Approximate Bayesian Reinforcement Learning for System Identification

Approximatives Bayesianisches Verstärkendes Lernen zur Systemidentifikation
Master-Thesis von Matthias Georg Schultheis aus Offenbach am Main
Juli 2019

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Approximate Bayesian Reinforcement Learning for System Identification
Approximatives Bayesianisches Verstärkendes Lernen zur Systemidentifikation

Vorgelegte Master-Thesis von Matthias Georg Schultheis aus Offenbach am Main

1. Gutachten: Prof. Jan Peters, Ph. D.
2. Gutachten: M. Sc. Hany Abdulsamad
3. Gutachten: M. Sc. Boris Belousov

Tag der Einreichung:

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Matthias Georg Schultheis, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Datum / Date: ______________________

Unterschrift / Signature: ______________________________________

Abstract

The application of machine learning techniques to the field of robotics has led to considerable progress towards the aim of realising autonomous robots. An autonomous robot, when not having specific tasks to be engaged in, should not remain idle, but prepare for upcoming tasks. One natural form of preparation is to explore the environment and accumulate knowledge in the form of a model of the world. A robot can use such a model to predict consequences of future actions which comes in handy for efficiently solving subsequent tasks. The central question we consider in this thesis is which actions the robot should execute to learn new aspects of the world and to improve its model optimally. We review existing work on exploration strategies and optimal decision making for information gain from several subfields of machine learning. As core of this thesis, we present a novel approach based on insights from active learning. The world is modelled as a Bayesian model and the optimal actions are formulated as the solution of an optimisation problem. To plan actions that are expected to improve the model, the expected model variance and the uncertainty of the resulting trajectory are considered as objectives, leading to two different problem formulations. Since an optimal solution for these problems is not tractable, we consider an approximation that can be solved efficiently using gradient-based trajectory planning methods. The resulting algorithms are compared to state-of-the-art exploration methods. Thanks to the natural representation of model uncertainty and trajectory planning, our proposed method is significantly faster, leads to lower model error and can be used to solve future tasks more reliably.

Zusammenfassung

Die Anwendung von maschinellen Lerntechniken auf den Bereich der Robotik hat zu erheblichen Fortschritten bei der Realisierung autonomer Roboter geführt. Ein autonomer Roboter sollte, wenn er keine spezifischen Aufgaben zu erledigen hat, nicht untätig bleiben, sondern sich auf anstehende Aufgaben vorbereiten. Eine natürliche Form der Vorbereitung ist es, die Umwelt zu erforschen und Wissen in Form eines Modells der Welt zu sammeln. Ein Roboter kann ein solches Modell verwenden, um die Folgen zukünftiger Aktionen vorherzusagen, was sich als nützlich erweist, um nachfolgende Aufgaben effizient zu lösen. Die zentrale Frage, die wir in dieser Arbeit betrachten, ist, welche Aktionen der Roboter ausführen sollte, um neue Aspekte der Welt zu lernen und sein Modell optimal zu verbessern. Wir überprüfen bestehende Arbeiten zu Explorationsstrategien und optimaler Entscheidungsfindung für den Informationsgewinn aus mehreren Teilbereichen des maschinellen Lernens. Als Kern dieser Arbeit stellen wir einen neuartigen Ansatz vor, der auf Erkenntnissen aus dem aktiven Lernen basiert. Die Welt wird als bayesianisches Modell modelliert und die optimalen Handlungen werden als Lösung eines Optimierungsproblems formuliert. Um Handlungen zu planen, von denen erwartet wird, dass sie das Modell verbessern, werden die erwartete Modellvarianz und die Unsicherheit der resultierenden Trajektorie als Ziele betrachtet, was zu zwei verschiedenen Problemstellungen führt. Da eine optimale Lösung für diese Probleme nicht umsetzbar ist, betrachten wir eine Approximation, die mit gradientenbasierten Methoden der Trajektorienplanung effizient gelöst werden kann. Die daraus resultierenden Algorithmen werden mit modernen Explorationsmethoden verglichen. Dank der natürlichen Repräsentation von Modellunsicherheit und Trajektorienplanung ist unser vorgeschlagenes Verfahren deutlich schneller, führt zu geringeren Modellfehlern und kann dazu verwendet werden, zukünftige Aufgaben zuverlässiger zu lösen.

Contents

1 Introduction

Within the last years, autonomous systems have found more and more their way into our daily lifes. While twenty years ago, self-driving cars, autonomous vacuum cleaners, and virtual personal agents seemed like a science fiction fantasy, nowadays these systems are already in widespread use. Many advances in these innovations were made possible by the progress in the field of machine learning. The research in this field aims at enabling agents to ”learn” autonomously without the need of programming complex behaviours by hand. One recent example is the program AlphaZero [1] that learns how to play the games chess, shogi, and go at superhuman level within a few hours of training.

Despite all the recent breakthroughs, it is still challenging for autonomous agents to learn a rich set of behaviour. Usually in reinforcement learning, agents are developed for learning to solve only a specific task based on a predefined reward function. This way, robots can solve quite complex tasks that require versatility and complex motor signals such as playing games like ball-in-a-cup [2] or table tennis [3]. However, in these cases the objective can be precisely formulated and the environment is predictable. Putting robots into complex diverse environments to do a variety of different tasks such as for housekeeping or caring for the elderly, on the other hand, is still a big challenge and currently not feasible. For building agents that can engage in a variety of tasks, they need to be made adaptable and able to accumulate knowledge on the world. To this end, an agent needs to learn a useful model of the environment that he can make use of to solve subsequent tasks more efficiently. Robots could benefit from learning about their own dynamics as well as about the environment that can afterwards be applied to a variety of problems.

In the specific setting we consider, the agent might already be in the environment before having to solve any tasks. In this case, it would be useful for it to explore the environment and to create an internal model of the world. One central question in this setting is how to make the agent to behave in an interested or curious way such that it learns a rich model. In the field of psychology, research on this behaviour in humans is known under the term *curiosity* [4] as subfield of *intrinsic motivation* [5]. It deals with the problem of how humans explore and enrich their knowledge to be more empowered for future situations. In the last thirty years, insights of these fields were tried to transfer to the field of machine learning for building agents that show curious behaviour, e. g. [6, 7]. Especially recently, pushed by the advances in the field of deep learning, very impressive results have been achieved. As an example, agents in video games learned to explore worlds solely based on images [8, 9]. These approaches, however, are usually reactive [10], i. e. the agent accidentally discovers an interesting state region and is reinforced to visit this region again in the near future. After exploring an initially interesting area, the agent remains attracted to this region and has to gradually unlearn moving to this area. This unlearning may require significant time and makes its exploratory behaviour sample-inefficient. This problem is known as ”over-commitment” and can be avoided by actively planning exploration [10].

1.1 Contributions

We present a method which actively plans for exploration to make the agent move into regions that are expected to improve the agent’s model. Our method makes use of ideas introduced in the field of active learning to select a sequence of actions that leads to trajectories of maximal utility for the model. To this end, we formulate two optimisation objectives. In the first one, actions are selected which are expected to reduce the uncertainty of the model most. In the second one, the objective is to maximize the entropy of the trajectory based on an internal dynamics model as proposed by uncertainty sampling. In each iteration of the resulting algorithm, the agent plans a trajectory which then is executed in the real system. The obtained data is then used to refine the model. The dynamics of the environments are modelled using a Bayesian model that naturally provides a measure of model uncertainty. As each sequence of actions is planned independently from the last ones given the model, there is no behaviour of ”over-commitment” and therefore explores the environment efficiently. In comparison to another recently proposed exploration algorithm which already provides remedy for this problem [10], our method is significantly faster, leads to lower model error, and can be used to solve future tasks more reliably.

1.2 Overview of Chapters

This thesis is structured as follows.

Chapter 2 introduces fundamental concepts that serve as basic ingredients for the method we propose. The concepts of reinforcement learning, regression, and planning are presented and measures of entropy and Fisher information are introduced.

Chapter 3 presents related methods that can be used to make optimal decisions leading to maximal information gain. This chapter therefore represents related work to our approach. The problem of exploration and information gain is ubiquitous in the fields of machine learning and optimal control for making optimal decisions. Hence, various subtopics such as active learning, dual control, and intrinsic motivation are introduced, and relevant work of these fields is presented and related to the objective considered in this thesis.

Chapter 4 introduces a novel approach for exploration to the end of learning a model. This approach makes use of a Bayesian model and planning methods to make decisions that are expected to improve the model.

Chapter 5 presents experiments where the approach of chapter 4 is compared to state-of-the art methods for exploration for model learning. In the comparison, random actions, an intrinsic motivation algorithms based on model-free reinforcement learning, andthe model-based active exploration (MAX) algorithm are considered.

Chapter 6 gives a conclusion of the insights discovered in this thesis as well as directions for future research on the problem of exploration for model learning.

2 Foundations

This chapter introduces fundamental concepts that serve as basic ingredients for the exploration method proposed in chapter 4. The concepts of reinforcement learning, planning, and regression are presented and measures of entropy and fisher information are introduced.

2.1 Model-based Reinforcement Learning

In model-based reinforcement learning [11], the task is to find a trajectory $\tau = \{s_0, a_0, s_1, a_1, \dots, a_{T-1}, s_T\}$ by minimizing some cost $c(\tau)$ for a certain time horizon T. Equivalently, instead of minimizing a cost function, a reward function $r(\tau) = -c(\tau)$ may be maximized. The trajectory has to be optimized such that it follows a model f, i. e. $s_{t+1} = f(s_t, a_t)$, $\forall t = 0, \dots, T-1$. Model and reward are either explicitly provided or estimated using past system interactions. Model learning in continuous space can be achieved by means of regression methods which are handled in greater detail section 2.3.

Once having a model of the system and reward function, one can use a planning algorithm to solve for the optimal trajectory under the model given reward function and initial state. Different formulations for of the objective are presented in the following section (2.2) of shooting methods.

2.2 Shooting Methods for Planning

Planning with continuous state-action models can be done using direct shooting methods [12]. In *single shooting methods*, the optimization problem is formulated as optimizing for the actions and computing the following states implicitly by plugging in the model:

$$\min_{a_0, \dots, a_{T-1}} \quad c(s_0, a_0, s_1, a_1, \dots, a_{T-1}, s_T)$$

where $s_t = f(s_{t-1}, a_{t-1})$ for $t = 1, \dots, T$.
In *multiple shooting methods* the states are also considered as optimization variables and consistency between states and model is ensured by inserting a constraint:

$$\begin{aligned} \min_{a_0, s_1, \dots, a_{T-1}, s_T} \quad & c(s_0, a_0, s_1, a_1, \dots, a_{T-1}, s_T) \\ \text{s.t.} \quad & s_t = f(s_{t-1}, a_{t-1}), \quad t = 1, \dots, T \end{aligned}$$

Multiple shooting methods are often considered superior to single shooting methods as lifting the problem to a higher dimension is known to often improve convergence [13]. Also, using multiple shooting offers the possibility to initialize the trajectory using some guess. A drawback of multiple shooting methods is that the problem gets much larger as more quantities have to be determined and additional constraints have to be considered. This drawback however is often compensated by the fact that it the problem becomes much sparser [13].

The formulated optimization problems can be directly solved using numerical optimizers such as IPOPT [14] together with frameworks like CasADi [13].

2.3 Regression Methods

In supervised learning, we are given inputs $\mathbf{x}$ with corresponding target values $\mathbf{y}$ with $\mathbf{y} = f(\mathbf{x}) + \epsilon$ for some unknown function f and optional noise ϵ. The goal of regression is to get an estimate of f when the targets are continuous which enables us to predict targets for new inputs $\mathbf{x}_*$.

2.3.1 Bayesian Linear Regression

Linear Regression models $f(\mathbf{x})$ to be a function of the form $\boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x})$ where $\boldsymbol{\theta}$ are parameters to learn and $\boldsymbol{\phi}(\mathbf{x})$ are fixed basis functions. If we assume Gaussian noise, the likelihood of observations is given by

$$p(\mathbf{y} \mid \boldsymbol{\theta}; \mathbf{x}) = \mathcal{N}\left(\mathbf{y} \mid \boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}), \Sigma\right).$$

A popular choice for the feature representation are radial basis functions (RBF) with centers equally distributed over the space [15]:

$$\boldsymbol{\phi}_{\mathrm{RBF}}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\nu}_{\mathrm{i}}\|^2}{2\sigma^2}\right)$$

An example of RBF features is shown in figure 2.1.

In Bayesian Linear Regression [15, 16], we incorporate a prior over $\boldsymbol{\theta}$ which enables us to maintain a full distribution over parameters $\boldsymbol{\theta}$ and thus to reason about the uncertainty of predictions $f(\mathbf{x}_*)$. To simplify posterior computation, the prior should be chosen conjugate to the likelihood, so we opt for a Gaussian prior over $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_0, \boldsymbol{\Sigma}_0)$$

By applying Gaussian identities [15] we obtain the posterior

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}) &\propto p(\mathbf{y} \mid \boldsymbol{\theta}; \mathbf{x}) p(\boldsymbol{\theta}) \\ &= \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \\ \text{with } \boldsymbol{\mu}_* &= \boldsymbol{\Sigma}_*\left(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \beta\boldsymbol{\Phi}^T\mathbf{Y}\right) \\ \boldsymbol{\Sigma}_*^{-1} &= \boldsymbol{\Sigma}_0^{-1} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} \end{aligned} \tag{2.1}$$

where feature inputs and targets of the training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1,\ldots,N}$ are aggregated into $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ respectively.
Once we have computed the posterior, we can evaluate the predictive distribution

$$\begin{aligned} p(\mathbf{y}_* \mid \mathbf{x}_*, \mathcal{D}) &= \int p(\mathbf{y}_* \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) \mathrm{d}\boldsymbol{\theta} \\ &= \mathcal{N}(\mathbf{y}_* \mid \boldsymbol{\mu}_*^T\boldsymbol{\phi}(\mathbf{x}_*), \sigma_*^2(\mathbf{x}_*)) \\ \text{with } \sigma_*^2(\mathbf{x}_*) &= \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x}_*)^T\boldsymbol{\Sigma}_*\boldsymbol{\phi}(\mathbf{x}_*). \end{aligned}$$

For now we considered only one-dimensional targets, i. e. $y_i \in \mathbb{R}$. However, the extension to multi-dimensional targets, $y_i \in \mathbb{R}^N$, is straight-forward. In this case, parameters $\boldsymbol{\theta}$ become a matrix. As sigma doesn't depend on the target values, it can be used directly for multiple outputs and only the mean $\boldsymbol{\mu}_*$ has to be shaped into a matrix by having $\boldsymbol{\mu}_0$ and $\mathbf{Y}$ as matrices.

If data flows in sequentially, the posterior can be iteratively updated. To do so, the last computed posterior becomes the prior and only the new data is considered for the likelihood. This procedure enables efficient computation in the case of sequential data and is usually denoted as *iterative least squares* [15].

2.3.2 Gaussian Processes

A common problem when applying linear regression is that it is necessary to engineer the features, i. e. a suitable number and form of features has to be determined. Gaussian Processes [17] offer a way to do Bayesian linear regression with implicitly using infinitely many features by replacing products of features by a kernel function.

Formally, the optimal approach [16] for making predictions for some new inputs $\mathbf{x}_*$ is to use a distribution $p(f \mid \mathcal{D})$ over possible functions given the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1,\ldots,N}$:

$$p(y_* \mid \mathbf{x}_*, \mathcal{D}) = \int p(y_* \mid \mathbf{x}_*, f) p(f \mid \mathcal{D}) \mathrm{d}f$$

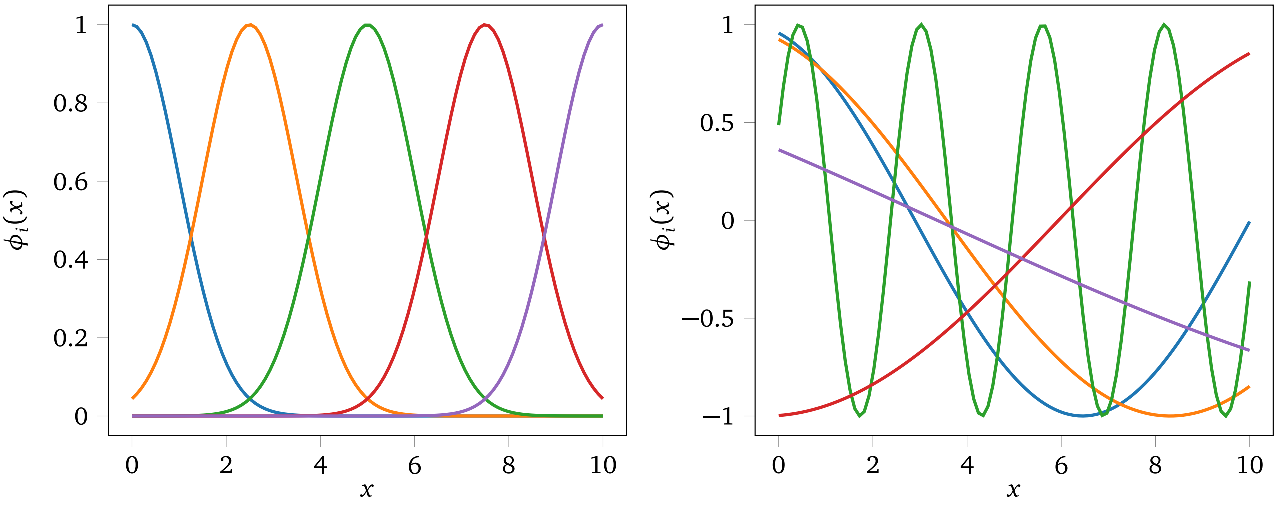


Figure 2.1: The plot shows radial basis (left) and random fourier (right) features for a one-dimensional x. In each plot there are five examples $\phi_i(x)$ shown. For RBF, σ was set to 1, for RFF ν was set to 1. In linear regression, these feature functions are linearly combined in order to approximate a continuous function.

A Gaussian Process (GP) [17] defines in this context a convenient and powerful [18] prior distribution $p(f \mid \mathscr{D})$ over functions. It assumes that the any finite set of points $\{\mathbf{x}_1, \dots \mathbf{x}_N\}$, induces a multivariate Gaussian distribution $p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_N))$. This distribution is fully specified by some mean $\boldsymbol{\mu}(\mathbf{x})$ and covariance $\boldsymbol{\Sigma}(\mathbf{x})$ given by $\boldsymbol{\Sigma}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ where κ is a positive definite kernel function. The prior over f defined by the GP can be converted into a posterior over functions when we observe some data. Even if GPs can be applied to both regression and classification tasks, we focus in this thesis only on the former case. In the regression setting, the process of inferring the posterior can be done in closed form and in $\mathscr{O}(N^3)$ time.

2.3.3 Gaussian Processes for Regression

We denote the GP that is the prior on the regression function by

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$$

where $m(\mathbf{x})$ is the mean function and $\kappa(\mathbf{x}, \mathbf{x}')$ the positive-definite kernel covariance function. The mean and kernel function can be chosen freely. A popular choice for the kernel function is the squared exponential kernel, also known as RBF kernel, which is given by

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)\right) + \sigma_y^2 \delta_{ij}$$

where $\mathbf{M}$ is a matrix determining the horizontal length scale, σ_f^2 the vertical scale and σ_y^2 the noise variance.

Suppose, we have a set of observations $\mathscr{D} = \{\mathbf{x}_i, y_i\}_{i=1,\dots,N}$ where $y_i = f(\mathbf{x}_i) + \epsilon_i$ are noisy observations. We assume the noise ϵ_i to be independent identically distributed Gaussian noise with variance σ_f^2. For convenience, we aggregate both inputs and outputs $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ and $\mathbf{y} = [y_1, \dots, y_N]$ respectively.

Given a set of new data points $\mathbf{X}_*$ of size $N_* \times D$ that we want to predict outputs $\mathbf{y}_*$ for, by the definition of the GP, the joint distribution can be written as

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{y}_* \end{pmatrix} \sim \mathscr{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{pmatrix}\right)$$

where $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I}_N$ is $N \times N$, $\mathbf{K}_* = \kappa(\mathbf{X}, \mathbf{X}_*)$ is $N \times N_*$, and $\mathbf{K}_{**} = \kappa(\mathbf{X}_*, \mathbf{X}_*)$ is $N_* \times N_*$.

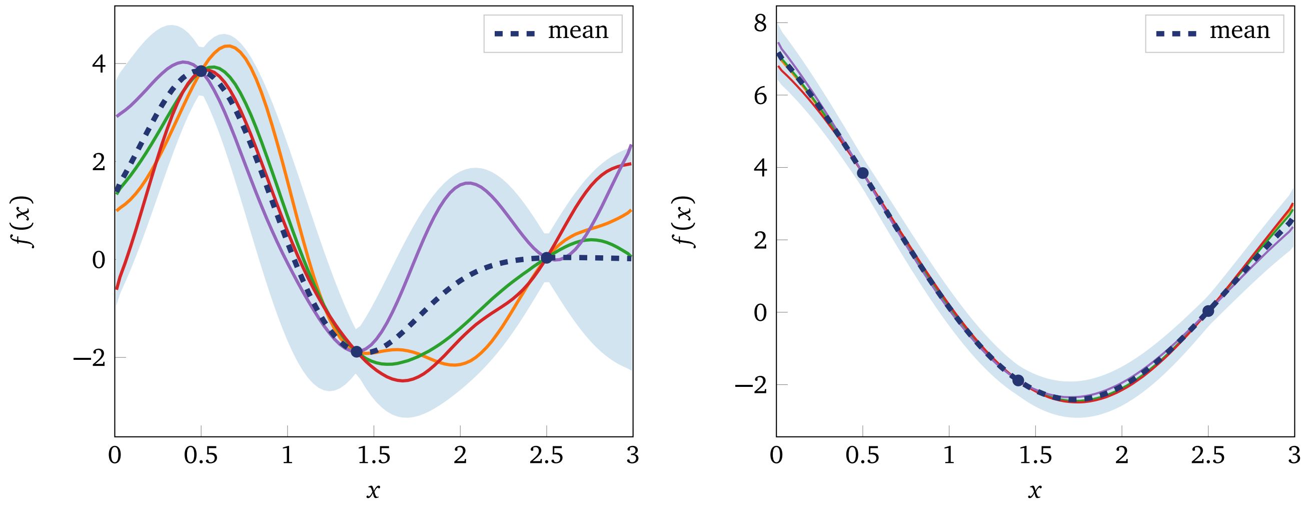


Figure 2.2: In this figure, Gaussian process fits on three data points (blue) using different vertical length-scales are shown. On the left, σ_f^2 was set to 0.12 whereas on the right $\sigma_f^2 = 0.2$. The parameter set for the right fit assumes higher smoothness of the underlying function. Even if the fit on the right has higher training data likelihood, depending on the knowledge about the true function either of the parameter choices could be more suitable.

By means of standard rules for conditioning Gaussians [16] the posterior can be derived as

$$\begin{aligned} p(\mathbf{y}_* \,|\, \mathbf{X}_*, \mathbf{X}, \mathbf{y}) &= \mathcal{N}(\mathbf{y}_* \,|\, \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \\ \boldsymbol{\mu}_* &= \boldsymbol{\mu}(\mathbf{X}_*) + \mathbf{K}^T \mathbf{K}_*^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{X})) \\ \boldsymbol{\Sigma}_* &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*. \end{aligned}$$

The posterior is the distribution of the prediction we obtain for $\mathbf{X}_*$ given our GP model.

2.3.4 Parameter Optimization

The performance of the GP model strongly depends on the choice of the kernel function. While the previously introduced RBF kernel is a common choice and usually leads to good results, it depends on parameters $\mathbf{M}$, σ_f and σ_y that have to be chosen thoroughly. Examples of different parameter choices are shown in figure 2.2. A common empirical Bayesian approach to optimize these parameters is to maximize the marginal likelihood [16]

$$p(\mathbf{y} \,|\, \mathbf{X}) = \int p(\mathbf{y} \,|\, \mathbf{X}, f) p(f \,|\, \mathbf{X}) \mathrm{d}f.$$

Using the definition of the Gaussian Process and the fact that the observations are independent given f we end up with the log likelihood

$$\log p(\mathbf{y} \,|\, \mathbf{X}) = \log \mathcal{N}(\mathbf{y} \,|\, \mathbf{0}, \mathbf{K}) = \underbrace{-\frac{1}{2}\mathbf{y}\mathbf{K}^{-1}\mathbf{y}}_{\text{data fit}} \underbrace{-\frac{1}{2}\log|\mathbf{K}|}_{\text{complexity}} \underbrace{-\frac{N}{2}\log(2\pi)}_{\text{const.}}$$

where we are assuming the mean is zero, for notational simplicity. This formula is differentiable w.r.t. the kernel function, thus, the gradient w.r.t. the kernel parameters can be computed. Any gradient-based optimization method, e. g. the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [19], can then be used to determine the parameters yielding maximal likelihood.

2.3.5 Sparse Gaussian Processes

As non-parametric method, the computational complexity for doing inference in Gaussian Processes depends on the number of training points. Inference for making new predictions lies in $\mathcal{O}(N^3)$ with N being the number of training examples which leads to slow performance if there are many data points. Sparse Gaussian Process methods decrease the complexity to $\mathcal{O}(M^2N)$ by approximating the dataset by a fixed number M of pseudo-inputs. The computation of $\mathbf{K}^{-1}$ in equation **??** is then approximated using the Nyström approximation $\mathbf{K}^{-1} \approx \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}$ where $\mathbf{K}_{MM}$ is the covariance matrix evaluated for the pseudo-inputs. This formulation requires only the inversion of a matrix of size $M \times M$ instead of an $N \times N$ matrix.

There have been presented numerous approaches to infer the pseudo-inputs. In [20] a subset of the data is greedily selected. [21] formulates the search for pseudo-input locations in form of marginal likelihood maximization. As most recent and currently popular approach, pseudo-inputs are modelled as variational parameters [22, 23] and a lower bound of the true log marginal likelihood is maximized. The pseudo-inputs are this way chosen to minimize the Kullback-Leibler between the approximated GP and the original one.

2.3.6 Random Fourier Features

One advantage of Gaussian Processes compared to Bayesian linear regression using RBF features is that they require little parameter tuning. Random Fourier Features (RFF), introduced by Rahimi [24], offer a powerful representation in randomized low-dimensional space that can be used to approximate a Gaussian Process with an exponentiated quadratic kernel. Empirically, they were shown to perform competitively with state-of-the-art kernel machines [25]. Their application in the linear regression set-up thus requires little tuning like GPs while offering a parametric model that can be used to carry out iterative linear regression.

The RFF representations $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^M$ of inputs $\mathbf{x} \in \mathbb{R}^N$ is given as presented in [25] as

$$\phi_i(\mathbf{x}) = \sin\left(\frac{\sum_{j=1}^{N} \mathbf{P}_{ij} s_j}{\nu_j} + \varphi_i\right)$$

where each element $\mathbf{P}_{ij}$ is drawn from $\mathcal{N}(0,1)$ and φ_i are random phase shifts which are sampled uniformly from the interval $[-\pi, \pi)$. $\boldsymbol{\nu}$ denotes bandwidth parameters for changing the frequency of the sinus. In [25] it was suggested to chose them approximately as the average pairwise distances between different observation vectors. A better alternative, however, is to tune them in order maximizing the data likelihood, analogously to section 2.3.4. To introduce a bias term, one can set $\mathbf{P}_{0\cdot} = 0$ and $\varphi_0 = 0$, so that $\phi_0(\mathbf{x}) = 1$.

2.4 Entropy

Entropy [26] is an information-theoretic measure that represents the amount of information needed to "encode" a distribution. In machine learning it is often used as measure of uncertainty [16]. The distribution with maximum entropy is the uniform distribution which shows the highest possible degree of uncertainty.

The entropy of a random variable $\mathcal{X}$ with distribution $p(\mathcal{X})$ is defined as

$$\mathbb{H}(\mathcal{X}) = -\int p(x) \log p(x) \mathrm{d}x.$$

For the Gaussian distribution $p_{\mathcal{N}}(\mathcal{X}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the entropy is given in closed form as

$$\mathbb{H}_{\mathcal{N}}(\mathcal{X}) = \frac{k}{2}\ln(2\pi e) + \frac{1}{2}\ln(|\boldsymbol{\Sigma}|).$$

Within this thesis, we will frequently use the concept of entropy for measuring the certainty of beliefs. For parameter estimates we want a distribution of low entropy so we are certain and its highly peaked..

2.5 Fisher Information

In parameter estimation problems, we want to get information about the parameters from a sample set of data. The Fisher information is an information measure describing the amount of information, data provides about an unknown parameter. For a random sample $\mathcal{X}$ with probability density $p(\mathcal{X} \mid \boldsymbol{\theta})$ for some model of data with parameters $\boldsymbol{\theta}$ the Fisher Information Matrix (FIM) is defined as

$$\mathcal{I}(\boldsymbol{\theta}) = -\mathop{\mathbb{E}}_{p(\mathbf{x}\mid\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x} \mid \boldsymbol{\theta})^T \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x} \mid \boldsymbol{\theta}) \tag{2.3}$$

$$= -\mathop{\mathbb{E}}_{p(\mathbf{x}\mid\boldsymbol{\theta})} \mathrm{H}_{\boldsymbol{\theta}} \log p(\mathbf{x} \mid \boldsymbol{\theta}). \tag{2.4}$$

A theorem, known as the Cramér Rao Lower Bound (CRLB) [27], states that the inverse of the Fisher information defines a lower bound on the variance of any unbiased estimator $\hat{\theta}$ of θ:

$$\mathbb{V}_{\theta}(\hat{\theta}) \geq \frac{1}{\mathscr{I}(\theta)} \tag{2.5}$$

Put into other words, the precision to which we can estimate θ is limited by the inverse Fisher information of the likelihood function. This theorem can be intuitively explained for maximum likelihood estimators as follows: If the variance $\mathbb{V}_{\theta}(\hat{\theta})$ is high, $\log p(x \mid \theta)$ is broadly shaped as we are unsure about the estimate. This means $\nabla_{\theta} \log p(x \mid \theta)$ changes slowly around the maximum likelihood estimate and results in a low Hessian, thus $\mathscr{I}(\theta)$ becomes small.

The concept of Fisher information is widely used in optimal experimental design. As estimator variance and Fisher information are reciprocally connected, minimizing the variance corresponds to maximizing the information.

3 Related Concepts

This chapter presents related methods that can be used to make optimal decisions leading to maximal information gain. It therefore introduces related work to our exploration approach presented in chapter . The problem of exploration and information gain is ubiquitous in the fields of machine learning and optimal control for making optimal decisions. We present relevant work of various subtopics which tackle the problem of exploration, i. e. active learning, dual control, and intrinsic motivation. Further, we take a look at concepts that are related in a broader sense, i. e. Bayesian optimisation and empowerment.

3.1 Active Learning

In the supervised learning setting, we need labelled data for learning a model. In many cases, the process of labelling is expensive, for example when manual labelling by human is necessary or if experiments in reality need to be conducted. In this case, it is useful to select a small dataset that labelled and provided to the learner leads to high model performance. Active learning [28], also denoted as optimal experimental design, tackles the problem on how to select the instances for this dataset by estimating the utility of instances.

Usually, this process is performed iteratively: By means of the current model and a measure of informativeness, the *acquisition function*, a fixed number of instances from a pool of unlabelled data is selected. These instances are labelled and used for refining the model. In stream-based or sequential active learning [29, 30] one instance at a time is sequentially presented to the the learner and it needs to decide whether to label this instance or to ignore it. In pool-based active learning [31], instead of receiving one instance at a time, the learner has to select from a larger set one or more instances for labelling. This thesis focuses on the latter use of active learning where one instance is selected at a time. There, the instance is chosen that maximises the acquisition function $\alpha(x)$

$$x^* = \underset{x}{\operatorname{argmax}}\, \alpha(x).$$

To selecting instances optimally for maximizing an objective, one would have to do an extensive search in form of planning in the space over labels of instances [32]. As this procedure is infeasible for continuous space and even shows very high complexity in discrete space, different heuristics were proposed. These heuristic acquisition functions measure the effects of querying instances only for the next time step. In the following an overview over different acquisition functions is given.

Uncertainty Sampling

As probably most simple and common [28] approach, uncertainty sampling selects the instances that the learner is most uncertain about. The intuition behind this method is that a learner profits most by receiving samples of which it is most unsure about the corresponding target value. For measuring the uncertainty, naturally a probabilistic model can be used and the unlabeled instance is chosen for which the model has highest output variance in its prediction. It is common to use entropy [26] as uncertainty measure and select the instance having maximal entropy:

$$\alpha_{\text{US}}(x) = -\int_y p(y \mid x) \log p(y \mid x)$$

There are also extensions for deterministic models, e. g. by using a committee of models [33] and select the instance of maximum disagreement between the models.

Compared to other acquisition functions, uncertainty sampling is simple from a computational viewpoint. A drawback of using this measure is that it likely fails in case of stochastic systems. There, the model will remain uncertain for instances having labels with high randomness. It will query these instances over and over again, even if the model cannot be improved anymore for these predictions. This problem is also faced in literature regarding intrinsic motivation (see section 3.3) and there known under the term *noisy-TV problem* [34].

Expected Model Change

Expected Model Change is a strategy that selects the the instance that would lead to the greatest change to the model. One example of this framework is Expected Gradient Length [35] which can be used if gradient-based training of model parameters is used. The underlying idea of this method is that the change of the model can be measured by the length of the training gradient. The learner should therefore query that instance x that – if labelled and added to the learner – results in the training gradient of largest magnitude. Since we do not know the target value in advance, the expectation over targets of the gradient length has to be taken. The acquisition function can then be formulated as

$$\alpha_{\mathrm{EMC}}(x) = -\int_y p(y \mid x) \, \|\nabla l_\theta(\mathscr{D} \cup \langle x, y \rangle)\|$$

where ∇l_θ is the gradient of the learner's objective function $l(\cdot)$ with respect to the model parameters θ and $\mathscr{D}$ the training data set.

The intuition behind this approach is that it chooses instances that likely influence most the model and have the highest impact on the parameters.

Expected Error Reduction

The acquisition function of expected error reduction measures how much the total prediction error of a model is reduced by adding an instance. The induced goal is to select instances that most reduce the error of the updated model. The future error of the model trained by the augmented set is estimated and the instance with expected minimal future error is chosen. Since the labels of the future observations are unknown, like in expected model change, the expected error over future models is considered. The acquisition function for expected error reduction is given by

$$\alpha_{\mathrm{EER}}(x) = \mathbb{E}_{y \sim p(y \mid x)} \sum_{x^* \in \mathscr{D}_{\mathrm{test}}} \mathbb{E}_{y^* \sim p(y^* \mid x^*)} \mathbb{E}_{\hat{y}^* \sim p_{+\langle x, y \rangle}(\hat{y}^* \mid x^*)} \left[(y^* - \hat{y}^*)^2 \right]$$

where $\mathscr{D}_{\mathrm{test}}$ denotes a set of test instances and $p_{+\langle x,y \rangle}$ is the probability distribution of the model p retrained using the augmented training set $\mathscr{D}_{\mathrm{train}} \cup \{(x, y)\}$. If we want to select instances greedily, this method is usually the most precise formulation of our intention for instance selection. In most cases, unfortunately, this method is infeasible as it requires estimating the expected future error (which is an expectation over the model) and model retraining for each candidate.

Expected Variance Reduction

Minimizing the expected total error directly as in Expected Error Reduction is expensive and not even the expectation over the model for a candidate can usually not be evaluated in closed form. One can still try to reduce the generalization error indirectly by minimizing the variance of the model. Geman et al. [36, 28] showed that one can decompose the learner's expected future squared error in regression problems as follows:

$$\mathbb{E}_{y \sim p} \mathbb{E}_{\hat{y} \sim \mathscr{D}} \left[(\hat{y} - y \mid x)^2 \right] = \underbrace{\mathbb{E}_p \left[(y - \mathbb{E}_p [y \mid x])^2 \right]}_{\text{noise}} + \underbrace{(\mathbb{E}_{\mathscr{D}}[\hat{y}] - \mathbb{E}_p[y \mid x])^2}_{\text{bias}} + \underbrace{\mathbb{E}_{\mathscr{D}} \left[(\hat{y} - \mathbb{E}_{\mathscr{D}}[\hat{y}])^2 \right]}_{\text{variance}}$$

where $\mathbb{E}_{\mathscr{D}}$ is the expectation over the training data set $\mathscr{D}$, $\hat{y}$ is the predicted model output for instance x, and y is the true value for x. The noise term describes the variance of the true label y given the instance x. The bias term represents the error of the model class itself. Both noise and bias are independent from the learner as noise does not depend on the model and the bias is constant for a specific model class. The only remaining term of the learners loss that the learner influences is the variance term. Minimizing the variance of the learner is therefore shown to decrease the future generalization error of the model. A motivated acquisition function to maximise is therefore the negative variance of the model:

$$\alpha_{\mathrm{EVR}}(x) = \mathrm{var}(\mathscr{M} \mid \mathscr{D} \cup \{x\})$$

that denotes the variance of the model $\mathscr{M}$ learned using the augmented data set $\mathscr{D} \cup \{x\}$. In the case of linear regression, the expected variance can be evaluated in closed form as it does not depend on the observation y of input x. In other cases like neural networks, there exist approximations similar to the fisher information (see section 2.5) as presented in [37, 38] which offer a closed-form formulation of $\alpha_{\mathrm{EVR}}(x)$. This acquisition function can thus be evaluated efficiently in comparison to Expected Error Reduction.

Connections to our approach

The problem considered in this thesis, to select actions for learning a to learn a model in the best possible way, can be formulated as an active learning problem: ”Which state-action pairs should be labelled with their next-states to get a rich dynamics model?” However, in contrast to the presented frame of active learning, only a part of the inputs can be controlled (the actions) while the states depend on past executions. Therefore, planning is required to reach interesting state regions and direct application of the presented acquisition functions is not possible. The observation that one can define an instance to be a full sequence of actions to execute, enables framing the problem of exploration as an active learning problem as presented here – but with the cost of making acquisition functions much harder to evaluate due to the long-time dependencies of the model. To make computation feasible, we will introduce an additional approximation in our method.

3.2 Dual Control and Bayesian Reinforcement Learning

For controlling an unknown system optimally, a controller has to pursue two goals [39]. First, it has to control the system as best as the current knowledge permits (exploitation). Second, the controller needs to get more information about the system in order to control the system better in future steps (exploration). This problem was named dual control by Feldbaum [40], nowadays also known under the term Bayesian reinforcement learning (BRL or Bayesian RL) [41, 42]. The main difficulty of dual control problems is that the two goals are conflicting. For exploration it is necessary to disturb the model evoking unexpected system while for controlling, the system has to be kept under control.

Traditional non-dual controllers are based on the certainty equivalence principle. This means, the system parameters are estimated and only the means of the estimation are used for control as if they were the true ones. If using this approach directly, learning happens only accidentally when the system behaviour differs from the controller’s expectation as there is no intentional exploration. For these controllers, exploration can still be evoked by introducing perturbation signals. If they are chosen small, the controller explores locally around the optimal trajectory. As controls are still not systematically chosen, these controllers are termed *non-dual* and learning is named *passive* [39].

While traditional adaptive controllers only need to consider past observations, an optimal dual controller has to incorporate expectations about future observations into planning. Feldbaum formulated the problem of dual control and its solution based on dynamic programming by combining the physical state and system parameters into an augmented dynamics description [40] and apply belief space planning. As dynamics are highly non-linear even in case of linear systems, the optimal solution is computationally very expensive and only tractable in few very special cases, e. g. [43]. Over the time many approximations of the dual control problem have been formulated, yielding suboptimal dual controllers.

The most-popular traditional approximation for the dual control problem is to revert to non-dual controllers as those acting under certainty equivalence with added perturbations [39]. Another approach aims at minimizing the loss function only one step ahead under certain constraints, such as limiting the minimum control signal [44] or the variance [45]. Also, some approximations were presented in the literature that make modifications of the loss function. In these approaches, terms reflecting the quality of parameter estimates were added to the loss function. Wittenmark and Elevitch [46] add a local measure of future system parameter variance as term to the loss function. As evaluating this term is difficult, they make a linearisation around the certainty equivalence solution and and use the local-linear approximation for optimization. Milito et al. [47] introduce an *innovation term* to the loss function measuring the information that observations contain about the system parameters. Both presented approaches that modify the loss function, however, have the assumption that the system is linear which limits their application quite heavily. A more detailed overview of traditional dual control methods is given by Wittenmark and can be found by the interested reader in [39].

Newer Bayesian reinforcement learning methods aim at introducing exploration more naturally by maintaining a probabilistic belief over the system knowledge. Bayesian reinforcement learning methods, in general, can be divided into *model-based* and *model-free* methods. Unfortunately, most model-based methods target discrete models, as these are much simpler than continuous problems. Examples for model-based discrete methods are Bayesian Exploration Exploitation Tradeoff in LEarning (BEETLE) [48], tree-search methods such as Bayesian Sparse Sampling (BSS) [49] or Bayes-adaptive Monte-Carlo planning (BAMCP) [50] that can be read about in [41]. One approach that can also applied to continuous systems is Bayesian dynamic programming [51] which is inspired by Thompson sampling (see section 3.4). There, a model is sampled from a posterior distribution over parameters and this model is solved to select actions. In [42], Klenske and Hennig present a tractable analytic approximation to solve the Bayesian reinforcement learning problem. They compute a solution based on the certainty equivalence principle and construct a local quadratic approximation around this trajectory of the effects due to future observations. Subsequently, they optimize the information gain without moving too far from the certainty equivalence solution. Their method still has high computational cost but is tractable and seems to provide good results. With POLO [52], Lowrey et al. propose a method for tackling the dual control problem in case of having access to the correct dynamics model of the system where only the rewards are unknown. This method combines local trajectory optimization in form of MPC with approximate value function learning to control the system to potential rewarding state regions. An ensemble of models is hereby used to model the value function and states causing most lack of agreement within the ensemble is regarded as promising which encourages exploration. The assumption of a given dynamics model is quite strong, however, for the case that it was given, this method was shown to speed up the learning process and allow for better policies than local solutions.

Model-free Bayesian reinforcement learning methods can be divided into value function methods and policy search methods. In these, a Bayesian belief over value function or policies, respectively, is constructed directly without learning an explicit model. The exploration is directed by the variety of rewards and the agent seeks to explore regions where it expects to receive high rewards. A detailed overview over these methods is given in [41].

The work in [53] propose a way to balance exploration and exploitation by maintaining separate exploration and exploitation policies and alternating them during execution. In their approach they assume the exploration policy (e. g. demonstrations with added random perturbations) given and update only the exploitation policy during executions. In the work presented in [54] the authors show a way to omit the need to perform explicit exploration entirely, however also by using demonstrations by a teacher. Using the demonstrations, they learn a model of the system and execute only ”exploitation policies” based on the model to learn a suitable policy.

Another way to solve the dual control problem is to use traditional reinforcement learning methods with adding exploration bonuses to the reward. Comparable with [46, 47], terms measuring the exploratory success are added to the loss function and the augmented system is solved using regular reinforcement learning methods. This concept is inspired by observations of human behaviour from the field of psychology, i. e. intrinsic motivation and curiosity. Through the additivity of the reward function, the problems of exploration and exploitation are separated and can be regarded independently. The whole research in this area focuses then on finding convenient measures of measuring exploration and even can be done without the goal of controlling the system to a certain goal. Due to the importance of exploration in this theses, the next section (3.3) is dedicated to these methods.

Connections to our approach

The objective to gain new knowledge in dual control enforces exploratory behaviour of the controller. Therefore, in principle, one should be able to solve the pure exploration problem by applying a method of dual control (or Bayesian reinforcement learning) by using a constant value as reward. This setting then is equal to problems where (very) sparse rewards are provided and the proposed methods usually fail. As presented in this section, the dual control problem cannot be solved exactly but approximations are necessary that focus on controlling the system using current knowledge and explore locally around the exploitation trajectories, e. g. [41, 42]. An exception is formed by the methods which adapt the objective function and consequently develop strategies for tackling the exploration problem independently. The classical methods [46, 47] that solve the dual control problem this way, however, are only applicable to linear systems that makes us target our attention to intrinsic motivation curiosity methods which are presented in the following section.

3.3 Exploration and Curiosity

It is important to implement autonomous agents some form of exploratory behaviour to gather knowledge about the system - to the end of learning a model of the environment or to learn a policy for solving a specific task. In this section, we will review literature that approached the this problem. After giving an overview over research in the field of control, we will focus on curiosity as conception modelling intrinsic motivation, a concept originating from psychology. Within the past years, this concept was transferred to the field of reinforcement learning to evoke exploratory behaviour in agents. While most research focuses on the application with model-free reinforcement learning methods we will also consider the few cases where model-based reinforcement learning was applied.

3.3.1 The Control Perspective: Optimal Input Design

In fact, the problem of choosing actions that are most beneficial to obtain a good knowledge about a system was known long before the introduction of the concept of curiosity into the field of machine learning. In control literature this problem is known as *optimal input design* and deals with the selection of input signals for learning a rich model of the system based on the system outputs. In early work, methods focused on minimizing a measure of the covariance of the system parameters to estimate, e. g. [55, 56]. Later, also measures of the quality of transfer function estimates (e. g.

[57]) and robust performance criteria (e. g. [58]) were introduced. Moore [59] introduced the concept of persistent excitation for choosing input signals that guarantee convergence rates for learning robust least squares models. Aoki [60] proposed a method for selecting input signals that minimise the estimation error of system parameters in case of noisy measurements. In this work, he derived a closed form solution for maximizing the Fisher information based on eigenvalues of an observation matrix. In the branch of optimal input design, near optimal approaches and bounds could be given, but these methods are usually restricted to very simple models like linear ones. A comprehensive detailed review of optimal input design methods was given by Gevers and Bombois in [61].

3.3.2 Introduction to Intrinsic Motivation / Curiosity: Links to Psychology

Curiosity and motivation has already been a research topic for philosophers and psychologist for a long time. One can observe that human beings, especially children, engage in activities of various types without receiving external rewards. In psychology, one differentiates between *intrinsic* and *extrinsic motivation* [5, 62]. A human is called extrinsically motivated if he pursues an activity to achieve a specific outcome which may be an external reward. Intrinsic motivation, on the other hand, denotes the motivation for doing an activity for its inherent satisfaction rather than for any other consequences. These activities often involve the motivation to discover new experiences that is denoted as *curiosity* [4]. It is assumed that intrinsic motivation is a drive for learning skills, knowledge and competences that might come handy for pursuing rewards in the future [63]. Performing an action without receiving an immediate reward but which is still useful for the human, e. g. learning, is usually not considered as intrinsic motivation in psychology. These forms of motivation are known as internalized forms of extrinsic motivation [64].

In the field of machine learning, the idea of inherent interest in performing an activity was adopted and is known under the term *intrinsic motivation*. However, in the machine learning setup it is not clear [65] what it exactly means that an idea is inherently interesting or enjoyable [5] for an autonomous agent. The idea is mostly realized in form of internalized extrinsic motivation by using rewards representing a measure of interestingness of an activity. Intrinsic motivation in machine learning serves mainly the purpose of directed exploration and is referred to as *curiosity*. It is mainly applied if rewards are extremely sparse or for model learning.

3.3.3 Curiosity in Model-free Reinforcement Learning

The first attempts to approach the problem of exploration in more complex models by applying the psychological concept of intrinsic motivation in machine learning were made by Schmidhuber [6]. His basic idea was to make use of reinforcement learning and provide a high reward to the agent whenever there occurs a large mismatch between its expectations of the adaptive world model and reality. To maximise the reward, the agent would then try to execute actions that lead to observations being most inconsistent with its model which leads to exploration behavior. The work in [7] implements a similar approach where an agent learns skills which are internally represented as a collection of so-called options. The agent has internal models of these option and tries to predict the outcome when executing them. As intrinsic reward, the error of the model by executing an option is chosen which makes the agent trying to choose options where the internal model is still non-reliable.

A problem with this formulation is the application in stochastic environments: In this case, the agent's controller will finally focus on parts of environment's dynamics that are random and thus inherently unpredictable. This problem is referred to as noisy-TV problem [34] as it may occur if the agent observes a TV emitting unpredictable content without doing anything. The underlying problem here is that the agent is reinforced in unpredictable situations although his model cannot be improved. To circumvent this problem, in [66] it was tried to learn a mapping from action sequences to expectations of future model improvements. In this approach, only action sequences that are probable to improve the model are encouraged. The encouragement is achieved by providing rewards for model changes or improvements measuring the learning process. Unfortunately, for more complicated models, e. g. deep neural networks, there don't seem to be computationally feasible mechanisms for measuring learning progress [9].

Another way to measure informativeness of an action is to provide exploration bonuses if novel states are reached. Bayesian exploration bonuses [67] measure novelty of state-action pairs as the number of times an agent has executed the action in that state. This approach approximates a Bayesian optimal solution and offers strong formal guarantees but requires an enumerable representation of the environment that renders it impractical for large-scale tasks. Other count-based exploration strategies can be found in [68, 69, 70] where different decreasing functions of the state visitation counts are proposed for computing exploration bonuses in enumerable environments. Attempts to extend count-based novelty measures to the continuous domain were made in [68] by generating pseudo counts using a density model over state-action pairs.

In the last years, exploration bonuses were applied to more complicated non-tabluar environments like video games by making use of deep learning. In [8] a dynamics model is learned based on a state representation obtained by an autoencoder on raw pixel data. The misprediction error of the dynamics model is used as measure of novelty that acts as exploration bonus for model-free reinforcement learning. The work in [9] builds on this approach but tries to circumvent the noisy-TV problem that is focusing on random parts in observations e. g. animations of background in a video game. In this approach a state representation is learned using a self-supervised inverse dynamics model. This model only captures information that is influenced by actions which makes the controller more robust against distracting objects. The prediction error of the dynamics model in this improved visual feature space is then used as reward for the agent.

In [71] the noisy-TV problem is tackled by exploration bonuses based on randomly generated prediction problems. More precisely, the novelty is measured by predicting the output of a fixed and randomly initialized neural network given the following state. It is expected that predictions of the output of the random network is less accurate for novel states that for ones often visited previously which then can be used as measure of novelty. This approach led to impressive results on difficult video game tasks as Montezuma's revenge.

VIME [72] models novelty as the predicted information gain of the belief over dynamics. This reward leads the agent going to states causing large updates to the dynamics model distribution and thus seeming surprising. To make computation of the intrinsic reward practical, the information gain is approximated using variational inference . The agent's dynamic is represented by a bayesian neural network.

3.3.4 Curiosity in Model-based Reinforcement Learning

The previously described methods all employ a reactive exploration strategy. There, the agent accidentally observes novel state-action regions (e. g. by means of prediction error) and only then decides to obtain more information about this region by obtaining a high reward. After exploring this region, the agent finally again has to unlearn the high reward of this region which makes the approaches not very sample-efficient. This problem was already spotted in [10] and called over-commitment. In this work, the problem was approached by using model-based reinforcement learning. An ensemble of neural networks was used as probabilistic model and trajectories were optimized to maximise an approximation of information gain using model-free reinforcement learning in the simulated model environment.

The work in [73] also proposes a more directed form of exploration using model-based reinforcement learning. In this work, an autoencoder on the visited state-action pairs is learned jointly with the system model in form of a neural network. The reconstruction error for state-action pairs serves as uncertainty measure which is motivated by the observation that autoencoders make lower error for the data it was trained on. For exploration, the state-action pairs leading to maximal reconstruction error in the autoencoder are chosen which representing the highest uncertainty of the current model. Our approach is quite similar to both approaches but uses Bayesian linear regression for learning the model which leads to a more natural representation of model uncertainty.

Another very related approach can be found in [74]. In this work, observations are used to learn a Bayesian model and actions are selected via value iteration to optimize various active learning acquisition functions, i. e. predicted information gain, predicted model change, predicted model improvement. In this approach, however, only discrete models were considered.

3.3.5 Further Measures for Specific Tasks

Finally, we take a breve look at further measures for exploration which only are applicable for learning a policy to solve a specific task. In the work of [75], one makes use of sensitivity analysis to find the actions that the current guess of the policy is most sensitive to. These actions are assumed to be particularly informative for refining the policy. In [76] the authors present an approach that makes use of deep generative models. In this approach, they consider the Jacobian of the likelihood to detect non-smooth transitions in latent space which indicate large changes in the movements. These non-smooth transitions indicate where the environment is not accurate modelled yet and new demonstrations for these areas are requested in form of demonstrations.

```
Data: statistical model for approximating objective function $f$
Result: predicted maximum of statistical model given data $\mathscr{D}_n$
initialize $\mathscr{D}_0 = \{\}$
for $i = 1, 2, \ldots$ do
    select $x_i$ by optimizing acquisition function $\alpha$: $x_i \leftarrow \arg\max_{x \in \mathscr{X}} \alpha(x; \mathscr{D}_{n-1})$
    query objective function $f$ to obtain $y_n$
    augment data: $\mathscr{D}_n \leftarrow \mathscr{D}_{n-1} \cup \{(x_n, y_n)\}$
    update statistical model using $\mathscr{D}_n$
```

Algorithm 1: Bayesian Optimization algorithm

3.4 Bayesian Optimization

Bayesian Optimization [77, 18, 78, 79] is a sequential model-based approach for solving continuous optimization problems globally in settings where the true objective function f is unknown. Bayesian Optimization is different from other procedures of global optimization in that it constructs a probabilistic model $\hat{f}$ for f and then makes use of the model to decide at which point to evaluate f next. For each decision, the information of previous iterations is still considered in form of the probabilistic model which makes this method very efficient in terms of function evaluations. Bayesian Optimization is therefore especially useful when evaluations of f are expensive.

Put formally, it finds the optimizer

$$x^* = \arg\min_{x \in \mathscr{X}} f(x)$$

where f is the unknown, possibly noisy objective function and $\mathscr{X}$ a compact set. Although we assume f to be unknown, we need to be able to evaluate it at arbitrary query points x producing stochastic unbiased outputs $y \in R$, i. e. $\mathbb{E}[y \mid f(x)] = f(x)$. In other words, we can only observe f through unbiased noisy point-wise observations y. After N queries, the algorithm makes a final recommendation representing the best estimate of the optimizer.

For finding the minimum, we describe our prior belief over the objective functions in form of a probabilistic model $\hat{f}$ and sequentially refine this model as data is observed. In each iteration, the next location for evaluation x is chosen such that it maximises an *acquisition function* $\alpha : \mathscr{X} \rightarrow \mathbb{R}$ which measures the utility of a data point x with regard to the current belief of the objective function. All evaluations are collected in a set $\mathscr{D}_n = (x_i, y_i)_{i=1,\ldots,n}$. A sketch of the algorithm is shown in algorithm 1.

Several acquisition functions have been proposed that are briefly presented in the following.

Thompson Sampling

In Thompson sampling [80], a function is sampled from the posterior belief over objective functions and this function is maximised. The point with the highest simulated reward is chosen as next candidate for querying the true objective function. The method can be formulated as

$$\alpha_{\mathrm{TS}}(\mathbf{x}) = f(\mathbf{x})$$
$$\text{where } f \sim \hat{f} \mid \mathscr{D}.$$

This method is one of the oldest and simplest acquisition functions. Empirical evaluations have shown good performance however it tends to explore aggressively, especially in high dimensional spaces [78].

Probability of Improvement

This acquisition function [81] estimates the probability of improving (PI) upon the current best function value observation y^*, i. e. measuring $p(f(\mathbf{x}) < y_*)$. This estimation is done by accumulating the posterior probability mass above y^* at $\mathbf{x}$. Maximizing this acquisition function produces the candidate showing maximal probability of improvement. If the posterior distribution of the belief $\hat{f}$ is Gaussian with mean μ and variance σ (like in Bayesian Linear Regression or Gaussian Processes), the acquisition function is given in closed-form by

$$\begin{aligned}\alpha_{\mathrm{PI}}(\mathbf{x}) &= p(\hat{f}(\mathbf{x}) < y^*) \\ &= \Phi\left(\frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)\end{aligned}$$

where Φ is the standard normal cumulative distribution function. This method may work well for optimization but focuses often too much on exploitation [78].

Expected Improvement

The acquisition function of probability of improvement treats all improvements over the current best observation y^* equal. One idea is to take into account also the amount of improvement and weight higher improvements more than smaller ones. This idea brings us to the formulation of expected improvement (EI) over the current best estimate y^*. If $\hat{f}$ is Gaussian again, the formula for EI can also be provided in closed form as

$$\begin{aligned}\alpha_{\mathrm{EI}}(\mathbf{x}) &= \mathbb{E}\left[(y^* - \hat{f}(\mathbf{x}))\,\mathbb{I}(\hat{f}(\mathbf{x}) < y^*)\right] \\ &= (y^* - \mu(\mathbf{x}))\,\Phi\left(\frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)\end{aligned}$$

where $\mathbb{I}$ is the indicator function and ϕ is the standard normal probability density function.

Upper Confidence Bound

In upper confidence bound (UCB) [82], one negotiates exploration and exploitation by being optimistic in the face of uncertainty [78]. More precisely, one selects the point that in the best case scenario (to a certain probability) maximises the objective function. To formalize this idea, the quantiles of the distribution of $\hat{f}$ are computed and the point with highest quantile is queried. The acquisition function can be given by

$$\alpha_{\mathrm{UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})$$

where β is a hyperparameter for selecting the quantile. For UCB one can derive formal bounds for the cumulative regret that one can read more about in the work of [83].

Entropy Search

The motivation for entropy search (ES) [84] is to reduce uncertainty over a distribution over the optimum $p(\mathbf{x}^* \mid \mathscr{D})$ that can derived from $\hat{f}$. In active learning, the goal is then to select the point $\mathbf{x}$ that is expected to cause the largest reduction in entropy to this distribution. The acquisition function can be formulated as

$$\alpha_{\mathrm{ES}}(\mathbf{x}) = \mathbb{H}(\mathbf{x}^\star|\mathscr{D}) - \mathbb{E}_y\mathbb{H}(\mathbf{x}^\star|\mathscr{D} \cup \{(\mathbf{x}, y)\})$$

where $\mathbb{H}(\mathbf{x}^\star \mid \mathscr{D})$ denotes the differential entropy of the distribution $p(\mathbf{x}^\star \mid \mathscr{D})$ derived from the surrogate objective model. Evaluating this acquisition function is not tractable in continuous space as even computing the distribution p cannot be done so approximations have to be made such as discretisation and sampling. There are also some different formulations as in predictive entropy search [85] or max-value entropy search [86] which are more efficient and robust compared to the original method.

Connections to our approach

The underlying problem in Bayesian Optimization is basically the same one as in active learning and dual control: We have to decide for a point to query (corresponds to the action to execute) in order to gain maximal information about the true minimum. A well-chosen selection must again show some trade-off between exploration and exploitation. According to the exploration goal, one should opt for points with high uncertainty regarding the outcome to learn something about the function to optimize in general. On the other hand, the choice should also reflect the goal of exploitation as points should be selected that are probable to minimise the objective function. Like in the case of active learning and dual control, the optimal solution could in principle be formulated as planning in a belief space [32]. Finding the exact optimal solution, however, is not tractable and most methods introduced in literature resort (again) to surrogate objectives such as optimizing an acquisition function representing some greedier objective.

3.5 Empowerment

Empowerment [87, 88] is like curiosity a subconcept of intrinsic motivation. It describes the desire of humans to get into conditions where they feel ”empowered” which means that they can actively control their future and have many different future pathways to choose between. Applied to the concept of autonomous agents, an agent showing empowering behaviour attempts to get into states from where it can easily reach many different states.

Empowerment can be formalized using concepts from information theory [87, 89]: The world is considered as information-theoretic channel converting actions into future sensory states. For providing a definition, we first have

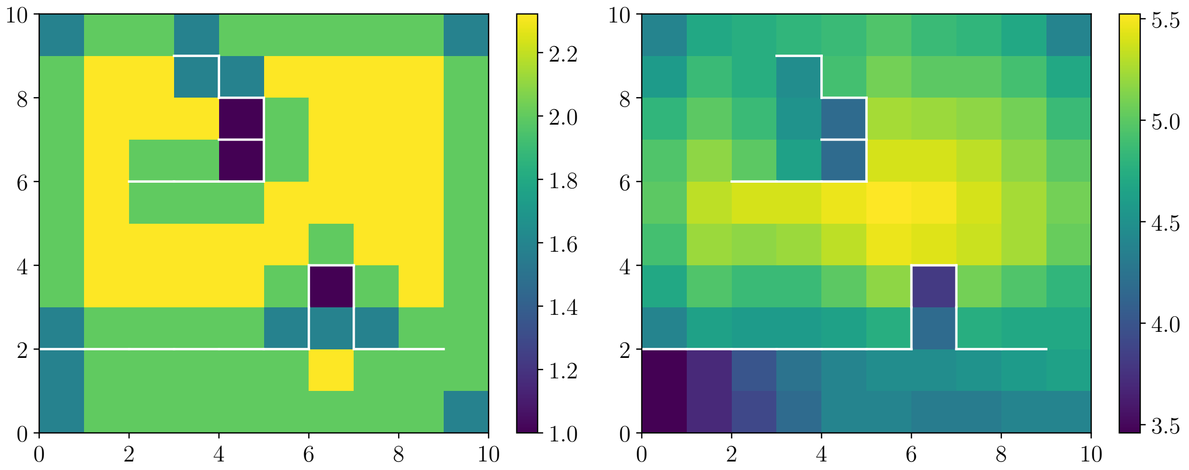


Figure 3.1: This figures shows an example of empowerment, *Klyubin's Maze World* [87]. In this example, an agent can move within a discrete world by executing the actions *stay*, *up*, *down*, *left*, and *right* that make him move into the denoted directions. In the world, there are walls (white) which block movements and make the agent stay in the current field. For each position, the 1-step (left) and 5-step (right) empowerment value are shown which measure the number of different states the agent can reach. 1-step empowerment only takes the next action into account while 5-step empowerment considers the possible states after five subsequent actions. While in case of 1-step empowerment, being close to a wall but not directly next to on one, does not limit the agent's options, 5-step empowerment captures differences in the number of options after taking several actions.

to introduce an information-theoretic measure called *mutual information* (MI). This measure is defined as the amount of information (measured in bits), the received signal on the average contains about the transmitted signal. It is given by

$$\mathcal{I}(\mathrm{x},\mathrm{y}) = \mathbb{E}_{p(y|x)p(x)}\left[\log\left(\frac{p(\mathrm{x},\mathrm{y})}{p(\mathrm{x})p(\mathrm{y})}\right)\right]$$

where $p(x)$ is the probability distribution of the transmitted signal and $p(y \mid x)$ the distribution characterizing the channel. In case of the autonomous agent where the information channel models the conversion from actions in a certain state into future states, we have $p(x) := \pi(a \mid s)$ as policy and $p(y \mid x) := p(s' \mid s, a)$ as transition model where s is the current state, s' is the following state and a the action.

We can now formulate a measure for empowerment using *channel capacity* which is defined as the maximum mutual information for the channel over all possible distributions of the transmitted signal:

$$\mathcal{E}(\mathrm{s}) = \max_{\omega} \mathcal{I}^{\omega}\left(\mathbf{a},\mathbf{s}'|\mathbf{s}\right) = \max_{\omega} \mathbb{E}_{p(s'|a,s)\omega(a|s)}\left[\log\left(\frac{p\left(\mathbf{a},\mathbf{s}'|\mathbf{s}\right)}{\omega(\mathbf{a}|\mathbf{s})p\left(\mathbf{s}'|\mathbf{s}\right)}\right)\right]$$

Put into words, this quantity measures how much information the agent can inject into the future states by means of his actions. Using this formalization, an agent shows high empowerment if each of his actions brings him into different but controllable states.

As introduced until now, only influences of the following action is considered. The extension to multiple actions is straight-forward, but it comes with an increased cost of computation. An example of empowerment in a grid world considering a different number of actions is shown in figure 3.1.

Empowerment was used as reward in reinforcement learning to represent a form of intrinsic motivation e. g. [89, 90]. Within this thesis, empowerment is considered as extension for future work. After having learned a model, an agent could seek to get into a "powerful" state to be prepared to solve a variety of following tasks efficiently.

4 Model-based Reinforcement Learning for System Identification

In this chapter we introduce a novel approach for exploration to the end of learning a model. This approach makes use of a Bayesian model and planning methods to make decisions that are expected to improve the agent's model.

4.1 Problem Formulation

Consider a dynamical system with state space $\mathscr{S} \subset \mathbb{R}^n$ and action space $\mathscr{A} \subset \mathbb{R}^k$. Denote a probabilistic model of the dynamics that an agent has about this system by $\mathscr{M}$. The probabilistic nature of the model enables us to evaluate the strength of belief in predictions. The problem that the agent has to solve is to decide for a sequence of actions $a_0, \ldots, a_{T-1}$ which, executed open-loop on the real system, provides the observations $\mathscr{X}_1 = \{s_t, a_t \rightarrow s_{t+1}\}_{t=0,\ldots,T-1}$ most informative for learning the dynamics. Having already a collection of transition observations $\mathscr{X}_0$, the actions should be chosen such that the model learned with the joint data set $\mathscr{X} = \mathscr{X}_0 \cup \mathscr{X}_1$ is expected to be maximally rich.

This problem can be seen as an active learning problem with the query point being a sequence of actions. This observation enables us to tackle the problem by using active learning approaches from section 3.1 like uncertainty sampling and expected variance reduction for solving the exploration problem.

4.2 Model Variance Measures

To measure the utility of transitions for evaluating active learning utility functions, one has to reason about the variance of our predictor model. There are two possible ways to measure this quantity.

4.2.1 Variance of Model Parameters

This variance formulation can be applied in case of Bayesian parametric models $\mathscr{M}$ that maintain a distribution over parameters $\boldsymbol{\theta}$. The variance of the dynamics model can then be defined as the variance of the probability distribution $p_{\mathscr{M}}(\boldsymbol{\theta} \mid \mathscr{X})$ over the parameters trained of the model on the data:

$$\operatorname{var}(\mathscr{M} \mid \mathscr{X}) = \boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \boldsymbol{V}_{p_{\mathscr{M}}}(\boldsymbol{\theta} \mid \mathscr{X})$$

Usually, the model consists of multiple parameters thus the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ of the parameter is a matrix. As we need a scalar model as measure to optimize, a better measure of model uncertainty is given by the entropy $\mathbb{H}(\boldsymbol{\theta} \mid \mathscr{X})$ of the parameter distribution. If the model parameters are Gaussian distributed, the entropy is given by

$$\mathbb{H}(\boldsymbol{\theta} \mid \mathscr{X}) = \frac{1}{2}\ln(|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|) + \frac{k}{2}\ln(2\pi e). \tag{4.1}$$

Further, if the parameters are independently distributed, the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is a diagonal matrix and the entropy is given by

$$\begin{aligned}\mathbb{H}(\boldsymbol{\theta} \mid \mathscr{X}) &= \frac{1}{2}\sum_i \ln(|\sigma_{\theta_i}|) + \frac{k}{2}\ln(2\pi e) \\ &\propto \sum_i \ln(|\sigma_{\theta_i}|).\end{aligned}$$

The latter formula can be seen as approximation of the entropy objective. In computationally complex cases, it might be advantageous to choose this approximate quantity as measure.

4.2.2 Variance of Predictive Distribution

In system identification of deterministic environments, our goal is usually to obtain a predictor that precisely models dynamics and thus is certain in its predictions. The entropy of the model parameters does not compulsorily reflect this measure as the model parameters can influence predictions to different extends. Another formulation is therefore to evaluate the variance of the predictive distribution directly. As the predictive distribution is dependent on state and action, we have to take the expectation over these variables. Usually, we don't know these distributions as they depend on the application. In this case we can opt for uniform distributions to make all state-action combinations are equally important.

Given the probabilistic transition function $p(s' \mid s, a)$ of the model $\mathscr{M}$, we can compute the expected (average) variance of the predictive distribution (given a fixed data set $\mathscr{X}$) by

$$\mathrm{var}_{\mathbb{V}}(\mathscr{M} \mid \mathscr{X}) = \mathop{\mathbb{E}}_{s,a} \mathbb{V}_p[s' \mid s, a].$$

This quantity is usually not solvable in closed form but we can approximate it using a set of N (e. g. uniformly-distributed) samples of state-action pairs:

$$\mathrm{var}_{\mathbb{V}}(\mathscr{M} \mid \mathscr{X}) \approx \frac{1}{2} \sum_{i=1}^{N} \mathbb{V}_p[s' \mid s_i, a_i] \tag{4.2}$$

MacKay [37] proposed using a fixed set of reference samples to measure the expected change in variance over them. This, however, was found to lead to local maxima at the reference points. In [91] it was therefore suggested to iteratively draw new reference point samples in each iteration.
Like in the model parameter variance case of the variance, to get a scalar objective, it is advantageous to resort again to entropy as variance measure:

$$\mathrm{var}_{\mathbb{H}}(\mathscr{M} \mid \mathscr{X}) \approx \frac{1}{2} \sum_{i=1}^{N} \mathbb{H}_p[s' \mid s_i, a_i]$$

Using predictive distribution variance for measuring utility has the advantage in comparison to model parameter variance that it compensates for parameter influences. A drawback is that it can usually only be evaluated approximately by sampling and therefore can become expensive when using this quantity as optimization objective.

4.3 Maximal Variance Reduction

When using the acquisition function for maximal variance reduction, one considers the expected variance of the future model and selects instances to minimise this quantity. This acquisition function was shown to lead to good results but it is quite expensive in computation [28]. Only the case where one considers solely effects due to the next action and ignores the further future, can be solved exactly. For the general case, one has to make approximations.

If we denote the probabilistic model transition function with p, the objective to minimize the expected variance of the future model can be formulated as

$$\min_{a_0, \ldots, a_{T-1}} \mathop{\mathbb{E}}_{p} \mathrm{var}(\mathscr{M} \mid \mathscr{X}_0 \cup \{s_t, a_t \rightarrow s_{t+1}\}_{t=0,\ldots,T-1}) \tag{4.3}$$

where the transitions follow the current model. In this formulation, s_t are random variables for $t = 1, \ldots, T-1$ which follow $s_{t+1} \sim p(s_{t+1} \mid s_t; a_t)$. $\mathrm{var}(\mathscr{M} \mid \mathscr{X})$ is a measure for the variance of the model as defined in formula (4.1) or (4.2). The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta} \mid \mathscr{X}}$ in dependence on the observations is given in equation (2.1) for the case of Bayesian linear regression.

The objective in (4.3) can in general not be evaluated in closed form as random state variables s_t are pushed through the (usually non-linear) probabilistic model p. One exception is the case of horizon $T = 1$ where one action is selected to improve the model greedily only for the next time step without further planning. To solve the general case, one would have to resort to approximate inference methods such as sampling or variational inference like applied in algorithm PILCO [92]. These approaches, however, are very costly for planning so we opt for a simpler approximation.

```
Data: number of episodes $N$, horizon $T$, initial model $\mathscr{M}_0$
Result: optimized model $\mathscr{M}_N$
for $i \leftarrow 1$ to $N$ do
    find actions $a_{0:T-1}$ that optimize (4.5) or (4.4) given the current model $\mathscr{M}_i$;
    execute $a_{0:T-1}$ in the environment and observe $s_{0:T}$;
    update the model $\mathscr{M}_{i+1}$ via (2.1) using $\mathscr{M}_i$ as the prior and $(a_{0:T-1}, s_{0:T})$ in the likelihood;
```

Algorithm 2: Receding Horizon Curiosity

We adopt a rather crude approximation known as the maximum likelihood observations assumption [93], which amounts to propagating only the mean of the state distribution. The optimization problem can then be formulated as

$$\begin{aligned} \min_{a_0,\ldots,a_{T-1}} \quad & \operatorname{var}(\mathscr{M} \mid \mathscr{X}_0 \cup \{\hat{s}_t, a_t \to \hat{s}_{t+1}\}_{t=0,\ldots,T-1}) \\ \text{s.t.} \quad & \hat{s}_{t+1} = \mathop{\mathbb{E}}_{p}[s_{t+1} \mid \hat{s}_t, a_t], \quad t = 1, \ldots, T-1, \\ & \hat{s}_0 = s_0 \end{aligned} \tag{4.4}$$

where we replaced random variables s_t by their expectation $\hat{s}_t$.
The crudeness of the approximation is offset by the computational advantage: More frequent replanning enabled by neglecting the expensive state uncertainty propagation allows the agent to efficiently compensate for unforeseen deviations from the planned trajectory.

Planning using the objective of maximal variance reduction requires differentiation through a matrix inversion due to the dependence of $\mathbf{\Sigma}_*$ on the kernel matrix inverse (2.1). Combined with the chain-like structure of state-action dynamics, gradient computation becomes quite expensive for larger models (e. g. $T > 100$, $M > 40$) which makes the uncertainty approach presented in the following section more appealing for these cases.

4.4 Uncertainty Sampling

Uncertainty sampling queries the instances that the model is most uncertain about. In our planning case, we optimize the rollout of T actions such that the trajectory has maximal entropy according to the model. Let $p(s' \mid s, a)$ denote the prediction of the probabilistic model $\mathscr{M}$ trained on the previous observed data $\mathscr{X}$. The objective can then be formulated as

$$\max_{a_0,\ldots,a_{T-1}} \quad \sum_{t=1,\ldots,T} \mathbb{E}_{s_{t-1} \sim p(s_{t-1} \mid a_{t-2},\ldots,a_0)} \mathbb{V}_p[s_t \mid s_{t-1}; a_{t-1}].$$

Like in the case of maximal variance reduction, for evaluating the objective exactly, we would have to propagate the distribution over states through the model. Thus, we solve again only an approximate version of this problem:

$$\begin{aligned} \max_{a_0,\ldots,a_{T-1}} \quad & \sum_{t=1,\ldots,T} \mathbb{V}_p[s_t \mid \hat{s}_{t-1}, a_{t-1}] \\ \text{s.t.} \quad & \hat{s}_t = \mathbb{E}_p[s_t \mid \hat{s}_{t-1}, a_{t-1}], \quad t = 1, \ldots, T \end{aligned} \tag{4.5}$$

The uncertainty sampling objective is usually much easier to solve than maximal variance reduction. In case of Bayesian linear regression models, for uncertainty sampling the used parameter variance matrix $\mathbf{\Sigma}_*$ remains constant thus doesn't depend on the states and actions to optimize for.

4.5 Optimization Procedure

Both optimization problems in (4.4) or (4.5) offer both objectives and constraints in closed form which are differentiable. Therefore, they can be solved using the multiple shooting method described in section 2.2. The complete exploration algorithm which we refer to as *Receding Horizon Curiosity* is summarized in algorithm 2.

5 Experiments

In this chapter we show experiments where we compare our proposed exploration algorithm, Receding Horizon Curiosity, against state-of-the art exploration methods. The method is compared to random actions, a model-free intrinsic motivation approaches with exploration bonuses as rewards, and to the model-based active exploration (MAX) algorithm.

5.1 Evaluated Algorithms

Receding Horizon Curiosity (RHC)

For Receding Horizon Curiosity, the optimization steps as presented in algorithm 2 were executed. The uncertainty sampling (RHC US) objective was considered for all experiments, while the method in the expected variance reduction (RHC EVR) formulation could only be applied to low-dimensional models. Planning was done by means of multiple shooting using the casadi framework [13].

Model-free Reinforcement Learning with Exploration Bonus (SAC)

Model-free reinforcement learning algorithms using popular exploration bonuses for exploration were applied to the environments. As underlying model-free reinforcement learning algorithm, the sample-efficient soft actor critic (SAC) [94] was put to use. For running the algorithm, the implementation by stable-baselines [95] was taken. As rewards, two common curiosity measures were used:
Firstly, the squared prediction error

$$r_{\text{pe}}(s_t, a_t) = (s_{t+1} - \mathbb{E}_p[s_{t+1} \mid s_t, a_t])$$

of model p trained on the data of the past episodes is considered as reward.
Secondly, as measure of information gain, the reduction of entropy

$$r_{\text{pe}}(s_t, a_t) = \mathbb{H}(\boldsymbol{\theta} \mid \mathcal{X}_t) - \mathbb{H}(\boldsymbol{\theta} \mid \mathcal{X}_{t+1})$$

was used as reward where $\mathcal{X}_t$ denotes the set of all observations until step t, $\mathbb{H}$ is the entropy and $\boldsymbol{\theta}$ the model parameters. We further tested DDPG [96] as model-free reinforcement learning algorithm on exploration bonus rewards but found only negligible differences to SAC in performance thus we omitted this algorithm from the evaluation. SAC was executed using the following hyper parameters: $\gamma = 0.99$, $\tau = 0.005$, the learning rate was set to 0.003, the buffer size to 0.0004, and a batch size of 64 was used.

Model-based Active Exploration (MAX)

We further applied the recently proposed model-based active exploration (MAX) algorithm [10] to the environments for learning a Bayesian linear regression model. This algorithm applies model-free reinforcement learning in a learned MDP instead of the true environment to obtain an exploratory policy and is therefore more sample-efficient. As rewards, it uses a measure of disagreement of a neural network ensemble.

Random (RAND)

In the environments, uniformly-random actions within the allowed bounds were applied. This method shows no directed way of exploration.

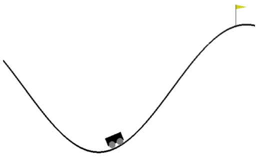
5.2 Experimental Setup

The experiments were carried out as follows: For RHC, according to algorithm 2 in each episode a sequence of action was computed and executed open-loop in the environment. As planning for the expected variance reduction objective is only tractable for small models, this objective formulation of RHC could only be considered in the mountaincar environment. In the case of MAX and SAC, the respective policies were applied in the environment. After each episode, a Bayesian linear regression model with random Fourier features was retrained using all observations of past episodes. After exploration, the models of each stage were evaluated. First, the mean likelihood of a trajectory test set under the learned model was computed. This test set consisted of trajectories based on uniformly-random sampled starting states to which a sequence of uniformly sampled actions was applied. Second, an environment-specific task was tried to be solved using the learned model. A sequence of actions for solving the task was planned using the multiple shooting method with the learned model, executed in the environment and the total cost of the trajectory calculated.

5.3 Environments

In the following, we present the environments that the algorithms were applied to for investigating their performance.

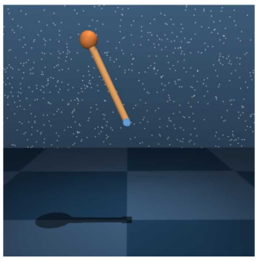
5.3.1 Mountain Car



The continuous mountain car in the openai gym implementation [97]. In this environment a car is positioned on a one-dimensional track in a valley between two mountains. One has to control the car by applying accelerations to drive up the mountain. As the car is underpowered, one needs to build up momentum by driving back and forth to accomplish the task.

The power of the engine was set to 10^{-3} and no speed limit was applied. Observations were of the form $(x, \dot{x})$ with x being the position of the car. At each start of the episode the car was set into the center of the valley with zero velocity. In exploration trials the episode was ended when the car reached the bounds of the environment or the number of 150 steps was reached. The evaluation task consisted of planning to drive to a goal on a mountain by using a cost for the last step of $10(x - x_{\text{goal}})^2 + 0.01a$ where a is the acceleration. For the model, the number of features was set to 20. Due to the rather low dimensionality, both objectives of minimal future model variance and maximal trajectory entropy could be realized.

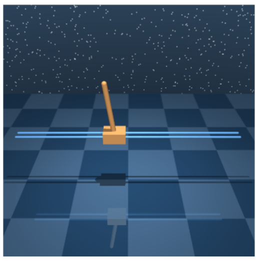
5.3.2 Pendulum



The inverted pendulum from the dm control suite [98]. The torque-limited actuator is 1/6-th as strong as required to lift the mass directly from the motionless starting state so that several swings are required to swing it up and balance.

Observations were of the form $\left(\cos(\theta), \sin(\theta), \dot{\theta}\right)$ where θ is the angle of the pendulum. In the beginning of each episode, the pendulum was set hanging down without velocity. Each episode consisted of 100 steps of 80 ms each. The evaluation task was to swing up the pendulum using a cost of $100\cos(\theta) + 0.1\sin(\theta) + 0.1\dot{\theta} + 0.001a$ for each step. For the model, the number of features was set to 90. As for this model, planning for maximal variance reduction was no longer feasible, only uncertainty sampling was considered as objective in RHC.

5.3.3 Cartpole



The cartpole from the dm control suite [98]. A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force to the cart.

Observations of the cartpole were given as $\left[x, \cos(\theta), \sin(\theta), \dot{x}, \dot{\theta}\right]$. The system was simulated with 50 Hz. Each episode started with the cartpole centered on the track without velocity and the pole hanging downwards. The episodes were ended if the cartpole was driven against the bounds or the a maximum step of 100 was reached. The evaluation task was to swing up the pole by providing a cost of $100x + 100\cos(\theta) + 0.1\sin(\theta) + 0.1\dot{x} + 0.1\dot{\theta} + 0.1a$ in each step. For the model, the number of features was set to 80.

5.4 Results

The results of the experiments are shown in Fig. 5.1. In each environment, RHC was the only algorithm that approximately reached within 20 episodes the lowest possible model log-likelihood as roughly indicated by the model performance using random transition samples. Both SAC variants almost performed to the same extend as RAND which can be explained by their *over-commitment* behaviour. High rewards are obtained for nearly all executed actions in the beginning which need to be unlearned over time moving to more distant areas. MAX did not suffer from this problem as it applies an active way of exploration, and therefore performed better than SAC and random exploration. In the mountaincar experiments, one can observe that RHC MVR is slightly superior to RHC US which is matches our intuition from

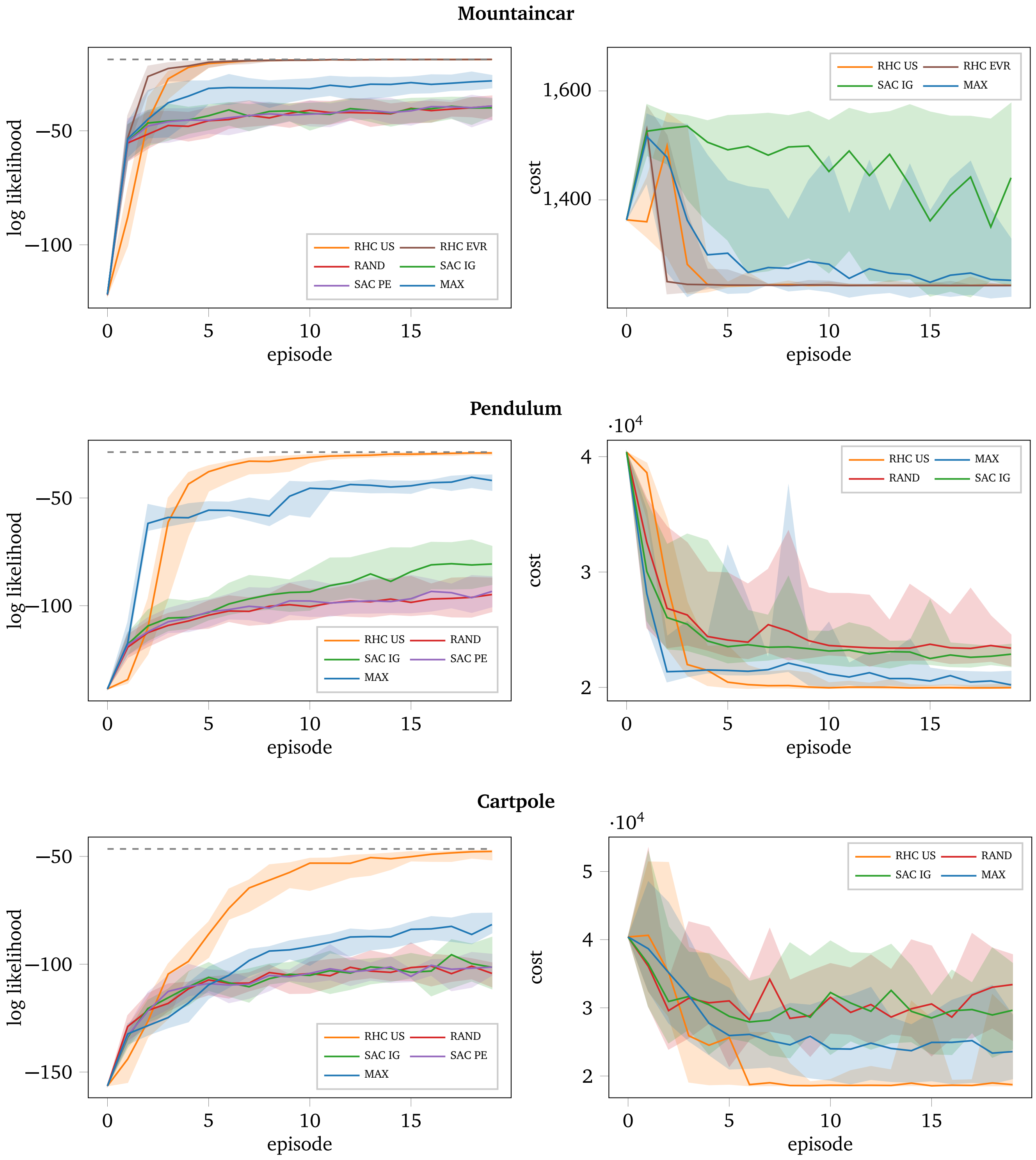


Figure 5.1: Evaluation of models learned with different exploration methods. In the figures on the left, the log likelihood of random 10-step trajectories was evaluated on the learned models. The median with 1st and 9th deciles over 20 different exploration runs are shown. The grey dashed lines indicate the likelihood obtained by a model trained on 10^4 uniform transition samples from the full state-action space and therefore approximates the lowest possible log-likelihood for this model class. Both RHC MVR (only mountaincar) and RHC US led within few episodes to models that reach this lower bound. In the figures on the right, the accumulative costs for solving tasks based on the learned models are shown.

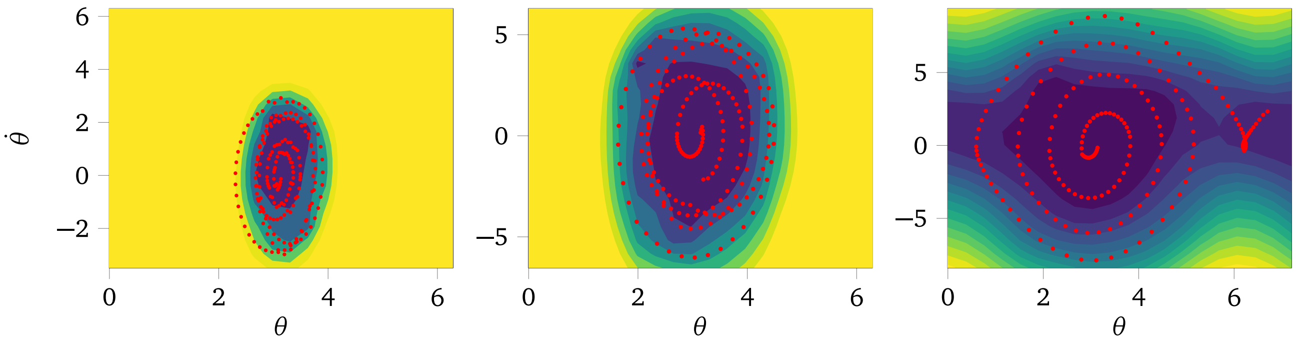


Figure 5.2: Exploration of RHC on the pendulum. Episodes 2, 3 and 6 – from left to right – are shown. The background indicates the entropy of predictions for applying zero action in the given state, assessed by the model trained on the observations from past episodes. Warm colors indicate high entropy (uncertainty). RHC tries to find a trajectory of maximum uncertainty which is consistent with the dynamics provided by model. In episode 6 the pendulum does a full swing-up to get into regions where the model is still inaccurately modelled.

the theoretical viewpoint. Regarding the evaluations for solving tasks, models learned by RHC can mostly successfully plan to solve the task. The other methods show higher varying model performance. Examples for trajectories generated by RHC under the uncertainty sampling objective are shown in Fig. 5.2.

6 Conclusion and Outlook

In this thesis we investigated the problem of exploration for learning forward models of the environment. To this end, we reviewed existing work on exploration strategies and optimal decision making for information gain from various subfields of machine learning. By applying insights from the field of active learning, we formulated two optimisation objectives to plan trajectories of maximal utility for the model. For making these problems tractable, we proposed an approximation which led to a novel algorithm for exploration. This algorithm was named *Receding Horizon Curiosity* (RHC) and can be used with either of the two objectives. While the objective of maximum variance reduction is superior to the one of uncertainty sampling from a theoretical viewpoint, the former is much harder to solve and thus can only be applied to very small problems. We compared our new method to state-of-the-art exploration algorithms, i. e. model-free reinforcement learning with exploration bonuses and the recently proposed model-based active exploration algorithm (MAX). Empirically, we found that RHC reached higher model likelihoods within few episodes and produced more reliably models for solving tasks. Additionally, it ran much faster than the main competitive algorithm, MAX.

One downside of our proposed algorithm in its current form is that it is limited to rather small models. As we do planning in form of multiple shooting, derivatives of the model need to be computed and Hessians approximated. An interesting future direction is to investigate how to incorporate models that scale well, e. g. Bayesian neural networks, and how to apply planning in this case.

Another interesting branch for future work is to investigate further approximations for the original optimisation problem that we presented in chapter 4. For obtaining a tractable planning problem, in this thesis we adopted a rather crude approximation using a maximum likelihood assumption. In this approximation, only the mean of the state distribution is propagated through the probabilistic model. Other approximate inference methods such as sampling and variational inference might be used to obtain better solutions but probably will increase the cost of computational complexity.

Finally, in our proposed algorithm, there is no stopping criterion. Even if the model cannot further be improved, an agent would still try to get to the state which is most informative in relation to other states. A better objective to prepare for upcoming tasks would be to then move to a state where it is able to quickly solve a variety of upcoming tasks. This behaviour is known as empowerment and a formalisation of that concept was briefly introduced in section 3.5. An interesting future direction would be to investigate how to combine both exploration and empowerment in a meaningful way.

Bibliography

[1] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[2] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189, 2011.

[3] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.

[4] Paul J Silvia. Curiosity and motivation. *The Oxford handbook of human motivation*, pages 157–166, 2012.

[5] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[6] Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991.

[7] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2005.

[8] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[9] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[10] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. *arXiv preprint arXiv:1810.12162*, 2018.

[11] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[12] Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*, volume 12. Springer Science & Business Media, 2013.

[13] Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, pages 1–36, 2018.

[14] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[15] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[16] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[17] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[18] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[19] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[20] Alex J Smola and Peter L Bartlett. Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625, 2001.

[21] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

[22] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[23] Alexander G de G Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. *Journal of Machine Learning Research*, 51:231–239, 2016.

[24] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[25] Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems*, pages 6550–6561, 2017.

[26] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[27] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[28] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[29] Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pages 566–573, 1990.

[30] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[31] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[32] Marc Toussaint. Bandits, global optimization, active learning, and bayesian reinforcement learning – understanding the common ground. *Autonomous Learning Summer School, Leipzig*, 2014.

[33] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[34] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[35] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[36] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[37] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[38] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[39] Björn Wittenmark. Adaptive dual control methods: An overview. In *Adaptive Systems in Control and Signal Processing 1995*, pages 67–72. Elsevier, 1995.

[40] AA Feldbaum. Dual control theory. i. *Avtomatika i Telemekhanika*, 21(9):1240–1249, 1960.

[41] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[42] Edgar D Klenske and Philipp Hennig. Dual control for approximate bayesian reinforcement learning. *Journal of Machine Learning Research*, 17(127):1–30, 2016.

[43] Jan Sternby. A simple dual control problem with an analytical solution. *IEEE Transactions on Automatic Control*, 21(6):840–844, 1976.

[44] DJ Hughes and OLR Jacobs. Turn-off, escape and probing in nonlinear stochastic control. In *IFAC Symposium Adaptive Control, Budapest, Hungary*, 1974.

[45] Jacob Alster and Pierre R Bélanger. A technique for dual adaptive control. *Automatica*, 10(6):627–634, 1974.

[46] Björn Wittenmark and Craig Elevitch. An adaptive control algorithm with dual features. *IFAC Proceedings Volumes*, 18(5):587–592, 1985.

[47] RRAP Milito, C Padilla, R Padilla, and D Cadorin. An innovations approach to dual control. *IEEE Transactions on Automatic Control*, 27(1):132–137, 1982.

[48] Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 697–704. ACM, 2006.

[49] Tao Wang, Daniel Lizotte, Michael Bowling, and Dale Schuurmans. Bayesian sparse sampling for on-line reward optimization. In *Proceedings of the 22nd international conference on Machine learning*, pages 956–963. ACM, 2005.

[50] Arthur Guez, David Silver, and Peter Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in neural information processing systems*, pages 1025–1033, 2012.

[51] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[52] Kendall Lowrey, Aravind Rajeswaran, Sham Kakade, Emanuel Todorov, and Igor Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

[53] Stephane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

[54] Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 1–8. ACM, 2005.

[55] Raman Mehra. Optimal input signals for parameter estimation in dynamic systems–survey and new results. *IEEE Transactions on Automatic Control*, 19(6):753–768, 1974.

[56] Martin B Zarrop. *Optimal experiment design for dynamic system identification*, volume 21. Springer, 1979.

[57] Lennart Ljung. Asymptotic variance expressions for identified black-box transfer function models. *IEEE transactions on Automatic Control*, 30(9):834–844, 1985.

[58] Henrik Jansson and Håkan Hjalmarsson. Convex computation of worst-case criteria with applications in identification and control. In *IEEE Conference on Decision and Control*, page 3132–3137. The Bahamas, 2004.

[59] J Moore. Persistence of excitation in extended least squares. *IEEE Transactions on Automatic Control*, 28(1):60–68, 1983.

[60] Masanao Aoki and RM Staley. On input signal synthesis in parameter identification. *Automatica*, 6(3):431–440, 1970.

[61] Michel Gevers and Xavier Bombois. Input design: From open-loop to control-oriented design. *IFAC Proceedings Volumes*, 39(1):1329–1334, 2006.

[62] Edward L Deci and Richard M Ryan. Intrinsic motivation. *The corsini encyclopedia of psychology*, pages 1–2, 2010.

[63] Robert W White. Motivation reconsidered: The concept of competence. *Psychological review*, 66(5):297, 1959.

[64] Edward L Deci and Richard M Ryan. The general causality orientations scale: Self-determination in personality. *Journal of research in personality*, 19(2):109–134, 1985.

[65] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

[66] Jürgen Schmidhuber. Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 1458–1463. IEEE, 1991.

[67] J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 513–520. ACM, 2009.

[68] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[69] Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2721–2730. JMLR. org, 2017.

[70] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[71] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[72] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016.

[73] Norman Di Palo and Harri Valpola. Improving model-based control and active exploration with reconstruction uncertainty optimization. *arXiv preprint arXiv:1812.03955*, 2018.

[74] Daniel Ying-Jeh Little and Friedrich Tobias Sommer. Learning and exploration in action-perception loops. *Frontiers in neural circuits*, 7:37, 2013.

[75] Arkady Epshteyn, Adam Vogel, and Gerald DeJong. Active reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 296–303. ACM, 2008.

[76] Nutan Chen, Alexej Klushyn, Alexandros Paraschos, Djalel Benbouzid, and Patrick Van der Smagt. Active learning based on data uncertainty and model sensitivity. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1547–1554. IEEE, 2018.

[77] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[78] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[79] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[80] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[81] Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[82] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[83] Nando De Freitas, Alex Smola, and Masrour Zoghi. Exponential regret bounds for gaussian process bandits with deterministic observations. *arXiv preprint arXiv:1206.6457*, 2012.

[84] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

[85] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

[86] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3627–3635. JMLR. org, 2017.

[87] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. All else being equal be empowered. In *European Conference on Artificial Life*, pages 744–753. Springer, 2005.

[88] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Keep your options open: an information-based driving principle for sensorimotor systems. *PloS one*, 3(12):e4018, 2008.

[89] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 2125–2133, 2015.

[90] Guido Montúfar, Keyan Ghazi-Zahedi, and Nihat Ay. Information theoretically aided reinforcement learning for embodied agents. *arXiv preprint arXiv:1605.09735*, 2016.

[91] David A Cohn. Neural network exploration using optimal experiment design. In *Advances in neural information processing systems*, pages 679–686, 1994.

[92] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[93] Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

[94] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[95] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

[96] 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.

[97] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[98] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[99] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[100] Dustin J Webb, Kyle L Crandall, and Jur van den Berg. Online parameter estimation via real-time replanning of continuous gaussian pomdps. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5998–6005. IEEE, 2014.

[101] 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.

[102] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[103] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

[104] J Andrew Bagnell, Sham M Kakade, Jeff G Schneider, and Andrew Y Ng. Policy search by dynamic programming. In *Advances in neural information processing systems*, pages 831–838, 2004.

[105] Oliver Kroemer, Renaud Detry, Justus Piater, and Jan Peters. Active learning using mean shift optimization for robot grasping. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2610–2615. IEEE, 2009.

[106] Antonio Morales, Eris Chinellato, Andrew H Fagg, and Angel Pasqual del Pobil. An active learning approach for assessing robot grasp reliability. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 485–490. IEEE, 2004.

[107] Sebastian B Thrun and Knut Möller. Active exploration in dynamic environments. In *Advances in neural information processing systems*, pages 531–538, 1992.

[108] BW Mel. A neurally-inspired connectionist approach to learning and performance in vision-based robot motion planning. *Technical Reprot CCSR-89-17, Center for Complex Systems Research, Beckman Institute, University of illinois*, 405, 1989.

[109] Michael C Mozer and Jonathan Bachrach. Discovering the structure of a reactive environment by exploration. In *Advances in neural information processing systems*, pages 439–446, 1990.

[110] Steven D Whitehead and Dana H Ballard. *A study of cooperative mechanisms for faster reinforcement learning*. University of Rochester, Department of Computer Science Rochester, NY, 1991.

[111] Sebastian B Thrun. Efficient exploration in reinforcement learning. 1992.

[112] Sebastian B Thrun and Knut Möller. *On planning and exploration in non-discrete environments*. GMD Sankt Augustin, Germany, 1991.

[113] Roger McFarlane. A survey of exploration strategies in reinforcement learning. *McGill University, http://www.cs.mcgill.ca/∼cs526/roger.pdf*, 2018.

[114] Jürgen Schmidhuber. Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer, 1991.

[115] Marco Wiering and Jürgen Schmidhuber. Efficient model-based exploration. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, volume 6, pages 223–228, 1998.

[116] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[117] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. Monte-carlo tree search: A new framework for game ai. In *AIIDE*, 2008.

[118] Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in Neural Information Processing Systems*, pages 206–214, 2012.

[119] Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[120] Jason Pazis and Ronald Parr. Pac optimal exploration in continuous space markov decision processes. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[121] Daniel E Berlyne. Conflict, arousal, and curiosity. 1960.

[122] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

[123] Rowan McAllister, Mark van der Wilk, and Carl Edward Rasmussen. Data-efficient policy search using pilco and directed-exploration. 2016.

[124] OLR Jacobs and JW Patchell. Caution and probing in stochastic control. *International Journal of Control*, 16(1):189–199, 1972.