max_{\mathbf{x}\in\mathcal{X}_k^D}\|\mathbf{\Phi}\mathbf{x}\|/\|\mathbf{x}\|\right), & (8.23) \end{aligned}$$

*where $\kappa$ and $\nu$ are the condition number and the smallest positive eigenvalue of the normalized empirical gram matrix $\frac{1}{\|\mathbf{X}\|_F}\mathbf{\Phi}\mathbf{X}^T\mathbf{X}\mathbf{\Phi}^T$.*

Theorem 8.3 can be further simplified by using concentration bounds on the operator norm of random projections. When $n \ll d$ and the features are not heavily correlated, we expect the smallest and biggest singular values of $\mathbf{X\Phi}^T / \|\mathbf{X}\|_F$ to be of order $\tilde{O}(\sqrt{1/d})$, since $\mathbf{\Phi}$ projects the norm approximately uniformly on all dimensions. Thus we have $\kappa = \tilde{O}(1)$ and $\nu = \tilde{O}(1/d)$. Projections are also norm-preserving when applied to the sparse space, and thus $\alpha = O(1)$ when $d = \tilde{O}(k \ln D)$. Assuming that $n = \tilde{O}(d^2)$, we can rewrite the bound on the error up to logarithmic terms as:

$$\tilde{O}\left( \|\mathbf{w}\| \, \|\mathbf{x}\|_\rho \sqrt{\frac{k \log D}{d}} \right) + \tilde{O}\left( \frac{d}{\sqrt{n}} \right). \tag{8.24}$$

The first term is a part of the bias due to the projection (excess approximation error). The rest is the combined variance terms that shrink with larger training sets (estimation error). We clearly observe the trade-off with respect to the compressed dimension $d$. With the assumptions discussed above, we can see that projection of size $d = \tilde{O}(k \log D)$ should be enough to guarantee arbitrarily small bias, as long as $\|\mathbf{w}\| \, \|\mathbf{x}\|_\rho$ is small.

Note that this bound matches that of Ghavamzadeh et al. (2010). The variance term is of order $\sqrt{d/n\nu}$. Thus, the dependence on the smallest eigenvalue of the gram matrix makes the variance term order $d/\sqrt{n}$ rather than the expected $\sqrt{d/n}$. We expect the use of ridge regression instead of OLS to remove this dependency and help with the convergence rate. The analysis of ridge regression within compressed spaces is an interesting topic of future work. One can make use of the recent developments in the analysis of ridge regression in random design settings (Hsu et al., 2012) and extend those results to the compressed regression case with sparse input features.

## 8.4 Proof of Compressed Regression Bounds

The proofs included in this section are based on the corresponding analysis in Fard et al. (2012) and Fard et al. (2013).

### 8.4.1 Proof of Theorem 8.1

*Proof of Theorem 8.1.* Let $\mathbf{\Psi}$ be the basis matrix under which $\mathcal{X}_k^D$ is sparse. Let $\mathbf{\Psi}_{.,i}$ be the $i$-th column of $\mathbf{\Psi}$. Using Theorem 7.2 over the finite set of signals $\{\mathbf{\Psi}_{.,i}\}_{i=1}^D$, and having $\|\mathbf{\Psi}_{.,i}\| = 1$, we get:

$$\Pr_{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}}\left\{\forall i : \left|(\mathbf{\Phi\Psi}_{.,i})^T(\mathbf{\Phi w}) - \mathbf{\Psi}_{.,i}\mathbf{w}\right| \leq \frac{\epsilon_{\text{prj}}}{\sqrt{k}}\|\mathbf{w}\|\right\} \geq 1-(4D+2)e^{-d\epsilon_{\text{prj}}^2/48k}. \quad (8.25)$$

By definition (see Equation 8.1), for any $\mathbf{x} \in \mathcal{X}_k^D$, we must have $\mathbf{x} = \sum_{i=1}^D \mathbf{z}_i \mathbf{\Psi}_{.,i}^T$ with $\mathbf{z} \in \mathbb{R}_k^D$. Thus with probability no less than $(4D+2)e^{-d\epsilon_{\text{prj}}^2/48k}$ for all $\mathbf{x} \in \mathcal{X}_k^D$:

$$\begin{aligned}\left|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right| &= \left|\sum_{i=1}^D \mathbf{z}_i((\mathbf{\Phi\Psi}_{.,i})^T(\mathbf{\Phi w}) - \mathbf{\Psi}_{.,i}\mathbf{w})\right| && (8.26)\\ &\leq \sum_{i=1}^D \left|\mathbf{z}_i((\mathbf{\Phi\Psi}_{.,i})^T(\mathbf{\Phi w}) - \mathbf{\Psi}_{.,i}\mathbf{w})\right| && (8.27)\\ &\leq \frac{\epsilon_{\text{prj}}}{\sqrt{k}}\|\mathbf{w}\|\sum_{i=1}^D |\mathbf{z}_i|. && (8.28)\end{aligned}$$

There are at most $k$ non-zero $\mathbf{z}_i$ and $\sum_{i=1}^D \mathbf{z}_i^2 = \|\mathbf{x}\|$. Therefore we must have $\sum_{i=1}^D |\mathbf{z}_i| \leq \sqrt{k}\|\mathbf{x}\|$. We thus get:

$$\Pr_{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}}\left\{\forall \mathbf{x} \in \mathcal{X}_k^D : \left|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right| \leq \epsilon_{\text{prj}}\|\mathbf{x}\|\|\mathbf{w}\|\right\} \geq 1-(4D+2)e^{-d\epsilon_{\text{prj}}^2/48k}, \quad (8.29)$$

which proves the Theorem by solving for $\xi$. □

### 8.4.2 Proof of Theorem 8.2

*Proof of Theorem 8.2.* Recall that by definition $f(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{x}^T\mathbf{w} + b_f(\mathbf{x})$. Define the bias due to projection as follows:

$$b_{\text{prj}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{x}^T\mathbf{w} - (\mathbf{\Phi x})^T(\mathbf{\Phi w}) \tag{8.30}$$

Using Theorem 8.1 we get that:

$$\mathop{\mathbb{P}\text{r}}_{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}} \left\{\forall \mathbf{x} \in \mathcal{X}_k^D : |b_{\text{prj}}(\mathbf{x})| \le \epsilon_{\text{prj}} \|\mathbf{x}\| \|\mathbf{w}\|\right\} \ge 1 - \xi_{\text{prj}}, \tag{8.31}$$

Adding the bias into the training set in Equation 8.5, we have the projected training set:

$$\mathbf{y} = (\mathbf{X\Phi})^T(\mathbf{\Phi w}) + \mathbf{b}_{\text{prj}} + \mathbf{b}_f + \eta, \tag{8.32}$$

such that for all $i$: $|(\mathbf{b}_{\text{prj}})_i| \le \epsilon_{\text{prj}} \|\mathbf{x}_i\| \|\mathbf{w}\|$ and $|(\mathbf{b}_f)_i| \le \epsilon_f \|\mathbf{x}_i\| \|\mathbf{w}\|$. Combining the bias terms we have:

$$\mathbf{y} = (\mathbf{X\Phi})^T(\mathbf{\Phi w}) + \mathbf{b} + \eta, \tag{8.33}$$

where we have $\forall i : |\mathbf{b}_i| \le (\epsilon_f + \epsilon_{\text{prj}}) \|\mathbf{x}_i\| \|\mathbf{w}\|$. Therefore for the COLS solution we have:

$$\begin{aligned}
\hat{\mathbf{w}}_{\mathbf{\Phi}} &= (\mathbf{X\Phi}^T)^\dagger \mathbf{y} &(8.34)\\
&= (\mathbf{X\Phi}^T)^\dagger \left((\mathbf{X\Phi})^T(\mathbf{\Phi w}) + \mathbf{b} + \eta\right) &(8.35)\\
&= \mathbf{\Phi w} + (\mathbf{X\Phi}^T)^\dagger \mathbf{b} + (\mathbf{X\Phi}^T)^\dagger \eta. &(8.36)
\end{aligned}$$

Thus for all $\mathbf{x} \in \mathcal{X}$ we have the error:

$$\left|\hat{f}(\mathbf{x}) - f_{\mathbf{\Phi}}(\mathbf{x})\right| = \left|(\mathbf{\Phi}\mathbf{x})^T \hat{\mathbf{w}}_{\mathbf{\Phi}} - (\mathbf{x}^T\mathbf{w} + b_f(\mathbf{x}))\right| \tag{8.37}$$

$$\leq \left|(\mathbf{\Phi}\mathbf{x})^T \left(\mathbf{\Phi}\mathbf{w} + (\mathbf{X}\mathbf{\Phi}^T)^\dagger \mathbf{b} + (\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\right) - \mathbf{x}^T\mathbf{w}\right| + \epsilon_f \left\|\mathbf{x}\right\| \left\|\mathbf{w}\right\| \tag{8.38}$$

$$\leq \left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \mathbf{b}\right| + \left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\right| + \left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{\Phi}\mathbf{w}) - \mathbf{x}^T\mathbf{w}\right| + \epsilon_f \left\|\mathbf{x}\right\| \left\|\mathbf{w}\right\| \tag{8.39}$$

$$\leq \left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \mathbf{b}\right| + \left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\right| + (\epsilon_{\text{prj}} + \epsilon_f) \left\|\mathbf{x}\right\| \left\|\mathbf{w}\right\|. \tag{8.40}$$

Line 8.38 uses the bound on the original bias $b_f$, and line 8.40 uses the bound on $b_{\text{prj}}$ described in Equation 8.31. For the first term of line 8.40 (part of prediction bias) on the right hand side, using $\forall i : |\mathbf{b}_i| \leq (\epsilon_f + \epsilon_{\text{prj}}) \left\|\mathbf{x}_i\right\| \left\|\mathbf{w}\right\|$ we have:

$$\left|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \mathbf{b}\right| \leq \left\|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger\right\| \left\|\mathbf{b}\right\| \tag{8.41}$$

$$\leq \left\|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger\right\| \left\|\mathbf{X}\right\|_F \left\|\mathbf{w}\right\| (\epsilon_f + \epsilon_{\text{prj}}). \tag{8.42}$$

For the second term in line 8.40 (prediction variance), we have that the expectation of $\underset{\eta\sim\mathcal{N}(\mathbf{0},\sigma_\eta^2\mathbf{I})}{\mathbb{E}}\left[(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\right] = 0$, as $\eta$ is independent of data and its expectation is $\mathbf{0}$. We also know that it is a weighted sum of normally distributed random variables, and thus is normal with the variance:

$$\underset{\eta\sim\mathcal{N}(\mathbf{0},\sigma_\eta^2\mathbf{I})}{\mathbb{V}\text{ar}}\left[(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\right] = \underset{\eta\sim\mathcal{N}(\mathbf{0},\sigma_\eta^2\mathbf{I})}{\mathbb{E}}\left[(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger \eta\eta^T((\mathbf{X}\mathbf{\Phi}^T)^\dagger)^T\mathbf{\Phi}\mathbf{x}\right] \tag{8.43}$$

$$= \sigma_\eta^2(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger((\mathbf{X}\mathbf{\Phi}^T)^\dagger)^T\mathbf{\Phi}\mathbf{x} \tag{8.44}$$

$$= \sigma_\eta^2 \left\|(\mathbf{\Phi}\mathbf{x})^T(\mathbf{X}\mathbf{\Phi}^T)^\dagger\right\|^2, \tag{8.45}$$

where in line 8.44 we used the i.i.d. assumption on the noise. Thereby we can bound $\left|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right|$ by the tail probability of the normal distribution as needed. Using a standard upper bound on the tail probability of normals, for any $0 < \xi_{\text{var}} \le \sqrt{2/e\pi}$:

$$\Pr_{\eta\sim\mathcal{N}(\mathbf{0},\sigma_\eta^2\mathbf{I})}\left\{\left|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right| \le \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\right\|\sigma_\eta\sqrt{\ln\frac{2}{\pi\xi_{\text{var}}^2}}\right\} \ge 1-\xi_{\text{var}}. \quad (8.46)$$

Substituting regression to the bias and the variance in Equation 8.40 proves the lemma. □

### 8.4.3 Proof of Theorem 8.3

We start by developing the following intermediate concentration lemmas based on Theorem 4.4 for fast mixing Markov chains:

**Lemma 8.4.** *Let $f$ be a measurable function on $\mathcal{X}$ whose values lie in $[0,b]$, $Z_{1:n} \sim C_n \in \mathcal{M}(\mathcal{X}^n)$ be a homogeneous Markov chain, taking values in $\mathcal{X}$ with forgetting time $\tau$. Let $\bar f = \frac{1}{n}\sum_{i=1}^n f(Z_i)$. For all $0<\xi<1$:*

$$\Pr_{Z_{1:n}\sim C_n}\left\{\bar f < 2\,\mathbb{E}\,\bar f + \frac{4b\tau}{n}\log\frac{1}{\xi}\right\} \ge 1-\xi. \quad (8.47)$$

*Proof.* Since $\mathbb{E}\,[z] \ge 0$, using Theorem 4.4 we have for any $\epsilon > 0$:

$$\begin{aligned}
\Pr\left\{\bar f - 2\,\mathbb{E}\,\bar f \ge \epsilon\right\} &= \Pr\left\{\bar f - \mathbb{E}\,\bar f \ge \mathbb{E}\,\bar f + \epsilon\right\} && (8.48)\\
&\le \exp\left(-\frac{(\mathbb{E}\,\bar f+\epsilon)^2\,n}{2b\tau(2\,\mathbb{E}\,\bar f+\epsilon)}\right) && (8.49)\\
&\le \exp\left(-\frac{(\mathbb{E}\,\bar f+\epsilon)\,n}{4b\tau}\right) && (8.50)\\
&\le \exp\left(-\frac{\epsilon\, n}{4b\tau}\right). && (8.51)
\end{aligned}$$

The lemma follows by solving for $\epsilon$. $\square$

**Lemma 8.5.** *Under the same conditions as in Lemma 8.4, for any* $0 < \xi < 1$*:*

$$\Pr_{Z_{1:n} \sim C_n} \left\{ \mathbb{E}\,\bar{f} < 2\bar{f} + \frac{8b\tau}{n} \log \frac{1}{\xi} \right\} \geq 1 - \xi. \tag{8.52}$$

*Proof.* Since $\mathbb{E}\,[z] \geq 0$, using Theorem 4.4 we have for any $\epsilon > 0$:

$$\begin{aligned} \Pr \left\{ \mathbb{E}\,\bar{f} - 2\bar{f} \geq \epsilon \right\} &= \Pr \left\{ \mathbb{E}\,\bar{f} - \bar{f} \geq (\mathbb{E}\,\bar{f} + \epsilon)/2 \right\} &(8.53)\\ &\leq \exp\left( -\frac{(\mathbb{E}\,\bar{f} + \epsilon)^2\, n}{8b\tau\, \mathbb{E}\,\bar{f}} \right) &(8.54)\\ &\leq \exp\left( -\frac{(\mathbb{E}\,\bar{f} + \epsilon)\, n}{8b\tau} \right) &(8.55)\\ &\leq \exp\left( -\frac{\epsilon\, n}{8b\tau} \right). &(8.56) \end{aligned}$$

The lemma follows by solving for $\epsilon$. $\square$

**Error Decomposition**

We use a similar decomposition of the error as in the proof of Theorem 8.2. Using Theorem 8.1 on the space $\mathcal{X}_k^D$ we have:

$$\mathbf{y} = (\mathbf{X}\mathbf{\Phi}^T)(\mathbf{\Phi}\mathbf{w}) + \mathbf{b} + \eta, \tag{8.57}$$

such that for all $i$: $|\mathbf{b}_i| \leq \epsilon_{\text{prj}}\, \|\mathbf{x}_i\|\, \|\mathbf{w}\|$ with probability $1 - \xi/4$ (with the definition of $\epsilon_{\text{prj}}$ in Theorem 8.3). $\mathbf{b}$ is the vector of bias due to the projection. The weighted

$\ell^2$ error in regression will thus be:

$$\begin{aligned}\left\|\hat{f}_{\mathbf{\Phi}}(\mathbf{x}) - f(\mathbf{x})\right\|_\rho &= \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\left((\mathbf{X\Phi}^T)(\mathbf{\Phi w}) + \mathbf{b} + \eta\right) - \mathbf{x}^T\mathbf{w}\right\|_\rho \\ &= \left\|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) + (\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\mathbf{b} + (\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta - \mathbf{x}^T\mathbf{w}\right\|_\rho \\ &\le \left\|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right\|_\rho + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\mathbf{b}\right\|_\rho + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right\|_\rho \\ &\le \epsilon_{\text{prj}}\left\|\mathbf{x}\right\|_\rho\left\|\mathbf{w}\right\| + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\mathbf{b}\right\|_\rho + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right\|_\rho \quad (8.58)\end{aligned}$$

The second term of Equation 8.58 is the regression to the bias, and the third term is the regression to the noise. We next present lemmas that bound these terms. Throughout this section we use the notation $\left\|\cdot\right\|_n \overset{\text{def}}{=} \left\|\cdot\right\|_{U(\{\mathbf{x}_i\})}$ to be the weighted $\ell^2$ norm uniform on the sample set $X$.

**Bounding the Regression to Bias Terms**

**Lemma 8.6 (Bounding regression to the bias).** *Assume the conditions of Theorem 8.3 and assume* $\forall \mathbf{x} \in \mathcal{X}_k^D : \left|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right| \le \epsilon_{prj}\left\|\mathbf{x}\right\|\left\|\mathbf{w}\right\|$*. Let* $\mathbf{b}$ *be the bias vector in Equation 8.57. For any* $0 < \xi < 1$*:*

$$\Pr_{\{(\mathbf{x}_i,\eta_i)\}_{i=1}^n \sim C_n}\left\{\left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\mathbf{b}\right\|_\rho \le \alpha\epsilon_{prj}\left\|\mathbf{w}\right\|\left(\left\|\mathbf{x}\right\|_\rho + 4.3\left\|\mathbf{x}\right\|_n\right)\sqrt{\frac{1}{d\nu}} + \alpha\epsilon_{prj}\left\|\mathbf{w}\right\|\sqrt{\frac{4d\tau}{n\nu}\log\frac{9d}{\xi}}\right\} \ge 1 - \xi/4. \quad (8.59)$$

*Proof.* Define $\mathbf{w}_b = (\mathbf{X\Phi})^\dagger\mathbf{b}$. We start by bounding the empirical norm $\left\|(\mathbf{\Phi x})^T\mathbf{w}_b\right\|_n$. Given that $\mathbf{X\Phi}^T\mathbf{w}_b$ is the OLS regression to the bias on the observed points, its sum of squared errors should not be greater than any other linear regression, including the vector $\mathbf{0}$, and thus $\left\|(\mathbf{\Phi x})^T\mathbf{w}_b - b_{\text{prj}}(\mathbf{x})\right\|_n \le \left\|b_{\text{prj}}(\mathbf{x})\right\|_n$.

Therefore we get:

$$\begin{aligned}
\left\|(\mathbf{\Phi x})^T \mathbf{w}_b\right\|_n &\leq \left\|(\mathbf{\Phi x})^T \mathbf{w}_b - b_{\text{prj}}(\mathbf{x})\right\|_n + \|b_{\text{prj}}(\mathbf{x})\|_n && (8.60)\\
&\leq 2\,\|b_{\text{prj}}(\mathbf{x})\|_n && (8.61)\\
&\leq 2\epsilon_{\text{prj}}\,\|\mathbf{w}\|\,\|\mathbf{x}\|_n\,. && (8.62)
\end{aligned}$$

Let $\mathcal{W} = \{\mathbf{u} \in \mathbb{R}^d \text{ s.t. } \|\mathbf{u}\| \leq 1\}$. Let $\mathcal{W}_\epsilon \subset \mathcal{W}$ be an $\epsilon$-grid cover of $\mathcal{W}$:

$$\forall \mathbf{v} \in \mathcal{W}:\ \exists \mathbf{u} \in \mathcal{W}_\epsilon \text{ s.t. } \|\mathbf{u} - \mathbf{v}\| \leq \epsilon. \tag{8.63}$$

It is easy to prove (see e.g. Chapter 13 of Lorentz et al. (1996)) that these conditions can be satisfied by choosing a grid of size $|\mathcal{W}_\epsilon| \leq (3/\epsilon)^d$ ($\mathcal{W}_\epsilon$ fills up the space within $\epsilon$ distance). Applying union bound to Lemma 8.5 with $f(\mathbf{x}) = ((\mathbf{\Phi w})^T\mathbf{u})^2$ for all elements in $\mathcal{W}_\epsilon$, we get with probability no less than $1 - \xi/4$, for all $\mathbf{u} \in \mathcal{W}_\epsilon$:

$$\left\|(\mathbf{\Phi x})^T\mathbf{u}\right\|_\rho^2 \leq 2\left\|(\mathbf{\Phi x})^T\mathbf{u}\right\|_n^2 + \frac{8\alpha^2\tau}{n}\log\frac{4|\mathcal{W}_\epsilon|}{\xi}, \tag{8.64}$$

which yields the following after simplification:

$$\left\|(\mathbf{\Phi x})^T\mathbf{u}\right\|_\rho \leq 2\left\|(\mathbf{\Phi x})^T\mathbf{u}\right\|_n + \alpha\sqrt{\frac{8\tau}{n}\log\frac{4|\mathcal{W}_\epsilon|}{\xi}}. \tag{8.65}$$

Let $\mathbf{w}'_b = \mathbf{w}_b / \|\mathbf{w}_b\|$. For any $\mathbf{X}$, since $\mathbf{w}'_b \in \mathcal{W}$, there exists $\mathbf{w}'' \in \mathcal{W}_\epsilon$ such that $\|\mathbf{w}'_b - \mathbf{w}''\| \leq \epsilon$. Therefore, under event in Equation 8.65, with the definition of $\alpha$

in Theorem 8.3 we have:

$$\begin{aligned}
\frac{\left\|(\mathbf{\Phi x})^T\mathbf{w}_b\right\|_\rho}{\|\mathbf{w}_b\|} &= \left\|(\mathbf{\Phi x})^T\mathbf{w}_b'\right\|_\rho && (8.66)\\
&\leq \left\|(\mathbf{\Phi x})^T(\mathbf{w}_b'-\mathbf{w}'')\right\|_\rho + \left\|(\mathbf{\Phi x})^T\mathbf{w}''\right\|_\rho && (8.67)\\
&\leq \|\mathbf{\Phi x}\|_\rho\,\|\mathbf{w}_b'-\mathbf{w}''\| + \sqrt{2}\left\|(\mathbf{\Phi x})^T\mathbf{w}''\right\|_n \\
&\quad +\alpha\sqrt{(8\tau/n)\log(4\,|\mathcal{W}_\epsilon|/\xi)} && (8.68)\\
&\leq \alpha\,\|\mathbf{x}\|_\rho\,\epsilon + \sqrt{2}\left\|(\mathbf{\Phi x})^T(\mathbf{w}''-\mathbf{w}_b')\right\|_n + \sqrt{2}\left\|(\mathbf{\Phi x})^T\mathbf{w}_b'\right\|_n \\
&\quad +\alpha\sqrt{(8\tau/n)\log(4\,|\mathcal{W}_\epsilon|/\xi)} && (8.69)\\
&\leq \alpha\,\|\mathbf{x}\|_\rho\,\epsilon + \sqrt{2}\alpha\,\|\mathbf{x}\|_n\,\epsilon + \sqrt{2}\left\|(\mathbf{\Phi x})^T\mathbf{w}_b'\right\|_n \\
&\quad +\alpha\sqrt{(8\tau/n)\log(4\,|\mathcal{W}_\epsilon|/\xi)} && (8.70)\\
&\leq \sqrt{2}\,\left\|(\mathbf{\Phi x})^T\mathbf{w}_b\right\|_n/\,\|\mathbf{w}_b\| \\
&\quad +\alpha\epsilon(\|\mathbf{x}\|_\rho + \sqrt{2}\,\|\mathbf{x}\|_n) + \alpha\sqrt{(8\tau/n)\log(4\,|\mathcal{W}_\epsilon|/\xi)}. && (8.71)
\end{aligned}$$

Line 8.68 uses Equation 8.65, and we use Equation 8.63 in lines 8.69 and 8.70.

Using the definition of $\mathbf{w}_b$ , we have that:

$$\begin{aligned}
\|\mathbf{w}_b\| &\leq \left\|(\mathbf{X\Phi}^T)^\dagger\right\|\,\|\mathbf{b}\| && (8.72)\\
&\leq \left\|(\mathbf{X\Phi}^T)^\dagger\right\|\,\epsilon_{\text{prj}}\,\|\mathbf{w}\|\,\|\mathbf{X}\|_F && (8.73)\\
&\leq \epsilon_{\text{prj}}\,\|\mathbf{w}\|\,\sqrt{1/\nu}. && (8.74)
\end{aligned}$$

Thus, using Equation 8.62 and substituting $\|\mathbf{w}_b\|$ in Equation 8.71, and letting $\epsilon = 1/\sqrt{d}$, we get:

$$\begin{aligned}\left\|(\mathbf{\Phi x})^T \mathbf{w}_b\right\|_\rho \leq\; & \sqrt{8}\epsilon_{\text{prj}} \|\mathbf{w}\| \|\mathbf{x}\|_n \\ & + \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \left(\|\mathbf{x}\|_\rho + \sqrt{2}\|\mathbf{x}\|_n\right)\sqrt{\frac{1}{d\nu}} \\ & + \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \sqrt{\frac{8\tau}{n\nu}\log\frac{4|\mathcal{W}_\epsilon|}{\xi}}. \end{aligned} \tag{8.75}$$

Using the properties of eigenvalues we have that $d\nu \leq trace(\mathbf{\Phi X}^T\mathbf{X\Phi}^T/\|\mathbf{X}\|_F)$ and thus $d\nu \leq \alpha^2$. Using $1 \leq \sqrt{\alpha^2/d\nu}$, $|\mathcal{W}_\epsilon| \leq (3\sqrt{d})^d$ and $d \geq 12$ we get:

$$\begin{aligned}\left\|(\mathbf{\Phi x})^T \mathbf{w}_b\right\|_\rho \leq\; & \sqrt{8}\epsilon_{\text{prj}} \|\mathbf{w}\| \|\mathbf{x}\|_n \sqrt{\frac{\alpha^2}{d\nu}} \\ & + \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \left(\|\mathbf{x}\|_\rho + \sqrt{2}\|\mathbf{x}\|_n\right)\sqrt{\frac{1}{d\nu}} \\ & + \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \sqrt{\frac{8\tau}{n\nu}\log\frac{4(3\sqrt{d})^d}{\xi}} \qquad (8.76) \\ \leq\; & \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \left(\|\mathbf{x}\|_\rho + (\sqrt{2}+\sqrt{8})\|\mathbf{x}\|_n\right)\sqrt{\frac{1}{d\nu}} \\ & + \alpha\epsilon_{\text{prj}} \|\mathbf{w}\| \sqrt{\frac{4d\tau}{n\nu}\log\frac{9d}{\xi}}. \qquad (8.77)\end{aligned}$$

□

**Bounding the Regression to Noise Terms**

To bound the regression to the noise, we need the following lemma on martingales:

**Lemma 8.7.** *Under the conditions of Theorem 8.3, let* $\mathbf{y}$ *be a vector of size* $n \times 1$*, in which row* $t$ *is a function of* $\mathbf{x}_t$*. Then with probability* $1 - \xi$ *we have:*

$$|\mathbf{y}^T \eta| \leq \eta_{\max} \|\mathbf{y}\| \sqrt{2 \log \frac{2}{\xi}}. \tag{8.78}$$

*Proof.* This is a simple application of generalized Azuma-Hoeffding concentration inequality on martingales. □

**Lemma 8.8 (Bounding regression to the noise).** *Under the conditions of Theorem 8.3:*

$$\Pr_{\{(\mathbf{x}_i,\eta_i)\}_{i=1}^n \sim C_n} \left\{ \left\|(\mathbf{x}^T \mathbf{\Phi})(\mathbf{X}\mathbf{\Phi})^\dagger \eta\right\|_\rho \leq \alpha \eta_{\max} \frac{\|\mathbf{x}\|_\rho}{\|\mathbf{x}\|_n} \sqrt{\frac{2\kappa d}{n\nu} \log \frac{8d}{\xi}} \right\} \geq 1 - \xi/4. \tag{8.79}$$

*Proof.* For all $i \in \{1, \ldots, d\}$, define the vector $\mathbf{e}_i^{d\times 1}$ to have 1 on the $i$th row and be 0 elsewhere. Using union bound on Lemma 8.7, we have with probability no less than $1 - \xi/4$:

$$\forall i : \left|(\mathbf{e}_i^T \mathbf{\Phi} \mathbf{X}^T)\eta\right| \leq \eta_{\max} \left\|\mathbf{X}\mathbf{\Phi}^T\right\| \sqrt{2 \log \frac{8d}{\xi}}. \tag{8.80}$$

For a fixed $x$, define $\mathbf{z}^T = (\mathbf{\Phi x})^T((\mathbf{X\Phi}^T)^T(\mathbf{X\Phi}^T))^{-1}$. We have:

$$\begin{aligned}
\left|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right| &= \left|\mathbf{z}^T(\mathbf{X\Phi}^T)^T\eta\right| && (8.81)\\
&= \left|\sum_{i=1}^{d}(\mathbf{z}^T\mathbf{e}_i)(\mathbf{e}_i^T\mathbf{\Phi X}^T)\eta\right| && (8.82)\\
&\leq \sum_{i=1}^{d}\left|\mathbf{z}^T\mathbf{e}_i\right|\left|(\mathbf{e}_i^T\mathbf{\Phi X}^T)\eta\right| && (8.83)\\
&\leq \eta_{\max}\left\|\mathbf{X\Phi}^T\right\|\sqrt{2\log\frac{8d}{\xi}}\left\|\mathbf{z}\right\|_1 && (8.84)\\
&\leq \eta_{\max}\left\|\mathbf{X\Phi}^T\right\|\sqrt{2d\log\frac{8d}{\xi}}\left\|\mathbf{z}\right\|, && (8.85)
\end{aligned}$$

where we used the inequality of Equation 8.80 in Line 8.84. We thus get:

$$\begin{aligned}
\left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right\|_\rho &\leq \eta_{\max}\left\|\mathbf{X\Phi}^T\right\|\left\|(\mathbf{\Phi x})^T(\mathbf{\Phi X}^T\mathbf{X\Phi}^T)^{-1}\right\|_\rho\sqrt{2d\log\frac{8d}{\xi}} && (8.86)\\
&\leq \alpha\,\eta_{\max}\left\|\mathbf{x}\right\|_\rho\left\|\mathbf{X\Phi}^T\right\|\left\|(\mathbf{\Phi X}^T\mathbf{X\Phi}^T)^{-1}\right\|\sqrt{2d\log\frac{8d}{\xi}} && (8.87)\\
&\leq \alpha\eta_{\max}\frac{\left\|\mathbf{x}\right\|_\rho}{\left\|\mathbf{x}\right\|_n}\sqrt{\frac{2\kappa d}{n\nu}\log\frac{8d}{\xi}}. && (8.88)
\end{aligned}$$

In Line 8.88, we used the relations between operator norms and eigenvalues of the normalized gram matrix. □

## Combining the Bounds

*Proof of Theorem 8.3.* To prove the theorem, we start by an application of Lemma 8.4 to $f(\mathbf{x}) = \|\mathbf{x}\|^2$. Assume $\forall \mathbf{x} \in \mathcal{X}_k^D : \|\mathbf{x}\|^2 \in [a,b]$. We get:

$$\Pr_{\{(\mathbf{x}_i,\eta_i)\}_{i=1}^n\sim C_n}\left\{\left\|\mathbf{x}\right\|_n^2 \leq 2\left\|\mathbf{x}\right\|_\rho^2 + \frac{4b\tau}{n}\log\frac{8}{\xi}\right\} \geq 1-\xi/8. \qquad (8.89)$$

Using the condition on $n$ from Theorem 8.3, we have that $\frac{4b\tau}{n}\log\frac{1}{\xi} \leq a \leq \|\mathbf{x}\|_\rho^2$. We thus get:

$$\underset{\{(\mathbf{x}_i,\eta_i)\}_{i=1}^n \sim C_n}{\mathbb{P}\mathrm{r}} \left\{ \|\mathbf{x}\|_n^2 \leq 3\|\mathbf{x}\|_\rho^2 \right\} \geq 1 - \xi/8. \tag{8.90}$$

With a similar argument using Lemma 8.5, we have:

$$\underset{\{(\mathbf{x}_i,\eta_i)\}_{i=1}^n \sim C_n}{\mathbb{P}\mathrm{r}} \left\{ \|\mathbf{x}\|_\rho^2 \leq 4\|\mathbf{x}\|_n^2 \right\} \geq 1 - \xi/8. \tag{8.91}$$

We also apply Theorem 8.1 with failure probability $\xi/4$. With the definition of $\epsilon_{\mathrm{prj}}$ in Theorem 8.3, we have:

$$\underset{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}}{\mathbb{P}\mathrm{r}} \left\{ \forall \mathbf{x} \in \mathcal{X}_k^D : \left|(\mathbf{\Phi x})^T(\mathbf{\Phi w}) - \mathbf{x}^T\mathbf{w}\right| \leq \epsilon_{\mathrm{prj}}\|\mathbf{x}\|\|\mathbf{w}\| \right\} \geq 1 - \xi/4 \tag{8.92}$$

We now union bound over Equations 8.90, 8.91, 8.92, regression to bias Lemma 8.6 and regression to noise Lemma 8.8. Substituting the bounds into the error composition of Equation 8.58 we have with probability $1-\xi$:

$$\begin{aligned}
\left\|\hat{f}_{\mathbf{\Phi}}(\mathbf{x}) - f(\mathbf{x})\right\|_\rho &\leq \epsilon_{\mathrm{prj}}\|\mathbf{w}\|\|\mathbf{x}\|_\rho + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\mathbf{b}\right\|_\rho + \left\|(\mathbf{\Phi x})^T(\mathbf{X\Phi}^T)^\dagger\eta\right\|_\rho \\
&\leq \epsilon_{\mathrm{prj}}\|\mathbf{w}\|\|\mathbf{x}\|_\rho \\
&+ \alpha\epsilon_{\mathrm{prj}}\|\mathbf{w}\|\left(\|\mathbf{x}\|_\rho + 4.3\sqrt{3}\|\mathbf{x}\|_\rho\right)\sqrt{\frac{1}{d\nu}} + \alpha\epsilon_{\mathrm{prj}}\|\mathbf{w}\|\sqrt{\frac{4d\tau}{n\nu}\log\frac{9d}{\xi}} \\
&+ \alpha\eta_{\max}\frac{2\|\mathbf{x}\|_n}{\|\mathbf{x}\|_n}\sqrt{\frac{2\kappa d}{n\nu}\log\frac{8d}{\xi}}.
\end{aligned} \tag{8.93}$$

Since $\alpha \geq 1$, we can simplify the above as follows:

$$\left\|\hat{f}_{\mathbf{\Phi}}(\mathbf{x}) - f(\mathbf{x})\right\|_\rho \leq 10\,\alpha\epsilon_{\text{prj}}\,\|\mathbf{w}\|\,\|\mathbf{x}\|_\rho\sqrt{\frac{1}{d\nu}} + 2\,\alpha\epsilon_{\text{prj}}\,\|\mathbf{w}\|\sqrt{\frac{d\tau}{n\nu}\log\frac{9d}{\xi}} + \sqrt{8}\,\alpha\eta_{\max}\sqrt{\frac{\kappa d}{n\nu}\log\frac{8d}{\xi}}. \qquad (8.94)$$

□

## 8.5 Empirical Evaluations

In this section, we aim to elucidate the conditions under which random projections are useful in sparse regression problems. This section only evaluates the effectiveness of compressed regression in the i.i.d. data collection setting with Gaussian noise. In the next chapter, we develop methods and empirical evaluations for compressed regression in non-i.i.d. settings.

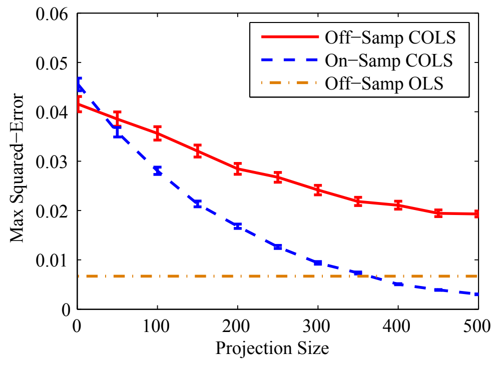
We start with synthetic regression datasets, in which the input vectors are sampled from a 1000-dimensional feature space ($D = 1000$), and at most 5% of the features are non-zero in any particular instance ($k = 50$). The target function is linear in the input features. The weight vector $\mathbf{w}$ of the target function is generated randomly from a normal distribution with diagonal covariance matrix. The details of the number of samples and the choice of the weight vector are varied in different test-cases.

### 8.5.1 Bias–Variance Trade-off

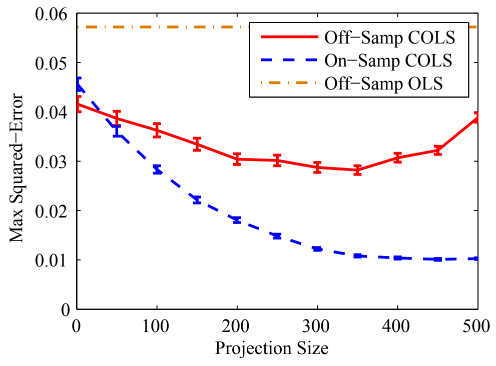
In order to be faithful to the worst-case theoretical analysis of Theorem 8.2, we study the bias–variance trade-off using the maximum squared-error on the training set (on-sample error) and testing set (off-sample error), as a function of the projection size, in different noise settings. To do so, we generate weight vectors $\mathbf{w}$ for the target function, in which each component is drawn independently from a normal distribution. Ten of the weights are generated using a larger variance, such that 80% of the norm of $\mathbf{w}$ is on those elements. This relatively sparse choice of $\mathbf{w}$ helps to illustrate the trade-off, as we discuss later. The features of the training and testing instances are also independently normally distributed on $k$ randomly chosen elements and are 0 elsewhere. Both the input values and the weight vector are of expected norm 1.

We consider two settings. In the first setting, the training set is smaller than the number of original dimensions ($n < D$), while in the second, there are more training examples than the number of dimensions ($n > D$). In each setting, we generate different levels of noise, quantified by the ratio between the standard deviation of the noise and that of the target function on the training set (similar to the concept of signal-to-noise ratio from signal processing). We plot the maximum squared error as a function of the dataset size.

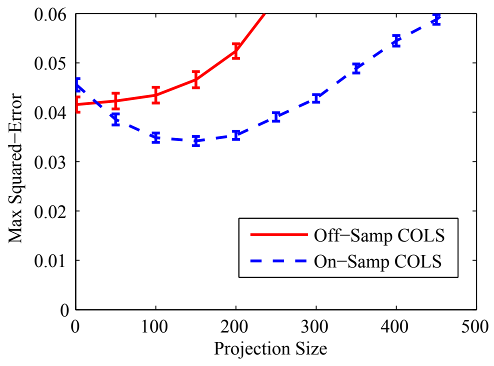
Figure 8–1 summarizes the trade-off, averaged over 100 runs, when the training and testing sets are of size $n = 800$. Even though OLS is not well defined in the original space when $n < D$, we can still use the pseudo-inverse of the input matrix to find the linear fit, among many possible solutions, that has the smallest norm for the weight vector (for convenience, we call this the OLS solution here). Because we are using a relatively sparse $\mathbf{w}$, we might get a fit that is better than the baseline constant predictor (i.e. projection of size 0). Figure 8–2 shows a



(a) Noise ratio $= 0.3$.



(b) Noise ratio $= 1.1$.

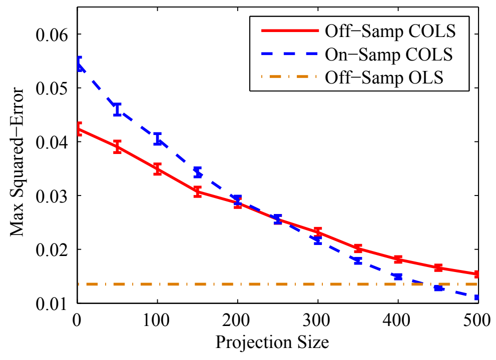


(c) Noise ratio $= 3.0$. OLS error $\simeq 0.29$.

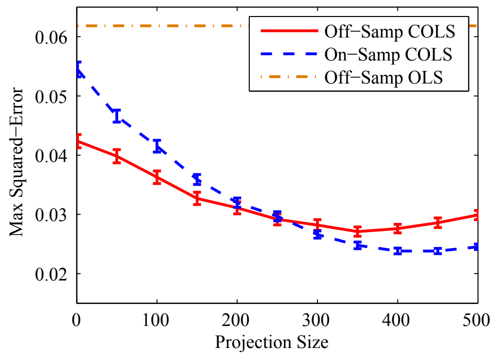
Figure 8–1: Error vs. projection size when $n = 800$ is less than $D = 1000$.

similar analysis for training sets slightly larger than the number of dimensions ($n = 2000$).

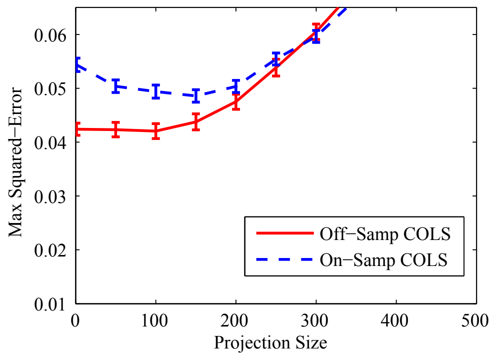
For small noise levels (Figures 8–1a and 8–2a), OLS in the original space works well and the plot shows a constant decrease in the error of the COLS predictor as we use larger random projections. In these cases, one should use as many features as possible (depending on the computational cost) for regression.



(a) Noise ratio = 1.1.



(b) Noise ratio = 2.4.



(c) Noise ratio = 5.0. OLS error $\simeq 0.40$.

Figure 8–2: Error vs. projection size when $n = 2000$ is more than $D = 1000$.

As we increase the noise level (Figures 8–1b and 8–2b), the bias–variance trade-off becomes apparent. Here we see that it is better to use random projections of intermediate size to optimize the trade-off. Finding the optimal projection size might be a challenge in practice. Error bounds such as the ones presented in this work give clues on the existence and values of such optimal sizes; or, one can use cross-validation to find the optimal projection size, as illustrated in these experiments.

Higher noise levels (Figures 8–1c and 8–2c) make the prediction problem impossible to solve, in which case the best regression is the constant baseline predictor. Note that the OLS solution in this case is significantly worse, so we do not plot it in order to keep the panels on the same scale.

We can conclude from this experiment that when the problem is not too easy (almost noiseless), or too difficult (with large noise), random projections of sizes smaller than the number of dimensions provide the optimal worst-case error rates. Even in the noiseless setting, if $D$ is very large, we might still want to use random projection as a form of feature selection/extraction, instead of $\ell^1$ regularization methods, which are significantly more computationally expensive (Efron et al., 2004).

### 8.5.2 Sparsity of Linear Coefficients

Next, we analyze the effect of the sparsity of the linear weight vector (rather than the sparsity of the inputs) on the worst-case prediction error. In this experiment, we fix the noise ratio and observe the bias–variance trade-off as we change the level of sparsity in the linear coefficients. We use a similar setting as described in the previous experiment. We generate the weight vector $\mathbf{w}$ by sampling from a Gaussian distribution with mean zero and unit diagonal covariance matrix, and then scale 10 of its elements such that they account for 50%, 70% and 90% of the norm of $\mathbf{w}$. The trade-off on the worst-case testing error and the error of the OLS predictor on the original and compressed space are shown in Figure 8–3.

The results indicate that the trade-off is present even when we use less concentrated weight vectors. However, random projections seem to have higher

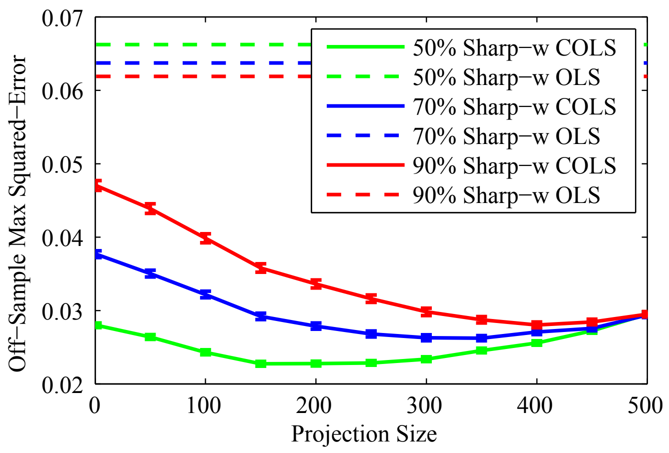


Figure 8–3: Error vs. projection size for different concentration of **w**, when $n = 2000$, $D = 1000$ and the noise ratio $= 2.4$.

effectiveness in the reduction of prediction error for highly concentrated weight vectors. This could be due to the fact that in the current setup, the inner-product between **w** and points in a sparse space is better preserved when **w** is sparse itself. Nevertheless, for a problem having large enough number of samples and features, the trade-off is expected to be apparent even for non-concentrated weight vectors, as the theory suggests.

### 8.5.3 Music Similarity Prediction

To assess the effectiveness of random projections as means of feature extraction for regression in high-dimensional spaces, we experiment with a music dataset for a similarity prediction task. The task is to predict the similarity between tracks of classical music using audio analysis. The internet radio service Last.fm (2014), provides a similarity measure between different tracks in its music database, which is calculated using the listening patterns from the users and the co-presence of tracks in playlists. For newly added songs, however, this value is unknown, and thus a predictor might be very useful for music recommendation or automatic playlist generation.

The regression problem is constructed as follows. We analyze 30 second audio samples of music tracks and apply frequency analysis on each 200ms segment. Scanning the entire dataset, we extract the top 50 common *chords*[1] . Using those chords, we then extract the top 1000 common *chord progressions*[2] of up to 8 segments. Having this set, we can check whether or not each of these progressions is present in a music track.

Our training set consists of pairs of music tracks along with a similarity score between 0 and 1 which is provided by Last.fm. For each pair of tracks in our training set, we construct a set of binary features corresponding to the simultaneous presence of a certain progression in track one, and another progression in track two. Therefore, there are $1000 \times 1000 =$ one million features for each training example. Using random projections, we reduce the dimension and apply OLS in the compressed space. We compare this with a naive baseline which randomly chooses features from the original one million dimensional space and applies OLS on the chosen features.

Figure 8–4 compares the mean squared-error of the discussed methods for the training and testing sets, using ten-fold cross-validation on 2000 training samples.

[1] Combinations of notes present in each segment.

[2] Patterns of chords happening in a sequence.

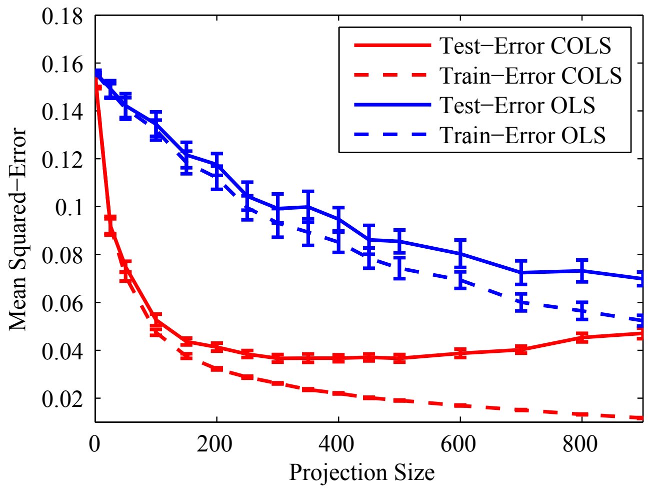


Figure 8–4: Error vs. number of features used for random projection and naive feature selection on the music dataset.

We use the average here instead of max, as we do not have access to the true target value and the worst case error contains a noise term.

The compressed regression method significantly outperforms the naive feature selection method, even with relatively small projection sizes. We expect the error of the naive OLS predictor to get minimized with a much larger number of selected features; however, solving the OLS for these larger problems would become computationally intractable.

This experiment shows that random projections can be useful in real world prediction tasks with very large feature spaces, while incurring relatively small computational cost. However, a more thorough analysis is required to compare this approach with other commonly used feature extraction methods (e.g. LASSO-type regression (Efron et al., 2004)), both in terms of error rates and computation feasibility. In the next chapter, we provide a comparison with one of the few computationally feasible alternatives, namely LSQR (Paige and Saunders, 1982), when working with time-dependent datasets in the RL domain.

## 8.6 Related Work

The analysis of regression with random compression has been done in different settings with different assumptions on the feature space and sampling distribution. Maillard and Munos (2009) provide an analysis without the sparsity assumption on the feature space (later corrected in Maillard and Munos (2012)), and show that the bias due to the projection can be controlled by changing the projection size. They conclude that an optimal projection size should be of order $O(\sqrt{n})$. Their bound, however, has a curious $\log(n)$ term in the bias that essentially grows with the number of samples.

Recently developed bounds in the fixed design settings (where the dataset is fixed and the error is calculated on the training set) avoid the $\log(n)$ term at the expense of having the bound scale with $1/\xi$ rather than $\log(1/\xi)$ (Kabán, 2013, 2014). Further development in this direction using the newly developed method of ridge regression analysis (Hsu et al., 2012) might be able to provide even tighter bounds in the fixed design setting.

The extension to the random design setting, similar to the analysis of Theorem 8.3, requires bounding the approximation error of linear regression in the compressed space. We use RIP type conditions with sparsity constraints to bound the approximation error of COLS on the sampling distribution. It is not clear if a random design analysis can be done without constraining the input space or the sampling distribution.

We have further extended the analysis of Theorem 8.3 to develop an error bound when using Predictive State Representation (PSR) in compressed spaces (Hamilton et al., 2013). PSRs are prediction models that can be formalized in terms of regression operators, and thus the COLS analysis presented here can be adapted to be used when working with PSRs with sparse transition models. We have also analysed PSRs when their linear operators have near-sparse spectral structures and shown how one can obtain even tighter bounds in such cases (Hamilton et al., 2014).

## 8.7 Discussion and Future Work

Theorem 8.2 gives an analysis on the worst-case prediction error of the OLS estimator built on the space induced by random projections from sparse spaces. We prove that random projections preserve the inner-product between sparse features and any fixed vector. This shows that near-linearity is preserved after the projection into smaller spaces, even for points that are never observed in the training set.

Leveraging the sparsity assumption in the input space, unlike previous analysis of random projection for regression tasks (Maillard and Munos, 2009, 2012), we provide worst-case bounds that hold regardless of the distribution of the training or testing sets. Most noticeably, we do not require i.i.d. sampling or similarity between the training and testing sets. This is particularly useful when the regressor is expected to perform well on data that is not sampled from the same distribution as the one on which it is trained (e.g. time series analysis and domain adaptation settings). The bound of Theorem 8.2 is, however, tighter when the data is well distributed and covers most of the problem space.

Theorem 8.3 provides a much tighter bound when the input is sampled from the same distribution that generated the data. Unlike the analysis of Maillard and Munos (2009, 2012), we do not require an i.i.d. sampling distribution. Instead, we assume the data is generated by a fast mixing Markov chain. This helps us use the bound when working with dependent time series data. We use Theorem 8.3 in the next chapter to analyse the performance of an RL value function estimator.

Under mild assumptions on the distribution of the training data, the bounds of both Theorem 8.2 and Theorem 8.3 reduce to a bias–variance trade-off on the prediction error, as a function of the projection size. Increasing the number of projected features reduces the approximation error of the OLS estimator on the induced space, but at the same time introduces some estimation error as the number of learning parameters increases.

The optimal choice for the error trade-off depends on the structure of the target function (sparsity of $\mathbf{w}$), and the noise level. Our analysis suggests an optimal projection size of $\tilde{O}(\sqrt{nk})$ for worst-case error minimization, resembling the suggested size of $\tilde{O}(\sqrt{n})$ for on-measure error minimization by Maillard and Munos (2009) (though their analysis is on-sample and with much larger constants). Depending on the noise level, the minimizer of expected error might be out of the feasible range of $1 \leq d \leq D$. This is manifested in our empirical evaluation of the method. With small noise, it is sometimes better to apply OLS on the original space (or if not possible, use projections as large as computationally feasible). For large noise levels and relatively small sample sizes, we are often in a situation where the best predictor is a constant one. Nevertheless, there is an important range of problems for which the COLS prediction vastly outperforms the prediction in the original space.

Our theoretical and empirical analysis of compressed OLS estimators provides some level of understanding of the usefulness of random projections in regression problems with sparse input signals. This sparsity is commonly observed in many fields of machine learning. There are many areas of future work in this domain. While $\ell^1$ regularized regression is not applicable in domains with large feature spaces due to its computational complexity, other types of linear estimators (e.g. $\ell^2$ regularized regression) should be analysed in the settings we examine. Some empirical comparison to ridge regression with LSQR solver (Paige and Saunders, 1982) is provided in the next chapter.

Since linear predictors are the building blocks of many learning algorithms, we expect random projections to be effective means of feature extraction when working with high dimensional data in many other fields of machine learning. These include the use of random projections in audio, video and time-series analysis, or with LSTD-type algorithms for high-dimensional reinforcement learning (Lazaric et al., 2010). In the next chapter, we study one such use of compressed OLS in the context of value function approximation in reinforcement learning.

# CHAPTER 9
# Compressed Reinforcement Learning

Policy evaluation, i.e. computing the expected return of a given policy, is at the core of many RL algorithms. In large problems, it is necessary to use function approximation in order to perform this task; a standard choice is to hand-craft parametric function approximators, such as a tile coding, radial basis functions or neural networks. The accuracy of parametrized policy evaluation depends crucially on the quality of the features used in the function approximator, and thus a lot of time and effort is spent on this step. The desire to make this process more automatic has led to significant recent work on feature generation and feature selection in RL (e.g. Di Castro and Mannor (2010); Kolter and Ng (2009a); Keller et al. (2006); Manoonpong et al. (2010); Geramifard et al. (2011))

An approach that offers good theoretical guarantees is to generate features in the direction of the Bellman error of the current value estimates (Bellman Error Based features, or BEBF). Successively adding exact BEBFs has been shown to reduce the error of a linear value function estimator at a rate similar to value iteration, which is the best one could hope to achieve (Parr et al., 2007). Unlike fitted value iteration (Boyan and Moore, 1995), which works with a fixed feature set, iterative BEBF generation gradually increases the complexity of the hypothesis space by adding new features and thus does not diverge, as long as the error in the generation does not cancel out the contraction effect of the Bellman operator (Parr et al., 2007). Several successful methods have been proposed for generating features related to the Bellman error (Geramifard et al., 2011; Di Castro and Mannor, 2010; Manoonpong et al., 2010; Parr et al., 2007; Keller et al., 2006). In practice however, these methods can be computationally expensive when applied in high dimensional input spaces.

With the emergence of more high-dimensional RL problems, it has become necessary to design and adapt BEBF-based methods to be more scalable and computationally efficient. In this chapter, we present an algorithm that uses the idea of applying random projections specifically in very large and sparse feature spaces. The idea is to iteratively project the original features into exponentially lower-dimensional spaces. We then apply linear regression in the smaller spaces, using temporal difference errors as targets, in order to approximate BEBFs.

We use the analysis of compressed linear regression from the previous chapter and apply it to the BEBF linear approximation scenario to provide error bounds for the resulting value function estimator. We also validate our proposed algorithm by a set of empirical evaluations in a high dimensional RL problem.

## 9.1 Bellman-Error-Based Features Generation

Recall that for a given estimate of the value function $\hat{V}^\pi$ the Bellman error is defined as:

$$e_{\hat{V}^\pi}(s) \overset{\text{def}}{=} (\mathbb{T}^\pi \hat{V}^\pi)(s) - \hat{V}^\pi(s). \tag{9.1}$$

Calculating the exact Bellman error requires full knowledge of the Bellman operator $\mathbb{T}^\pi$. As discussed before, one way to estimate the value of $e_{\hat{V}^\pi}(s)$ is through the observed TD-errors. Recall that the TD-error of an estimated value function $\hat{V}_t$ at after observing the $t$-th transition and reward is defined as:

$$\delta_t = R_t + \gamma \hat{V}_t(S_{t+1}) - \hat{V}_t(S_t). \tag{9.2}$$

It is easy to show that the expectation of the temporal difference at $\mathbf{x}_t$ equals the Bellman error at that point (Sutton and Barto, 1998). TD-errors are thus proxies to estimating the Bellman error.

Several methods have been developed to estimate the Bellman error and use it as a basis for value function approximation. Using temporal differences, Menache et al. (2005) introduced two algorithms to construct basis functions for linear function approximation. Keller et al. (2006) applied neighbourhood component analysis as a dimensionality reduction technique to construct a low dimensional state space based on TD-errors. In their work, they iteratively add features that would help predict the Bellman error. Parr et al. (2007) later showed that any BEBF extraction method with small angular error will provably tighten the approximation error of the value function estimate.

Online BEBF extraction methods have also been studied in the RL literature. The *incremental Feature Dependency Discovery* (iFDD) is a fast online algorithm to extract non-linear binary features for linear function approximation (Geramifard et al., 2011). In their work, one keeps a list of candidate features (non-linear combination of two active features); the features most correlated with the TD-error are then added to the approximator.

We note that these algorithms, although theoretically interesting, are difficult to apply to very large state spaces or need specific domain knowledge to generate good features. The problem lies in the large estimation error when predicting BEBFs in high-dimensional state spaces. Our proposed solution leverages the use of simple random projections to alleviate this problem.

## 9.2 Compressed Bellman-Error-Based Features Generation

In this section, we propose a new method to generate BEBFs using linear regression in a small space induced by random projections. We first project the state features into a much smaller space and then regress a hyperplane to the TD-errors. For simplicity, we assume that regardless of the current estimate of the value function, the Bellman error is always linearly representable in the original feature space. This seems like a strong assumption, but is true, for example, in many discretized state-spaces, and is also likely to hold in very high dimensional feature spaces[1] .

Linear function approximators can be used to estimate the value of a given state. Let $V_m$ be an estimated value function described in a linear space defined by a feature set $\{\psi_i\}_{i=1}^{m}$. Parr et al. (2007) show that if we add a new BEBF $\psi_{m+1} = e_{V_m}$ to the feature set, (with mild assumptions) the approximation error on the new linear space spanned by $\{\psi_i\}_{i=1}^{m+1}$ shrinks by a factor of $\gamma$. They also

[1] For the more general case, the analysis of BEBF generation can be done with respect to the *projected* Bellman error (Parr et al., 2007). We assume linearity of the Bellman error to simplify the derivations.

show that if we can estimate the Bellman error within a constant angular error, $\cos^{-1}(\gamma)$, the error will still shrink.

Estimating the Bellman error by regressing to temporal differences in high-dimensional sparse spaces can result in large prediction error. This is due to the large estimation error of regression in high dimensional spaces. However, as discussed in Theorem 8.1, random projections were shown to exponentially reduce the dimension of a sparse feature space, only at the cost of a controlled constant bias. A variance analysis along with proper mixing conditions can also bound the estimation error due to the variance in MDP returns. The computational cost of the estimation is also much smaller when the regression is applied in the compressed space.

### 9.2.1 General CBEBF Algorithm

Theorem 8.3 provided an analysis for linear regression performed in a space induced by random projections. We see that with sparse input features, we expect random projections of sizes much smaller than the nominal dimensionality of the problem to be sufficient for the sake of compressed linear regression.

In light of these results, we propose the *Compressed Bellman Error Based Feature Generation* algorithm (CBEBF). The algorithm iteratively constructs new features using compressed linear regression to the TD-errors, and uses these features with a policy evaluation algorithm to update the estimate of the value function. We assume our input space is featurized using a function $\varphi$. Therefore $\varphi(S)$ is the base feature vector of the state $S$. Instead of using $\varphi(S)$ as the basis for value function approximation, we construct the value function using generated BEBFs in a compressed space.

**Algorithm 1** CBEBF algorithm

```
Input: Sample of transitions $\mathcal{D}_n = \{(S_t, A_t, R_t, S'_t)\}_{t=1}^n$
Input: Base feature function $\varphi : \mathcal{S} \to \mathbb{R}^D$
Input: Number of BEBFs: $m$
Input: Projection size schedule: $d_1, d_2, \ldots, d_m$
Output: $V$: estimate of the value function

procedure CBEBF($\mathcal{D}_n, \varphi, m, d_{1:m}$)
    Initialize $V_1(S)$ to be 0 for all $S \in \mathcal{S}$.
    Initialize the set of CBEBF features $\Psi \leftarrow \{\}$.
    for $i \leftarrow 1$ to $m$ do
        Calculate TD-errors: $\delta_t = r_t + \gamma V_i(S'_t) - V_i(S_t)$.
        Generate random projection $\mathbf{\Phi}^{d_i \times D}$ satisfying Equation 7.9.
        Apply compressed regression (COLS) using projection $\mathbf{\Phi}$:
            Let $\hat{\mathbf{w}}_{\mathbf{\Phi}}^{d_i \times 1}$ be the OLS parameter in the compressed space,
            using $\mathbf{\Phi}\varphi(S_t)$ as inputs and $\delta_t$ as outputs.
        Add $\hat{e}_m(S) \overset{\text{def}}{=} \hat{\mathbf{w}}_{\mathbf{\Phi}}^T \mathbf{\Phi}\varphi(S)$ to $\Psi$.
        Apply policy evaluation on feature sets $\Psi$ to obtain $V_{i+1}$.
    end for
    return $V_m$
end procedure
```

In theory, the optimal number of BEBFs and the schedule of projection sizes can be determined in advance. But we show in the next section that, under certain conditions, logarithmic size projections should be enough to guarantee the reduction of error in value function prediction at each step. This makes the algorithm very attractive when it comes to computational and memory complexity, as the regression at each step is only on a small projected feature space. As we discuss in our empirical analysis, the algorithm is fast and robust with respect to the selection of parameters.

One can view the above algorithm as a model selection procedure that gradually increases the complexity of the hypothesis space by adding more BEBFs to the feature set. This means that the procedure has to be stopped at some point to avoid over-fitting. This is relatively easy to do, as one can use a validation set and compare the estimated values against the empirical returns. The generation of BEBFs should stop when the validation error starts to rise.

### 9.2.2 Simplified CBEBF as Regularized Value Iteration

Note that in CBEBF, we can use any type of value function approximation to estimate the value function in each iteration. To simplify the bias–variance analysis and avoid multiple levels of regression, we present here a simplified version of the CBEBF algorithm (SCBEBF). In the simplified version, instead of storing generated features in each iteration, new features are added to the value function approximator with constant weight 1. Therefore, the value estimate is simply the sum of all generated BEBFs (i.e. $V_{i+1}(S) = \sum_{f \in \Psi} f(S)$).

As compared to the general CBEBF, the simplified version trivially has lower computational complexity per iteration, as it avoids an extra level of regression based on the features. It can also avoid storing all the generated features by simply keeping the sum of all previously generated coefficients $\mathbf{\Phi}^T \hat{\mathbf{w}}_{\mathbf{\Phi}}$'s (since sum of linear functions is a linear function).

**Algorithm 2** SCBEBF algorithm

```
Input: Sample of transitions $\mathcal{D}_n = \{(S_t, A_t, R_t, S'_t)\}_{t=1}^n$
Input: Base feature function $\varphi : \mathcal{S} \to \mathbb{R}^D$
Input: Number of BEBFs: $m$
Input: Projection size schedule: $d_1, d_2, \ldots, d_m$
Output: $V$: estimate of the value function

procedure SCBEBF($\mathcal{D}_n, \varphi, m, d_{1:m}$)
    Initialize value estimator coefficients $\mathbf{w}^{D\times 1} \leftarrow \mathbf{0}$.
    for $i \leftarrow 1$ to $m$ do
        Calculate TD-errors: $\delta_t = r_t + \gamma \mathbf{w}^T \varphi(S'_t) - \mathbf{w}^T \varphi(S_t)$.
        Generate random projection $\mathbf{\Phi}^{d_i \times D}$ satisfying Equation 7.9.
        Apply compressed regression (COLS) using projection $\mathbf{\Phi}$:
            Let $\hat{\mathbf{w}}_{\mathbf{\Phi}}^{d_i \times 1}$ be the OLS parameter in the compressed space,
            using $\mathbf{\Phi}\varphi(S_t)$ as inputs and $\delta_t$ as outputs.
        Update $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{\Phi}^T \hat{\mathbf{w}}_{\mathbf{\Phi}}$
    end for
    return $V(S) = \mathbf{w}^T \varphi(S)$
end procedure
```

It is important to note that once we use linear value function approximation, the entire BEBF generation process can be viewed as a regularized value iteration algorithm. Each iteration of the algorithm is a regularized Bellman backup which is linear in the features. The coefficients of this linear backup are confined to a lower-dimensional random subspace implicitly induced by the random projection used in each iteration. Regularized value iteration with $\ell^1$ and $\ell^2$ regularization employ a similar idea (Farahmand et al., 2009b,a). However, the regularization in those methods can be characterized as bounding a norm of the regressed weight vector, as opposed to optimizing in an unbounded lower-dimensional subspace.

## 9.3 Finite Sample Analysis of Simplified CBEBF

This section provides a finite sample analysis of the simplified CBEBF algorithm. We first look at the error of the BEBF generated at each iteration compared to the true Bellman error at that iteration. Using assumptions on the Bellman error, we then extend the analysis to contraction of the error after multiple iterations.

In order to provide such analysis, we need to have an assumption on the range of observed TD-errors. This is usually possible by assuming that the current estimate of the value function is bounded, which is easy to enforce by truncating any estimate of the value function between 0 and $V_{\max} = R_{\max}/(1-\gamma)$.

The following theorem, based on the fast-mixing framework of Theorem 8.3, shows how well we can estimate the Bellman error by regression to the TD-errors in a compressed space. It highlights the bias–variance trade-off with respect to the choice of the projection size. As stated before, to simplify the derivations, we assume that the true Bellman error $e(S)$ is linear in the features $\varphi(S)$.

**Theorem 9.1.** *Let $\mathbf{\Phi}^{d\times D} \sim \mu_{\mathbf{\Phi}}$ with $D > d \geq 12$ be a random projection that satisfies the concentration of measure inequality of Equation 7.9. Assume for all $s \in S$: $\varphi(s) \in \mathcal{X}_k^D$ and $\hat{V}(s) \in [0, V_{\max}]$. Let $\hat{e}_{\mathbf{\Phi}}(s) \overset{\text{def}}{=} \hat{\mathbf{w}}_{\mathbf{\Phi}}^T \mathbf{\Phi}\varphi(s)$ be the CBEBF estimator of the Bellman error $e(s) \overset{\text{def}}{=} \mathbf{w}^T\varphi(s)$ of $\hat{V}$. Assume $\mathcal{D}_n = \{(S_t, A_t, R_t, S'_t)\}_{t=1}^n$ is generated from a Markov chain $C_n$, resulting from a sampling policy $\pi$, with forgetting time $\tau$ and stationary distribution $\rho(S)$ on the states. For any $0 < \xi < 1/4$, if $n \geq 4\,\frac{\sup_{S\in\mathcal{S}}\|\varphi(S)\|^2}{\inf_{S\in\mathcal{S}}\|\varphi(S)\|^2} \ln\frac{8}{\xi}$:*

$$\Pr_{\substack{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}}\\ D_n\sim C_n}} \left\{ \|\hat{e}_{\mathbf{\Phi}}(S) - e(S)\|_\rho \leq \epsilon_{bias} + \epsilon_{var} \right\} \geq 1 - \xi_{prj}, \tag{9.3}$$

*defining:*

$$\begin{aligned}
\epsilon_{bias} &= 10\,\alpha\epsilon_{prj}\,\|\mathbf{w}\|\,\|\varphi(S)\|_\rho \sqrt{\frac{1}{d\nu}}, && (9.4)\\
\epsilon_{var} &= 2\,\alpha\epsilon_{prj}\,\|\mathbf{w}\| \sqrt{\frac{d\tau}{n\nu}\log\frac{9d}{\xi}} + 4\sqrt{2}\,\alpha V_{\max}\sqrt{\frac{\kappa d}{n\nu}\log\frac{8d}{\xi}} && (9.5)\\
\epsilon_{prj} &= \sqrt{\frac{48k}{d}\ln\frac{16D+8}{\xi}}, && (9.6)\\
\alpha &= \max\left(1, \max_{s\in\mathcal{S}} \|\mathbf{\Phi}\varphi(s)\| \,/\, \|\varphi(s)\|\right), && (9.7)
\end{aligned}$$

*where $\kappa$ and $\nu$ are the condition number and the smallest positive eigenvalue of $\frac{1}{\|\mathbf{X}\|_F}\mathbf{\Phi}\mathbf{X}^T\mathbf{X}\mathbf{\Phi}^T$, and $\mathbf{X}^{n\times D}$ is the $\varphi$-feature matrix of $\mathcal{D}_n$.*

*Proof.* Define $\eta_t \overset{\text{def}}{=} \delta_t - e(S_t)$. Since $\hat{V}(s) \in [0, V_{\max}]$, we have $|\eta_t| \leq 2V_{\max}$. The noise terms have also conditional mean 0 and form a martingale difference series. Forming a chain $\{(\varphi(S_t), \eta_t)\}_{t=1}^{n}$, we apply Theorem 8.3 to get the bound. □

Similar to the analysis of Theorem 8.3, we can simplify the bound of Theorem 9.1 with similar assumptions on the distribution of the features. The bound reduces to:

$$\tilde{O}\left(\|\mathbf{w}\| \, \|\varphi(S)\|_\rho \sqrt{\frac{k \log D}{d}}\right) + \tilde{O}\left(\frac{d}{\sqrt{n}}\right). \tag{9.8}$$

Similarly, the first term is a part of the excess approximation error and the second is the estimation error. We observe a trade-off with respect to the compressed dimension $d$. With the assumptions discussed before, we can see that projection of size $d = \tilde{O}(k \log D)$ should be enough to guarantee arbitrarily small bias, as long as $\|\mathbf{w}\| \, \|\varphi(S)\|_\rho$ is small. Thus, the bound is tight enough to prove reduction in the error as new BEBFs are added to the feature set.

The dependence on the norm of $\mathbf{w}$ is conjectured to be tight by the compressed sensing literature (Davenport et al., 2010), making this bound asymptotically the best one can hope for. This dependence also points out an interesting link between our method and $\ell^2$-regularized LSTD. We expect ridge regression to be favourable in cases where the norm of the weight vector is small. The upper bound on the error of compressed regression is also smaller when the norm of $\mathbf{w}$ is small. Kabán (2013, 2014) provides some discussion on the connections between ridge regression and compressed regression in the fixed design setting.

As mentioned before, our simplified version of the algorithm does not store the generated BEBFs (such that it could later apply value function approximation over them). It adds up all the features with weight 1 to approximate the value function. Therefore our analysis is different from that of Parr et al. (2007). The following lemma (adaptation of results in Parr et al. (2007)) provides a sufficient condition for the shrinkage of the error in the value function prediction:

**Lemma 9.2.** *Let $V^\pi$ be the value function of a policy $\pi$ imposing stationary measure $\rho$ in states, and let $e_V$ be the Bellman error under policy $\pi$ for an estimate $V$. Given a BEBF $\hat{e}_V$ satisfying:*

$$\left\|\hat{e}_V(S) - e_V(S)\right\|_\rho \leq \epsilon \left\|e_V(S)\right\|_\rho, \tag{9.9}$$

*we have that:*

$$\left\|V^\pi(S) - (V(S) + \hat{e}_V(S))\right\|_\rho \leq (\gamma + \epsilon + \epsilon\gamma) \left\|V^\pi(S) - V(S)\right\|_\rho. \tag{9.10}$$

The proof is included in Section 9.4. Note that the above shows a sufficient condition for a $(\gamma + \epsilon + \epsilon\gamma)$-contraction in the error after each iteration of the SCBEBF algorithm. Theorem 9.1 (simplified in Equation 9.8) does not state the error in terms of $\left\|e_V(S)\right\|_\rho = \left\|\mathbf{w}^T\varphi(S)\right\|_\rho$, as needed by this lemma, but rather does it in terms of $\|\mathbf{w}\| \left\|\varphi(S)\right\|_\rho$. Therefore, if there is a large gap between these terms, we cannot expect to see shrinkage in the error (we can only show that the error can be shrunk to a bounded uncontrolled constant). Ghavamzadeh et al. (2010) and Maillard and Munos (2012, 2009) provide some discussion on the cases were $\left\|\mathbf{w}^T\varphi(S)\right\|_\rho$ and $\|\mathbf{w}\| \left\|\varphi(S)\right\|_\rho$ are expected to be close. These cases include when the features are rescaled orthonormal basis functions and also with specific classes of wavelet functions.

The following lemma (proved in Section 9.4) finishes the analysis of error shrinkage in SCBEBF algorithm, using assumptions on the behaviour of the $\|\mathbf{w}\| \, \|\varphi(S)\|_\rho$ term.

**Lemma 9.3.** *Assume the conditions of Theorem 9.1. Further assume for some constants $c_1, c_2, c_3 \geq 1$:*

$$\|\mathbf{w}\| \leq c_1 \left\|\mathbf{w}^T\varphi(S)\right\|_\rho \quad \text{and} \quad \|\varphi(S)\|_\rho \leq c_2 \left\|\mathbf{w}^T\varphi(S)\right\|_\rho \quad \text{and} \quad 1/\nu \leq c_3 d,$$

*There exist universal constants $c_4$ and $c_5$, such that for any $\gamma_0 \in (\gamma, 1)$ and $\xi \in (0, 1/4)$, if:*

$$d \geq \alpha^2 c_1^2 c_2^2 c_3 c_4 \left(\frac{1+\gamma}{\gamma_0-\gamma}\right)^2 k \log\frac{D}{\xi} \quad \text{and} \quad n \geq (\tau + \alpha^2 c_2^2 c_3 \delta_{\max}^2 \kappa) c_5 \left(\frac{1+\gamma}{\gamma_0-\gamma}\right)^2 d^2 \log\frac{d}{\xi},$$

*then with the addition of the estimated compressed BEBF $\hat{e}_{\mathbf{\Phi}}$, we have:*

$$\Pr_{\substack{\mathbf{\Phi}\sim\mu_{\mathbf{\Phi}} \\ D_n\sim C_n}} \left\{ \|V^\pi(S) - (V(S) + \hat{e}_{\mathbf{\Phi}}(S))\|_\rho \leq \gamma_0 \|V^\pi(S) - V(S)\|_\rho \right\} \geq 1 - \xi. \tag{9.11}$$

Lemma 9.3 shows that with enough sampled transitions, using random projections of size $d = \tilde{O}\left((\frac{1+\gamma}{\gamma_0-\gamma})^2 k \log\frac{D}{\xi}\right)$ guarantees contraction in the error by a factor of $\gamma_0$. Using union bound over $m$ iterations of the algorithm, we prove that projections of size $d = \tilde{O}\left((\frac{1+\gamma}{\gamma_0-\gamma})^2 \log\frac{mD}{\xi}\right)$ and a sample of transitions of size $n = \tilde{O}\left((\frac{1+\gamma}{\gamma_0-\gamma})^2 d^2 \log\frac{md}{\xi}\right)$ are enough to shrink the error by a factor of $\gamma_0^m$ after $m$ iterations.

## 9.4 Proof of SCBEBF Error Lemmas

*Proof of Lemma 9.2.* We have that $V^\pi$ is the fixed point to the Bellman operator (i.e. $\mathbb{T}^\pi V^\pi = V^\pi$), and that the operator is a contraction with respect to the weighted $\ell^2$ norm on the stationary distribution $\rho$ (Van Roy, 1998):

$$\left\|\mathbb{T}^\pi V(S) - \mathbb{T}^\pi V'(S)\right\|_\rho \leq \gamma \left\|V(S) - V'(S)\right\|_\rho . \tag{9.12}$$

We thus have:

$$\left\|V^\pi(S) - (V(S) + \hat{e}_V(S))\right\|_\rho \tag{9.13}$$

$$\leq \left\|V^\pi(S) - \mathbb{T}^\pi V(S)\right\|_\rho + \left\|(\mathbb{T}^\pi V(s) - V(s)) - \hat{e}_V(S)\right\|_\rho \tag{9.14}$$

$$\leq \left\|\mathbb{T}^\pi V^\pi(S) - \mathbb{T}^\pi V(S)\right\|_\rho + \epsilon \left\|\mathbb{T}^\pi V(S) - V(S)\right\|_\rho \tag{9.15}$$

$$\leq \gamma \left\|V^\pi(S) - V(S)\right\|_\rho + \epsilon \left\|\mathbb{T}^\pi V(S) - \mathbb{T}^\pi V^\pi(S)\right\|_\rho + \epsilon \left\|V^\pi(S) - V(S)\right\|_\rho \tag{9.16}$$

$$\leq (\gamma + \epsilon\gamma + \epsilon) \left\|V^\pi(S) - V(S)\right\|_\rho . \tag{9.17}$$

In Line 9.15 we used the error assumption of the lemma and the fix point property of the Bellman operator. □

*Proof of Lemma 9.3.* Let $\epsilon = (\gamma_0 - \gamma)/(1+\gamma)$, $c_4 = 25600$ and $c_5 = 64$. Substituting $d$, and $n$ into Theorem 9.1, with probability $1 - \xi$ we get after simplification: $\left\|\hat{e}_{\mathbf{\Phi}}(S) - e_V(S)\right\|_\rho \leq \epsilon \left\|e_V(S)\right\|_\rho$. Proof follows immediately by an application of Lemma 9.2. □

## 9.5 Empirical Evaluations

We conduct a series of experiments to evaluate the performance of our algorithm and compare it against viable alternatives. Experiments are performed using a simulator that models an autonomous helicopter in the flight regime close to hover (Ng et al., 2006). Our goal is to evaluate the value function associated with the manually tuned policy provided with the simulator. We let the helicopter free fall for 5 time-steps before the policy takes control. We then collect 100 transitions while the helicopter hovers. We run this process multiple times to collect more trajectories on the policy.

The original state space of the helicopter domain consists of 12 continuous features. 6 of these features corresponding to the velocities and position, capture most of the data needed for policy evaluation. We use tile-coding on these 6 features as follows: 8 randomly positioned grids of size $16 \times 16 \times 16$ are placed over forward, sideways and downward velocity. 8 grids of similar structure are placed on features corresponding to the hovering coordinates. The constructed feature space is thus of size 65536. Note that our choice of tile-coding for this domain is for demonstration purposes.

Since the true value function is not known in our case, we evaluate the performance of the algorithm by measuring the *normalized return prediction error* (NRPE) on a large test set. Let $U(S_i)$ be the empirical return observed for $S_i$ in a testing trajectory, and $\bar{U}$ be its average over the testing measure $\mu(S)$. We define $\text{NRPE}(V) = \|U(S) - V(S)\|_\mu / \left\|U(S) - \bar{U}\right\|_\mu$. Note that the best constant predictor has NRPE $= 1$.

We start by an experiment to observe the behaviour of the prediction error in SCBEBF as we run more iterations of the algorithm. We collect 3000 sample transitions for training. We experiment with 3 schedules for the projection size:

(1) Fix $d = 300$ for 300 steps. (2) Fix $d = 30$ for 300 steps. (3) Let $d$ decrease with each iteration $i$: $d_i \overset{\text{def}}{=} \lfloor 300e^{-i/30} \rfloor$.

Figure 9–1 shows the error averaged over 5 runs. When $d$ is fixed to a large number, the prediction error drops rapidly, but then rises due to over-fitting. This problem can be mitigated by using a smaller fixed projection size at the cost of slower convergence. In our experiments, we find a gradual decreasing schedule to provide fast and robust convergence with minimal over-fitting effects.

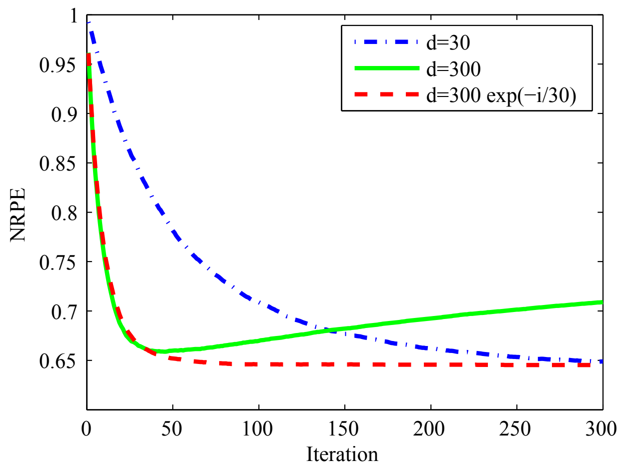


Figure 9–1: NRPE of SCBEBF for $n = 3000$ and different number of projections, under different choices of $d$, averaged over 5 runs. 95% confidence intervals of the averages are tight (less than 0.005 in width) and are not shown.

We next compare SCBEBF against other alternatives. There are only a few methods that can be compared against our algorithm due to the high dimensional feature space. We compare against Compressed LSTD (CLSTD) (Ghavamzadeh et al., 2010), $\ell^2$-Regularized LSTD using a Biconjugate gradient solver (L2-LSTD), and $\ell^1$-Regularized LSTD using LARS-TD (Kolter and Ng, 2009a) with a

Biconjugate gradient solver in the inner loop (L1-LSTD). These conjugate gradient solvers exploit the sparsity of the feature space to converge faster to the solution of linear equations (Barrett et al., 1987). We avoided online and stochastic gradient type methods as they are not very efficient in sample complexity.

We compare the described methods while increasing the size of the training set. The projection schedule for SCBEBF is set to $d = \lfloor 500e^{-i/300} \rfloor$ for all sample sizes [2] . The regularization parameter of L2-LSTD was chosen among a small set of values using 1/5 of the training data as validation set. Due to memory and time constraints, the optimal choice of parameters could not be set for CLSTD and L1-LSTD. The maximum size of projection for CLSTD and the maximum number of non-zero coefficients for L1-LSTD was set to 3000. CLSTD would run out of memory and L1-LSTD would take multiple hours to run if we increase these limits.

The results, averaged over 5 runs, are shown in Figure 9–2. We see that L2-LSTD outperforms other methods, closely followed by SCBEBF. Not surprisingly, L1-LSTD and CLSTD are not competitive here as they are suboptimal with the mentioned constraints. This is a consequence of the fact that these algorithms scale worse with respect to memory and time complexity.

We conjecture that L2-LSTD is benefiting from the sparsity of the features space, not only in running time (due to the use of conjugate gradient solvers), but also in sample complexity. This makes L2-LSTD an attractive choice when the

[2] The schedule of the projection sizes is different from the first experiment since the training sets can be larger

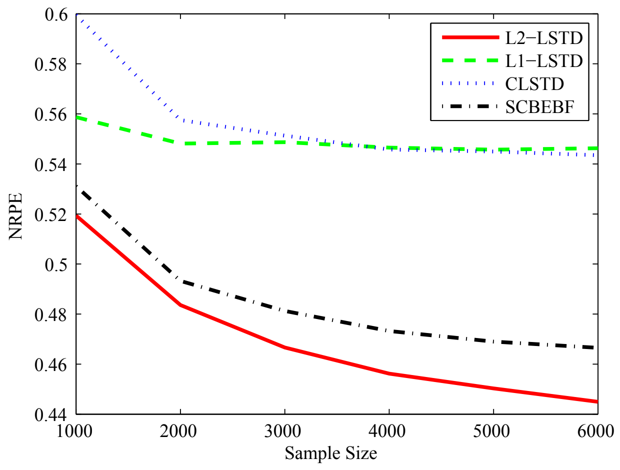


Figure 9–2: Right: Comparison of the prediction error of different methods for varying sample sizes. 95% confidence intervals of averages are tight (less than 0.005 in width) and are not shown.

features are observed in the sparse basis. However, if the features are sparse in some unknown basis (observation is not sparse), then the time complexity of any linear solver in the observation basis can be prohibitive. SCBEBF, however, scales much better in such cases as the main computation is done in the compressed space.

To highlight this effect, we construct an experiment in which we gradually increase the number of non-zero features using a change of basis. This is done by partitioning the features into random blocks and multiplying each block by a randomly generated basis. The size of these blocks will influence the observed sparsity of the features. Note that since we only apply a change of basis to the features, the error of both L2-LSTD and SCBEBF remain unchanged as

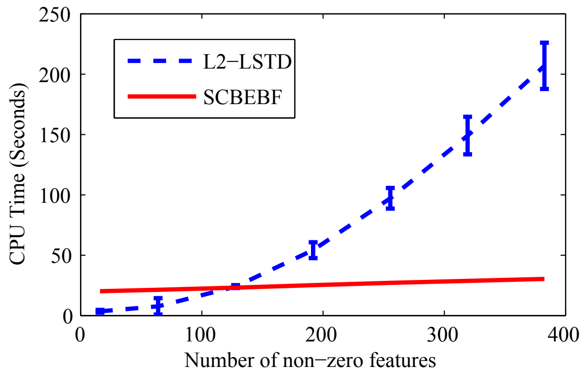


Figure 9–3: Runtime of L2-LSTD and SCBEBF with varying observation sparsity.

predicted by the theory. We thus only compare the running times as we change the observation sparsity.

Figure 9–3 shows the CPU time used by each methods with sample size of 3000, averaged over 5 runs (using Matlab on a 3.2GHz Quad-Core Intel Xeon processor). We run 100 iterations of SCBEBF with $d = \lfloor 300e^{-i/30} \rfloor$ (as in the first experiment), and set the regularization parameter of L2-LSTD to the optimal value. We can see that the running time L2-LSTD quickly becomes prohibitive with the decreased observation sparsity, whereas the running time of SCBEBF grows very slowly (and provably linearly) with the number of non-zero features.

## 9.6 Related Work

Random projections have been studied extensively in signal processing (Candès et al., 2006; Candès and Wakin, 2008) as well as machine learning (Maillard and Munos, 2012, 2009; Zhou et al., 2007). In reinforcement learning, Ghavamzadeh et al. (2010) have used random projections in conjunction with

LSTD and have shown that this can reduce the estimation error, at the cost of a controlled bias.

Instead of compressing the feature space for LSTD, we focus on the BEBF generation setting, which offers better scalability and more flexibility in practice. Our algorithm is well suited for sparse feature spaces, naturally occurring in domains with audio and video inputs (Olshausen et al., 2001), and also in tile-coded and discretized spaces.

Compared to other regularization approaches to RL (Kolter and Ng, 2009a; Farahmand et al., 2009b; Johns et al., 2010), our random projection method does not require complex optimization, and thus is faster and more scalable. If features are observed in the sparse basis, then conjugate gradient solvers can be used for regularized value function approximation. However, CBEBF seems to have better performance with smaller sample sizes and provably works under any observation basis.

We have also applied compressive regression to the PSR learning problem, along with the commonly used spectral learning methods used for learning the PSR parameters (Hamilton et al., 2013). An extended analysis shows that with the typical sparse structure in PSR dynamical models, we can gain both in computational and sample complexity by applying regression in a compressed space (Hamilton et al., 2014).

## 9.7 Discussion and Future Work

This chapter introduced a simple, fast and robust feature extraction algorithm for policy evaluation in sparse and high dimensional state spaces. Using recent results on the properties of random projections, we prove that in sparse spaces, random projections of sizes logarithmic in the original dimension are sufficient to preserve linearity. Therefore, BEBFs can be generated on compressed spaces induced by small random projections. Our finite sample analysis provides guarantees on the reduction in prediction error after the addition of such BEBFs.

Our assumption of the linearity of the Bellman error in the original feature space might be too strong for some problems. This assumption is used to simplify the analysis. However, most of the discussion can be rephrased in terms of the *projected Bellman error*, and we expect this approach to carry through and provide more general results (e.g. see Parr et al. (2007)).

Our empirical analysis on a high dimensional space with unknown true value function shows that CBEBF vastly outperforms LSTD with random projections and easily scales to large problems. It is also more consistent (i.e., has lower variance) and has a much smaller memory complexity. Both methods are significantly more sample-efficient than gradient-based TD methods. We expect this behaviour to be common for many RL problems. However, more empirical analysis should be done to confirm this hypothesis. Since the focus of this work is on feature extraction with minimal domain knowledge, using agnostic random projections, we avoided the commonly used toy problem domains with known structure in the value function (e.g. mountain car (Sutton and Barto, 1998)), where algorithms can exploit what is known about the problem.

Finding the optimal choice of the projection size schedule and the number of iterations is an interesting subject of future research. We expect the use of cross-validation to suffice for the selection of the optimal parameters, due to the robustness that we observed in the results of the algorithm. A tighter theoretical bound might also help provide an analytical, closed-form answer to how parameters should be selected. One would expect a slow reduction in the projection size to be favourable.

# CHAPTER 10
# Conclusion and Discussion

Reinforcement learning has seen substantial improvements in the past few years, with many researchers working on both the theory and the application sides of the problem. Many algorithms have been developed to more efficiently handle larger and more complex problems, and have shown relatively good performance on traditional RL benchmarks. But as of writing of this thesis, RL is not yet the go-to method when it comes to real-world sequential planing and decision making problems. This is partly because the general RL problem is very difficult or too costly to solve in practice. Principled methods fail to scale to realistic large domains, and those with partial practical success depend on heuristics with no real theoretical guarantees.

To make RL algorithms practical, we need ways to control and help with the inherent complexity of the learning problem. This can be done in two general manners: using domain-dependent knowledge of the problem to kick start the learning process, and using domain-independent assumptions that are observed or are likely to hold in most problems. Domain dependent knowledge can come from human experts or from previous experience in similar RL domains. In the absence of such knowledge, we can resort to methods that exploit common structures or assumptions in real-world learning problems.

With the goal of making RL more practical in costly real-world problems, this thesis presents two new classes of model-selection and regularization techniques suited for handling unknown dynamics and utilities in the reinforcement learning problem. The PAC-Bayesian methods make use of domain knowledge in form of a prior distribution. We provide model-section methods and frequentist analysis on PAC-Bayesian RL when a prior is provided either on the model dynamics or the utility of actions. We also use agnostic random projection to extract features in domains where we have little domain knowledge and can only assume a sparse feature space.

PAC-Bayesian analysis of RL is often favourable to Bayesian RL methodology, since it can provide frequentist guarantees regardless of the correctness of the given prior distribution. It is also favourable to pure frequentist methods, since it can leverage the prior knowledge incorporated in the prior when it matches with the observed data. It has the superior sample efficiency of a Bayesian method and the strong theoretical guarantees of a frequentist one.

PAC-Bayesian analysis is particularly interesting for value function prediction, as it provides simultaneous bounds over all posterior measures. As we discussed in Chapter 6, this can be used to derive margin bounds for the model-selection problem in RL. The PAC-Bayesian analysis can also be used in parametric policy search methods, where a prior is provided on the parameter space, to produce a simultaneous error bound for all parametric policies. Combined with importance weighting methods, one can can hope to derive PAC RL policy search algorithms with global performance guarantees. Design and analysis of such learning method are interesting avenues of future work.

When little is known about the dynamics of the underlying Markovian process or the utility of actions, one can only leverage the knowledge about the observed features or rewards to help with the model selection and regularization. A discussed in Chapters 8 and 9, many problem domains have a sparse input structure under a known or unknown basis. Such sparse input structures allow us to use techniques from the compressed sensing literature to project the data into a lower-dimensional space and still retain enough information to construct a linear function approximator with controllable induced bias.

Using compressed regression techniques developed in Chapter 8, we construct a feature extractor for the RL problem that avoids working with the original high dimensional state features. By applying regression in the compressed space, the generated features are closer to the target of the regression by the reduction in the estimation error, at the expense of a small and controllable approximation error. Our empirical results on a domain with non-linear dynamics and a large discretized state-space show-cased the use of such methods to improve value function approximation under minimal domain knowledge. A thorough empirical analysis and comparison of the proposed method with non-linear value function approximations techniques could help in better assessing the effectiveness of compression in the overall prediction error.

Since linear regression is at the heart of many machine learning algorithms, other methods can also benefit from the analysis of compressed regression under input sparsity assumptions. We have thus far applied and analysed the method in the RL setting with value function approximation and with PSR parameter learning problem (Hamilton et al., 2013, 2014). In both cases, we observe substantial gains in computational and sample complexity of the learning method.

With the ever-increasing size of input data, both in the number of observed features and the sample count, we need methods that can scale better with the input dimension size and can run in linear time with respect to the sample size. Random projections, in particular, can provide fast and robust dimensionality reduction and still retain enough data to avoid degradation in the overall output.

Under sparsity assumptions of the data and with batch learning methods, we showed that the projection size can be significantly smaller than the original dimensionality of the system and still provide similar or better results. The analysis of random compression with online learning methods can extend these results to linear time algorithms. Gradient-descent methods with linear value function approximation have shown to be very effective with many instances of the RL problems, but they tend to be highly sample inefficient. One can hope to get better sample complexity with these methods when they are adapted to work in a compressed feature space.

This work helps bring a new set of methods to the regularization arsenal available for solving the RL problem under unknown dynamics. We studied both domain-dependent methods based on prior distributions, and agnostic compression techniques using minimal assumptions on the domain and the structure of the features. The wide gap in between these extremes needs to be filled in with regularization that benefit from the domain knowledge in forms other than a prior distribution or use assumptions other than the sparsity conditions on the observed features. These remain to be interesting areas for future work.

# Appendices

# APPENDIX A
# Notation Guide

Below is a list of notation throughout the thesis. Other notations used later in this work will be explained as needed.

## A.1 Basic Notation

- Random variables (if not vectors) are represented by capital letters (e.g. $S_t$).
- Observed values are represented by lower case letter (e.g. $s_t$).
- Sets are represented by curly capital letters (e.g. $\mathcal{A}$).
- The set of all measures over a set $\mathcal{A}$ is represented by $\mathcal{M}(\mathcal{A})$.
- Matrices and vectors are represented by bold letter (e.g. $\mathbf{U}$, $\mathbf{v}$).
- $S_{i:j}$ represent the series $S_i, S_{i+1}, \ldots, S_j$.
- $\mathbf{U}_i$ and $\mathbf{U}_{\cdot,j}$ are the $i$th row and the $j$th column of $\mathbf{U}$ respectively.
- $\mathbf{I}$ is the identity matrix of appropriate size.
- $\mathbf{0}$ is the zero vector of appropriate size.

## A.2 Probability and Expectation

- $\mathbb{P}(X)$ denotes the probability distribution over $X$ implied by the context.
- $\underset{X\sim\mu}{\mathbb{P}\mathrm{r}}\{e(X)\}$ denotes the probability of event $e(X)$ when $X$ is sampled from the distribution $\mu$.
- $\underset{X\sim\mu}{\mathbb{E}}[f(X)]$ denotes the expected value of $f(X)$ when $X$ is sampled from the distribution $\mu$.

- $\underset{X\sim\mu}{\mathbb{V}\mathrm{ar}}\left[f(X)\right]$ denotes the variance of $f(X)$ when $X$ is sampled from the distribution $\mu$.
- $\mu(\mathcal{A})$ is the mass of the measure $\mu$ on the set $\mathcal{A}$. $\mu(X)$ is the density function of measure $\mu$ (if defined).
- $\|\rho-\mu\|_{\mathrm{TV}} = \sup_{\mathcal{A}} |\rho(\mathcal{A})-\mu(\mathcal{A})|$ is the total variation distance between two probability measures $\rho$ and $\mu$.

To simplify the notation, the distribution can be dropped if it is implied by the context.

## A.3 Norms and Projections

- $\|f(X)\|_{\nu}$, $\|\mathbf{v}\|$ and $\|\mathbf{U}\|$ denote the $\ell^2$ norm over measurable functions, vectors and matrices respectively:

$$\|f(X)\|_{\nu}^2 \overset{\text{def}}{=} \underset{X\sim\nu}{\mathbb{E}}\left[(f(X))^2\right], \tag{A.1}$$

$$\|\mathbf{v}\|^2 \overset{\text{def}}{=} \sum_i \mathbf{v}_i^2, \tag{A.2}$$

$$\|\mathbf{U}\| \overset{\text{def}}{=} \sup_{\mathbf{v},\|\mathbf{v}\|=1} \|\mathbf{U}\mathbf{v}\|. \tag{A.3}$$

- $\|f(X)\|_{\infty}$, $\|\mathbf{v}\|_{\infty}$ is the infinity norm:

$$\|f(X)\|_{\infty} \overset{\text{def}}{=} \sup_X f(X), \tag{A.4}$$

$$\|\mathbf{v}\|_{\infty} \overset{\text{def}}{=} \sup_i \mathbf{v}_i. \tag{A.5}$$

- $\|\mathbf{v}\|_1$ denotes the $\ell^1$ norm:

$$\|\mathbf{v}\|_1 \overset{\text{def}}{=} \sum_i |\mathbf{v}_i|. \tag{A.6}$$

- $\|\mathbf{v}\|_0$ denotes the is Donoho's zero "norm", indicating the number of non-zero elements in a vector:

$$\|\mathbf{v}\|_0 \overset{\text{def}}{=} \sum_i \mathbb{I}_{\{\mathbf{v}_i \neq 0\}}. \tag{A.7}$$

- $\|\mathbf{U}\|_F$ denotes Frobenius norm:

$$\|\mathbf{U}\|_F^2 \overset{\text{def}}{=} \sum_{i,j} \mathbf{U}_{i,j}^2. \tag{A.8}$$

- $\Pi_\nu^{\mathcal{F}}$ and $\Pi^{\mathcal{F}}$ denotes projections with respect to the $\ell^2$ norm:

$$\Pi_\nu^{\mathcal{F}} f \overset{\text{def}}{=} \inf_{f' \in \mathcal{F}} \|f' - f\|_\nu , \tag{A.9}$$

$$\Pi^{\mathcal{F}} \mathbf{v} \overset{\text{def}}{=} \inf_{\mathbf{v}' \in \mathcal{F}} \|\mathbf{v}' - \mathbf{v}\| . \tag{A.10}$$

## A.4 Other Miscellaneous Notation

- $\nabla_\theta f_\theta$ denotes the gradient of $f$ with respect to parameter vector $\theta$.
- $\mathbf{U}^\dagger$ is the Moore–Penrose pseudoinverse of $\mathbf{U}$.
- $(f)_+ = \max(f, 0)$ is the positive part of $f$.
- $\tilde{O}(\cdot)$ is the asymptotic $O(\cdot)$ notation with term logarithmic in relevant variables removed.

# References

D. Achlioptas. Database-friendly random projections. In *ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2001.

A. Ambroladze, E. Parrado-Hernandez, and J. Shawe-Taylor. Tighter PAC-Bayes bounds. In *Advances in Advances in Neural Information Processing Systems*, 2006.

D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4): 343–370, 1988. ISSN 0885-6125.

A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, 2009.

P. Auer and R. Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1):55–65, 2010.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

A. Banerjee. On bayesian bounds. In *International Conference on Machine learning*, 2006.

R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

R. G. Baraniuk and M. B. Wakin. Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1):51–77, 2009.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. Number 43. Society for Industrial and Applied Mathematics, 1987.

P. Bartlett and W. Maass. Vapnik-Chervonenkis dimension of neural nets. *The handbook of brain theory and neural networks*, pages 1000–1003, 1995.

R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming (Optimization and Neural Computation Series, 3)*. Athena Scientific, 1996. ISBN 1886529108.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery. Vol*, 36(4):929–965, 1989.

V. S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

J. Boyan and A. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Advances in Neural Information*

*Processing Systems*, 1995.

J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002.

R. I. Brafman and M. Tennenholtz. R-max – A general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

E. Brunskill, B. Leffler, L. Li, M. Littman, and N. Roy. CORL: A continuous-state offset-dynamics reinforcement learner. In *Uncertainty in Artificial Intelligence*, 2008.

K. Bush, J. Pineau, A. Guez, B. Vincent, G. Panuccio, and M. Avoli. Dynamic representations for adaptive neurostimulation treatment of epilepsy. In *Fourth International Workshop on Seizure Prediction*, 2009.

E. Candès and M. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

E. J. Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

M. A. Davenport, M. B. Wakin, and R. G. Baraniuk. Detection and estimation with compressive measurements. Technical report, Dept. of ECE, Rice University, 2006.

M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk. Signal processing with compressive measurements. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):445–460, 2010.

D. Di Castro and S. Mannor. Adaptive bases for reinforcement learning. *Machine Learning and Knowledge Discovery in Databases*, 6321:312–327, 2010.

D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52 (4):1289–1306, 2006.

M. O. G. Duff. *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst, 2002.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: The Gaussian process approach to temporal difference learning. In *International Conference on Machine Learning*, 2003.

Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with Gaussian processes. In *International Conference on Machine Learning*, 2005.

D. Ernst, P. Geurts, L. Wehenkel, and M. L. Littman. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(4):503–556, 2005.

A. Farahmand and C. Szepesvári. Model selection in reinforcement learning. *Machine Learning*, 85:299–332, 2011.

A. Farahmand, M. Ghavamzadeh, C. Szepesvári, and S. Mannor. Regularized fitted Q-iteration for planning in continuous-space Markovian decision problems. In *American Control Conference*, 2009a.

A. M. Farahmand, M. Ghavamzadeh, C. Szepesvári, and S. Mannor. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, 2009b.

M. Fard, Y. Grinberg, J. Pineau, and D. Precup. Compressed least-squares regression on sparse spaces. In *AAAI Conference on Artificial Intelligence*, 2012.

M. M. Fard and J. Pineau. PAC-Bayesian model selection for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2010.

M. M. Fard, J. Pineau, and C. Szepesvári. PAC-Bayesian policy evaluation for reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

M. M. Fard, Y. Grinberg, A. Farahmand, J. Pineau, and D. Precup. Bellman error based feature generation using random projections on sparse spaces. In *Advances in Neural Information Processing Systems*, 2013.

C. Fiechter. Efficient reinforcement learning. In *Annual Conference on Computational Learning Theory*, 1994.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, 1995.

A. Geramifard, F. Doshi, J. Redding, N. Roy, and J. How. Online discovery of feature dependencies. In *International Conference on Machine Learning*, 2011.

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. A PAC-Bayes risk bound for general loss functions. In *Advances in Advances in Neural Information Processing Systems*, 2006.

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning*, 2009.

M. Ghavamzadeh, A. Lazaric, O. Maillard, and R. Munos. LSTD with random projections. In *Advances in Neural Information Processing Systems*, 2010.

P. W. Glynn. Likelilood ratio gradient estimation: an overview. In *Conference on Winter Simulation*, 1987.

W. L. Hamilton, M. M. Fard, and J. Pineau. Modelling sparse dynamical systems with compressed predictive state representations. In *International Conference on Machine Learning*, volume 28, 2013.

W. L. Hamilton, M. M. Fard, and J. Pineau. Efficient learning and planning with compressed predictive states. *Journal of Machine Learning Research (Under Review)*, 2014.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Verlag, 2009.

D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial intelligence*, 36(2):177–221, 1988.

C. Hegde, M. Wakin, and R. Baraniuk. Random projections for manifold learningproofs and analysis. In *Advances in Neural Information Processing Systems*, 2007.

M. Heger. Consideration of risk in reinforcement learning. In *International Conference on Machine Learning*, 1994.

R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers. *IEEE Transactions on Information Theory*, 48(12):3140–3150, 2002.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory*, 2012.

T. Jie and P. Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Advances in Neural Information Processing Systems*, 2010.

J. Johns, C. Painter-Wakefield, and R. Parr. Linear complementarity for regularized policy evaluation and improvement. In *Advances in Advances in Neural Information Processing Systems*, 2010.

W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

A. Kabán. A new look at compressed ordinary least squares. In *International Conference on Data Mining Workshops (ICDMW)*, 2013.

A. Kabán. New bounds on compressive linear least squares regression. In *International Conference on Artificial Intelligence and Statistics*, 2014.

S. Kakade, M. Kearns, and J. Langford. Exploration in metric state spaces. In *International Conference on Machine Learning*, 2003.

M. Kearns and D. Koller. Efficient reinforcement learning in Factored MDPs. In *International Joint Conference on Artificial Intelligence*, 1999.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

P. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *International Conference on Machine Learning*, 2006.

J. Kolter and A. Ng. Regularization and feature selection in least-squares temporal difference learning. In *International Conference on Machine Learning*, 2009a.

J. Z. Kolter and A. Y. Ng. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning*, 2009b.

M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003. ISSN 1532-4435.

J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005. ISSN 1532-4435.

J. Langford and M. Seeger. Bounds for averaging classifiers. Technical report, CMU-CS-01-102, Carnegie Mellon University, School of Computer Science, 2001.

J. Langford and J. Shawe-Taylor. PAC-Bayes and margins. In *Advances in Advances in Neural Information Processing Systems*, 2002.

J. Langford, M. Seeger, and N. Megiddo. An improved predictive accuracy bound for averaging classifiers. In *International Conference on Machine Learning*, 2001.

Last.fm. Web API. `http://www.last.fm/api`, Jan 2014.

A. Lazaric, M. Ghavamzadeh, R. Munos, et al. Finite-sample analysis of lstd. In *International Conference on Machine Learning*, 2010.

B. London, B. Huang, B. Taskar, and L. Getoor. Pac-bayes generalization bounds for randomized structured prediction. In *NIP Workshop on Perturbation, Optimization and Statistics*, 2013.

G. Lorentz, M. von Golitschek, and Y. Makovoz. *Constructive approximation: advanced problems*, volume 304. Springer Berlin, 1996.

H. R. Maei and R. S. Sutton. Gq ($\lambda$): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Conference on Artificial General Intelligence*, 2010.

H. R. Maei, C. Szepesvári, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems*, 2009.

H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *International Conference on Machine Learning*, 2010.

O. Maillard and R. Munos. Compressed least-squares regression. In *Advances in Neural Information Processing Systems*, 2009.

O.-A. Maillard and R. Munos. Linear regression with random projections. *The Journal of Machine Learning Research*, 13(1):2735–2772, 2012.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, IEEE Conference on*, 2008.

S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

ISSN 0025-1909.

P. Manoonpong, F. Wörgötter, and J. Morimoto. Extraction of reward-related feature space using correlation-based and reward-based learning methods. *Neural Information Processing. Theory and Algorithms*, 6443:414–421, 2010.

D. McAllester. A pac-bayesian tutorial with a dropout bound. *CoRR*, abs/1307.2118, 2013.

D. A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3): 355–363, 1999a.

D. A. McAllester. PAC-Bayesian model averaging. In *Annual Conference on Computational Learning Theory*, 1999b.

F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation. In *International Conference on Machine Learning*, 2008.

I. Menache, S. Mannor, and N. Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1): 215–238, 2005.

S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2009.

J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, pages 363–372. Springer, 2006.

B. Olshausen, P. Sallee, and M. Lewicki. Learning sparse image codes using a wavelet pyramid architecture. In *Advances in Advances in Neural Information Processing Systems*, 2001.

D. Ormoneit and Ś. Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

C. Paige and M. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8 (1):43–71, 1982.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value-function approximation. In *International Conference on Machine Learning*, 2007.

G. A. Rummery and M. Niranjan. On-line q-learning using connectionist systems. Technical report, 1994.

P. M. Samson. Concentration of measure inequalities for Markov chains and $\phi$-mixing processes. *Annals of Probability*, 28(1):416–461, 2000.

M. Sato and S. Kobayashi. Variance-penalized reinforcement learning for risk-averse asset allocation. In *International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents*, 2000.

K. Scheinberg and I. Rish. Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach. *Machine Learning and Knowledge Discovery in Databases*, 6323:196–212, 2010.

Y. Seldin, N. Cesa-Bianchi, F. Laviolette, P. Auer, J. Shawe-Taylor, and J. Peters. PAC-Bayesian analysis of the exploration-exploitation trade-off. In *On-line Trading of Exploration and Exploitation 2, ICML workshop*, 2011a.

Y. Seldin, F. Laviolette, J. Shawe-Taylor, J. Peters, and P. Auer. Pac-bayesian analysis of martingales and multiarmed bandits. Technical report, 2011b.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. Pac-bayesian inequalities for martingales. *Information Theory, IEEE Transactions on*, 58(12):7086–7093, 2012.

J. Shawe-Taylor and R. Williamson. A PAC analysis of a Bayesian estimator. In *Annual Conference on Computational Learning Theory*, 1997.

S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38 (3):287–308, 2000.

A. Strehl and M. Littman. An empirical evaluation of interval estimation for Markov decision processes. In *International Conference on Tools with Artificial Intelligence*, 2004.

A. Strehl and M. Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8): 1309 – 1331, 2008. ISSN 0022-0000.

A. L. Strehl and M. L. Littman. A theoretical analysis of model-based interval estimation. In *International Conference on Machine Learning*, 2005.

M. J. A. Strens. A Bayesian Framework for Reinforcement Learning. In *International Conference on Machine Learning*, 2000.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning*, 2009.

C. Szepesvari. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

B. Van Roy. *Learning and value function approximation in complex decision processes*. PhD thesis, Massachusetts Institute of Technology, 1998.

V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982. ISBN 0387907335.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *International Conference on Machine Learning*, 2005.

C. J. C. H. Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge, 1989.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. Technical report, Information Theory Research Group, HP Laboratories, 2003.

M. Wiering and J. Schmidhuber. Efficient model-based exploration. In *International Conference on Simulation of Adaptive Behavior*, 1998.

R. J. Williams and L. C. Baird. Tight performance bounds on greedy policies based on imperfect value functions. Technical Report NU–CCS-93-14, Northeastern University, Nov 1993.

J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:210–227, 2008.

S. Zhou, J. Lafferty, and L. Wasserman. Compressed regression. In *Advances in Advances in Neural Information Processing Systems*, 2007.

## KEY TO ABBREVIATIONS

AI: Artificial Intelligence

BEBF: Bellman Error Based feature

BR: Bellman Residual loss

BRM: Bellman Residual Minimization

CBEBF: Compressed Bellman Error Based Feature Generation

CDF: Cumulative Density Function

CS: Compressed Sensing

CSP: Compressed Signal Processing

GPI: Generalized Policy Iteration

GTD: Gradient Temporal Difference learning

i.i.d.: Independent and Identically Distributed

iFDD: Incremental Feature Dependency Discovery

JL (Lemma): Johnson–Lindenstrauss (Lemma)

LARS: Least Angle Regression

LMS: Least-Mean Squares

LSPI: Least-Squares Policy Iteration

LSTD: Least-Squares Temporal Difference

MBIE-EB: Model Based Interval Estimation with Exploration Bonus

MBIE: Model-Based Interval Estimation

| | |
|---|---|
| MDP: | Markov Decision Process |
| PAC: | Probably Approximately Correct |
| PB: | Projected Bellman loss |
| PDF: | Probability Density Function |
| PI: | Policy Iteration |
| PSR: | Predictive State Representation |
| RIP: | Restricted Isometry Property |
| RL: | Reinforcement Learning |
| SCBEBF: | Simplified Compressed Bellman Error Based Feature Generation |
| SRM: | Structural Risk Minimization |
| SVM: | Support Vector Machines |
| TD: | Temporal Difference |
| TDC: | Temporal Difference learning with Corrections |
| VC Dimension: | Vapnik–Chervonenkis Dimension |