Université de Lille

Thèse De Doctorat

Spécialité **Informatique**

Présenté par

Mahsa Asadi

Identifying structure in online and collaborative learning problems.

Identifier la structure des problèmes d'apprentissage en ligne et collaboratif.

Thèse soutenue le 25 novembre 2022 devant le jury composé de :

Matthieu Geist	Professeur, Université de Lorraine, Google Brain	Rapporteur
Hachem Kadri	Professeur agrégé,Laboratoire d'Informatique, & Systèmes, Aix-Marseille Université	Rapporteur
Ciara Pike-Burke	Maître de conférences, Department of Mathematics, Imperial College London	Examinateur
Laetitia Jourdan	Professeur, CRIStAL, Université de Lille	Examinateur & Président
Aurélien Bellet	Chargé de recherche, CRIStAL, Inria	Co-Directeur de Thèse
Marc Tommasi	Professeur, CRIStAL, Université de Lille	Directeur de Thèse
Odalric Maillard	Chargé de recherche, CRIStAL, Inria	Invité

Laboratoire CRIStAL, équipe Magnet

France

Declaration of Authorship

I, Mahsa Asadi, declare that this thesis titled, "Identifying structure in online and collaborative learning problems" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Mahsa Asadi

Date: October 2022

"*Jungle doesn't know its countless dimensions yet*"

Sohrab Sepehri

LILLE UNIVERSITY

Abstract

Inria

Doctor of Philosophy

Identifying structure in online and collaborative learning problems

by Mahsa Asadi

Nowadays it is commonplace to deal with large scale problems and should we take problem structure into account, it could assist us toward improving learning performance. In this work, we have proposed approaches that take into account the structure in two settings: (i) model-based reinforcement learning problems where we have reduced the regret (ii) online personalized mean estimation problems where we have reduced the sample complexity for mean estimation.

LILLE UNIVERSITY

Résumé

Inria

Doctor of Philosophy

Identifier la structure des problèmes d'apprentissage en ligne et collaboratif

by Mahsa Asadi

De nos jours, il est courant de traiter des problèmes à grande échelle et si nous tenons compte de la structure du problème, cela pourrait nous aider à améliorer les performances d'apprentissage. Dans ce travail, nous avons proposé des approches qui prennent en compte la structure dans deux contextes: (i) des problèmes d'apprentissage par renforcement basés sur un modèle où nous avons réduit le regret (ii) des problèmes d'estimation moyenne personnalisée en ligne où nous avons réduit la complexité de l'échantillon pour la moyenne estimation.

Acknowledgements

First of all, I would like to thank my supervisors Marc Tommasi and Aurélien Bellet who supported me at all times during my PhD. We explored various domains of machine learning together and it was definitely a nice journey. I learned a lot not only science and research but also about life from them. I am grateful for Marc's patience and the times we spent probing different problems and the discussions we had. And I am extremely thankful for Aurélien's encouragements and positiveness that motivated me to learn more and go further in research.

Secondly, I would like to thank Odalric Maillard who has not been my official supervisor but supervised me during this time. He taught me a lot about reinforcement learning and concentration inequalities. He always encouraged me to follow my passions and helped me doing so.

The next person I would like to thank is Sadegh Talebi who has been a supportive friend and nice teacher giving me constructive feedback about my work. Working with him for our first publication has been a nice team working experience.

I would like to thank my friends and colleagues at MAGNET team, Mariana Vargas Vieyra, Cesar Sabater, Arijus Pleska, Brij Mohan Lal Srivastava, William de Vazelhes, Carlos Jorge Zubiaga Pena, Mikaela Keller, Moitree Basu, Onkar Pandit and Mathieu Dehouck and also my friends and colleagues at SCOOL(SequeL) team, Emilie Kaufmann, Michal Valko, Ronan Fruit, Edouard Leurent, Lilian Besson, Omar Darwiche Domingues, Xuedong Shang, Mathieu Seurin, Pierre Ménard, for being there during these PhD years, their friendship and kindness.

I was also lucky to find great friends in city of Lille where I stayed during my PhD who made life more enjoyable: Zahra Rad, Negissa Ebadi, Naghme Mobini, Ehsan Enferad, Shirin Enferad, Nastaran Esmailpour, Faezeh Ghasemi, MohammadHadi Fazeli, Amir Pourmoghaddam and Masoud Hasanvand.

I would also like to thank Abbas Mehrabian and Tohid Ahdifar for their life guidance and wise words and helping me a lot from far away in Canada.

And finally I would like to thank my mother, father and brother for their love and support.

Mahsa Asadi

Contents

List of Figures

List of Tables

To my mother and father who have always been there for me

Chapter 1

Introduction

With the growing use of large scale machine learning problems, the need for huge amount of data and also more time for the computation increases. This is needed to learn the machine learning model of the problem. This data can be huge and collection can be challenging. Of course, general machine learning algorithms assume that there exists structure so that we can learn a model using machine learning and the data. However, sometimes there exists some additional underlying form of structure in the machine learning problem. Therefore, one solution to the aforementioned problem is to identify and leverage the underlying structure so as to aggregate data and reuse for parts of the problem with similar structure. As a result, the models learn with less samples and require less time to collect those samples.

In this thesis, we have proposed novel approaches that exploit the notion of structure in two different online learning settings. The first contribution (Chapter 3) aims to improve the performance of model-based reinforcement learning (RL) approaches. We introduce a notion of similarity in order to reveal a potential structure in an RL problem. We show that the learning performance of these RL problems can gain advantage from the structure of the problem. The second contribution (Chapter 4) is to design collaborative algorithms to learn online personalized mean estimation in a network. An important building block is then to identify agents acquiring data from similar distributions. Again, we can exploit the structure revealed by the similarity to improve estimation models of the agents in the network. This is particularly difficult to do in an *online* setting, in which data becomes available sequentially over time and this is one of the challenges we face. Moreover, it is of great importance to point out that we do not share the data among agents and rather we exchange some statistics about the data. This helps in keeping the privacy of the users in case the data is sensitive and/or too large for efficient transmission.

The results in this thesis are also a first step towards a complementary line of research that is to design collaborative strategies for learners in a network. Learning in isolation and local processing is an option but it suffers from slow time when data arrives slowly. In that case, collaborative strategies can be investigated in order to increase statistical power and accelerate learning. Such collaborative approaches are broadly referred to as federated learning (Kairouz et al., 2021). Since the setting where (a lot of) agents evolve in different environment is widely spread (personal digital devices, Internet of Things, ubiquitous computing), the agent can collaborate with ones having similar environment. Our results prove that identifying structures in online learning problems can be efficient. Therefore, they can be at the root of the important problem to identify good peers based on the similarity of their data distribution when it arrives in an online fashion.

1.1 Summary of Contributions

It is considered "online learning" (Cesa-Bianchi and Lugosi, 2006) when the data becomes available sequentially over time and we learn from it. One important criterion to evaluate an online learning approach is called regret. Generally speaking, regret is the difference of evaluated agent's performance and that for the optimal agent as it measures how much the evaluated agent regrets.

Equivalence structure in reinforcement learning. Let us dive into online learning in reinforcement learning problems (Szepesvári, 2010). Here, the regret is defined as the difference between the total reward obtained using optimal policy and that received by the agent (learner). One can find the concrete definition of regret for reinforcement learning problems in Chapter 2, Def. 13.

In the first step we tackle model-based reinforcement learning (RL) problems and we assume to have a single agent in a huge environment. The goal of this agent is to reduce the regret. One component of learning in model-based RL problems is the model of the environment and should we improve learning this model, we will reduce the regret as well. By help of equivalent state-action pairs, we can better approximate agent model. To faster learn the model, we introduce a notion of similarity by taking advantage of problem structure and the statistical behavior of the state-action pair transition function. This leads to an equivalence relation and aggregating the information of similar state-action pairs reduces the regret. Leveraging an equivalence property knowing the model of the environment has been investigated in several studies. To the best of our knowledge, this is the first work providing regret bounds for RL when an equivalence structure is efficiently exploited. In other words, the bounds are studied when the model of environment is no longer known. We present C-UCRL as a natural modification of UCRL2 for RL and for the case of known equivalence structure, we show that C-UCRL improves over UCRL2 in terms of regret by a factor of $\sqrt{SA/C}$ in models with S states, A actions and C classes which corresponds to massive improvement when $C \ll SA$.

Collaborative learning in personalized mean estimation network. A machine learning approach is called decentralized when there exists a network of agents who are learning collaboratively but there is no centralized server. Here collaboration in machine learning is similar to what we have for other disciplines where collaborative learning is when working in a group of two or more to achieve a common goal (Roberts, 2004). In the second step, we analyze a decentralized collaborative online learning scenario for personalized mean estimation problem. We consider a complete graph of agents where each agent is doing mean estimation on each node. The means can be the same or different and each agent receives one sample at each time step from a σ-sub-Gaussian distribution and therefore, in an online fashion. The agent is also allowed to query one other neighbour at each time step. The goal of the learner is to estimate its mean as fast as possible. Collaboration is possible among agents and each agent is searching for the similar peers within the network; i.e. peers with the same mean, to enhance its learning performance by sharing of information. We assume the existence of an underlying class structure where agents in the same class have the same mean value and also an extension which considers agents that are η close to each other. We use optimism in face of uncertainty principle to find out if two agents are in the same class. Moreover, we provide class estimation and mean estimation time complexity.

1.2 Published Work

The contributions of this thesis has resulted into one publications and one preprint currently under review:

1. Asadi, M., Talebi, M.S., Bourel, H. and Maillard, O.A., 2019, October. Model-Based Reinforcement Learning Exploiting State-Action Equivalence. In Asian Conference on Machine Learning (pp. 204-219). PMLR. **[Best Student Paper Award]**

2. Asadi, M., Bellet, A., Maillard, O.A. and Tommasi, M., 2022, Collaborative Algorithms for Online Personalized Mean Estimation. [Under review in Transaction on Machine Learning Research (TMLR)]

1.3 Outline

The rest of this manuscript is organized into four chapters.

Chapter 2: Preliminaries. In Chapter 2, we introduce the background materials required to understand our contributions. We describe some concentration inequalities and review some key problem settings and associated algorithms in online learning: multi-armed bandits, pure exploration bandits and reinforcement learning problems.

Chapter 3: Model-based Reinforcement Learning Exploiting State-Action Equivalence. In Chapter 3, we focus on model-based reinforcement learning problems and propose a measure of state-action equivalence. Using the introduced notion of equivalence, we design algorithms that leverage this structure to improve the regret of UCRL2, a popular model-based reinforcement learning algorithm.

Chapter 4: Collaborative Algorithms for Online Personalized Mean Estimation. In Chapter 4, we consider a network of agents, each trying to estimate its personalized mean in an online fashion. We consider problems where several agents have the underlying same mean and introduce a collaborative learning algorithm that uses a notion of optimistic distance to identify the corresponding classes of agents. We provide theoretical guarantees for class estimation time complexity and mean estimation time complexity.

Chapter 5: Conclusion. This is the last chapter where we summarize our contributions and discuss possible paths for future work.

Chapter 2

Preliminaries

In this chapter, we introduce some background necessary to understand the contributions of this thesis. First, in Section 2.1, we investigate concentration inequalities. Random variables are assigned values randomly drawn from their distributions. Although these values are random and there is a fluctuation, their behaviour can be understood for specific distributions. More specifically, a concentration inequality is about bounding the distance of these random variables from their expectation by a certain amount Afterwards, in Section 2.2, we introduce the main ideas of online learning and present multi-armed bandits, pure exploration setting and reinforcement learning.

2.1 Concentration Inequalities

In this section, we are going to probe the world of concentration inequalities. We aim to provide the reader with high level understanding of concentration inequalities and highlight some of the cases when they can be used. A more detailed and advanced investigation of the topic can be found at Boucheron, Lugosi, and Massart (2013). Specifically, we are going to provide high-probability bounds for:

- A random variable X;
- Sum of independent random variables X_i: $\sum_{i=1}^{n} X_i$;
- Subgaussian random variables.

2.1.1 Probability and Measure Theoretic Prerequisites

There exists a number of concepts in probability and measure theory that are required before going into concentration inequalities, many of which are borrowed from Lattimore and Szepesvári (2020).

Definition 1 (σ-algebra)**.** *Let us denote the outcome space by Ω and define a σ-algebra to be a set $\mathcal{F} \subset 2^{\Omega}$ which satisfies three properties: it includes Ω, it is closed under complement and is a countable union.*

The σ-algebra is used when defining the notions of measure, probability measure and probability space.

Definition 2 (Measure)**.** *A measure f is a function from a σ-algebra to $\mathbb{R}$.*

The elements of a σ-algebra $\mathcal{F}$ are called measurable sets and they are measured in a sense that f assigns values to them.

Definition 3 (Measurable space)**.** *The pair $(\Omega, \mathcal{F})$ of an outcome space and a σ-algebra is called a measurable space.*

Definition 4 (Probability measure). *A function $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ (where $\mathcal{F}$ is a σ-algebra) is a probability measure if*

- $\mathbb{P}(\Omega) = 1$,
- $\mathbb{P}(A) \geq 0$ *and* $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$ *for all* $A \in \mathcal{F}$,
- $\mathbb{P}(\cup_i A_i) = \sum_i \mathbb{P}(A_i)$ *for all countable collections of disjoint sets* $\{A_i\}_i$ *with* $A_i \in \mathcal{F}$ *for all* i.

Definition 5 (Probability space). *The triplet $(\Omega, \mathcal{F}, \mathbb{P})$ consisting of an outcome space, a σ-algebra and a probability measure is called a probability space.*

Definition 6 (Random Variable). *A random variable X is a function which maps the outcome space Ω to the set of real numbers $\mathbb{R}$.*

One important quantity in probability theory is the expectation of a random variable or its mean value.

Definition 7 (Expectation). *The expected value of X is defined as its integral with respect to $\mathbb{P}$ (assuming that this integral exists):*

$$\mathbb{E}[X] = \int_\Omega X(\omega) d\mathbb{P}(\omega). \tag{2.1}$$

A very useful inequality in probability is Boole's inequality, stated in the next lemma.

Lemma 1 (Boole's inequality (Union Bound)). *For a finite countable set of events $A_1, A_2, A_3, \ldots, A_n$, we have:*

$$\mathbb{P}(\cup_i^n A_i) \leq \sum_i^n \mathbb{P}(A_i).$$

Sub-Gaussian random variables are commonly used and we are going to refer to them in both Chapter 3 and 4. So, here is the definition:

Definition 8 (σ-sub-Gaussian). *A random variable $X \in \mathbb{R}$ is said to be sub-Gaussian with variance proxy σ^2 if $\mathbb{E}[X] = 0$ and its moment generating function $\mathbb{E}[\exp(\lambda X)]$ satisfies*

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\sigma^2 \lambda^2}{2}\right), \forall \lambda \in \mathbb{R}.$$

Having introduced σ-sub-Gaussian random variables, the lemma below shows that any random variable that is bounded uniformly is actually sub-Gaussian with a variance that depends on the size of its support:

Lemma 2 (Hoeffding's lemma(1963)). *Let X be a random variable such that $\mathbb{E}(X) = 0$ and $X \in [a, b]$ almost surely. Then, for any $\lambda \in \mathbb{R}$, it holds:*

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\lambda^2 (b-a)^2}{8}\right).$$

Note that, as a consequence, any bounded random variable X in $[a, b]$ is $\frac{(b-a)}{2}$-sub-Gaussian.

We now introduce the notions of filtration, martingale and martingale difference sequence, which will be used to control the regret of the reinforcement learning algorithm C-UCRL we propose in Chapter 3.

Definition 9 (Filtration). *Given a measurable space $(\Omega, \mathcal{F})$ a filtration is a sequence $\mathbb{F} = (\mathcal{F}_t)_{t=1}^n$ of sub-σ-algebras of $\mathcal{F}$ where $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ for all $t < n$. For a sequence of random variables $X_1, \dots, X_n$ on $(\Omega, \mathcal{F}, \mathbb{P})$ and a filtration $\mathbb{F} = (\mathcal{F}_t)_{t=1}^n$, we say that the sequence $(X_t)_{t=1}^n$ is $\mathbb{F}$-adapted if X_t is $\mathcal{F}_t$-measurable for all $1 \leq t \leq n$.*

It is worth mentioning that t here can be seen as time. A martingale is then a sequence of random variables for which at a particular time, the conditional expectation of the next value in the sequence is equal to the present value.

Definition 10 (Martingale). *An $\mathbb{F}$-adapted sequence of random variables $(X_t)_{t\in\mathbb{N}_+}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is an $\mathbb{F}$-adapted martingale if:*

1. *X_t is integrable for all $t \in \mathbb{N}_+$,*
2. *$\mathbb{E}[X_t|\mathcal{F}_{t-1}] = X_{t-1}$ almost surely for all $t \in \{2, 3, \dots\}$.*

Definition 11 (Martingale Difference Sequence). *An $\mathbb{F}$-adapted sequence of random variables $(X_t)_{t\in\mathbb{N}_+}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a martingale difference sequence (MDS) if:*

1. *X_t is integrable for all $t \in \mathbb{N}_+$,*
2. *$\mathbb{E}[X_t|\mathcal{F}_{t-1}] = 0$ almost surely for all $t \in \{2, 3, \dots\}$.*

Note that if X_t is a martingale, then the sequence of random variables $(Y_t)_{t\in\mathbb{N}_+}$ defined as $Y_t = X_{t+1} - X_t$ is an MDS.

2.1.2 Guarantees on a Single Random Variable

We want to find out the probability δ that a random variable X is bounded by ε, i.e.:

$$\mathbb{P}(X \geq \varepsilon) \leq \delta. \tag{2.2}$$

Depending on the assumptions we make on the distribution of X, we can find different guarantees (Bertsekas and Tsitsiklis, 2002; Boucheron, Lugosi, and Bousquet, 2003). One of the most generic result is known as Markov inequality.

Theorem 1 (Markov inequality). *For any non-negative random variable X and any $\varepsilon > 0$, we have:*

$$\mathbb{P}(X \geq \varepsilon) \leq \frac{\mathbb{E}[X]}{\varepsilon}. \tag{2.3}$$

Proof. Since $X \geq 0$, we have:

$$\begin{aligned}\mathbb{E}[X] &= \int_0^\infty x\mathbb{P}(x)\, d_x = \int_0^\varepsilon x\mathbb{P}(x)\, d_x + \int_\varepsilon^\infty x\mathbb{P}(x)\, d_x, \\ &\geq \int_\varepsilon^\infty x\mathbb{P}(x)\, d_x \geq \varepsilon\mathbb{P}(X \geq \varepsilon).\end{aligned}$$

Therefore,

$$\mathbb{P}(X \geq \varepsilon) \leq \frac{\mathbb{E}[X]}{\varepsilon}.$$

□

Markov inequality can be extended to *non-negative non-decreasing* functions.

Theorem 2 (Extending Markov)**.** *Let φ be a non-negative and non-decreasing function of the non-negative reals. Then for any $\varepsilon \geq 0$ such that $\varphi(\varepsilon) > 0$ we have:*

$$\mathbb{P}\left(X \geq \varepsilon\right) \leq \mathbb{P}\left(\varphi(X) \geq \varphi(\varepsilon)\right) \leq \frac{\mathbb{E}[\varphi(X)]}{\varphi(\varepsilon)}.$$

Proof. Since φ is non-decreasing, for every $a < b$ on the domain we have $\varphi(a) < \varphi(b)$. Therefore, if $X \geq \varepsilon$ and φ is non-decreasing, then $\varphi(X) \geq \varphi(\varepsilon)$. Now, if $(X \geq \varepsilon) \subseteq (\varphi(X) \geq \varphi(\varepsilon))$, then $\mathbb{P}\left(X \geq \varepsilon\right) \leq \mathbb{P}\left(\varphi(X) \geq \varphi(\varepsilon)\right)$. Adding the non-negativity assumption, we can apply Markov inequality and the result is obtained. □

Let us see few applications of this extension: the Chernoff method and Chebyshev inequality.

Theorem 3 (Chernoff method)**.** *Let X be a real valued random variable, for any $\lambda > 0$:*

$$\mathbb{P}\left(X \geq \varepsilon\right) \leq \exp\left(-\lambda\varepsilon\right)\mathbb{E}[\exp\left(\lambda X\right)]. \tag{2.4}$$

The idea of Chernoff method is to write down an equivalent inequality replacing the random variable with the moment generating function of the random variable, which is obtained by choosing $\varphi(Y) = \exp\left(\lambda Y\right)$.

Theorem 4 (Chebyshev's inequality)**.** *For any random variable X, we have:*

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \varepsilon\right) \leq \frac{\mathrm{Var}(X)}{\varepsilon^2}. \tag{2.5}$$

Proof. This is another extension of Markov's inequality obtained by choosing $\varphi(t) = t^2$ and defining a random variable $Y = |X - \mathbb{E}[X]|$. Therefore,

$$\mathbb{P}\left(|X - \mathbb{E}[X]| \geq \varepsilon\right) = \mathbb{P}\left(Y \geq \varepsilon\right) \leq \mathbb{P}\left(Y^2 \geq \varepsilon^2\right) \leq \frac{\mathbb{E}[|X - \mathbb{E}[X]|^2]}{\varepsilon^2} = \frac{\mathrm{Var}(X)}{\varepsilon^2}. \tag{2.6}$$

□

2.1.3 Guarantees on a Sequence of Random Variables

Now, instead of a single random variable, we are going to consider a sequence of n random variables. A classic result is Hoeffding inequality, which provides a guarantee on the deviation of the sum of bounded independent random variables from its expected value without any assumption on the random variables' distribution.

Theorem 5 (Hoeffding inequality)**.** *Let $X_1, \ldots, X_n$ be independent bounded random variables with $X_i \in [a, b]$ for all i, where $-\infty < a < b < \infty$. Then for all $\varepsilon \geq 0$:*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i]) \geq \varepsilon\right) \leq \exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right). \tag{2.7}$$

Next, we introduce the Laplace method, where random variables are i.i.d but n is no longer a constant and is itself a random variable.

Lemma 3 (Time uniform concentration inequalities (Laplace method))**.** *Let $(X_t)_{t\in\mathbb{N}_+}$ be a sequence of i.i.d. real-valued random variables bounded in [0, 1], with mean μ. For all $\delta \in (0, 1)$, it holds*

$$\mathbb{P}\left(\exists n \in \mathbb{N}, \frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq \sqrt{(1 + \frac{1}{n})\frac{\ln(\sqrt{n+1}/\delta)}{2n}}\right) \leq \delta, \tag{2.8}$$

$$\mathbb{P}\left(\exists n \in \mathbb{N}, \mu - \frac{1}{n}\sum_{i=1}^{n} X_i \geq \sqrt{(1+\frac{1}{n})\frac{\ln(\sqrt{n+1}/\delta)}{2n}}\right) \leq \delta. \tag{2.9}$$

Lemma 4. *Let μ_a^t be the mean value of t independent real-valued random variables with the true mean μ_a and is σ-sub-gaussian. For all $\delta \in (0,1)$, it holds:*

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \mu_a^t - \mu_a \geq \sigma\sqrt{\frac{2}{t}(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta \ , \tag{2.10}$$

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \mu_a - \mu_a^t \geq \sigma\sqrt{\frac{2}{t}(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta \ . \tag{2.11}$$

Proof. The two inequalities are proved in the same way as a direct consequence of Maillard, 2019, Lemma 2.7 therein. Let $Y_1, \ldots, Y_t$ be a sequence of independent real-valued random variables where for each $s \leq t$, Y_s has mean μ_s and is σ_s-sub-Gaussian, then for all $\delta \in (0,1)$, it holds that

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \sum_{s=1}^{t}(Y_s - \mu_s) \geq \sqrt{2\sum_{s=1}^{t}\sigma_s^2(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta \ .$$

When all random variables Y_s have the same mean μ_a and variance σ, we have

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \sum_{s=1}^{t}(Y_s - \mu_a) \geq \sqrt{2t\sigma^2(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta,$$

Taking the average rather than the sum, i.e. dividing both sides by t we obtain that:

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \sum_{s=1}^{t}(\frac{Y_s}{t} - \frac{\mu_a}{t}) \geq \sqrt{\frac{2}{t}\sigma^2(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta,$$

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \sum_{s=1}^{t}\frac{Y_s}{t} - \mu_a \geq \sqrt{\frac{2}{t}\sigma^2(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta \ .$$

And denoting $\mu_a^t = \sum_{s=1}^{t}\frac{Y_s}{t}$, we conclude

$$\mathbb{P}\left(\exists t \in \mathbb{N}, \mu_a^t - \mu_a \geq \sigma\sqrt{\frac{2}{t}(1+\frac{1}{t})\ln(\sqrt{t+1}/\delta)}\right) \leq \delta \ .$$

□

2.1.4 Guarantees on Sub-Gaussian Random Variables

We can provide tighter bounds if we know about the properties of the distribution. Having introduced the subgaussianity assumption, here, we provide tighter bounds.

Intuitively, if X is distributed like a Gaussian with zero mean and variance σ^2, then X is σ-sub-Gaussian. Furthermore, if $\mathbb{E}[X] = 0$ and $|X| \leq B$ almost surely for some $B \geq 0$, then X is B-sub-Gaussian. The notion of σ-subgaussianity gives us good characteristics and helps us to better control the bounds over the random variables.

Theorem 6. *If X is a σ-sub-Gaussian, then for any $\varepsilon \geq 0$,*

$$\mathbb{P}\left(X \geq \varepsilon\right) \leq \exp\Big(-\frac{\varepsilon^2}{2\sigma^2}\Big). \tag{2.12}$$

Proof. For any constant $\lambda > 0$, we have

$$\begin{aligned}\mathbb{P}\left(X \geq \varepsilon\right) &= \mathbb{P}\left(\exp(\lambda X) \geq \exp(\lambda\varepsilon)\right) &(2.13)\\ &\leq \mathbb{E}[\exp\left(\lambda X\right)]\exp\left(-\lambda\varepsilon\right) &(2.14)\end{aligned}$$

by Markov inequality. Then, using the subgaussianity assumption:

$$\mathbb{P}\left(X \geq \varepsilon\right) \leq \exp\Big(\frac{\lambda^2\sigma^2}{2} - \lambda\varepsilon\Big). \tag{2.15}$$

Finally, to make the bound as tight as possible, we set $\lambda = \frac{\varepsilon}{\sigma^2}$. □

To go towards bounding sequences of sub-Gaussian random variables, the following lemma is going to be useful.

Lemma 5. *Suppose that X is σ-sub-Gaussian, and X_1 and X_2 are independent and σ_1 and σ_2-sub-Gaussian, then:*

- $\mathbb{E}[X] = 0$ *and* $\mathrm{Var}(X) \leq \sigma^2$.
- cX *is* $|c|\sigma$*-sub-Gaussian for all* $c \in \mathbb{R}$.
- $X_1 + X_2$ *is* $\sqrt{\sigma_1^2 + \sigma_2^2}$*−sub-Gaussian.*

Corollary 1. *Assume that $X_i - \mu$ are independent, σ-sub-Gaussian random variables. Then for any $\varepsilon \geq 0$,*

$$\mathbb{P}\left(\widehat{\mu} \geq \mu + \varepsilon\right) \leq \exp\Big(-\frac{n\varepsilon^2}{2\sigma^2}\Big), \tag{2.16}$$

and

$$\mathbb{P}\left(\widehat{\mu} \leq \mu - \varepsilon\right) \leq \exp\Big(-\frac{n\varepsilon^2}{2\sigma^2}\Big), \tag{2.17}$$

where $\widehat{\mu} = \frac{1}{n}\sum_{t=1}^{n} X_t$.

Proof. $X_i - \mu$ is σ-sub-Gaussian, so using Theorem 6 we have:

$$\mathbb{P}\left(X_i - \mu \geq \varepsilon\right) \leq \exp\Big(-\frac{\varepsilon^2}{2\sigma^2}.\Big)$$

Using Lemma 5, $\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu) = \widehat{\mu} - \mu$ is $\frac{1}{\sqrt{n}}\sigma$-sub-Gaussian and therefore:

$$\mathbb{P}\left(\widehat{\mu} - \mu \geq \varepsilon\right) \leq \exp\Big(-\frac{n\varepsilon^2}{2\sigma^2}\Big).$$

The second statement follows by symmetry. □

An alternative way of writing the above result shows that for any $\delta \in [0, 1]$,

$$\mathbb{P}\left(\mu \leq \widehat{\mu} + \sqrt{\frac{2\sigma^2\log(1/\delta)}{n}}\right) \geq 1 - \delta, \tag{2.18}$$

and similarly

$$\mathbb{P}\left(\mu \geq \widehat{\mu} - \sqrt{\frac{2\sigma^2 \log(1/\delta)}{n}}\right) \geq 1 - \delta. \tag{2.19}$$

2.2 Online Learning

Generally speaking, online learning is about learning from data that is received sequentially over time. It has attracted a lot of attention in modern days: since online learning algorithms process data one sample at a time, they provide efficient solutions for large-scale problems.

The contributions of this thesis are related to three types of online learning problems: multi-armed bandits, pure exploration bandits, and model-based reinforcement learning. We briefly review them below.

2.2.1 Multi-armed Bandits

A multi-armed bandit problem consists of an environment and a learner. In the environment, there exists k different arms each with a distribution ν_a and its corresponding mean μ_a. The learner has a set of k actions $\{1, 2, \ldots, k\}$, each of which corresponding to choosing one of the arms. At time t, the learner chooses an action (arm) a_t and receives a reward $r_t \sim \nu_{a_t}$. This goes on over a horizon of T rounds. At time t, the history H_t of agent is the set of actions and rewards $a_1, r_1, \ldots, a_{t-1}, r_{t-1}$ received until time $t-1$.

Definition 12. *A policy or learning algorithm $\mathbb{A}$ is a mapping from history H_t to action a_t.*

The objective of the learner is to find a policy that maximizes the cumulative reward $\sum_{t=1}^{T} r_t$ over all T rounds. One of the broadly-used evaluation measures is the regret.

Definition 13 (Regret)**.** *The regret $\mathfrak{R}(\mathbb{A}, T)$ of a learner with policy $\mathbb{A}$ is the difference between the total expected reward using the best policy and the total expected reward collected by the learner over T rounds:*

$$\mathfrak{R}(\mathbb{A}, T) = T \max_{a \in \mathcal{A}} \mu_a - \mathbb{E}[\sum_{t=1}^{T} r_t], \tag{2.20}$$

where the expectation is with respect to randomness of the environment and the policy.

There exists many different algorithms to solve a multi-armed bandit problem. We present here the classic Upper Confidence Bound algorithm (UCB) which is based on the UCRL2 (Auer, Cesa-Bianchi, and Fischer, 2002). It states that we should assume that the environment behaves as nicely as possible. This means that the algorithm should consider the best possible values of the environment parameters that are compatible with the observations so far. For an arm a and a time t, let us denote by $N_a^{(t-1)} = \sum_{1 \leq t' \leq t-1} \mathbb{1}_{\{a_{t'} = a\}}$ the number of observations for arm a before time t, and by

$$\widehat{\mu}_a^{(t-1)} = \frac{1}{N_a^{(t-1)}} \sum_{\substack{1 \leq t' \leq t-1 \\ a_{t'} = a}} r_a, \tag{2.21}$$

Algorithm 1 UCB(δ) algorithm (Lattimore and Szepesvári, 2020)

1: **Input** k and δ
2: Choose each action once
3: **while** $t > k$ **do**
4: Choose action $a_t = \text{argmax}_a \, \text{UCB}_a(t-1, \delta)$
5: $t = t + 1$
6: **end while**

the estimator of the mean μ_a obtained by averaging the $N_a^{(t-1)}$ rewards obtained so far. Using the concentration inequality (2.18), we can bound the deviation of $\widehat{\mu}_a^{(t-1)}$ from μ_a by:

$$\text{UCB}_a(t-1, \delta) = \widehat{\mu}_a^{(t-1)} + \sqrt{\frac{2\log(1/\delta)}{N_a^{(t-1)}}}. \tag{2.22}$$

As shown in Algorithm 1, the UCB policy consists in choosing the arm with the highest value for $\text{UCB}_a(t-1, \delta)$. Note that $\text{UCB}_a(t-1, \delta)$ is always an optimistic estimation for the estimator $\widehat{\mu}_a$ for all the arms because we estimate the mean as the upper bound of the confidence. Therefore either the optimal policy is chosen or the arms have not been explored enough. Assuming without loss of generality that the optimal policy is to choose arm 1 (i.e., $\mu_1 > \mu_a$ for all $a \neq 1$), if we had access to the true means of the arms, we would have clearly chosen arm 1. However, since we do not have access to the true values, we are estimating them using the upper confidence bound of each empirical mean of the arms. Therefore, the condition (for not optimal arms) on which UCB has found the optimal policy is:

$$\widehat{\mu}_a(t-1) + \sqrt{\frac{2\log(1/\delta)}{N_a(t-1)}} \leq \mu_1 \approx \widehat{\mu}_1(t-1) + \sqrt{\frac{2\log(1/\delta)}{N_1(t-1)}}, \quad \forall a \neq 1$$

So, when all optimistic estimations for suboptimal arms is less than that of the optimal arm, we have reached the optimal policy solution.

From another perspective, we can say an arm is chosen for either 1) it has a large $\widehat{\mu}_a^{(t-1)}$ or 2) it has not been explored a lot (small $N_a^{(t-1)}$) and thus it has a large confidence bound $\sqrt{\frac{2\log(1/\delta)}{N_a^{(t-1)}}}$.

The regret of UCB algorithm, setting $\delta = \frac{1}{T^2}$, is bounded by $\mathfrak{R}(\text{UCB}, T) \leq 8\sqrt{Tk\log(T)} + 3\sum_{i=1}^{k} \Delta_a$ where $\Delta_a = \mu_1 - \mu_a$ gives the gap between arm a and the optimal arm (Lattimore and Szepesvári, 2020).

2.2.2 Pure Exploration Bandits

In multi-armed bandit problems, we seek to maximize the cumulative reward by balancing exploration (of the arms that we have not explored a lot) and exploitation (of the arms that we have already chosen a lot and seem to provide large rewards). Let us now imagine a setting in which you do not have to pay for each exploration. For instance, consider that there exists k different foods and T units of budget. We do not want to save money and thus we do not care about using all the budget. So, we can try as many different foods as we want so that we find the very best food in the end. This represents the pure exploration setting.

Algorithm 2 `Uniform Exploration` algorithm (Lattimore and Szepesvári, 2020)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Choose $a_t = 1 + (t \mod k)$
3: **end for**
4: Choose $a_{T+1} = \text{argmax}_{a \in [k]} \widehat{\mu}_a(T)$

One way to evaluate a policy in a pure exploration setting is the *simple* regret:

$$\mathfrak{R}(\mathbb{A}, T)^{\text{SIMPLE}} = \mathbb{E}[\Delta_{a_{T+1}}], \tag{2.23}$$

where the expectation is with respect to the randomness of the environment and the policy, T is the exploration budget, $\mathbb{A}$ is the policy, $\Delta_a = \mu_1 - \mu_a$ (assuming as above that a_1 is the optimal arm) and a_{T+1}, the action chosen at time step $T+1$.

There exists a plethora of algorithms to solve the pure exploration setting. Algorithm 2 shows the simple uniform exploration policy (Bubeck, Munos, and Stoltz, 2009) which explores each arm for some time in a round-robin fashion. It then chooses the best arm accordingly to the empirical mean estimates $\widehat{\mu}_1(T), \ldots, \widehat{\mu}_k(T)$ as defined in (2.21). It can be shown that $\mathfrak{R}(\text{UE}, T)^{\text{SIMPLE}} \leq C\sqrt{\frac{k \log(k)}{T}}$ for $C > 0$. For more detailed explanations and to elaborate examples of pure exploration algorithms and their analysis, one can refer to Lattimore and Szepesvári (2020).

2.2.3 Reinforcement Learning (RL)

The multi-armed and pure exploration bandit problems described in the previous sections are particular instances of a general framework known as Reinforcement Learning (RL) (Sutton and Barto, 1998). RL considers a more complex notion of environment through the additional notion of *state*. At a high level, reinforcement learning problem is characterized by a set of states $s \in \mathcal{S}$, a set of actions $a \in \mathcal{A}$ and rewards given the state and action $r(s, a)$. A policy is then a mapping from the perceived state of environment to an action when being in a particular state. The goal is to maximize the cumulative reward along the trajectory or minimize regret compared to the optimal policy.

General Setting

More precisely, a reinforcement learning problem can be formalized by a Markov Decision Process (MDP).

Definition 14 (Markov Decision Process)**.** *An Markov Decision Process (MDP) is described by four components* $(\mathcal{S}, \mathcal{A}, p, \nu)$*:*

- $\mathcal{S}$ *is the set of possible states of the environment,*
- $\mathcal{A}$ *is the set of possible actions,*
- $p(s_{t+1}|s_t, a_t)$ *is the transition probability from a state* s_t *given an action* a_t*, to the next state* s_{t+1}*,*
- $\nu(s_t, a_t)$ *is the reward distribution function of the problem given state* s_t *and action* a_t*.*

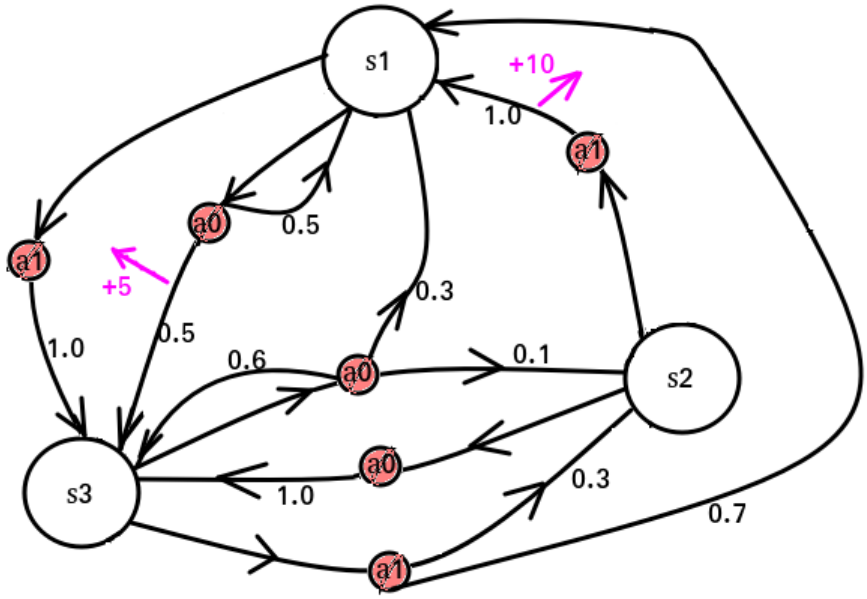


FIGURE 2.1: An example of Markov Decision Process.

MDPs allow to characterize reinforcement learning problems that satisfy the so-called *Markov Property*: their state signal compactly summarizes the past without degrading the ability to predict the future, i.e. $p(.|s_t, a_t) = p(.|s_t, a_t, \ldots, s_1, a_1)$ where a_i is an action and s_i is a state. An example MDP is presented in Figure 2.1, which has three states s_1, s_2, s_3 and two actions a_0, a_1. The transition probability from state s_1 with action a_0 to state s_3 is 0.5 and to state s_1 is 0.5. The reward of being in state s_2 and choosing action a_1 is +10. Another example of MDP is called RiverSwim MDP (Strehl and Littman, 2008). It consists of L states $\{s_1, s_2, \ldots, s_L\}$ and 2 actions {Right, Left}. The transition probability of going to state s_{i-1} from state s_i with action Left is higher than transiting to state s_{i+1} from state s_i with action Right. An illustration of transition probabilities and rewards of RiverSwim MDP can be seen in Figure 3.2.

Given a problem instance (MDP), a reinforcement learning algorithm considers an agent (learner) starting at some state $s_1 \in \mathcal{S}$ at time $t = 1$ and generally proceeds as follows. At each time step $t \in \mathbb{N}$, the agent chooses one action $a \in \mathcal{A}$ in its current state s_t based on its past decisions and observations. When executing action a_t in state s_t, the agent receives a random reward drawn independently from distribution $\nu(s_t, a_t)$, whose mean is $\mu(s_t, a_t)$. The state changes then to $s_{t+1} \sim p(\cdot|s_t, a_t)$, and a new decision step begins. The goal of the agent is to maximize the cumulative reward gathered during the course of interaction with the environment.

For a known MDP, the optimal (deterministic) policy can be found by the value iteration algorithm (Algorithm 3). The optimal policy is derived from the *value function* u, where $u(s)$ is equal to expected total reward for an agent starting from s and represents how good it is for an agent to be in state s. Value iteration computes (up to desired precision θ) the value function with a dynamic programming approach. Unfortunately, the MDP is usually unknown to the agent: in particular, the transition probabilities p and the reward distribution ν are unknown. Therefore, there is a need for exploration: the agent have to learn p and ν by trying different actions and recording the realized rewards and state transitions. We will see below how to balance exploitation (via value iteration) and exploration with the UCRL2 algorithm.

Algorithm 3 Value Iteration (up to desired precision $\theta > 0$)

Initialize u arbitrarily (e.g., $u(s) = 0$ for all $s \in \mathcal{S}^+$)
$\Delta = 0$
while $\Delta > \theta$ **do**
  $\Delta = 0$
  for each $s \in \mathcal{S}$ **do**
    $\nu \leftarrow u(s)$
    $u(s) \leftarrow \max_a \sum_{s'} p(s'|s,a)[\nu(s,a) + \gamma u(s')]$
    $\Delta \leftarrow \max(\Delta, |\nu - u(s)|)$
  end for
end while
Output a deterministic policy π, such that
$\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r|s,a)[r + \gamma u(s')]$

The performance of the learner can be assessed through the notion of regret with respect to an optimal oracle, being aware of p and ν and the actions a_t selected at different time steps, is chosen by the algorithm $\mathbb{A}$. More formally, as in Jaksch, Ortner, and Auer (2010) under a learning algorithm $\mathbb{A}$, we define the T-step regret of the network as the sum of all agents rewards (Kok and Vlassis, 2006):

$$\mathfrak{R}(\mathbb{A}, T) := T g_\star - \sum_{t=1}^{T} r_t(s_t, a_t)\,,$$

where $g_\star$ denotes the *average reward* (or *gain*[1]) attained by an optimal policy, and where a_t is chosen by $\mathbb{A}$ as a function of $((s_{t'}, a_{t'})_{t'<t}, s_t)$. Alternatively, the objective of the learner is to minimize the regret, which calls for balancing between exploration and exploitation.

What we described so far corresponds to the so-called *undiscounted* MDP setting, where we used average-reward criterion and regret to evaluate the algorithms. However, for the case of infinite horizon MDP, one typically considers the *discounted* MDP setting where the value function for a policy π is defined as (He, Zhou, and Gu, 2021; Liu and Su, 2020):

$$u_t^\pi(s) = \mathbb{E}\Big[\sum_{i=0}^{\infty} \gamma^i r(s_{t+i}, a_{t+i}) | s_t = s\Big],$$

and $0 < \gamma < 1$ is the discount factor. In the discounted setting, we value the immediate reward more than the more distant rewards. Having defined the value function, the regret is defined as:

$$\mathfrak{R}(\mathbb{A}, T) := \sum_{t=1}^{T} \big(u_t^\star(s_t) - u_t^{\mathbb{A}}(s_t)\big),$$

where $u_t^\star$ is the value function for the optimal policy and $u_t^{\mathbb{A}}$ is the value function corresponding to the agent's policy.

Model-based Reinforcement Learning

In this thesis, we will focus on model-based reinforcement learning, which refers to a family of RL algorithms in which we also estimate a model of the underlying MDP

[1] See, e.g., Puterman (2014) for background material on MDPs.

Algorithm 4 EVI$(\mu, p, N, \varepsilon, \delta)$ (Jaksch, Ortner, and Auer, 2010)

1: **Initialize:** $u^{(0)} = 0, u^{(-1)} = -\infty$, $n = 0$
2: **while** $\max_s(u^{(n)}(s) - u^{(n-1)}(s)) - \min_s(u^{(n)}(s) - u^{(n-1)}(s)) > \varepsilon$ **do**
3: For all (s,a), set $\mu'(s,a) = \mu(s,a) + \beta'_{N(s,a)}(\delta)$
4: For all (s,a), set $p'(\cdot|s,a) \in \text{argmax}_{q \in \mathcal{P}(s,a)} \sum_{x \in \mathcal{S}} q(x) u^{(n)}(x)$ where

$$\mathcal{P}(s,a) := \left\{ q \in \Delta^S : \|q - p(\cdot|s,a)\|_1 \leq \beta_{N(s,a)}(\delta) \right\}$$

5: For all s, update $u^{(n+1)}(s) = \max_{a \in \mathcal{A}} \left(\mu'(s,a) + \sum_{x \in \mathcal{S}} p'(x|s,a) u^{(n)}(x) \right)$
6: For all s, update $\pi_{n+1}(s) \in \text{argmax}_{a \in \mathcal{A}} \left(\mu'(s,a) + \sum_{x \in \mathcal{S}} p'(x|s,a) u^{(n)}(x) \right)$
7: Set $n = n + 1$
8: **end while**
9: **Output:** π_{n+1}

(i.e., the state-action transition probabilities and reward values of the problem).

We present here a popular model-based RL algorithm, UCRL2 (Jaksch, Ortner, and Auer, 2010). Like the UCB algorithm for multi-armed bandits introduced in Section 2.2.1, UCRL2 relies on the "optimism in face of uncertainty" principle. At a high level, UCRL2 maintains the set $\mathcal{M}_{t,\delta}$ of MDPs at time t [2] that is constructed by using the confidence bounds over transition probabilities p and reward μ models. It then implements the optimistic principle by trying to compute policy $\overline{\pi}_t^+ = \text{argmax}_{\pi:\mathcal{S}\to\mathcal{A}} \max_{M \in \mathcal{M}_{t,\delta}} g_\pi^M$, where g_π^M denotes the average reward or gain of policy π in MDP M.

The UCRL2 algorithm, at the start of each episode sets the time for the starting time of episode k: $t_k = t$. Then it counts the number of observations and sum of rewards for each state action pair up to time t_k. Afterwards it counts the number of observations for each state action and next state and using these, it calculates $\widehat{\mu}_{t_k}(s,a)$ and $\widehat{p}_{t_k}(.|s,a)$. Let $\mathcal{M}_{t,\delta}$ be set of all MDPs with transition probabilities $\widehat{p}_t(.|s,a)$ close to $p'_t(.|s,a)$ and reward function $\widehat{\mu}_t(s,a)$ close to $\mu'_t(s,a)$; that is,

$$\|\widehat{p}_t(\cdot|s,a) - p'(\cdot|s,a)\|_1 \leq \beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right), \ \forall s,a$$

$$|\widehat{\mu}_t(s,a) - \mu'(s,a)| \leq \beta'_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right), \ \forall s,a$$

where $\beta'_n(\delta)$ represents a confidence bound, S is the number of states and A is the number of actions. The confidence bound can be obtained from concentration inequalities introduced in Section 2.1.

UCRL2 relies on an extension of value iteration to find a near-optimal policy for the set of plausible MDPs $\mathcal{M}_{t,\delta}$, namely *Extended Value Iteration* (EVI) shown in Algorithm 4. EVI considers a set of plausible MDPs instead of one and chooses the one that is optimistically best (lines 3-4 of Algorithm 4), thereby implementing exploration in a way similar to UCB for multi-armed bandits (Section 2.2.1). EVI builds a near-optimal policy π_t^+ for the "optimistic" MDP $\widetilde{M}_t$ (with mean rewards $\widetilde{\mu}_t$ and transition probabilities $\widetilde{p}_t$ as constructed in lines 3-4) such that:

[2] This set is described by the Weismann confidence bounds combined with the Laplace method, see also Section 3.4. The original UCRL2 algorithm in (Jaksch, Ortner, and Auer, 2010) uses looser confidence bounds relying on union bounds instead of the Laplace method.

$$g_{\pi_t^+}^{\widetilde{M}_t} \geq \max_{\pi, M \in \mathcal{M}_{t,\delta}} g_\pi^M - \varepsilon.$$

At this step, we are going to explain an important property of the EVI algorithm which is used to bound the regret of the introduced algorithm later on in chapter 3:

Lemma 6. *At last iteration i of EVI (algorithm 4) with ε as input,*

$$\left|\left(g_{\pi_t^+}^{\widetilde{M}_t} - \widetilde{\mu}_t(s, \pi_t^+(s))\right) - \left(\sum_x \widetilde{p}_t(x|s, \pi_t^+(s)) u_t^{(i)}(x) - u_t^{(i)}(s)\right)\right| \leq \varepsilon, \qquad \forall s \in \mathcal{S}.$$

Proof. Using the same argument as in the proof of Jaksch, Ortner, and Auer, 2010, Theorem 2, the value function $u_t^{(i)}$ computed by EVI at the last iteration i satisfies: $\max_s u_t^{(i)}(s) - \min_s u_t^{(i)}(s) \leq D$. Moreover, the convergence criterion of EVI implies

$$|u_t^{(i+1)}(s) - u_t^{(i)}(s) - g_t| \leq \varepsilon, \qquad \forall s \in \mathcal{S}. \tag{2.24}$$

as stated in Jaksch, Ortner, and Auer (2010). By the design of `EVI` , we have $u_t^{(i+1)}(s) = \widetilde{\mu}_t(s, \pi_t^+(s)) + \sum_x \widetilde{p}_t(x|s, \pi_t^+(s)) u_t^{(i)}(x)$. Substituting this into (2.24) gives the result. □

The statement can be expressed in a matrix form defining $\mathbf{g}_t = g_{\pi_t^+}^{\widetilde{M}_t} \mathbf{1}$, where size of $\mathbf{1}$ is $|\mathcal{S}|$, $\widetilde{\boldsymbol{\mu}}_t = \left(\widetilde{\mu}_t(s, \pi_t^+(s))\right)_{s \in \mathcal{S}}$ and $\widetilde{\mathbf{P}}_t = \left(\widetilde{p}_t(x|s, \pi_t^+(s))\right)_{s,x \in \mathcal{S}}$, we can rewrite the above inequality as:

$$\left|\mathbf{g}_t - \widetilde{\boldsymbol{\mu}}_t - (\widetilde{\mathbf{P}}_t - \mathbf{I}) u_t^{(i)}\right| \leq \varepsilon \mathbf{1} \ ,$$

where size of $\mathbf{I}$ is $|\mathcal{S}| \times |\mathcal{S}|$ and therefore

$$\mathbf{g}_t - \widetilde{\boldsymbol{\mu}}_t - (\widetilde{\mathbf{P}}_t - \mathbf{I}) u_t^{(i)} \leq \varepsilon \mathbf{1} \ .$$

Finally, UCRL2 does not recompute π_t^+ at each time step. Instead, it proceeds in internal episodes (indexed by $k \in \mathbb{N}$), and computes π_t^+ only at the starting time t_k of each episode and with precision $\varepsilon = \frac{1}{\sqrt{t_k}}$, where $t_1 = 1$ and for all $k > 1$, $t_k = \min\left\{t > t_{k-1} : \exists s, a, V_{t_{k-1}:t}(s,a) \geq N_{t_{k-1}}(s,a)\right\}$, where $V_{t_1:t_2}(s,a)$ denotes the number of observations of pair (s,a) between time t_1+1 and t_2. The pseudo-code for UCRL2 is shown in Algorithm 5.

The regret of UCRL2 depends on the notion of diameter of an MDP which is used in chapter 3.

Definition 15 (Diameter)**.** *Let us consider the stochastic process built using a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ performing on an MDP M with initial state s. Let $T(s'|M, \pi, s)$ be the random variable showing the first time state s' is reached in this stochastic process. The diameter of M is defined as:*

$$D(M) := \max_{s \neq s' \in \mathcal{S}} \min_{\pi : \mathcal{S} \to \mathcal{A}} \mathbb{E}[T(s'|M, \pi, s)].$$

Theorem 7. *With probability of at least $1 - \delta$ it holds that for any initial state $s \in \mathcal{S}$ and any $T > 1$, the regret of UCRL2 is bounded by*

$$\mathfrak{R}(M, \text{UCRL2}, s, T) \leq 34 D(M) S \sqrt{AT \log\left(\frac{T}{\delta}\right)}.$$

Algorithm 5 UCRL2(δ) with input parameter $\delta \in (0, 1]$ (Jaksch, Ortner, and Auer, 2010)

Initialize: For all (s, a), set $N_0(s, a) = 0$ and $V_0(s, a) = 0$. Set $t_0 = 0$, $t = 1$, $k = 1$, and observe the initial state s_1
for episodes $k \geq 1$ **do**
  Set $t_k = t$
  Set $N_{t_k}(s, a) = N_{t_{k-1}}(s, a) + V_k(s, a)$ for all (s, a)
  Compute empirical estimates $\widehat{\mu}_{t_k}(s, a)$ and $\widehat{p}_{t_k}(\cdot|s, a)$ for all (s, a)
  Compute $\pi^+_{t_k} = \texttt{EVI}\left(\widehat{\mu}_{t_k}, \widehat{p}_{t_k}, N_{t_k}, \frac{1}{\sqrt{t_k}}, \frac{\delta}{SA}\right)$ — see Algorithm 4
  while $V_k(s_t, \pi^+_{t_k}(s_t)) < \max\{1, N_{t_k}(s_t, \pi^+_{t_k}(s_t))\}$ **do**
    Play action $a_t = \pi^+_{t_k}(s_t)$, observe the next state s_{t+1} and reward $r_t(s_t, a_t)$
    Set $V_k(s_t, a_t) = V_k(s_t, a_t) + 1$
    Set $t = t + 1$
  end while
end for

where S is the number of states and A represents the number of actions.

In the next chapter, we will introduce a notion of problem structure in MDPs and we will rely on this to improve the theoretical guarantees of model-based RL algorithms such as UCRL2.

Chapter 3

Model-based Reinforcement Learning Exploiting State-Action Equivalence

This chapter presents the first contribution of this thesis, where we propose an approach to identify some structure in state-action pairs of reinforcement learning problems and show how to leverage this structure to reduce the regret (Asadi et al., 2019).

3.1 Introduction

As explained in Section 2.2.3, in Reinforcement Learning (RL) problems, agents interact with an unknown environment in a single stream of observations, with the aim of maximizing the cumulative reward gathered over the course of experience. The environment is typically modeled as a Markov Decision Process (MDP, see Definition 14), with finite state and action spaces. In order to act optimally (or nearly so), the agents needs to learn the parameters of the MDP using the observations from the environment. The agents thus faces a fundamental trade-off between *exploitation vs. exploration*: namely, whether to gather more experimental data about the consequences of the actions (exploration) or acting consistently with past observations to maximize the rewards (exploitation); see Sutton and Barto, 1998. Over the past two decades, a plethora of studies have addressed the above RL problem, either in the undiscounted setting where the goal is to minimize the regret (e.g., Bartlett and Tewari, 2009; Jaksch, Ortner, and Auer, 2010; Gheshlaghi Azar, Osband, and Munos, 2017), or in the discounted setting (as in e.g., Strehl and Littman, 2008) with the goal of bounding the sample complexity of exploration as defined by Kakade (2003). In most practical situations however, the state-space of the underlying MDP is too large: directly applying the state-of-the-art RL algorithms, for instance from the above works, would lead to a prohibitive regret or sample complexity.

The work of this chapter is motivated by the observation that the underlying MDP in RL problems is often endowed with some *structure* that could be exploited to learn more effectively. Specifically, we consider RL problems where the state-action space of the MDP exhibits some *equivalence structure*. This is quite typical of many MDPs in various application domains. Consider for instance grid-world MDPs. The action-space is $\{u, d, l, r\}$. Playing action $a = u$ moves the current state *up* with probability 0.8, does not change the current state with probability 0.1, and moves left or right with the same probability 0.05. Walls act as *reflectors*: when the next state is a wall, the transition probability of it is added to that of the current state. Other actions are defined in a similar way. Finally, the goal-state is put in the bottom-right corner of the MDP, where the learner is given a reward of 1. In such grid-world

MDP, taking action *up* from state s or *right* from state s' when both are away from any wall may result in similar transitions (typically, move towards the target state with some probability, and stay still or transit to other neighboring states with the remaining probability); see Figure 3.1 in Section 3.3. We are interested in identifying and exploiting such structure in order to speed up the learning process. We do so by aggregating the information of state-action pairs in the same equivalence class when estimating the transition probabilities or reward function of the MDP.

We note that leveraging an equivalence structure is popular in the MDP literature; see Ravindran and Barto (2004), Li, Walsh, and Littman (2006), and Abel, Hershkowitz, and Littman (2016). However, most notions are unfortunately not well adapted to the RL setup, that is when the underlying MDP is *unknown*. In particular, amongst those considering such structures, to our knowledge, none has provided performance guarantees in terms of regret or sample complexity (see below for a brief review). In contrast, our goal is to find a near-optimal policy, with controlled regret or sample complexity. To this end, we follow a model-based approach, which is popular in the RL literature (see Section 2.2.3), and aim at providing a generic approach capable of exploiting this structure to reduce regret or sample complexity.

3.2 Related work

There is a rich literature on state-abstraction (or state-aggregation) in MDPs; we refer to Li, Walsh, and Littman (2006) on earlier methods, and to Abel, Hershkowitz, and Littman (2016) for a good survey of recent approaches. Ravindran and Barto (2004) introduces aggregation based on homomorphisms of the model, but with no algorithm nor regret analysis. Dean, Givan, and Leach (1997) and Givan, Dean, and Greig (2003) consider a partition of state-space of MDPs based on the notion of *stochastic bi-simulation*, which is a generalization of the notion of bi-simulation from the theory of concurrent processes to stochastic processes. This path is further followed in Ferns, Panangaden, and Precup (2004) and Ferns, Panangaden, and Precup (2011), where *bi-simulation metrics* for capturing similarities are presented. Bi-simulation metrics can be thought of as quantitative analogues of the equivalence relations, and suggest to resort to optimal transport, which is intimately linked with our notions of similarity and equivalence (see Definition 16). However, these powerful metrics have only been studied in the context of a *known* MDP, and not the RL setup. The approach in Anand et al. (2015) is similar to our work in that it considers state-action equivalence. Unlike the present paper, however, it does not consider orderings, transition estimation errors, or regret analysis. Another relevant work to our approach is Ortner (2013) on aggregation of states (but not of pairs, and with no ordering) based on concentration inequalities, a path that we follow. We also mention the work of Brunskill and Li (2013) and Mandel et al. (2016), where clustering of the state-space is studied. As other relevant works, we refer to Leffler, Littman, and Edmunds (2007), where *relocatable action model* is introduced, and to Diuk, Li, and Leffler (2009) that studies RL in the simpler setting of factored MDPs. We also mention interesting works revolving around complementary RL questions including the one on selection amongst different state representations in Ortner, Maillard, and Ryabko (2014) and on state-aliasing in Hallak, Di-Castro, and Mannor (2013).

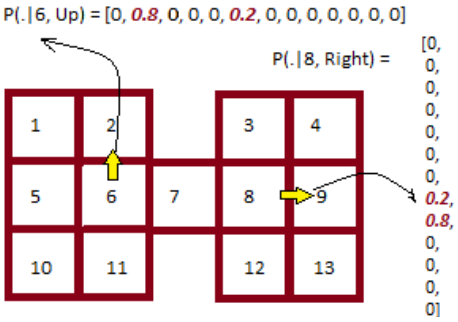


FIGURE 3.1: A grid-world MDP showing similar transitions from pairs $(6, \mathsf{Up})$ and $(8, \mathsf{Right})$.

3.3 Similarity and Equivalence classes

We now present a precise definition of the equivalence structure considered in this work. Let us consider an MDP $M = (\mathcal{S}, \mathcal{A}, p, \nu)$ where $\mathcal{S} = \{1, \dots, S\}$ (see Definition 14) and let $\mu(s, a)$ be the mean of the rewards when action $a \in \mathcal{A}$ is performed from state $s \in \mathcal{S}$. We first introduce a notion of similarity between state-action pairs. It is based on the probability distribution p of reaching a next state given a current state and an action. However, building an equivalence relation by comparing p directly can be too fine-grained and will miss some symmetries or identities that could be exploited when learning the MDP. For instance, consider the grid-like environment depicted in Figure 3.1, an MDP called 2-rooms MDP with 13 states and 4 actions {Up, Down, Left, Right}. Going up in state 6 and going right in state 8 has the same probability of success (arriving respectively in state 2 and state 9). We propose to capture this kind of structure by considering permutations on the set of states. Let $\sigma_{s,a} : \mathcal{S} \to \mathcal{S}$ be a permutation of states such that

$$p(\sigma_{s,a}(1)|s, a) \geq p(\sigma_{s,a}(2)|s, a) \geq \cdots \geq p(\sigma_{s,a}(S)|s, a).$$

We refer to $\sigma_{s,a}$ as a *profile mapping* (or for short, *profile*) for (s, a), and denote by $\boldsymbol{\sigma} = (\sigma_{s,a})_{s,a}$ the set of profile mappings of all pairs. Then the similarity in the MDP M is defined in the following way.

Definition 16 (Similar state-action pairs)**.** *The pair (s', a') is said to be* ε-similar *to the pair (s, a), for $\varepsilon = (\varepsilon_p, \varepsilon_\mu) \in \mathbb{R}_+^2$, if*

$$\|p(\sigma_{s,a}(\cdot)|s, a) - p(\sigma_{s',a'}(\cdot)|s', a')\|_1 \leq \varepsilon_p \quad \text{and} \quad |\mu(s, a) - \mu(s', a')| \leq \varepsilon_\mu .$$

Clearly, $(0, 0)$-similarity defines an equivalence relation on the state-action space $\mathcal{S} \times \mathcal{A}$.

Definition 17 (Equivalence classes)**.** *The $(0, 0)$-similarity defines the* equivalence structure *of $\mathcal{S} \times \mathcal{A}$. We denote it by $\mathcal{C}$, and let C to be the number of equivalence classes of $\mathcal{C}$.*

Definitions 16 and 17 are illustrated in Figure 3.1. The state-action pairs $(6, \mathsf{Up})$ and $(8, \mathsf{Right})$ are equivalent up to a permutation: let the permutation $\sigma_{(6,\mathsf{Up})}$ be such that $\sigma_{(6,\mathsf{Up})}(2) = 9$, $\sigma_{(6,\mathsf{Up})}(6) = 8$, and $\sigma(i)_{(6,\mathsf{Up})} = i$ for all $i \notin \{2, 6\}$. Now $p(\sigma(x)|6, \mathsf{Up}) = p(x|8, \mathsf{Right})$ for all $x \in \mathcal{S}$, and thus, the pairs $(8, \mathsf{Right})$ and $(6, \mathsf{Up})$ belong to the same equivalence class.

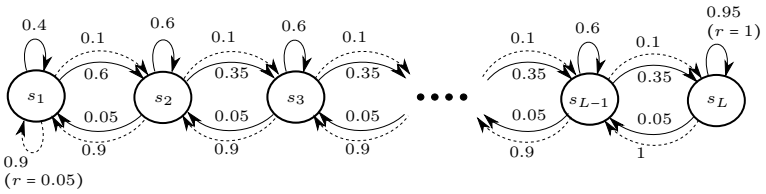


FIGURE 3.2: The L-state *Ergodic RiverSwim* MDP.

Remark 1. *Crucially, the equivalence relation is not only stated about states, but about* state-action pairs. *For instance, pairs* $(6, \mathsf{Up})$ *and* $(8, \mathsf{Right})$ *in this example are in the same class although they correspond to playing different actions in different states.*

Remark 2. *The profile mapping* $\sigma_{s,a}$ *in Definition 16 may not be unique in general, especially if distributions have sparse supports. For ease of presentation, in the sequel we assume that the restriction of* $\sigma_{s,a}$ *to the support of* $p(\cdot|s,a)$ *is uniquely defined. We also remark that Definition 16 can be easily generalized by replacing the* $\|\cdot\|_1$ *norm with another notion of distance or divergence, such as the KL divergence, squared Euclidean distance, etc.*

In many environments considered in RL with large state and action spaces, the number C of equivalent classes of state-action pairs using Definitions 16–17 stays small even when $S \times A$ is large, thanks to the profile mappings. This is the case in typical grid-world MDPs such as that in Figure 3.1 as well as in classic *RiverSwim* problems (Strehl and Littman, 2008). For example, in *Ergodic RiverSwim* with L states (Figure 3.2),[1] we have $C = 6$. This remarkable feature suggests that leveraging this structure may yield significant speed-up in terms of learning guarantees, if exploited well.

We stress that other notions of similarity from the RL literature do not scale well. For instance, in Ortner (2013), a partition $\mathcal{S}_1, \dots \mathcal{S}_n$ of the state-space $\mathcal{S}$ is considered to define an aggregated MDP, which satisfies, for all $i \in \{1, \dots, n\}$,

$$\begin{aligned} \forall s, s' \in \mathcal{S}_i, \forall a \in \mathcal{A}, \qquad & \mu(s,a) = \mu(s',a)\,, \\ \forall j \in \{1, \dots, n\}, \qquad & \textstyle\sum_{s'' \in \mathcal{S}_j} p(s''|s,a) = \sum_{s'' \in \mathcal{S}_j} p(s''|s',a)\,. \end{aligned}$$

This readily prevents any two states s, s' such that $p(\cdot|s,a)$ and $p(\cdot|s',a)$ have disjoint supports from being in the same set $\mathcal{S}_i$. Thus, since in a grid-world MDP, where transitions are local, the number of pairs with disjoint support is (almost linearly) increasing with S, this implies a potentially large number of classes for grid-worlds with many states. A similar criticism can be formulated for the approach of Anand et al. (2015), even though it considers sets of state-action pairs instead of states only, thus slightly reducing the total number of classes.

3.4 Equivalence-Aware Confidence Sets

We are now ready to present an approach that defines confidence sets for p and μ taking into account the equivalence structure in the MDP. Recall that the MDP is unknown and we estimate the true one by iterative updates to the model. The use of confidence bounds in a model-based approach is related to strategies implementing the *optimism in the face of uncertainty* principle. As explained in Section 2.2.3,

[1] An MDP is ergodic if any state is reachable from any other state by following any policy.

these approaches rely on maintaining a set of plausible MDPs (models) that are consistent with the observations gathered so far and contains the true MDP M with high probability. The plausibility is represented by confidence intervals on the model probability distribution $p(.|s,a)$ and the mean of ν. Any algorithm of this kind takes one action at each time step and updates the confidence bounds (and thus the set of plausible MDPs) at some point in time based on the available observations. Exploiting the equivalence structure of the MDP could then allow to obtain a more precise estimation of mean reward μ and transition probabilities p of the MDP by *aggregating* observations from various state-action pairs in the same class. This, in turn, would yield *smaller* (hence, better) sets of models, and thus a smaller time needed to reach an ε-optimal solution.

Notations. We introduce some necessary notations (some of which were already used in Chapter 2). Under a given RL algorithm, for a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, we denote by $N_t(s,a)$ the total number of observations of (s,a) up to time t. Let us define $\widehat{\mu}_t(s,a)$ as the empirical mean reward built using $N_t(s,a)$ i.i.d. samples from $\nu(s,a)$, and $\widehat{p}_t(\cdot|s,a)$ as the empirical distribution built using $N_t(s,a)$ i.i.d. observations from $p(\cdot|s,a)$. For a set $c \subseteq \mathcal{S} \times \mathcal{A}$, we denote by $N_t(c)$ the total number of observations of state-action pairs in c up to time t, that is $N_t(c) = \sum_{(s,a)\in c} N_t(s,a)$. We further denote by $\widehat{\mu}_t(c)$ and $\widehat{p}_t(\cdot|c)$ the empirical mean reward and transition probability built using these $N_t(c)$ samples, respectively; we provide precise definitions of $\widehat{\mu}_t(c)$ and $\widehat{p}_t(\cdot|c)$ later on in this section where they are needed.

For a given confidence parameter δ and time t, we write $\mathsf{CB}_{t,\delta}$ (resp. $\mathsf{CB}'_{t,\delta}$) to denote the confidence set for the true transition probability function p (resp. the true reward function μ) centered at $\widehat{p}_t$ (resp. $\widehat{\mu}_t$). The definition of the confidence sets are such that p (resp. μ) belongs to $\mathsf{CB}_{t,\delta}$ (resp. $\mathsf{CB}'_{t,\delta}$) for all t with high probability $1-\delta$. In other words:

$$\mathbb{P}\left(\bigcap_t p \in \mathsf{CB}_{t,\delta} \wedge \mu \in \mathsf{CB}'_{t,\delta}\right) \geq 1 - 2\delta\,. \tag{3.1}$$

Note that both $\mathsf{CB}_{t,\delta}$ and $\mathsf{CB}'_{t,\delta}$ depend on $N_t(s,a), (s,a) \in \mathcal{S} \times \mathcal{A}$. We consider the following confidence sets[2] used in several model-based RL algorithms, e.g., Jaksch, Ortner, and Auer (2010) and Dann, Lattimore, and Brunskill (2017):

$$\begin{aligned}
\mathsf{CB}_{t,\delta} &= \left\{p' : \|\widehat{p}_t(\cdot|s,a) - p'(\cdot|s,a)\|_1 \leq \beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\ \forall s,a\right\}, \\
\mathsf{CB}'_{t,\delta} &= \left\{\mu' : |\widehat{\mu}_t(s,a) - \mu'(s,a)| \leq \beta'_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\ \forall s,a\right\},
\end{aligned} \tag{3.2}$$

where

$$\beta_n(\delta) = \sqrt{\frac{2(1+\frac{1}{n})\log\left(\sqrt{n+1}\frac{2^S-2}{\delta}\right)}{n}} \quad \text{and} \quad \beta'_n(\delta) = \sqrt{\frac{(1+\frac{1}{n})\log(\sqrt{n+1}/\delta)}{2n}}\,,\ \forall n. \tag{3.3}$$

The above confidence sets are derived by combining Hoeffding's (Hoeffding, 1963) and Weissman's (Weissman et al., 2003) concentration inequalities with the Laplace method (Peña, Lai, and Shao, 2008; Abbasi-Yadkori, Pál, and Szepesvári, 2011) discussed in Section 2.1.3, Lemma 3. It enables to handle the random stopping times,

[2] Our approach can be extended to other concentration inequalities as well.

and to control the confidence sets with a tight bound We refer to Maillard (2019) for further discussion on the Laplace method and random stopping times.

We can now write $\mathcal{M}_{t,\delta}$ to denote the set of plausible MDPs at time t, which may be generically expressed as

$$\mathcal{M}_{t,\delta} = \{(\mathcal{S}, \mathcal{A}, p', \nu') : p' \in \mathsf{CB}_{t,\delta} \text{ and } \mu' \in \mathsf{CB}'_{t,\delta}\}, \tag{3.4}$$

and satisfies that the true MDP M belongs to $\mathcal{M}_{t,\delta}$ for all t with probability larger than $1 - 2\delta$.

Remark 3. *As the bounds for μ and p are similar, techniques to control μ will be similar to the one used for p and the class equivalence only introduces a simple conjunction. To simplify the presentation, in the next sections, we assume the mean reward function μ is known.*[3]

We now provide modifications to $\mathsf{CB}_{t,\delta}$ in order to exploit the equivalence structure $\mathcal{C}$, when the learner knows $\mathcal{C}$ in advance. The case of an unknown $\mathcal{C}$ is addressed in Section 3.5.

3.4.1 Case 1: Known Classes and Profiles

Assume that an oracle provides the learner with a perfect knowledge of the equivalence classes $\mathcal{C}$ as well as profiles $\boldsymbol{\sigma} = (\sigma_{s,a})_{s,a}$. In this ideal situation, the knowledge of $\mathcal{C}$ and $\boldsymbol{\sigma}$ allows to straightforwardly aggregate observations from all agents in the same class to build more accurate estimates of p and μ. Recall that the permutations $\sigma_{s,a}$ sort transition probabilities such that

$$p(\sigma_{s,a}(1)|s,a) \geq p(\sigma_{s,a}(2)|s,a) \geq \cdots \geq p(\sigma_{s,a}(S)|s,a).$$

Formally, for a class $c \in \mathcal{C}$, we define the transition probabilities given the class c as:

$$\widehat{p}_t^{\boldsymbol{\sigma}}(x|c) = \frac{1}{N_t(c)} \sum_{(s,a)\in c} N_t(s,a)\widehat{p}_t\big(\sigma_{s,a}(x)|s,a\big), \qquad \forall x \in \mathcal{S}, \tag{3.5}$$

where we recall that $N_t(c) = \sum_{(s,a)\in c} N_t(s,a)$. The superscript $\boldsymbol{\sigma}$ in (3.5) indicates that the aggregate empirical distribution $\widehat{p}_t^{\boldsymbol{\sigma}}$ depends on $\boldsymbol{\sigma}$. Note that $\widehat{p}_t^{\boldsymbol{\sigma}}$ is such that $\widehat{p}_t^{\boldsymbol{\sigma}}(1|c)$ is the weighted combination for all $(s,a) \in c$ of all largest probabilities given (s,a). We change the confidence set (3.2) by modifying the ℓ_1 bound there as follows:

$$\|\widehat{p}_t^{\boldsymbol{\sigma}}(\cdot|c) - p'(\sigma_{s,a}(\cdot)|s,a)\|_1 \leq \beta_{N_t(c)}\big(\tfrac{\delta}{C}\big), \quad \forall c \in \mathcal{C}, \forall (s,a) \in c. \tag{3.6}$$

We further define $\mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) = \big\{p' : (3.6) \text{ holds}\big\}$, where the dependence on $\mathcal{C}$ and $\boldsymbol{\sigma}$ stresses the fact that they are provided as input. By construction, we now have that with probability $1 - \delta$, for all time t the true transition p belongs to $\mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma})$:

$$\mathbb{P}\left(\bigcap_t p \in \mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma})\right) \geq 1 - \delta\,. \tag{3.7}$$

Remark 4. *It is crucial to remark that the above confidence set does not use elements of $\mathcal{C}$ as "meta states" (i.e., replacing the states with classes), as considered for instance in the literature on state-aggregation. Instead, the classes are only used to* group *observations from different sources and build more refined estimates for each pair. In*

[3] This is a common assumption in the RL literature; see, e.g., Bartlett and Tewari (2009).

particular, the plausible MDPs are built using the original state-space $\mathcal{S}$ and action-space $\mathcal{A}$, unlike in, e.g., Ortner (2013).

3.4.2 Case 2: Known Classes, Unknown Profiles

Now we consider a more realistic setting when the oracle provides $\mathcal{C}$ to the learner, but $\boldsymbol{\sigma}$ is unknown. In this more challenging situation, we need to *estimate* profiles as well. Given time t, we find an *empirical profile mapping* (or for short, empirical profile) $\sigma_{s,a,t}$ satisfying

$$\widehat{p}_t(\sigma_{s,a,t}(1)|s,a) \geq \widehat{p}_t(\sigma_{s,a,t}(2)|s,a) \geq \cdots \geq \widehat{p}_t(\sigma_{s,a,t}(S)|s,a)\,,$$

and define $\boldsymbol{\sigma}_t = (\sigma_{s,a,t})_{s,a}$. We then build the modified empirical estimate in a similar fashion to (3.5): for any $c \in \mathcal{C}$,

$$\widehat{p}_t^{\boldsymbol{\sigma}_t}(x|c) = \frac{1}{N_t(c)} \sum_{(s,a)\in c} N_t(s,a)\widehat{p}_t(\sigma_{s,a,t}(x)|s,a)\,, \quad \forall x \in \mathcal{S}. \tag{3.8}$$

Then, we can modify the ℓ_1 bound in (3.2) as follows:

$$\|\widehat{p}_t^{\boldsymbol{\sigma}_t}(\cdot|c) - p'(\sigma_{s,a}(\cdot)|s,a)\|_1 \leq \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a')\beta_{N_t(s',a')}\left(\tfrac{\delta}{C}\right), \quad \forall c \in \mathcal{C}, \forall (s,a) \in c, \tag{3.9}$$

This yields the modified confidence set $\mathsf{CB}_{t,\delta}(\mathcal{C}) = \left\{p' : (3.9) \text{ holds}\right\}$ that uses only $\mathcal{C}$ as input. To show the validity of this confidence set, we will need the following lemma.

Lemma 7 (Non-expansive ordering)**.** *Let p and q be two discrete distributions, defined on the same alphabet $\mathcal{S}$, with respective profile mappings σ_p and σ_q. Then,*

$$\|p(\sigma_p(\cdot)) - q(\sigma_q(\cdot))\|_1 \leq \|p - q\|_1\,.$$

Proof. Obviously we have $\|p-q\|_1 = \|p(\sigma_p(\cdot)) - q(\sigma_p(\cdot))\|_1$. Therefore the lemma holds if $\|p(\sigma_p(\cdot)) - q(\sigma_q(\cdot))\|_1 \leq \|p(\sigma_p(\cdot)) - q(\sigma_p(\cdot))\|_1$.

The proof is based on the fact that one can obtain σ_p from σ_q by a finite sequence of elementary permutations $\sigma_q = \sigma_0, \sigma_1, \ldots, \sigma_k = \sigma_p$ such that for all $i \in \{0, \ldots, k-1\}$, $\|p(\sigma_{i+1}(\cdot)) - q(\sigma_i(\cdot))\|_1 \leq \|p(\sigma_{i+1}(\cdot)) - q(\sigma_{i+1}(\cdot))\|_1$.

Indeed, elementary permutations only exchange 2 indices as in many sort algorithms. The sequence is defined such that for all $i \in \{0, \ldots, k-1\}$, there exists 2 distinct states s_1 and s_2 such that $\sigma_i(s_1) < \sigma_i(s_2)$, $\sigma_i(s_1) = \sigma_{i+1}(s_2)$, $\sigma_i(s_2) = \sigma_{i+1}(s_1)$ and $\sigma_i(s) = \sigma_{i+1}(s)$ for every $s \in \mathcal{S} \setminus \{s_1, s_2\}$. Note that $p(\sigma_i(s_1)) > p(\sigma_i(s_2))$ (and therefore $p(\sigma_{i+1}(s_1)) \leq p(\sigma_{i+1}(s_2))$ and $q(\sigma_i(s_1)) \leq q(\sigma_i(s_2))$.

Therefore it suffices to show that for all $i \in \{0, \ldots, k-1\}$,

$$\sum_{x=1,2} |p(\sigma_{i+1}(s_x)) - q(\sigma_i(s_x))| \leq \sum_{x=1,2} |p(\sigma_{i+1}(s_x)) - q(\sigma_{i+1}(s_x))|. \tag{3.10}$$

Let us denote $a = p(\sigma_{i+1}(s_1))$ and $b = p(\sigma_{i+1}(s_2))$ and $c = q(\sigma_{i+1}(s_1))$, $d = q(\sigma_{i+1}(s_2))$. Then (3.10) rewrites

$$|a-d| + |b-c| \leq |a-c| + |b-d|, \tag{3.11}$$

where $a < b$ and $d < c$. The proof then examines the 6 cases of all possible total orderings of the 4 values.

- if $a < b < d < c$ then $|a-c| + |b-d| = |a-d| + |d-c| + |b-d| = |a-d| + |b-c|$,

- if $a < d < b < c$ then $|a-c|+|b-d| = |a-d|+|b-c|+2|b-d| \geq |a-d|+|b-c|$,
- if $d < a < b < c$ then $|a-c|+|b-d| = |a-d|+|b-c|+2|a-b| \geq |a-d|+|b-c|$,
- if $a < d < c < b$ then $|a-c|+|b-d| = |a-d|+|b-c|+2|d-c| \geq |a-d|+|b-c|$,
- if $d < a < c < b$ then $|a-c|+|b-d| = |a-d|+|b-c|+2|a-c| \geq |a-d|+|b-c|$,
- if $d < c < a < b$ then $|a-c|+|b-d| = |a-c|+|b-a|+|a-d| = |a-d|+|b-c|$,

and therefore all cases satisfy (3.11) which concludes the proof. □

Corollary 2. *The modified confidence set* $\mathsf{CB}_{t,\delta}(\mathcal{C})$ *contains the true transition function* p *with probability at least* $1-\delta$*, uniformly over all time* t*, i.e.:*

$$\mathbb{P}\left(\bigcap_t p \in \mathsf{CB}_{t,\delta}(\mathcal{C})\right) > 1-\delta\,.$$

Proof. We need to show that Equation 3.9 holds for $p' = p$, the true transition probability function. Replacing $\widehat{p}_t^{\boldsymbol{\sigma}^t}(\cdot|c)$ by its definition in Equation 3.8 we have:

$$\begin{aligned}
&\|\widehat{p}_t^{\boldsymbol{\sigma}^t}(\cdot|c) - p(\sigma_{s,a}(\cdot)|s,a)\|_1 \\
&\quad= \Big\| \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a')\widehat{p}_t(\sigma_{s',a',t}(\cdot)|s',a') - p(\sigma_{s,a}(\cdot)|s,a) \Big\|_1\,, \\
&\quad= \Big\| \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a')\widehat{p}_t(\sigma_{s',a',t}(\cdot)|s',a') \\
&\qquad - \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a')p(\sigma_{s,a}(\cdot)|s,a) \Big\|_1\,, \\
&\qquad\qquad\qquad\qquad \text{(because } N_t(c) = \textstyle\sum_{(s',a')\in c} N_t(s',a')\text{)} \\
&\quad\leq \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a') \|\widehat{p}_t(\sigma_{s',a',t}(\cdot)|s',a') - p(\sigma_{s,a}(\cdot)|s,a)\|_1\,, \\
&\quad\leq \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a') \|\widehat{p}_t(\sigma_{s',a',t}(\cdot)|s',a') - p(\sigma_{s',a'}(\cdot)|s',a')\|_1\,, \\
&\qquad\qquad\qquad\qquad \text{(since } (s,a)\in c\text{)} \\
&\quad\leq \frac{1}{N_t(c)} \sum_{(s',a')\in c} N_t(s',a') \|\widehat{p}_t(\cdot|s',a') - p(\cdot|s',a')\|_1\,, \qquad \text{(from Lemma 7)}
\end{aligned}$$

Now, since $p \in \mathsf{CB}_{t,\delta}(\mathcal{C},\boldsymbol{\sigma})$, then $\|\widehat{p}_t(\cdot|s',a') - p(\cdot|s',a')\|_1 \leq \beta_{N_t(s',a')}\left(\frac{\delta}{C}\right)$. Therefore, $\|\widehat{p}_t^{\boldsymbol{\sigma}^t}(\cdot|c) - p(\sigma_{s,a}(\cdot)|s,a)\|_1 \leq \frac{1}{N_t(c)}\sum_{(s',a')\in c} N_t(s',a')\beta_{N_t(s',a')}\left(\frac{\delta}{C}\right)$. Consequently, by definition of the confidence bound in (3.9), $p \in \mathsf{CB}_{t,\delta}(\mathcal{C})$.

□

3.5 Unknown Classes: The ApproxEquivalence Algorithm

In this section, we turn to the most challenging situation when both $\mathcal{C}$ and $\boldsymbol{\sigma}$ are unknown to the learner. To this end, we first introduce an algorithm, which we call ApproxEquivalence, that finds an approximate equivalence structure in the MDP by grouping transition probabilities based on statistical tests. ApproxEquivalence is inspired by Khaleghi et al. (2016), which provides a method for clustering time series. Interestingly enough, ApproxEquivalence does not require the knowledge of the number of classes in advance.

We first introduce some definitions. Given $u \subseteq \mathcal{S} \times \mathcal{A}$ and $v \subseteq \mathcal{S} \times \mathcal{A}$, we define the distance between u and v as

$$d(u, v) = \|p^{\boldsymbol{\sigma}_u}(\cdot|u) - p^{\boldsymbol{\sigma}_v}(\cdot|v)\|_1 , \tag{3.12}$$

where $\boldsymbol{\sigma}_u$ is the true mapping for u and u's probability value is the average of the samples of all the state-action pairs in u. In this context, we will refer to u as a *center* since it is the mean of the state-action pairs in u.

ApproxEquivalence relies on finding subsets of $\mathcal{S} \times \mathcal{A}$ that are *statistically close* in terms of the distance function $d(\cdot, \cdot)$. As $d(\cdot, \cdot)$ is unknown, ApproxEquivalence relies on a lower confidence bound on it. For $t \in \mathbb{N}$, we define

$$\varepsilon_{u,t} = \frac{1}{N_t(u)} \sum_{s,a \in u} N_t(s, a) \beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right) \tag{3.13}$$

and let $p_t^{\boldsymbol{\sigma}_{u,t}}(\cdot|u)$ be the empirical estimation of transition probability for center u ordered using the empirical mapping $\boldsymbol{\sigma}_{u,t}$. Note that $\boldsymbol{\sigma}_{u,t}$ is the empirical mapping while $\boldsymbol{\sigma}_u$ is the true mapping for center u.

For $u \subseteq \mathcal{S} \times \mathcal{A}$ and $v \subseteq \mathcal{S} \times \mathcal{A}$, we define the *lower-confidence distance function* between u and v as

$$\widehat{d}_{t,\delta}(u, v) = \left\|\widehat{p}_t^{\boldsymbol{\sigma}_{u,t}}(\cdot|u) - \widehat{p}_t^{\boldsymbol{\sigma}_{v,t}}(\cdot|v)\right\|_1 - \varepsilon_{u,t} - \varepsilon_{v,t} , \tag{3.14}$$

where $\widehat{p}_t^{\boldsymbol{\sigma}_{u,t}}(\cdot|u)$ is the empirical estimation of transition probability for center u ordered using the empirical mapping $\boldsymbol{\sigma}_{u,t}$. When t and δ are clear from the context we use a lighter notation $\widehat{d}(u, v)$ instead of $\widehat{d}_{t,\delta}(u, v)$. We stress that, unlike $d(\cdot, \cdot)$, $\widehat{d}(\cdot, \cdot)$ is not a distance function.

Definition 18 (PAC Neighbor)**.** *For a given equivalence structure $\mathcal{C}$, and given $u \in \mathcal{C}$, we say that $v \in \mathcal{C}$ is a* PAC Neighbor *of u if it satisfies $\widehat{d}(u, v) \leq 0$. We further define $\mathcal{N}(u) = \left\{v \in \mathcal{C} \setminus \{u\} : \widehat{d}(u, v) \leq 0\right\}$ as the set of all PAC Neighbors of u.*

Definition 19 (PAC Nearest Neighbor)**.** *For a given equivalence structure $\mathcal{C}$ and $u \in \mathcal{C}$, we define the* PAC Nearest Neighbor *of u (when it exists) as:*

$$\mathsf{Near}(u, \mathcal{C}) \in \operatorname*{argmin}_{z \in \mathcal{N}(u)} \widehat{d}(u, z) .$$

We can now describe our algorithm ApproxEquivalence, which proceeds as follows. At time t, it receives as input a parameter $\alpha > 1$ that controls the level of aggregation, as well as $N_t(s, a)$ for all state-action pairs $(s, a) \in \mathcal{S} \times \mathcal{A}$. Recall that N_t is a function from state action pairs in $\mathcal{S} \times \mathcal{A}$ to $\mathbb{N}$, and we have extended N_t to classes of state action pairs $u \subseteq \mathcal{S} \times \mathcal{A}$ such that $N_t(u) = \sum_{(s,a) \in u} N_t(s, a)$. Also, $|u|$ denotes the number of elements of u. Starting from the trivial partition of $\{1, \ldots, SA\}$ into $\mathcal{C}^0 = \left\{\{1\}, \ldots, \{SA\}\right\}$, the algorithm builds a coarser partition by iteratively merging elements of $\mathcal{C}^0$ that are *statistically close*. More precisely, the algorithm first sorts the elements in a non-increasing order of sample size $N_t(u)$, so as to promote agents with the tightest confidence intervals. Then, starting from u with the largest $N_t(u)$, it finds the PAC Nearest Neighbor v of u, that is $v = \mathsf{Near}(u, \mathcal{C}^0)$. If $\frac{1}{\alpha} \leq \frac{N_t(u)/|u|}{N_t(v)/|v|} \leq \alpha$, the algorithm merges u and v, thus leading to a novel partition $\mathcal{C}^1$, which contains the new cluster $u \cup v$, and removes u and v. The algorithm continues this procedure with the next element of $\mathcal{C}^0$, until exhaustion, thus finishing the creation of the novel partition $\mathcal{C}^1$ of $\{1, \ldots, SA\}$. ApproxEquivalence continues this process, by ordering

Algorithm 6 ApproxEquivalence

Input: N, α

Initialization: $\mathcal{C}^0 \leftarrow \{\{1\},\{2\},\dots,\{SA\}\}$;
repeat
  $k \leftarrow k+1$
  $\mathcal{C}^k \leftarrow \mathcal{C}^{k-1}$
  Merged $\leftarrow \varnothing$
  for all $u \in \mathcal{C}^{k-1}$ in a non decreasing order of their $N(u)$ **do**
    if $N(u) \neq 0$ and $u \notin$ Merged and $\mathsf{Near}(u, \mathcal{C}^{k-1}) \neq \varnothing$ **then**
      $v \leftarrow \mathsf{Near}(u, \mathcal{C}^{k-1})$
      if $N(v) \neq 0$ and $v \notin$ Merged and $\frac{1}{\alpha} \leq \frac{N(u)/|u|}{N(v)/|v|} \leq \alpha$ **then**
        $\mathcal{C}^k \leftarrow \mathcal{C}^k \setminus (\{u\},\{v\}) \cup \{u \cup v\}$
        Merged $\leftarrow$ Merged $\cup \{v\}$
      end if
    end if
  end for
until $\mathcal{C}^k = \mathcal{C}^{k-1}$

Output: $\mathcal{C}^k$

the elements of $\mathcal{C}^1$ in a non-increasing order, and carrying out similar steps as before, yielding the new partition $\mathcal{C}^2$ and so on, until some iteration k where $\mathcal{C}^{k+1} = \mathcal{C}^k$ (convergence). The pseudo-code of ApproxEquivalence is shown in Algorithm 6.

Remark 5. *Since at each iteration, either two or more subsets are merged, the algorithm ApproxEquivalence stops after, at most, $SA-1$ steps.*

The purpose of the condition $\frac{1}{\alpha} \leq \frac{N_t(u)/|u|}{N_t(v)/|v|} \leq \alpha$ is to ensure the optimism of the RL algorithm that uses ApproxEquivalence as a subroutine (see our C-UCRL algorithm in Section 3.6). This condition prevents merging centers (classes) whose numbers of samples differ a lot, i.e., one for which we are almost certain with one for which we are uncertain and want to act optimistically. Consider the following example (borrowed from Ortner, 2013) where there exists two centers in $\mathcal{C}^k$: one (s_1,a_1) has few samples (and thus optimistically good) and the other (s_2,a_2) has many, and they are in the same approximated class. If the two centers in fact belong to two different classes, the RL algorithm needs to sample (s_1,a_1) more so that it can ultimately separate the two centers. However, since they are currently in the same approximated class, the algorithm may not sample (s_1,a_1) anymore and this will prevent making the distinction between the two centers. Using $\alpha < \infty$, we only merge the centers that have a sufficiently similar number of observations (and thus sufficiently similar confidence). We observe in the experiments that α was needed to force sampling very optimistic state/action pairs for which we only have a few samples. We note that a very similar condition (with $\alpha = 2$) was used in Ortner (2013) for state-aggregation.

We now provide a theoretical guarantee for the correctness of ApproxEquivalence. Since we do not bound the regret for this case, we consider the case $\alpha \to \infty$. The result relies on the following separability assumption.

Assumption 1 (Separability)**.** *There exists some $\Delta > 0$ such that*

$$\forall c \neq c' \in \mathcal{C}, \forall \ell \in c, \forall \ell' \in c', \quad d(\{\ell\},\{\ell'\}) \geq \Delta\,.$$

Note that $\Delta = \min\big\{d(\{\ell\},\{\ell'\}) : \ell,\ell' \in \mathcal{S}\times\mathcal{A} \text{ and } \ell,\ell' \text{ are not in the same class}\big\}$. Define

$$\mathcal{E} = \bigcap_{t\in\mathbb{N}}\bigcap_{s,a}\Big\{\|p(\cdot|s,a) - \widehat{p}_t(\cdot|s,a)\|_1 \le \beta_{N_t(s,a)}\big(\tfrac{\delta}{SA}\big)\Big\}.$$

Recall from Equation (3.1) that $\mathbb{P}(\mathcal{E}) \ge 1-\delta$.

We can now prove our main result, which shows the correctness of ApproxEquivalence when enough samples have been collected.

Theorem 8. *Under Assumption 1, provided that* $\min_{s,a} N_t(s,a) > f^{-1}(\Delta)$*, where* $f : n \mapsto 4\beta_n(\frac{\delta}{SA})$*, ApproxEquivalence with the choice* $\alpha \to \infty$ *outputs the correct equivalence structure* $\mathcal{C}$ *of state-action pairs with probability at least* $1-\delta$*.*

Proof. Fix $t \ge 1$, and consider $\alpha \to \infty$. Assume that $\min_{s,a} N_t(s,a) > f^{-1}(\Delta)$, and that $\mathcal{E}$ holds (which occurs with probability at least $1-\delta$).

To prove the theorem, we first establish by induction on k that $\forall u \in \mathcal{C}^k$ and $\forall (s_1,a_1),(s_2,a_2) \in u$, $\exists v \in \mathcal{C}$ such that $(s_1,a_1),(s_2,a_2) \in v$: we say that $\mathcal{C}^k$ is a *valid partition*. Then, to conclude on the proof, we will show that when the algorithm stops, there are no two states in the same class of $\mathcal{C}$ that have not been merged.

For $k = 0$, since $\mathcal{C}^0$ contains all singletons, $\mathcal{C}^0$ is a valid partition.

For $k \ge 1$, assuming that $\mathcal{C}^{k-1}$ is valid, we need to show that $\mathcal{C}^k$ is also valid. To prove this, it is sufficient to show if two centers u and v from $\mathcal{C}^{k-1}$ are merged, then they are in the same class of $\mathcal{C}$ (i.e. $d(u,v) = 0$). To this end, consider $u, v \in \mathcal{C}^{k-1}$ that are merged by the algorithm in round k, so that $u \cup v \in \mathcal{C}^k$. We need to show that $d(u,v) = 0$. Starting from the definition of d in (3.12), we have

$$\begin{aligned} d(u,v) &= \|p^{\boldsymbol{\sigma}_u}(\cdot|u) - p^{\boldsymbol{\sigma}_v}(\cdot|v)\|_1 \\ &\le \|p^{\boldsymbol{\sigma}_u}(\cdot|u) - \widehat{p}_t^{\,\boldsymbol{\sigma}_{u,t}}(\cdot|u)\|_1 + \|p^{\boldsymbol{\sigma}_v}(\cdot|v) - \widehat{p}_t^{\,\boldsymbol{\sigma}_{v,t}}(\cdot|v)\|_1 + \|\widehat{p}_t^{\,\boldsymbol{\sigma}_{u,t}}(\cdot)|u) - \widehat{p}_t^{\,\boldsymbol{\sigma}_{v,t}}(\cdot|v)\|_1 . \end{aligned} \tag{3.15}$$

The term $\big\|\widehat{p}_t^{\,\boldsymbol{\sigma}_u}(\cdot|u) - p^{\boldsymbol{\sigma}_{u,t}}(\cdot|u)\big\|_1$ is upper bounded as follows:

$$\begin{aligned} \big\|\widehat{p}_t^{\,\boldsymbol{\sigma}_u}(\cdot|u) - p^{\boldsymbol{\sigma}_{u,t}}(\cdot|u)\big\|_1 &= \sum_{x\in\mathcal{S}} |\widehat{p}_t^{\,\boldsymbol{\sigma}_{u,t}}(x|u) - p^{\boldsymbol{\sigma}_u}(x|u)| \\ &= \sum_{x\in\mathcal{S}} \left| \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a)\Big(\widehat{p}_t(\sigma_{s,a,t}(x)|s,a) - p(\sigma_{s,a}(x)|s,a)\Big)\right| \\ &\le \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a) \sum_{x\in\mathcal{S}} \Big|\widehat{p}_t(\sigma_{s,a,t}(x)|s,a) - p(\sigma_{s,a}(x)|s,a)\Big| \\ &\le \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a) \big\|\widehat{p}_t(\sigma_{s,a,t}(\cdot)|s,a) - p(\sigma_{s,a}(\cdot)|s,a)\big\|_1 \\ &\le \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a) \big\|\widehat{p}_t(\cdot|s,a) - p(\cdot|s,a)\big\|_1 , \qquad \text{(Lemma 7)} \\ &\le \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a)\beta_{N_t(s,a)}\big(\tfrac{\delta}{SA}\big), \qquad \text{(From (3.2), under } \mathcal{E}) \\ &= \varepsilon_{u,t} \qquad \text{(From (3.13))} \end{aligned}$$

A similar argument yields $\|\widehat{p}_t^{\,\boldsymbol{\sigma}_{u,t}}(\cdot)|u) - \widehat{p}_t^{\,\boldsymbol{\sigma}_{v,t}}(\cdot|v)\|_1 \le \varepsilon_{v,t}$. By construction, $v = \mathsf{Near}(u,\mathcal{C}^k)$, and since $\alpha \to \infty$, the condition in Line 9 of the algorithm is satisfied.

Thus, $\left\|\widehat{p}_t^{\boldsymbol{\sigma}_{u,t}}(\cdot|u) - \widehat{p}_t^{\boldsymbol{\sigma}_{v,t}}(\cdot|v)\right\|_1 - \varepsilon_{u,t} - \varepsilon_{v,t} \leq 0$. Using (3.15), we arrive at

$$\begin{aligned} d(u,v) &\leq \varepsilon_{u,t} + \varepsilon_{v,t} + \left\|\widehat{p}_t^{\boldsymbol{\sigma}_{u,t}}(\cdot|u) - \widehat{p}_t^{\boldsymbol{\sigma}_{v,t}}(\cdot|v)\right\|_1 \\ &\leq 2\varepsilon_{u,t} + 2\varepsilon_{v,t}. \end{aligned}$$

By the assumption that $\min_{s,a} N_t(s,a) > f^{-1}(\Delta)$, we have

$$\begin{aligned} \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a)\beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right) &\leq \frac{1}{N_t(u)} \sum_{(s,a)\in u} N_t(s,a)\beta_{f^{-1}(\Delta)}\left(\tfrac{\delta}{SA}\right) \\ &= \beta_{f^{-1}(\Delta)}\left(\tfrac{\delta}{SA}\right). \end{aligned}$$

We deduce that we have $\varepsilon_{u,t}$ and $\varepsilon_{v,t}$ lower than $\beta_{f^{-1}(\Delta)}\left(\frac{\delta}{SA}\right)$ and therefore $d(u,v) < 4\beta_{f^{-1}(\Delta)}\left(\frac{\delta}{SA}\right)$. By definition of f, $\Delta = f(f^{-1}(\Delta)) = 4\beta_{f^{-1}(\Delta)}\left(\frac{\delta}{SA}\right)$. By Assumption 1 (separability), we deduce that $d(u,v) = 0$, which concludes the first part of the proof.

As noted in Remark 5, we know that the algorithm stops after at most SA rounds, say at round K. To finalize the proof, it remains to show that when the algorithm stops, there are no two states in the same class of $\mathcal{C}$ that have not been merged. In other words, for any two distinct $u, v \in \mathcal{C}^K$, we need to show that $d(u,v) > 0$. If no merge happens, from line 7 of the algorithm, we have

$$\widehat{d}(u,v) = \|\widehat{p}(\sigma_{u,t}(\cdot)|u) - \widehat{p}(\sigma_{v,t}(\cdot)|v)\|_1 - \varepsilon_{u,t} - \varepsilon_{v,t} > 0\,. \tag{3.16}$$

Now, we need to show that $d(u,v) > 0$. We have

$$\begin{aligned} \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 \leq \|p(\sigma_u(\cdot)|u) - p(\sigma_v(\cdot)|v)\|_1 + \\ \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - p(\sigma_u(\cdot)|u)\|_1 + \|p(\sigma_v(\cdot)|v) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1\,. \end{aligned} \tag{3.17}$$

Hence,

$$\begin{aligned} d(u,v) &= \|p(\sigma_u(\cdot)|u) - p(\sigma_v(\cdot)|v)\|_1 && \text{(Def. (3.12))} \\ &\geq \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 - \|p(\sigma_u(\cdot)|u) - \widehat{p}_t(\sigma_{u,t}(\cdot)|u)\|_1 \\ &\quad - \|p(\sigma_v(\cdot)|v) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 && \text{(from (3.17))} \\ &\geq \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 - \|p(\sigma_u(\cdot)|u) - \widehat{p}_t(\sigma_u(\cdot)|u)\|_1 \\ &\quad - \|p(\sigma_v(\cdot)|v) - \widehat{p}_t(\sigma_v(\cdot)|v)\|_1 && \text{(Lemma 7)} \\ &= \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 - \|p(\cdot|u) - \widehat{p}_t(\cdot|u)\|_1 \\ &\quad - \|p(\cdot|v) - \widehat{p}_t(\cdot|v)\|_1 \\ &\geq \|\widehat{p}_t(\sigma_{u,t}(\cdot)|u) - \widehat{p}_t(\sigma_{v,t}(\cdot)|v)\|_1 - \varepsilon_{u,t} - \varepsilon_{v,t} = \widehat{d}(u,v) > 0\,. && \text{(from (3.16))} \end{aligned}$$

Therefore, since all the partitions constructed by the algorithm are valid, and there is no possibility of merging further at round K, $\mathcal{C}^K = \mathcal{C}$ and the proof is done. □

Proposition 1. *Under the conditions of Theorem 8, we always have* $\frac{\max(1,f^{-1}(\Delta))}{t} \leq \frac{N_t(u)/|u|}{N_t(v)/|v|} \leq \frac{t}{\max(1,f^{-1}(\Delta))}$. *Therefore, it is sufficient to choose* $\alpha \geq \frac{t}{\max(1,f^{-1}(\Delta))}$ *in* ApproxEquivalence *for Theorem 8 to hold.*

Proof. First, we know from Lines 7 and 9 of ApproxEquivalence that $N(u)$ and $N(v)$ are not zero. We have:

$$\begin{aligned}
&\max(1, f^{-1}(\Delta)) \le N_t(s,a) \le t \quad \Leftrightarrow \\
\sum_{s',a' \in u} \max(1, f^{-1}(\Delta)) \le &\sum_{s',a' \in u} N_t(s,a) \le \sum_{s',a' \in u} t \quad \Leftrightarrow \\
&|u| \max(1, f^{-1}(\Delta)) \le N_t(u) \le |u| t \quad \Leftrightarrow \\
&\max(1, f^{-1}(\Delta)) \le \frac{N_t(u)}{|u|} \le t \quad \Leftrightarrow \\
&\frac{1}{t} \le \frac{|u|}{N_t(u)} \le \frac{1}{\max(1, f^{-1}(\Delta))}.
\end{aligned}$$

Combining the two last lines, we have that for any two $u, v \subseteq \mathcal{S} \times \mathcal{A}$, $\frac{\max(1, f^{-1}(\Delta))}{t} \le \frac{N_t(u)/|u|}{N_t(v)/|v|} \le \frac{t}{\max(1, f^{-1}(\Delta))}$. Therefore it is sufficient to choose $\alpha \ge \frac{t}{\max(1, f^{-1}(\Delta))}$ for the condition in Line 9 of ApproxEquivalence to always be satisfied. □

Remark 6. *There exists a trade off between choosing a small α to guarantee optimism and doing regret minimization, or a large α to do consistent clustering. Therefore, appropriately choosing α does not have a trivial solution and is an open problem.*

3.6 Application: The C-UCRL Algorithm

This section is devoted to presenting some applications of equivalence-aware confidence sets introduced in Section 3.4. We present C-UCRL, a natural modification of UCRL2 (Section 2.2.3). The main difference between C-UCRL and UCRL2 is that C-UCRL is capable of exploiting the equivalence structure of the MDPs while estimating it. It takes advantage from this equivalence structure to aggregate the information of similar state-action pairs and better estimate the environment model. This improves the regret of the algorithm by a factor of $\sqrt{\frac{SA}{C}}$. We consider variants of C-UCRL depending on which information is available to the learner in advance.

3.6.1 C-UCRL: Known Equivalence Structure

Here we assume that the learner knows $\mathcal{C}$ and $\boldsymbol{\sigma}$ in advance, and provide a variant of UCRL2 (Section 2.2.3, Algorithm 5), referred to as C-UCRL (Algorithm 7), capable of exploiting the knowledge on $\mathcal{C}$ and $\boldsymbol{\sigma}$. Given δ, at time t, C-UCRL uses the following set of models

$$\mathcal{M}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) = \left\{ (\mathcal{S}, \mathcal{A}, p', \nu') : p' \in \mathsf{Pw}(\mathcal{C}) \text{ and } p'_{\mathcal{C}} \in \mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) \text{ and } \mu' \in \mathsf{CB}'_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) \right\},$$

where $\mathsf{Pw}(\mathcal{C})$ denotes the state-transition functions that are piece-wise constant on $\mathcal{C}$, and where $p'_{\mathcal{C}}$ denotes the function induced by $p' \in \mathsf{Pw}(\mathcal{C})$ over $\mathcal{C}$ (that is $p'_{\mathcal{C}}(\cdot|c) = p'(\cdot|s,a)$ for all $(s,a) \in c$). Recall that in previous sections, as indicated in Remark 3, we considered for conciseness that the reward ν was known. In this section, the proof is given also for the case where the reward function is unknown to the learner. C-UCRL defines

$$t_{k+1} = \min \left\{ t > t_k : \exists c \in \mathcal{C} : \textstyle\sum_{(s,a) \in c} V_{t_k:t}(s,a) \ge N_{t_k}(c) > 0 \right\},$$

Algorithm 7 C-UCRL$(\mathcal{C}, \boldsymbol{\sigma}, \delta)$ with input parameter $\delta \in (0,1]$

1: **Initialize:** For all $c_0 \in \mathcal{C}$, set $N_0(c) = 0$ and $V_0(c) = 0$. Set $t_0 = 0$, $t = 1$, $k = 1$, and observe the initial state s_1
2: $s_t = s_1$
3: **for** episodes $k \geq 1$ **do**
4: Set $t_k = t$
5: Set $N_{t_k}(c) = N_{t_k - 1}(c) + V_{k-1}(c)$ for all c
6: Compute empirical estimates $\widehat{\mu}^{\boldsymbol{\sigma}}_{t_k}(c)$ and $\widehat{p}^{\boldsymbol{\sigma}}_{t_k}(\cdot|c)$ for all c
7: Compute $\pi^{+}_{t_k} = \text{EVI}\Big(\widehat{\mu}^{\boldsymbol{\sigma}}_{t_k}, \widehat{p}^{\boldsymbol{\sigma}}_{t_k}, N_{t_k}, \frac{1}{\sqrt{t_k}}, \frac{\delta}{C}\Big)$ — see Algorithm 4
8: $a_t = \pi^{+}_{t_k}(s_t)$
9: $c_t \in \mathcal{C}$ is the class containing (s_t, a_t)
10: **while** $V_k(c_t) < \max\{1, N_{t_k}(c_t)\}$ **do**
11: Play action $a_t = \pi^{+}_{t_k}(s_t)$, observe the next state s_{t+1} and reward $r_t(s_t, a_t)$
12: Set $c_t \in \mathcal{C}$ to be the class containing (s_t, a_t)
13: Set $V_k(c_t) = V_k(c_t) + 1$
14: Set $t = t + 1$
15: **end while**
16: **end for**

where $V_{t_k:t}(s,a)$ denotes the number of observations from time t_k to t and $N_{t_k}(s,a)$ is the number of observations until time t_k of the state action pair (s,a).

We note that forcing the condition $p' \in \mathsf{Pw}(\mathcal{C})$ may be computationally difficult. To ensure an efficient implementation, we use the same EVI algorithm of UCRL2, where for $(s,a) \in c$, we replace $\widehat{p}_t(\cdot|s,a)$ and $\beta_{N_t(s,a)}(\frac{\delta}{SA})$ respectively with $\widehat{p}^{\boldsymbol{\sigma}}_t(\cdot|c)$ and $\beta_{N_t(c)}(\frac{\delta}{C})$. Let us now explain the C-UCRL algorithm. It starts with zero initialization of N and V for all classes and the choice of s_1 as the starting state. For each episode k, we set the start time of episode k as t_k, we represent the number of observations for episode k by $N_{t_k}(c)$, the sum of the number of observations until episode t_{k-1} and the number of observations in episode $k-1$ (line 5). After computing the empirical estimates for distribution means and transition probabilities, we call the extended value iteration (EVI) to compute the policy $\pi^{+}_{t_k}$ (line 7). We find the class c_t in which the state action pair (s_t, a_t) falls (line 9). As long as the number of observations at episode k for class c_t remains smaller than the number of observations until episode t_k for class c_t (line 10), we play action a_t and observe the next state s_{t+1} and reward $r_t(s_t, a_t)$ (line 11), find the state-action's corresponding class c_t, increase the number of observations for episode k (line 13), and transit to time $t+1$ (line 14).

To bound the number of episodes produced by the algorithm we use the following lemma.

Lemma 8 (Number of episodes)**.** *The number $m(T)$ of episodes of C-UCRL up to time $T \geq C$ is upper bounded by*

$$m(T) \leq C \log_2\left(\tfrac{8T}{C}\right).$$

Proof. The proof uses similar steps as in the proof of Proposition 18 in Jaksch, Ortner, and Auer (2010). Recall that given c, $N_T(c)$ and $V_k(c) = V_{t_k}(c)$ denote as the total number of state-action class observations, up to step T and for episode k, respectively. For any c, let $K(c)$ denote the number of episodes where a state-action pair from c is sampled: $K(c) = \sum_{k=1}^{m(T)} \mathbb{1}_{\{V_k(c) > 0\}}$. It is worth mentioning that if $N_k(c) > 0$ and $V_k(c) = N_{t_k}(c)$, by the design of the algorithm, $N_{t_{k+1}}(c) = 2N_{t_k}(c)$ (due to line 5

and 10 of Algorithm 7). Hence, for all class c

$$\begin{aligned} N_{t_{m(T)}}(c) &= \sum_{k=1}^{m(T)} V_k(c) \\ &\geq 1 + \sum_{k:V_k(c)=N_{t_k}(c)} N_{t_k}(c) \\ &\geq 1 + \sum_{i=1}^{K(c)} 2^{i-1} \\ &= 2^{K(c)}. \end{aligned}$$

where the first 1 in the second step is due to the first step of line 10 in Algorithm 7. If $N_{t_{m(T)}}(c) = 0$, then $K(c) = 0$, so that $N_{t_{m(T)}}(c) \geq 2^{K(c)} - 1$ for all c. Thus,

$$T = \sum_{c\in\mathcal{C}} N_{t_{m(T)}}(c) \geq \sum_{c\in\mathcal{C}} \left(2^{K(c)} - 1\right). \tag{3.18}$$

On the other hand, an episode has happened when either $N_{t_k}(c) = 0$ or $N_{t_k}(c) = V_k(c)$. Therefore, $m(T) \leq 1 + C + \sum_{c\in\mathcal{C}} K(c)$ and consequently, $\sum_{c\in\mathcal{C}} K(c) \geq m(T) - 1 - C$. Because 2^x is convex, by Jensen's inequality, we have $\frac{1}{C}\sum_{c\in\mathcal{C}} 2^{K(c)} > 2^{\sum_{c\in\mathcal{C}} \frac{K(c)}{C}}$. Hence we obtain

$$\sum_{c\in\mathcal{C}} 2^{K(c)} \geq C 2^{\sum_{c\in\mathcal{C}} \frac{K(c)}{C}} \geq C 2^{\frac{m(T)-1}{C} - 1}. \tag{3.19}$$

Putting together Equations 3.18 and 3.19, we obtain $T \geq C\left(2^{\frac{m(T)-1}{C} - 1} - 1\right)$. Therefore,

$$\begin{aligned} m(T) &\leq 1 + C + C\log_2\left(\tfrac{T+C}{C}\right) \\ &\leq 1 + C + C\log_2\left(\tfrac{2T}{C}\right) \\ &\leq 1 + 2C + C\log_2\left(\tfrac{T}{C}\right) \\ &\leq 3C + C\log_2\left(\tfrac{T}{C}\right) \\ &\leq C\log_2\left(\tfrac{8T}{C}\right), \end{aligned}$$

for $T > C$, thus concluding the proof. □

We first provide the following time-uniform concentration inequality to control a bounded martingale difference sequence which is an extension of the Azuma-Hoeffding lemma (Jaksch, Ortner, and Auer, 2010).

Lemma 9 (Time-uniform Azuma-Hoeffding). *Let $(X_t)_{t\geq 1}$ be a martingale difference sequence (Definition 11) bounded by b for some $b > 0$ (that is, $|X_t| \leq b$ for all t). Then, for all $\delta \in (0,1)$,*

$$\mathbb{P}\left(\exists T \in \mathbb{N} : \sum_{t=1}^{T} X_t \geq b\sqrt{\frac{1}{2}(T+1)\log\left(\sqrt{T+1}/\delta\right)}\right) \leq \delta.$$

A modification of the analysis of Jaksch, Ortner, and Auer (2010) yields the following theorem.

Theorem 9 (Regret of C-UCRL). *Given $\mathcal{C}$, $\boldsymbol{\sigma}$ and for any $\delta \in (0,1)$ such that with probability higher than $1-\delta$, uniformly over all time horizon T,*

$$\mathfrak{R}(\text{C-UCRL}(\mathcal{C}, \boldsymbol{\sigma}, \delta'), T) \leq 20D\sqrt{CT\left(S + \log(C\sqrt{T+1}/\delta')\right)} + DC\log_2\left(\tfrac{8T}{C}\right).$$

where $\delta' = \delta/4$, C is the number of classes and D is the diameter of the MDP.

Proof. To simplify notations, we define the short-hand $J_k = J_{t_k}$ for various random variables that are fixed within a given episode k (for example $\mathcal{M}_{k,\delta'} = \mathcal{M}_{t_k,\delta'}$). Denote by $m(T)$ the number of episodes initiated by the algorithm up to time T. Using the definition of regret where $g_\star$ is the optimal gain, we have

$$\begin{aligned}\mathfrak{R}(T) &= \sum_{t=1}^{T} g_\star - \sum_{t=1}^{T} r_t(s_t, a_t) \\ &= \sum_{t=1}^{T} (g_\star - \mu(s_t, a_t)) + \sum_{t=1}^{T} (\mu(s_t, a_t) - r_t(s_t, a_t)) \,.\end{aligned}$$

The first part of the above equation is bounded by $\sum_{s,a} N_{m(T)}(s,a)(g_\star - \mu(s,a))$ and for the second part, note that we can define Lemma 9's X_t as $(\mu(s_t,a_t) - r_t(s_t,a_t))$ and according to definition, it is a MDS. Therefore, Lemma 9 holds and for all $\delta' \in (0,1)$

$$\mathfrak{R}(T) \le \sum_{s,a} N_{m(T)}(s,a)(g_\star - \mu(s,a)) + \sqrt{\tfrac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')}\,, \tag{3.20}$$

with probability at least $1 - \delta'$. We have

$$\begin{aligned}\sum_{s,a} N_{t_{m(T)}}(s,a)(g_\star - \mu(s,a)) &= \sum_{k=1}^{m(T)} \sum_{s,a} \sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}_{\{s_t=s, a_t=a\}} \big(g_\star - \mu(s,a)\big) \\ &= \sum_{k=1}^{m(T)} \sum_{s,a} V_k(s,a)\big(g_\star - \mu(s,a)\big)\,.\end{aligned}$$

Defining $V_k(c) = \sum_{s,a} V_k(s,a)$ for $c \in \mathcal{C}$, we further obtain

$$\sum_{s,a} N_{t_{m(T)}}(s,a)(g_\star - \mu(s,a)) = \sum_{k=1}^{m(T)} \sum_{c \in \mathcal{C}} V_k(c)\big(g_\star - \mu(c)\big)\,,$$

where we have used that $\mu(s,a)$ has constant value $\mu(c)$ for all $(s,a) \in c$. For $1 \le k \le m(T)$, we define the regret of episode k as $\Delta_k = \sum_{c \in \mathcal{C}} V_k(c)\big(g_\star - \mu(c)\big)$. Hence, with probability at least $1 - \delta'$,

$$\mathfrak{R}(T) \le \sum_{k=1}^{m(T)} \Delta_k + \sqrt{\tfrac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')}\,.$$

Of course the objective of our algorithm is to minimize the regret. However, to help doing so, it also tries for searching for the true model M among a set of plausible MDPs $\mathcal{M}_{k,\delta'}$. We say an episode is *good* if $M \in \mathcal{M}_{k,\delta'}$ (that is, the set $\mathcal{M}_{k,\delta'}$ of plausible MDPs contains the true model), and *bad* otherwise.

Control of the regret due to bad episodes ($M \notin \mathcal{M}_{k,\delta'}$). Due to using time-uniform instead of time-instantaneous confidence bounds, we can show that with high probability, all episodes are good for $T \in \mathbb{N}$. Indeed, using the definition of the confidence sets in (3.2), for any $\delta' \in (0,1)$, we have

$$\mathbb{P}\left(\bigcap_t p \in \mathsf{CB}_{t,\delta'}(\mathcal{C}, \boldsymbol{\sigma}) \wedge \mu \in \mathsf{CB}'_{t,\delta'}(\mathcal{C}, \boldsymbol{\sigma})\right) \ge 1 - 2\delta'.$$

We have $M \in \mathcal{M}_{k,\delta'}$ for all k in $[1, m(T)]$ with probability at least $1 - 2\delta'$ and

$$\sum_{k=1}^{m(T)} \Delta_k \mathbb{1}_{\{M \notin \mathcal{M}_{k,\delta'}\}} = 0\,,$$

that is, bad episodes do not contribute to the regret.

Control of the regret due to good episodes ($M \in \mathcal{M}_{k,\delta'}$). We closely follow Jaksch, Ortner, and Auer (2010) and decompose the regret to control the transition and reward functions. At a high level, we make two major modifications: (i) we use the time-uniform bound stated in Lemma 9 to control the martingale difference sequence that appears; and (ii) as the stopping criterion of C-UCRL slightly differs from that of UCRL2, we use Lemma 8 to control the number $m(T)$ of episodes. Consider a good episode k (hence, $M \in \mathcal{M}_{k,\delta'}$). From Jaksch, Ortner, and Auer (2010), we know that the EVI algorithm outputs a policy π_k^+ and $\widetilde{M}_{k,\delta'}$ satisfying $g_{\pi_k^+}^{\widetilde{M}_{k,\delta'}} \geq g_\star - \frac{1}{\sqrt{t_k}}$. Let us define $g_k = g_{\pi_k^+}^{\widetilde{M}_{k,\delta'}}$. It then follows that

$$\Delta_k = \sum_{c \in \mathcal{C}} V_k(c)\big(g_\star - \mu(c)\big) \leq \sum_{c \in \mathcal{C}} V_k(c)\big(g_k - \mu(c)\big) + \sum_{c \in \mathcal{C}} \frac{V_k(c)}{\sqrt{t_k}}\,. \tag{3.21}$$

Defining $V_k = \big(V_k\big(s, \pi_k^+(s)\big)_{s \in \mathcal{S}}$ and combining Lemma 6 with (3.21) in the last step yields

$$\begin{aligned}
\Delta_k &\leq \sum_{s,a} V_k(s,a)\big(g_k - \mu(s,a)\big) + \sum_{s,a} \frac{V_k(s,a)}{\sqrt{t_k}} \\
&= \sum_{s,a} V_k(s,a)\big(g_k - \widetilde{\mu}_k(s,a)\big) + \sum_{s,a} V_k(s,a)\big(\widetilde{\mu}_k(s,a) - \mu(s,a)\big) + \sum_{s,a} \frac{V_k(s,a)}{\sqrt{t_k}} \\
&\leq V_k^\top (\widetilde{\mathbf{P}}_k - \mathbf{I}) u_k^{(i)} + \sum_{s,a} V_k(s,a)\big(\widetilde{\mu}_k(s,a) - \mu(s,a)\big) + 2 \sum_{s,a} \frac{V_k(s,a)}{\sqrt{t_k}}\,,
\end{aligned}$$

where i is the last step of the loop in the EVI algorithm (page 16) and $\widetilde{\mathbf{P}}_k = \big(\widetilde{p}_t(x|s, \pi_k^+(s))\big)_{s,x \in \mathcal{S}}$. Similarly to Jaksch, Ortner, and Auer (2010), we define $w_k(s) = u_k^{(i)}(s) - \frac{1}{2}(\min_{s'} u_k^{(i)}(s') + \max_{s'} u_k^{(i)}(s'))$ for all $s \in \mathcal{S}$. Note that $(\widetilde{\mathbf{P}}_k - \mathbf{I})(\min_{s'} u_k^{(i)}(s') + \max_{s'} u_k^{(i)}(s'))$ is 0 due to the fact that $\widetilde{\mathbf{P}}_k$ is row-stochastic. Therefore we have $V_k^\top (\widetilde{\mathbf{P}}_k - \mathbf{I}) u_k^{(i)} = V_k^\top (\widetilde{\mathbf{P}}_k - \mathbf{I}) w_k^{(i)}$ and thus

$$\Delta_k \leq V_k^\top (\widetilde{\mathbf{P}}_k - \mathbf{I}) w_k + \sum_{s,a} V_k(s,a)\big(\widetilde{\mu}_k(s,a) - \mu(s,a)\big) + 2 \sum_{s,a} \frac{V_k(s,a)}{\sqrt{t_k}}\,. \tag{3.22}$$

The second term in the right-hand side can be upper bounded as follows. Fix pair (s,a) and let $c_{s,a}$ denote the cluster to which (s,a) belongs. The fact $M \in \mathcal{M}_{k,\delta'}$

implies that $\mu \in \mathsf{CB}'_{t,\delta'}(\mathcal{C}, \boldsymbol{\sigma})$, so

$$\begin{aligned}\widetilde{\mu}_k(s,a) - \mu(s,a) &\leq |\widetilde{\mu}_k(s,a) - \widehat{\mu}_k(s,a)| + |\widehat{\mu}_k(s,a) - \mu(s,a)| \leq 2\beta'_{N_{t_k}(c_{s,a})}(\tfrac{\delta'}{C}) \\ &= 2\sqrt{\frac{1}{2N_{t_k}(c_{s,a})}\Big(1+\frac{1}{N_{t_k}(c_{s,a})}\Big)\log\Big(C\sqrt{N_{t_k}(c_{s,a})+1}/\delta'\Big)} \\ &\leq 2\sqrt{\frac{1}{N_{t_k}(c_{s,a})}\log\big(C\sqrt{T+1}/\delta'\big)}\,,\end{aligned}$$

where we have used $1 \leq N_{t_k}(c_{s,a}) \leq T$ in the last inequality. Using this bound and noting that $t_k \geq N_{t_k}(c)$, we obtain

$$\Delta_k \leq V_k^\top(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k + 2\Big(\sqrt{\log\big(C\sqrt{T+1}/\delta'\big)} + 1\Big)\sum_{c\in\mathcal{C}}\frac{V_k(c)}{\sqrt{N_{t_k}(c)}}\,. \tag{3.23}$$

In what follows, we derive an upper bound on $V_k^\top(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k$. Similarly to Jaksch, Ortner, and Auer (2010), we consider the following decomposition:

$$V_k^\top(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k = \underbrace{V_k^\top(\widetilde{\mathbf{P}}_k - \mathbf{P}_k)w_k}_{L_1(k)} + \underbrace{V_k^\top(\mathbf{P}_k - \mathbf{I})w_k}_{L_2(k)}\,.$$

We upper bound $L_1(k)$ as follows:

$$\begin{aligned}L_1(k) &\leq \sum_{s'}\sum_{s,a} V_k(s,a)\big(\widetilde{p}_k(s'|s,a) - p(s'|s,a)\big)w_k(s') \\ &\leq \sum_{s,a} V_k(s,a)\|\widetilde{p}_k(\cdot|s,a) - p(\cdot|s,a)\|_1\|w_k\|_\infty\end{aligned}$$

Note that $\|w_k\|_\infty \leq \frac{D}{2}$ according to Jaksch, Ortner, and Auer (2010) where D is the diameter. The confidence interval (3.2) implies that $\|\widetilde{p}_k(\cdot|s,a) - p(\cdot|s,a)\|_1 \leq \beta_{N_{t_k}(c)}$ and thus

$$\begin{aligned}L_1(k) &\leq \frac{D}{2}\sum_{s,a} V_k(s,a)\beta_{N_{t_k}(c_{s,a})}(\tfrac{\delta'}{C}) \\ &= \frac{D}{2}\sum_{c\in\mathcal{C}} V_k(c)\beta_{N_{t_k}(c)}(\tfrac{\delta'}{C}) \\ &\leq \frac{D}{2}\sum_{c\in C} V_k(c)\sqrt{2(1+\frac{1}{N_{t_k}})\frac{\log\big(C2^S\sqrt{T+1}/\delta'\big)}{N_{t_k}(c)}} \\ &\leq 2D\sqrt{\log\big(C2^S\sqrt{T+1}/\delta'\big)}\sum_{c\in\mathcal{C}}\frac{V_k(c)}{\sqrt{N_{t_k}(c)}}\,.\end{aligned} \tag{3.24}$$

To upper bound $L_2(k)$, similarly to the proof of Jaksch, Ortner, and Auer (2010, Theorem 2), we define the sequence $(X_t)_{t\geq 1}$, with $X_t = (p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}})w_{k_t}\mathbb{1}_{\{M\in\mathcal{M}_{k_t,\delta'}\}}$, for all t, where k_t denotes the episode containing step t and $\mathbf{e}_j$ is a vector where it is one at index j and zero otherwise. Note that $\mathbb{E}[X_t|s_1,a_1,\ldots,s_t,a_t] = 0$. Indeed when $M \notin \mathcal{M}_{k_t,\delta'}$, $X_t = 0$ and $\mathbb{E}[X_t|s_1,a_1,\ldots,s_t,a_t] = 0$, and when $M \in \mathcal{M}_{k_t,\delta'}$,

$$\begin{aligned}X_t &= (p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}})w_{k_t} \\ &= (p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}})\big(u_{k_t}^{(i)}(s) - \tfrac{1}{2}(\min_{s'} u_{k_t}^{(i)}(s') + \max_{s'} u_{k_t}^{(i)}(s'))\big)\,.\end{aligned}$$

Since w_{k_t} is measurable given $s_1, a_1, \dots, s_t, a_t$, it comes out of expectation and it remains to calculate $\mathbb{E}[p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}} | s_1, a_1, \dots, s_t, a_t] = \sum_{s' \in \mathcal{S}} p(s'|s_t,a_t) - p(s'|s_t,a_t) = 0$. So $(X_t)_{t \geq 1}$ is martingale difference sequence. Furthermore, $|X_t| \leq D$. Indeed, for all t, by the Hölder inequality,

$$|X_t| \leq \|p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}}\|_1 \|w_{k(t)}\|_\infty \leq \Big(\|p(\cdot|s_t,a_t)\|_1 + \|\mathbf{e}_{s_{t+1}}\|_1 \Big) \frac{D}{2} = D \,.$$

Using similar steps as in Jaksch, Ortner, and Auer (2010), for any k with $M \in \mathcal{M}_{k,\delta'}$, we have that:

$$L_2(k) \leq \sum_{t=t_k}^{t_{k+1}-1} X_t + D \,,$$

so that $\sum_{k=1}^{m(T)} L_2(k) \leq \sum_{t=1}^{T} X_t + m(T)D$. Therefore, by Lemma 9, we deduce that with probability at least $1-\delta'$

$$\begin{aligned} \sum_{k=1}^{m(T)} L_2(k) &\leq D\sqrt{\frac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')} + m(T)D \\ &\leq D\sqrt{\frac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')} + DC\log_2(\tfrac{8T}{C}) \,, \end{aligned} \tag{3.25}$$

where the last step follows from Lemma 8.

Final control. Combing (3.23)–(3.25) and summing over all episodes gives

$$\begin{aligned} &\sum_{k=1}^{m(T)} \Delta_k \mathbb{1}_{\{M \in \mathcal{M}_{k,\delta'}\}} \\ &\quad \leq \sum_{k=1}^{m(T)} L_1(k) + \sum_{k=1}^{m(T)} L_2(k) + 2\Big(\sqrt{\log\big(C\sqrt{T+1}/\delta'\big)} + 1\Big) \sum_{k=1}^{m(T)} \sum_{c \in \mathcal{C}} \frac{V_k(c)}{\sqrt{N_{t_k}(c)}} \\ &\quad \leq 2\Big(D\sqrt{\log\big(C2^S\sqrt{T+1}/\delta'\big)} + \sqrt{\log\big(C\sqrt{T+1}/\delta'\big)} + 1\Big) \sum_{k=1}^{m(T)} \sum_{c \in \mathcal{C}} \frac{\nu_k(c)}{\sqrt{N_{t_k}(c)}} \\ &\quad + D\sqrt{2(T+1)\log\big(\sqrt{T+1}/\delta'\big)} + DC\log_2(\tfrac{8T}{C}) \,, \end{aligned} \tag{3.26}$$

with probability at least $1-\delta'$. To upper bound the right-hand side, we recall the following lemma.

Lemma 10 (Jaksch, Ortner, and Auer, 2010, Lemma 19)**.** *For any sequence of numbers $z_1, z_2, \dots, z_n$ with $0 \leq z_k \leq Z_{k-1} = \max\{1, \sum_{i=1}^{k-1} z_i\}$,*

$$\sum_{k=1}^{n} \frac{z_k}{\sqrt{Z_{k-1}}} \leq \big(\sqrt{2}+1\big)\sqrt{Z_n} \,.$$

Note that $N_{t_k}(c) = \sum_{k'<k} V_{k'}(c)$. Hence, applying Lemma 10 gives

$$\sum_{c \in \mathcal{C}} \sum_{k=1}^{m(T)} \frac{V_k(c)}{\sqrt{N_k(c)}} \leq \sum_{c \in \mathcal{C}} \big(\sqrt{2}+1\big)\sqrt{N_{t_{m(T)}}(c)} \leq \big(\sqrt{2}+1\big)\sqrt{CT} \,,$$

where the last step follows from Jensen's inequality and $\sum_c N_{t_{m(T)}}(c) = T$. Therefore,

$$\sum_{k=1}^{m(T)} \Delta_k \mathbb{1}_{\{M\in\mathcal{M}_k\}} \leq D\sqrt{2(T+1)\log\left(\sqrt{T+1}/\delta'\right)} + DC\log_2(\tfrac{8T}{C}) \\ + 2\left(\sqrt{2}+1\right)\left(D\sqrt{\log\left(C2^S\sqrt{T+1}/\delta'\right)} + \sqrt{\log\left(C\sqrt{T+1}/\delta'\right)}+1\right)\sqrt{CT},$$

with probability of at least $1-\delta'$.

Finally and at a high level, we have three sources of error leading to $4\delta'$ probability terms. The first bad event E_1 is from the regret decomposition that makes appear one high probability term in Equation 3.20: with probability $1-\delta'$, we have

$$\sum_{t=1}^{T}(\mu(s_t,a_t) - r_t(s_t,a_t)) > \sqrt{\frac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')}\,.$$

Then, the second good event E_2 is the probability that $M \in M_{k,\delta'}$ which is true with probability $1-2\delta'$. And the third bad event E_3, the remainder martingale difference term controlled by the diameter D in Equation 3.25, that also adds one δ' term: with probability $1-\delta'$

$$\sum_{k=1}^{m(T)} (p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}}) w_{k_t} \mathbb{1}_{\{M\in\mathcal{M}_{k_t}\}} = \sum_{t=1}^{T} X_t > D\sqrt{\frac{1}{2}(T+1)\log(\sqrt{T+1}/\delta')}\,.$$

Finally, the regret of C-UCRL is controlled on an event of probability higher than $1-\delta'-2\delta'-\delta' = 1-4\delta'$, uniformly over all T. Therefore, for any $\delta \in (0,1)$, taking $\delta' = \delta/4$, the regret of C-UCRL with δ' as parameter is controlled on an event of probability higher than $1-\delta$, uniformly over all T, by

$$\begin{aligned}
\mathfrak{R}(T) &\leq 2\left(\sqrt{2}+1\right)\left(D\sqrt{\log\left(C2^S\sqrt{T+1}/\delta'\right)} + \sqrt{\log\left(C\sqrt{T+1}/\delta'\right)}+1\right)\sqrt{CT} \\
&\quad + D\sqrt{\tfrac{1}{2}(T+1)\log\left(\sqrt{T+1}/\delta'\right)} + DC\log_2(\tfrac{8T}{C}) + \sqrt{\tfrac{1}{2}(T+1)\log\left(\sqrt{T+1}/\delta'\right)} \\
&\leq 6\left(D\sqrt{CT\left(S+\log\left(C\sqrt{T+1}/\delta'\right)\right)} + \sqrt{CT\log\left(C\sqrt{T+1}/\delta'\right)} + \sqrt{CT}\right) \\
&\quad + (D+1)\sqrt{\tfrac{1}{2}(T+1)\log\left(\sqrt{T+1}/\delta'\right)} + DC\log_2(\tfrac{8T}{C}) \\
&\leq 20D\sqrt{CT\left(S+\log(C\sqrt{T+1}/\delta')\right)} + DC\log_2(\tfrac{8T}{C})\,,
\end{aligned}$$

thus completing the proof. □

Remark 7. *In the above theorem to bound the regret, if we assume that reward ν is known, we obtain a probability of $1-3\delta'$ instead of $1-4\delta'$.*

Theorem 9 bounds the regret of C-UCRL which shows that efficiently exploiting the knowledge of $\mathcal{C}$ and $\boldsymbol{\sigma}$ yields an improvement over the regret bound of UCRL2 (stated in Theorem 7) by a factor of $\sqrt{SA/C}$, which could be a huge improvement when $C \ll SA$. This is the case in, for instance, many grid-world environments, as illustrated below.

Examples of MDP. In this part, we illustrate the notion of similarity and equivalence class presented in Definitions 16-17 on some grid-world environments. For this purpose, we consider grid-world MDPs as described in the beginning of the chapter. Below, we show four examples of grid-world environments defined according to the

Environment	States	5×5	7×7	9×9	100×100
2-Room (Fig 3.3)	SA	100	196	324	4×10^4
	C	4	4	4	4
4-Room (Fig 3.4)	SA	100	196	324	4×10^4
	C	3	3	3	3

TABLE 3.1: Number of state-action pairs compared to the number of classes in two types of grid-like environments.

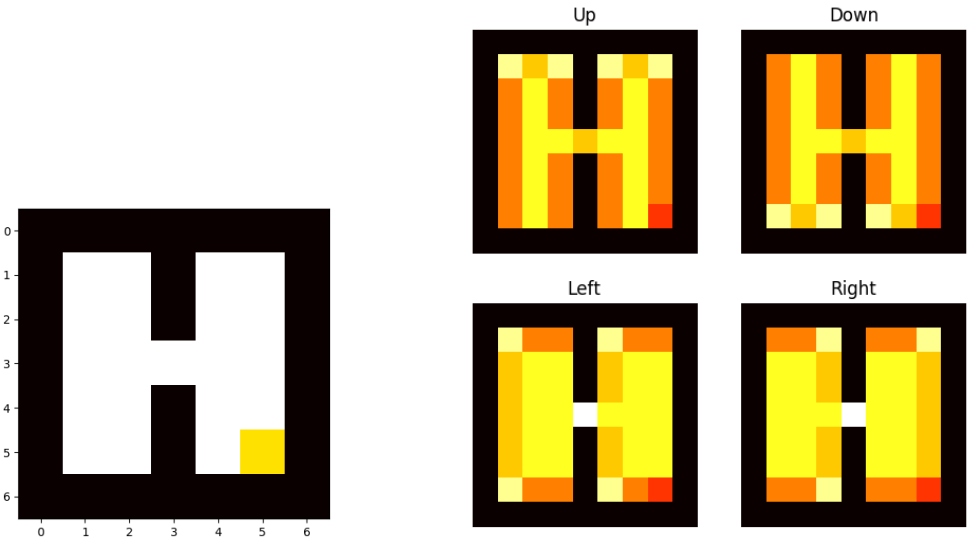


FIGURE 3.3: Left: Two-room grid-world (left) with walls in black, and goal state in yellow. Right: equivalence classes for state-action pairs (one color per class).

above scheme, with different numbers of state-action pairs. The number SA of state-action pairs in the simple 2-room and 4-room MDPs shown in (Figures 3.3 and 3.4) changes as the grid size grows while the number C of classes stays almost fixed (see Table 3.1), showing the advantage of our proposed approach. More complex examples are shown in Figures 3.5 and 3.6 and as can be seen, the number of classes are still very small compared to the very large grid world.

3.6.2 $\widehat{\text{C}}$-UCRL: Unknown Equivalence Structure

Now we consider the case where $\mathcal{C}$ is unknown to the learner. In order to accommodate this situation, we use ApproxEquivalence in order to estimate the equivalence structure.

We introduce $\widehat{\text{C}}$-UCRL (Algorithm 8), which proceeds similarly to C-UCRL but does not know the class and the true profile mapping. At each time t, ApproxEquivalence outputs $\mathcal{C}_t$ as an estimate of the true equivalence structure $\mathcal{C}$. Then, C-UCRL uses the following set of models taking $\mathcal{C}_t$ as input:

$$\mathcal{M}_{t,\delta}(\mathcal{C}_t) = \left\{ (\mathcal{S}, \mathcal{A}, p', \nu) : p' \in \mathsf{Pw}(\mathcal{C}_t) \text{ and } p'_{\mathcal{C}_t} \in \mathsf{CB}_{t,\delta}(\mathcal{C}_t) \right\}.$$

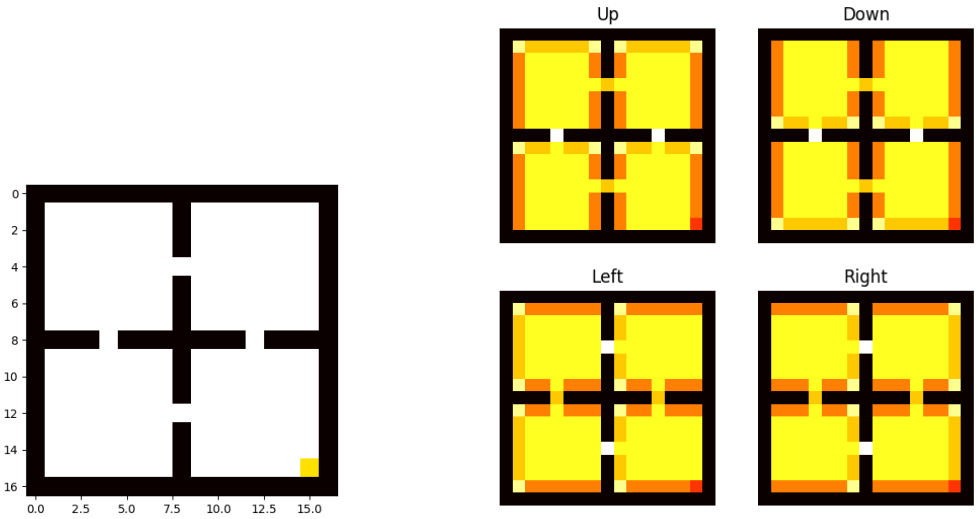


FIGURE 3.4: Left: Four-room grid-world (left) with walls in black, and goal state in yellow. Right: equivalence classes for state-action pairs (one color per class).

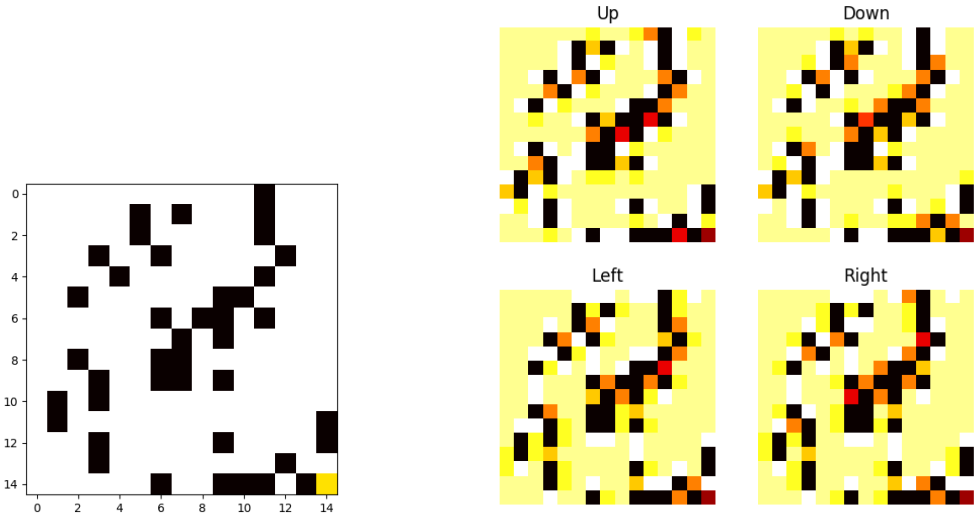


FIGURE 3.5: Left: A more complex grid-world (left) with walls in black, and goal state in yellow. Right: equivalence classes for state-action pairs (one color per class).

Then, it sets the starting step of episode $k+1$ as:

$$t_{k+1} = \min \Big\{ t > t_k : \exists c \in \mathcal{C}_{t_k}, \sum_{(s,a)\in c} V_{t_k:t}(s,a) \geq N_{t_k}(c) > 0 \text{ or } \exists s, a,\ V_{t_k:t}(s,a) \geq N_{t_k}(s,a) > 0 \Big\}.$$

Using the disjunction and the two given cases above provides more statistical confidence for the case where the true profile mappings and class are not given. Since we do not know the classes, we need to check the state-action pairs. Intuitively speaking, this helps to separate the optimistically joined classes, especially in the beginning of

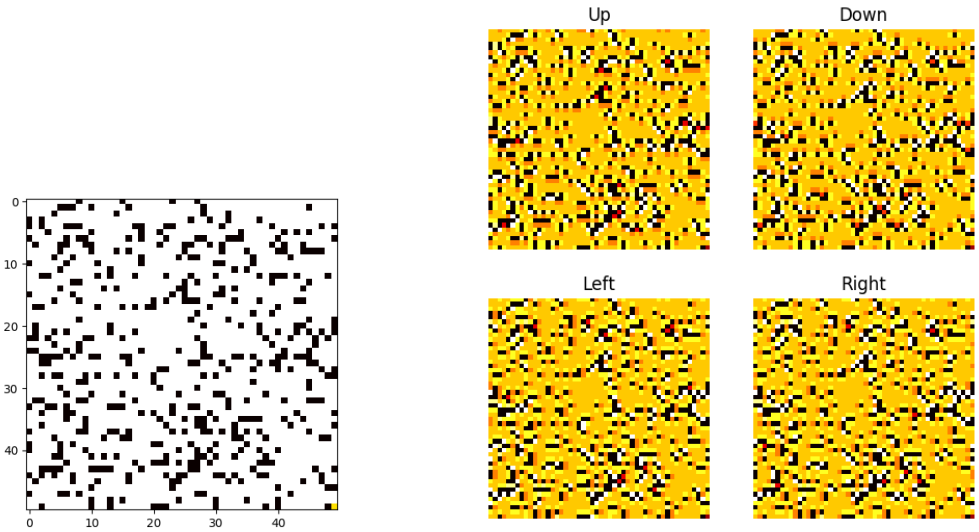


FIGURE 3.6: Left: A more complex grid-world (left) with walls in black, and goal state in yellow. Right: equivalence classes for state-action pairs (one color per class).

procedure. Consider the case when two centers are merged and one is sampled more while the other remains the same. The sampled center's data is doubled and it is a good time to check our modeling again while considering the merged center, we still need more sample to check the model. This is not important for the case that we know the classes in advance, because we cannot incorrectly merge pairs that are in the same class.

Remark 8. *Note that $\mathcal{M}_{t,\delta}(\mathcal{C}) \neq \mathcal{M}_{t,\delta}(\mathcal{C}_t)$ as we may have $\mathcal{C}_t \neq \mathcal{C}$. Nonetheless, the design of ApproxEquivalence, which relies on confidence bounds, ensures that $\mathcal{C}_t$ is informative enough, in the sense that $\mathcal{M}_{t,\delta}(\mathcal{C}_t)$ could be much smaller (hence, better) than a set of models that one would obtain by ignoring equivalence structure; this is also validated by the numerical experiments in ergodic environments in the next subsection.*

3.6.3 Numerical Experiments

We conduct numerical experiments to examine the performance of the proposed variants of C-UCRL, and compare it to that of UCRL2-L[4]. In our experiments with unknown mappings and classes, for the subroutine ApproxEquivalence called by $\widehat{\text{C}}$-UCRL, we set $\varepsilon_{u,t} = \beta_{N_t(u)}\left(\frac{\delta}{3SA}\right)$ for $u \in \mathcal{S} \times \mathcal{A}$ where δ is the risk parameter and we have used $\frac{\delta}{3}$ since the reward is assumed to be known. Recall that $\varepsilon_{u,t}$ is used in the definition of both $\widehat{d}(\cdot,\cdot)$ and $\mathsf{CB}_{t,\delta}(\mathcal{C}_t)$ (see Equations 3.14 and 3.9).

Remark 9. *To avoid over-grouping, we empirically use a more constrained definition of PAC Neighbor that the one needed in our theory (Definition 18). Specifically, for v to be a PAC Neighbor of u, in addition to requiring $\widehat{d}(u,v) \leq 0$, we also require that $\widehat{d}(\{j\},\{j'\}) \leq 0$ for all $j \in u$ and $j' \in v$, and $\widehat{d}(\{j\}, u \cup v) \leq 0$ for all $j \in u \cup v$.*

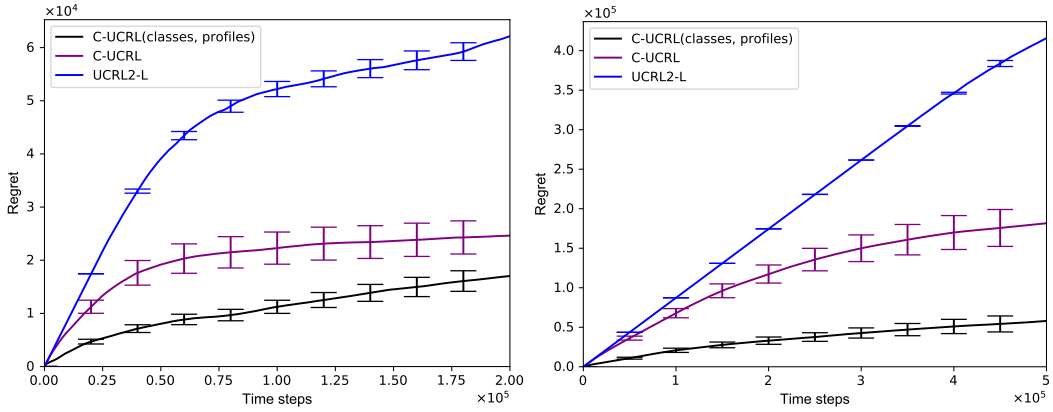
[4] UCRL2-L is a variant of UCRL2, which uses confidence bounds derived by combining Hoeffding's and Weissman's inequalities with the Laplace method, as in (3.2). We stress that UCRL2-L attains a smaller regret than the original UCRL2 of Jaksch, Ortner, and Auer (2010).

Algorithm 8 $\widehat{\text{C}}$-UCRL with input parameter $\delta \in (0,1], \alpha$

Initialize: For all (s,a), set $N_0(s,a) = 0$ and $V_0(s,a) = 0$. For all $c_0 \in \mathcal{C}$, set $N_0(c) = 0$ and $V_0(c) = 0$. Set $t_0 = 0$, $t = 1$, $k = 1$, and observe the initial state s_1
$s_t = s_1$
for episodes $k \geq 1$ **do**
  Set $t_k = t$
  Set $N_{t_k}(s,a) = N_{t_{k-1}}(s,a) + V_k(s,a)$ for all (s,a)
  Set $N_{t_k}(c) = N_{t_{k-1}}(c) + V_k(c)$ for all $c \in \mathcal{C}_{t_{k-1}}$
  Compute empirical estimates $\boldsymbol{\sigma}_{t_k}$
  Find $\mathcal{C}_{t_k}$ using ApproxEquivalence(N_t, α)
  Set $N_{t_k}(c) = N_{t_{k-1}}(c) + V_{k-1}(c)$ for all $c \in \mathcal{C}_{t_k}$
  Compute empirical estimates $\widehat{\mu}_{t_k}^{\boldsymbol{\sigma}_{t_k}}(c)$ and $\widehat{p}_{t_k}^{\boldsymbol{\sigma}_{t_k}}(\cdot|c)$ for all $c \in \mathcal{C}_{t_k}$
  Compute $\pi_{t_k}^{+} = \texttt{EVI}\left(\widehat{\mu}_{t_k}^{\boldsymbol{\sigma}_{t_k}}, \widehat{p}_{t_k}^{\boldsymbol{\sigma}_{t_k}}, N_{t_k}, \frac{1}{\sqrt{t_k}}, \frac{\delta}{C}\right)$ — see Algorithm 4
  $a_t = \pi_{t_k}^{+}(s_t)$
  $c_t \in \mathcal{C}$ is the class containing (s_t, a_t)
  while $V_k(c_t) < \max\{1, N_{t_k}(c_t)\}$ and $V_k(s_t, \pi_{t_k}^{+}(s_t)) < \max\{1, N_{t_k}(s_t, \pi_{t_k}^{+}(s_t))\}$ **do**
    Play action $a_t = \pi_{t_k}^{+}(s_t)$, observe the next state s_{t+1} and reward $r_t(s_t, a_t)$
    Set $c_t \in \mathcal{C}_{t_k}$ to be the class containing (s_t, a_t)
    Set $V_k(s_t, a_t) = V_k(s_t, a_t) + 1$
    Set $V_k(c_t) = V_k(c_t) + 1$
    Set $t = t + 1$
  end while
end for

In the first set of experiments, we examine the regret of various algorithms in ergodic environments. Specifically, we consider the ergodic *RiverSwim* MDP, shown in Figure 3.2, with 25 and 50 states. In both cases, we have $C = 6$ classes. In Figures 3.7(a) and 3.7(b), we plot the regret against time steps under C-UCRL, $\widehat{\text{C}}$-UCRL, and UCRL2-L executed in the aforementioned environments. The results are averaged over 100 independent runs, and the 95% confidence intervals are shown. All algorithms use $\delta = 0.05$, and for $\widehat{\text{C}}$-UCRL, we use $\alpha = 4$. As the curves show, the proposed C-UCRL algorithms significantly outperform UCRL2-L. As expected, since it is given the true classes and profile mappings, C-UCRL attains the smallest regret. In particular, in the 25-state environment and at the final time step, C-UCRL attains a regret smaller than that of UCRL2-L by a factor of approximately $\sqrt{SA/C} = \sqrt{50/6} \approx 2.9$, thus verifying Theorem 9. Similarly, we may expect an improvement in regret by a factor of around $\sqrt{SA/C} = \sqrt{100/6} \approx 4.1$ in the other environment. We however get a better improvement (by a factor of around 8), which can be attributed to the increase in the regret of UCRL2-L due to a long burn-in phase (i.e., the phase before the algorithm starts learning). $\widehat{\text{C}}$-UCRL, which does not know the true classes and profile mappings in advance, has larger regret than C-UCRL but still largely outperforms UCRL2-L.

We now turn our attention to the quality of approximate equivalence structure produced by ApproxEquivalence (Algorithm 6), which is used as a sub-routine of $\widehat{\text{C}}$-UCRL. To this aim, we introduce two performance measures to assess the clustering quality. The first one is defined as the total number of state-action pairs that are *mis-clustered*, normalized by the total number SA of pairs. We refer to this measure as the *mis-clustering ratio*. More precisely, let $\mathcal{C}_t$ denote the empirical equivalence structure output by ApproxEquivalence at time t. For a given $c \in \mathcal{C}_t$, we consider the



(a) 25-state Ergodic RiverSwim (b) 50-state Ergodic RiverSwim

FIGURE 3.7: Regret of various algorithms in Ergodic RiverSwim environments.

restriction of $\mathcal{C}$ to c which is the intersection of c and $\mathcal{C}$, denoted by $\mathcal{C}|c$. We find $\ell(c) \in \mathcal{C}|c$ that has the largest cardinality: $\ell(c) \in \operatorname{argmax}_{x \in \mathcal{C}|c} |x|$. Now, we define

$$\text{mis-clustering ratio at time } t = \frac{1}{SA} \sum_{c \in \mathcal{C}_t} |c \setminus \ell(c)| .$$

Note that the mis-clustering ratio falls in $[0, 1]$ as $\sum_{c \in \mathcal{C}_t} |c| = SA$ for all t. Our second performance measure accounts for the error in the aggregated empirical transition probability due to mis-clustered pairs. We refer to this measure as *mis-clustering bias*. Precisely speaking, for a given pair $e \in \mathcal{S} \times \mathcal{A}$, we denote by $c_e \in \mathcal{C}_t$ the set containing e in $\mathcal{C}_t$. Then, we define the mis-clustering bias at time t as

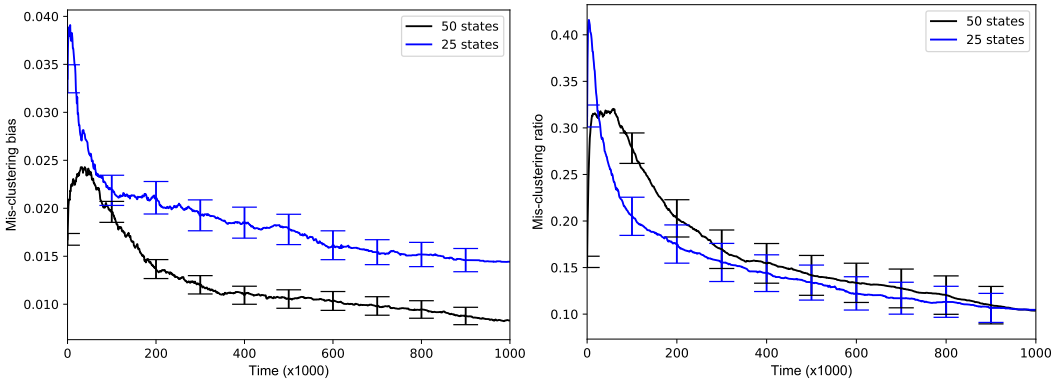
$$\text{mis-clustering bias at time } t = \sum_{c \in \mathcal{C}_t} \sum_{e \notin \ell(c)} \|\widehat{p}_t(\cdot|c_e) - \widehat{p}_t(\cdot|c_e \setminus \{e\})\|_1 .$$

Remark 10. *As explained on page 28, $\widehat{\text{C}}$-UCRL could lead to a biased estimation of p when the number of samples is lower than the bound provided in Theorem 8. Nevertheless, such a bias can be controlled thanks to α (we set $\alpha = 4$ in our experiments), see Figure 3.8 as well as in the sub-linear regret plots of the algorithm (Figures 3.7 and 3.9).*

In Figures 3.8(a) and 3.8(b), we plot the "mis-clustering bias" and "mis-clustering ratio" for the empirical equivalence structures produced in the previous experiments. We observe on the figures that the errors in terms of the aforementioned performance measures reduce. These errors do not vanish quickly, thus indicating that the generated empirical equivalence structures do not agree with the true one. Yet, they help reduce uncertainty in the transition probabilities, and, in turn, reduce the regret; we refer to Remark 8 for a related discussion.

In the second set of experiments, we consider two communicating environments[5]: *4-room grid-world* (with 49 states) and *RiverSwim* (with 25 states). In Figures 3.9(a) and 3.9(b), we plot the regret against time steps under C-UCRL, $\widehat{\text{C}}$-UCRL, and UCRL2-L, and similarly to the previous case, we set $\delta = 0.05$ and $\alpha = 4$. The results are

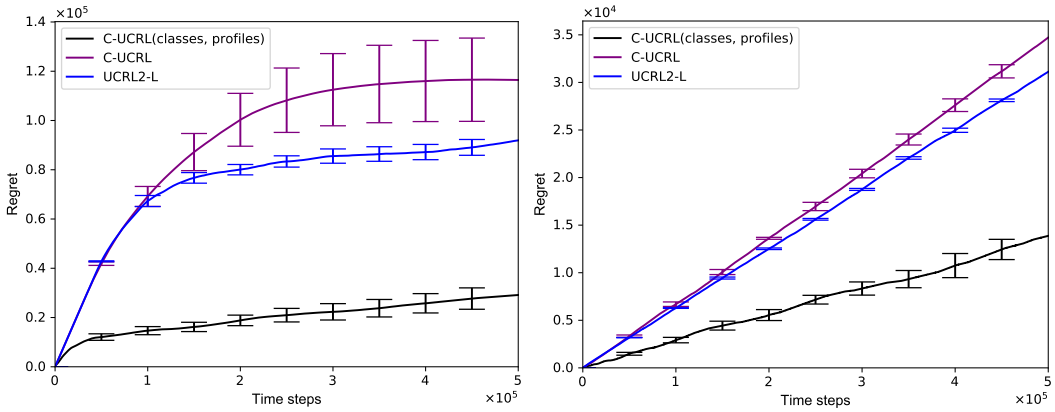
[5] An MDP is called communicating if for any two states s_1, s_0, there exists a policy such that the expected number of steps to reach s_0 from s_1 is finite



(a) Mis-clustering bias (b) Mis-clustering ratio

FIGURE 3.8: Assessment of quality of approximate equivalence structures for Ergodic RiverSwim with 25 and 50 states.

averaged over 100 independent runs, and the 95% confidence intervals are shown. In both environments, C-UCRL significantly outperforms UCRL2-L. However, $\widehat{\text{C}}$-UCRL attains a regret which is slightly worse than that of UCRL2-L. This can be attributed to the fact that ApproxEquivalence is unable to find an accurate enough equivalence structure in these non-ergodic environments.



(a) 25-state RiverSwim (b) Grid-World

FIGURE 3.9: Regret of various algorithms in communicating environments.

Chapter 4

Collaborative Algorithms for Online Personalized Mean Estimation

In this chapter, we tackle the online personalized mean estimation problem in a network of collaborative agents. Each agent has access to a (personal) process that generates samples from a real-valued distribution and seeks to estimate its mean. We study the case where some of the distributions have the same mean, and the agents are allowed to actively query information from other agents. The goal is to design an algorithm that enables each agent to improve its mean estimate thanks to communication with other agents. The means as well as the number of distributions with same mean are unknown, which makes the task nontrivial. We introduce a novel collaborative strategy to solve this online personalized mean estimation problem. We analyze its time complexity and introduce variants that enjoy good performance in numerical experiments. We also extend our approach to the setting where clusters of agents with similar means seek to estimate the mean of their cluster.

4.1 Introduction

With the wide spreading of personal digital devices, ubiquitous computing and IoT (Internet of Things), the need for decentralized and collaborative computing has become more pregnant. Indeed, devices are first of all designed to collect data and this data may be sensitive and/or too large to be transmitted. Therefore, it is often preferable to keep the data on-device, where it has been collected. Local processing on a single device is a always possible option but learning in isolation has slow convergence time. In that case, collaborative strategies can be investigated to improve statistically and faster learning. In recent years, such collaborative approaches have been broadly referred to as federated learning (Kairouz et al., 2021).

The data collected at each device reflects the specific usage, production patterns and objective of the associated agent. Therefore, we must solve a set of *personalized tasks over heterogeneous data distributions*. Even though the tasks are personalized, collaboration can play a significant role in reducing the time complexity and accelerating learning in presence of agents who share similar objectives. An important building block to design collaborative algorithms is then to identify agents acquiring data from the same (or similar) distribution. This is particularly difficult to do in an *online* setting, in which data becomes available sequentially over time.

In this work, we explore this challenging objective in the context of a new problem: *online personalized mean estimation*. Formally, each agent continuously receives data

from a *personal* σ-sub-Gaussian distribution and aims to construct an accurate estimation of its mean as fast as possible. At each step, each agent receives a new sample from its distribution but is also allowed to query the current local average of another agent. To enable collaboration, we assume the existence of an underlying *class structure* where agents in the same class have the same mean value. We also consider a relaxed assumption where the means of agents in a class are close (but not necessarily equal). Such assumptions are natural in many real-world applications (Adi et al., 2020). A simple example is that of in different environments, monitoring parameters such as temperature in order to accurately estimate their mean (see for instance Mateo et al., 2013). Another example is collaborative filtering, where the goal is to estimate user preferences by leveraging the existence of clusters of users with similar preferences (Su and Khoshgoftaar, 2009). Crucially, the number of classes and their cardinality are unknown to the agents and must be discovered in an online fashion.

We propose collaborative algorithms to solve this problem, where agents identify the class they belong to in an online fashion so as to better and faster estimate their own mean by assigning weights to other agents' estimates. Our approach is grounded in Probably Approximately Correct (PAC) theory, allowing agents to iteratively discard agents in different classes with high confidence. We provide a theoretical analysis of our approach by bounding the time required by an agent to correctly estimate its class with high probability, as well as the time required by an agent to estimate its mean to the desired accuracy. Our results highlight the dependence on the gaps between the true means of agents in different classes, and show that in some settings our approach achieves nearly the same time complexity as an oracle who would know the classes beforehand. Our numerical experiments on synthetic data are in line with our theoretical findings and show that some empirical variants of our approach can further improve the performance in practice.

The chapter is organized as follows. Section 4.2 discusses the related work on federated learning and collaborative online learning. In Section 4.3, we formally describe the problem setting and introduce relevant notations. In Section 4.4, we introduce our algorithm and its variants. Section 4.5 presents our theoretical analysis of the proposed algorithm in terms of class and mean estimation time complexity. Section 4.6 is devoted to illustrative numerical experiments. Section 4.7 extends our approach to the case where classes consist of agents with similar(but not necessarily equal) means and agents seek to estimate the mean of their class.

4.2 Related Work

Over the last few years, collaborative estimation and learning problems involving several agents with local datasets have been extensively investigated under the broad term of Federated Learning (FL) (Kairouz et al., 2021). While traditional FL algorithms learn a global estimate for all agents, more personalized approaches have recently attracted a lot of interest (see for instance Vanhaesebrouck, Bellet, and Tommasi, 2017; Smith et al., 2017; Fallah, Mokhtari, and Ozdaglar, 2020; Sattler, Müller, and Samek, 2020; Hanzely et al., 2020; Marfoq et al., 2021, and references therein). However, these approaches are not suitable for online learning and often lack clear statistical assumptions on the relation between local data distributions.

In the online setting, the work on collaborative learning has largely focused on multi-armed bandits (MAB). Most approaches however consider a *single* MAB instance which is solved collaboratively by multiple agents. Collaboration between agents can be implemented through broadcast messages to all agents (Hillel et al.,

2013; Tao, Zhang, and Zhou, 2019), via a server (Wang et al., 2020b), or relying only on local message exchanges over a network graph (Sankararaman, Ganesh, and Shakkottai, 2019; Martínez-Rubio, Kanade, and Rebeschini, 2019; Wang et al., 2020a; Landgren, Srivastava, and Leonard, 2021). Other approaches do not allow explicit communication but instead consider a collision model where agents receive no reward if several agents pull the same arm (Boursier and Perchet, 2019; Wang et al., 2020a). In any case, all agents aim at solving the *same* task.

Some recent work considered collaborative MAB settings where the arm means vary across agents. Extending their previous work (Boursier and Perchet, 2019), Boursier et al. (2019) consider the case where arm means can vary among players. Under their collision model, the problem reduces to finding a one-to-one assignment of agents to arms. In Shi and Shen (2021), the local arm means of each agent are IID random realizations of fixed global means and the goal is to solve the global MAB using only observations from the local arms with an algorithm inspired from traditional FL. Similarly, Karpov and Zhang (2022) extend the work of Tao, Zhang, and Zhou (2019) by considering different local arm means for each agent with the goal to identify the arm with largest aggregated mean. Shi, Shen, and Yang (2021) introduce a limited amount of personalization by extending the model of Shi and Shen (2021) to optimize a mixture between the global and local MAB objectives. Réda, Vakili, and Kaufmann (2022) further consider a *known* weighted combination of the local MAB objectives. A crucial difference with our work is that there is no need to discover relations between local distributions to solve the above problems.

Another related problem is to identify a graph structure on top of the arms in MAB. Kocák and Garivier (2020) and Kocák and Garivier (2021) construct a similarity graph while solving the best arm identification problem, but consider only a single agent. In contrast, our work considers a multi-agent setting with personalized estimation tasks, and our approach discovers similarities across agents' tasks in an online manner.

4.3 Problem Setting

We consider a mean estimation problem involving A agents. The goal of each agent $a \in [A] = \{1, 2, \dots, A\}$ is to estimate the mean μ_a of a personal distribution ν_a over $\mathbb{R}$. In this work, we assume that there exists $\sigma \geq 0$ such that each ν_a is σ-sub-Gaussian (Def. 8). This classical assumption captures a property of strong tail decay, and includes in particular Gaussian distributions (in that case, the smallest possible σ^2 corresponds to the variance) as well as any distribution supported on a bounded interval (e.g., Bernoulli distributions).

We consider an online and collaborative setting where data points are received sequentially and agents can query each other to learn about their respective distributions. Agents should be thought of as different user devices which operate in parallel. Therefore, they all receive a new sample and query another agent at each time step.

Formally, we assume that time is synchronized between agents and at each time step t, each agent a receives a new sample x_a^t from its personal distribution ν_a with mean μ_a, which is used to update its local mean estimate $\bar{x}_{a,a}^t = \frac{1}{t}\sum_{t'=1}^{t} x_a^{t'}$. It also chooses another agent l to *query*. As a response from querying agent l, agent a receives the local average $\bar{x}_{l,l}^t$ of agent l (i.e., the average of t independent samples from the personal distribution ν_l) and stores it in its memory $\bar{x}_{a,l}^t$ along with the corresponding number of samples $n_{a,l}^t = t$. Each agent a thus keeps a memory

$[(\bar{x}^t_{a,1}, n^t_{a,1}), \dots, (\bar{x}^t_{a,A}, n^t_{a,A})]$ of the last local averages (and associated number of samples) that it received from other agents. The information contained in this memory is used to compute an *estimate* μ^t_a of μ_a at each time t. Our goal is to design a query and estimation procedure for each agent.

As described above, note that when an agent queries another agent at time t, it does not receive one sample from this agent (as e.g. in multi-armed bandits), but receives the full statistics of observations of this agent up to time t. This has considerably much more information than in typical MAB settings, and naturally requires specific strategies.

To measure the performance of an algorithm, we rely on the following notion of (ε, δ)-convergence in probability (Bertsekas and Tsitsiklis, 2002; Wasserman, 2013), which we recall below.

Definition 20 (PAC-convergence)**.** *An estimation procedure for agent a is called (ε, δ)-convergent if there exists $\tau_a \in \mathbb{N}$ such that the probability that the mean estimator μ^t_a of agent a is ε-distant from the true mean for any time $t > \tau_a$ is at least $1 - \delta$:*

$$\mathbb{P}\left(\forall t > \tau_a : |\mu^t_a - \mu_a| \leq \varepsilon\right) > 1 - \delta. \tag{4.1}$$

While it is easy to design (ε, δ)-convergent estimation procedure for a single agent taken in isolation, the goal of this work is to propose collaborative algorithms where agents benefit from information from other agents by taking advantage of the relation between the agents' distributions. This will allow them to build up more accurate estimations in less time, i.e., with smaller time complexity τ_a.

Specifically, to foster collaboration between agents, we consider that the set of agents $[A]$ is partitioned into equivalence classes that correspond to agents with the same mean.[1] In real scenarios, these classes may represent sensors in the same environment, objects with the same technical characteristics, users with the same behavior, etc. This assumption makes it possible for an agent to design strategies to identify other agents in the same class and to use their estimates in order to speed up the estimation of his/her own mean. Formally, we define the class of a as the set of agents who have the same mean as a.

Definition 21 (Similarity class)**.** *The* similarity class *of agent a is given by:*

$$\mathcal{C}_a = \{l \in [A] : \Delta_{a,l} = 0\},$$

where $\Delta_{a,l} = |\mu_a - \mu_l|$ is the gap between the means of agent a and agent l.

The gaps $\{\Delta_{a,l}\}_{a,l \in [A]}$ define the problem structure. We consider that the agents do not know the means, the gaps, or even the number of underlying classes. Hence the classes $\{\mathcal{C}_a\}_{a \in [A]}$ are completely unknown. This makes the problem quite challenging.

4.4 Proposed Approach

In this section, we first introduce some of the key technical components used in our approach, and then present our proposed algorithm.

[1] In Section 4.7, we will consider a relaxed version of this assumption where classes consist of agents with *similar* (not necessarily equal) means.

4.4.1 Main Concepts

In our approach, each agent a computes confidence intervals $I_{a,l} = [\bar{x}^t_{a,l} - \beta_\delta(n^t_{a,l}), \bar{x}^t_{a,l} + \beta_\delta(n^t_{a,l})]$ for the mean estimators $[\bar{x}^t_{a,1}, \ldots, \bar{x}^t_{a,A}]$ that it holds in memory at time t. The generic confidence bound $\beta_\delta(n^t_{a,l})$ takes as input the number of samples $n^t_{a,l}$ seen for agent l at time t, and δ corresponds to the risk parameter so that with probability at least $1-\delta$ the true mean μ_l falls within the confidence interval $I_{a,l}$.

Agent a will use these confidence intervals to assess whether another agent l belongs to the class $\mathcal{C}_a$ through the evaluation of an optimistic distance defined below.

Definition 22 (Optimistic distance)**.** *The* optimistic distance *with agent l from the perspective of agent a is defined as:*

$$d^t_{a,\delta}(l) = |\bar{x}^t_{a,a} - \bar{x}^t_{a,l}| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}). \tag{4.2}$$

The "optimistic" terminology is justified by the fact that $d^t_{a,\delta}(l)$ is, with high probability, a *lower bound* on the distance between the true means μ_a and μ_l of distributions ν_a and ν_l. Recall that two agents belong to the same class if the distance of their true mean is zero. Since these values are unknown, the idea of the above heuristic is to provide a proxy based on observed data and high probability confidence bounds. In particular, we will adopt the *Optimism in Face of Uncertainty Principle (OFU)* (see Auer, Cesa-Bianchi, and Fischer, 2002) and consider that two agents may be in the same class if the optimistic distance is zero or less. Hence, we define an optimistic notion of class accordingly.

Definition 23. *The* optimistic similarity class *from the perspective of agent a at time t is defined as:*

$$\mathcal{C}^t_a = \{l \in [A] : d^t_{a,\delta}(l) \leq 0\}.$$

Having introduced the above concepts, we can now present our algorithm.

4.4.2 Algorithm

The collaborative mean estimation algorithm we propose, called ColME, is given in Algorithm 9 (taking the perspective of agent a). For conciseness, we consider that $\beta_\delta(0) = +\infty$. At each step t, agent a performs three main steps.

In the **Perceive** step, the agent receives a sample from its distribution and updates its local average together with the number of samples.

In the **Query** step, agent a selects another agent following a query strategy given as a parameter to the ColME algorithm. Agent a runs the choose_agent function to select another agent l and asks for its current local estimate to update its memory. We propose two variants for the choose_agent function:

- Round-Robin: cycle over the set $[A]$ of agents one by one in a fixed order.
- Restricted-Round-Robin: like round-robin but ignores agents that are not in the set of optimistically similar agents $\mathcal{C}^t_a$.

The focus on round-robin-style strategies is justified by the information structure of our problem setting, which is very different from classic bandits. Indeed, querying an agent at time t produces an estimate computed on the t observations collected by this agent so far. The choice of variant (Round-Robin or Restricted-Round-Robin) will affect the class identification time complexity, as we shall discuss later.

Algorithm 9 ColME

Parameters: agent a, time horizon H, risk δ, weighting scheme α, and query strategy choose_agent

$\forall l \in [A]$: $\bar{x}^0_{a,l} \leftarrow 0$, $n^0_{a,l} \leftarrow 0$
$\mathcal{C}^0_a \leftarrow \{l \in [A] : d^0_{a,\delta}(l) \leq 0\} = [A]$
for $t = 1, \dots, H$ **do**
  $\forall l \in [A]$: $\bar{x}^t_{a,l} \leftarrow \bar{x}^{t-1}_{a,l}$, $n^t_{a,l} \leftarrow n^{t-1}_{a,l}$
  Perceive:
    Receive sample $x^t_a \sim \nu_a$
    $\bar{x}^t_{a,a} \leftarrow \bar{x}^{t-1}_{a,a} \times \frac{t-1}{t} + x^t_a \times \frac{1}{t}$, $n^t_{a,a} \leftarrow t$
  Query:
    $\mathcal{C}^t_a \leftarrow \{l \in [A] : d^t_{a,\delta}(l) \leq 0\}$
    Query agent $l =$ choose_agent$(\mathcal{C}^t_a)$ to get $\bar{x}^t_{l,l}$
    $\bar{x}^t_{a,l} \leftarrow \bar{x}^t_{l,l}$, $n^t_{a,l} \leftarrow t$
  Estimate:
    $\mathcal{C}^t_a \leftarrow \{l \in [A] : d^t_{a,\delta}(l) \leq 0\}$
    $\mu^t_a \leftarrow \sum_{l \in \mathcal{C}^t_a} \alpha^t_{a,l} \times \bar{x}^t_{a,l}$
end for

Output: μ^H_a

Finally, in the **Estimate** step, agent a computes the optimistic similarity class $\mathcal{C}^t_a$ based on available information, and constructs its mean estimate as a weighted aggregation of the local averages of agents that belong to $\mathcal{C}^t_a$. We propose different weighting mechanisms:

Simple weighting. This is a natural weighting mechanism for aggregating samples:

$$\alpha^t_{a,l} = \frac{n^t_{a,l}}{\sum_{l \in C^t_a} n^t_{a,l}}.$$

Soft weighting. This is a heuristic weighting mechanism which leverages the intuition that the more the confidence intervals of two agents overlap, the more likely that they are in the same class. Moreover, the smaller the union of the agent means, the more confident we are that the agents are in the same class. In other words, we are not equally confident about all the agents that are selected for estimation, and this weighting mechanism incorporates this information:

$$\alpha^t_{a,l} = n^t_{a,l} \frac{|I_{a,a} \cap I_{a,l}|}{|I_{a,a} \cup I_{a,l}|} \times \frac{1}{Z_{\text{soft}}},$$

where $Z_{\text{soft}} = \sum_{i \in C^t_a} \frac{n^t_{a,i} |I_{a,a} \cup I_{a,i}|}{|I_{a,a} \cap I_{a,i}|}$ is a normalization factor.

Aggressive weighting. This is an extension of the previous soft weighting mechanism that is more selective. Not only does it consider the overlap and intersection of the agents' confidence intervals, but it also requires the size of the intersection to be larger than half the size of both confidence intervals from the two agents. Let us denote the binary value associated with this condition by $E_{a,l} =$

$\mathbb{1}_{\{|I_{a,a}\cap I_{a,l}|>\min\{\beta_\delta(n^t_{a,l}),\beta_\delta(n^t_{a,a})\}\}}$. Then

$$\alpha^t_{a,l} = n^t_{a,l}\frac{|I_{a,a}\cap I_{a,l}|}{|I_{a,a}\cup I_{a,l}|}\times\frac{E_{a,l}}{Z_{\text{agg}}},$$

where $Z_{\text{agg}} = \sum_{i\in C^t_a}\frac{n^t_{a,i}|I_{a,a}\cup I_{a,i}|\times E_{a,i}}{|I_{a,a}\cap I_{a,i}|}$ is a normalization factor.

4.4.3 Baselines

We introduce two baselines that will be used to put the performance of our approach into perspective, both theoretically and empirically.

Local estimation. Estimates are computed without any collaboration, using only samples received from the agent's own distribution, i.e. $\mu^t_a = \bar{x}^t_{a,a}$.

Oracle weighting. The agent knows the true class $\mathcal{C}_a$ via an oracle and uses the simple weighting $\alpha^t_{a,l} = \frac{n^t_{a,l}}{\sum_{l\in\mathcal{C}_a} n^t_{a,l}}$.

4.5 Theoretical Analysis

In this section, we provide a theoretical analysis of our algorithm ColME for the query strategy Restricted-Round-Robin and the simple weighting scheme. Specifically, we bound the time complexity in probability for both class and mean estimation.

A key aspect of our analysis is to characterize when the optimistic similarity class (Definition 23) coincides with the true classes. We show that this is the case when two conditions hold. First, for a given agent a, we need the confidence interval computed by a about agent l to contain the true mean μ_l for all $l \in A$.

Definition 24. *We define the following events:*

$$E^t_a = \bigcap_{l\in[A]} |\bar{x}^t_{a,l} - \mu_l| \le \beta_\delta(n^t_{a,l}), \tag{4.3}$$

$$E_a = \bigcap_{t\in\mathbb{N}} E^t_a. \tag{4.4}$$

We can guarantee that E_a holds with high probability via an appropriate parameterization of confidence intervals. We use the so-called Laplace method (Maillard, 2019).

Lemma 11. *Let $\delta \in (0,1)$, $a \in [A]$. Setting $\beta_\delta(n) = \sigma\sqrt{2\frac{1}{n}\times(1+\frac{1}{n})\ln(\sqrt{n+1}/\gamma(\delta))}$ with $\gamma(\delta) = \frac{\delta}{8\times A}$, we have:*

$$\mathbb{P}(E_a) \ge 1 - \frac{\delta}{8}. \tag{4.5}$$

Proof. Let us recall that $E_a = \bigcap_{t\in\mathbb{N}}\bigcap_{l\in[A]} |\bar{x}^t_{a,l} - \mu_l| \le \beta_\delta(n^t_{a,l})$. Then

$$\begin{aligned}
\mathbb{P}(E_a) &= 1 - \mathbb{P}(\bar{E_a}),\\
&= 1 - \mathbb{P}\left(\exists t\in\mathbb{N},\, \exists l\in[A] : |\bar{x}^t_{a,l} - \mu_l| > \beta_\delta(n^t_{a,l})\right),\\
&\ge 1 - \sum_{l\in[A]} \mathbb{P}\left(\exists t\in\mathbb{N} : |\bar{x}^t_{a,l} - \mu_l| > \beta_\delta(n^t_{a,l})\right).
\end{aligned}$$

defining $\gamma(\delta) = \frac{\delta}{8\times A}$ and using Lemma 4,

$$\mathbb{P}(E_a) \geq 1 - \sum_{l\in[A]} \mathbb{P}\Big(\exists t \in \mathbb{N} : |\bar{x}^t_{a,l} - \mu_l| > \sigma\sqrt{\frac{2}{n^t_{a,l}} \times (1 + \frac{1}{n^t_{a,l}}) \ln(\sqrt{n^t_{a,l}+1}/\gamma(\delta))}\Big),$$
$$\geq 1 - \sum_{l\in[A]} \gamma(\delta) = 1 - \sum_{l\in[A]} \frac{\delta}{8A} = 1 - \frac{\delta}{8}.$$ □

The second condition is that agent's a memory about the local estimates of other agents should contain enough samples. Let us denote by $\lceil \beta_\delta^{-1}(x) \rceil$ the smallest integer n such that $x > \beta_\delta(n)$.

Definition 25. *From the perspective of agent a and at time t, event G^t_a is defined as:*

$$G^t_a = \bigcap_{l\in[A]} n^t_{a,l} > n^\star_{a,l}, \tag{4.6}$$

$$\text{where } n^\star_{a,l} = \begin{cases} \lceil \beta_\delta^{-1}(\frac{\Delta_{a,l}}{4}) \rceil & \text{if } l \notin \mathcal{C}_a, \\ \lceil \beta_\delta^{-1}(\frac{\Delta_a}{4}) \rceil & \text{otherwise,} \end{cases}$$

with $\Delta_a = \min_{l\in[A]\setminus\mathcal{C}_a} \Delta_{a,l}$.

Note that the required number of samples is inversely proportional to the gaps between the means of agents in different classes. Having enough samples and knowing that the true means fall within the confidence bounds, we can show that the class-estimation rule $d^t_{a,\delta}(l) \leq 0$ indicates the membership of l in $\mathcal{C}_a$.

Lemma 12 (Class membership rule)**.** *Under $E^t_a \wedge G^t_a$ and $\forall l \in [A]$ and at time t: $d^t_{a,\delta}(l) > 0 \iff l \in [A]\setminus\mathcal{C}_a$.*

Proof. From Lemma 14, we directly have one implication. For the other one, if $l \notin \mathcal{C}_a$ because G^t_a holds, we have $\forall l \in [A]$, $n^t_{a,l} \geq n^\star_{a,l}$, therefore we can apply Lemma 13 and we directly have $d^t_{a,\delta}(l) > 0$. □

Using the above lemma, we obtain the following result for the time complexity of class estimation.

Theorem 10 (ColME class estimation time complexity)**.** *For any $\delta \in (0,1)$, employing Restricted-Round-Robin query strategy, we have:*

$$\mathbb{P}\left(\exists t > \zeta_a : \mathcal{C}^t_a \neq \mathcal{C}_a\right) \leq \frac{\delta}{8}, \quad \text{with } \zeta_a = n^\star_{a,a} + A - 1 - \sum_{l\in[A]\setminus\mathcal{C}_a} \mathbb{1}_{\{n^\star_{a,a} > n^\star_{a,l}+A-1\}}. \tag{4.7}$$

Proof. From Lemma 15 and Lemma 12, we deduce that if E_a holds and knowing that $\mathcal{C}^t_a = \{l \in [A] : d^t_{a,\delta}(l) \leq 0\}$ then $\forall t > \zeta_a$, $\mathcal{C}_a = \mathcal{C}^t_a$. Hence, $\mathbb{P}\left(\forall t > \zeta_a, \mathcal{C}_a = \mathcal{C}^t_a\right) \geq \mathbb{P}(E_a) \geq 1 - \delta/8$ using Lemma 11. □

Let us first remark that

$$d^t_{a,\delta}(l) = |\bar{x}^t_{a,a} - \bar{x}^t_{a,l}| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}),$$
$$= |(\bar{x}^t_{a,a} - \mu_a) - (\bar{x}^t_{a,l} - \mu_l) + (\mu_a - \mu_l)| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}).$$

As a consequence we have

$$d^t_{a,\delta}(l) \leq \Delta_{a,l} + |\bar{x}^t_{a,a} - \mu_a| + |\bar{x}^t_{a,l} - \mu_l| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}). \tag{4.8}$$

$$d^t_{a,\delta}(l) \geq \Delta_{a,l} - |\bar{x}^t_{a,a} - \mu_a| - |\bar{x}^t_{a,l} - \mu_l| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}). \tag{4.9}$$

Lemma 13. *Under E_a, $\forall l \in [A]$, if $l \notin \mathcal{C}_a$ then $\forall n^t_{a,l} \geq n^\star_{a,l} = \lceil \beta^{-1}_\delta(\frac{\Delta_{a,l}}{4}) \rceil$ we have $d^t_{a,\delta}(l) > 0$.*

Proof. Under E_a, we have $|\bar{x}^t_{a,l} - \mu_l| \leq \beta_\delta(n^t_{a,l})$ and $|\bar{x}^t_{a,a} - \mu_a| \leq \beta_\delta(n^t_{a,a})$. Since $n^t_{a,a} \geq n^t_{a,l}$, we also have $\beta_\delta(n^t_{a,a}) \leq \beta_\delta(n^t_{a,l})$. Hence, using (4.9), $d^t_{a,\delta}(l) \geq \Delta_{a,l} - 2\beta_\delta(n^t_{a,a}) - 2\beta_\delta(n^t_{a,l}) \geq \Delta_{a,l} - 4\beta_\delta(n^t_{a,l})$. If $l \in \mathcal{C}_a$ then $\Delta_{a,l} = 0$ and since $\beta_\delta(n^t_{a,l}) > 0$ we cannot ensure that $\Delta_{a,l} - 4\beta_\delta(n^t_{a,l}) > 0$. If $l \notin \mathcal{C}_a$ then to ensure that $d^t_{a,\delta}(l) \geq \Delta_{a,l} - 4\beta_\delta(n^t_{a,l}) > 0$, we need that $\frac{\Delta_{a,l}}{4} > \beta_\delta(n^t_{a,l})$ and hence $n^\star_{a,l} = \lceil \beta^{-1}_\delta(\frac{\Delta_{a,l}}{4}) \rceil$.[2] □

Lemma 14. *Under E_a, $\forall l \in [A]$, $\forall t \in \mathbb{N}$, if $l \in \mathcal{C}_a$ then $d^t_{a,\delta}(l) \leq 0$.*

Proof. Again, recall that under E^t_a, we have $|\bar{x}^t_{a,l} - \mu_l| \leq \beta_\delta(n^t_{a,l})$ and $|\bar{x}^t_{a,a} - \mu_a| \leq \beta_\delta(n^t_{a,a})$. Hence, using (4.8), $d^t_{a,\delta}(l) \leq \Delta_{a,l} + \beta_\delta(n^t_{a,a}) + \beta_\delta(n^t_{a,l}) - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}) = \Delta_{a,l}$. If $l \in \mathcal{C}_a$ then $\Delta_{a,l} = 0$ and thus $d^t_{a,\delta}(l) \leq 0$.

□

Lemma 15. *Under E_a, and using Restricted-Round-Robin algorithm, G^t_a holds when $t > \zeta_a$ where*

$$\zeta_a = n^\star_{a,a} - 1 + A - \sum_{l \in [A] \setminus \mathcal{C}_a} \mathbb{1}_{\{n^\star_{a,a} > n^\star_{a,l} - 1 + A\}}.$$

Proof. According to Algorithm 9, $\mathcal{C}^0_a = [A]$ and an agent is eliminated from set $\mathcal{C}^t_a$ at time t if $d^t_{a,\delta}(l) = |\bar{x}^t_{a,l} - \bar{x}^t_{a,a}| - \beta_\delta(n^t_{a,a}) - \beta_\delta(n^t_{a,l}) > 0$. According to Lemma 13, the time required to eliminate agent l from the class $\mathcal{C}_a$ is at least $n^\star_{a,l}$. If agent l is queried at time $n^\star_{a,l} - 1$, then using Restricted-Round-Robin (or round robin), we are sure that it will be removed from $\mathcal{C}^t_a$ for all t larger than $n^\star_{a,l} - 1 + A$.

Let us consider h being an agent such that $n^\star_{a,h} = \max_{l \in [A] \setminus \mathcal{C}_a} n^\star_{a,l}$. By definition, $\Delta_{a,h} = \min_{l \in [A] \setminus \mathcal{C}_a} \Delta_{a,l}$ and $n^\star_{a,h}$ can be denoted by $n^\star_{a,a}$.

In the case of round robin, we are sure that G^t_a will be true when $t \geq n^\star_{a,a} - 1 + A$. But using Restricted-Round-Robin, since the loop ignores agents not in $\mathcal{C}^t_a$, we have that G^t_a holds when $t > \zeta_a$ where

$$\zeta_a = n^\star_{a,a} - 1 + A - \sum_{l \in [A] \setminus \mathcal{C}_a} \mathbb{1}_{\{n^\star_{a,a} > n^\star_{a,l} - 1 + A\}}.$$

□

In the worst case, the class estimation time complexity ζ_a for agent a is equal to the number of samples required to distinguish agent a from the one who has smallest nonzero gap Δ_a to a, plus $A - 1$ since all others agents that are not in $\mathcal{C}_a$ could require the same number of samples. The last term in (4.7) accounts for agents that require less samples and had thus been eliminated before, which reflects the gain of using Restricted-Round-Robin query strategy over Round-Robin. When we have enough samples (at least ζ_a), Theorem 10 guarantees that we correctly learn the class ($\mathcal{C}_a = \mathcal{C}^t_a$) with high probability. We build upon this result to quantify the mean estimation time complexity of our approach.

[2] In extremely rare cases, the expression $\beta^{-1}_\delta(\frac{\Delta_{a,l}}{4})$ could be an integer and we should add 1 to get a strict inequality. But for conciseness of the expression, we omit the $+1$ in the definition of $n^\star_{a,l}$.

Theorem 11 (ColME mean estimation time complexity)**.** *Given the risk parameter δ, using the Restricted-Round-Robin query strategy and simple weighting, the mean estimator μ_a^t of agent a is $(\varepsilon, \frac{\delta}{4})$-convergent, that is:*

$$\mathbb{P}\left(\forall t > \tau_a : |\mu_a^t - \mu_a| \leq \varepsilon\right) > 1 - \frac{\delta}{4}, \quad \text{with } \tau_a = \max(\zeta_a, \frac{\lceil \beta_\delta^{-1}(\varepsilon) \rceil}{|\mathcal{C}_a|} + \frac{|\mathcal{C}_a| - 1}{2}). \tag{4.10}$$

Proof. Let us assume that at time t we have $\mathcal{C}_a^t = \mathcal{C}_a$. Therefore

$$\mu_a^t = \sum_{l \in \mathcal{C}_a} \bar{x}_{a,l}^t \alpha_{a,l}^t = \frac{\sum_{l \in \mathcal{C}_a} \bar{x}_{a,l}^t n_{a,l}^t}{\sum_{l \in \mathcal{C}_a} n_{a,l}^t}.$$

Remark that $\sum_{l \in \mathcal{C}_a} \bar{x}_{a,l}^t n_{a,l}^t$ is the sum of all $n_{a,l}^t$ samples received by all agents l in $\mathcal{C}_a$. In other words, μ_a^t is the estimation of μ_a with $\sum_{l \in \mathcal{C}_a} n_{a,l}^t$ examples. Hence in order to have $|\mu_a^t - \mu_a| \leq \varepsilon$ when E_a holds, we should have $\beta(\sum_{l \in \mathcal{C}_a} n_{a,l}^t) \leq \varepsilon$. Let us see at what time denoted by $n_{\varepsilon,a}$ we have $\lceil \beta^{-1}(\varepsilon) \rceil = \sum_{l \in \mathcal{C}_a} n_{a,l}^t$. With Algorithm 13 using Restricted-Round-Robin, we know that when $\mathcal{C}_a^t = \mathcal{C}_a$, then only members of $\mathcal{C}_a$ are queried. Therefore,

$$\lceil \beta^{-1}(\varepsilon) \rceil = n_{\varepsilon,a} + (n_{\varepsilon,a} - 1) + \cdots + (n_{\varepsilon,a} - |\mathcal{C}_a| + 1) = |\mathcal{C}_a| n_{\varepsilon,a} - \frac{|\mathcal{C}_a| - 1}{2} |\mathcal{C}_a|,$$

$$n_{\varepsilon,a} = \frac{\lceil \beta^{-1}(\varepsilon) \rceil}{|\mathcal{C}_a|} + \frac{|\mathcal{C}_a| - 1}{2}.$$

As a summary, if E_a holds, then we have $\forall t \geq n_{\varepsilon,a}$, $\mathcal{C}_a^t = \mathcal{C}_a$ implies that $|\mu_a^t - \mu_a| \leq \varepsilon$. Now, following Theorem 10, we have $\mathbb{P}\left(\exists t > \zeta_a : \mathcal{C}_a^t \neq \mathcal{C}_a\right) \leq \frac{\delta}{8}$. Since $\tau_a = \max(\zeta_a, n_{\varepsilon,a}) \geq \zeta_a$, then $\mathbb{P}\left(\exists t > \tau_a : |\mu_a^t - \mu_a| > \varepsilon\right) \leq \frac{\delta}{8} + \mathbb{P}\left(\bar{E}_a\right) = \frac{\delta}{4}$. □

Several comments are in order. First, recall that collaboration induces a bias in mean estimation before class estimation time. Because the problem structure is unknown, any collaborative algorithm that aggregate observations from different agents will suffer from this bias, but the bias vanishes as soon as the class is estimated and we outperform local estimation.

Then, to interpret the guarantees provided by Theorem 11, it is useful to compare them with the local estimation baseline, which has time complexity $\lceil \beta_\delta^{-1}(\varepsilon) \rceil$. Inspecting (4.10), we see that our approach is faster than local estimation by a factor of order $|\mathcal{C}_a|$ as long as the time ζ_a needed to correctly identity the class $\mathcal{C}_a$ is smaller than $\lceil \beta_\delta^{-1}(\varepsilon) \rceil$, that is:

$$\varepsilon < \beta_\delta(n_{a,a}^\star + A - 1 - \sum_{l \in [A] \setminus \{\mathcal{C}_a\}} \mathbb{1}_{\{n_{a,a}^\star > n_{a,l}^\star + A - 1\}}). \tag{4.11}$$

This condition relates the desired precision of the solution ε to the problem structure captured by the gaps $\{\Delta_{a,l}\}_{l \in [A]}$ between the true means through $\{n_{a,l}^\star\}_{l \in [A]}$ (see Definition 25). We will see in our experiments that our theory predicts quite well whether an agent empirically benefits from collaboration.

Remarkably, when (4.11) is satisfied (i.e., for large enough gaps or small enough ε), the speed-up achieved by our approach is nearly optimal. Indeed, the time complexity of the oracle weighting baseline introduced in Section 4.4.3 is precisely $\frac{\lceil \beta_\delta^{-1}(\varepsilon) \rceil}{|\mathcal{C}_a|} + \frac{|\mathcal{C}_a| - 1}{2}$. In a full information setting where agent a would know $\mathcal{C}_a$ *and* would also have access

to up-to-date samples from all agents at each step, the time complexity would be $\frac{\lceil \beta_\delta^{-1}(\varepsilon) \rceil}{|\mathcal{C}_a|}$.

4.6 Numerical Results

In this section, we provide numerical experiments on synthetic data to illustrate our theoretical results and assess the practical performance of our proposed algorithms.

4.6.1 Experimental Setting

We consider $A = 200$ agents, a time horizon of 2500 steps and a risk parameter $\delta = 0.001$. The personal distributions of agents are all Gaussian with variance $\sigma^2 = 0.25$ and belong to one of 3 classes with means 0.2, 0.4 and 0.8. The class membership of each agent (and thus the value of its true mean) is chosen uniformly at random among the three classes. We thus obtain roughly balanced class sizes.

We consider several variants of our algorithm ColME: we compare query strategies Round-Robin and Restricted-Round-Robin with simple weighting, and also evaluate the use of soft and aggressive weighting schemes in the Restricted-Round-Robin case. This gives 4 variants of our algorithm: Round-Robin, Restricted-Round-Robin, Soft-Restricted-Round-Robin and Aggressive-Restricted-Round-Robin.

Regarding competing approaches, we recall that our setting is novel and we are not aware of existing algorithms addressing the same problem. We can however compare against two baseline strategies. The Local baseline corresponds to the case of no collaboration. On the other hand, the Oracle baseline represents an upper bound on the achievable performance by any collaborative algorithm as it is given as input the true class membership of each agent and thus does not need to perform class estimation.

All algorithms are compared across 20 random runs corresponding to 20 different samples. In a given run, at each time step, each agent receives the same sample for all algorithms.

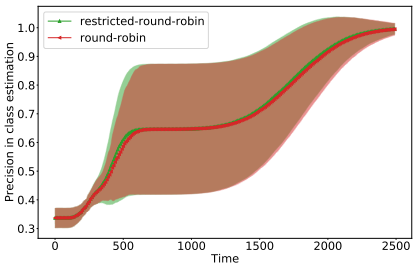
4.6.2 Class Estimation

We first focus on the performance in class estimation. In this experiment, only Round-Robin and Restricted-Round-Robin are shown since the different weighting schemes have no effect on class estimation.

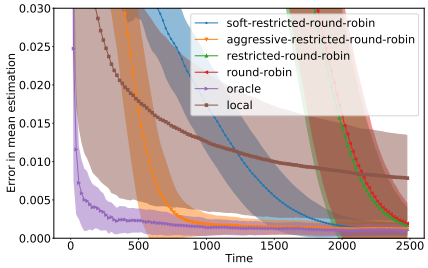
To measure how well an agent a estimates its true class $\mathcal{C}_a$ with its heuristic class $\mathcal{C}_a^t$ at a given time t, we consider the precision computed as follows:

$$\text{precision}_{\mathcal{C}_a^t} = \frac{|\mathcal{C}_a^t \cap \mathcal{C}_a|}{|\mathcal{C}_a^t|}. \tag{4.12}$$

We compute the average and standard deviation of (4.12) across runs, and then average these over all agents. Figure 4.1(a) shows how the precision of class estimation varies across time as agents progressively remove others from their heuristic class and eventually identify their true class. We can see that the classes 0.2 and 0.8 are separated very early, quickly followed by 0.4 and 0.8 and finally, after sufficiently many samples have been collected, the pair with the smallest gap (0.4 and 0.2). We also observe that Round-Robin and Restricted-Round-Robin only differ slightly in the last time steps before classes are identified, as captured by Eq. (4.7).



(a) Class estimation precision over all agents



(b) Error in mean estimation over all agents

FIGURE 4.1: Results on a 3-class problem (Gaussian distributions with true means 0.2, 0.4, 0.8). Thanks to our collaborative algorithms (Soft-Restricted-Round-Robin, Aggressive-Restricted-Round-Robin, Restricted-Round-Robin, Round-Robin), agents are able to estimate their true class (Fig. 4.1(a)) and thereby obtain accurate mean estimates much more quickly than using purely local estimation (Fig. 4.1(b)).

4.6.3 Mean Estimation

We now turn to our main objective: mean estimation. The error of an agent a at time t is evaluated as the absolute difference of its mean estimate with its true mean:

$$\text{error}_a^t = |\mu_a^t - \mu_a|. \tag{4.13}$$

Similar to above, we compute the average and standard deviation of this quantity across runs, and then for each time step we report in Figure 4.1(b) the average of these quantities across all agents for the different algorithms (Soft-Restricted-Round-Robin, Aggressive-Restricted-Round-Robin, Restricted-Round-Robin, Round-Robin, Oracle, and Local).

As expected, all variants of CollME suffer from mean estimation bias in the early steps (due to optimistic class estimation). However, as the estimated class of each agent gets more precise (see Figure 4.1(a)), agents progressively eliminate this bias and eventually learn estimates with similar error and variance as the Oracle baseline. On the other hand, Local does not have estimation bias (hence achieves smaller error on average in early rounds) but exhibits much higher variance, and its average error converges very slowly towards zero. These results show the ability of our collaborative algorithms to construct highly accurate mean estimates much faster than without collaboration. We can also see that Soft-Restricted-Round-Robin and Aggressive-Restricted-Round-Robin converge much quicker to low error estimates than Restricted-Round-Robin. This shows that our proposed heuristic weighting schemes successfully reduce the relative weight given to agents that actually belong to different classes well before they are identified as such with sufficient confidence. The aggressive weighting scheme is observed to perform best in practice.

Finally, we quantitatively compare the convergence time of different algorithms with an empirical measure inspired by our theoretical PAC criterion (Definition 20). We define the *empirical convergence* time of an agent as the earliest time step where the estimation error of the agent always stays lower than some ε:

$$\text{conv}_a(\varepsilon) = \min\{\tau \in \mathbb{N} : \forall t \geq \tau, \text{error}_a^t \leq \varepsilon\}. \tag{4.14}$$

We denote by $\text{conv}(\varepsilon)$ the average of the above quantity across all runs and all agents.

Algorithm	conv(0.1)	conv(0.01)
Round-Robin	696.81 ± 514.85	1623.69 ± 705.93
Restricted-Round-Robin	685.45 ± 516.19	1601.49 ± 720.53
Soft-Restricted-Round-Robin	124.92 ± 72.83	1097.95 ± 487.76
Aggressive-Restricted-Round-Robin	82.67 ± 48.97	$\mathbf{491.43 \pm 180.30}$
Local	$\mathbf{41.11 \pm 38.77}$	1924.14 ± 600.26
Oracle	$\mathit{5.19 \pm 3.44}$	$\mathit{87.80 \pm 53.10}$

TABLE 4.1: Empirical convergence time (see Eq. 4.14) of different algorithms for a target estimation error of $\varepsilon = 0.1$ (unfavorable regime) and $\varepsilon = 0.01$ (favorable regime). We see that our approach largely outperforms the local estimation baseline in the favorable regime and remains competitive in the unfavorable regime.

Table 4.1 reports the empirical convergence time for two values of ε with standard deviations across runs. These values were chosen to reflect the two different regimes suggested by our theoretical analysis. Indeed, recall that our theory gives a criterion to predict whether our collaborative algorithms will outperform the Local baseline: this is the case when the desired accuracy of the solution ε is small enough for the given problem instance (see Eq. 4.11). For the problem considered here, Eq. (4.11) gives that Restricted-Round-Robin will outperform Local for all agents as soon as ε is smaller than 0.049. We thus choose $\varepsilon = 0.01$ as the favorable regime (where we should beat Local) and $\varepsilon = 0.1$ as the unfavorable regime. The results in Table 4.1 are consistent with our theory. All variants of our algorithms outperform Local for $\varepsilon = 0.01$, while Local is better for $\varepsilon = 0.1$ as agents can reach this precision using only their own samples faster than they can reliably estimate their class. Overall, Restricted-Round-Robin performs marginally better than Round-Robin, while Soft-Restricted-Round-Robin and Aggressive-Restricted-Round-Robin significantly outperform Round-Robin and Restricted-Round-Robin in both cases. Note that Aggressive-Restricted-Round-Robin performs almost as good as Local in the unfavorable regime. These results again show the relevance of our collaborative algorithms and heuristic weighting schemes.

4.7 Extension to Imperfect Classes

So far we have assumed that two agents are in the same class if their personal distributions have *exactly* the same mean, which can be restrictive for some use-cases. In this section, we show that we can extend the problem setup and our approach to the case where *two agents are considered to be in the same class if their means are close enough* and agents seek to *estimate the mean of their class*.

Formally, we define a new notion of similarity class parameterized by a radius η, which generalizes our previous notion introduced in Definition 21.

Definition 26. *Given $\eta > 0$, the* η-similarity class *of agent a is given by:*

$$\mathcal{C}_{\eta,a} = \{l \in [A] : \Delta_{a,l} \leq \eta\},$$

where $\Delta_{a,l} = |\mu_a - \mu_l|$ is the gap between the means of agent a and agent l.

This notion of "imperfect" similarity class allows to capture situations where *clusters* of agents have similar (but not necessarily equal) means. Such discrepancies

between the means of agents in the same class may for instance stem from the presence of local measurement bias (e.g., due to local variations in the environment, see Taghavi et al., 2016). They can also be used to model groups of agents with similar preferences, behavior, or goals, in applications like collaborative filtering (Su and Khoshgoftaar, 2009),

In this context, it is natural to slightly redefine the estimation objective. Instead of estimating its personal mean μ_a as considered so far, each agent a aims to estimate the mean of its class:

$$\mu_{\eta,a} = \frac{1}{|\mathcal{C}_{\eta,a}|} \sum_{l \in \mathcal{C}_{\eta,a}} \mu_l. \tag{4.15}$$

For instance, in the presence of (centered) local measurement bias, estimating the class mean (instead of the local mean) allows to debias the estimate.

Remark 11 (Non-separated clusters)**.** *We do not formally require that the η-similarity classes form separated clusters, in the sense that for three distinct agents $a, l, i \in [A]$ we may have simultaneously $i \in \mathcal{C}_{\eta,a}$, $i \in \mathcal{C}_{\eta,l}$ and $\mathcal{C}_{\eta,a} \neq \mathcal{C}_{\eta,l}$. This happens when $\Delta_{a,i} \leq \eta$, $\Delta_{l,i} \leq \eta$ and $\eta < \Delta_{a,l} \leq 2\eta$. In this case, the "class" of an agent simply corresponds to a ball of radius η around its mean, which potentially overlaps with others and thus violates the transitivity property of equivalence classes. For consistency with the rest of the paper and with an slight abuse of terminology, we continue to use the term "class". Although the case of separated clusters appears more natural, we note that our proposed approach will still work in the non-separated setting, in the sense that agents will correctly estimate the mean of their class as defined in Eq. 4.15.*

Based on the above, we can adapt the notion of optimistic similarity class (Definition 4.2) and the condition on the number of samples required for this optimistic class to coincide with the true class (Definition 25) by incorporating η.

Definition 27. *The* η-optimistic similarity class *from the perspective of agent a at time t is defined as:*

$$\mathcal{C}^t_{\eta,a} = \{l \in [A] : d^t_{a,\delta}(l) \leq \eta\}.$$

Definition 28. *From the perspective of agent a and at time t, event $G^t_{\eta,a}$ is defined as:*

$$G^t_{\eta,a} = \bigcap_{l \in [A]} n^t_{a,l} > n^\eta_{a,l}, \tag{4.16}$$

$$\text{where } n^\eta_{a,l} = \begin{cases} \lceil \beta_\delta^{-1}(\frac{\Delta_{a,l}-\eta}{4}) \rceil & \text{if } l \notin \mathcal{C}_a, \\ \lceil \beta_\delta^{-1}(\frac{\Delta_{\eta,a}-\eta}{4}) \rceil & \text{otherwise,} \end{cases}$$

with $\Delta_{\eta,a} = \min_{l \in [A] \setminus \mathcal{C}_{\eta,a}} \Delta_{a,l}$.

Lemma 16 (Class membership rule)**.** *Under $E^t_a \wedge G^t_{\eta,a}$ and $\forall l \in [A]$ and at time t: $d^t_{a,\delta}(l) > \eta \iff l \in [A] \setminus \mathcal{C}_{\eta,a}$.*

We can see from the above that ruling out an agent l from the optimistic class $\mathcal{C}_{\eta,a}$ requires more samples for larger η, which is expected as the size of the confidence interval needs to be smaller to make this decision reliably.

With these tools in place, we can use our collaborative mean estimation algorithm ColME (Algorithm 9) presented before, with only minor modifications: we simply need to replace the notion of optimistic similarity class by the η-version of

Definition 27, and compute the estimate $\mu^t_{\eta,a}$ at time t using a simple *class-uniform weighting scheme* $\alpha^t_{a,l} = \frac{1}{|\mathcal{C}^t_{\eta,a}|}$ to match the objective in Eq. 4.15. We refer to this algorithm as η-ColME. Note that η becomes a parameter of the algorithm, allowing to choose the desired radius for the class structure.

We can now extend the class and mean estimation complexity of η-ColME.

Theorem 12 (η-ColME class estimation time complexity)**.** *For any $\delta \in (0,1)$, employing Restricted-Round-Robin query strategy, we have:*

$$\mathbb{P}\left(\exists t > \zeta^\eta_a : \mathcal{C}^t_{\eta,a} \neq \mathcal{C}_{\eta,a}\right) \leq \frac{\delta}{8}, \quad \textit{with } \zeta^\eta_a = n^\eta_{a,a} + A - 1 - \sum_{l \in [A] \setminus \mathcal{C}_{\eta,a}} \mathbb{1}_{\{n^\eta_{a,a} > n^\eta_{a,l} + A - 1\}}. \tag{4.17}$$

The proof of Theorem 12 follows the same step as that of Theorem 10, up to replacing the 0 threshold by η. We only state the intermediate lemmas (which are adaptations of Lemmas 13-14-15) and omit the detailed proof.

Lemma 17. *Under E_a, $\forall l \in [A]$, if $l \notin \mathcal{C}_{\eta,a}$ then $\forall n^t_{a,l} \geq n^\eta_{a,l} = \lceil \beta^{-1}_\delta(\frac{\Delta_{a,l}-\eta}{4}) \rceil$ we have $d^t_{a,\delta}(l) > \eta$.*

Lemma 18. *Under E_a, $\forall l \in [A]$, $\forall t \in \mathbb{N}$, if $l \in \mathcal{C}_{\eta,a}$ then $d^t_{a,\delta}(l) \leq \eta$.*

Lemma 19. *Under E_a, and using Restricted-Round-Robin algorithm, $G^t_{\eta,a}$ holds when $t > \zeta^\eta_a$ where*

$$\zeta^\eta_a = n^\eta_{a,a} - 1 + A - \sum_{l \in [A] \setminus \mathcal{C}_{\eta,a}} \mathbb{1}_{\{n^\eta_{a,a} > n^\eta_{a,l} - 1 + A\}}.$$

Here, we detail the proof of Theorem 13 :

Theorem 13 (η-ColME mean estimation time complexity)**.** *Given the risk parameter δ, using the Restricted-Round-Robin query strategy and class-uniform weighting (while employing $\mathcal{C}_{\eta,a}$), the mean estimator μ^t_a of agent a is $(\varepsilon, \frac{\delta}{4})$-convergent, that is:*

$$\mathbb{P}\left(\forall t > \tau^\eta_a : |\mu^t_{\eta,a} - \mu_{\eta,a}| \leq \varepsilon\right) > 1 - \frac{\delta}{4}, \quad \textit{with } \tau^\eta_a = \max(\zeta^\eta_a, \beta^{-1}_\delta(\varepsilon) + |\mathcal{C}_{\eta,a}| - 1). \tag{4.18}$$

Proof. Since $t > \tau^\eta_a > \zeta^\eta_a$, at time t we have $\mathcal{C}^t_{\eta,a} = \mathcal{C}_{\eta,a}$. Therefore

$$\mu^t_a = \sum_{l \in \mathcal{C}_{\eta,a}} \bar{x}^t_{a,l} \alpha^t_{a,l} = \frac{\sum_{l \in \mathcal{C}_{\eta,a}} \bar{x}^t_{a,l}}{|\mathcal{C}_{\eta,a}|}.$$

Remark that μ^t_a is not equivalent to the average of all the samples of agents in $\mathcal{C}_{\eta,a}$: it is the average of the mean values for each agent in $\mathcal{C}_{\eta,a}$. Therefore, although some agents may have more samples than the others, all are assigned uniform weights. We would like to have $|\mu^t_{\eta,a} - \mu_{\eta,a}| \leq \varepsilon$. When E_a holds, we can rewrite this as

$$|\mu^t_{\eta,a} - \mu_{\eta,a}| = |\frac{1}{|\mathcal{C}_{\eta,a}|} \sum_{l \in \mathcal{C}_{\eta,a}} \bar{x}^t_{a,l} - \mu_l| \leq \frac{1}{|\mathcal{C}_{\eta,a}|} \sum_{l \in \mathcal{C}_{\eta,a}} |\bar{x}^t_{a,l} - \mu_l| \leq \varepsilon$$

Therefore, we need:

$$\sum_{l \in \mathcal{C}_{\eta,a}} |\bar{x}^t_{a,l} - \mu_l| \leq |\mathcal{C}_{\eta,a}| \times \varepsilon,$$

A sufficient condition for the above inequality to hold is to ensure that each term is bounded by ε:

$$\forall l \in \mathcal{C}_{\eta,a} : |\bar{x}^t_{a,l} - \mu_l| \leq \varepsilon \tag{4.19}$$

This is achieved when $\beta_\delta(n^t_{a,l}) < \varepsilon$ for all $l \in \mathcal{C}_{\eta,a}$. Since we are using Restricted-Round-Robin and also that $\mathcal{C}^t_{\eta,a} = \mathcal{C}_{\eta,a}$, the number of samples required for each agent in $\mathcal{C}_{\eta,a}$ are $n^t_{a,1}$, $n^t_{a,1}-1$, $n^t_{a,1}-2$, $\ldots$, $n^t_{a,1}-|\mathcal{C}_{\eta,a}|+1$ where we consider the one with the maximum number of observations to have index 1 for notation simplicity (which corresponds to index a). For Eq. 4.19 to hold, it is thus sufficient to have:

$$\beta_\delta^{-1}(\varepsilon) < n^t_{a,a} - |\mathcal{C}_{\eta,a}| + 1$$

$$\beta_\delta^{-1}(\varepsilon) + |\mathcal{C}_{\eta,a}| - 1 < n^t_{a,a}$$

Therefore $\tau^\eta_a = \max(\zeta^\eta_a, \beta_\delta^{-1}(\varepsilon) + |\mathcal{C}_{\eta,a}| - 1)$.

As a summary, if E_a holds, then we have $\forall t \geq \tau^\eta_a$, $\mathcal{C}^t_{\eta,a} = \mathcal{C}_{\eta,a}$ implies that $|\mu^t_{\eta,a} - \mu_{\eta,a}| \leq \varepsilon$. Now, following Theorem 12, we have $\mathbb{P}\left(\exists t > \zeta^\eta_a : \mathcal{C}^t_{\eta,a} \neq \mathcal{C}_{\eta,a}\right) \leq \frac{\delta}{8}$. Since $\tau^\eta_a = \max(\zeta^\eta_a, n^\eta_{\varepsilon,a}) \geq \zeta^\eta_a$, then $\mathbb{P}\left(\exists t > \tau^\eta_a : |\mu^t_a - \mu_{\eta,a}| > \varepsilon\right) \leq \frac{\delta}{8} + \mathbb{P}\left(\bar{E}_a\right) = \frac{\delta}{4}$. □

The results are similar as for the "perfect" class case (Theorems 10-11) except that they involve η-dependent quantities. Note that for large enough gaps or small enough precision ε (similar to Eq. 4.11), we again achieve an optimal speed since the time complexity of an oracle weighting baseline that would know the true classes beforehand is $\beta_\delta^{-1}(\varepsilon) + |\mathcal{C}_{\eta,a}| - 1$.

Chapter 5

Conclusion & Future Work

In this chapter we summarize our contributions and also discuss the future work and possible extensions of the work presented.

5.1 Conclusion

In this work, we presented ways to leverage structure for online and collaborative learning problems.

In the first setting which revolves around model-based reinforcement learning problems, we introduced a similarity measure of state-action pairs, which induces a notion of equivalence of profile distributions in the state-action space of a Markov Decision Process. In the case of a known equivalence structure, we presented confidence sets incorporating such knowledge that are provably tighter than their corresponding counterparts ignoring equivalence structure. In the case of an unknown equivalence structure, we presented an algorithm, based on confidence bounds, that seeks to estimate an empirical equivalence structure for the MDP. In order to illustrate the usefulness of our developments, we further introduced C-UCRL, which is a natural modification of UCRL2 using the presented confidence sets. We showed that when the equivalence structure is known to the learner, C-UCRL attains a regret smaller than that of UCRL2 by a factor of $\sqrt{SA/C}$ in communicating MDPs, where C denotes the number of classes. In the case of an unknown equivalence structure, we showed through numerical experiments that in ergodic environments, C-UCRL outperforms UCRL2 significantly. The regret analysis in this case is much more complicated, and we leave it for future work.

In the second setting, we addressed the challenging problem of online personalized mean estimation in a network of learners with heterogeneous data distributions. We presented collaborative online algorithms where each agent learns the set (class) of agents who shares the same objective and uses this knowledge to speed up the estimation of its personalized mean. The collaborative learning approaches presented work better than learning individually on this problem. We provided PAC-style guarantees for the class and mean estimation time complexity of our algorithms, which improve upon the case where there is no collaboration. In addition, we introduced a number of sample weighting mechanisms to decrease the bias in the early rounds of learning, whose benefit is shown empirically.

5.2 Future Work

There exists a number of possible extensions for the first setting we explored in order to go towards more decentralized cases. A simple extension of Chapter 3 is to consider a multi-agent setting for our problem where agents collaborate to solve a

reinforcement learning problem by sharing their observations of the state-action pairs. As first step, let us assume that each agent has access to the observations of the other agents. In order to learn the policy for the problem, it is important to coordinate the policies so that not every agent samples the same state-action pair and learn as fast as possible. This should be done while minimizing the communication between agents. As the next direction, the assumption that all agents have access to each other's observations need to be relaxed and the observations can be synchronized every now and then. Moreover, the agents can have personalized problems while sharing some similarity with others. In this case, it would be important what to communicate with other agents as well. As a more concrete improvement of the main setting in Chapter 3, we believe that our confidence sets can be combined with model-based algorithms for the discounted setup, which we expect to yield improved performance in terms of sample complexity both in theory and practice.

In the second setting, our work initiates the study of online, collaborative and personalized estimation and learning problems, which we believe to be a promising area for future work. First, we would like to provide a theoretical analysis of the soft and aggressive weighting schemes, which is challenging as the effect of these heuristics occurs before the class has been correctly identified. Second, we can extend the work to a more realistic scenario where the underlying communication graph is not complete. In this case we can take into account the similarities between the neighbors of neighbors as well. Finally, the problem could be extended to cases where each agent aims to solve a personalized machine learning task (Vanhaesebrouck, Bellet, and Tommasi, 2017) based on the data it receives online. In that case, a structure in the distribution conditioned by the outputs of the learned models can be inferred, introducing an interesting exploration-exploitation dilemma in the learning task. For instance, the task could be regression and if the used loss function is mean squared error, then this setting can be used for solving policy gradient reinforcement learning tasks as well (Asadi, 2016).

Bibliography

Abbasi-Yadkori, Y., D. Pál, and Cs. Szepesvári (2011). "Improved algorithms for linear stochastic bandits". In: *Proc. of NIPS*, pp. 2312–2320.

Abel, D., D. Hershkowitz, and M. L. Littman (2016). "Near Optimal Behavior via Approximate State Abstraction". In: *Proc. of ICML*, pp. 2915–2923.

Adi, Erwin et al. (2020). "Machine learning and data analytics for the IoT". In: *Neural Computing and Applications* 32.20, pp. 16205–16233.

Anand, Ankit et al. (2015). "ASAP-UCT: Abstraction of State-Action Pairs in UCT". In: *Proc. of IJCAI*, pp. 1509–1515.

Asadi, Mahsa (2016). "Distributed Multitask Learning". MA thesis. Shiraz University.

Asadi, Mahsa et al. (2019). "Model-Based Reinforcement Learning Exploiting State-Action Equivalence". In: *Asian Conference on Machine Learning*. PMLR, pp. 204–219.

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer (2002). "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2, pp. 235–256.

Bartlett, P. L. and A. Tewari (2009). "REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs". In: *Proc. of UAI*, pp. 35–42.

Bertsekas, Dimitri P and John N Tsitsiklis (2002). *Introduction to probability*. Vol. 1. Athena Scientific Belmont, MA.

Boucheron, Stéphane, Gábor Lugosi, and Olivier Bousquet (2003). "Concentration inequalities". In: *Summer school on machine learning*. Springer, pp. 208–240.

Boucheron, Stéphane, Gábor Lugosi, and Pascal Massart (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

Boursier, Etienne and Vianney Perchet (2019). "SIC - MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits". In: *NeurIPS*.

Boursier, Etienne et al. (2019). "A practical algorithm for multiplayer bandits when arm means vary among players". In: *arXiv preprint arXiv:1902.01239*.

Brunskill, E. and L. Li (2013). "Sample Complexity of Multi-task Reinforcement Learning". In: *Proc. of UAI*, p. 122.

Bubeck, Sébastien, Rémi Munos, and Gilles Stoltz (2009). "Pure exploration in multi-armed bandits problems". In: *International conference on Algorithmic learning theory*. Springer, pp. 23–37.

Cesa-Bianchi, Nicolo and Gábor Lugosi (2006). *Prediction, learning, and games*. Cambridge university press.

Dann, C., T. Lattimore, and E. Brunskill (2017). "Unifying PAC and Regret: Uniform PAC Bounds for Episodic Reinforcement Learning". In: *Proc. of NIPS*, pp. 5711–5721.

Dean, T., R. Givan, and S. Leach (1997). "Model reduction techniques for computing approximately optimal solutions for Markov decision processes". In: *Proc. of UCAI*, pp. 124–131.

Diuk, C., L. Li, and B. R. Leffler (2009). "The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning". In: *Proc. of ICML*, pp. 249–256.

Fallah, Alireza, Aryan Mokhtari, and Asuman Ozdaglar (2020). "Personalized federated learning: A meta-learning approach". In: *NeurIPS*.

Ferns, N., P. Panangaden, and D. Precup (2004). "Metrics for finite Markov decision processes". In: *Proc. of UAI*, pp. 162–169.

— (2011). "Bisimulation metrics for continuous Markov decision processes". In: *SIAM Journal on Computing* 40.6, pp. 1662–1714.

Gheshlaghi Azar, M., I. Osband, and R. Munos (2017). "Minimax Regret Bounds for Reinforcement Learning". In: *Proc. of ICML*, pp. 263–272.

Givan, R., T. Dean, and M. Greig (2003). "Equivalence notions and model minimization in Markov decision processes". In: *Artificial Intelligence* 147.1-2, pp. 163–223.

Hallak, A., D. Di-Castro, and S. Mannor (2013). "Model selection in Markovian processes". In: *Proc. of ACM SIGKDD*, pp. 374–382.

Hanzely, Filip et al. (2020). "Lower bounds and optimal algorithms for personalized federated learning". In: *Proc. of NeurIPS*.

He, Jiafan, Dongruo Zhou, and Quanquan Gu (2021). "Nearly minimax optimal reinforcement learning for discounted MDPs". In: *Advances in Neural Information Processing Systems* 34, pp. 22288–22300.

Hillel, Eshcar et al. (2013). "Distributed Exploration in Multi-Armed Bandits". In: *NIPS*.

Hoeffding, W. (1963). "Probability inequalities for sums of bounded random variables". In: *Journal of the American Statistical Association* 58.301, pp. 13–30.

Jaksch, T., R. Ortner, and P. Auer (2010). "Near-optimal regret bounds for reinforcement learning". In: *JMLR* 11, pp. 1563–1600.

Kairouz, Peter et al. (2021). "Advances and Open Problems in Federated Learning". In: *Foundations and Trendső in Machine Learning* 14.12, pp. 1–210.

Kakade, S. (2003). "On the sample complexity of reinforcement learning". PhD thesis. University of London London, England.

Karpov, Nikolai and Qin Zhang (2022). "Collaborative Best Arm Identification with Limited Communication on Non-IID Data". In: *arXiv preprint arXiv:2207.08015*.

Khaleghi, A. et al. (2016). "Consistent algorithms for clustering time series". In: *JMLR* 17.1, pp. 94–125.

Kocák, Tomáš and Aurélien Garivier (2020). "Best Arm Identification in Spectral Bandits". In: *arXiv preprint arXiv:2005.09841*.

— (2021). "Epsilon Best Arm Identification in Spectral Bandits". In: *IJCAI*.

Kok, Jelle R and Nikos Vlassis (2006). "Collaborative multiagent reinforcement learning by payoff propagation". In: *Journal of Machine Learning Research* 7, pp. 1789–1828.

Landgren, Peter, Vaibhav Srivastava, and Naomi Ehrich Leonard (2021). "Distributed cooperative decision making in multi-agent multi-armed bandits". In: *Automatica* 125, p. 109445.

Lattimore, Tor and Csaba Szepesvári (2020). *Bandit algorithms*. Cambridge University Press.

Leffler, B. R., M. L. Littman, and T. Edmunds (2007). "Efficient reinforcement learning with relocatable action models". In: *Proc. of AAAI*. Vol. 7, pp. 572–577.

Li, L., T. J. Walsh, and M. L. Littman (2006). "Towards a unified theory of state abstraction for MDPs". In: *ISAIM*.

Liu, Shuang and Hao Su (2020). "Regret bounds for discounted mdps". In: *arXiv preprint arXiv:2002.05138*.

Maillard, Odalric-Ambrym (2019). "Mathematics of statistical sequential decision making". PhD thesis. Université de Lille, Sciences et Technologies. Chap. 2, p. 34.

Mandel, T. et al. (2016). "Efficient Bayesian Clustering for Reinforcement Learning." In: *Proc. of IJCAI*, pp. 1830–1838.

Marfoq, Othmane et al. (2021). "Federated Multi-Task Learning under a Mixture of Distributions". In: *Proc. of NeurIPS*.

Martínez-Rubio, David, Varun Kanade, and Patrick Rebeschini (2019). "Decentralized cooperative stochastic bandits". In:

Mateo, Fernando et al. (2013). "Machine learning methods to forecast temperature in buildings". In: *Expert Systems with Applications* 40.4, pp. 1061–1068.

Ortner, R. (2013). "Adaptive aggregation for reinforcement learning in average reward Markov decision processes". In: *Annals of Operations Research* 208.1, pp. 321–336.

Ortner, R., O.-A. Maillard, and D. Ryabko (2014). "Selecting near-optimal approximate state representations in reinforcement learning". In: *Proc. of ALT*, pp. 140–154.

Peña, V. H., T. L. Lai, and Q.-M. Shao (2008). *Self-normalized processes: Limit theory and Statistical Applications*. Springer Science & Business Media.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Ravindran, B. and A. G. Barto (2004). "Approximate homomorphisms: A framework for non-exact minimization in Markov decision processes". In: *Proc. of KBCS*.

Roberts, Tim S (2004). *Online collaborative learning: Theory and practice*. IGI Global.

Réda, Clémence, Sattar Vakili, and Emilie Kaufmann (2022). "Near-Optimal Collaborative Learning in Bandits". In: *arXiv preprint arXiv:2206.00121*.

Sankararaman, Abishek, Ayalvadi Ganesh, and Sanjay Shakkottai (2019). "Social learning in multi agent multi armed bandits". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.3, pp. 1–35.

Sattler, Felix, Klaus-Robert Müller, and Wojciech Samek (2020). "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints". In: *IEEE Transactions on Neural Networks and Learning Systems*.

Shi, Chengshuai and Cong Shen (2021). "Federated Multi-Armed Bandits". In: *AAAI*.

Shi, Chengshuai, Cong Shen, and Jing Yang (2021). "Federated Multi-armed Bandits with Personalization". In: *AISTATS*.

Smith, Virginia et al. (2017). "Federated Multi-Task Learning". In: *NIPS*.

Strehl, Alexander L and Michael L Littman (2008). "An analysis of model-based interval estimation for Markov decision processes". In: *Journal of Computer and System Sciences* 74.8, pp. 1309–1331.

Su, Xiaoyuan and Taghi M. Khoshgoftaar (2009). "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence*.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction*. MIT Press Cambridge.

Szepesvári, Csaba (2010). "Algorithms for reinforcement learning". In: *Synthesis lectures on artificial intelligence and machine learning* 4.1, pp. 1–103.

Taghavi, Ehsan et al. (2016). "A practical bias estimation algorithm for multisensor-multitarget tracking". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.1, pp. 2–19.

Tao, Chao, Qin Zhang, and Yuan Zhou (2019). "Collaborative Learning with Limited Interaction: Tight Bounds for Distributed Exploration in Multi-armed Bandits". In: *FOCS*.

Vanhaesebrouck, Paul, Aurélien Bellet, and Marc Tommasi (2017). "Decentralized collaborative learning of personalized models over networks". In:

Wang, Po-An et al. (2020a). "Optimal algorithms for multiplayer multi-armed bandits". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 4120–4129.

Wang, Yuanhao et al. (2020b). "Distributed Bandit Learning: Near-Optimal Regret with Efficient Communication". In: *ICLR*.

Wasserman, Larry (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Weissman, T. et al. (2003). "Inequalities for the L1 deviation of the empirical distribution". In: *Hewlett-Packard Labs, Technical Report*.