Advance Access publication on January 12, 2010
doi:10.1093/comjnl/bxp122

Prospects for Bandit Solutions in Sensor Management

NICOS G. PAVLIDIS[1,*], NIALL M. ADAMS[2], DAVID NICHOLSON[3] AND DAVID J. HAND[1,2]

[1]*Institute for Mathematical Sciences, Imperial College London, South Kensington Campus, London SW7 2AZ, UK*
[2]*Department of Mathematics, Imperial College London, South Kensington Campus, London SW7 2AZ, UK*
[3]*Advanced Technology Centre, BAE Systems, Sowerby Building, PO Box 5, Filton, Bristol BS34 7QW, UK*
[*]*Corresponding author: n.pavlidis@imperial.ac.uk*

Sensor management in information-rich and dynamic environments can be posed as a sequential action selection problem with side information. To study such problems we employ the dynamic multi-armed bandit with covariates framework. In this generalization of the multi-armed bandit, the expected rewards are time-varying linear functions of the covariate vector. The learning goal is to associate the covariate with the optimal action at each instance, essentially learning to partition the covariate space adaptively. Applications of sensor management are frequently in environments in which the precise nature of the dynamics is unknown. In such settings, the sensor manager tracks the evolving environment by observing only the covariates and the consequences of the selected actions. This creates difficulties not encountered in static problems, and changes the exploitation–exploration dilemma. We study the relationship between the different factors of the problem and provide interesting insights. The impact of the environment dynamics on the action selection problem is influenced by the covariate dimensionality. We present the surprising result that strategies that perform very little or no exploration perform surprisingly well in dynamic environments.

Keywords: multi-armed bandits with covariates; sensor management; action-selection; dynamic environment

Received 11 August 2009; revised 1 December 2009
Handling editor: Nick Jennings

1. INTRODUCTION

Sensor management refers to the theory and application of dynamic resource allocation in a diverse system of sensors and sensing modalities [1]. We are concerned with active sensor management problems in which the sequence of sensor actions is determined in an online and adaptive manner based on the information gathered from previous actions. Active sensor management can be viewed as a sequential decision-making problem in that at each time one action from a set of actions is selected, to satisfy certain objectives. The fundamental challenge that arises in this type of problem is that the consequences of each action are uncertain and must be learnt. Learning in this context happens through interaction with the *environment*, which consists of everything that is external to the sensor manager [2]. This learning process is heavily influenced by the sequence of action selections since only the consequences of the selected action are observed.

The multi-armed bandit (MAB) [3] is the simplest formulation of a sequential decision problem that captures these characteristics. In its most basic formulation, the MAB involves an agent that at each *play* selects one of K actions and receives a reward derived from the probability distribution associated with this action. The rewards from all the other actions are not revealed. The goal is to identify the action with the maximum expected reward within the minimum number of plays [2]. The MAB constitutes the minimal formalization of the trade-off between *exploration* (trying different actions to identify the best) and *exploitation* (choosing the action believed to be the best in order to maximize reward) that arises in all sequential decision problems. For this reason the MAB has been extensively studied [4–6] and employed in numerous diverse applications including clinical trials [7], adaptive routing [8], real-world data retrieval problems with redundant sources [9], economics [10] and sensor management [1]. In the following, we adopt the MAB terminology and refer to the consequences

of selecting an action as the reward from this action, and the goal of the sensor manager becomes the selection of the action with the highest expected reward. While the MAB minimalism admits tractability and insight, it misses details that are necessary for application to many realistic problems [11]. An important aspect, ignored in the MAB formulation, is that the agent can observe side information prior to each action selection. This information can be used to determine which action to select. Another aspect is that the consequences of different actions may change over time.

We address these issues in the context of sensor management applications. In the problems we consider, the expected reward of each action is a function of a set of covariates that are observed prior to each action selection. A simple example is a sensor used to monitor and assess the threat posed by a number of vehicles. Actions in this setting correspond to assigning the sensor to observe different vehicles, and the objective of the sensor manager is to select the one that presents the highest threat currently. Instead of assuming that the expected threat of each vehicle is constant it is more realistic to relate it to a number of covariates such as its location relative to the sensor and a number of locations of interest, the identity currently assigned to the vehicle and the route it has followed. A more complicated scenario involves a potentially large number of agents equipped with diverse sensing equipment deployed in a particular area. At each time one agent is capable of sensing and a central sensor manager decides which threat should be monitored. Some of the additional covariates that influence the reward from each action in this scenario are the type of the sensing agent, the capabilities of the sensor it is equipped with and its position relative to the threats. A detailed discussion of the sensor management problem and various alternative approaches is provided in [12]. In [13], sensor management in a defence application context is discussed.

Sequential decision problems in which the reward from an action is modified by covariates arise in a number of fields other than sensor management. Some indicative examples include matching advertisements to web-page content on the World Wide Web [11, 14], adaptive generation of multimedia messages [15] and video compression [16]. Indeed, settings in which no information is available are rare in practice [11].

Two important properties of all the aforementioned applications are first, that the consequences of each action are felt immediately, and second, the selected action does not affect the subsequent realizations of the covariates. Therefore, we are not dealing with the complete reinforcement learning problem [2]. In such problems, the objective is to learn which is the optimal action, i.e. the action that yields the maximum expected reward, for all possible realisations of the covariate vector. A rule that associates covariates to actions is called a *policy*, and learning the optimal policy is essentially learning how to partition covariate space. We refer to this type of problem as multi-armed bandits with covariates (MABC) [17]. Recently, a number of different formulations of this problem have been proposed, and algorithms have been developed for static versions of this problem [14, 17–19].

Sensor management is typically deployed in a dynamic environment, which is not typical of most work on MABC. In the present context, a dynamic environment is one in which the relationship between the covariates and the expected reward from each action changes over time. In most real life applications, it is hard to a priori specify the precise type and speed of the underlying process that governs change. Moreover, some problems may be characterized by periods of relatively small change, followed by periods in which the environment undergoes major change. It is therefore important to develop approaches that accommodate the possibility of time-variations without requiring the knowledge of the precise type or even the presence of dynamics.

In dynamic MABC problems, the relationship between the rewards and the covariates changes sufficiently to induce a time-varying optimal policy. Without knowing or imposing assumptions concerning the nature of the dynamics, the evolution of the environment must be tracked via the covariate and the rewards observed from the selected actions. This induces significant new problems which are not present in the static case. According to the formulation of the MABC, only the reward from the selected action is observed at each play. In the static case, this does not affect the estimation of the reward functions, in the sense that estimation accuracy is not affected by the timing of each observation. In the dynamic case, it creates a missing data problem which has a substantial impact on estimation. Similarly, the role of covariate dimensionality changes with the introduction of dynamics. Under these conditions it is hard to derive analytical results for the dynamic MABC. Therefore, in our analysis in Section 3 we resort to simulation.

The introduction of dynamics also affects the distinction between exploration and exploitation. In a static environment, there is relatively little information to be gained from exploitation because actions that yield the highest expected reward (*greedy actions*) are frequently selected and hence there is little uncertainty about them. In a dynamic environment, the selection of any action, even a greedy action, yields new information simply because reward functions change. Hence, there is an exploratory component in exploitation. A second consideration is related to the cost of exploration. In a static environment, the cost of an exploration step is the difference between the expected reward from the greedy action and the (probably) smaller reward from the exploratory action. In a dynamic environment, there is the additional cost of missing one observation from the evolution of a greedy action. This cost can have repercussions in subsequent plays as well.

The paper is organized as follows: in Section 2 the formulation of the MABC is presented; the next section is devoted to the presentation of the empirical methodology and results first on an artificial dynamic MABC problem and then on a sensor management problem related to monitoring and

assessing threats; the paper ends with concluding remarks in Section 4.

2. MABC WITH LINEAR REWARD FUNCTIONS

In our formulation of the MABC, we assume that a covariate vector is observed at each play, $t = 1, 2, \ldots, T$. The true reward function for each action, $\alpha_i = \alpha_1, \ldots, \alpha_N$, is linear with an additive noise term:

$$r_{\alpha_i}(x(t)) = \beta_{i,0} + \sum_{j=1}^{d} \beta_{i,j} x_j(t) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2). \quad (1)$$

It is often the case that the reward function of an action can be well approximated by a linear function [2, 14, 19]. The assumption of linearity also allows us to focus on the action selection problem without confounding it with the general function approximation problem. Even under this assumption, however, the estimation of time-varying reward functions is a challenging task, as will be shown in Section 3. More complicated scenarios can be constructed by allowing the distribution of the noise term in the reward functions, ϵ_i in Equation (1), to be skewed. In this work, however, we restrict our attention to the normal distribution.

A *policy* is a rule that associates covariates with actions, $\pi : \mathbb{R}^d \to \mathcal{A}$. The learning problem is to identify the optimal policy, i.e. to identify the action with the highest expected reward for all possible realizations of the covariate vector. A simple instance of the MABC with the expected reward from each action being a linear function of the covariates with three-arms and a one-dimensional covariate is illustrated in Fig. 1.

In this example, the optimal policy is to select action 1 when $x \geqslant 0$, and action 2 when $x < 0$. Nowhere is it optimal to select action 3. We call actions that are suboptimal for all x *globally suboptimal*. We also define the *degree of optimality* of an action as the probability that this action will be optimal. The degree of optimality of an action is therefore equal to the probability that the covariate vector lies in the region in which this action yields the highest expected reward. Because the covariate is one-dimensional in the present example, the optimal action changes at a point, which we call a *decision point*. In higher dimensions, regions characterized by different optimal action are separated by hyperplanes, called *decision surfaces*.

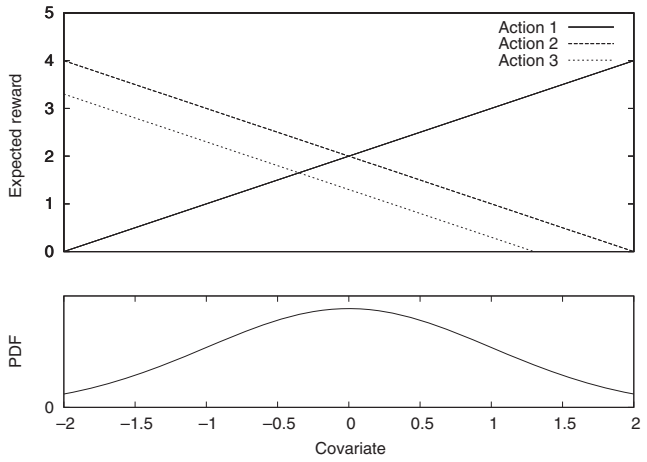

FIGURE 1. Expected reward functions of a three-armed MABC with a one-dimensional covariate.

We introduce dynamics in the MABC by having the coefficients of all the reward functions, $\beta_i = (\beta_{i,0}, \beta_{i,1}, \ldots, \beta_{i,d})^\top$, change at each play, while the distribution of the covariate is constant. We do not consider a time-varying covariate distribution because it plays no role in the transformation of a static sequential decision-making problem into a dynamic problem. The defining characteristic of a dynamic sequential decision-making problem is that the optimal policy changes over time. The optimal policy associates covariates with the best action. This association is not affected by the covariate distribution. This claim should not be misunderstood to imply that introducing a time-varying covariate distribution will not affect performance, or the difficulty of the learning task.

2.1. Estimation of time-varying reward functions

When a model of the dynamics is available, the sensor manager can employ a sequential learning algorithm to infer its parameters. For instance, if the coefficients of the reward functions evolve at each time-step according to a linear model, then its parameters can be estimated through the online expectation maximization algorithm [20]. In the dynamic MABC problem, an accurate model of the dynamics can improve not only the estimation accuracy of the reward function parameters, but also provide valuable information for the action selection process. For example, with an estimate of the variance of the reward from each action upper-confidence, bound type of strategies [4] can be deployed. However, applications in which the type of dynamics is unknown or even the existence of dynamics is uncertain are frequently encountered in sensor management. At present we focus on dynamic problems in which the sensor manager has no knowledge of the nature of the dynamics. In the MABC framework, it is very hard to construct an accurate model for the dynamics with no prior knowledge and within a finite number of plays. The major difficulty is due to the fact that there are typically many actions while only the response from the selected action is observed at any time. Furthermore, the sampling frequency of each action is variable due to action selection. We thus assume that the sensor manager is not attempting to explicitly model, or learn, the dynamics. Avoiding the explicit modelling of the dynamics has the advantage that it enables the application of the same methodology to problems in which the type or speed of dynamics can change.

In the absence of a model of the dynamics, the sensor manager has to track the evolution of the reward functions

only through the observed covariates and rewards. To this end, sophisticated approaches such as Gaussian processes [21] can be employed. The advantage of Gaussian processes is that they provide an estimate of the variance of each reward, but they are computationally demanding. Instead, we consider a simpler, but computationally efficient, adaptive linear filter, namely the Adaptive Recursive Least Squares (ARLS) algorithm [22, 23] to track the evolution of the reward functions. The RLS algorithm is one of the most popular adaptive filters [23, 24]. It handles time-varying systems by incorporating the concept of *forgetting*, according to which older data are gradually discarded in favour of more recent observations. The standard linear least squares (LLS) estimation of the coefficients of the linear relationship between the response, r, and the covariate vector, $x \in \mathbb{R}^d$, is given by [23]:

$$\hat{\beta}(t) = \left(X(t)^\top X(t)\right)^{-1} \left(X(t)^\top R(t)\right), \tag{2}$$

where $X(t) \in \mathbb{R}^{T\times(d+1)}$ is the data matrix storing in its kth row the covariate vector $x(k)^\top$, and $R(t) \in \mathbb{R}^T$ is the vector of desired responses. With the introduction of forgetting the time-average correlation matrix, $X(t)^\top X(t)$, becomes

$$\begin{aligned} X(t)^\top X(t) &= \sum_{i=1}^{t} \lambda^{t-i} x(i)^\top x(i) + \delta\lambda^t I, \\ &= \lambda X(t-1)^\top X(t-1) + x(t)x(t)^\top, \end{aligned} \tag{3}$$

where $\lambda \in (0, 1)$ is the *forgetting factor*, and $\delta\lambda^t I$ is a regularization term that ensures that the matrix is nonsingular at all stages of the computation. As λ tends to unity, past and present observations become equally weighted and for the special case that $\lambda = 1$ RLS becomes equivalent to LLS. In contrast, as λ tends to zero, recent data become more influential, and the impact of past data is progressively reduced. The appropriate choice of λ is critical for the accuracy of tracking a time-varying system. The classical RLS algorithm uses a constant forgetting factor. In the ARLS algorithm, λ is updated at each iteration using a stochastic gradient descent algorithm to minimize the squared a priori estimation error [22, 23, 25]:

$$\lambda(t) = \lambda(t-1) + \alpha\xi(t)\psi(t-1)^\top x(t)\Big|_{\lambda_-}^{\lambda_+}, \tag{4}$$

where $\xi(t) = r(t) - \hat{\beta}(t-1)x(t)$ is the a priori estimation error, $\psi(t)$ is the derivative of the estimated coefficient vector with respect to $\lambda(t-1)$, $\psi(t) = \partial\hat{\beta}(t)/\partial\lambda$ and α is the step-size. The bracket followed by λ_+ and λ_- indicates truncation. The upper level of truncation, λ_+, can be set close to unity. It is the lower level of truncation, λ_-, however, that plays a more crucial role, and this value needs to be determined empirically. In all the experiments the value of λ_- was 0.7.

The most computationally demanding step in the estimation of $\hat{\beta}(t)$ in Equation (2) is the inversion of $X(t)^\top X(t)$. This is particularly relevant in our MABC formulation because the coefficient vector of the selected action must be updated at each play. Employing the matrix inversion lemma [26] the inversion of the matrix $X(t)^\top X(t)$ at each new covariate-response pair can be avoided. Instead, the inverse $G(t) = (X(t)^\top X(t))^{-1}$ can be recursively computed from $G(t-1)$ reducing the computational cost from $\mathcal{O}(d^3)$ to $\mathcal{O}(d^2)$. The estimated coefficient vector, $\hat{\beta}(t)$, can also be incrementally updated [23].

ARLS is capable of tracking time-varying linear equations. It can handle different and variable speeds of change by adapting the degree of forgetting, without requiring the a priori determination of this crucial parameter [23]. Finally, the low computational cost renders it suitable for sequentially updating the estimated coefficients. These characteristics render ARLS a suitable adaptive filter for the purposes of the dynamic MABC problem.

3. EMPIRICAL METHODOLOGY AND RESULTS

In this section, we discuss the experimental results obtained for the dynamic MABC with linear reward functions. We first study an artificial dynamic MABC problem that we used to investigate the impact of different factors, like the covariate dimensionality and the speed of the dynamics. Next we present a sensor management problem related to monitoring and assessing threat.

Before discussing the two problems we provide the definitions of the two measures that will be used to assess the performance of different action selection strategies. The first performance measure that we employ is a transformation of the regret metric, which is the standard evaluation measure used in the MAB literature. Regret is defined as the difference between the maximum expected reward and the expected reward from the selected action, $\mathbb{E}[r_{\alpha(t)}(x(t))]$ [4, 19]:

$$T \max_{\alpha_i\in\mathcal{A}} \mathbb{E}[r_{\alpha_i}(x(t))] - \sum_{i=1}^{T} \mathbb{E}[r_{\alpha(t)}(x(t))].$$

The expectation is over the distribution of the reward given the covariate vector at play t. We apply a standard linear transformation to bound the range of the regret metric in $[0, 1]$, and refer to this transformation as the *normalized expected regret* measure:

$$\frac{1}{T}\sum_{i=1}^{T} \frac{\max_{\alpha_i\in\mathcal{A}} \mathbb{E}[r_{\alpha_i}(x(t))] - \mathbb{E}[r_{\alpha(t)}(x(t))]}{\max_{\alpha_i\in\mathcal{A}} \mathbb{E}[r_{\alpha_i}(x(t))] - \min_{\alpha_i\in\mathcal{A}} \mathbb{E}[r_{\alpha_i}(x(t))]}. \tag{5}$$

Normalized expected regret at each play is defined as the difference between the maximum expected reward and the expected reward from the selected action, $\mathbb{E}[r_{\alpha(t)}(x(t))]$, divided by the difference between the maximum and the minimum expected reward. The second measure we consider is the

proportion of best action [2], which is defined as:

$$\frac{1}{T}\sum_{i=1}^{T}\left\{\alpha(t) = \text{argmax}_{\alpha_i \in \mathcal{A}}\mathbb{E}[r_{\alpha_i}(x(t))]\right\}, \qquad (6)$$

where $\{\cdot\}$ is the indicator function. Compared with normalized expected regret this is a much harder evaluation criterion, because the maximum penalty is received for selecting any action other than the best. To receive the equivalent penalty by the normalized expected regret criterion, the action with the lowest reward must be selected.

3.1. Artificial dynamic environment

In this set of experiments, the covariate vector is obtained from a multivariate normal distribution, $x(t) \sim \mathcal{N}(\mu, \Sigma)$, with the elements of μ chosen uniformly in $[-5, 5]$. Without loss of generality, we consider diagonal covariance matrices Σ with each diagonal element in the range $[3d, 6d]$. The performance of any action selection strategy depends on the relationship between the covariance matrix Σ of the covariate vector and the variance of the noise term of the reward equations σ_i^2 in Equation (1). To quantify this relationship we use the *covariance to noise ratio*, CNR $= \|\Sigma\|_1/\sigma_i^2$, where $\|\Sigma\|_1$ is the 1-norm of Σ. The larger the value of CNR the more informative observations become about the regression line, and vice versa. To render comparable the results from experiments with different covariate dimensionality we fix the value of CNR.

The simplest artificial model for inducing time-varying coefficient vectors is the random walk, $\beta_{i,j}(t) = \beta_{i,j}(t-1) + \eta$, with $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$. For our present purposes the problem with this model is that the occurrence of large values of $|\eta|$ frequently causes one action to dominate the others, or conversely to become globally suboptimal, over long periods. As a consequence the decision-making problem frequently becomes trivial. An alternative model for the dynamics is one in which the coefficients are randomly perturbed around an equilibrium value, $\beta_{i,j}(t) = \beta_{i,j}^{\text{eq}} + \eta$. However, the decision-making problem associated with this type of change is static. The exponential smooth transition autoregressive (ESTAR) [27] model provides a compromise between the two aforementioned types of variation. Two interpretations of the ESTAR model are possible. On one hand, it can be thought of as a regime-switching model that allows for two regimes (random walk and mean reversion) with a smooth transition between them. On the other hand, the ESTAR model can be said to allow for a 'continuum' of regimes [28].

We employ a simple ESTAR formulation, in which the jth coefficient of the ith action at time t is determined by:

$$\beta_{i,j}(t) = \beta_{i,j}^{\text{eq}} + (\beta_{i,j}(t-1) - \beta_{i,j}^{\text{eq}}) \times \exp\left(-\gamma(\beta_{i,j}(t-1) - \beta_{i,j}^{\text{eq}})^2\right) + \eta, \qquad (7)$$

where $\beta_{i,j}^{\text{eq}}$ is the equilibrium value of $\beta_{i,j}$, which we set to $\beta_{i,j}(0)$, η is an independent and identically distributed random variable $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ and $\gamma \in (0, \infty)$ is called the smoothness parameter. The ESTAR model of Equation (7) implies a nonlinear and symmetric adjustment of $\beta_{i,j}(t)$ for deviations from its equilibrium value, $\beta_{i,j}^{\text{eq}}$. For large deviations the process becomes mean reverting and forces $\beta_{i,j}(t)$ towards $\beta_{i,j}^{\text{eq}}$, while for small deviations the process exhibits near random walk behaviour. Overall, although $\beta_{i,j}(t)$ is globally stationary, it can exhibit a high degree of persistence.

Before discussing the balance between exploitation–exploration in dynamic MABC problems, it is important to address two issues. The first is the capability of the ARLS filter to track the evolution of time-varying reward function in the presence of missing data. The second issue is related to the impact of the dimensionality of the covariate on the optimal policy.

3.1.1. Speed of change and missing data

By the formulation of the problem, at each play, t, the sensor manager observes only the response from the selected action. The rewards from all the other actions are not observed. In a static environment, this fact does not affect estimation. In the case of continuous dynamics, however, this is no longer true. As an example, consider a linear function whose coefficients follow a random walk. Observing the response from this function every second play is equivalent to observing at each play a random walk process with twice the variance. This issue is further complicated by the fact that the sampling frequency of each action is variable due to the action selection strategy, and ARLS is fundamentally not suited to variable sampling frequency.

To investigate the tracking ability of the ARLS filter, we monitor the values of the forgetting factor, $\lambda(t)$, with respect to the speed of change and the sampling frequency. As mentioned in Section 2.1 the ARLS filter adapts $\lambda(t)$ at each iteration using stochastic gradient descent on the squared a priori estimation error, Equation (4) [23]. We consider a time-varying linear equation whose coefficients evolve at each iteration according to the ESTAR model of Equation (7). The speed of change is determined by σ_η^2, the variance of the noise term η in Equation (7). At each iteration, the response from the linear equation is observed with probability p, where p determines the sampling frequency.

The results of this experiment for $\sigma_\eta^2 \in [0, 0.2]$, $p \in [0, 1]$ and $d = 4, 14$ are presented in Fig. 2. The left column of the figure depicts the mean value of $\lambda(t)$ over 5000 iterations, and the right column depicts the variance. Figure 2 shows that when the probability of observing the response is high, $\lambda(t)$ assumes values significantly lower than unity. In these cases, ARLS is tracking the evolution of the equation by assigning more weight to recent data compared with old data. As p declines and σ_η^2 increases, the average $\lambda(t)$ increases and in the limit that p tends to zero the average $\lambda(t)$ tends to unity. When $\lambda(t)$ tends to unity ARLS weights past and recent observations equally, performing an estimation equivalent to that of standard least

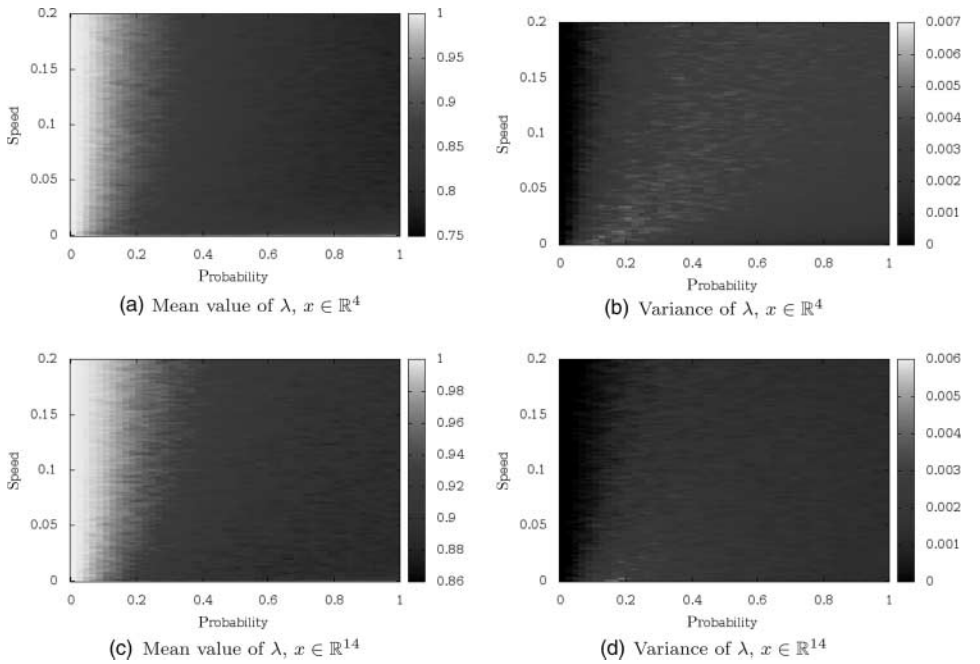

FIGURE 2. Mean and variance of λ over 5000 iterations with respect to σ_η^2 and the probability of observing the response at each iteration.

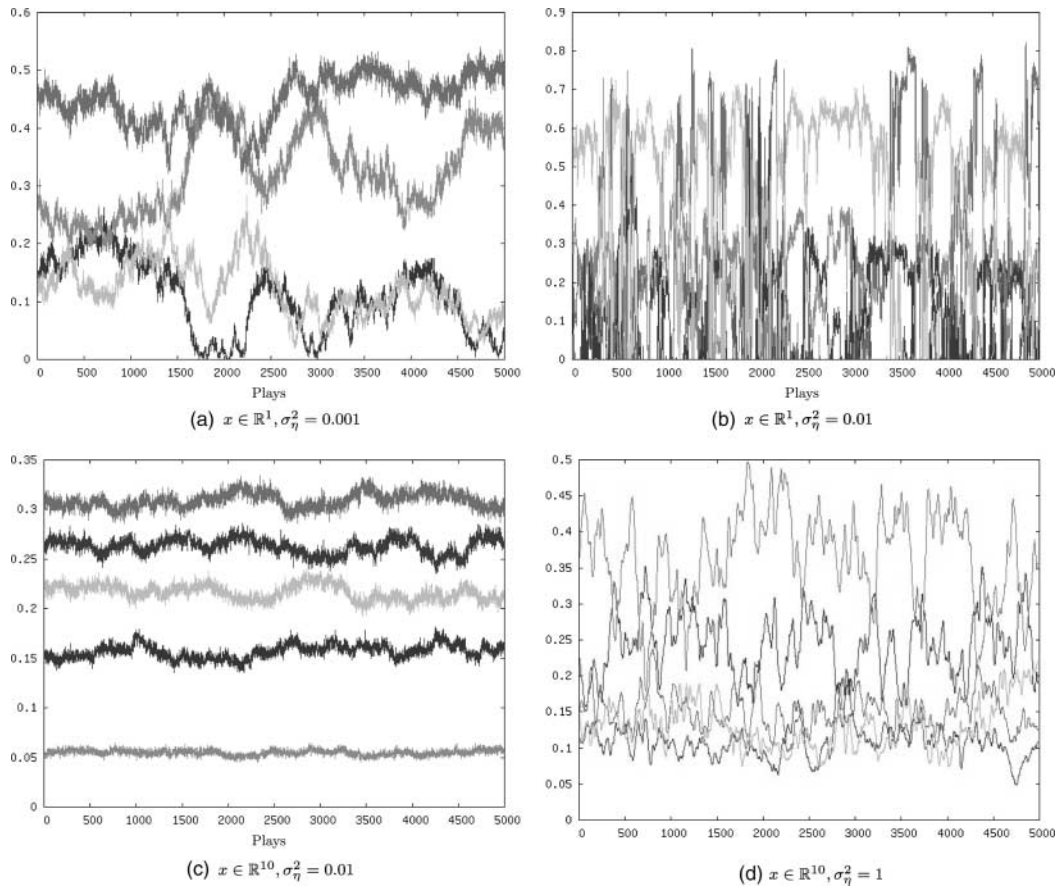

FIGURE 3. Probability of each action being optimal for $x \in \mathbb{R}^1$ and $x \in \mathbb{R}^{10}$ and different values of σ_η^2.

squares. In these cases, the effective speed of change is too large for the filter to track the time variations and treats the phenomenon as static. Therefore, the ability of the ARLS filter to perform accurate estimation constrains the decision-making accuracy.

3.1.2. *Evolution of optimality*

The purpose of introducing dynamics in the MABC is to render the decision-making problem dynamic. In other words, we are interested in problems in which the optimal policy changes over time. A time-varying optimal policy is characterized by changing degrees of optimality of actions over time. The speed of change in Equation (7) is determined by the variance of the disturbance term σ_η^2. However, as Fig. 3 shows, the impact of a particular value of σ_η^2 on the evolution of the optimality of different actions depends largely on the dimensionality of the covariate.

For a five action problem, Fig. 3 illustrates the evolution of the proportion of times each action is optimal over 5000 plays for $x \in \mathbb{R}^1$ and $x \in \mathbb{R}^{10}$ and different values of σ_η^2, given the scaling of the covariance matrix of the covariate distribution Σ with respect to dimensionality that we use. It is clear from the figure that for a given value of σ_η^2 the impact of the dynamics is much more pronounced in lower dimensions. For $x \in \mathbb{R}^1$ a value of $\sigma_\eta^2 = 10^{-3}$ is sufficient to induce the ranking of the optimality of different actions to change gradually, whereas a value of $\sigma_\eta^2 = 10^{-2}$ renders the changes in ranking so frequent and abrupt that tracking their evolution is very difficult. On the other hand, for $x \in \mathbb{R}^{10}$ a value of σ_η^2 equal to unity is required to induce a change in the ranking of different actions similar to that for the case of $x \in \mathbb{R}$ and $\sigma_\eta^2 = 10^{-2}$.

3.1.3. *Optimal degree of exploration*

It appears intuitive that in a continuously changing environment a higher degree of exploration is required compared with the static case. In the static case, the expected returns from exploration decrease over time, whereas this does not necessarily hold in dynamic problems. As pointed out in Section 1, there are several differences in the trade-off between exploitation and exploration when dynamics are introduced. First, because all the actions are continuously changing, selecting a greedy action contributes to knowledge acquisition and hence there is an element of exploration in exploitation. Second, performing an exploratory action selection causes a missing observation for a greedy action. Thus, the cost and character of exploration is different in the dynamic case.

The rationale behind exploration in a static environment is that by improving one's knowledge of the environment higher rewards can be accrued in the future. For this reason, in a static environment with known time horizon it is always advantageous to perform an exploration phase first, followed by an exploitation phase [11]. This rationale is not valid in a continually changing environment, simply because the future differs from the present.

Retaining the convention of characterizing as exploration the selection of a non-greedy action, then for exploration to be beneficial in a dynamic environment it must enable the detection of changes in the optimal policy, that would not have been detected (or would have been detected much later) had the greedy strategy been used. Moreover, the cost of exploration needs to be compensated by the higher reward induced by the more accurate identification of optimal actions.

At this point it is useful to consider the distinction between different types of change made by [29] for the dynamic MAB. According to [29] the three types of changes that can occur in a MAB are: (i) the best action remains the same, but its reward changes; (ii) the reward of another action increases to the extent that it exceeds that of the current best and (iii) the reward of the best action decreases to the point that it becomes lower than that of another action. In the context of the MABC there is no unique best action, but the aforementioned categorization is directly applicable if instead of the best action we consider the actions that are part of the optimal policy (i.e. actions with positive

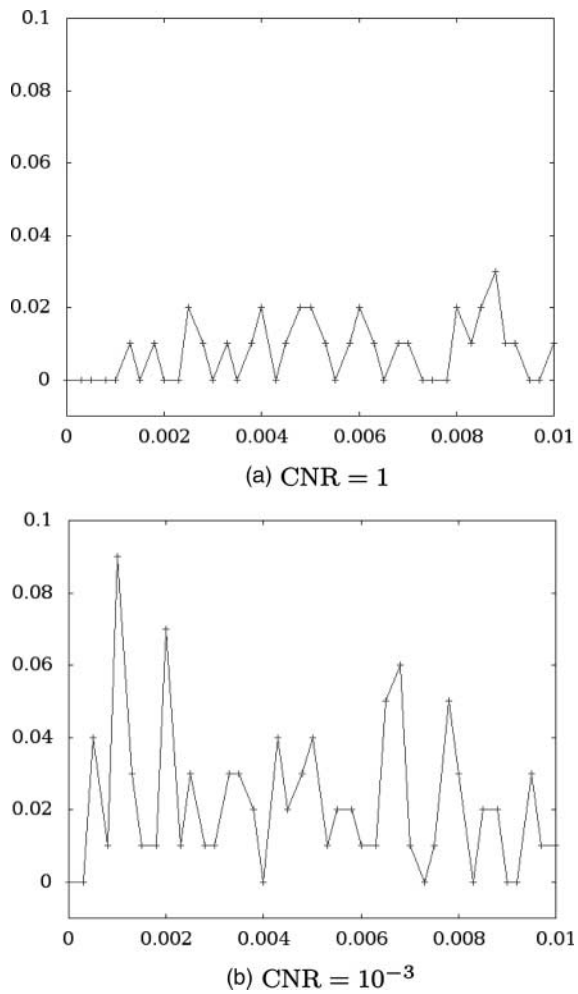

FIGURE 4. Optimal degree of exploration, ε, with respect to the speed of change, σ_η^2, for $x \in \mathbb{R}$.

degree of optimality). Assuming that the true parameters of all the reward functions are known at the start of a simulation, then in terms of the above categorization, only a change of type (ii) requires exploration. The other types of change, which refer to changes in the reward functions of one of the optimal actions, can be handled by greedy selection as long as an appropriate estimation technique is used.

As shown in Section 3.1.2 the dimensionality of the covariate determines to a large extent the impact of the speed of change on the action selection problem. Values of σ_η^2 that produce abrupt and frequent changes in the degree of optimality of different actions in low dimensions have little impact in higher dimensions. Since simulation results for different values of d are not comparable, we provide simulation results for two cases, $x \in \mathbb{R}$ and $x \in \mathbb{R}^{10}$. For $x \in \mathbb{R}$ the values of σ_η^2 considered are in the range [0, 0.01], while for $x \in \mathbb{R}^{10}$, σ_η^2 is in [0, 1]. The range of σ_η^2 is chosen so that the problem exhibits behaviour ranging from static to abruptly and frequently changing optimal policy. In both cases the number of actions is set to 20. Note that for $x \in \mathbb{R}^{10}$, the maximum value of σ_η^2 is five times larger than the maximum value considered in the experiments performed concerning the behaviour of the ARLS filter. Moreover, the number of actions is so large that there are bound to be actions that are very rarely observed. Equivalent to exact initialization in the static case we make the assumption that the true reward equations are known at time zero.

We consider the ε-greedy [30] action selection strategy with $\varepsilon \in [0, 0.5]$ with a step size of 10^{-2}. According to this strategy, at each play the greedy action is selected with probability $(1 - \varepsilon)$, while with the remaining probability, ε, a random action is selected. Note that without making explicit assumptions about the nature of the dynamics, the sensor manager cannot construct confidence bounds for the reward of each action. Hence, upper confidence bound-type strategies [4, 19] are not applicable.

The optimal value of ε with respect to the mean proportion of best action for the case of $x \in \mathbb{R}$ is depicted in Fig. 4 for CNR $= 1, 10^{-3}$. The optimal degree of exploration with respect to the normalized expected regret measure is always zero. A plot similar to Fig. 4 is not produced for the case of $x \in \mathbb{R}^{10}$ because the optimal ε is zero for all the values of σ_η^2. For $x \in \mathbb{R}$, the optimal ε with respect to the proportion of best action exhibits variability

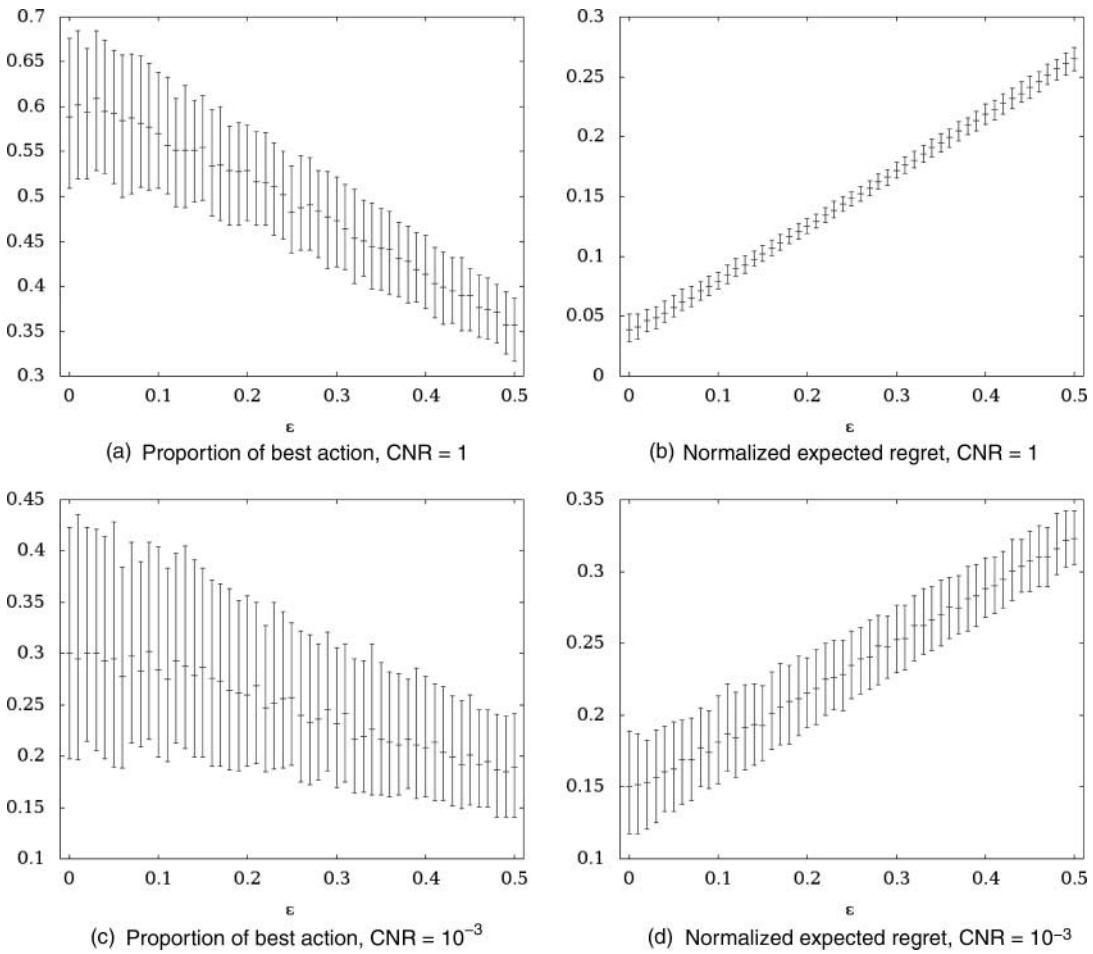

FIGURE 5. Error bars for the performance of different ε-greedy strategies for the case of 20 actions and $x \in \mathbb{R}$.

but is always low, <0.1. Figure 5 depicts the performance of all the ε-greedy strategies, for the cases in which the highest optimal ε is obtained, namely $\sigma_\eta^2 = 0.0088$ for CNR $= 1$, and $\sigma_\eta^2 = 0.001$ for CNR $= 10^{-3}$. Figure 5 shows that for higher values of CNR, there is a clear tendency for performance with respect to the proportion of best action to degrade as the degree of exploration increases. The interquartile ranges of the ε-greedy strategies with very small ε overlap and hence the difference between any one of them and the one with the highest median performance is not substantial. As the value of CNR decreases the choice of ε appears to affect performance less. On the other hand, the superior performance of strategies that perform little or no exploration is clear when normalized expected regret is considered. Performance with respect to this criterion exhibits much less variability because it is a continuous measure.

The performance of the optimal ε-greedy strategy for the different values of σ_η^2 and CNR $= 1$ is illustrated in Fig. 6. Mean performance with respect to both measures is declining while variability increases as σ_η^2 increases. Even in cases in which the dynamics induce frequent and abrupt changes, the optimal action selection strategy manages to perform an order of magnitude better than random action selection with respect to the proportion of best action.

3.2. Sensor management application

We investigate a simple formulation of a sensor management problem that involves monitoring and assessing the threat posed by different sources in an urban environment. We formulate the problem as follows. From the onset K threats that must be monitored are identified. We consider a discrete time model in which threats move at each time-step, but the type of motion is unknown. The sensor manager selects sequentially which threat to monitor. Each threat poses a danger to a number of locations of interest. The term locations of interest is used generically to refer to different types of entities, including static

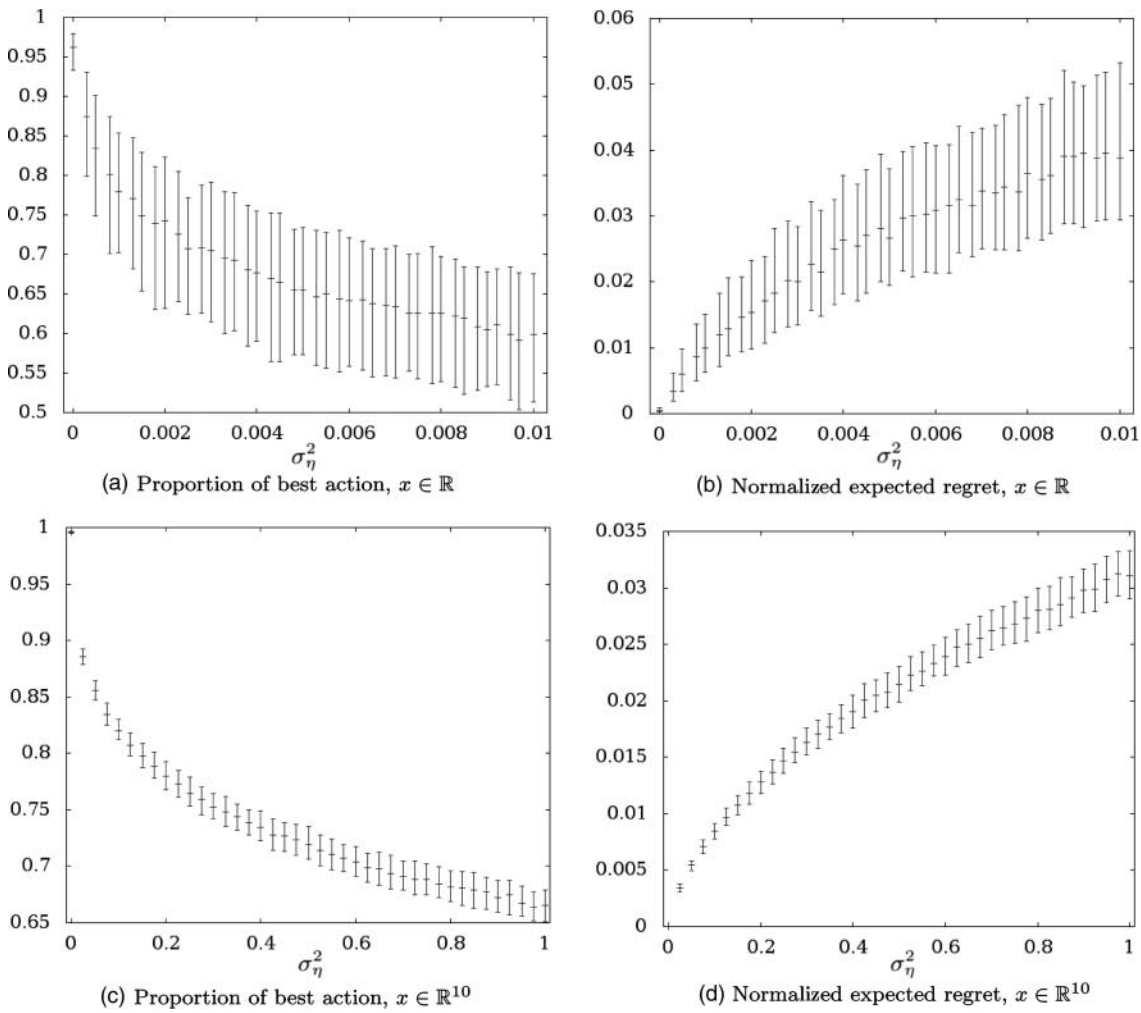

FIGURE 6. Average proportion of best action and normalized expected regret for MABC with 20 actions for $x \in \mathbb{R}^1$ (top row) and $x \in \mathbb{R}^{10}$ (bottom row) with respect to the value of σ_η^2.

entities, like buildings, and moving entities such as vehicles. The threat level is a function of a number of factors, including the type/identity of the threat, its distance from a number of locations of interest, the importance/priority of each location of interest, and its previous activities. The contribution of some factors to the current threat level is constant over time. Such examples are the type/identity of a threat, the danger that a type of threat poses to different types of locations of interest and the priorities of the locations. Other factors, like the distance of a threat from each location of interest change at each time-step. The latter set of factors form the covariate vector upon which the sensor manager conditions its decision at each time-step.

The precise form of the threat function for each type of threat is context dependent. We set the *current threat level* of each threat, $c_i(t)$, to be a linear function of the distance of the threat to each of the L locations interest:

$$c_i(t) = \sum_{j=1}^{L} \beta_{i,j}(t) d_{i,j}(t) + \varepsilon_i. \tag{8}$$

We make the general assumption that the closer a threat is to a particular location the higher its potential to inflict damage. Therefore, the coefficients, $\beta_{i,j}(t)$, are nonpositive.

The current assessment of each threat is also influenced by its history of threat levels. Since only one threat is monitored at each time, this information cannot be part of the covariate vector. Instead, we incorporate this dependence by having the coefficients $\beta_{i,j}(t)$ be functions of an *overall threat level* measure, $l_i(t)$, that captures the history of threat levels. This measure is bounded in $[1, 5]$ with 1 referring to the lowest possible threat and 5 representing the greatest threat. The overall threat level is updated by fixed increments of h. In particular, if the current threat level exceeds the previous threat level by a margin m, then $l_i(t)$ is increased by h, and vice versa. If the margin is not exceeded in any direction, the overall threat level is unchanged.

$$l_i(t+1) = \begin{cases} l_i(t) + h, & \text{if } \Delta c_i(t) > m, \\ l_i(t) - h, & \text{if } \Delta c_i(t) < m, \\ l_i(t), & \text{otherwise.} \end{cases} \tag{9}$$

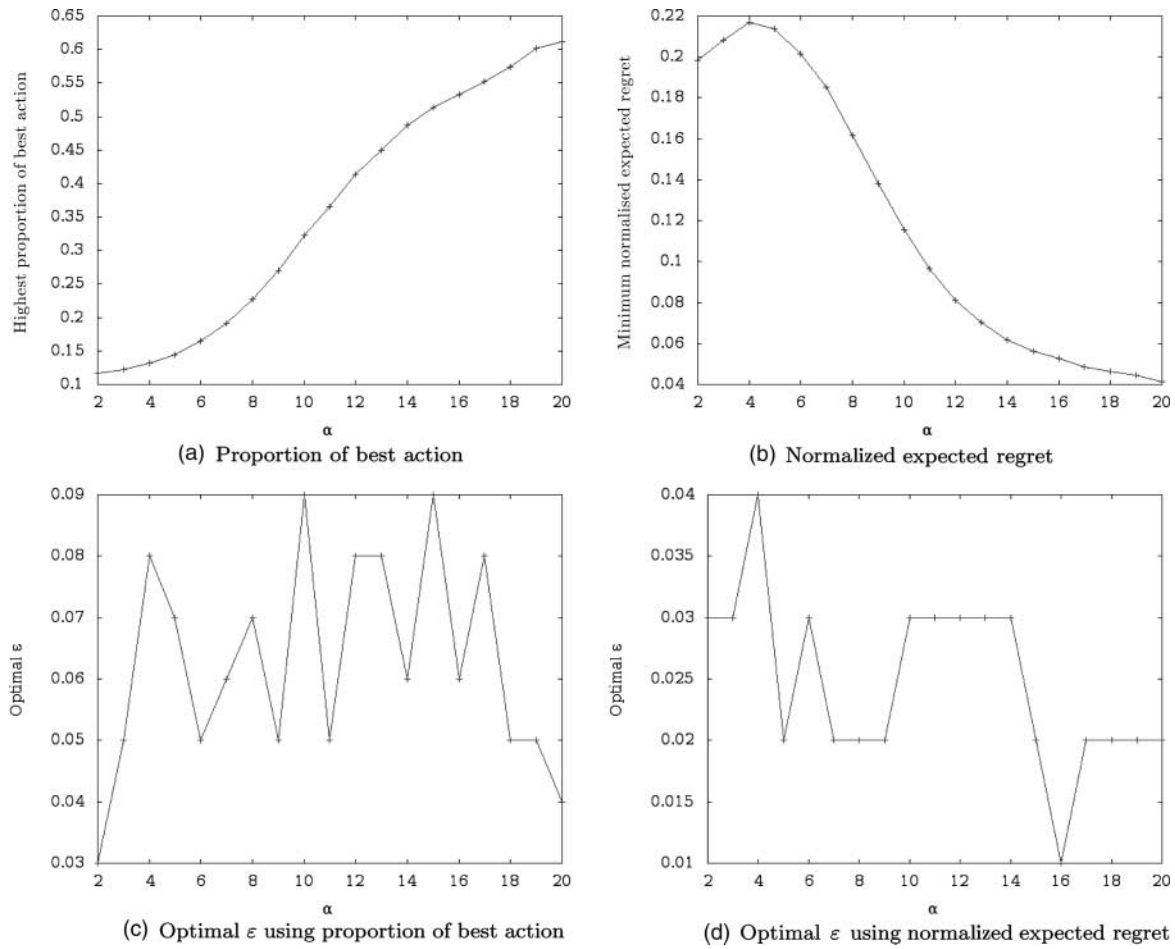

FIGURE 7. Optimal degree of exploration, ε, with respect to the speed of change parameter α.

We let the coefficients $\beta_{i,j}(t)$ be inversely related to $l_i(t)$:

$$\beta_{i,j}(t) = \frac{\beta_{i,j}(0)}{\alpha + l_i(t)}, \quad \beta_{i,j}(0) < 0. \tag{10}$$

In this formulation, $\beta_{i,j}(0)$ reflects the contribution of all the static parameters of the scenario (such as the type of the threat, and the relative priority of each location). The influence of changes of $l_i(t)$ on the current values of the coefficients is controlled by α. Smaller values of α increase the impact of a change of $l_i(t)$, on the value of $\beta_{i,j}(t)$ and vice versa. Hence, the speed of the dynamics is determined by the parameter α. The sensor manager is unaware of this deterministic scheme, but instead attempts to track the evolution of the coefficient vectors through the observed covariate vector and the reward obtained at each instance.

As in the previous section, we investigate the exploitation–exploration dilemma in this setting using the ε-greedy strategy, with $\varepsilon \in [0, 0.5]$. The top row of Fig. 7 shows the maximum median proportion of best action, and the minimum normalized cumulative regret for different values of α on a problem with 20 threats. As expected, increasing the speed of the dynamics, i.e. smaller values of α, reduces the best attainable performance. An exception occurs for small values of α were an increase appears to impair the optimal median performance with respect to normalized cumulative regret. This can be attributed to the higher variability of performance that is observed when the speed of dynamics increases, which will be discussed below. The bottom row of the figure depicts the values of ε that yielded the best performance with respect to the two criteria. As in the artificial dynamic MABC the optimal value of ε is lower when normalized expected regret is used as the performance criterion. The optimal value of ε is below 0.1 even when the mean proportion of best action is used as performance criterion.

A more detailed illustration of the overall performance induced by different ε-greedy strategies is provided in Fig. 8. This figure depicts the performance of ε-greedy strategies with $\varepsilon \in [0, 0.5]$ on a problem with 20 threats, for two values of α. The top row of the figure corresponds to $\alpha = 20$, and the bottom row corresponds to $\alpha = 2$. Clearly, increasing the speed of the dynamics (i.e. a lower value of α) impairs the performance of the sensor manager for all the values of ε. However, irrespective of the value of α, there appears a distinct pattern in performance

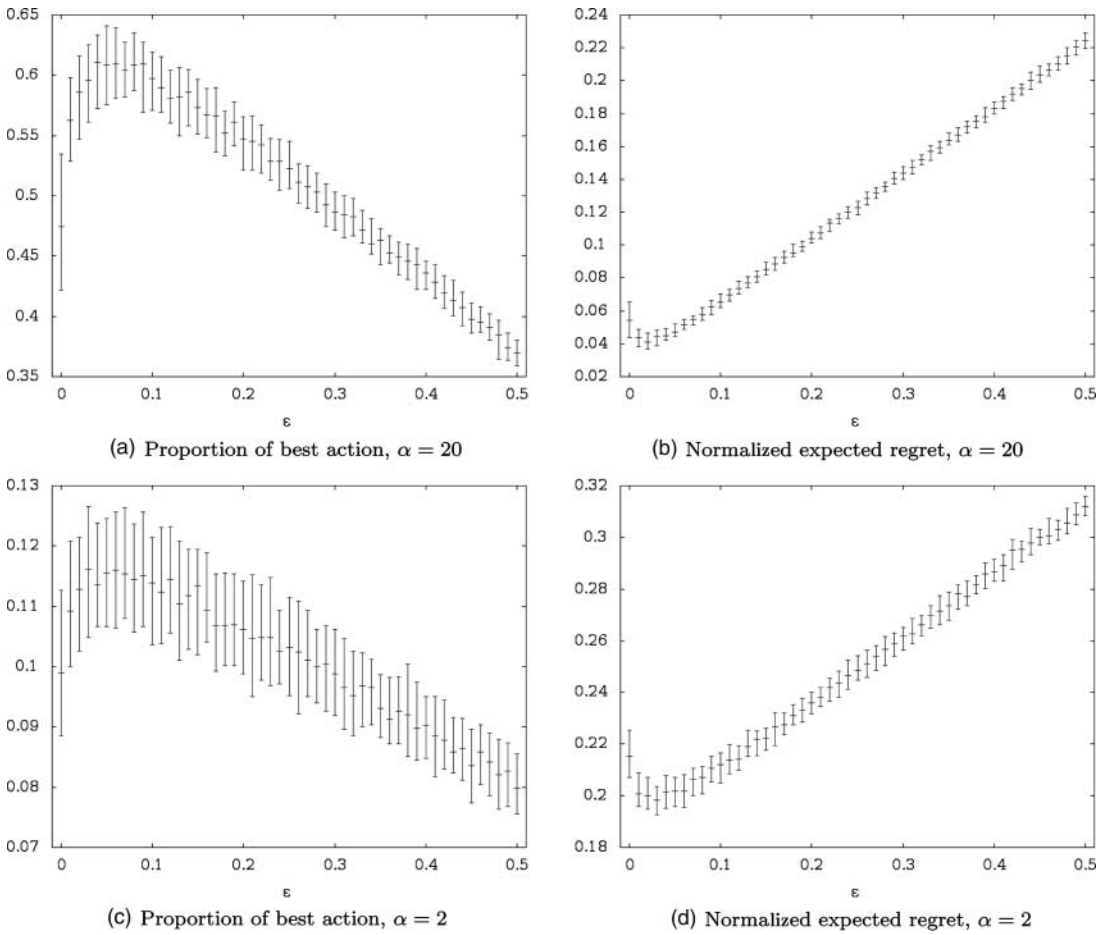

FIGURE 8. Average proportion of best action and cumulative normalized expected regret in sensor management simulation with 20 threats for $\alpha = 40$ (top row) and $\alpha = 2$ (bottom row).

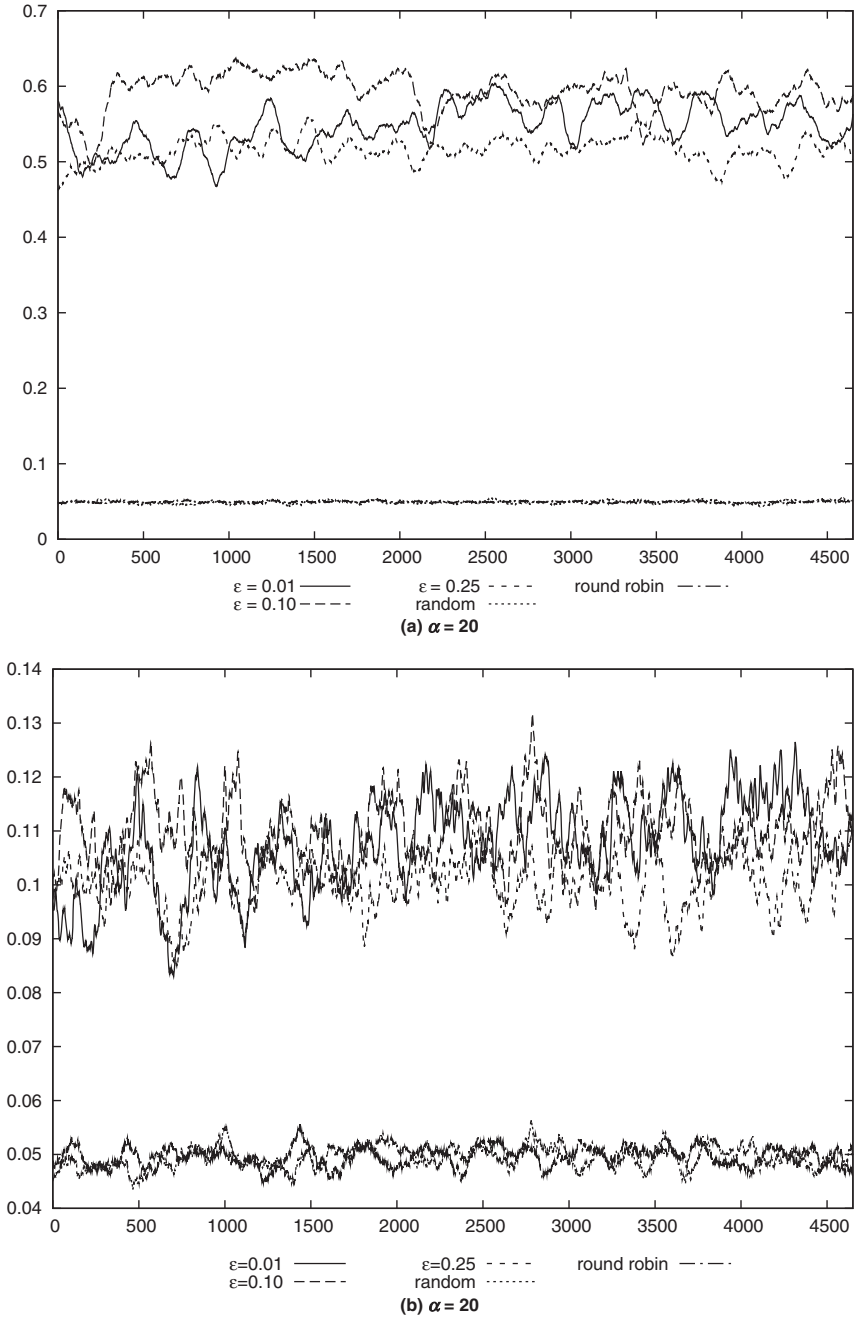

FIGURE 9. Proportion of best action over time for $\alpha = 2, 20$.

with respect to ε. Introducing a marginal degree of exploration improves performance over the purely greedy strategy with respect to both measures. Increasing ε further can cause an increase in median performance, but the interquartile ranges of the performance of strategies with small ε overlap substantially. For values of ε exceeding 0.1 there is an apparent trend of performance degradation.

Figure 9 illustrates the evolution of performance of three ε-greedy strategies and two benchmark strategies, namely random and round robin assignment. By ignoring the covariate information both random and round robin assignment identify the optimal action at each time-step with a constant probability of $1/N$, where N is the number of actions. For the two values of α considered the three ε-greedy strategies are capable of identifying the optimal action more accurately. For the case of slower dynamics, $\alpha = 20$, the 0.01-greedy strategy improves its performance over time and eventually identifies the optimal action almost as well as the 0.1-greedy strategy. Figure 9b shows

that as the speed of dynamics increases the performance of different strategies becomes indistinguishable.

4. CONCLUDING REMARKS

In this work, we investigate sensor management problems in which the consequences of different actions are affected by a set of covariates that are observed prior to each action selection. In our formulation, the consequences of the actions are immediately felt, and selected actions do not affect subsequent realizations of the covariates. The sensor manager's goal is to learn the optimal policy, which is a rule that associates covariate vectors to the optimal action. Sensor management is typically deployed in dynamic environments, where there is frequently uncertainty concerning the underlying process that governs change. In the present context, a dynamic environment is characterized by changing relationships between the covariates and the rewards. The sensor manager therefore has to revise its policy over time to attain satisfactory performance. The dynamic multi-armed bandit with covariates is a natural implementation of this framework. At present we restrain our attention to the case that the rewards are linear functions of the covariates.

We study the case in which the sensor manager does not know and is not trying to learn the nature of the dynamics, but instead employs an adaptive filter to estimate the parameters of the reward functions. The incorporation of dynamics has important repercussions for both the estimation and the action selection problems. Introducing dynamics blurs the distinction between exploitation and exploration. Because the rewards are changing, there is an exploratory component in every action selection, even in exploitation, as information about the evolution of a greedy action is obtained. Furthermore, in a changing environment the future differs from the past and the present. Therefore, the reasoning that exploration improves the knowledge of the environment and enables better action selection in the future may not be valid.

Experimental results show that when the speed of change is large and the sampling frequency of the time-varying equation is low the adaptive filter behaves like standard LLS. Therefore, in problems in which accurate estimates of the reward functions are required, an explicit model, or a learning mechanism, is necessary to handle dynamics. We also show that increasing the dimensionality of the covariate, while holding the rate of change constant, renders the estimation problem harder, but not the decision-making problem.

Through simulation we show that for different types of dynamics and a wide range of parameter settings, the optimal behaviour of the sensor manager is to be almost completely greedy. Therefore, in dynamic problems in which the type of dynamics is unknown the cost of exploration appears to exceed the benefit. Surprisingly, this is true for both slowly and rapidly changing environments. In slowly changing environments the changes in the optimal policy are so gradual that monitoring suboptimal actions degrades overall performance. In contrast, in rapidly changing environments there is no time for the sensor manager to accrue the benefits from learning more accurately the current configuration of the reward functions. Therefore, the intuitive argument that in a changing environment more exploration is required to timely identify changes in the optimal policy does not seem to hold in this case. To perform accurate action selection in dynamic sensor management applications without a model of the dynamics requires either that the speed of the dynamics is relatively slow or that the type of dynamics is such that the sensor manager has to effectively distinguish between few actions. The experimental results suggest that as the speed of change increases the performance of all the action selection strategies degrades and approaches that of random action selection. To improve performance in these cases, or to render this performance degradation more gradual a highly accurate model of the dynamics is necessary.

In this work, we studied a single agent action selection problem. An extension of this work would be to consider multi-agent action selection in uncertain and dynamic environments. In this context, at each time each agent selects an action and the joint actions determine the overall reward. Such extensions have been investigated in the ALADDIN project. For example, a sensor management application of this formulation is the determination of the sampling frequency of each sensor in a sensor field, with the overall reward being a measure of tracking accuracy. In this case, each sensor sets its sampling frequency so as to obtain regular measurements of the target, but also avoid network congestion that causes delays and therefore performance deterioration [31]. An online multi-agent action selection approach based on MAB is also relevant to disaster recovery applications [32, 33] where a population of agents, for example mobile sensors, must coordinate their actions to optimize collective information acquisition.

FUNDING

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership, under EPSRC grant EP/C548051/1. D. H. was partially supported by a Royal Society Wolfson Research Merit Award.

REFERENCES

[1] Hero III, A.O., Castañón, D.A., Cochran, D. and Kastella, K. (eds) (2008) *Foundations and Applications of Sensor Management*. Springer.

[2] Sutton, R.S. and Barto, A.G. (1998) *Reinforcement Learning: An Introduction*. MIT Press.

[3] Robbins, H. (1952) Some aspects of the sequential design of experiments. *Bull. Am. Math. Soc.*, **55**, 527–535.

[4] Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002) Finite-time analysis of the multi-armed bandit problem. *Mach. Learn.*, **47**, 235–256.

[5] Even-dar, E., Mannor, S. and Mansour, Y. (2006) Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *J. Mach. Learn. Res.*, **7**, 1079–1105.

[6] Lai, T.L. and Robbins, H. (1985) Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, **6**, 4–22.

[7] Hardwick, J.P. and Stout, Q.F. (1991) Bandit strategies for ethical sequential allocation. *Computing Science and Statistics*, **23**, 421–424.

[8] Awerbuch, B. and Kleinberg, R. (2004) Adaptive Routing with End-to-End Feedback: Distributed Learning and Geometric Approaches. *36th ACM Symp. Theory Computing (STOC 2004)*, Chicago, USA, pp. 45–53.

[9] Vermorel, J. and Mohri, M. (2005) Multi-armed Bandit Algorithms and Empirical Evaluation. In Gama, J., Camacho, R., Brazdil, P., Jorge, A. and Torgo, L. (eds.), *Lecture Notes in Artificial Intelligence*, Springer, Berlin, pp. 437–448.

[10] Bergemann, D. and Välimäki, J. (2006) Bandit Problems. Technical Report. HECER—Helsinki Center of Economic Research, University of Helsinki.

[11] Langford, J. and Zhang, T. (2008) The Epoch–Greedy Algorithm for Multi-Armed Bandits with Side Information. In Platt, J., Koller, D., Singer, Y. and Roweis, S. (eds), *Advances in Neural Information Processing Systems 20*, pp. 817–824. MIT Press.

[12] Xiong, N. and Svensson, P. (2002) Multi-sensor management for information fusion: issues and approaches. *Inf. Fusion*, **3**, 163–186.

[13] Nicholson, D. (2009) Defence industry applications of multi-agent information fusion and control. *Comput. J.* (this issue).

[14] Abe, N., Biermann, A.W. and Long, P.M. (2003) Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, **37**, 263–293.

[15] Biermann, A.W. and Long, P.M. (1996) The Composition of Messages in Speech-Graphics Interactive Systems. *Int. Symp. Spoken Dialogue*, Philadelphia, USA, pp. 97–100.

[16] Hoang, D.T., Long, P.M. and Vitter, J.S. (1996) Efficient Cost Measures for Motion Compensation at Low Bit Rates. *IEEE Data Compression Conf. (DCC 1996)*, Snowbird, Utah, pp. 102–111.

[17] Yang, Y. and Zhu, D. (2002) Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *Ann. Stat.*, **30**, 100–121.

[18] Auer, P. (2000) An Improved Algorithm for Learning Linear Evaluation Functions. In Cesa-Bianchi, N. and Goldman, S.A. (eds) *13th Annual Conf. Computational Learning Theory (COLT 2000)*, Palo Alto, CA, USA, pp. 118–125.

[19] Auer, P. (2002) Using confidence bounds for exploitation–exploration trade-offs. *J. Mach. Learn. Res.*, **3**, 397–422.

[20] Cappeé, O. and Moulines, E. (2009) On-line expectation–maximization algorithm for latent data models. *J. R. Stat. Soc. B*, **71**, 593–614.

[21] Rasmussen, C. and Williams, C. (2006) *Gaussian Processes for Machine Learning*. MIT Press.

[22] Benveniste, A., Priouret, P. and Métivier, M. (1990) *Adaptive Algorithms and Stochastic Approximations*. Springer.

[23] Haykin, S. (1996) *Adaptive Filter Theory* (3rd edn). Prentice-Hall International.

[24] Benesty, J. and Huang, Y. (eds) (2003) *Adaptive Signal Processing—Applications to Real-World Problems*. Springer.

[25] Buche, R. and Kushner, H.J. (2005) Adaptive optimization of least-squares tracking algorithms: With applications to adaptive antenna arrays for randomly time-varying mobile communications systems. *IEEE Trans. Automat. Contr.*, **50**, 1749–1760.

[26] Kailath, T. (1980) *Linear Systems*. Prentice-Hall.

[27] Haggan, V. and Ozaki, T. (1981) Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model. *Biometrika*, **68**, 189–196.

[28] van Dijk, D., Teräsvirta, T. and Franses, P.H. (2002) Smooth transition autoregressive models a survey of recent developments. *Econom. Rev.*, **21**, 1–47.

[29] Hartland, C., Gelly, S., Baskiotis, N., Teytaud, O. and Sebag, M. (2006) Multi-armed Bandit, Dynamic Environments and Meta-bandits. *Online Trading of Exploration and Exploitation Workshop, NIPS 2006*, Vancouver, Canada.

[30] Watkins, C.J.C.H. (1989) Learning from Delayed Rewards. PhD Thesis, Cambridge University, UK.

[31] Tasoulis, D., Adams, N. and Hand, D. (2009) Selective fusion of out-of-sequence measurements. *Inf. Fusion*, doi:10.1016/j.inffus.2009.06.002, in press.

[32] Stranders, R., Farinelli, A., Rogers, A. and Jennings, N. (2009) Decentralised Coordination of Mobile Sensors Using the Max-sum Algorithm. *21st Int. Joint Conf. Artificial Intelligence (IJCAI)*, Pasadena, CA, USA, pp. 299–304.

[33] Stranders, R., Rogers, A. and Jennings, N. (2008) A Decentralized, On-Line Coordination Mechanism for Monitoring Spatial Phenomena with Mobile Sensors. *2nd Int. Workshop on Agent Technology for Sensor Networks*, Estoril, Portugal, pp. 9–15.