Intelligenza Artificiale xx (20xx) x–xx
DOI 10.3233/IA-150080
IOS Press

Sparse multi-task reinforcement learning

Daniele Calandriello[a,*], Alessandro Lazaric[a] and Marcello Restelli[b]
[a]*Team SequeL, INRIA Lille – Nord Europe, France*
[b]*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy*

Abstract. In multi-task reinforcement learning (MTRL), the objective is to simultaneously learn multiple tasks and exploit their similarity to improve the performance w.r.t. single-task learning. In this paper we investigate the case when all the tasks can be accurately represented in a linear approximation space using the same small subset of the original (large) set of features. This is equivalent to assuming that the weight vectors of the task value functions are *jointly sparse*, i.e., the set of their non-zero components is small and it is shared across tasks. Building on existing results in multi-task regression, we develop two multi-task extensions of the fitted Q-iteration algorithm. While the first algorithm assumes that the tasks are jointly sparse in the given representation, the second one learns a transformation of the features in the attempt of finding a more sparse representation. For both algorithms we provide a sample complexity analysis and numerical simulations.

Keywords: Reinforcement learning, multi-task learning, sparsity, feature learning, theoretical guarantees

1. Introduction

Reinforcement learning (RL) and approximate dynamic programming (ADP) [2, 26] are effective approaches to solve the problem of decision-making under uncertainty. Nonetheless, they may fail in domains where a relatively small amount of samples can be collected (e.g., in robotics where samples are expensive or in applications where human interaction is required, such as in automated rehabilitation). Fortunately, the lack of samples can be compensated by leveraging on the presence of multiple related tasks (e.g., different users). In this scenario, usually referred to as multi-task reinforcement learning (MTRL), the objective is to simultaneously solve multiple tasks and exploit their similarity to improve the performance w.r.t. single-task learning (we refer to [28] and [16] for a comprehensive review of the more general setting of transfer RL). In this setting, many approaches have been proposed, which mostly differ for the notion of similarity leveraged in the multi-task learning process. In [30] the transition and reward kernels of all the tasks are assumed to be generated from a common distribution and samples from different tasks are used to estimate the generative distribution and, thus, improving the inference on each task. A similar model, but for value functions, is proposed in [17], where the parameters of all the different value functions are assumed to be drawn from a common distribution. In [25] different shaping function approaches for Q-table initialization are considered and empirically evaluated, while a model-based approach that estimates statistical information on the distribution of the Q-values is proposed in [27]. Similarity at the level of the MDPs is also exploited in [18], where samples are transferred from source to target tasks. Multi-task reinforcement learning approaches have been also applied in partially observable environments [19].

In this paper we investigate the case when all the tasks can be accurately represented in a linear approximation space using the same small subset of the original (large) set of features. This is equivalent to assuming that the weight vectors of the task value functions are *jointly sparse*, i.e., the set of their non-zero components

*Corresponding author: Daniele Calandriello, Team SequeL, INRIA Lille – Nord Europe, France. E-mail: daniele.calandriello@inria.fr.

is small and it is shared across tasks. We can illustrate the concept of shared sparsity using the blackjack card game. The player can rely on a very large number of features such as: value and color of the cards in the player's hand, value and color of the cards on the table and/or already discarded, different scoring functions for the player's hand (e.g., sum of the values of the cards) and so on. The more the features, the more likely it is that the corresponding feature space could accurately represent the optimal value function. Nonetheless, depending on the rules of the game (i.e., the reward and dynamics), a very limited subset of features actually contribute to the value of a state and we expect the optimal value function to display a high level of sparsity. Furthermore, if we consider multiple tasks differing for the behavior of the dealer (e.g., the value at which she stays) or slightly different rule sets, we may expect such sparsity to be shared across tasks. For instance, if the game uses an infinite number of decks, features based on the history of the cards played in previous hands have no impact on the optimal policy for any task and the corresponding value functions are all jointly sparse in this representation.

The main limitation of this formulation is that it forces all tasks to be jointly sparse, and the set of useful features is not know in advance. Therefore whenever a new task is added, the sparisty scenario may be significantly affected. On the one hand, adding tasks may improve the sample complexity by reducing the overall variance. On the other hand, if the new task requires features that were useless up to that point (i.e., it is not jointly sparse), then it would not help identifying the set of useful features, and in addition the set of useful features would grow larger. In the second part of the paper we will introduce a generalization of the concept of joint sparsity to tackle this problem. The sparsity of the linear weights in the solution is highly dependant on the particular feature space chosen for the problem. We will try to learn a transformation of the features in order to build a new feature space where the solution has its sparsest representation. Intuitively, this will correspond to generalizing the assumption of correlation through shared sparsity (shared support) to the more general assumption of linear correlation between tasks, and provide us more flexibility in choosing which tasks can be added to the problem. This concept will be explored in more detail in the remarks of Sections 4 and 5 after we will have formalized the notation and introduced the main results.

In this paper we first introduce the notion of sparse MDPs in Section 3. Then we build on existing results in multi-task regression [1, 20] to develop two multi-task extensions of the fitted Q-iteration algorithm. While the first algorithm (Section 4) assumes that the tasks are jointly sparse in the given representation, the second algorithm (Section 5) performs a transformation of the given features in the attempt of finding a more sparse representation. For both algorithms we provide a sample complexity analysis and numerical simulations both in a continuous chain–walk domain and in the blackjack game (Section 6).[1]

2. Preliminaries

2.1. Multi-task reinforcement learning (MTRL)

A Markov decision process (MDP) is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, R, P, \gamma)$, where the state space $\mathcal{X}$ is a bounded closed subset of the Euclidean space, the action space $\mathcal{A}$ is finite (i.e., $|\mathcal{A}| < \infty$), $R : \mathcal{X} \times \mathcal{A} \to [0, 1]$ is the reward of a state-action pair, $P : \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$ is the transition distribution over the states achieved by taking an action in a given state, and $\gamma \in (0, 1)$ is a discount factor. A deterministic policy $\pi : \mathcal{X} \to \mathcal{A}$ is a mapping from states to actions. We denote by $\mathcal{B}(\mathcal{X} \times \mathcal{A}; b)$ the set of measurable state-action functions $f : \mathcal{X} \times \mathcal{A} \to [-b; b]$ absolutely bounded by b. Solving an MDP corresponds to computing the optimal action-value function $Q^* \in \mathcal{B}(\mathcal{X} \times \mathcal{A}; Q_{\max} = 1/(1-\gamma))$, defined as the largest expected sum of discounted rewards that can be collected in the MDP and fixed point of the optimal Bellman operator $\mathcal{T} : \mathcal{B}(\mathcal{X} \times \mathcal{A}; Q_{\max}) \to \mathcal{B}(\mathcal{X} \times \mathcal{A}; Q_{\max})$ defined as $\mathcal{T}Q(x, a) = R(x, a) + \gamma \sum_y P(y|x, a) \max_{a'} Q(y, a')$. The optimal policy is finally obtained as the greedy policy w.r.t. the optimal value function as $\pi^*(x) = \arg\max_{a \in \mathcal{A}} Q^*(x, a)$. In this paper we study the multi-task reinforcement learning (MTRL) setting where the objective is to solve T tasks, defined as $\mathcal{M}_t = (\mathcal{X}, \mathcal{A}, P_t, R_t, \gamma_t)$ with $t \in [T] = \{1, \dots, T\}$, with the same state-action space, but different dynamics P_t and goals R_t. The objective of MTRL is to exploit possible relationships between tasks to improve the performance w.r.t. single-task learning. In particular, we choose linear fitted Q-iteration as the single-task baseline and we propose multi-task extensions tailored to exploit the sparsity in the structure of the tasks.

[1] We refer the reader to the technical report [5] for more details about the theoretical results, which are mostly based on existing results in multi-task regression and so they were omitted from this version.

Algorithm 1 Linear FQI with fixed design and fresh samples at each iteration in a multi-task setting.

input: Input sets $\left\{\mathcal{S}_t = \{x_i\}_{i=1}^{n_x}\right\}_{t=1}^{T}$, tol, K
output: W_a^K, $b_{a,t}^K$
  Initialize $w^0 \leftarrow \mathbf{0}$, $k = 0$
  do
   $k \leftarrow k + 1$
   for $a \leftarrow 1, \ldots, |\mathcal{A}|$ **do**
    for $t \leftarrow 1, \ldots, T$ **do**
     for $i \leftarrow 1, \ldots, n_x$ **do**
      Sample $r_{i,a,t}^k = R_t(x_{i,t}, a)$ and $y_{i,a,t}^k \sim P_t(\cdot|x_{i,t}, a)$
      Compute $z_{i,a,t}^k = r_{i,a,t}^k + \gamma \max_{a'} \widetilde{Q}_t^k(y_{i,a,t}^k, a')$
     end for
     Build new dataset $\mathcal{D}_{a,t}^k = \{(x_{i,t}, a), z_{i,a,t}^k\}_{i=1}^{n_x}$
    end for
    Compute $\widehat{W}_a^k$ by multi-task regression on the datasets $\{\mathcal{D}_{a,t}^k\}_{t=1}^{T}$ (see Eqs. 4,10, or 13)
   end for
  while $\left(\max_a \left\|W_a^k - W_a^{k-1}\right\|_2 \geq tol\right)$ **and** $k < K$

2.2. Fitted Q-iteration with linear function approximation

Whenever $\mathcal{X}$ and $\mathcal{A}$ are large or continuous, we need to resort to approximation schemes to learn a near-optimal policy. One of the most popular ADP methods is the fitted-Q iteration (FQI) algorithm [7], which extends value iteration to approximate action-value functions. While exact value iteration proceeds by iterative applications of the Bellman operator (i.e., $Q^k = \mathcal{T}Q^{k-1}$), in FQI, each iteration approximates $\mathcal{T}Q^{k-1}$ by solving a regression problem. Among possible instances, here we focus on a specific implementation of FQI in the fixed design setting with linear approximation and we assume access to a generative model of the MDP. Since the action space $\mathcal{A}$ is finite, we approximate an action-value function as a collection of $|\mathcal{A}|$ independent state-value functions. We introduce a d_x-dimensional state-feature vector $\phi(\cdot) = [\varphi_1(\cdot), \varphi_2(\cdot), \ldots, \varphi_{d_x}(\cdot)]^\mathsf{T}$ with $\varphi_i : \mathcal{X} \to \mathbb{R}$ such that $\sup_x ||\phi(x)||_2 \leq L$, while the corresponding state-action feature vector is

$$\psi(x, a) = \big[\underbrace{0, \cdots, 0}_{\substack{(a-1)\times d_x \\ \text{times}}}, \varphi_1(x), \ldots, \varphi_{d_x}(x), \underbrace{0, \cdots, 0}_{\substack{(|\mathcal{A}|-a)\times d_x \\ \text{times}}}\big]^\mathsf{T},$$

with dimension $d = |\mathcal{A}| \times d_x$. From ϕ we construct a linear approximation space for action-value functions as $\mathcal{F} = \{f_w(\cdot, \cdot) = \psi(\cdot, \cdot)^\mathsf{T} w, w \in \mathbb{R}^d\}$ where the weight vector w can be decomposed as $w = [w_1, \ldots, w_{|\mathcal{A}|}]$ so that for any $a \in \mathcal{A}$, we have $f_w(\cdot, a) = \phi(\cdot)^\mathsf{T} w_a$. F$Q$I receives as input a fixed set of states $\mathcal{S} = \{x_i\}_{i=1}^{n_x}$ (*fixed design setting*) and the space $\mathcal{F}$. Starting from $w^0 = \mathbf{0}$ defining the function $\widehat{Q}^0$, at each iteration k, FQI first draws a (fresh) set of samples $(r_{i,a}^k, y_{i,a}^k)_{i=1}^{n_x}$ from the generative model of the MDP for each action $a \in \mathcal{A}$ on each of the states $\{x_i\}_{i=1}^{n_x}$ (i.e., $r_{i,a}^k = R(x_i, a)$ and $y_{i,a}^k \sim P(\cdot|x_i, a)$). From the samples, $|\mathcal{A}|$ independent training sets $\mathcal{D}_a^k = \{(x_i, a), z_{i,a}^k\}_{i=1}^{n_x}$ are generated, where

$$z_{i,a}^k = r_{i,a}^k + \gamma \max_{a'} \widehat{Q}^{k-1}(y_{i,a}^k, a'), \tag{1}$$

and $\widehat{Q}^{k-1}(y_{i,a}^k, a')$ is computed using the weight vector learned at the previous iteration as $\psi(y_{i,a}^k, a')^\mathsf{T} w^{k-1}$ (or equivalently $\phi(y_{i,a}^k)^\mathsf{T} w_{a'}^{k-1}$). Notice that each $z_{i,a}^k$ is an unbiased sample of $\mathcal{T}\widehat{Q}^{k-1}$ and it can be written as

$$z_{i,a}^k = \mathcal{T}\widehat{Q}^{k-1}(x_i, a) + \eta_{i,a}^k, \tag{2}$$

where $\eta_{i,a}^k$ is a zero-mean noise bounded in the interval $[-Q_{\max}; Q_{\max}]$. Then FQI solves $|\mathcal{A}|$ linear regression problems, each fitting the training set $\mathcal{D}_a^k$ and it returns vectors $\widehat{w}_a^k$, which lead to the new action value function $f_{\widehat{w}^k}$ with $\widehat{w}^k = [\widehat{w}_1^k, \ldots, \widehat{w}_{|\mathcal{A}|}^k]$. Notice that at each iteration the total number of samples is $n = |\mathcal{A}| \times n_x$. The process is repeated until a fixed number of iterations K is reached or no significant change in the weight vector is observed. Since in principle $\widehat{Q}^{k-1}$ could be unbounded (due to numerical issues in the regression step), in computing the samples $z_{i,a}^k$ we can use a function $\widetilde{Q}^{k-1}$ obtained by truncating $\widehat{Q}^{k-1}$ within $[-Q_{\max}; Q_{\max}]$. In order to simplify the notation, we also introduce the matrix form of the elements used by FQI as $\Phi = [\phi(x_1)^\mathsf{T}; \cdots; \phi(x_{n_x})^\mathsf{T}] \in \mathbb{R}^{n_x \times d_x}$, $\Phi_a'^k = [\phi(y_{i,a}^k)^\mathsf{T}; \cdots; \phi(y_{n_x,a}^k)^\mathsf{T}] \in \mathbb{R}^{n_x \times d_x}$, $R_a^k = [r_{1,a}^k, \ldots, r_{n_x,a}^k] \in \mathbb{R}^{n_x}$, and the vector $Z_a^k = [z_{1,a}^k, \ldots, z_{n_x,a}^k] \in \mathbb{R}^{n_x}$ obtained as

$$Z_a^k = R_a^k + \gamma \max_{a'} (\Phi_{a'}'^k w_{a'}^{k-1}).$$

The convergence and the performance of FQI are studied in detail in [21] in the case of bounded approximation space, while linear FQI is studied in [18, Thm. 5] and [24 Lemma 5]. When moving to the multi-task setting, we consider different state sets $\{\mathcal{S}_t\}_{t=1}^{T}$ and each of the previous terms is defined for each task $t \in [T]$ as Φ_t^k, $\Phi_{a,t}'^k$, $R_{a,t}^k$, $Z_{a,t}^k$ and we denote by $\widehat{W}^k \in \mathbb{R}^{d \times T}$ the matrix with vector $\widehat{w}_t^k \in \mathbb{R}^d$ as the t–th column. The general structure of FQI in a multi-task setting is reported in Fig. 1.

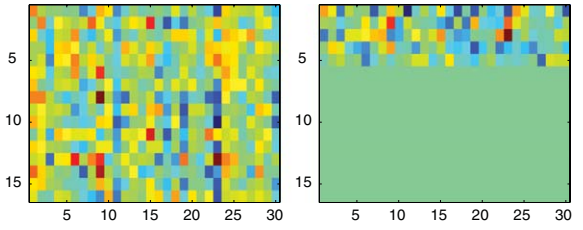


Fig. 1. Visualization of $\|W\|_{2,1}$ penalties (high on the left and low on the right).

Finally, we also introduce the following matrix notation. For any matrix $W \in \mathbb{R}^{d\times T}$, $[W]_t \in \mathbb{R}^d$ is the t-th column and $[W]^i \in \mathbb{R}^T$ the i-th row of the matrix, $\mathrm{Vec}(W)$ is the $\mathbb{R}^{dT}$ vector obtained by stacking the columns of the matrix one on top of each other, $\mathrm{Col}(W)$ is its column-space and $\mathrm{Row}(W)$ is its row-space. In addition to the classical ℓ_2, ℓ_1 norm for vectors, we also use the trace (or nuclear norm) $\|W\|_* = \mathrm{trace}((WW^\mathsf{T})^{1/2})$, the Frobenius norm $\|W\|_F = (\sum_{i,j}[W]_{i,j}^2)^{1/2}$ and the $\ell_{2,1}$-norm $\|W\|_{2,1} = \sum_{i=1}^d \|[W]^i\|_2$. We denote by $\boldsymbol{O}^d$ the set of orthonormal matrices. Finally, for any pair of matrices V and W, $V \perp \mathrm{Row}(W)$ denotes the orthogonality between the spaces spanned by the two matrices.

3. Fitted Q–iteration in sparse MDPs

Depending on the regression algorithm employed at each iteration, FQI can be designed to take advantage of different characteristics of the functions at hand, such as smoothness (ℓ_2–regularization) and sparsity (ℓ_1–regularization). In this section we consider the standard high–dimensional regression scenario and we study the performance of FQI under sparsity assumptions. Define the greedy policy w.r.t. a Q^k function as $\pi^k(x) = \arg\max_a Q^k(x,a)$. We start with the following assumption.

Assumption 1. The linear approximation space $\mathcal{F}$ is such that for any function $f_{w^k} \in \mathcal{F}$, the Bellman operator $\mathcal{T}$ can be expressed as

$$\begin{aligned}&\mathcal{T} f_{w^k}(x,a)\\ &= R(x,a) + \gamma \mathbb{E}_{x'\sim P(\cdot|x,a)}\left[Q(x',\pi^k(x'))\right]\\ &= \psi(x,a)^\mathsf{T} w^R + \gamma \psi(x,a)^\mathsf{T} P_\psi^{\pi^k} w^k, \qquad (3)\end{aligned}$$

where π^k is greedy w.r.t. f_{w^k}.

The main consequence of this assumption is that the image of the Bellman operator is contained in $\mathcal{F}$, since it can be computed as the product between features $\psi(x,a)$ and a vector of weights w^R and $P_\psi^{\pi^k} w^k$. This implies that after enough applications of the Bellman operator, the function $f_{w^*} = Q^*$ will belong to $\mathcal{F}$ as a combination $\psi(x,a)^\mathsf{T} w^*$. The assumption encodes the intuition that in the high–dimensional feature space $\mathcal{F}$ induced by ψ, the transition kernel P, and therefore the system dynamics, can be expressed as a linear combination of the features using the matrix $P_\psi^{\pi^k}$. This condition is usually satisfied whenever the space $\mathcal{F}$ is spanned by a very large set of features that allows it to approximate a wide range of different functions, including the reward and transition kernel. The matrix $P_\psi^{\pi^k}$ is dependent on the previous Q^k approximation through the π^k policy, and on the feature representation ψ, since it effectively encodes the operator $\int_{x'} P(dx'|x,a)Q^k(x',\pi^k(x'))dx'$. Under this assumption, at each iteration of FQI, there exists a weight vector w^k such that $\mathcal{T}\widehat{Q}^{k-1} = f_{w^k}$ and an approximation of the target function f_{w^k} can be obtained by solving an ordinary least-squares problem on the samples in $\mathcal{D}_a^k$. Unfortunately, it is well known that OLS fails whenever the number of samples is not sufficient w.r.t. the number of features (i.e., $d > n$). For this reason, Asm. 1 is often joined together with a sparsity assumption. Let $J(w) = \{i = 1,\dots,d : w_i \neq 0\}$ be the set of s non-zero components of vector w (i.e., $s = |J(w)|$) and $J^c(w)$ be the complementary set. In supervised learning, the LASSO is effective in exploiting the sparsity assumption that $s \ll d$ and dramatically reduces the sample complexity, so that the squared prediction error of $\tilde{O}(d/n)$ of OLS decreases to $\tilde{O}(s \log d/n)$ for LASSO (under specific assumptions), thus moving from a linear dependency on the number of features to a linear dependency only on the features that are actually useful in approximating the target function. A detailed discussion about LASSO, its implementation and theoretical guarantees can be found in [4] and [12]. In RL the idea of sparsity has been successfully integrated into policy evaluation [9, 13, 15, 23] but rarely in the full policy iteration. In value iteration, it can be easily integrated in FQI by approximating the target weight vector w_a^k through LASSO as[2]

[2]Notice that when performing linear regression, it is important to include a constant feature to model the offset of the function. To avoid regularizing this term in the optimization, we subtract its average from the target of the regression, and then add it again when evaluating the function. For this reason at iteration k we may also store a bias $b_a^k \in \mathbb{R}$ for each action. Once the algorithm terminates it returns the weights

$$\widehat{w}_a^k = \arg\min_{w\in\mathbb{R}^{d_X}} \frac{1}{n_x}\sum_{i=1}^{n_x}\left(\phi(x_i)^\mathsf{T} w - z_{i,a}^k\right)^2 + \lambda||w||_1. \tag{4}$$

While this integration is technically simple, the conditions on the MDP structure that imply sparsity in the value functions are not fully understood. In fact, we could simply assume that the optimal value function Q^* is sparse in $\mathcal{F}$, with s non-zero weights, thus implying that $d - s$ features capture aspects of states and actions that do not have any impact on the actual optimal value function. Nonetheless, this would not provide any guarantee about the actual level of sparsity encountered by FQI through iterations, where the target functions f_{w^k} may not be sparse at all. For this reason we need stronger conditions on the structure of the MDP. In [6,11], it has been observed that state features that do not affect either immediate rewards or future rewards through the transition kernel can be discarded without loss of information about the value function. Thus, we introduce the following assumption.[3]

Assumption 2. (Sparse MDPs). Given the set of states $\mathcal{S} = \{x_i\}_{i=1}^{n_x}$ used in FQI, there exists a set J (set of useful features) for MDP $\mathcal{M}$, with $|J| = s \ll d$, such that for any $i \notin J$, and any policy π

$$\left[P_\psi^\pi\right]^i = 0, \tag{5}$$

and there exists a function $f_{w^R} = R$ such that $J(w^R) \subseteq J$.

Assumption 2 implies that not only the reward functions are all sparse, but also that the features that are useless (i.e., features not in J) have no impact on the dynamics of the system. Building on the previous interpretation of P_ψ^π as the linear representation of the transition kernel embedded in the high-dimensional space $\mathcal{F}$, we can see that the assumption corresponds to imposing that the matrix P_ψ^π has all its rows corresponding to features outside of J set to 0. This in turn means that the future state-action vector $\mathbb{E}[\psi(x', a')^\mathsf{T}] = \psi(x, a)^\mathsf{T} P_\psi^\pi$ depends only on the features in J. In the blackjack scenario illustrated in the introduction, this assumption is verified by features related to the history of the cards played so far. In fact, if we consider an infinite number of decks, the feature indicating whether an ace has already been played is not used in the definition of the reward function and it is completely unrelated to the other features and, thus it does not contribute to the optimal value function. As an example of what constitutes a group of similar tasks, we can consider the control of a humanoid robot. Humanoid robots are equipped with a large number of sensors (both internal and external) and actuators that allow them to perform a wide variety of tasks. In tasks such as grasping objects, writing with a pen, tying knots, and other manipulation tasks, the controller needs to consider information about the surrounding environment and information relative to position, speed, and acceleration of joints in robot upper-body. This means that all the information coming from sensors positioned in the legs of the robot can be ignored since they are not relevant to accomplish such tasks. So, in the humanoid robot context, manipulation tasks can be referred to as a group of "similar" tasks since they share a subset of features that are relevant for solving the different control problems. Although this may appear as an extreme scenario, similar configurations may often happen in robotic problems (or other domains where physical systems are considered) in which starting from the raw reading of the sensors (e.g., position and speed), features are built by taking polynomials of the basic state variables. In this situation, it is often the case that only few polynomials are actually useful (e.g., the dynamics may be linear in position and speed), while other polynomials could be discarded without preventing the learning of the optimal action-value function. Since such structure could be shared across multiple tasks, then the shared-sparisty assumption would be verified and multi-task approaches could be very effective. Two important considerations on this Assumption can be derived by a closer look to the sparsity pattern of the matrix P_ψ^π. Since the sparsity is required at the level of the rows, this does not mean that the features that do not belong to J have to be equal to 0 after each transition. Instead, their value will be governed simply by the interaction with the features in J. This means that the features outside of J can vary from completely unnecessary features with no dynamics, to features that are redundant to those in J to describe the evolution of the system. Another important point is the presence of linear dependency among the non-zero rows in P_ψ^π. Because it is often the case that we do not have access to the P_ψ^π matrix, it is possible that in practice dependent features are introduced in the high-dimensional setting. In this case we could select only an independent subset of them to be included in J and remove the remaining, but this can not be easily done

$\widehat{w}_a^K$ together with the bias b_a^K, that can be used to determine the policy in any state.

[3]Notice that this assumption can be interpreted as an explicit sufficient condition for feature independency in the line of [11, Equation 5], where a completely implicit assumption is formalized. Furthermore, a similar assumption has been previously used in [10] where the transition P is embedded in a RKHS.

in practice without full access to the model. For the rest of the paper we assume for simplicity that the sparsity pattern J is unique. As we will see later, the presence of multiple possible P_ψ^π matrices and sparsity patterns J is not a problem for the regression algorithms that we use, and we will provide a longer discussion after introducing more results on sparse regression in Remark 2 of Theorem 1. Assumption 2, together with Asm. 1, leads to the following lemma.

Lemma 1. *Under Assumptions 1 and 2, the application of the Bellman operator $\mathcal{T}$ to any function $f_w \in \mathcal{F}$, produces a function $f_{w'} = \mathcal{T} f_w \in \mathcal{F}$ such that $J(w') \subseteq J$.*

Proof. As stated in Assumption 1, $\mathcal{F}$ is closed under the Bellman operator $\mathcal{T}$, i.e., $f_w \in \mathcal{F} \Rightarrow \mathcal{T} f_w \in \mathcal{F}$. We also introduced the $P_\psi^{\pi^k}$ matrix that represents the expected transition kernel in the High-Dimensional space. Using this assumption, we have that, given a vector w^k, for all $x \in \mathcal{X}$ there exists a w^{k+1} such that

$$\begin{aligned} f_{w^{k+1}}(x,a) &= \psi(x,a)^\mathsf{T} w^{k+1} \\ &= \psi(x,a)^\mathsf{T} w^R + \gamma \psi(x,a) P_\psi^{\pi^k} w^k \\ &= \mathcal{T} f_{w^k}. \end{aligned}$$

Clearly vector $w^{k+1} = w^R + P_\psi^{\pi^k} w^k$ satisfies this condition. Under Assumption 2, we know that it exists a set of useful features J. Moreover, the assumption implies that the rows of the matrix $P_\psi^{\pi^k}$ corresponding to features outside the J set are equal to 0. The product $P_\psi^{\pi^k} w^k$ will therefore follow the same sparsity pattern of J, irregardless of w^k. This, in addition to the fact that $J(w^R) \subseteq J$, proves the lemma. □

The previous lemma guarantees that, at any iteration k of FQI, the target function $f_{w^k} = \mathcal{T}\widehat{Q}^{k-1}$ has a number of non-zero components $|J(w^k)| \leq s$. We are now ready to analyze the performance of LASSO-FQI over iterations. In order to make the following result easier to compare with the multi-task results in Sections 4 and 5, we analyze the accuracy of LASSO-FQI averaged over multiple tasks (which are solved independently). For this reason we consider that the previous assumptions extend to all the MDPs $\{\mathcal{M}_t\}_{t=1}^T$ with a set of useful features J_t such that $|J_t| = s_t$ and average sparsity $\bar{s} = (\sum_t s_t)/T$. The quality of the action-value function learned after K iterations is evaluated by computing the corresponding greedy policy $\pi_t^K(x) = \arg\max_a Q_t^K(x,a)$ and comparing its performance to the optimal policy. In particular, the performance loss is measured w.r.t. a target distribution $\mu \in \mathcal{P}(\mathcal{X} \times \mathcal{A})$. To provide performance guarantees we have first to introduce an assumption used in [3] to derive theoretical guarantees for LASSO.

Assumption 3. (Restricted Eigenvalues (RE)). For any $s \in [d]$, there exists $\kappa(s) \in \mathbb{R}^+$ such that:

$$\min \left\{ \frac{\|\Phi\Delta\|_2}{\sqrt{n}\,\|\Delta_J\|_2} \,:\, \Delta \in \mathbb{R}^d \backslash \{\mathbf{0}\}, \; |J| \leq s, \, \|\Delta_{J^c}\|_1 \leq 3\,\|\Delta_J\|_1 \right\} \geq \kappa(s), \quad (6)$$

where n is the number of samples, and J^c denotes the complement of the set of indices J.

Theorem 1. (LASSO-FQI). *Let the tasks $\{\mathcal{M}_t\}_{t=1}^T$ and the function space $\mathcal{F}$ satisfy assumptions 1, 2 and 3 with average sparsity $\bar{s} = \sum_t s_t / T$ and features bounded $\sup_x ||\phi(x)||_2 \leq L$. If LASSO-FQI (Algorithm 1 with Equation 4) is run independently on all T tasks for K iterations with a regularizer $\lambda = \delta Q_{\max} \sqrt{\frac{\log d}{n}}$, for any numerical constant $\delta > 8$, then, with probability at least $(1 - 2d^{1-\delta/8})^{KT}$, the performance loss is bounded as*

$$\frac{1}{T}\sum_{t=1}^T \left\| Q_t^* - Q_t^{\pi_t^K} \right\|_{2,\mu}^2 \leq \mathcal{O}\left(\frac{1}{(1-\gamma)^4} \left[\frac{Q_{\max}^2 L^2}{\kappa_{\min}^4(\bar{s})} \frac{\bar{s}\log d}{n} + \gamma^K Q_{\max}^2 \right] \right), \quad (7)$$

where $\kappa_{\min}(\bar{s}) = \min_t \kappa(s_t)$.

Remark 1 (concentrability terms). Unlike similar analyses for FQI (see e.g., [21]), no concentrability term appears in the previous bound. This is possible because at each iteration LASSO provides strong guarantees about the accuracy in approximating the weight vector of the target function by bounding the error $||w_t^k - \widehat{w}_t^k||_2$. This, together with the boundedness of the features $||\phi(x)||_2 \leq L$, provides an ℓ_∞-norm bound on the prediction error $||f_{w_t^k} - f_{\widehat{w}_t^k}||_{2,\infty}$ which allows for removing the concentrability terms relative to the propagation of the error.

Remark 2 (assumptions). Intuitively, Assumption 3 gives us a weak constraint on the representation capability of the data. In an OLS approach, the rank of the matrix $\Phi^\mathsf{T}\Phi$ is required to be strictly greater than 0. This can be expressed also as $\|\Phi\Delta\|_2 / \|\Delta\|_2 > 0$, because the minimum quantity that this expression can take is equal to the smallest singular value of Φ. In a LASSO setting, the number of features d is usually much larger than the number of samples, and the matrix

$\Phi^\mathsf{T}\Phi$ is often not full rank. The RE Assumption forces a much weaker restriction focusing on a condition on $\|\Phi\Delta\|_2 / \|\Delta_J\|_2$, where in the denominator the norm $||\Delta_J||_2$ only focuses on the components of Δ in the set J. This vector is composed only by the non-zero groups of variable, and intuitively this norm will be larger than the smallest eigenvalue of the part of the matrix Φ related to the non-zero groups. $\kappa(s)$ is therefore a lower bound on the capability of the matrix Φ to represent a solution not for the full OLS problem, but only for the sparse subspace that truly supports the target function. A number of sufficient conditions are provided in [29], among them one of the most common, although much stronger than the RE, is the Restricted Isometry Condition. Assumptions 1 and 2 are specific to our setting and may provide a significant constraint on the set of MDPs of interest. Assumption 1 is introduced to give a more explicit interpretation for the notion of sparse MDPs. In fact, without Assumption 1, the bound in Equation 7 would have an additional approximation error term similar to standard approximate value iteration results (see e.g., [21]). Assumption 2 is a potentially very loose sufficient condition to guarantee that the target functions encountered over the iterations of LASSO–FQI have a minimum level of sparsity. More formally, the necessary condition needed for Thm. 1 is that for any $k \leq K$, the weight w_t^{k+1} corresponding to $f_{w_t^{k+1}} = \mathcal{T} f_{w_t^k}$ (i.e., the target function at iteration k) is such that there exists $s \ll d$ such that $\max_{k\in[K]} \max_{t\in[T]} s_t^k \leq s$ where $s_t^k = |J(w_t^{k+1})|$. Such condition can be obtained under much less restrictive assumptions than Assumption 2 at the cost of a much lower level of interpretability (see e.g., [11]). Without this necessary condition, we may expect that, even with sparse Q_t^*, LASSO–FQI may generate through iterations some regression problems with little to no sparsity, thus compromising the performance of the overall process. Nonetheless, we recall that LASSO is proved to return approximations which are as sparse as the target function. As a result, to guarantee that LASSO–FQI is able to take advantage of the sparsity of the problem, it may be enough to state a milder assumption that guarantees that $\mathcal{T}$ never reduces the level of sparsity of a function below a certain threshold and that the Q_t^* functions are sparse. As discussed in the definition of Assumption 2, we decided to consider $J(w_t^k)$ to be unique for each task. This is not guaranteed to hold when the rows of the matrix $P_\phi^{\pi^k}$ that are in J are not linearly independent. Nonetheless, if we consider that at each step the new weight vector w^{k+1} is chosen to be sparse, we see that LASSO will naturally disregard linearly correlated lines in order to produce a sparser solution. On the other hand, not all sparsity patterns can be recovered from the actual samples that we use for regression. In particular, we can only recover patterns for which Assumption 3 holds. Therefore the LASSO guarantees hold for the sparsity pattern $J(w^{k+1})$ such that the ratio $|J(w^{k+1})|/\kappa^4(J(w^{k+1}))$ is most favorable, while the patterns that do not satisfy Assumption 3 have a 0 denominator and are automatically excluded from the comparison. Finally, we point out that even if "useless" features (i.e., features that are not used in Q_t^*) do not satisfy Equation 5 and are somehow correlated with other (useless) features, yet their weights would be discounted by γ at each iteration (since not "reinforced" by the reward function). As a result, over iterations the target functions would become "approximately" as sparse as Q_t^* and this, together with a more refined analysis of the propagation error as in [8], would possibly return a result similar to Thm. 1. We leave for future work a more thorough investigation of the extent to which these assumptions can be relaxed.

Proof. We recall from Asm. 1 and Lemma 1, that at each iteration k and for each task t, samples $z_{i,a,t}^k$ can be written as

$$z_{i,a,t}^k = f_{w_t^k}(x_{i,t}, a) + \eta_{i,a,t}^k = [\Phi_t]_i w_{a,t}^k + \eta_{i,a,t}^k,$$

where $w_a^k \in \mathbb{R}^d$ is the vector that contains the weight representing exactly the next value function for each task. With this reformulation we made explicit the fact that the samples are obtained as random observations of linear functions evaluated on the set of points in $\{\mathcal{S}_t\}_{t\in[T]}$. Thus we can directly apply the following proposition.

Proposition 1 [3]. *For any task $t \in [T]$, any action $a \in \mathcal{A}$ and any iteration $k < K$, let $w_{a,t}^k$ be sparse such that $|J(w_{a,t})| \leq s_t^k$ and satisfy Assumption 3 with $\kappa_t^k = \kappa(s_t^k)$. Then if Equation 4 is run independently on all T tasks with a regularizer $\lambda = \delta Q_{\max}\sqrt{\frac{\log d}{n}}$, for any numerical constant $\delta > 2\sqrt{2}$, then with probability at least $1 - d^{1-2\delta^2/8}$, the function $f_{\widehat{w}_{a,t}^k}$ computed in Equation 4 has an error bounded as*

$$\left\| w_{a,t}^k - \widehat{w}_{a,t}^k \right\|_2^2 \leq \frac{256\delta^2 Q_{\max}^2}{\kappa^4(s_t^k)} \frac{s_t^k \log d}{n}. \qquad (8)$$

In order to prove the final theorem we need to adjust previous results from [21] to consider how this error is propagated through iterations. We begin by recalling the intermediate result from [21] about the propagation

of error through iterations adapted to the case of action-value functions. For any policy π, given the right-linear operator $P_t^\pi : \mathcal{B}(\mathcal{X}\times\mathcal{A}) \to \mathcal{B}(\mathcal{X}\times\mathcal{A})$

$$(P_t^\pi Q)(x,a) = \int_y P_t(y|x,a) \sum_b \pi(b|x) Q(y,b),$$

we have that after K iterations for each task $t \in [T]$

$$|Q_t^* - Q_t^{\pi_t^K}| \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \Bigg[\sum_{k=0}^{K-1} \alpha_k A_{tk} |\varepsilon_t^k| + \alpha_K A_{tK} |Q_t^* - Q_t^0| \Bigg],$$

with

$$\alpha_k = \begin{cases} \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}}, & \text{for } 0 \le k < K, \\ \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}}, & \text{for } k = K \end{cases}$$

$$A_{tk} = \frac{1-\gamma}{2}(I - \gamma P_t^{\pi_t^K})^{-1} \times \left[(P_t^{\pi_t^*})^{K-k} + P_t^{\pi_t^K} P_t^{\pi_t^{K-1}} \dots P_t^{\pi_t^{k+1}} \right],$$

$$A_{tK} = \frac{1-\gamma}{2}(I - \gamma P_t^{\pi_t^K})^{-1} \times \left[(P_t^{\pi_t^*})^{K+1} + P_t^{\pi_t^K} P_t^{\pi_t^{K-1}} \dots P_t^{\pi_t^0} \right].$$

and with the state-action error $\varepsilon_t^k(y,b) = \widehat{Q}^k(y,b) - \mathcal{T}_t \widehat{Q}^{k-1}(y,b)$ measuring the approximation error of action value functions at each iteration. We bound the error in any state $y \in \mathcal{X}$ and for any action $b \in \mathcal{A}$ as

$$\begin{aligned} |\varepsilon_t^k(y,b)| &= |f_{w_t^k}(y,b) - f_{\widehat{w}_t^k}(y,b)| \\ &= |\phi(y)^\mathsf{T} w_{b,t}^k - \phi(y)^\mathsf{T} \widehat{w}_{b,t}^k| \\ &\le ||\phi(y)||_2 ||w_{b,t}^k - \widehat{w}_{b,t}^k||_2 \\ &\le L ||w_{b,t}^k - \widehat{w}_{b,t}^k||_2, \end{aligned}$$

We notice that the operators A_{tk}, once applied to a function in a state-action pair (x,a), are well-defined distributions over states and actions and thus we can rewrite the previous expression as

$$\begin{aligned} &|Q_t^* - Q_t^{\pi_t^K}| \\ &\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \left[\sum_{k=0}^{K-1} \alpha_k A_{tk} L \max_b ||w_{b,t}^k - \widehat{w}_{b,t}^k||_2 + 2\alpha_K A_{tK} Q_{\max} \right] \\ &\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \left[\sum_{k=0}^{K-1} \alpha_k L \max_b ||w_{b,t}^k - \widehat{w}_{b,t}^k||_2 + 2\alpha_K Q_{\max} \right]. \end{aligned} \tag{9}$$

Taking the average value, and introducing the bound in Proposition 1 we have that

$$\frac{1}{T}\sum_{t=1}^T \left\| Q_t^* - Q_t^{\pi_t^K} \right\|_{2,\mu}^2 \le \left[\frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \right]^2 \times \left[\sum_{k=0}^{K-1} \alpha_k L^2 Q_{\max}^2 \frac{1}{T} \sum_{t=1}^T \frac{s_t^k}{\kappa^4(s_t^k)} \frac{\log d}{n} + 2\alpha_K Q_{\max}^2 \right].$$

holds. Since from Lemma 1, $s_t^k \le |J_t| = s_t$ for any iteration k, this proves the statement. □

4. Group-LASSO fitted Q–iteration

After introducing the concept of MDP sparsity in Section 3, we now move to the multi-task scenario and we study the setting where there exists a suitable representation (i.e., set of features) under which all the tasks can be solved using roughly the same set of features, the so-called *shared sparsity* assumption. We consider that assumptions 1 and 2 hold for all the tasks $t \in [T]$, such that each MDP $\mathcal{M}_t$ is characterized by a set J_t such that $|J_t| = s_t$. We denote by $J = \cup_{t=1}^T J_t$ the union of all the useful features across all the tasks and we state the following assumption.

Assumption 4. We assume that the joint useful features across all the tasks are such that $|J| = \tilde{s} \ll d$.

This assumption implies that the set of features "useful" for at least one of the tasks is relatively small compared to d. As a result, we have the following result.

Lemma 2. *Under Assumptions 2 and 4, at any iteration k, the target weight matrix $W^k \in \mathbb{R}^{d\times T}$ is such that $J(W^k) \le \tilde{s}$, where $J(W) = \cup_{t=1}^T J([W^k]_t)$.*

Proof. By Lemma 1, we have that for any task t, at any iteration k, $J([W^k]_t) \subseteq J_t$, thus $J(W^k) = \cup_{t=1}^T J([W^k]_t) \subseteq J$ and the statement follows. □

Finally, we notice that in general the number of jointly non-zero components cannot be smaller than in each task individually as $\max_t s_t \le \tilde{s} \le d$. In the following we introduce a multi-task extension of FQI

where the samples coming from all the tasks contribute to take advantage of the shared sparsity assumption to reduce the sample complexity and improve the average performance.

4.1. The algorithm

In order to exploit the similarity across tasks stated in Asm. 4, we resort to the Group LASSO (GL) algorithm [12, 20], which defines a joint optimization problem over all the tasks. GL is based on the observation that, given the weight matrix $W \in \mathbb{R}^{d\times T}$, the norm $\|W\|_{2,1}$ measures the level of shared-sparsity across tasks. In fact, in $\|W\|_{2,1}$ the ℓ_2-norm measures the "relevance" of feature i across tasks, while the ℓ_1-norm "counts" the total number of relevant features, which we expect to be small in agreement with Asm. 4. In Fig. 1 we provide a visualization on the case when $\|W\|_{2,1}$ is small and large. Building on this intuition, we define the GL–FQI algorithm in which, using the notation introduced in Section 2.2, the optimization problem solved by GL at each iteration for each action $a \in \mathcal{A}$ is

$$\widehat{W}_a^k = \arg\min_{W_a} \sum_{t=1}^{T} \left\| Z_{a,t}^k - \Phi_t w_{a,t} \right\|_2^2 + \lambda \|W_a\|_{2,1}. \quad (10)$$

Further details on the implementation of GL–FQI are reported in [5].

4.2. Theoretical analysis

The multi-task regularized approach of GL–FQI is designed to take advantage of the shared-sparsity assumption at each iteration and in this section we show that this may lead to reduce the sample complexity w.r.t. using LASSO in FQI for each task separately. Before reporting the analysis of GL–FQI, we need to introduce a technical assumption defined in [20] for GL.

Assumption 5 (Multi-Task Restricted Eigenvalues). For any $s \in [d]$, there exists $\kappa(s) \in \mathbb{R}^+$ such that:

$$\min \left\{ \frac{\|\Phi \operatorname{Vec}(\Delta)\|_2}{\sqrt{n}\, \|\operatorname{Vec}(\Delta_J)\|_2} : \Delta \in \mathbb{R}^{d\times T} \setminus \{\mathbf{0}\}, \quad (11) \right.$$
$$\left. |J| \le s, \|\Delta_{J^c}\|_{2,1} \le 3 \|\Delta_J\|_{2,1} \right\} \ge \kappa(s),$$

where n is the number of samples, J^c denotes the complement of the set of indices J, and Φ indicates the block diagonal matrix composed by the union of the T sample matrices Φ_t.

This assumption provides us with similar guarantees as Prop. 1.

Proposition 2 [20]. *For any action $a \in \mathcal{A}$ and any iteration $k < K$, let W_a^k be sparse such that $|J(W_a^k)| \le \widetilde{s}^k$ and satisfy Assumption 5 with $\kappa_t^k = \kappa(2s_t^k)$. Then if Equation [10] is run with a regularizer $\lambda = \frac{LQ_{\max}}{\sqrt{nT}} \left(1 + \frac{(\log d)^{\frac{3}{2}+\delta}}{\sqrt{T}}\right)^{\frac{1}{2}}$, for any numerical constant $\delta > 0$, then with probability $\Omega(1 - \log(d)^{-\delta})$, the function $f_{\widehat{w}_{a,t}^k}$ computed in Equation 4 has an error bounded as*

$$\frac{1}{T}\sum_{t=1}^{T} \left\| [W_a^k]_t - [\widehat{W}_a^k]_t \right\|_2^2$$
$$= \frac{1}{T} \left\| \operatorname{Vec}(W_a^k) - \operatorname{Vec}(\widehat{W}_a^k) \right\|_2^2$$
$$\le \frac{160 L^2 Q_{\max}^2}{\kappa_{Td}^4(2\tilde{s})} \frac{\tilde{s}}{n} \left(1 + \frac{(\log d)^{3/2+\delta}}{\sqrt{T}}\right).$$

Similar to Theorem 1 we evaluate the performance of GL–FQI as the performance loss of the returned policy w.r.t. the optimal policy and we obtain the following performance guarantee. The proof is similar to Thm. 1, using Prop. 2 instead of 1.

Theorem 2 (GL–FQI). *Let the tasks $\{\mathcal{M}_t\}_{t=1}^T$ and the function space $\mathcal{F}$ satisfy assumptions 1, 2, 4, and 5 with joint sparsity $\tilde{s}$ and features bounded $\sup_x ||\phi(x)||_2 \le L$. If* GL–F$Q$I *(Algorithm 1 with Equation 10) is run jointly on all T tasks for K iterations with a regularizer $\lambda = \frac{LQ_{\max}}{\sqrt{nT}} \left(1 + \frac{(\log d)^{\frac{3}{2}+\delta}}{\sqrt{T}}\right)^{\frac{1}{2}}$, for any numerical constant $\delta > 0$, then with probability at least*

$$1 - K \frac{4\sqrt{\log(2d)[64\log^2(12d)+1]^{1/2}}}{(\log d)^{3/2+\delta}} \simeq 1 - K\log(d)^{-\delta},$$

the performance loss is bounded as

$$\frac{1}{T}\sum_{t=1}^{T} \left\| Q_t^* - Q_t^{\pi_t^K} \right\|_{2,\mu}^2 \quad (12)$$
$$\le \mathcal{O}\left(\frac{1}{(1-\gamma)^4} \left[\frac{L^2 Q_{\max}^2}{\kappa^4(2\tilde{s})} \frac{\tilde{s}}{n} \left(1 + \frac{(\log d)^{3/2+\delta}}{\sqrt{T}}\right) + \gamma^K Q_{\max}^2 \right] \right).$$

Remark 1 (comparison with LASSO-FQI). We first compare the performance of GL–FQI to single-task FQI with LASSO regularization at each iteration. Ignoring

all the terms in common with the two methods, constants, and logarithmic factors, we can summarize their bounds as

$$\text{GL–FQI}: \ \widetilde{\mathcal{O}}\Big(\frac{\tilde{s}}{n}\big(1+\frac{\log d}{\sqrt{T}}\big)\Big),$$

$$\text{LASSO-FQI}: \ \widetilde{\mathcal{O}}\Big(\frac{\bar{s}\log d}{n}\Big),$$

where $\bar{s} = 1/T\sum_t s_t$ is the average sparsity. The first interesting aspect of the bound of GL–FQI is the role played by the number of tasks T. In LASSO–FQI the "cost" of discovering the s_t useful features is a factor $\log d$, while GL–FQI has a factor $1 + \log(d)/\sqrt{T}$, which decreases with the number of tasks. This illustrates the advantage of the multi–task learning dimension of GL–FQI, where all the samples of all tasks actually contribute to discovering useful features, so that the more the number of features, the smaller the cost. In the limit, we notice that when $T \to \infty$, the bound for GL–FQI does not depend on the dimensionality of the problem anymore. The other aspect of the bound that should be taken into consideration is the difference between $\bar{s}$ and $\tilde{s}$. In fact, if the shared-sparsity assumption does not hold, we can construct cases where the number of non-zero features s_t is very small for each task, but the *union* $J = \cup_t J_t$ is still a full set, so that $\tilde{s} \approx d$. In this case, GL–FQI cannot leverage on the shared sparsity across tasks and it may perform significantly worse than LASSO–FQI. This is the well–known *negative transfer* effect that happens whenever the wrong assumption over tasks is enforced thus worsening the single-task learning performance.

Remark 2 (assumptions). Assumption 5 is a rather standard (technical) assumption in Group-LASSO and RL and it is discussed in detail in the respective literature. The shared sparsity assumption (Assumption 4) is at the basis of the idea of the joint optimization defined in GL–FQI.

5. Feature learning fitted Q–iteration

Unlike other properties such as smoothness, the sparsity of a function is intrinsically related to the specific *representation* used to approximate it (i.e., the function space $\mathcal{F}$). While Assumption 2 guarantees that $\mathcal{F}$ induces sparsity for each task independently, Assumption 4 requires that all the tasks share the same useful features in the given representation. As discussed in Rem. 1, whenever this is not the case, GL–FQI may be affected by negative transfer and perform worse than LASSO–FQI. In this section we further investigate an alternative notion of sparsity in MDPs and we introduce the Feature Learning fitted Q-iteration (FL–FQI) algorithm, and derive finite–sample bounds.

5.1. Sparse representations and low rank approximation

Since the poor performance of GL–FQI may be due to a representation (i.e., definition of the features) which does not lead to similar tasks, it is natural to ask the question whether there exists an alternative representation (i.e., a different set of features) that induces a high-level of shared sparsity. Let as assume that there exists a linear space $\mathcal{F}^*$ defined by features ϕ^* such that the weight matrix of the optimal Q-functions is $A^* \in \mathbb{R}^{d\times T}$ such that $J(A^*) = s^* \ll d$. As shown in Lemma 2, together with Assumptions 2 and 4, this guarantees that at any iteration $J(A^k) \leq s^*$. Given the set of states $\{\mathcal{S}_t\}_{t=1}^T$, let Φ and Φ^* be the feature matrices obtained by evaluating ϕ and ϕ^* on the states. We assume that there exists a linear transformation of the features of $\mathcal{F}^*$ to the features of $\mathcal{F}$ such that $\Phi = \Phi^* U$ with $U \in \mathbb{R}^{d_x\times d_x}$. In this setting, at each iteration k and for each task t, the samples used to define the regression problem can be formulated as noisy observations of $\Phi^* A_a^k$ for any action a. Together with the transformation U, this implies that there exists a weight matrix W^k defined in the original space $\mathcal{F}$ such that $\Phi^* A_a^k = \Phi^* U U^{-1} A_a^k = \Phi W_a^k$ with $W_a^k = U^{-1} A_a^k$. It is clear that, although A_a^k is indeed sparse, any attempt to learn W_a^k using GL would fail, since W_a^k may have a very low level of sparsity. On the other hand, an algorithm able to learn a suitable transformation U, it may be able to recover the representation Φ^* (and the corresponding space $\mathcal{F}^*$) and exploit the high level of sparsity of A_a^k. This additional step of representation or *feature learning* introduces additional complexity, but allows to relax the strict assumption on the joint sparsity $\tilde{s}$. In particular, we are interested in the special case when the feature transformation is obtained using an orthogonal matrix U. Our assumption is formulated as follows.

Assumption 6. There exists an orthogonal matrix $U \in \boldsymbol{O}^d$ (the block matrix obtained by having transformation matrices $U_a \in \boldsymbol{O}^{d_x}$ for each action $a \in \mathcal{A}$ on the diagonal) such that the weight matrix A^* obtained as a transformation of W^* (i.e., $A^* = U^{-1}W^*$) is jointly sparse, i.e., has a set of "useful" features $J(A^*) = \cup_{t=1}^T J([A^*]_t)$ with $|J(A^*)| = s^* \ll d$.

Coherently with this assumption, we adapt the multi-task feature learning (MTFL) problem defined in [1] and at each iteration k for any action a we solve the optimization problem

$$(\widehat{U}_a^k, \widehat{A}_a^k) = \tag{13}$$

$$\arg\min_{U_a \in \boldsymbol{O}^d, A_a \in \mathbb{R}^{d\times T}} \sum_{t=1}^{T} ||Z_{a,t}^k - \Phi_t U_a [A_a]_t||^2 + \lambda \|A\|_{2,1}\,.$$

In order to better characterize the solution to this optimization problem, we study more in detail the relationship between A^* and W^* and analyze the two directions of the equality $A^* = U^{-1}W^*$. When A^* has s^* non-zero rows, then any orthonormal transformation W^* will have at most rank $r^* = s^*$. This suggests that instead of solving the joint optimization problem in Equation 13 and explicitly recover the transformation U, we may directly try to solve for low-rank weight matrices W. Then we need to show that a low-rank W^* does indeed imply the existence of a transformation to a jointly-sparse matrix A^*. Assume W^* has low rank r^*. It is then possible to perform a standard singular value decomposition $W^* = U\Sigma V = UA^*$. Because Σ is diagonal with r^* non-zero entries, A^* will have r^* non-zero rows. It is important to notice that A^* will not be an arbitrary matrix, but since it is the product of an orthonormal matrix with a diagonal matrix, it will have exactly r^* orthogonal rows. Although this construction shows that a low-rank matrix W^* may imply a sparse matrix A^*, the constraint coming from the SVD argument and the fact that A^* has orthogonal rows may prevent from finding the representation that indeed leads to the most sparse matrix (i.e., the matrix recovered from the SVD decomposition of a low-rank W may lead to a matrix A which is not as sparse as the A^* defined in Assumption 6). Fortunately, we can show that this is not the case by construction. Assume that starting from W^* an arbitrary algorithm produces a sparse matrix $A' = U^{-1}W^*$, with sparsity s'. Again, given a SVD decomposition $A' = U'\Sigma'V' = U'A''$. Because the rank r' of matrix A' is surely equal or smaller than s', we have that by construction A'' is an orthogonal matrix with at most s' non-zero rows. Finally, since $A'' = U'^{-1}A' = U'^{-1}U^{-1}W^*$, and since $U'^{-1}U^{-1}$ is still an orthonormal transformation, it is always possible to construct an orthogonal sparse matrix A^* that is not less sparse than any non-orthogonal alternatives. Based on such observations, it is possible to derive the following equivalence (the proof is mostly based on the results from [1] and it is available in full detail in [5]).

Proposition 3. *Given $A, W \in \mathbb{R}^{d\times T}$, $U \in \boldsymbol{O}^d$, the following equality holds*

$$\min_{A,U} \sum_{t=1}^{T} ||Z_{a,t}^k - \Phi_t U_a [A_a]_t||^2 + \lambda \|A\|_{2,1}$$

$$= \min_{W} \sum_{t=1}^{T} ||Z_{a,t}^k - \Phi_t [W_a]_t||^2 + \lambda \|W\|_1. \tag{14}$$

The relationship between the optimal solutions is $W^* = UA^*$.

In words the previous proposition states the equivalence between solving a feature learning version of GL and solving a nuclear norm (or trace norm) regularized problem. This penalty is equivalent to an ℓ_1-norm penalty on the singular values of the W matrix, thus forcing W to have low rank.

This is motivated by the fact that if there exists a representation $\mathcal{F}^*$ in which A^* is jointly sparse and that can be obtained by transformation of $\mathcal{F}$, then the rank of the matrix $W^* = U^{-1}A^*$ corresponds to the number of non-zero rows in A^*, i.e., the number of useful features. Notice that assuming that W^* has low rank can be also interpreted as the fact that either the task weights $[W^*]_t^*$ (the columns of W^*) or the features weights $[W^*]^i$ (the rows of W^*) are linearly correlated. In the first case, it means that there is a small dictionary, or basis, of core tasks that is able to reproduce all the other tasks as a linear combination. As a result, Assumption 6 can be reformulated as $\text{Rank}(W^*) = s^*$. Building on this intuition we define the FL–FQI algorithm that is identical to the GL–FQI except for the optimization problem, which is now replaced by Equation [14].

5.2. Theoretical analysis

Our aim is to obtain a bound similar to Theorem 2 for the new FL-FQI Algorithm. We begin by introducing a slightly stronger assumption on the data available for regression.

Assumption 7 (Restricted Strong Convexity). Under Assumption 6, let $W^* = UDV^\mathsf{T}$ be a singular value decomposition of the optimal matrix W^* of rank s^*, and U^{s^*}, V^{s^*} the submatrices associated with the top r singular values. Define $\mathcal{B} = \{\Delta \in \mathbb{R}^{d\times T} : \text{Row}(\Delta) \perp U^{s^*}$ and $\text{Col}(\Delta) \perp V^{s^*}\}$, and the projection operator onto this set $\Pi_{\mathcal{B}}$. There exists a positive constant κ such that

$$\min \left\{ \frac{\|\Phi \operatorname{Vec}(\Delta)\|_2^2}{2nT\|\operatorname{Vec}(\Delta)\|_2^2} : \Delta \in \mathbb{R}^{d\times T}, \right. \tag{15}$$
$$\left. \|\Pi_{\mathcal{B}}(\Delta)\|_1 \leq 3\|\Delta - \Pi_{\mathcal{B}}(\Delta)\|_1 \right\} \geq \kappa$$

The RSC assumption plays a central role in recent developements in high-dimensional statistics in regression, matrix completion and compressed sensing [22]. The corresponding proposition is

Lemma 3. *For any action $a \in \mathcal{A}$ and any iteration $k < K$, let W_a^k satisfy Assumption 6 with* $\operatorname{Rank}(W_a^k) \leq s^*$*, Assumption 7 with κ and $T > \mathcal{O}(\log n)$. Then if Equation 14 is run with a regularizer $\lambda \geq 2LQ_{\max}\sqrt{\frac{d+T}{n}}$ for any numerical constant $\delta > 0$ and the noise is symmetric*[4]*, then there exists constants c_1 and c_2 such that with probability at least $1 - c_1 \exp\{c_2(d+T)\}$ the function $f_{\widehat{w}_{a,t}^k}$ computed in Equation 4 has an error bounded as*

$$\frac{1}{T}\sum_{t=1}^{T} \left\| [W_a^k]_t - [\widehat{W}_a^k]_t \right\|_2^2$$
$$= \frac{1}{T}\|\widehat{W} - W^*\|_F^2 \leq \frac{4048L^2Q_{\max}^2 r(d+T)}{T\kappa^2 n}$$

We can now derive the main result of this section.

Theorem 3 (FL–FQI). *Let the tasks $\{\mathcal{M}_t\}_{t=1}^T$ and the function space $\mathcal{F}$ satisfy assumptions 1, 2, 6, and 7 with $s^* = \operatorname{Rank}(W^*)$, features bounded $\sup_x ||\phi(x)||_2 \leq L$ and $T > \mathcal{O}(\log n)$. If* FL–FQI *(Algorithm 1 with Equation 13) is run jointly on all T tasks for K iterations with a regularizer $\lambda \geq 2LQ_{\max}\sqrt{\frac{d+T}{n}}$, then there exist constants c_1 and c_2 such that with probability at least $(1 - c_1\exp\{c_2(d+T)\})^K$, the performance loss is bounded as*

$$\frac{1}{T}\sum_{t=1}^{T} \left\| Q_t^* - Q_t^{\pi_t^K} \right\|_{2,\rho}^2$$
$$\leq \mathcal{O}\left(\frac{1}{(1-\gamma)^4} \left[\frac{Q_{\max}^2 L^4}{\kappa^2} \frac{s^*}{n} \left(1 + \frac{d}{T}\right) + \gamma^K Q_{\max}^2 \right] \right).$$

[4]The requirement on the noise to be drawn from a symmetric distribution can be easily relaxed but the cost of a much more complicated proof. In fact, with an asymmetric noise, the truncation argument used in the proof of Lemma 3 would introduce a bias. Nonetheless, this would only translate in higher order terms in the bound and they would not change the overall dependency on the critical terms.

Remark 1 (comparison with GL-FQI). From the previous bound, we notice that FL–FQI does not directly depend on the shared sparsity $\tilde{s}$ of W^* but on its rank, that is the value s^* of the most jointly-sparse representation that can be obtained through an orthogonal transformation U of the given features X. As commented in the previous section, whenever tasks are somehow *linearly dependent*, even if the weight matrix W^* is dense and $\tilde{s} \approx d$, the rank s^* may be much smaller than d, thus guaranteeing a dramatic performance improvement over GL–FQI. On the other hand, learning a new representation comes at the cost of increasing the dependency on d. In fact, the factor $1 + \log(d)/\sqrt{T}$ in GL–FQI, becomes $1 + d/T$, implying that many more tasks are needed for FL–FQI to construct a suitable representation (i.e., compute weights with low rank). This is not surprising since we added a $d \times d$ matrix U in the optimization problem and a larger number of parameters needs to be learned. As a result, although significantly reduced by the use of trace-norm instead of $\ell_{2,1}$-regularization, the negative transfer is not completely removed. In particular, the introduction of new tasks, that are not linear combinations of the previous tasks, may again increase the rank s^*, corresponding to the fact that no alternative jointly-sparse representation can be constructed. Another way to interpret this is by imagining a small set of fundamental tasks. For example, recalling the humanoid robot case mentioned in Section 3, let us consider the basic tasks of grasping, picking up, and throwing. If, starting from the features that we provide, it is possible to extract a concise description of the optimal value function (e.g. in the rotated feature space) for all of these basic tasks and more complex tasks have optimal value functions that can be well approximated by a linear combination of the solutions to the basic tasks, then samples collected from the latter can be effectively reused to learn solutions to the former.

Remark 2 (assumptions). Assumption 7 is directly obtained from [22]. Intuitively, the top s^* singular values play the role of the non-zero groups, the space $\mathcal{B}$ is perpendicular to the non-zero part of the column space and row space (i.e., the submatrix of Φ with positive κ in RE). Then the residual $\Delta - \Pi_{\mathcal{B}}(\Delta)$ (that is parallel to the space spanned by the top s^* singular values because is perpendicular to $\mathcal{B}$) must be greater than the projection. This is similar to $\|\Delta_{J^c}\|_{2,1} \leq 3\|\Delta_J\|_{2,1}$ where we have spaces parallel and perpendicular to the top r subspace instead of group J and its complement.

6. Experiments

We investigate the empirical performance of GL–FQI, and FL–FQI and compare their results to single-task LASSO–FQI. First in Section 6.1 we report a detailed analysis in the chain walk domain, while in Section 6.2 we consider a more challenging blackjack domain.

6.1. Chain walk

In the chain walk domain, the agent is placed on a line and needs to reach a goal from a given starting position. The chain is a continuous interval with range $[0, 8]$, and the goal can be situated at any point in the interval $[2, 6]$. The agent has 2 actions at her disposal, a_1 and a_2, that correspond to a step in each direction. When choosing action a_1 the state of the environment, represented by the agent's position, transitions from x to $x' = x + 1 + \epsilon$ (respectively $x' = x - 1 + \epsilon$ for a_2), with ϵ a Gaussian noise. Given a goal $g = y$, the agent receives a reward 0 for every step, and a reward 1 when the future state x' is close to g, according to the formula $|x' - y| \leq 0.5$.

We generate T tasks by randomly selecting a position for the goal from $\mathcal{U}(2, 6)$, and we randomly select $n = 30$ samples for each task, starting from random positions and taking a random action. We force the inclusion of at least two transitions with reward equal to 1 to characterize each task. The average regret, evaluated by taking a set of random points $\{x_i\}_{i=1}^N$ and simulating many trajectories following the proposed policy and the optimal policy, is computed as:

$$\widetilde{R} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{E}_{\pi_t^*}\left[\sum_{j=0}^{K}\gamma^j r_j | x_0 = x_i\right] - \mathbb{E}_{\pi_t}\left[\sum_{j=0}^{K}\gamma^j r_j | x_0 = x_i\right]\right]. \quad (16)$$

We define two experiments to test GL–FQI and FL–FQI. In both cases, the chain is soft-discretized by defining 17 evenly spaced radial basis functions $\mathcal{N}(x_i, 0.05)$ on $[0, 8]$. To these 17 informative dimensions, we added noisy features $\mathcal{U}(-0.25, 0.25)$, for a total $d \in 17, \ldots, 2048$. In the first experiment, the features are inherently sparse, because the noisy dimensions are uncorrelated with the tasks. Since $s = 17 \ll d$ we expect a clear advantage of GL–FQI over LASSO. The averages and confidence intervals for regret are plotted in Figure 2. As expected, the GL–FQI solution outperforms LASSO–FQI when the number of tasks increases. In particular we can see that when $T = 10$, the term $\log(d)/\sqrt{T}$ remains small and the performance of GL–FQI remains stable.

In the second experiment, we introduced a rotation in the features, by randomly generating an orthonormal matrix U. This rotation combines the RBFs and the noise, and $\tilde{s}$ grows, although the rank s^* remains small. Results are reported in Fig. 3, where, as expected, the low rank approximation found by FL–FQI is able to solve the tasks much better than GL–FQI, which assumes joint sparsity. Moreover, we can see that the stability to the number of noisy dimensions grows when T increases, but not as much as in the first experiment.

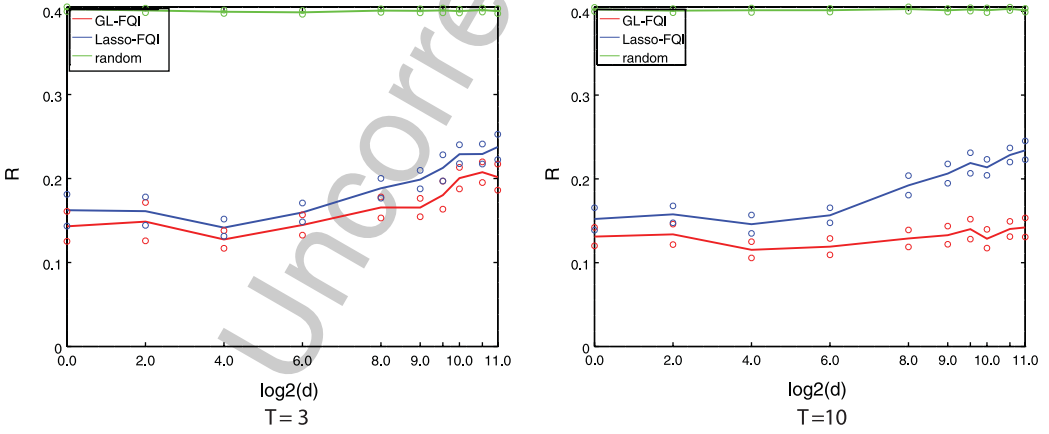


Fig. 2. Results of the first experiment in the chain walk domain comparing GL–FQI and LASSO–FQI. On the y axis we have the average regret computed according to Equation (16). On the x axis we have the total number of dimensions d, including noise dimensions, on a logarithmic scale. For each graph, T corresponds to the number of tasks learned at the same time in the experiment.

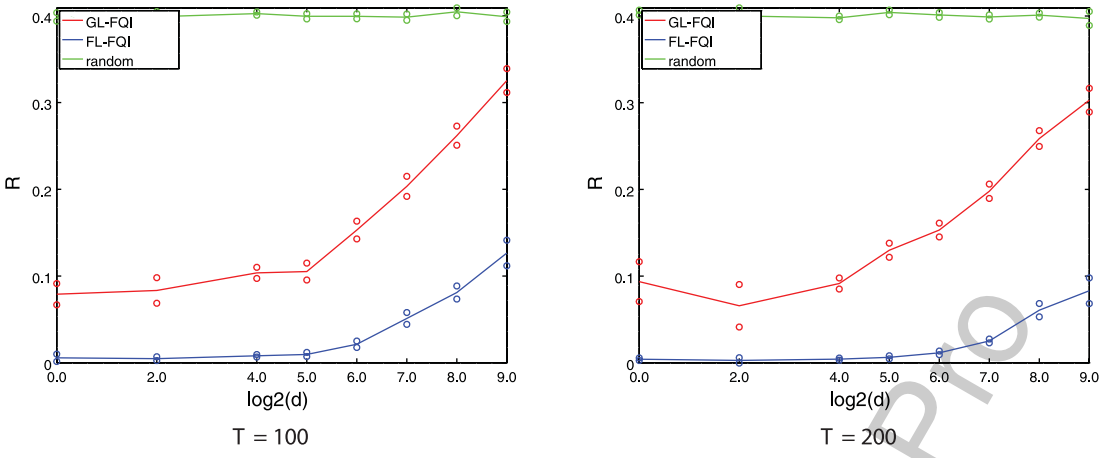


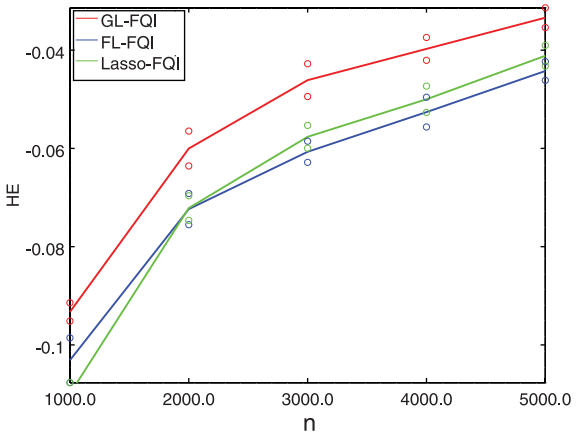
Fig. 3. Results of the second experiment in the chain walk domain comparing GL–FQI and FL–FQI. On the y axis we have the average regret computed according to Equation (16). On the x axis we have the total number of dimensions d, including noise dimensions, on a logarithmic scale. For each graph, T corresponds to the number of tasks learned at the same time in the experiment.

6.2. Black jack

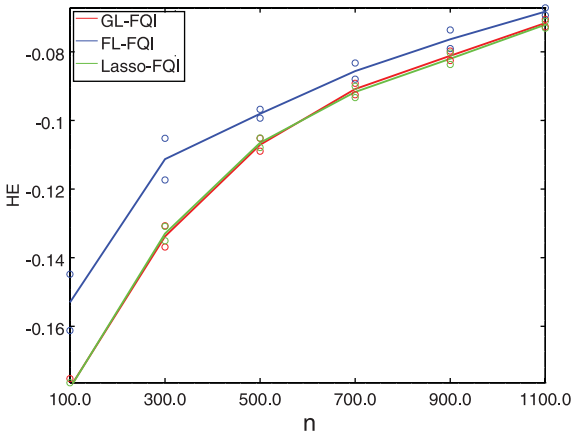
We consider two variants of the more challenging blackjack domain. In both variants the player can choose to *hit* to obtain a new card or *stay* to end the episode, while the two settings differ in the possibility of performing a *double* (doubling the bet) on the first turn. We refer to the variant with the *double* option as the *full variant*, while the other is the *reduced variant*. After the player concludes the episode, the dealer hits until a fixed threshold is reached or exceeded. Different tasks can be defined depending on several parameters of the game, such as the number of decks, the threshold at which the dealer stays and whether she hits when the threshold is reached exactly with a *soft* hand.

Full variant experiment. In the first experiment we consider the full variant of the game. The tasks are generated by selecting 2, 4, 6, 8 decks, by setting the stay threshold at {16, 17} and whether the dealer hits on soft, for a total of 16 tasks. We define a very rich description of the state space with the objective of satisfying Asm. 1. At the same time this is likely to come with a large number of useless features, which makes it suitable for sparsification. In particular, we include the player hand value, indicator functions for each possible player hand value and dealer hand value, and a large description of the cards not dealt yet (corresponding to the history of the game), under the form of indicator functions for various ranges. In total, the representation contains $d = 212$ features. We notice that although none of the features is completely useless (according to the definition in Asm. 2), the features related with the history of the game are unlikely to be very useful for most of the tasks defined in this experiment. We collect samples from up to 5000 episodes, although they may not be representative enough given the large state space of all possible histories that the player can encounter and the high stochasticity of the game. The evaluation is performed by simulating the learned policy for 2,000,000 episodes and computing the average House Edge (HE) across tasks. For each algorithm we report the performance for the best regularization parameter λ in the range {2, 5, 10, 20, 50}. Results are reported in Fig. 4a. Although the set of features is quite large, we notice that all the algorithms succeed in learning a good policy even with relatively few samples, showing that all of them can take advantage of the sparsity of the representation. In particular, GL–FQI exploits the fact that all 16 tasks share the same useless features (although the set of useful feature may not overlap entirely) and its performance is the best. On the other hand, FL–FQI suffers from the increased complexity of representation learning, which in this case does not lead to any benefit since the initial representation is already sparse. Nonetheless, it is interesting to note that the performance of FL–FQI is comparable to single-task LASSO–FQI.

Reduced variant experiment. In the second experiment we construct a representation for which we expect the weight matrix to be dense. In particular, we only consider the value of the player's hand and of the dealer's hand and we generate features as the Cartesian product of these two discrete variables plus a feature indicating whether the hand is soft, for a total of 280 features. Similar to the previous setting, the tasks



(a) Full variant of blackjack



(b) Reduced variant of blackjack

Fig. 4. Results of the experiment comparing FL–FQI, GL–FQI and LASSO–FQI. On the y axis we have the average house edge (HE) computed across tasks. On the x axis we have the total number of episodes used for training.

are generated with 2, 4, 6, 8 decks, whether the dealer hits on soft, and a larger number of stay thresholds in {15, 16, 17, 18}, for a total of 32 tasks. We used regularizers in the range {0.1, 1, 2, 5, 10}. Since the history is not included, the different number of decks influences only the probability distribution of the totals. Moreover, limiting the actions to either *hit* or *stay* further increases the similarity among tasks. Therefore, we expect to be able to find a dense, low-rank solution. The results in Fig. 4b confirms this guess, with FL–FQI performing significantly better than the other methods. In addition, GL–FQI and LASSO–FQI perform similarly, since the dense representation penalizes both single-task and shared sparsity. This was also observed by the fact that both methods favor low values of λ, indicating that the sparse-inducing penalties are not effective.

7. Conclusions

We studied the problem of multi-task reinforcement learning under shared sparsity assumptions across the tasks. GL–FQI extends the FQI algorithm by introducing a Group-LASSO step at each iteration and it leverages over the fact that all the tasks are expected to share the same small set of useful features to improve the performance of single-task learning. Whenever the assumption is not valid, GL–FQI may perform worse than LASSO–FQI. With FL–FQI we take a step further and we learn a transformation of the given representation that could guarantee a higher level of shared sparsity. This also corresponds to find a low-rank approximation and to identify a set of *core* tasks that can be used as a basis for learning all the other tasks. While the theoretical guarantees derived for the presented methods provide a solid argument for their soundness, preliminary empirical results suggest that they could be a useful alternative to single-task learning in practice. Future work will be focused on providing a better understanding and a relaxation of the theoretical assumptions and on studying alternative multi-task regularization formulations such as in [31] and [14].

Acknowledgments

This work was supported by the French Ministry of Higher Education and Research, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 270327 (project CompLACS), and the French National Research Agency (ANR) under project ExTra-Learn n.ANR-14-CE24-0010-01.

References

[1] A. Argyriou, T. Evgeniou and M. Pontil, Convex multi-task feature learning, *Machine Learning* **73**(3) (2008), 243–272.
[2] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.
[3] P.J. Bickel, Y. Ritov and A.B. Tsybakov, Simultaneous analysis of lasso and dantzig selector, *The Annals of Statistics*, 2009, pp. 1705–1732.
[4] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 1st edition, 2011.

[5] D. Calandriello, A. Lazaric and M. Restelli, Sparse Multi-task Reinforcement Learning. In https://hal.inria.fr/hal-01073513, 2014.

[6] A. Castelletti, S. Galelli, M. Restelli and R. Soncini-Sessa, Treebased feature selection for dimensionality reduction of largescale control systems. *In 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2011, pp. 11–15.

[7] D. Ernst, P. Geurts, L. Wehenkel and M.L. Littman, Tree-based batch mode reinforcement learning, *Journal of Machine Learning Research* **6**(4), 2005.

[8] A.M. Farahmand, R. Munos and C. Szepesvári, Error propagation for approximate policy and value iteration. In NIPS, 2010, pp. 568–576.

[9] M. Ghavamzadeh, A. Lazaric, R. Munos, M. Hoffman, et al. Finite-sample analysis of lasso-td, *In International Conference on Machine Learning*, 2011.

[10] S. Grunewalder, G. Lever, L. Baldassarre, M. Pontil and A. Gretton, Modelling transition dynamics in mdps with rkhs embeddings, *In Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, (2012).

[11] H. Hachiya and M. Sugiyama, Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information. *In Machine Learning and Knowledge Discovery in Databases*, 2010.

[12] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning*, Springer, 2009.

[13] M. Hoffman, A. Lazaric, M. Ghavamzadeh and R. Munos, Regularized least squares temporal difference learning with nested ℓ_2 and ℓ_1 penalization. In *EWRL*, 2012, pp. 102–114.

[14] L. Jacob, G. Obozinski and J.-P. Vert, Group lasso with overlap and graph lasso. *In Proceedings of the International Conference on Machine Learning*, ACM, 2009, pp. 433–440.

[15] J.Z. Kolter and A.Y. Ng, Regularization and feature selection in least-squares temporal difference learning. *In Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[16] A. Lazaric, Transfer in reinforcement learning: a framework and a survey. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*. Springer, 2011.

[17] A. Lazaric and M. Ghavamzadeh, Bayesian multi-task reinforcement learning. *In Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML-2010)*, 2010.

[18] A. Lazaric and M. Restelli, Transfer from multiple MDPs. *In Proceedings of the Twenty-Fifth Annual Conference on Neural Information Processing Systems (NIPS'11)*, 2011.

[19] H. Li, X. Liao and L. Carin, Multi-task reinforcement learning in partially observable stochastic environments, *Journal of Machine Learning Research* **10** (2009), 1131–1186.

[20] K. Lounici, M. Pontil, S. Van De Geer, A.B. Tsybakov, et al. Oracle inequalities and optimal inference under group sparsity, *The Annals of Statistics* **39**(4) (2011), 2164–2204.

[21] R. Munos and C. Szepesvári, Finite-time bounds for fitted value iteration, *The Journal of Machine Learning Research* **9** (2008), 815–857.

[22] S. Negahban, M.J. Wainwright, et al. Estimation of (near) low-rank matrices with noise and high-dimensional scaling, *The Annals of Statistics* **39**(2) (2011), 1069–1097.

[23] C. Painter-Wakefield and R. Parr, Greedy algorithms for sparse reinforcement learning, In *ICML*, 2012.

[24] B. Scherrer, V. Gabillon, M. Ghavamzadeh and M. Geist, Approximate modified policy iteration, *In Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 – July 1 2012*, 2012.

[25] M. Snel and S. Whiteson, Multi-task reinforcement learning: Shaping and feature selection. *In Proceedings of the European Workshop on Reinforcement Learning (EWRL)*, 2011.

[26] R.S. Sutton and A.G. Barto, *Introduction to reinforcement learning*, MIT Press, 1998.

[27] F. Tanaka and M. Yamamura, Multitask reinforcement learning on the distribution of mdps. *In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2003)*, 2003, pp. 1108–1113.

[28] M.E. Taylor and P. Stone, Transfer learning for reinforcement learning domains: A survey, *Journal of Machine Learning Research* **10**(1) (2009), 1633–1685.

[29] S.A. Van De Geer, P. Bühlmann, et al. On the conditions used to prove oracle results for the lasso, *Electronic Journal of Statistics* **3** (2009), 1360–1392.

[30] A. Wilson, A. Fern, S. Ray and P. Tadepalli, Multi-task reinforcement learning: A hierarchical Bayesian approach. *In Proceedings of ICML 24*, 2007, pp. 1015–1022.

[31] Y. Zhang and J.G. Schneider, Learning multiple tasks with a sparse matrix-normal penalty. In *NIPS*, 2010, pp. 2550–2558.