

Chapter 2

The Explore–Exploit Dilemma in Nonstationary Decision Making under Uncertainty

Allan Axelrod and Girish Chowdhary

Abstract It is often assumed that autonomous systems are operating in environments that may be described by a stationary (time-invariant) environment. However, real-world environments are often nonstationary (time-varying), where the underlying phenomena changes in time, so stationary approximations of the nonstationary environment may quickly lose relevance. Here, two approaches are presented and applied in the context of reinforcement learning in nonstationary environments. In Sect. 2.2, the first approach leverages reinforcement learning in the presence of a changing reward-model. In particular, a functional termed the Fog-of-War is used to drive exploration which results in the timely discovery of new models in nonstationary environments. In Sect. 2.3, the Fog-of-War functional is adapted in real-time to reflect the heterogeneous information content of a real-world environment; this is critically important for the use of the approach in Sect. 2.2 in real world environments.

2.1 Introduction

When making decisions under uncertainty, one of the fundamental questions that arises in reinforcement learning (RL) is whether the autonomous agent should utilize their (still imperfect) knowledge to decide on the action to take (exploit), or whether the agent should try out something new (explore) to gain more knowledge. The resulting explore–exploit dilemma is common to almost all facets of adaptive decision making in the real world (Wilson et al. 1996). The explore–exploit dilemma is therefore the study of the trade-off between trying something new in the hope of better reward and risking disappointment, or worse, failure. This dilemma is famously embodied by the N-armed bandit problem, where a gambler must decide whether to keep playing at a slot machine, or to try playing at a new one

A. Axelrod (✉) · G. Chowdhary
Oklahoma State University, 218 Engineering North, Stillwater, OK 74078, USA
e-mail: allanma@okstate.edu

G. Chowdhary
e-mail: girish.chowdhary@okstate.edu

(Koulouriotis and Xanthopoulos 2008; Garivier and Moulines 2008; Granmo and Berg 2010; Gur et al. 2014). The explore–exploit dilemma is widely regarded as one of the fundamental problems of adaptive decision making under uncertainty, and is specifically deeply studied in the field of reinforcement learning (Sutton and Barto 1998; Garivier and Moulines 2008; Gur et al. 2014; Koulouriotis and Xanthopoulos 2008; Granmo and Berg 2010; Geramifard et al. 2013). Fundamental results in RL show that sufficient exploration is necessary for the RL agent to find optimal policies (Watkins and Dayan 1992; Tsitsiklis and Roy 1997; Busoniu et al. 2010). Many of the techniques proposed to tackle the exploration-exploitation dilemma rely on some measure of “knownness” or “experience” of the agent. In essence, these techniques favor exploration in the early stage of the learning problem—when the agent knows little about the environment— and exploitation in the latter stages of the learning problem (Walsh et al. 2010; Kakade et al. 2003; Grande et al. 2014; Kolter and Ng 2009; Pazis and Parr 2013). For example, in the context of the well-known on-policy temporal-difference based techniques SARSA (Tsitsiklis and Roy 1997; Geramifard et al. 2013), one of the classical results in RL states that an ε -greedy strategy leads to convergence to the optimal value function. In the ε -greedy policy, the RL agent takes a random action with a probability ε and chooses an optimal action with the current estimate of the value function with a probability $1 - \varepsilon$. The value ε is decreased as a function of time or the number of state-action-reward samples analyzed. Similarly, in the context of the actor–critic temporal-difference techniques for reinforcement learning, a similar framework is adopted wherein asymptotic convergence to some stationary and optimal exploitative policy is expected (Grondman et al. 2012).

An assumption that is implicit to almost all “knownness” based techniques however is that the environment in which the agent operates is stationary, that is, it is not changing with time. This assumption enables the agent to utilize state visitation index to determine whether it is likely to have gained enough experience to trust the model it has built. However, many real-world environments are often dynamically changing, that is they are nonstationary. When the environment is nonstationary, state-visitation index based ideas, or open-loop ε -greedy type ideas are not likely to succeed. One of the key open problems in decision making in dynamically changing environments therefore is how to tackle the explore–exploit dilemma in the presence of time-variations. In particular, when the environment is dynamically changing, the following two questions become increasing important: (1) What signals should the agent use from the environment to adjust its bias towards exploration or exploitation when the underlying reward, transition, or observation models are dynamically changing, and (2) Can the agent learn to anticipate change in the environment and utilize a *meta-model* on expected change to pro actively adapt its exploration-exploitation bias.

In this book chapter, we take the position that when the environment is dynamically changing, an open loop explore–exploit strategy will not lead to optimal decision making by the agent. In fact, the agent will have to continuously question whether the model or policies that it has learned are still applicable, and decide to learn new policies if it perceives that the environment has changed. This leads to closed-loop exploration-exploitation strategies in which the agent uses the dif-

ference between expected and actual transitions or rewards as a feedback signal to tackle the explore–exploit dilemma. In the study of human behavior in handling exploration–exploitation tradeoff, it has been observed that a combination of random and information-directed behaviors is often most successful when the environment is uncertain and dynamically changing (Reverdy et al. 2012; Daw et al. 2006; Wilson et al. 2014). Building on the same vein of thought, we will show that a combination of two classes of explore–exploit strategies: *reactive* and *proactive*, are essential in handling the explore–exploit tradeoff in nonstationary environments. The reactive strategy depends purely on the agent adapting its exploration strategy after it has detected that the environment has changed. However, this strategy leads to behaviors that are impervious to changes in other parts of the state-space that are not visited by the agent when it is executing a previously learned optimal strategy. For example, a purely reactive strategy cannot detect the emergence of better options in some other part of the state-space due to a change in the environment. We argue that in order to handle this, the agent needs to anticipate when it is optimal to engage in exploratory behavior. The main question then becomes: What should be the optimal frequency of the proactive exploration? We show that, this question can be addressed by enabling the agent to build a model on the expected change in the environment and use this knowledge to adapt its behavior in anticipation.

To ground our discussion, we will focus on two case studies and associated recently introduced algorithms for nonstationary decision making: In the first case study, we investigate the problem of finding optimal policies using model based RL when the reward function is nonstationary; specifically, the problem of finding least-visible paths for Unmanned Aerial Systems (UAS) over populated areas with changing population densities. We will discuss the Gaussian Process Non-Bayesian Clustering (GP-NBC) based Model-Based Reinforcement Learning (MBRL) (Allamaraju et al. 2014) for solving nonstationary RL problems, and introduce the Fog-of-War (FoW) functional, designed to dilute the agent’s predictive confidence in the learned models. The FoW functional is essentially a model for *anticipating* interesting change, which has been empirically shown to be useful in the context of the explore–exploit dilemma.

In the second case study we consider the problem of monitoring a massive scale spatiotemporally evolving stochastic process with a set of Unattended Ground Sensors (UGSs) with limited communication range. A UAS is utilized to ferry data from these sensors, and the problem of interest is to develop a predictive model of the accumulation of high value events so that the UAS can prioritize which sensors to visit in any given flight sortie. This problem can be viewed in the N-armed bandit framework, for which an optimal solution can be found through a linear program. The solution approach involves a more expressive model of the FoW functional, realized using a Cox-Gaussian process prior on the expected arrival rate of high-value events. The Exploitation by Informed Exploration between Isolated Operatives (EIEIO) approach is discussed which provides the mathematical specification of the model and associated learning algorithms (Axelrod and Chowdhary 2015). The

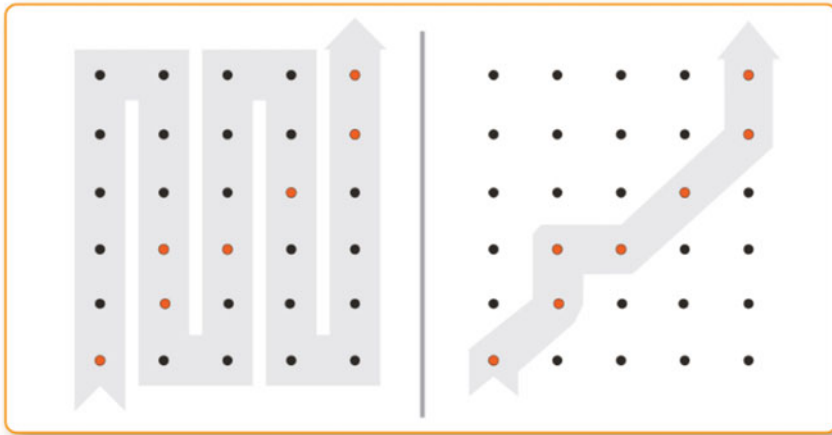


Fig. 2.1 The *arrows* in the *left* and *right* panel are sequential and informed search policies, respectively. The orange locations (*dots*) are information-rich relative to the surrounding locations. The aim for EIEIO is to learn the informed search policy

EIEIO algorithm efficiently learns to *anticipate* interesting change in a nonstationary environment, which yields a Bayes-optimal exploration strategy for nonstationary environments and provides a foundation for resolving the explore–exploit dilemma in nonstationary environments as shown in Fig. 2.1.

2.1.1 Decision Making and Control under Uncertainty in Nonstationary Environments

Markov Decision Processes (MDPs) have been studied for solving sequential-decision making problems over an array of uncertain domains, including navigation, complex games such as Chess or Go, nonlinear dynamical systems, and scheduling and resource allocation. Reinforcement Learning (RL) is concerned with solving MDPs when the underlying reward and transition models are unknown. A detailed review of state-of-the-art in MDPs and RL is provided in a recent survey (Geramifard et al. 2013). Most of the existing work in MDPs and RL focuses on the case where the reward and the transition models are non time varying. Here, we provide a brief summary of the RL/MDP problem for the time-varying (nonstationary) RL case (Yu and Mannor 2009; Abdoos et al. 2011).

A nonstationary MDP (N-MDP) is a tuple: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}(t), \mathcal{R}(t), \gamma, t \rangle$, with state space $\mathcal{S}$, action set $\mathcal{A}$, transition function $\mathcal{T} = p(s_{t+1}|s_t, a_t)$, reward function $\mathcal{R}(s, a) \mapsto \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. Note here that unlike traditional MDP formulations, the transition and reward models changes with time

indexing variable t . A deterministic nonstationary *policy* $\pi : \mathcal{S} \rightarrow \text{Pr}[\mathcal{A}]$ maps states to actions. Together with the initial state s_0 , a policy forms a *trajectory* $\zeta = \{[s_0, a_0, r_0], [s_1, a_1, r_1], \dots\}$ where $a_t = \pi(s_t)$, and both r_t and s_t are sampled from the reward and transition functions. The main difference here is that the Markov process trajectory ζ has a nonstationary distribution since both the transition model and the reward model are changing with time.

The state-action *value function* or *Q-function* of each state-action pair under policy π is defined as $Q^\pi(s, a, t) = \mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t, t)$ which is the expected sum of temporally-discounted rewards obtained starting from state s , taking action a and following π thereafter. A class of N-MDPs can be represented as Partially-Observable Markov Decision Processes (POMDPs) (Choi et al. 2001).

Reinforcement learning is concerned with finding optimal policies π even when the reward model $\mathcal{R}(t)$ or the transition model $\mathcal{T}(t)$ are unknown. A model-based RL (MBRL) agent simultaneously learns a predictive model of $\mathcal{T}(t)$, $\mathcal{R}(t)$ and utilizes it to compute the optimal policy. Because MBRL allows for prior knowledge to be naturally incorporated and leveraged for directed exploration (Thrun 1992; Wiering 1999; Boone 1997; Hester and Stone 2013; Ross and Pineau 2012; Vlassis et al. 2012). Whereas, a model-free RL agent directly utilizes state-action-reward tuples to determine the optimal policy, or equivalently, the optimal Q function. Several algorithms for solving RL have been surveyed in Geramifard et al. (2013).

2.2 Case Study 1: Model-Based Reinforcement Learning for Nonstationary Environments

This case study is concerned with the development of human-aware UAS path planning algorithms that minimize the likelihood of UAS flying over areas with high human density. Given a stationary (non-time varying) reward model encapsulating the likelihood of being seen by humans as function of the expected human density, the problem can be posed in the RL framework. Hence, if an a priori known and stationary reward model $\mathcal{R}(t)$ is available, then several existing path planning algorithms, including sample-based approaches (Karaman et al. 2011) or Markov Decision Process (MDP) (Thrun et al. 2005) based approaches can be used. Even if such a model is built from historic datasets, it would be difficult to ensure that this model is continually updated to account for nonstationary environments. Therefore, rather than relying solely on historic datasets, it is beneficial to maintain a real-time updated model of the population density. However, the main challenge is that human population densities change as a function of time of the day, the time of the year, or other seasonal considerations.

In this section, we present a nonstationary Markov Decision Process (MDP) formulation and model-based reinforcement learning approach as a solution to the problem. The resultant algorithm learns and maintains a separate model for each distinguishable distribution using Gaussian Process Non-Bayesian Clustering based

Model-Based Reinforcement Learning (GP-NBC-MBRL). The main benefit of using the Gaussian Process (GP) Bayesian Nonparametric (BNP) is that the model grows with the data, and little prior domain knowledge needs to be assumed. A non-Bayesian hypothesis testing based algorithm is used to cluster and classify GP cost models in real-time, and a model-based policy iteration algorithm (Lagoudakis and Parr 2003) is used to solve the MDP associated with each reward model.

The GP-NBC-MBRL leverages a variance-based exploration-exploitation strategy, which is driven by a *Fog-of-War* functional. The addition of the Fog-of-War functional encourages the UAS to explore regions which it has not explored *recently* by diluting its predictive confidence. This addition is crucial to ensure that the agent is able to detect changes in the environment. The integrated solution architecture that follows is quite general, and can be applied to other nonstationary planning and control problems, such as path planning in presence of nonstationary obstacles. Furthermore, it is the first GP based BNP nonstationary MDP solution architecture that explicitly handles the trade-off between exploration and exploitation in dynamically evolving environments.

2.2.1 Gaussian Process Regression and Clustering

A GP is defined as a collection of random variables such that every finite subset is jointly Gaussian and where every draw from a GP is a function. The joint Gaussian condition means that GPs are completely characterized by their second order statistics (Rasmussen and Williams 2005); i.e. the function estimate is defined using a mean function $\hat{m}$ and a covariance function $\hat{\Sigma}$. When Δ follows a Gaussian process model, then

$$\Delta(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \quad (2.1)$$

where $m(\cdot)$ is the mean function, and $k(\cdot, \cdot)$ is a real-valued, positive definite covariance kernel function (Scholkopf and Mullert 1999). For the sake of clarity of exposition, we will assume that $\Delta(z) \in \mathbb{R}$; the extension to the multidimensional case is straightforward.

GP Regression

A GP regression (GPR) is completely characterized by its second order statistics (Rasmussen and Williams 2005), and the mean is assumed to lie in the class of functions

$$\mathcal{G} = \left\{ g(\cdot) \in \mathbb{R}^{\mathcal{X}} \mid g(\cdot) = \sum_{i=1}^{\infty} \alpha_i k(z_i, \cdot) \right\}, \quad (2.2)$$

where $\mathcal{X} = \mathbb{R}^n$, $\alpha_i \in \mathbb{R}$, $z_i \in \mathcal{X}$. This assumption imposes a smoothness prior on the mean and makes the problem amenable to analysis though the representer theorem (Scholkopf et al. 2001).

The posterior (sometimes called the predictive) distribution, obtained by conditioning the joint Gaussian prior distribution over the observation z_{t+1} , is computed by

$$p(y_{\tau+1}|Z_\tau, y_\tau, z_{\tau+1}) \sim \mathcal{N}(\hat{m}_{\tau+1}, \hat{\Sigma}_{\tau+1}), \quad (2.3)$$

where

$$\hat{m}_{\tau+1} = \beta_{\tau+1}^T k_{z_{\tau+1}} \quad (2.4)$$

$$\hat{\Sigma}_{\tau+1} = k_{\tau+1}^* - k_{z_{\tau+1}}^T C_\tau k_{z_{\tau+1}} \quad (2.5)$$

are the updated mean and covariance estimates, respectively, and where $C_\tau := (K(Z_\tau, Z_\tau) + \omega^2 I)^{-1}$, $\beta_{\tau+1} := C_\tau y_\tau$, and $Z_\tau = \{z_1, \dots, z_\tau\}$ is a set of state measurements.

Since both Z_τ and y_τ grow with data, computing the inverse becomes computationally intractable over time. Therefore, to adapt GPs for the online setting, we use the efficient and recursive scheme introduced in Csató and Oppé (2002). This is known as the kernel linear independence test, and is computed by

$$\gamma_{\tau+1} = \left\| \sum_{i=1}^{\tau} \alpha_i \psi(z_i) - \psi(z_{\tau+1}) \right\|_{\mathcal{H}}^2. \quad (2.6)$$

The scalar $\gamma_{\tau+1}$ is the length of the residual of $\psi(z_{\tau+1})$ projected onto the subspace $\mathcal{F}_{Z_\tau}$. When $\gamma_{\tau+1}$ is larger than a specified threshold, then a new data point should be added to the data set. The coefficient vector α minimizing (2.6) is given by $\alpha_\tau = K_{Z_\tau}^{-1} k_{z_{\tau+1}}$, meaning that

$$\gamma_{\tau+1} = k_{\tau+1}^* - k_{z_{\tau+1}}^T \alpha_\tau. \quad (2.7)$$

This restricted set of selected elements, called the *basis vector set*, is denoted by $\mathcal{BV}$. When incorporating a new data point into the GP model, the inverse kernel matrix can be recomputed with a rank-1 update.

GP Clustering

The inference method in the previous section assumes that the data arises from a single model. In many real-world applications however, this is not a valid assumption. In particular, an algorithm that can determine that the current model does not match the source of data can be very useful, especially in the context of nonstationary reward inference. Grande et al. Grande (2014) presented such an algorithm in context of regression and clustering. Here we use it for solving nonstationary MDPs. The overall algorithm is described in Algorithm 2.1; see Grande (2014) for more details. For a GP, the log likelihood of a subset of points y can be evaluated as

Algorithm 2.1 GP Clustering

Input: Initial data (X, Y) , lps size l , model deviation η
 Initialize GP Model 1 from (X, Y) .
 Initialize set of least probable points $S = \emptyset$.
while new data is available **do**
 Denote the current model by M_c .
 If data is unlikely with respect to M_c , include it in S .
 if $|S| = l$ **then**
 for each model M_i **do**
 Calculate log-likelihood of data points S using having been generated from current model M_i (2.8) $\log(S|M_i)$, and find highest likelihood model M_h , making M_h current model.
 Create new GP M_S from S .
 if $\frac{1}{l}(\log(S|M_S) - \log(S|M_c)) > \eta$ **then**
 Add M_S as a new model.
 end if
 end for
 end if
end while

$$\log P(y | x, M) = -\frac{1}{2}(y - \mu(x))^T \Sigma_{xx}(y - \mu(x)) - \log |\Sigma_{xx}|^{1/2} + C, \quad (2.8)$$

where $\mu(x) = K(X, x)^T (K(X, X) + \omega_n^2 I)^{-1} Y$ is the mean prediction of the model M and $\Sigma_{xx} = K(x, x) + \omega_n^2 I - K(X, x)^T (K(X, X) + \omega_n^2 I)^{-1} K(X, x)$ is the conditional variance plus the measurement noise. The log-likelihood contains two terms which account for the deviation of points from the mean, $\frac{1}{2}(y - \mu(x))^T \Sigma_{xx}(y - \mu(x))$, as well as the relative certainty in the prediction of the mean at those points $\log |\Sigma_{xx}|^{1/2}$. Our algorithm, at all times, maintains a set of points S which are considered unlikely to have arisen from the current GP model M_c . The set S is used to create a new GP M_S , which is tested against the existing models M_i using a non-Bayesian hypothesis test to determine whether the new model M_S merits instantiation as a new model. This test is defined as

$$\frac{P(y | M_i)}{P(y | M_j)} \underset{\hat{M}_j}{\overset{\hat{M}_i}{\geq}} \eta \quad (2.9)$$

where $\eta = (1 - p)/p$, and $p = P(M_i)$. If the quantity on the left hand side is greater than η , then the hypothesis M_i (i.e. that the data y is better represented by M_i) is chosen, and vice versa.

2.2.2 GP-NBC-MBRL Solution to Nonstationary MDPs

The above ideas are leveraged to create an algorithm for nonstationary MDPs with unknown reward models. Based on the current model of the reward, a decision is made to either (a) explore the state space to gather more information for the reward, or b) exploit the model by solving the MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \hat{r}_{t_i})$. While step (b) is clear, the question arises on *how* to efficiently explore $\mathcal{S}$ so as to minimize time spent following a potentially suboptimal policy.

GP Exploration

Recall that in GP inference, the predictive variance is an indication of the GP's confidence in its estimate at a given location. Therefore, in areas that have been unexplored, the predictive variance is high and vice versa. In the example, a GP is initialized over a domain $D \subset \mathbb{R}^2$, and an agent explores the space, using samples to update the GP and reduce the predictive covariance from t_1 to t_{20} . The *exploration reward* $\hat{r}_{e_{t_i}}$ at time t_i is then defined as

$$\hat{r}_{e_{t_i}}(x) = \hat{\Sigma}_{t_i}(x) + \Upsilon_{t_i}(x), \quad (2.10)$$

where $\hat{\Sigma}_{t_i}(x)$ is the covariance of the GP over $\mathcal{S}$, and Υ_{t_i} is a functional over $\mathcal{H}$ which we refer to as the *FoW* functional. The latter increases the predictive variance as a function of time, so as to induce a tunable forgetting factor into (2.10). This exploration reward is used to create a new *exploration MDP* $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \hat{r}_{e_{t_i}})$, which can be solved in order to explore $\mathcal{S}$. With this intuition in place, we have a natural rule to make a decision on when to explore, by computing the quantity

$$s_e(t_i) = \kappa - \frac{1}{\text{volume}(D)} \int_D \hat{r}_{e_{t_i}}, \quad (2.11)$$

where κ is the largest value of the variance (1 for the Gaussian kernel) and $D \subset \mathcal{S}$. The quantity s_e is a measure of the space explored at an instant t_i . If this is above a certain threshold, the agent should only exploit its knowledge of $\mathcal{S}$ instead of exploring.

Putting it all together, we get the algorithm shown in Algorithm 2.2.

2.2.3 Example Experiment

An example motivated by human-aware UAS deployment is considered, where human population densities are reflected as a cost in the reward models and the human population densities change with time. Unknown to the agent, the human population densities in time are captured by in four distinct reward models where the transition between models may be randomized. The goal of this example is to

Algorithm 2.2 Nonstationary MDP Solver

Initialize: Initial data (X, Y) , lps size l , model deviation η , GP parameters (σ, ω_n^2) , space exploration threshold φ .

while new data (x_i, y_i) is available **do**

Update GP cluster using Algorithm 2.1 (Grande 2014).

Compute exploration reward $\hat{r}_{e_{t_i}}$ using (2.10).

Compute space explored s_e using (2.11).

if $s_e > \varphi$ **then**

Solve exploitation MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \hat{r}_{t_i})$ to get path.

else

Solve exploration MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \hat{r}_{e_{t_i}})$ to get path.

end if

end while

demonstrate that the UAS can learn an optimal policy for each model, as well as identify when the model changes, in an online fashion.

For this example, the human population density in a 50×50 gridworld are considered. The human population densities switch periodically every 200 runs for the first 4 times for each model, and then randomly. The reward samples are stochastic draws from the current human population intensity, so the reward changes with a change in the model. Both the GP Clustering (GPC)-based planner and the single-GP Regression (GPR)-based planner process each reward and state location to build a GP based generative model of the reward. A highly efficient policy-iteration based planner (cha 2012) may be used to compute the (nearly) optimal policy at the beginning of each run using the learned reward model.

Experimental Results

Figure 2.2 presents the average reward accumulated and its variance by two planners for each of the four models, one planner which does not use clustering (GPC) and the other planner that does (GPR). The horizontal axis indicates the number of times each model is active and the vertical axis indicates the cumulative reward the agent has accumulated over the time the underlying model was active. Notice that the first model the algorithms are initialized, in this case model 1, that both algorithms perform very similarly. One reason for this is because GPR tends to learn model 1 quickly as this is the model it was initialized in. However the GPC outperforms GPR in subsequent models, in this case models 2, 3 and 4. In fact, the GPC based planner should find the near optimal policy.

The performance of the GPC algorithm is shown in Fig. 2.3. Note that GPC quickly identifies whether the underlying model is similar to a one it has learned before and quickly learns new reward models when the model is not recognized. A key advantage of GPC is that the UAS does not have to spend a lot of time exploring. The net effect is that the UAS obtains consistently high mission scores, and a long-term improvement in performance can be seen.

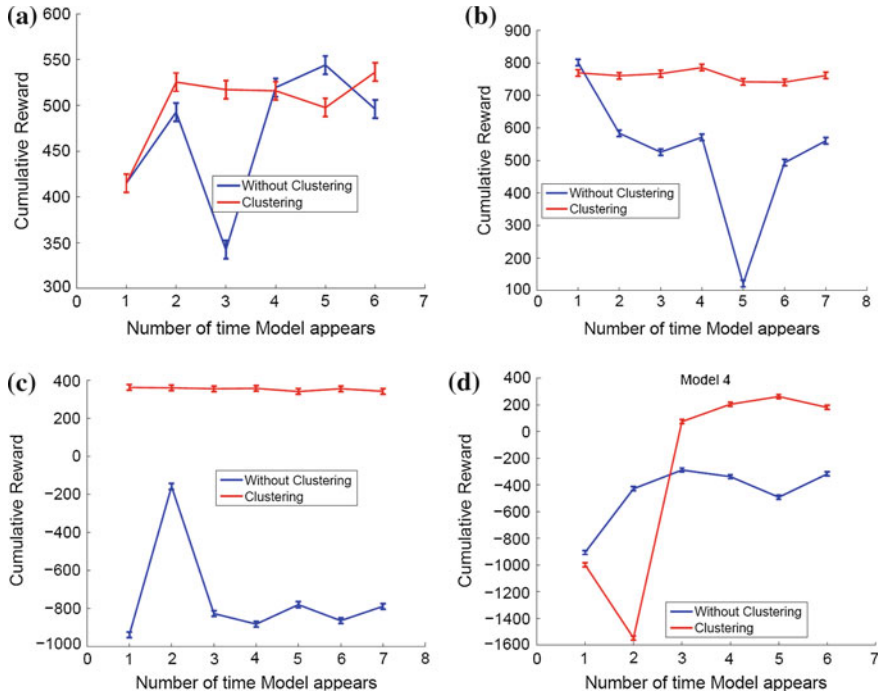
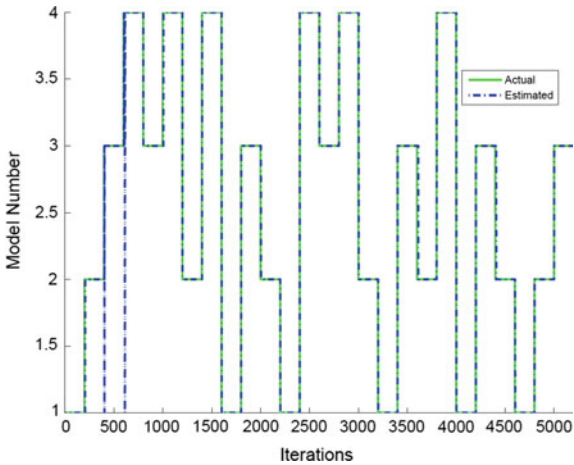


Fig. 2.2 Comparison between Total Rewards for each model. **a** Model 1. **b** Model 2. **c** Model 3. **d** Model 4

Fig. 2.3 Plot indicating the actual model being tracked by the estimated model. At the 200th and 400th run new models are introduced, the algorithm quickly detects them after a brief misclassification. Note that the algorithm clusters the underlying reward model quickly and reliably afterwards



2.2.4 Summary of Case Study 1

We presented a flexible and adaptive architecture to plan UAS paths to minimize the likelihood of the UAS being seen by humans. Our approach handles the potential shift in human population densities during the day, season, or time of the year by formulating the planning problem as a nonstationary MDP with unknown and switching reward models, and provides a model-based reinforcement learning solution to the problem. In our architecture, the different underlying reward models generated by the likelihood of being seen by humans are learned using an efficient Gaussian Process clustering algorithm. Sufficient exploration to make clustering decisions was induced through a novel *FoW* factor that encourages re-visiting of areas not recently visited. The learned reward models are then used in conjunction with a policy iteration algorithm to solve the nonstationary MDP. The proposed architecture is validated in a long-duration experiment with over 5500 simulated and real UAS path planning instances. The results show that the ability of the GP Clustering based planner to learn and recognize previously seen population densities allows our architecture to minimize unnecessary exploration and improve performance over the long term over a traditional GP regression based approach. Although developed in context of human aware path planning, our architecture is quite general, and it can be extended to solving other nonstationary MDPs as well.

2.3 Case Study 2: Monitoring Spatiotemporally Evolving Processes using Unattended Ground Sensors and Data-Ferrying UAS

Exploration of the domain is required by any reinforcement learning algorithm to ensure that it obtains sufficient information to compute the optimal policy (Busoniu et al. 2010). The exploration strategy in GP-NBC-MBRL is to compute a policy that guides the agents to areas of the domain where it has the least confidence in the model. However, as discussed in Sect. 2.2, relying simply on the GP variance is not enough, because the GP variance updates in closed-form GP algorithms (Csató and Oppel 2002; Rasmussen and Williams 2005) do not take into account the non-stationarity in the domain. The *FoW* forgetting introduced in (2.10) adds another metric on learned model confidence by ensuring that the agent revisits parts of the domain that it has not recently visited. Yet, exploration is costly, because this is when the agent is likely to accumulate negative reward. Therefore, the optimal long-term strategy should be to minimize exploration and maximize exploitation by learning the available Value-of-Information (VoI) and identifying similarities in the underlying models.

List of symbols and notations	
τ_i^n	Timestamp of n^{th} UAS visit to node i
t	Time $t \in [\tau_i^n, \tau_i^{n+1}]$
S	Set of Unattended Ground Sensors (UGS)
$\eta(\cdot)$	Reachable Subset of UGS
K	Number of UGS in Distributed Network
κ	Endurance Constraint $\kappa \leq K$
i	Index of UGS node
V_i	The Value-of-Information (VoI) of UGS i
V_i^+	The VoI at the Next Time Instance
$f(\cdot)$	Total VoI collected in a flight sortie
$\lambda_i(\cdot)$	Positively-Valued Random Variable
$\bar{\lambda}_i(\cdot)$	Mean of Poisson process (Pois)
$m_i(\cdot)$	Mean of Gaussian Process (GP)
$k_i(\cdot, \cdot)$	Covariance Kernel of GP
α	Gamma Distribution Shape Parameter
β	Gamma Distribution Rate Parameter
M	Number of Draws from Distribution
Δt_i	Time Between Visits $\Delta t := t - \tau_i^n$
σ^2	Variance of Gaussian Distribution and GP
$V_{GP,i}(\cdot)$	Draw from GP
$\bar{\lambda}_{c,i}(\cdot)$	Data-Based FoW Term for UGS i
$\lambda_{r,i}(\cdot)$	Randomly Seeded Poisson Parameter
ε	Gaussian White Noise

2.3.1 Connecting the FoW Functional to Data

The FoW functional used in GP-NBC-MBRL uniformly dilutes the predictive covariance of the GP. However, uniformly diluting the covariance can be conservative, especially if arrival of new events is spatiotemporally correlated. In this case, a uniformly diluted covariance may not reflect the expected information accumulation in an evolving environment, which may have states evolve at distinct non-constant rates. Also, no data-driven way of inferring the rate at which the predictive covariance should be diluted has been presented. Therefore, we turn our attention to Exploitation by Informed Exploration between Isolated Operatives (EIEIO), which connects the FoW functional to data-based observations by learning a generative model for the information entropy in a nonstationary environment.

The EIEIO algorithm in the following case study is applied to a restless bandit problem in the context of data-ferrying in a nonstationary environment, where Unattended Ground Sensors (UGS) with the highest informatic value are prioritized for each sortie. The data-ferrying agent is randomly initialized, and converges to a stochastic Gittins index policy as the agent learns to *anticipate* the informatic value of each UGS. Simulation results show EIEIO applied on simulated and real-world data sets.

2.3.2 Problem Definition

The general problem we are interested in is distributed inference and monitoring over a spatio-temporally varying measure y that changes with spatial variable $x \in \mathbb{R}^2$ and temporal variable $t \in \mathbb{T}$. For simplicity, we include a set of independent resource constrained unattended ground sensors (UGS) S , indexed by the variable i , which provide measurements of y at various locations $x_i \in \mathbb{R}^2$. The total number of sensors in the network is denoted by K . The measurements at each of these locations is denoted by the random variable y_i , which generates a temporally evolving stochastic process Y_t across all of the nodes. The stochastic process is time-varying, however, its rate of change need not be the same across all nodes. Consequently, not all nodes have new information at all times. We assume that the Value-of-Information (VoI), V_i of a node can be captured by an information-theoretic metric such as Kullback–Leibler (KL) divergence or Renyi divergence (Mu et al. 2014). Note that due to spatio-temporal variations, V_i is a temporally dependent random variable that takes positive values.

The UGS can leverage in-situ resources to operate over an extended duration of time, but it is assumed that they have a limited range of communication. Therefore, it is assumed that the ground sensors do not have sufficient power to communicate with a central hub, which leads to clusters of ground sensors which may be able to talk to each other but do not form a completely connected network. Instead, a data-ferrying agent, such as a UAV, needs to physically ferry the data between the ground sensors. However, the data-ferrying agent itself has a limited endurance, and can only visit a subset $\eta \subset S$ consisting of κ of the ground sensors in any given flight sortie. Hence, we may pose the problem as a restless bandit where η arms are played at each time step, and $\mathcal{Y}$ is a binary vector indicating which arms are being played. We leverage

Problem 2.1 Determine the subset η —the most informative set of UGS to visit in each flight sortie—by solving the following mathematical program

$$\begin{aligned} \eta(t) = \underset{\mathcal{Y}(t)}{\operatorname{argmin}} \quad & \sum_{i=1}^K \mathbb{E}(V_i(t)) - \mathbb{E}(V_i(t))\mathcal{Y}^{(i)}(t) \\ \text{s.t.} \quad & \|\mathcal{Y}(t)\|_1 = \kappa \end{aligned} \tag{2.12}$$

It is in general difficult to determine η because the expected VoI at each node is not known. Furthermore, without visiting the nodes it is not possible to glean what the VoI at each node could be. A proactive planning strategy for this problem is to build a model on the expected accumulated VoI at each node, and utilize this model to plan *anticipated change* at each of the sensing locations. A model on the expected VoI must be positively valued, with the exception of simultaneously-collected samples, and must model the expected VoI with respect to the time since last visit to each UGS. A notable random process with strikingly similar constraints is the Poisson process. Furthermore, there’s a comprehensive body of literature which highlights advantageous information-theoretic properties of the Poisson process (Rubin 1974; Kontoyiannis et al. 2005; Harremoës 2001; Johnson 2007; Harremoës et al. 2007,

2003; Harremoës and Ruzankin 2004; Guo et al. 2008; Prabhakar and Gallager 2003; Bedekar and Azizoglu 1998; Coleman et al. 2008), and it has been shown that information entropy may itself be considered as a random variable (Leśniewicz 2014), so we treat observations of VoI as the likelihood of a Poisson process; i.e.

$$\mathbb{E}(V_i(t)) = \mathbb{E} \left(\int_{\tau_i^n}^t \sum_j^r s_j(\lambda_{j,i}(t)) dt \right), \quad (2.13)$$

where $\lambda_{j,i}(t)$ are positively valued random variables denoting the Poisson arrival and departure rates of an informative event type j at UGS i worth s_j bits per net informative arrival. Note that there are r types of informative events, which relates to the notion of embodied cognition, in that only the types of sensors used may affect what informative events can be observed, and r need not be known.

The model is designed to capture the effect of the arrival of an unlabeled set of informative events at a particular node on the available VoI. Every time the node is visited, that is when $i \in \eta$, the VoI of that node gets reset to zero, at all other times, the informative events are allowed to accumulate. Furthermore, the variable V_i is modeled as the likelihood of a class of Poisson processes. Although the Poisson process is a well-known discrete random process, the likelihood of the Poisson process may be a continuous quantity, thus we are permitted an elegant continuous model for the available VoI in a large set of UGS.

The model is also inspired by the queueing-theoretic application of Poisson process priors in determining the arrival rate of packets in communication networks (Karagiannis et al. 2004). In a Poisson process, the inter-arrival times A_n of new events are exponentially distributed with a rate parameter $\hat{\lambda} : P(A_n \leq t) = 1 - e^{-\hat{\lambda}t}$ (Frost and Melamed 1994). When $\hat{\lambda}$ is constant, the Poisson process is termed homogeneous. A homogeneous Poisson process can be used to model the situation when the VoI accumulation is expected to be constant across the nodes.

Poisson processes can accommodate much more general forms of $\hat{\lambda}_i$. In an inhomogeneous Poisson process, $\hat{\lambda}_i(t)$ is spatio-temporally varying with each location i and time t . A general model for spatio-temporally varying Poisson processes is the Cox process. In the Cox process, $\hat{\lambda}_i$ is drawn from a stochastic process. Recall that existing Cox process models require a priori output scaling or domain specification (Møller et al. 1998; Zhou et al. 2014; Adams et al. 2009; Gunter et al. 2014). Since an upper bound on λ or the number of UGS may not be known known a priori, we introduce a new Bayesian Nonparametric model termed *Cox-Gaussian Process (CGP)*, which models the accumulated VoI at a location V_i using a Gaussian process prior:

$$\begin{aligned} \int_{\tau_i^n}^t \sum_j^r s_j(\lambda_{j,i}(t) - \tilde{\lambda}_{j,i}(t)) dt &\sim \text{Pois}(V_i(t)) \\ V_i(t) &\sim GP_i(m_i(t), k_i(t, t')), \end{aligned} \quad (2.14)$$

where $Pois(\cdot)$ is the Poisson process and $GP_i(\cdot, \cdot)$ is a Gaussian process with the mean m_i and covariance kernel $k(\cdot, \cdot)$. Another key benefit of this model is that the GPs can evolve to accommodate a changing number and distribution of UGS in the sensor network.

2.3.3 Solution Methods

While the mathematical program in (2.12) is used for each of the following solution approaches, the method used to calculate (2.13) for the Poisson process methods in Sect. “Poisson Process Methods” is

$$\mathbb{E}(V_i(t)) = \mathbb{E}(V_{Pois,i}(t)), \quad (2.15)$$

where

$$\mathbb{E}(V_{Pois,i}(t)) = \int_{\tau_i^n}^t \hat{\lambda}_i(t) dt, \quad (2.16)$$

which, for homogeneous Poisson processes, reduces to

$$\mathbb{E}(V_{Pois,i}(t)) = \Delta t_i \hat{\lambda}_i(t), \quad (2.17)$$

where

$$\Delta t_i = t - \tau_i^n. \quad (2.18)$$

Note that Δt_i provides the context for the multi-play contextual bandit formulated in (2.12).

Poisson Process Methods

The following solution approaches assume that the generative VoI model may be described by a homogeneous Poisson process. This assumption, as well as Assumption 1, will be relaxed in Sect. “Poisson–Cox Gaussian Processes”. Note that the homogeneous Poisson process provides a Bayesian baseline for the Poisson–Cox Gaussian process (P-CGP) formulated in Sect. “Poisson–Cox Gaussian Processes”.

Assumption 1 The VoI can only increase between visits; i.e.

$$V_i(t) \geq V_i(\tau_i^n) \quad \forall t \in [\tau_i^n, \tau_i^{n+1}]. \quad (2.19)$$

Poisson Sampling

Since, each VoI sample corresponds to a sample of a Poisson process likelihood, we obtain a linear model of the available VoI using the homogeneous Poisson process. Since the Gamma distribution, $G(\cdot)$, is the conjugate prior of a Poisson distribution, the Gamma distribution is used as the prior distribution.

The resulting posterior distribution is

$$G(\hat{\lambda}_i(\tau_i^{n+1})|\alpha_i(\tau_i^{n+1}), \beta_i(\tau_i^{n+1})) \propto \text{Pois}\left(\frac{M_{i,n} V_i(\tau_i^{n+1})}{\Delta t_i}\right) G(\hat{\lambda}_i(\tau_i^n)|\alpha_i(\tau_i^n), \beta_i(\tau_i^n)), \quad (2.20)$$

where the posterior parameters are used to update the VoI accumulation rate $\hat{\lambda}_i$ as

$$\hat{\lambda}_i(\tau_i^{n+1}) = \frac{\alpha_i(\tau_i^n) + \frac{M_{i,n} V_i(\tau_i^{n+1})}{\Delta t_i}}{\beta_i(\tau_i^n) + M_{i,n}} \quad (2.21)$$

and the posterior parameters are to be reused as the prior for subsequent measurements at static agent x_i ; i.e.

$$\alpha_i(\tau_i^{n+1}) = \alpha_i(\tau_i^n) + \frac{M_{i,n} V_i(\tau_i^{n+1})}{\Delta t_i} \quad (2.22)$$

and

$$\beta_i(\tau_i^{n+1}) = \beta_i(\tau_i^n) + M_{i,n}, \quad (2.23)$$

where

$$M_{i,n} = 1 \quad \forall t. \quad (2.24)$$

Linear Poisson Sampling

The linear Poisson process treats a sample at time t as the result of $M_{i,n}(\cdot)$ independent and identically distributed (iid) draws from a homogeneous Poisson process. Here we use $M_{i,n}(\tau_i^{n+1}) = \Delta t_i$ to emulate a sensor that samples in iteration time, but other sampling paradigms may be accommodated. Hence, (2.21)–(2.23) simplify to

$$\hat{\lambda}_i(\tau_i^{n+1}) = \frac{\alpha_i(\tau_i^n) + V_i(\tau_i^{n+1})}{\beta_i(\tau_i^n) + \Delta t_i}, \quad (2.25)$$

$$\alpha_i(\tau_i^{n+1}) = \alpha_i(\tau_i^n) + V_i(\tau_i^{n+1}), \quad (2.26)$$

and

$$\beta_i(\tau_i^{n+1}) = \beta_i(\tau_i^n) + \Delta t_i, \quad (2.27)$$

respectively.

Poisson–Cox Gaussian Processes

We introduce two novel Cox Gaussian processes which use the Poisson sampling variants of the FoW functional (Allamaraju et al. 2014) and the sparse online Gaussian process (Csató and Oppel 2002). The FoW functional is typically used to artificially

increase the predictive covariance of a reinforcement learning agent such that the agent transitions from exploiting some nonstationary model to exploration, rather than assuming that the unobserved regions of the state space do not change. Similarly, the sparse online GP initially expects the VoI to be 0 until nearby samples are observed, so the FoW term is used to increase the expected VoI so that (2.12) causes the data-ferrying agent to visit each UGS until the GP is sufficiently confident in predicting the expected accumulation of VoI. Hence, the FoW functional is considered (Allamaraju et al. 2014). However, the FoW functional is not connected to data and is identically valued for each sensor location i .

We connect the FoW functional to data by using Poisson processes, thereby yielding a data-driven FoW term in a Bayesian nonparametric Poisson–Cox Gaussian process (P-CGP). As a result, we have two novel P-CGPs; one which uses the Poisson Sampling scheme, and another that uses the Linear Poisson sampling scheme.

The method to calculate (2.13) for the P-CGPs in Sect. “Poisson–Cox Gaussian Processes” is

$$\mathbb{E}(V_i(t)) = \sigma_{2,i}^2 \mathbb{E}(V_{GP,i}(t)) + (1 - \sigma_{2,i}^2) \mathbb{E}(V_{Pois,i}(t)), \quad (2.28)$$

where Eq. (2.13) is restated as

$$\mathbb{E}(V_{GP,i}(t)) \sim GP_i(m_i(\Delta t_i), k_i(\Delta t_i, \Delta t'_i)). \quad (2.29)$$

Poisson–Cox Gaussian Process Sampling

Drawing inspiration from the Poisson sampling method in Sect. “Poisson Sampling”, we replace the uniform FoW term from Allamaraju et al. (2014) with the output from (2.21). Thus the underlying heterogeneity of the UGS VoI model is leveraged, even when the predictive confidence of the GP-mean is low. A summary of the P-CGP is provided in Algorithm 2.3.

Algorithm 2.3 Poisson-CGP Sampling

Input: $t, \tau_i^n, \hat{\lambda}_{c,i}$
Output: $\mathbb{E}(V_i(t+1))$
for each $i \in S$ **do**
 assign $\Delta t_i := t - \tau_i^n$
 if $i \in \eta(t)$ **then**
 print $V_i(\Delta t_i)$
 return $\hat{\lambda}_{c,i}$ from (2.21)
 Reset Counter $\Delta t_i := 1$
 else
 Increment Counter $\Delta t_i := \Delta t_i + 1$
 end if
 Get GP Prediction of VoI; i.e. $V_{GP,i}(\Delta t_i)$
 Get Poisson or Linear Poisson Prediction of $\hat{\lambda}_i$; i.e. $\lambda_{c,i}$
 $\bar{\lambda}(t+1) := (\sigma^2)(\lambda_{c,i} \cdot \Delta t_i) + (1 - \sigma^2)V_{GP,i}(\Delta t_i)$
end for

Linear Poisson–Cox Gaussian Process Sampling

Similarly, the Linear Poisson sampling method in Sect. “Linear Poisson Sampling” is used to replace the uniform FoW term from Allamaraju et al. (2014) with the output from (2.25). Thus, not only is the underlying heterogeneity of the UGS VoI model is leveraged, even when the predictive confidence of the GP-mean is low, but also approximately homogeneous phenomena may be rapidly and accurately approximated.

2.3.4 Simulation Results

The EIEIO algorithm is applied in the context of distributed sensing applications that require data-ferrying such as the detection of carbon dioxide or temperature readings over a large operational area. We assume that the UAS is able to visit a subset of up to 6 UGS per episode; i.e. $\kappa = 6$ for the simulated data sets and the Intel Berkeley Research lab data set.

Baseline-Comparison Study of Poisson-Based Methods

A baseline-Comparison study is shown in Figs. 2.4 and 2.5. The baseline methods included are a random exploration method and a sequential exploration method. Table 2.1 provides a concise summary of the performance gain achieved by each sampling method, with respect to the performance of the sequential sampling method. Note that the random sampling method, on average, performs just as well as sequential sampling on the real-world data sets.

Validation on Simulated Data Sets

The mobile agent first explores the data available from a subset ($\kappa = 6$) of the 50 deployed UGS ($K = 50$), and the initialization parameters for each method is ($\hat{\alpha}_i = 10$, $\hat{\beta}_i = 1$). The parameters for the homogeneous Poisson process generative VoI model are again seeded randomly for the simulations. The underlying parameter evolution for the Cox process generative VoI simulation is defined as

$$\lambda_i(t) = \left| \frac{\lambda_{r,i}}{8} (\sin(0.5t\lambda_{r,i}) + 1) + \varepsilon(\mu, \sigma_1^2) \right|, \quad (2.30)$$

where $\lambda_i(t)$ is the true underlying Poisson parameter, $\lambda_{r,i}$ is seeded randomly for the simulation and $\varepsilon(\cdot, \cdot)$ is Gaussian white noise; i.e. $\varepsilon \sim N(\mu, \sigma_1^2)$, where $\mu = 0$ and $\sigma_1^2 = 10$.

The baseline-Comparison for the developed methods in Sects. “Poisson Process Methods” and “Poisson–Cox Gaussian Processes” for a homogeneous Poisson process generative VoI model is shown in Fig. 2.4a–c. The baseline-Comparison for the developed methods in Sects. “Poisson Process Methods” and “Poisson–Cox Gaussian Processes” for a generative Cox process VoI model is shown in Fig. 2.4b–

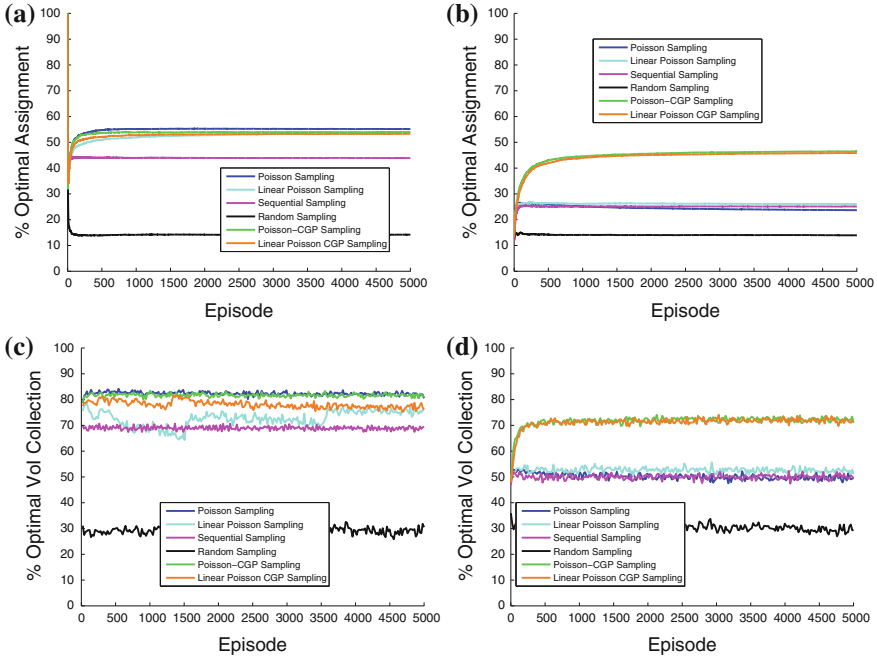


Fig. 2.4 **a** and **b** showcase the categorical accuracy of the Poisson-based methods in the presence of significant noise. Likewise, **c** and **d** showcase the efficacy of the Poisson-based methods in optimally reducing the available VoI. The percentage optimality refers to the performance of the unsupervised learning algorithm to that of a system with perfect situational awareness

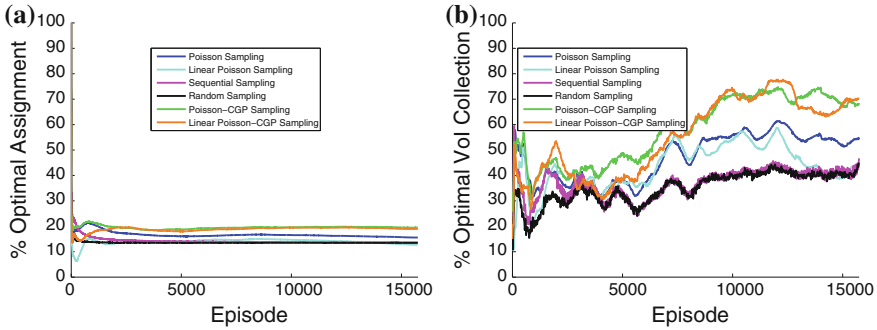


Fig. 2.5 The Intel Berkeley Research lab data set using (2.33). (Bodik et al. 2004). **a** Categorical Accuracy. **b** Percent of Maximum Score

d. The simulated gains in categorical accuracy (i.e. the average identification of the most valuable subset of UGS) and VoI collection relative to the sequential sampling method are shown below for all data sets.

Table 2.1 The percent performance gain of each Poisson-based method on each simulated and real-world data set is shown, with respect to the performance of the sequential sampling baseline

Sampling Method	Data Sets (% Gain in VoI)/(% Gain in Categorical Accuracy)		
	Homogeneous	Cox	Intel
Poisson	+15/ + 10	+0/ − 2	+10/ + 2
Linear Poisson	+8/ + 7	+4/ + 2	+0/ + 0
Poisson-CGP	+14/ + 8	+22/ + 20	+20/ + 5
Linear Poisson-CGP	+8/ + 7	+22/ + 20	+20/ + 5

Validation on the Intel Berkeley Data Set

As the results in Fig. 2.4 are the result of artificially generated VoI metrics, we then apply EIEIO on the Intel Berkeley Research Lab spatio-temporal temperature data set (Bodik et al. 2004), see Fig. 2.5. The goal of the experiment is to demonstrate that EIEIO works on real-world data sets. Although the data from the Intel Berkeley Research Lab is not Gaussian-distributed, we leverage the Central Limit Theorem to treat the observations at each UGS as belonging to a Gaussian-distributed likelihood; i.e. $y_i \sim N(\mu, \sigma_3^2)$. For simplicity, each UGS models local observations using the Gaussian distribution as the conjugate prior (Gelman et al. 2014). The Bayesian update is

$$\mu_{\hat{q}} = \frac{\frac{\sigma_3^2}{M} \mu_{\hat{p}} + \sigma_{\hat{p},i}^2 \bar{y}_i}{\frac{\sigma_3^2}{M} + \sigma_{\hat{p},i}^2}, \quad (2.31)$$

and

$$\sigma_{\hat{q}}^{-2} = \left(\frac{\sigma_3^2}{M} + \sigma_{\hat{q}}^2 \right)^{-1}, \quad (2.32)$$

where $\mu_{\hat{p}}$ is the prior mean, $\mu_{\hat{q}}$ is the posterior mean, $\sigma_{\hat{p},i}^2$ is the prior variance, and $\sigma_{\hat{q},i}^2$ is the posterior variance.

When the UAS visits each UGS, the local model of the UGS is received and the VoI is calculated using KL divergence (Hershey and Olsen 2007). The KL divergence, D_{KL} , for scalar normal distributions (i.e. $d = 1$) is

$$D_{KL}(\hat{q}||\hat{p}) = 0.5 \left[\log\left(\frac{\sigma_{\hat{p}}^2}{\sigma_{\hat{q}}^2}\right) + Tr[\sigma_{\hat{p}}^{-2} \sigma_{\hat{q}}^2] - d + (\mu_{\hat{q}} - \mu_{\hat{p}})^T \sigma_{\hat{p}}^{-2} (\mu_{\hat{q}} - \mu_{\hat{p}}) \right]. \quad (2.33)$$

2.3.5 Summary of Case Study 2

An endurance-constrained and model-based information-theoretic approach is formulated for a UAS to ferry data between distributed UGS. The EIEIO algorithm allows the UAS to *anticipate* the VoI available in the distributed UGS, even in non-stationary environments, hence the optimal subset of UGS, η , may be predicted for each data-ferrying sortie. The VoI model is realized in the form of two completely Bayesian P-CGPs which do not require any a priori knowledge to implement. The P-CGPs and Poisson process approaches are incorporated into EIEIO and compared on simulated and real-world data sets. Furthermore, an optimal sampling scheme is available when there is a known sample cost for each UGS.

Acknowledgments This work is sponsored by the Department of Energy Award Number DE-FE0012173, the Air Force Office of Scientific Research Award Number FA9550-14-1-0399, and the Air Force Office of Scientific Research Young Investigator Program Number FA9550-15-1-0146.

References

- Abdoos M, Mozayani N, Bazzan AL (2011) Traffic light control in non-stationary environments based on multi agent q-learning. In: 2011 14th international IEEE conference on, IEEE Intelligent Transportation Systems (ITSC), pp 1580–1585
- Adams RP, Murray I, MacKay DJ (2009) Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp 9–16
- Allamaraju R, Kingravi H, Axelrod A, Chowdhary G, Grande R, Crick C, Sheng W, How J (2014) Human aware path planning in urban environments with nonstationary mdps. In: IEEE international conference on robotics and automation, Hong Kong, China
- Axelrod A, Chowdhary G (2015) Adaptive algorithms for autonomous data-ferrying in nonstationary environments. In: AIAA Aerospace science and technology forum, Kissimmee, FL
- Bedekar AS, Azizoglu M (1998) The information-theoretic capacity of discrete-time queues. IEEE Trans Inf Theory 44(2):446–461
- Bodik P, Hong W, Guestrin C, Madden S, Paskin M, Thibaux R (2004) Intel lab data. Technical report
- Boone G (1997) Efficient reinforcement learning: Model-based acrobot control. In: Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, IEEE, vol 1, pp 229–234
- Busoniu L, Babuska R, Schutter BD, Ernst D (2010) Reinforcement learning and dynamic programming using function approximators, 1st edn. CRC Press
- Choi SP, Yeung DY, Zhang NL (2001) Hidden-mode markov decision processes for nonstationary sequential decision making. In: Sequence learning, Springer, pp 264–287
- Coleman TP, Kiyavash N, Subramanian VG (2008) The rate-distortion function of a poisson process with a queueing distortion measure. In: Data Compression Conference, DCC 2008, IEEE, pp 63–72
- Csató L, Opper M (2002) Sparse on-line Gaussian processes. Neural Comput 14(3):641–668
- Daw ND, O'Doherty JP, Dayan P, Seymour B, Dolan RJ (2006) Cortical substrates for exploratory decisions in humans. Nature 441(7095):876–879

- Frost VS, Melamed B (1994) Traffic modeling for telecommunications networks. *IEEE Commun Mag* 32(3):70–81
- Garivier A, Moulines E (2008) On upper-confidence bound policies for non-stationary bandit problems. arXiv preprint [arXiv:08053415](https://arxiv.org/abs/08053415)
- Gelman A, Carlin JB, Stern HS, Rubin DB (2014) Bayesian data analysis, vol 2. Taylor & Francis
- Geramifard A, Walsh TJ, Tellex S, Chowdhary G, Roy N, How JP (2013) A tutorial on linear function approximators for dynamic programming and reinforcement learning. *Foundations and Trends® in Machine Learning* 6(4): 375–451. doi:[10.1561/22000000042](https://doi.org/10.1561/22000000042)
- Grande RC (2014) Computationally efficient gaussian process changepoint detection and regression. Ph.D thesis, Massachusetts Institute of Technology
- Grande RC, Walsh TJ, How JP (2014) Sample efficient reinforcement learning with Gaussian processes. URL http://acl.mit.edu/papers/Grande14_ICML.pdf
- Granmo OC, Berg S (2010) Solving non-stationary bandit problems by random sampling from sibling kalman filters. In: *Trends in applied intelligent systems*, Springer, pp 199–208
- Grondman I, Busoniu L, Lopes GA, Babuska R (2012) A survey of actor–critic reinforcement learning: Standard and natural policy gradients. *IEEE Trans Syst, Man, and Cybern, Part C: Appl Rev* 42(6):1291–1307
- Gunter T, Lloyd C, Osborne MA, Roberts SJ (2014) Efficient bayesian nonparametric modelling of structured point processes. arXiv preprint [arXiv:14076949](https://arxiv.org/abs/14076949)
- Guo D, Shamai S, Verdú S (2008) Mutual information and conditional mean estimation in poisson channels. *IEEE Trans Inf Theory* 54(5):1837–1849
- Gur Y, Zeevi A, Besbes O (2014) Stochastic multi-armed-bandit problem with non-stationary rewards. In: *Advances in neural information processing systems*, pp 199–207
- Harremoës P (2001) Binomial and poisson distributions as maximum entropy distributions. *IEEE Trans Inf Theory* 47(5):2039–2041
- Harremoës P, Ruzankin P (2004) Rate of convergence to poisson law in terms of information divergence
- Harremoës P, Johnson O, Kontoyiannis I (2007) Thinning and the law of small numbers. *IEEE International Symposium on Information Theory, ISIT 2007*. IEEE, pp 1491–1495
- Harremoës P, Vignat C et al (2003) A nash equilibrium related to the poisson channel. *Commun Inf Syst* 3(3):183–190
- Hershey JR, Olsen PA (2007) Approximating the kullback leibler divergence between gaussian mixture models. In: *ICASSP (4)*, pp 317–320
- Hester T, Stone P (2013) Texlore: real-time sample-efficient reinforcement learning for robots. *Mach learning* 90(3):385–429
- Johnson O (2007) Log-concavity and the maximum entropy property of the poisson distribution. *Stoch Process Appl* 117(6):791–802
- Kakade S, Langford J, Kearns M (2003) Exploration in metric state spaces
- Karagiannis T, Molle M, Faloutsos M, Broido A (2004) A nonstationary poisson view of internet traffic. In: *INFOCOM 2004. Twenty-third annual joint conference of the IEEE computer and communications societies*. IEEE, vol 3, pp 1558–1569
- Karaman S, Walter M, Perez A, Frazzoli E, Teller S (2011) Anytime motion planning using the RRT*. In: *International conference on robotics and automation*. IEEE, pp 1478–1483
- Kolter JZ, Ng AY (2009) Near-bayesian exploration in polynomial time. In: *Proceedings of the 26th annual international conference on machine learning*. ACM, pp 513–520
- Kontoyiannis I, Harremoës P, Johnson O (2005) Entropy and the law of small numbers. *IEEE Trans Inf Theory* 51(2):466–472
- Koulouriotis DE, Xanthopoulos A (2008) Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Appl Math Comput* 196(2):913–922
- Lagoudakis MG, Parr R (2003) Least-squares policy iteration. *J Mach Learn Res* 4:1107–1149, URL <http://dl.acm.org/citation.cfm-id=945365.964290>
- Leśniewicz M (2014) Expected entropy as a measure and criterion of randomness of binary sequences. *Przegląd Elektrotechniczny* 90(1):42–46

- Markov decision processes (MDP) toolbox (2012). <http://www7.inra.fr/mia/T/MDPtoolbox/MDPtoolbox.html>
- Møller J, Syversveen AR, Waagepetersen RP (1998) Log gaussian cox processes. *Scand J Stat* 25(3):451–482
- Mu B, Chowdhary G, How J (2014) Efficient distributed sensing using adaptive censoring-based inference. *Automatica*
- Pazis J, Parr R (2013) Pac optimal exploration in continuous space markov decision processes
- Prabhakar B, Gallager R (2003) Entropy and the timing capacity of discrete queues. *IEEE Trans Inf Theory* 49(2):357–370
- Rasmussen C, Williams C (2005) Gaussian processes for machine learning (Adaptive Computation and Machine Learning). The MIT Press
- Reverdy P, Wilson RC, Holmes P, Leonard NE (2012) Towards optimization of a human-inspired heuristic for solving explore–exploit problems. In: CDC, pp 2820–2825
- Ross S, Pineau J (2012) Model-based bayesian reinforcement learning in large structured domains. arXiv preprint [arXiv:12063281](https://arxiv.org/abs/1206.3281)
- Rubin I (1974) Information rates and data-compression schemes for poisson processes. *IEEE Trans Inf Theory* 20(2):200–210
- Scholkopf B, Herbrich R, Smola A (2001) A generalized representer theorem. In: Helmbold D, Williamson B (eds) Computational learning theory., Lecture notes in computer science Springer, Berlin, pp 416–426 URL http://dx.doi.org/10.1007/3-540-44581-1_27
- Scholkopf B, Mullert KR (1999) Fisher discriminant analysis with kernels. Proceedings of the IEEE signal processing society workshop neural networks for signal processing IX. Madison, WI, USA, pp 23–25
- Sutton R, Barto A (1998) Reinforcement learning, an introduction. MIT Press, Cambridge
- Thrun S, Burgard W, Fox D, et al (2005) Probabilistic robotics, vol 1. MIT press Cambridge
- Thrun SB (1992) Efficient exploration in reinforcement learning. Carnegie-Mellon University, Technical report
- Tsitsiklis JN, Roy BV (1997) An analysis of temporal difference learning with function approximation. *IEEE Trans Autom Control* 42(5):674–690
- Vlassis N, Ghavamzadeh M, Mannor S, Poupart P (2012) Bayesian reinforcement learning. In: Reinforcement learning, Springer, pp 359–386
- Walsh TJ, Goschin S, Littman ML (2010) Integrating sample-based planning and model-based reinforcement learning. In: AAAI
- Watkins CJCH, Dayan P (1992) Q-learning. *J Mach Learn* 16:185–202
- Wiering MA (1999) Explorations in efficient reinforcement learning. Ph.D thesis, University of Amsterdam/IDSIA
- Wilson RC, Geana A, White JM, Ludvig EA, Cohen JD (2014) Humans use directed and random exploration to solve the explore–exploit dilemma. *J Exp Psychol: Gen* 143(6):2074
- Wilson SW, et al (1996) Explore/exploit strategies in autonomy. In: From animals to animats 4: Proceedings of the 4th international conference on simulation of adaptive behavior, pp 325–332
- Yu JY, Mannor S (2009) Online learning in markov decision processes with arbitrarily changing rewards and transitions. In: International conference on game theory for networks, GameNets’ 09. IEEE, pp 314–322
- Zhou Z, Matteson DS, Woodard DB, Henderson SG, Micheas AC (2014) A spatio-temporal point process model for ambulance demand. arXiv preprint [arXiv:14015547](https://arxiv.org/abs/1401.5547)

Handling Uncertainty and Networked Structure in Robot
Control

Busoniu, L.; Tamás, L. (Eds.)

2015, XXVIII, 388 p. 172 illus., 146 illus. in color. With
online files/update., Hardcover

ISBN: 978-3-319-26325-0