What the hell is the title of this paper?

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

Recent advancements have allowed reinforcement learning (RL) [17] to achieve impressive results in a wide variety of complex tasks, ranging from Atari [13], to the game of Go [16], to the control of sophisticated robotics systems [5, 11, 10]. The main limitation is that RL algorithms still require an enormous amount of experience samples before successfully learning such complicated tasks. One of the most promising solutions is transfer learning, which focuses on reusing past knowledge available to the agent in order to reduce the sample-complexity for learning new tasks. In the typical settings of transfer in RL [19], the agent is assumed to have already solved a set of *source tasks* generated from some unknown distribution. Then, given a *target task* drawn from the same distribution, or a slightly different one, the agent can rely on knowledge from the source tasks to speed up the learning process. This constitutes a significant advantage over plain RL, where the agent learns each new task from scratch independently of previous learning experience. Several algorithms have been proposed in the literature to transfer experience samples [9, 18], policies/options [3, 7], rewards [6], features [1], parameters [2, 4], and so on. We refer the reader to [19] for a thorough survey on transfer in RL.

Under the assumption that tasks follow a certain distribution, an intuitive choice for designing a transfer algorithm is to attempt at characterizing the uncertainty over the target task. Then, an ideal algorithm would leverage prior knowledge from the source tasks to interact with the target task in such a way that this uncertainty is reduced as quickly as possible. This simple intuition makes Bayesian methods appealing approaches for transfer in RL, and many previous works have been proposed in this direction. [20] assume tasks share similarities in their dynamics and rewards and propose a hierarchical Bayesian model for the distribution of these two elements. Similarly, [8] assume tasks are similar in their value functions and design a different hierarchical Bayesian model for transferring such information. More recently, [2], and its extension [4], consider tasks whose dynamics are governed by some hidden parameters, and propose efficient Bayesian models for quickly learning such parameters in new tasks. However, all these algorithms require specific, and sometimes restrictive, assumptions (e.g., on the distributions involved or the function approximators adopted), which might limit their practical applicability. The importance of having transfer algorithms that alleviate the need of strong assumptions and that can be easily adapted to different contexts motivates us to take a more general approach.

probabilistic? distributional?

Similarly to [8], we assume tasks to share similarities in their value functions and use the given source tasks to learn a distribution over such functions. Then, we use this distribution as a prior for learning the target task and we propose and efficient variational approximation of the corresponding posterior. Leveraging on recent ideas from randomized value functions [14], we design a Thompson Sampling-based algorithm which efficiently explores the target task by repeatedly sampling from the posterior

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

and acting greedily w.r.t. (with respect to) the sampled value function. We show that our approach is very general, in the sense that it does not require any specific choice of function approximator or prior/posterior distribution models. We propose a finite-sample analysis that theoretically validates our approach, while empirically evaluating its performance on three domains with increasing level of difficulty.

2 Preliminaries

We consider a distribution $\mathcal{D}$ over tasks, where each task is modeled as a discounted Markov Decision Process (MDP). We define an MDP as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, p_0, \gamma \rangle$, where $\mathcal{S}$ is the state-space, $\mathcal{A}$ is a finite set of actions, $\mathcal{P}(\cdot|s,a)$ is the distribution of the next state s' given that action a is taken in state s, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, p_0 is the initial-state distribution, and $\gamma \in [0,1)$ is the discount factor. We assume the reward function to be uniformly bounded by a constant $R_{max} > 0$. A deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ is a mapping from states to actions. At the beginning of each episode of interaction, the initial state s_0 is drawn from p_0. Then, the agent takes the action $a_0 = \pi(s_0)$, receives a reward $\mathcal{R}(s_0, a_0)$, transitions to the next state $s_1 \sim \mathcal{P}(\cdot|s_0, a_0)$, and the process is repeated. The goal is to find the policy maximizing the long-term return over a possibly infinite horizon: $\max_\pi J(\pi) \triangleq \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t \mid \mathcal{M}, \pi]$. To this end, we define the optimal value function $Q^*(s,a)$ as the expected return obtained by taking action a in state s and following an optimal policy thereafter. Then, an optimal policy π^* is a policy that is greedy with respect to the optimal value function, i.e., $\pi^*(s) = \operatorname{argmax}_a Q^*(s,a)$ for all states s. It can be shown (e.g., [15]) that Q^* is the unique fixed-point of the optimal Bellman operator T defined by $TQ(s,a) = \mathcal{R}(s,a) + \gamma \mathbb{E}_{\mathcal{P}}[\max_{a'} Q(s',a')]$ for any value function Q. From now on, we adopt the term Q-function to denote any plausible value function, i.e., any function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ uniformly bounded by $\frac{R_{max}}{1-\gamma}$.

Should we provide a more transfer-oriented definition?

We consider a parametric family of Q-functions, $\mathcal{Q} = \left\{ Q_{\boldsymbol{w}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \mid \boldsymbol{w} \in \mathbb{R}^K \right\}$, and we assume each function in $\mathcal{Q}$ to be uniformly bounded by $\frac{R_{max}}{1-\gamma}$. When learning the optimal value function, a quantity of interest is how close a given function $Q_{\boldsymbol{w}}$ is to the fixed-point of the Bellman operator. A possible measure is its Bellman error (or Bellman residual), defined by $B(\boldsymbol{w}) \triangleq TQ_{\boldsymbol{w}} - Q_{\boldsymbol{w}}$. Notice that $Q_{\boldsymbol{w}}$ is optimal if, and only if, $B(\boldsymbol{w})(s,a) = 0$ for all s, a. If we assume the existence of a distribution ν over $\mathcal{S} \times \mathcal{A}$, an appealing objective is to directly minimize the squared Bellman error of $Q_{\boldsymbol{w}}$ under ν, denoted by $\|B(\boldsymbol{w})\|_\nu^2$. Unfortunately, it is well-known that an unbiased estimator of this quanitity requires two independent samples of the next state s' for each s, a (e.g., [12]). In practice, the Bellman error is typically replaced by the TD error $b(\boldsymbol{w})$, which approximates the former using a single transition sample, $b(\boldsymbol{w}) = r + \gamma \max_{a'} Q_{\boldsymbol{w}}(s',a') - Q_{\boldsymbol{w}}(s,a)$. Finally, given a dataset $D = \langle s_i, a_i, r_i, s'_i \rangle_{i=1}^N$ of N samples, the squared TD error is computed as $\|B(\boldsymbol{w})\|_D^2 = \frac{1}{N}\sum_{i=1}^N (r_i + \gamma \max_{a'} Q_{\boldsymbol{w}}(s'_i, a') - Q_{\boldsymbol{w}}(s_i, a_i))^2 = \frac{1}{N}\sum_{i=1}^N b_i(\boldsymbol{w})^2$. Whenever the distinction is clear from the context, with slight abuse of terminology, we refer to the squared Bellman error and squared TD error as Bellman error and TD error, respectively.

Should we define norms before?

3 Variational Transfer Learning

In this section, we describe our variational approach to transfer in RL. In Section 3.1, we start by introducing our algorithm from a high-level perspective, in such a way that it can be used for any choice of prior and posterior distributions. Then, in Sections 3.2 and 3.3, we propose practical implementations based on Gaussians and mixtures of Gaussians, respectively. We conclude with some considerations on how to optimize the proposed objective in Sec. 3.4.

3.1 Algorithm

We begin with a simple consideration: the distribution $\mathcal{D}$ over tasks clearly induces a distribution over optimal Q-functions. Since, for any MDP, learning its optimal Q-function is sufficient for solving the problem, one can safely replace the distribution over tasks with the distribution over their optimal value functions. In particular, we reduce the latter to a distribution $p(\boldsymbol{w})$ over weights.

over the parameters of the function approximator, perhaps?

Assume that we are given a dataset $D = \{(s_i, a_i, s'_i, r_i) \mid i = 1, 2, \ldots N\}$ of samples from some task τ that we want to solve. Then, the posterior distribution over weights given such dataset can be computed by applying Bayes theorem as $p(\boldsymbol{w}|D) \propto p(D|\boldsymbol{w})p(\boldsymbol{w})$. Unfortunately, this cannot be directly used in practice since we do not have a model of the likelihood $p(D|\boldsymbol{w})$. In such case, it is very common to make strong assumptions on the MDPs or the Q-functions so as to get tractable posteriors []. However, in our transfer settings all distributions involved depend on the family of tasks under consideration, and making such assumptions is likely to limit the applicability of the approach. Thus, we take a different approach to derive a more general and meaningful solution. Recall that our final goal is to move all probability mass over the weights minimizing some empirical loss measure, which in our case is the TD error $\|B(\boldsymbol{w})\|_D^2$. Then, given a prior $p(\boldsymbol{w})$, we know from PAC-Bayesian theory that the optimal Gibbs posterior takes the form []:

Cite somebody

Cite Catoni 2007

$$q(\boldsymbol{w}) = \frac{e^{-\Lambda \|B(\boldsymbol{w})\|_D^2} p(\boldsymbol{w})}{\int e^{-\Lambda \|B(\boldsymbol{w}')\|_D^2} p(d\boldsymbol{w}')} \tag{1}$$

for some parameter $\Lambda > 0$. Since Λ is typically chosen to increase with the number of samples N, in the remaining we set it to $\lambda^{-1}N$, for some constant $\lambda > 0$. Notice that, whenever the term $e^{-\Lambda \|B(\boldsymbol{w})\|_D^2}$ can be interpreted as the actual likelihood of D, q becomes a classic Bayesian posterior. Although we now have an appealing distribution, the integral at the denominator of (1) is intractable to compute even for simple Q-function models. Thus, we propose a variational approximation $q_{\boldsymbol{\xi}}$ by considering a simpler family of distributions parameterized by $\boldsymbol{\xi} \in \Xi$. Then, our problem reduces to finding the variational parameters $\boldsymbol{\xi}$ such that $q_{\boldsymbol{\xi}}$ minimizes the Kullback-Leibler (KL) divergence w.r.t. the Gibbs posterior q. From the theory of variational inference (e.g., []), this can be shown to be equivalent to minimizing the well-known (negative) *evidence lower bound* (ELBO):

cite

$$\min_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{w} \sim q_{\boldsymbol{\xi}}} \left[\|B(\boldsymbol{w})\|_D^2 \right] - \frac{\lambda}{N} KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \mid\mid p(\boldsymbol{w})\right) \tag{2}$$

Intuitively, the approximate posterior balances the trade-off between placing probability mass over those weights $\boldsymbol{w}$ that have low TD error (first term), and staying close to the prior distribution (second term). Assuming that we are able to compute the gradients of (2) w.r.t. the variational parameters $\boldsymbol{\xi}$, our objective can be easily optimized with any stochastic optimization algorithm.

We now highlight our general transfer procedure in Alg. 1, while deferring a description of practical implementations with specific choices for the distributions involved to the next two sections. Given a set of weights $\mathcal{W}_s$ from the source tasks, we start by estimating the prior distribution (line 1) and we initialize the variational parameters by minimizing the KL divergence w.r.t. such distribution[1] (line 2). Then, at each time step of interaction, we re-sample the weights from the current approximate posterior and act greedily w.r.t. the corresponding Q-function (lines 7,8). This resembles the well-known Thompson Sampling approach adopted in multi-armed bandits []and allows our algorithm to efficiently explore the target task. In some sense, at each time we guess what is the task we are trying to solve based on our current belief and we act as if such guess were actually true. After collecting and storing the new experience (lines 9-11), we draw a batch of samples from the replay buffer and a batch of weights from the posterior (line 12). We use these to approximate the negative ELBO, compute its gradient, and finally update the variational parameters (lines 13-15).

Cite

The main advantage of our approach is that it exploits knowledge from the source tasks to perform an efficient adaptive exploration. Intuitively, during the first steps of interaction, our algorithm has no idea about what is the current task. However, it can rely on the learned prior to take early informed decisions. As the learning process goes on, it will quickly figure out which task is being solved, thus moving all probability mass over the weights minimizing the TD error. From that point, sampling from the posterior is approximately equivalent to deterministically taking such weights, and no more exploration will be performed. Finally, notice the generality of the proposed approach: as far as the objective $\mathcal{L}$ is differentiable in the variational parameters $\boldsymbol{\xi}$, and its gradients can be efficiently computed, any function approximator for the Q-functions and any family for the prior and posterior distributions can be adopted. For the latter, we describe two practical choices in the next two sections.

3.2 Gaussian Variational Transfer

We now restrict ourselves to a specific choice of the prior and posterior families that makes our algorithm very efficient and easy to implement. We assume that optimal Q-functions according to our

[1] If the prior and approximate posterior were in the same family of distributions we could simply set $\boldsymbol{\xi}$ to the prior parameters, however this does not always hold in practice.

Algorithm 1 Variational Transfer

Require: Target task τ, source Q-function weights $\mathcal{W}_s$, batch sizes M_D and $M_{\mathcal{W}}$, prior weight λ

```
Estimate prior $p(\boldsymbol{w})$ from $\mathcal{W}_s$
Initialize variational parameters: $\boldsymbol{\xi} \leftarrow \operatorname{argmin}_{\boldsymbol{\xi}} KL(q_{\boldsymbol{\xi}}||p)$
Initialize replay buffer: $D = \emptyset$
repeat
    Sample initial state: $s_0 \sim p_0^{(\tau)}$
    while $s_h$ is not terminal do
        Sample weights: $\boldsymbol{w} \sim q_{\boldsymbol{\xi}}(\boldsymbol{w})$
        Take action $a_h = \operatorname{argmax}_a Q_{\boldsymbol{w}}(s_h, a)$
        Observe transition $s_{h+1} \sim \mathcal{P}^{(\tau)}(\cdot|s_h, a_h)$
        Collect reward $r_h = \mathcal{R}^{(\tau)}(s_h, a_h)$
        Add sample to the replay buffer: $D \leftarrow D \cup \langle s_h, a_h, r_h, s_{h+1}\rangle$
        Sample batch $D' = \langle s_i, a_i, r_i, s'_i\rangle_{i=1}^{M_D}$ from $D$ and $\mathcal{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_{M_{\mathcal{W}}}\}$ from $q_{\boldsymbol{\xi}}$
        Approximate objective: $\mathcal{L}(\boldsymbol{\xi}) = \frac{1}{M_{\mathcal{W}}} \sum_{\boldsymbol{w} \in \mathcal{W}} \|B(\boldsymbol{w})\|_{D'}^2 - \frac{\lambda}{N} KL\left(q_{\boldsymbol{\xi}} \mid\mid p\right)$
        Compute the gradient $\nabla_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\xi})$
        Update $\boldsymbol{\xi}$ in the direction of $\nabla_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\xi})$ using any stochastic optimizer (e.g., ADAM)
    end while
until forever
```

task distribution (or better, their weights) follow a multivariate Gaussian law. That is, we model the prior as $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and we learn its parameters from the set of source weights using, e.g., maximum likelihood estimation (with small regularization to make sure the covariance is positive definite). Then, our variational family is the set of all well-defined Gaussian distributions, i.e., the variational parameters are $\Xi = \left\{(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \mid \boldsymbol{\mu} \in \mathbb{R}^K, \boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}, \boldsymbol{\Sigma} \succ 0\right\}$. Additionally, to prevent the covariance from going not positive definite, we consider its Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ and learn the lower-triangular Cholesky factor L instead.

3.3 Mixture of Gaussian Variational Transfer

Although the Gaussian assumption of the previous section is very appealing as it allows for a simple and efficient way of computing the variational objective and its gradients, we believe that such assumption almost never holds in practice. In fact, even for families of tasks in which the reward and transition models follow a Gaussian law, the Q-values might be far from it. Depending on the family of tasks under consideration and, since we are learning a distribution over weights, on the chosen function approximator, the prior might have arbitrarily complex shapes. When the information loss due to the Gaussian approximation becomes too severe, the algorithm is likely too fail at transferring knowledge, thus reducing to almost random exploration. We now propose a variant to successfully solve this problem, while keeping the algorithm simple and efficient enough to be applied in practice. In order to capture arbitrarily complex distributions, we use a kernel estimator []for learning our prior. Assume we are given a set $\mathcal{W}_s$ of weights from the source tasks. Then, our estimated prior places a single isotropic Gaussian over each weight: $p(\boldsymbol{w}) = \frac{1}{|\mathcal{W}_s|} \sum_{\boldsymbol{w}_s \in \mathcal{W}_s} \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_s, \sigma_p^2 \boldsymbol{I})$[2]. This takes the form of a mixture of Gaussians with equally weighted components. Consistently with the prior, we model our approximate posterior as a mixture of Gaussians. However, we allow a different number of components (typically much less than the prior's) and we adopt full covariances instead of only diagonals, so that our posterior has the potential to match complex distributions with less components. Using C components, our posterior is $q_{\boldsymbol{\xi}}(\boldsymbol{w}) = \frac{1}{C} \sum_{i=1}^{C} \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, with variational parameters $\boldsymbol{\xi} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_C, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_C)$. Once again, we learn Cholesky factors instead of full covariances.

Although this new model has the potential to capture much more complex distributions, it poses a major complication: the KL divergence between two mixture of Gaussians is well-known to have no closed-form equation. To solve this issue, we can rely on an upper bound to such quantity, so that the negative ELBO we are optimizing still represents an upper bound on the KL between the approximate and true posterior. However, this turns out to be non-trivial as well. In fact, it is very easy to bound

[2] Notice that this is slightly different than the typical kernel estimator (e.g., [])

the KL between two mixtures with the KLs between each couple of components. However, the loss of information is such that minimizing the upper bound via gradient methods converges to a local optimum in which all components tend to go to the same point, thus almost reducing to the single Gaussian case. To solve this issue, we adopt the variational upper bound proposed in [], which we found to be able to preserve the needed information. We report it here for the sake of completeness. See the original paper for the proof.

Cite UB

Theorem 1. *Let $p = \sum_i c_i^{(p)} f_i^{(p)}$ and $q = \sum_j c_j^{(q)} f_j^{(q)}$ be two mixture of Gaussian distributions, where $f_i^{(p)} = \mathcal{N}(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)})$ denotes the i-th component of p, $c_i^{(p)}$ denotes its weight, and similarly for q. Introduce two vectors $\chi^{(1)}$ and $\chi^{(2)}$ such that $c_i^{(p)} = \sum_j \chi_{j,i}^{(2)}$ and $c_j^{(q)} = \sum_i \chi_{i,j}^{(1)}$. Then:*

$$KL(p||q) \leq KL(\chi^{(2)}||\chi^{(1)}) + \sum_{i,j} \chi_{j,i}^{(2)} KL(f_i^{(p)}||f_j^{(q)}) \tag{3}$$

Our new algorithm replaces the KL with the above-mentioned upper bound. Each time we require its value, we have to recompute the parameters $\chi^{(1)}$ and $\chi^{(2)}$ that tighten the bound. As shown in [], this can be achieved by a simple fixed-point procedure. Furthermore, both terms in the approximate negative ELBO are now linear combinations of functions of the variational parameters for different components, thus their gradients can be straightforwardly derived from the ones of the Gaussian case.

cite them again

3.4 Optimizing the TD error

From Sections 3.2 and 3.3, we know that differentiating the negative ELBO $\mathcal{L}$ w.r.t. $\boldsymbol{\xi}$ requires differentiating $\|B(\boldsymbol{w})\|_D^2$ w.r.t. $\boldsymbol{w}$. Unfortunately, the TD error is well-known to be non-differentiable due to the presence of the max operator. This rarely represents a problem since typical value-based algorithms are actually semi-gradient methods, i.e., they do not differentiate the targets (see, e.g., Chapter 11 of []). However, our transfer settings are rather different than common RL. In fact, our algorithm is likely to always start from Q-functions that are very close to the optimum, and the only thing that needs to be done is to adapt the weights in a direction of lower error (i.e., higher likelihood) so as to quickly converge to the solution of the task that is being solved. Unfortunately, this property cannot be guaranteed for most semi-gradient algorithms. Even worse, many online RL algorithms combined with complex function approximators (e.g., DQNs) are well-known to be unstable, especially when approaching the optimum, and require a lot of tuning and tricks to work well. This is obviously an undesirable property in our case, as we only aim at adapting already good solutions. Thus, we consider using a residual gradient algorithm (after []). In order to differentiate the targets, we replace the optimal Bellman operator with the mellow Bellman operator introduced in [], which adopts a softened version of max called *mellowmax*:

Cite Sutton

Baird

Cite MM

$$\operatorname*{mm}_a Q_{\boldsymbol{w}}(s,a) = \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_a e^{\kappa Q_{\boldsymbol{w}}(s,a)} \tag{4}$$

where κ is a hyperparameter and $|\mathcal{A}|$ is the number of actions. The mellow Bellman operator, which we denote as $\widetilde{T}$, has several appealing properties that make it suitable for our settings: (i) it converges to the maximum as $\kappa \to \infty$, (ii) it has a unique fixed point, and (iii) it is *differentiable*. Denoting by $\widetilde{B}(\boldsymbol{w}) = \widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}$ the Bellman residual w.r.t. the mellow Bellman operator $\widetilde{T}$, we have that the corresponding TD error, $\left\|\widetilde{B}(\boldsymbol{w})\right\|_D^2$, is now differentiable with respect to $\boldsymbol{w}$. Although residual algorithms have the sought guaranteed convergence, they are typically much slower than their semi-gradient counterpart. This problem was addressed in [], where the authors proposed a simple remedy consisting in projecting the gradient in a direction that sill guarantees convergence but is also closer to the one of the semi-gradient, thus achieving higher learning speed. This can be easily done by including a parameter $\psi \in [0,1]$ in the TD error gradient such that:

Baird again

$$\nabla_{\boldsymbol{w}} \left\|\widetilde{B}(\boldsymbol{w})\right\|_D^2 = \frac{2}{N} \sum_{i=1}^{N} b_i(\boldsymbol{w}) \left(\gamma\psi\nabla_{\boldsymbol{w}} \operatorname*{mm}_{a'} Q_{\boldsymbol{w}}(s_i', a') - \nabla_{\boldsymbol{w}} Q_{\boldsymbol{w}}(s_i, a_i)\right) \tag{5}$$

where $b_i(\boldsymbol{w}) = r_i + \gamma \operatorname{mm}_{a'} Q_{\boldsymbol{w}}(s_i', a') - Q_{\boldsymbol{w}}(s_i, a_i)$. Notice that ψ trades-off between the semi-gradient ($\psi = 0$) and the full residual gradient ($\psi = 1$). A good criterion for choosing such parameter is to start with values close to zero (to have faster learning) and move to higher values when approaching the optimum (to guarantee converge). Since in our case we are likely to start from this latter point, we consider only higher values (e.g., above 0.5).

4 Theoretical Analysis

In this section, we theoretically analyze our variational transfer algorithm...

A first important question that we need to answer is whether replacing max with mellow-max in the Bellman operator constitutes a strong approximation or not. It has been proved [] that the mellow Bellman operator is a contraction under the L_∞-norm and, thus, has a unique fixed-point. However, how such fixed-point differs from the one of the optimal Bellman operator remains an open question. Since mellow-max monotonically converges to max as $\kappa \to \infty$, it would be desirable if the corresponding operator also monotonically converged to the optimal one. We confirm that this property actually holds in the following theorem.

Cite MM

Theorem 2. *Let V be the fixed-point of the optimal Bellman operator T, and Q the corresponding action-value function. Define the action-gap function $g(s)$ as the difference between the value of the best action and the second best action at each state s. Let $\widetilde{V}$ be the fixed-point of the mellow Bellman operator $\widetilde{T}$ with parameter $\kappa > 0$ and denote by $\beta > 0$ the inverse temperature of the induced Boltzmann distribution (as in []). Let ν be a probability measure over the state-space. Then, for any $p \geq 1$:*

Cite MM

$$\left\|V - \widetilde{V}\right\|_{\nu,p}^{p} \leq \frac{2R_{max}}{(1-\gamma)^2}\left\|1 - \frac{1}{1+|\mathcal{A}|\,e^{-\beta g}}\right\|_{\nu,p}^{p} \tag{6}$$

Theorem 3. *Let Q^* be the fixed-point of the optimal Bellman operator T. Define the action-gap function $g(s)$ as the difference between the value of the best action and the second best action at each state s. Let $\widetilde{Q}$ be the fixed-point of the mellow Bellman operator $\widetilde{T}$ with parameter $\kappa > 0$ and denote by $\beta_\kappa > 0$ the inverse temperature of the induced Boltzmann distribution (as in []). Let ν be a probability measure over the state-action space. Then, for any $p \geq 1$:*

Cite MM

$$\left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^{p} \leq \frac{2\gamma R_{max}}{(1-\gamma)^2}\left\|\frac{1}{1+\frac{1}{|\mathcal{A}|}e^{\beta_\kappa g}}\right\|_{\nu,p}^{p} \tag{7}$$

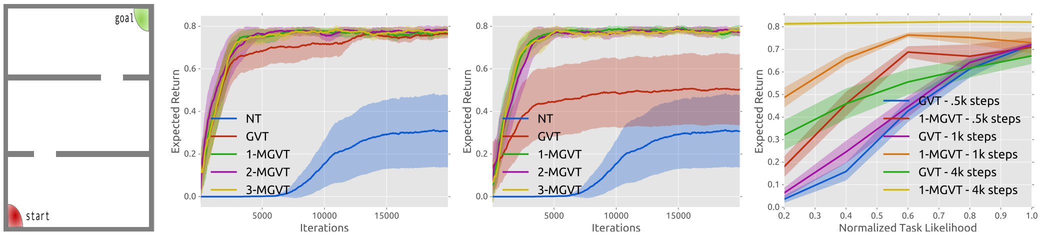
5 Related Works

Our approach is mostly related to []. Although we both assume the tasks to share similarities in their value functions, [] consider only linear approximators and adopt a hierarchical Bayesian model of the corresponding weights' distribution, which is assumed Gaussian. On the other hand, our variational approximation allows for more general distribution families and can be combined with non-linear approximators. Furthermore, [] propose a Dirichlet process model for the case where weights cluster into different classes, which relates to our mixture formulation of Sec. 3.3 and proves again the importance of capturing more complicated task distributions. Another related approach is [], where the authors propose a hierarchical Bayesian model for the distribution over tasks. Differently from our approach and [], they consider a distribution over transition probabilities and rewards, rather than value functions. In the same spirit of our method, they consider a Thompson sampling-based procedure which, at each iteration, samples a new task from the posterior and solves it, thus providing efficient exploration via the transferred knowledge. However, [] consider only finite MDPs, which poses a severe limitation in the algorithm's applicability. On the contrary, our approach can handle high dimensional tasks by allowing powerful function approximators.

Cite also HIP

Should we add more here or just leave a quick citation in the introduction?

The idea of exploration via value function randomization we build upon was first proposed in the RL community by []. The authors extend the well-known LSTD [] by adopting Bayesian linear regression to model the uncertainty over the predicted value function weights, and use that to perform a Thompson sampling-like procedure. Such approach was recently extended by [], where the approximator is replaced by a Bayesian neural network, leading to an algorithm capable of solving much more complicated problems. Both these algorithms rely on the Gaussian assumption and, since they work in plain RL settings, have no informative prior available. On the other hand, our variational approximation allows more complex distributions (e.g., mixtures) to be adopted, while knowledge from the source tasks allows us to learn very informative priors. Finally, the variational approximation technique we consider in this work has been previously adopted in [] to approximate



(a) (b) (c) (d)

Figure 1: (a) the rooms environment, (b) transfer from 10 source tasks with both doors moving, (c) transfer from 10 source tasks with only one door moving, and (d) transfer performance as a function of how likely the target task is according to the prior.

an intractable posterior over the transition dynamics, which is then used to drive efficient exploration via an information-maximizing procedure.

6 Experiments

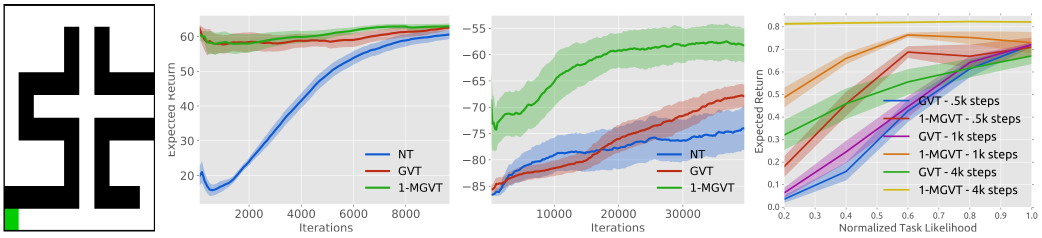
In this section, we provide an experimental evaluation of our approach. We begin by analyzing the behavior of our algorithm on a toy problem in Sec. 6.1. Then, in Sec. 6.2, we consider two classic control benchmarks, namely Cartpole and Mountain Car. We conclude with a complex maze navigation task in Sec. 6.3. In all experiments, we compare our Gaussian variational transfer algorithm (GVT) and the version using mixture of Gaussians with c components (c-MGVT) to plain no-transfer RL (NT) with ϵ-greedy exploration. To the best of our knowledge, no existing transfer algorithm is directly comparable to our approach from an experimental perspective.

Any better motivation?

6.1 The Rooms Problem

We consider an agent navigating in the rooms environment depicted in Fig. **??**. The agent starts in the bottom-left corner and must move from one room to another to reach the goal position in the top-right corner. The rooms are connected by small doors whose positions are unknown to the agent. The state-space is modeled as a 10×10 continuous grid, while the action-space is the set of 4 movement directions (up, right, down, left). After each action, the agent moves by 1 in the chosen direction and the final position is corrupted by Gaussian noise with 0.2 standard deviation. In case the agent hits a wall, its position remains unchanged. The reward is 1 when reaching the goal (after which the process terminates) and 0 otherwise, while the discount factor is $\gamma = 0.99$. We consider a distribution over tasks in which doors have a fixed width of 1, while their positions are sampled uniformly in $[0.5, 9.5]$. Since the agent does not know where the doors are located in advance and receives only very sparse feedback, it must efficiently explore the environment to figure out (i) their positions, and (ii) how to reach the goal. While this might be a complicated problem for plain RL, our transfer algorithm should be able to quickly figure out the door positions. In fact, notice that, although different, the optimal Q-functions for all tasks share some similarities. For example, once the agent has passed the last door before the goal, the Q-values are exactly the same in all tasks. This does not hold for positions nearby the start state. However, it is clear that there should be a preference over actions up and right, rather than down and left (which are worse in all tasks). Thus, we expect our algorithm to efficiently explore any target task.

In order to prove that our guesses are correct, we generate a set of 50 source tasks for the three-room environment of Fig. **??** by sampling both door positions uniformly, and solve all of them by directly minimizing the TD error as presented in Sec. 3.4. In order to make sure that their solutions are accurate enough, we allow sampling the initial state uniformly in the environment and run until convergence. Then, we use our algorithms to transfer from 10 source tasks sampled from the previously generated set. The average return over the last 50 learning episodes as a function of the number of iterations is shown in Fig. **??**. Each curve is the result of 20 independent runs, each resampling the target and source tasks. 95% confidence intervals are shown. Further details on the parameters adopted in this experiment are given in App. **??**. As expected, the no-transfer (NT) algorithm fails at learning the task in so few iterations due to the limited exploration provided by an



(a) (b) (c) (d)

Figure 2: (a) Sample Maze, (b) transfer from 10 sources in Cartpole, (c) transfer from 10 source tasks in Mountain Car, and (d) transfer from 10 sources in Mazes 0, 6, 13—see App. **??**

ϵ-greedy policy. On the other hand, all our algorithms achieve a significant speed-up and are able to converge to the optimal performance in few iterations, with GVT being slightly slower. Interestingly, we notice that there is no advantage in adopting more than 1 component for the posterior in MGVT. This is intuitive since, when the algorithm quickly figures out which task is being solved, it will move all components in the same direction.

To better understand the differences between GVT and MGVT, we now consider transferring from a slightly different distribution than the one from which target tasks are drawn. We generate again 50 source tasks but this time with the bottom door fixed in the center and the other one moving. Then, we repeat the previous experiment, allowing both doors to move when sampling target tasks. The results are shown in Fig. **??**. Interestingly, MGVT seems almost unaffected by this change, proving that is has sufficient representation power to generalize to slightly different task distributions. The same does not hold for GVT, which now is not able to solve many of the sampled target tasks,thus the very high variance. This proves again that assuming Gaussian distributions can pose severe limitations in our transfer settings.

Finally, we analyze the transfer performance as a function of how likely the target task is according to the prior. We consider a two-room version of the environment of Fig. **??**. Differently from before, we generate tasks by sampling the door position from a Gaussian with mean 5, and standard deviation 1.8, so that tasks where the door is near the sides are very unlikely. Fig. **??** shows the performance reached by GVT and MGVT with 1 component at fixed iterations as a function of how likely the target task is according to such distribution. As expected GVT achieves poor performance on very unlikely tasks, even after many iterations. In fact, estimating a single Gaussian distribution definitely entails some information loss, especially about the unlikely tasks. On the other hand, MGVT keeps such information and, consequently, performs much better. Perhaps not surprisingly, MGVT reaches the optimal performance in $4k$ iterations no matter what task is being solved.

6.2 Classic Control

To evaluate the performance of our transfer algorithms in more common benchmarks, we chose to use the well-known classic control environments: Cartpole and Mountain Car as defined in [17]. In the case of Cartpole we generated 20 source tasks by uniformly sampling the cart mass, the pole mass and length. Instead, for Mountain Car, we generated the 20 sources by uniformly sampling the base speed of the car. We kept a discount factor of $\gamma = 0.99$ for both environments. Moreover, to train the source tasks we chose to use a Multilayer Perceptron (MLP) as approximator. We trained all source tasks until convergence and to transfer, in both settings, we used GVT and 1-MGVT choosing randomly 10 of the trained sources.

In Fig. **??**, we show the results of transfer in Cartpole from which it is easily noticed a near-optimal jump-start in both GVT and 1-MGVT that during the following iterations converges to the optimal performance. In the other hand, transfer in Mountain Car as shown in Fig. **??**, a more complicated task than Cartpole, causes GVT to struggle to solve the target task but 1-MGVT reaches convergence within the same number of iterations to the optimal performances.

6.3 Maze Navigation

For this setting we propose a maze environment, such as those in Fig. **??**, for a robotic agent to navigate. We created a set of 20 mazes of size 10×10 with continuous space within which we ensured four groups: each group with the goal position in one of the corners of the maze. In order to simulate a robotic agent, we set an action space that allows to control the linear and angular motion by a fixed distance of 0.5 (*move forward*) and a fixed angle of $\pm 22.5^{\text{o}}$ (*rotate counter- and clockwise*). Moreover, we enhanced the state space to include, in addition to the absolute position and orientation, a set of distance measures to the nearest obstacles from the agent's perspective with maximum range of 2 and a field of vision from -90^{o} to 90^{o} relative to its orientation. In this way, we provide the agent with information that a robot would normally collect from LIDARs, sonars and other sensors alike. Finally, in the same field of perception, we provide a boolean vector indicating whether the goal is within the range of observation. Regarding rewards, the agent gets 1 whenever it enters the goal area and 0 in any other situation and the MDP has a discount factor of $\gamma = 0.99$.

The main motivation for this set-up is that a robot is able to exploit locally-sensed information in addition to its estimation of the actual position—as opposed to the Rooms problem presented above—and it is such information that the robotic agent could learn to use and, also, exploit in different maze configurations leveraging its previous experience. Moreover, simple variations in the maze structure could, potentially, produce big differences in the value function with respect to the ones seen in previous tasks which makes the setting very good to assess how robust our algorithm is to negative transfer.

In App **??**he mazes used are enumerated. These were solved using Deep Q Networks (DQN) [] with a multilayer perceptron with hidden layers 32×32 and, in order to perform the knowledge transfer, the target maze would be excluded from the source tasks given to the algorithm. Finally, the MGVT algorithm, with one component for the posterior distribution, was used to solve the target tasks shown in Fig. *.

cite DQN?

7 Conclusion

References

[1] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065, 2017.

[2] Finale Doshi-Velez and George Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, page 1432. NIH Public Access, 2016.

[3] Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727. ACM, 2006.

[4] Taylor W Killian, Samuel Daulton, George Konidaris, and Finale Doshi-Velez. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, pages 6250–6261, 2017.

[5] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.

[6] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 489–496. ACM, 2006.

[7] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning.

[8] Alessandro Lazaric and Mohammad Ghavamzadeh. Bayesian multi-task reinforcement learning. In *ICML-27th International Conference on Machine Learning*, pages 599–606. Omnipress, 2010.

[9] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551. ACM, 2008.

[10] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[11] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[12] Odalric-Ambrym Maillard, Rémi Munos, Alessandro Lazaric, and Mohammad Ghavamzadeh. Finite-sample analysis of bellman residual minimization. In *Proceedings of 2nd Asian Conference on Machine Learning*, pages 299–314, 2010.

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[14] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

[15] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[17] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[18] Matthew E Taylor, Nicholas K Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 488–505. Springer, 2008.

[19] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

[20] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022. ACM, 2007.

A Proofs

Theorem 2. *Let V be the fixed-point of the optimal Bellman operator T, and Q the corresponding action-value function. Define the action-gap function $g(s)$ as the difference between the value of the best action and the second best action at each state s. Let $\widetilde{V}$ be the fixed-point of the mellow Bellman operator $\widetilde{T}$ with parameter $\kappa > 0$ and denote by $\beta > 0$ the inverse temperature of the induced Boltzmann distribution (as in []). Let ν be a probability measure over the state-space. Then, for any $p \geq 1$:*

Cite MM

$$\left\| V - \widetilde{V} \right\|_{\nu,p}^{p} \leq \frac{2R_{max}}{(1-\gamma)^2} \left\| 1 - \frac{1}{1 + |\mathcal{A}|\, e^{-\beta g}} \right\|_{\nu,p}^{p} \tag{6}$$

Proof. We begin by noticing that:

$$\begin{aligned} \left\| V - \widetilde{V} \right\|_{\nu,p}^{p} &= \left\| TV - \widetilde{T}\widetilde{V} \right\|_{\nu,p}^{p} \\ &= \left\| TV - \widetilde{T}V + \widetilde{T}V - \widetilde{T}\widetilde{V} \right\|_{\nu,p}^{p} \\ &\leq \left\| TV - \widetilde{T}V \right\|_{\nu,p}^{p} + \left\| \widetilde{T}V - \widetilde{T}\widetilde{V} \right\|_{\nu,p}^{p} \\ &\leq \left\| TV - \widetilde{T}V \right\|_{\nu,p}^{p} + \gamma \left\| V - \widetilde{V} \right\|_{\nu,p}^{p} \end{aligned}$$

where the first inequality follows from Minkowsky's inequality and the second one from the contraction property of the mellow Bellman operator. This implies that:

$$\left\| V - \widetilde{V} \right\|_{\nu,p}^{p} \leq \frac{1}{1-\gamma} \left\| TV - \widetilde{T}V \right\|_{\nu,p}^{p} \tag{8}$$

Let us bound the norm on the right-hand side separately. In order to do that, we will bound the function $\left| TV(s) - \widetilde{T}V(s) \right|$ point-wisely for any state s. By applying the definition of the optimal and mellow Bellman operators, we obtain:

$$\begin{aligned} \left| TV(s) - \widetilde{T}V(s) \right| &= \left| \max_a \{R(s,a) + \gamma \mathbb{E}\left[V(s')\right]\} - \operatorname*{mm}_a \{R(s,a) + \gamma \mathbb{E}\left[V(s')\right]\} \right| \\ &= \left| \max_a Q(s,a) - \operatorname*{mm}_a Q(s,a) \right| \end{aligned}$$

Recall that applying the mellow-max is equivalent to computing an expectation under a Boltzmann distribution with inverse temperature β induced by κ []. Thus, we can write:

Cite MM

$$\begin{aligned} \left| \max_a Q(s,a) - \operatorname*{mm}_a Q(s,a) \right| &= \left| \sum_a \pi^*(a|s) Q(s,a) - \sum_a \pi_\beta(a|s) Q(s,a) \right| \\ &= \left| \sum_a Q(s,a) \left(\pi^*(a|s) - \pi_\beta(a|s) \right) \right| \\ &\leq \sum_a |Q(s,a)| \left| \pi^*(a|s) - \pi_\beta(a|s) \right| \\ &\leq \frac{R_{max}}{1-\gamma} \sum_a \left| \pi^*(a|s) - \pi_\beta(a|s) \right| \end{aligned} \tag{9}$$

where π^* is the optimal (deterministic) policy w.r.t. Q and π_β is the Boltzmann distribution induced by Q with inverse temperature β:

$$\pi_\beta(a|s) = \frac{e^{\beta Q(s,a)}}{\sum_{a'} e^{\beta Q(s,a')}}$$

Denote by $a_1(s)$ the optimal action for state s under Q. We can then write:

$$\begin{aligned}\sum_a |\pi^*(a|s) - \pi_\beta(a|s)| &= |\pi^*(a_1(s)|s) - \pi_\beta(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi^*(a|s) - \pi_\beta(a|s)| \\ &= |1 - \pi_\beta(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi_\beta(a|s)| \\ &= 2\,|1 - \pi_\beta(a_1(s)|s)| \qquad (10)\end{aligned}$$

Finally, let us bound this last term:

$$\begin{aligned}|1 - \pi_\beta(a_1(s)|s)| &= \left|1 - \frac{e^{\beta Q(s,a_1(s))}}{\sum_{a'} e^{\beta Q(s,a')}}\right| \\ &= \left|1 - \frac{e^{\beta(Q(s,a_1(s)) - Q(s,a_2(s)))}}{\sum_{a'} e^{\beta(Q(s,a') - Q(s,a_2(s)))}}\right| \\ &= \left|1 - \frac{e^{\beta g(s)}}{\sum_{a'} e^{\beta(Q(s,a') - Q(s,a_2(s)))}}\right| \\ &= \left|1 - \frac{e^{\beta g(s)}}{e^{\beta g(s)} + \sum_{a' \neq a_1(s)} e^{\beta(Q(s,a') - Q(s,a_2(s)))}}\right| \\ &\leq \left|1 - \frac{e^{\beta g(s)}}{e^{\beta g(s)} + |\mathcal{A}|}\right| \\ &= \left|1 - \frac{1}{1 + |\mathcal{A}|\, e^{-\beta g(s)}}\right| \qquad (11)\end{aligned}$$

Combining Eq. (14), (15), and (16), we obtain:

$$\left|\max_a Q(s,a) - \operatorname*{mm}_a Q(s,a)\right| \leq \frac{2R_{max}}{1-\gamma} \left|1 - \frac{1}{1 + |\mathcal{A}|\, e^{-\beta g(s)}}\right|$$

Taking the norm and plugging this into Eq. (12) concludes the proof. □

Theorem 3. *Let Q^* be the fixed-point of the optimal Bellman operator T. Define the action-gap function $g(s)$ as the difference between the value of the best action and the second best action at each state s. Let $\widetilde{Q}$ be the fixed-point of the mellow Bellman operator $\widetilde{T}$ with parameter $\kappa > 0$ and denote by $\beta_\kappa > 0$ the inverse temperature of the induced Boltzmann distribution (as in []). Let ν be a probability measure over the state-action space. Then, for any $p \geq 1$:*

Cite MM

$$\left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^p \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} \left\|\frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g}}\right\|_{\nu,p}^p \qquad (7)$$

Proof. We begin by noticing that:

$$\begin{aligned}\left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^p &= \left\|TQ^* - \widetilde{T}\widetilde{Q}\right\|_{\nu,p}^p \\ &= \left\|TQ^* - \widetilde{T}Q^* + \widetilde{T}Q^* - \widetilde{T}\widetilde{Q}\right\|_{\nu,p}^p \\ &\leq \left\|TQ^* - \widetilde{T}Q^*\right\|_{\nu,p}^p + \left\|\widetilde{T}Q^* - \widetilde{T}\widetilde{Q}\right\|_{\nu,p}^p \\ &\leq \left\|TQ^* - \widetilde{T}Q^*\right\|_{\nu,p}^p + \gamma \left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^p\end{aligned}$$

where the first inequality follows from Minkowsky's inequality and the second one from the contraction property of the mellow Bellman operator. This implies that:

$$\left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^p \leq \frac{1}{1-\gamma} \left\|TQ^* - \widetilde{T}Q^*\right\|_{\nu,p}^p \qquad (12)$$

Let us bound the norm on the right-hand side separately. In order to do that, we will bound the function $\left|TQ^*(s,a) - \widetilde{T}Q^*(s,a)\right|$ point-wisely for any state s, a. By applying the definition of the optimal and mellow Bellman operators, we obtain:

$$\begin{aligned}
\left|TQ^*(s,a) - \widetilde{T}Q^*(s,a)\right| &= \left|R(s,a) + \gamma\mathbb{E}\left[\max_{a'} Q^*(s',a')\right] - R(s,a) - \gamma\mathbb{E}\left[\operatorname*{mm}_{a'} Q^*(s',a')\right]\right| \\
&= \gamma\left|\mathbb{E}\left[\max_{a'} Q^*(s',a')\right] - \mathbb{E}\left[\operatorname*{mm}_{a'} Q^*(s',a')\right]\right| \\
&\leq \gamma\mathbb{E}\left[\left|\max_{a'} Q^*(s',a') - \operatorname*{mm}_{a'} Q^*(s',a')\right|\right] \qquad (13)
\end{aligned}$$

Thus, bounding this quantity reduces to bounding $|\max_a Q^*(s,a) - \text{mm}_a Q^*(s,a)|$ point-wisely for any s. Recall that applying the mellow-max is equivalent to computing an expectation under a Boltzmann distribution with inverse temperature β_κ induced by κ []. Thus, we can write:

Cite MM

$$\begin{aligned}
\left|\max_a Q^*(s,a) - \operatorname*{mm}_a Q^*(s,a)\right| &= \left|\sum_a \pi^*(a|s)Q^*(s,a) - \sum_a \pi_{\beta_\kappa}(a|s)Q^*(s,a)\right| \\
&= \left|\sum_a Q^*(s,a)\left(\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)\right)\right| \\
&\leq \sum_a |Q^*(s,a)|\,|\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \\
&\leq \frac{R_{max}}{1-\gamma}\sum_a |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \qquad (14)
\end{aligned}$$

where π^* is the optimal (deterministic) policy w.r.t. Q^* and π_{β_κ} is the Boltzmann distribution induced by Q^* with inverse temperature β_κ:

$$\pi_\beta(a|s) = \frac{e^{\beta_\kappa Q^*(s,a)}}{\sum_{a'} e^{\beta_\kappa Q^*(s,a')}}$$

Denote by $a_1(s)$ the optimal action for state s under Q^*. We can then write:

$$\begin{aligned}
\sum_a |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| &= |\pi^*(a_1(s)|s) - \pi_{\beta_\kappa}(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \\
&= |1 - \pi_{\beta_\kappa}(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi_{\beta_\kappa}(a|s)| \\
&= 2\,|1 - \pi_{\beta_\kappa}(a_1(s)|s)| \qquad (15)
\end{aligned}$$

Finally, let us bound this last term:

$$\begin{aligned}
|1 - \pi_{\beta_\kappa}(a_1(s)|s)| &= \left|1 - \frac{e^{\beta_\kappa Q^*(s,a_1(s))}}{\sum_{a'} e^{\beta_\kappa Q^*(s,a')}}\right| \\
&= \left|1 - \frac{e^{\beta_\kappa (Q^*(s,a_1(s)) - Q^*(s,a_2(s)))}}{\sum_{a'} e^{\beta_\kappa (Q^*(s,a') - Q^*(s,a_2(s)))}}\right| \\
&= \left|1 - \frac{e^{\beta_\kappa g(s)}}{\sum_{a'} e^{\beta_\kappa (Q^*(s,a') - Q^*(s,a_2(s)))}}\right| \\
&= \left|1 - \frac{e^{\beta_\kappa g(s)}}{e^{\beta_\kappa g(s)} + \sum_{a' \neq a_1(s)} e^{\beta_\kappa (Q^*(s,a') - Q^*(s,a_2(s)))}}\right| \\
&\leq \left|1 - \frac{e^{\beta_\kappa g(s)}}{e^{\beta_\kappa g(s)} + |\mathcal{A}|}\right| \\
&= \left|\frac{1}{1 + \frac{1}{|\mathcal{A}|}e^{\beta_\kappa g(s)}}\right| \qquad (16)
\end{aligned}$$

Combining Eq. (14), (15), and (16), we obtain:

$$\left|\max_a Q(s,a) - \operatorname*{mm}_a Q(s,a)\right| \leq \frac{2R_{max}}{1-\gamma}\left|\frac{1}{1+\frac{1}{|\mathcal{A}|}e^{\beta_\kappa g(s)}}\right|$$

Finally, using Eq. (13) we get:

$$\left|TQ^*(s,a) - \widetilde{T}Q^*(s,a)\right| \leq \frac{2\gamma R_{max}}{1-\gamma}\left|\frac{1}{1+\frac{1}{|\mathcal{A}|}e^{\beta_\kappa g(s)}}\right|$$

Taking the norm and plugging this into Eq. (12) concludes the proof. □

Lemma 1. *Let p and ν denote probability measures over Q-functions and state-action pairs, respectively. Assume Q^* is the unique fixed-point of the optimal Bellman operator T. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a Q-function Q, the following holds:*

$$\|Q - Q^*\|_\nu^2 \leq \frac{\mathbb{E}_p\left[\|B(Q)\|_\nu^2\right]}{(1-\gamma)\delta} \tag{17}$$

Proof. First notice that:

$$\begin{aligned}\|Q - Q^*\| &= \|Q + TQ - TQ - TQ^*\| \\ &\leq \|Q - TQ\| + \|TQ - TQ^*\| \\ &\leq \|Q - TQ\| + \gamma\|Q - Q^*\| \\ &= \|B(Q)\| + \gamma\|Q - Q^*\|\end{aligned}$$

which implies that:

$$\|Q - Q^*\| \leq \frac{1}{1-\gamma}\|B(Q)\|$$

Then we can write:

$$P\left(\|Q - Q^*\| > \epsilon\right) \leq P\left(\|B(Q)\| > \epsilon(1-\gamma)\right) \leq \frac{\mathbb{E}_p\left[\|B(Q)\|_\nu^2\right]}{(1-\gamma)\epsilon}$$

Settings the right-hand side equal to δ and solving for ϵ concludes the proof. □

Corollary 1. *Let p and ν denote probability measures over Q-functions and state-action pairs, respectively. Assume $\widetilde{Q}$ is the unique fixed-point of the mellow Bellman operator $\widetilde{T}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a Q-function Q, the following holds:*

$$\left\|Q - \widetilde{Q}\right\|_\nu^2 \leq \frac{\mathbb{E}_p\left[\left\|\widetilde{B}(Q)\right\|_\nu^2\right]}{(1-\gamma)\delta} \tag{18}$$

Lemma 2. *Assume Q-functions belong to a parametric space of functions bounded by $\frac{R_{max}}{1-\gamma}$. Let p and q be arbitrary distributions over the parameter space $\mathcal{W}$, and ν be a probability measure over $\mathcal{S}\times\mathcal{A}$. Consider a dataset D of N samples and define $v(\boldsymbol{w}) \triangleq \mathbb{E}_\nu\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]\right]$. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following two inequalities hold simultaneously:*

$$\mathbb{E}_q\left[\|B(\boldsymbol{w})\|_\nu^2\right] \leq \mathbb{E}_q\left[\|B(\boldsymbol{w})\|_D^2\right] - \mathbb{E}_q\left[v(\boldsymbol{w})\right] + \frac{\lambda}{N}KL(q||p) + 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}} \tag{19}$$

$$\mathbb{E}_q\left[\|B(\boldsymbol{w})\|_D^2\right] \leq \mathbb{E}_q\left[\|B(\boldsymbol{w})\|_\nu^2\right] + \mathbb{E}_q\left[v(\boldsymbol{w})\right] + \frac{\lambda}{N}KL(q||p) + 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}} \tag{20}$$

Proof. From Hoeffding's inequality we have:

$$P\left(\left|\mathbb{E}_{\nu,\mathcal{P}}\left[\|B(\boldsymbol{w})\|_D^2\right]-\|B(\boldsymbol{w})\|_D^2\right|>\epsilon\right)\leq 2exp\left(-\frac{2N\epsilon^2}{\left(2\frac{R_{max}}{1-\gamma}\right)^4}\right)$$

which implies that, for any $\delta>0$, with probability at least $1-\delta$:

$$\left|\mathbb{E}_{\nu,\mathcal{P}}\left[\|B(\boldsymbol{w})\|_D^2\right]-\|B(\boldsymbol{w})\|_D^2\right|\leq 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}}$$

Under independence assumptions, the expected TD error can be re-written as:

$$\begin{aligned}\mathbb{E}_{\nu,\mathcal{P}}\left[\|B(\boldsymbol{w})\|_D^2\right]&=\mathbb{E}_{\nu,\mathcal{P}}\left[\frac{1}{N}\sum_{i=1}^{N}(r_i+\gamma\,\mathrm{mm}_{a'}\,Q_{\boldsymbol{w}}(s_i',a')-Q_{\boldsymbol{w}}(s_i,a_i))^2\right]\\&=\mathbb{E}_{\nu,\mathcal{P}}\left[(R(s,a)+\gamma\,\mathrm{mm}_{a'}\,Q_{\boldsymbol{w}}(s',a')-Q_{\boldsymbol{w}}(s,a))^2\right]\\&=\mathbb{E}_{\nu}\left[\mathbb{E}_{\mathcal{P}}\left[b(\boldsymbol{w})^2\right]\right]\\&=\mathbb{E}_{\nu}\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]+\mathbb{E}_{\mathcal{P}}\left[b(\boldsymbol{w})\right]^2\right]\\&=v(\boldsymbol{w})+\|B(\boldsymbol{w})\|_\nu^2\end{aligned}$$

where $v(\boldsymbol{w})\triangleq\mathbb{E}_{\nu}\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]\right]$. Thus:

$$\left|\|B(\boldsymbol{w})\|_\nu^2+v(\boldsymbol{w})-\|B(\boldsymbol{w})\|_D^2\right|\leq 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}}\tag{21}$$

From the change of measure inequality [], we have that, for any measurable function $f(\boldsymbol{w})$ and any two probability measures p and q:

Find a reference for this

$$\log\mathbb{E}_p\left[e^{f(\boldsymbol{w})}\right]\geq\mathbb{E}_q\left[f(\boldsymbol{w})\right]-KL(q||p)$$

Thus, multiplying both sides of (21) by $\lambda^{-1}N$ and applying the change of measure inequality with $f(\boldsymbol{w})=\lambda^{-1}N\left|\|B(\boldsymbol{w})\|_\nu^2+v(\boldsymbol{w})-\|B(\boldsymbol{w})\|_D^2\right|$, we obtain:

$$\mathbb{E}_q\left[f(\boldsymbol{w})\right]-KL(q||p)\leq\log\mathbb{E}_p\left[e^{f(\boldsymbol{w})}\right]\leq 4\frac{R_{max}^2\lambda^{-1}N}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}}$$

where the second inequality holds since the right-hand side of (21) does not depend on $\boldsymbol{w}$. Finally, we can explicitly write:

$$\mathbb{E}_q\left[\left|\|B(\boldsymbol{w})\|_\nu^2+v(\boldsymbol{w})-\|B(\boldsymbol{w})\|_D^2\right|\right]\leq\frac{\lambda}{N}KL(q||p)+4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}}$$

from which the lemma follows straightforwardly. □

Lemma 3. *Let p be a prior distribution over the parameter space $\mathcal{W}$, and ν be a probability measure over $\mathcal{S}\times\mathcal{A}$. Assume $\widehat{\xi}$ is the minimizer of $ELBO(\xi)=\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_D^2\right]+\frac{\lambda}{N}KL(q_\xi||p)$ for a dataset D of N samples. Define $v(\boldsymbol{w})\triangleq\mathbb{E}_{\nu}\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]\right]$. Then, for any $\delta>0$, with probability at least $1-\delta$:*

$$\mathbb{E}_{q_{\widehat{\xi}}}\left[\|B(\boldsymbol{w})\|_\nu^2\right]\leq\inf_{\xi\in\Xi}\left\{\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_\nu^2\right]+\mathbb{E}_{q_\xi}\left[v(\boldsymbol{w})\right]+2\frac{\lambda}{N}KL(q_\xi||p)\right\}+2\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{N}}$$

Proof. Let us use Lemma 2 for the specific choice $q = q_{\widehat{\xi}}$. From Eq. (19), we have:

$$\begin{aligned}
\mathbb{E}_{q_{\widehat{\xi}}}\left[\|B(\boldsymbol{w})\|_\nu^2\right] &\le \mathbb{E}_{q_{\widehat{\xi}}}\left[\|B(\boldsymbol{w})\|_D^2\right] - \mathbb{E}_{q_{\widehat{\xi}}}\left[v(\boldsymbol{w})\right] + \frac{\lambda}{N}KL(q_{\widehat{\xi}}||p) + 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}} \\
&\le \mathbb{E}_{q_{\widehat{\xi}}}\left[\|B(\boldsymbol{w})\|_D^2\right] + \frac{\lambda}{N}KL(q_{\widehat{\xi}}||p) + 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}} \\
&= \inf_{\xi\in\Xi}\left\{\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_D^2\right] + \frac{\lambda}{N}KL(q_\xi||p)\right\} + 4\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{2N}}
\end{aligned}$$

where the second inequality holds since $v(\boldsymbol{w}) > 0$, while the equality holds from the definition of $\widehat{\xi}$. We can now use Eq. (20) to bound $\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_D^2\right]$, thus obtaining:

$$\mathbb{E}_{q_{\widehat{\xi}}}\left[\|B(\boldsymbol{w})\|_\nu^2\right] \le \inf_{\xi\in\Xi}\left\{\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_\nu^2\right] + \mathbb{E}_{q_\xi}\left[v(\boldsymbol{w})\right] + 2\frac{\lambda}{N}KL(q_\xi||p)\right\} + 2\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{N}}$$

This concludes the proof. □

B Additional Details on the Algorithms

B.1 Gaussian Variational Transfer

Under Gaussian distributions, all quantities of interest for using Alg. 1 can be computed very easily. The KL divergence between the prior and approximate posterior can be computed in closed-form as:

$$KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w})\mid\mid p(\boldsymbol{w})\right) = \frac{1}{2}\left(\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}|} + \mathrm{Tr}\left(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}\right) + (\boldsymbol{\mu}-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_p) - K\right) \tag{22}$$

for $\boldsymbol{\xi} = (\boldsymbol{\mu}, \boldsymbol{L})$ and $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$. Its gradients with respect to the variational parameters are []:

Cite matrix cookboox

$$\nabla_{\boldsymbol{\mu}}KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w})\mid\mid p(\boldsymbol{w})\right) = \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_p) \tag{23}$$

$$\nabla_{\boldsymbol{L}}KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w})\mid\mid p(\boldsymbol{w})\right) = \boldsymbol{\Sigma}_p^{-1}\boldsymbol{L} - (\boldsymbol{L}^{-1})^T \tag{24}$$

Finally, the gradients w.r.t. the expected likelihood term of the variational objective (2) can be computed using the reparameterization trick (e.g., []):

Cite deep-mind and another

$$\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{L}\boldsymbol{L}^T)}\left[||B(\boldsymbol{w})||_D^2\right] = \mathbb{E}_{\boldsymbol{v}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{w}}||B(\boldsymbol{w})||_D^2\right] \quad \text{for } \boldsymbol{w} = \boldsymbol{L}\boldsymbol{v}+\boldsymbol{\mu} \tag{25}$$

$$\nabla_{\boldsymbol{L}}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{L}\boldsymbol{L}^T)}\left[||B(\boldsymbol{w})||_D^2\right] = \mathbb{E}_{\boldsymbol{v}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{w}}||B(\boldsymbol{w})||_D^2\cdot\boldsymbol{v}^T\right] \quad \text{for } \boldsymbol{w} = \boldsymbol{L}\boldsymbol{v}+\boldsymbol{\mu} \tag{26}$$

B.2 Mixture of Gaussian Variational Transfer

C Additional Details on the Experiments

C.1 The Rooms Problem

C.2 Classic Control

C.3 Maze Navigation

Transfer approaches to mention:

- Taylor 2009: survey
- Konidaris and Barto transfer shaped reward functions PROs: this allows the agent to have a more goal-directed behavior, thus limiting unnecessary random exploration CONs: applicable only to simple problems, does not scale

- Wilson 2007 propose a hierarchical Bayesian model for the distribution over tasks PROs it explicitly models the uncertainty over which tasks are being solved, thus quickly adapting to new ones and allowing informed exploration decisions to be taken
- Lazaric 2008 transfer samples PROs select good samples, can scale to continuous domains CONs eps greedy is used
- Taylor 2008 transfer samples for model-based RL PROs good exploration via model-based RL CONs does not scale, can negatively transfer when tasks are very different
- Lazaric 2010 propose a hierarchical bayesian model for the distribution over value functions PROs quickly adapts to new tasks, non-parameteric (GP) models CONs scaling, strong assumptions (GPTD)
- Fernandez and veloso 2006 propose an exploration strategy based on probabilistic policy reuse PROs good exploration (?) CONs Do not try to figure out which policies are good or bad
- Brunskill propose a method to transfer in model based RL (E3) - PROs theory, exploration CONs scaling
- Barreto use successor features PROs simple, theory CONs eps-greedy exploration

Exploration approaches to mention:

- Osband 2014 propose a method to efficiently explore via randomized value functions
- Osband 2016 adapt such algorithm to DQNs
- Houthooft 2016 use variational inference to approximate the posterior distribution over parameters of the dynamics, and use that to drive exploration
- Azizzadenesheli 2018 extend Osband 2016 to use Bayesian DQN instead. Still makes Gaussian assumptions though