Appl Intell
DOI 10.1007/s10489-012-0346-z

Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the Goore Game

Ole-Christoffer Granmo · Sondre Glimsdal

Abstract The two-armed bandit problem is a classical optimization problem where a decision maker sequentially pulls one of two arms attached to a gambling machine, with each pull resulting in a random reward. The reward distributions are unknown, and thus, one must balance between exploiting existing knowledge about the arms, and obtaining new information. Bandit problems are particularly fascinating because a large class of real world problems, including routing, Quality of Service (QoS) control, game playing, and resource allocation, can be solved in a decentralized manner when modeled as a system of interacting gambling machines.

Although computationally intractable in many cases, Bayesian methods provide a standard for optimal decision making. This paper proposes a novel scheme for decentralized decision making based on the Goore Game in which each decision maker is inherently Bayesian in nature, yet avoids computational intractability by relying simply on updating the hyper parameters of sibling conjugate priors, and on random sampling from these posteriors. We further report theoretical results on the variance of the random rewards experienced by each individual decision maker. Based on these theoretical results, each decision maker is able to accelerate its own learning by taking advantage of the increasingly more reliable feedback that is obtained as exploration gradually turns into exploitation in bandit problem based learning.

Extensive experiments, involving QoS control in simulated wireless sensor networks, demonstrate that the accelerated learning allows us to combine the benefits of conservative learning, which is high accuracy, with the benefits of hurried learning, which is fast convergence. In this manner, our scheme outperforms recently proposed Goore Game solution schemes, where one has to trade off accuracy with speed. As an additional benefit, performance also becomes more stable. We thus believe that our methodology opens avenues for improved performance in a number of applications of bandit based decentralized decision making.

Keywords Bandit problems · Goore Game · Bayesian learning · Decentralized decision making · Quality of service control · Wireless sensor networks

1 Introduction

The conflict between exploration and exploitation is a well-known problem in reinforcement learning, and other areas of artificial intelligence. The *Two-Armed Bandit* (TAB) problem captures the essence of this conflict. In brief, a decision maker sequentially pulls one of two arms attached to a gambling machine, with each pull resulting in a random reward. The reward distributions are unknown, and thus one must balance between exploiting existing knowledge about the arms, and obtaining new information.

1.1 Thompson sampling

In [8] we reported a *Bayesian* technique for solving bandit like problems, revisiting the *Thompson Sampling* [21] principle pioneered in 1933. This revisit lead to novel schemes for handling multi-armed, and dynamic (restless) bandit

A preliminary version of this paper was presented at IEA/AIE'11, the 2011 International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Syracuse, NY, USA, in June 2011.

O.-C. Granmo (✉) · S. Glimsdal
Department of Information and Communication Technology, University of Agder, Post Box 422, 4604 Kristiansand, Norway
e-mail: ole.granmo@uia.no

S. Glimsdal
e-mail: sondre.glimsdal@uia.no

Published online: 10 April 2012

problems [9, 13, 14], and empirical results demonstrated the advantages of these techniques over established top performers. Furthermore, we provided theoretical results stating that the original technique is instantaneously self-correcting and that it converges to only pulling the optimal arm, with probability as close to unity as desired. In addition to the theoretical convergence results found in [8], May et al. recently reported an alternative proof strategy for establishing convergence properties for *optimistic* Bayesian sampling [16]. As a further testimony to the renewed importance of the Thompson Sampling principle, Wang et al. [24] combined so-called sparse sampling with Bayesian exploration, enabling efficient searching of the arm selection space using a sparse look-ahead tree. In [4], Dimitrakakis derived optimal decision thresholds for the multi-armed bandit problem, for both the infinite horizon discounted reward case, and for the finite horizon undiscounted case. Later on, a modern Bayesian look at the multi-armed bandit problem was also taken in [16, 20]. Promising recent application areas for Thompson Sampling include Bayesian click-through rate optimization for sponsored search advertising [7] and web site optimization [6, 20].

1.2 Decentralized decision making and multi-armed bandits

Multiple *interacting* bandits problems are particularly fascinating because they can be used to model, and efficiently solve a large class of real world decentralized decision making problems, such as QoS-control in wireless sensor networks [15], routing [19], game playing [5], combinatorial optimization [1, 10], and resource allocation [11, 12]. In decentralized decision making problems, however, a certain phenomenon renders current bandit problem based solutions sub-optimal. Specifically, multiple decentralized decision makers are simultaneously exploring a collection of interacting bandits. This means that the variances of the reward distributions of each bandit problem are governed by the current level of exploration being manifested in the system as a whole. In other words, the variance of the reward distributions will be fluctuating with the degree of exploration taking place. Thus, initially, when exploration typically is significant, each decision maker should be correspondingly more conservative or cautious when interpreting the received rewards. Otherwise, by being too reckless, the decision maker may be led astray early on, converging to a sub-optimal decision.

The traditional approach to dealing with the above described fluctuation of reward distribution variance is to make learning sufficiently conservative. The purpose is to minimize the chance of each decision maker converging prematurely. Obviously, the disadvantages of this approach is the corresponding loss in learning speed caused by being too conservative also when exploration calms down. A recent approach deals with this problem indirectly by incorporating a Kalman filter into the decision making [9], allowing each decision maker to track changing reward distributions. Thus, too reckless learning initially is offset by the "forgetting" mechanism of the Kalman filter. This means that premature convergence is hindered. Yet, this tracking of changing reward distributions also means that exploration never stops. The decision makers will, as a result, never converge to a single optimal decision.

1.3 Paper contributions and organization

In this paper, we propose a novel scheme for solving one particular class of decentralized decision making problems, namely, the *Goore Game* (GG) [22]. The GG has applications within QoS control in wireless sensor networks, and we described both the GG and recent applications in Sect. 2. We then proceed to introduce a scheme for *Accelerated Decentralized Learning in Two-Armed Bandit Based Decision Making (ADL-TAB)* in Sect. 3. The ADL-TAB scheme directly and specifically addresses fluctuating reward distribution variances. To achieve this, we derive theoretical results that characterize the variance of the random rewards experienced by each individual decision maker. Based on these theoretical results, each decision maker is able to accelerate its own learning as follows. When a decision maker chooses which arm to pull, it also submits a measurement of its degree of exploration, which we refer to as *arm selection variance*. In turn, along with the random reward it receives from the arm pull, it also receives a signal that reflects the current aggregated level of exploration being manifested in the system. Using this signal, each decision maker accelerates learning by taking advantage of the increasingly more reliable feedback that can be obtained when exploration gradually turns into exploitation. In Sect. 4, we demonstrate empirically that the accelerated learning allows us to combine the benefits of conservative learning, which is high accuracy, with the benefits of hurried learning, which is fast convergence. We also achieve significant performance benefits when applying ADL-TAB to QoS control in wireless sensor networks that are dynamically changing through a stochastic sensor birth-death process. In brief, our scheme clearly outperforms recently proposed Goore Game solution schemes, where one has to trade off accuracy with speed. Finally, in Sect. 5, we conclude and provide pointers to further research.

2 The Goore Game (GG)

The GG is one of the most fascinating games studied in the field of artificial intelligence, and our presentation here is based on the exposition found in [18]. We describe the GG using the following informal formulation given in [17]:

Imagine a large room containing N cubicles and a raised platform. One person (voter) sits in each cubicle and a Referee stands on the platform. The Referee conducts a series of voting rounds as follows. On each round the voters vote "Yes" or "No" (the issue is unimportant) simultaneously and independently (they do not see each other) and the Referee counts the fraction, λ, of "Yes" votes. The Referee has a uni-modal performance criterion $G(\lambda)$, which is optimized when the fraction of "Yes" votes is exactly λ^*. The current voting round ends with the Referee awarding a dollar with probability $G(\lambda)$ and assessing a dollar with probability $1 - G(\lambda)$ to every voter independently. On the basis of their individual gains and losses, the voters then decide, again independently, how to cast their votes on the next round.

The game has many interesting and fascinating features which render it both non-trivial and intriguing. These are listed below:

1. The game is a non-trivial *non-zero-sum* game.
2. Unlike the games traditionally studied in the AI literature (like Chess, Checkers, Lights-Out, etc.) the game is essentially a *distributed* game.
3. The players of the game are ignorant of all of the parameters of the game. All they know is that they have to make a choice, for which they are either rewarded or penalized. They have no clue as to how many other players there are, how they are playing, or even of how/why they are rewarded/penalized.
4. The stochastic function used to reward or penalize the players can be completely arbitrary, as long as it is unimodal.

The literature concerning the GG is sparse. It was initially studied in the general learning domain, and, as far as we know, was for a long time merely considered as an interesting pathological game. Recently, however, the GG has found important applications within two main areas, namely, Quality of Service (QoS) control in wireless sensor networks [3] and within cooperative mobile robotics, as summarized in [2].

The GG has found applications within the field of wireless sensor networks, as explained briefly here. Consider a base station that collects data from a sensor network. The sensors of the network are battery driven and have been dropped from the air, leaving some of them non-functioning. The functioning sensors can either be switched on or off, and since they are battery-driven, it is expedient that they should be turned off whenever possible. The base station, on the other hand, has been set to maintain a certain resolution (i.e., QoS), and therefore requires that Q sensors are switched on. Unfortunately, it does not know the number of functioning sensors, and it is only able to contact them by means of a broadcast, leaving it unable to address them individually. This leaves us with the following challenge: *How can the base station turn on exactly Q sensors, only by means of its limited broadcast capability?*

Iyer et al. [15] proposed a scheme where the base station provided broadcasted QoS feedback to the sensors of the network. Using this model, the above problem was solved by modeling it as a GG [23]. From the GG perspective, a sensor is seen as a voter that chooses between transmitting data or remaining idle in order to preserve energy. Thus, in essence, each sensor takes the role of a GG player that either votes "On" or "Off", and acts accordingly. The base station, on the other hand, is seen as the GG Referee with a uni-modal performance function $G(\cdot)$ whose maximum is found at Q normalized by the total number of sensors available. The "trick" is to let the base station (1) count the number of sensors that have turned on, and (2) use the broadcast mechanism to distribute, among the sensors, the corresponding reward based on the probability obtained from $G(\cdot)$. The application of the GG solution to the field of sensor networks is thus both straightforward and obvious.

Furthermore, Tung and Kleinrock [23] have demonstrated how the GG can be used for coordinating groups of mobile robots (also called "mobots") that have a restricted ability to communicate. The main example application described in [23] consists of a fixed number of mobots that can either (1) collect pieces of ore from a landscape, or (2) sort already collected ore pieces. The individual mobots vary with respect to how fast they collect and how fast they sort these pieces of ore. In this context, the GG is used to make sure that the mobots choose their action so as to maximize the throughput of the overall collection and sorting system.

Other possible cooperative robotics applications include controlling a moving platform and guarding a specified perimeter [2]. In all of these cases, the solution to the problem in question would essentially utilize the solution to the GG in a plug-and-play manner.

3 Accelerated decentralized learning in two-armed bandit based decision making (ADL-TAB)

This paper proposes a novel scheme for decentralized decision making in which each decision maker is inherently Bayesian in nature, yet avoids computational intractability by relying simply on updating the hyper parameters of sibling conjugate priors, and on random sampling from these posteriors. Based on the sibling conjugate priors, we also measure the current degree of exploration and exploitation being manifested in the system as a whole. This allows each decision maker to accelerate its learning by taking advantage of the increasingly more reliable feedback that can be obtained when exploration gradually turns into exploitation.

3.1 Bayesian sampling for two-armed normal bandits (BS-TANB)

At the heart of our decentralized decision making scheme, we find a Bayesian Sampling approach to Two-Armed Normal Bandits (BS-TANB) problems. A unique feature of BS-TANB is its computational simplicity, achieved by relying *implicitly* on Bayesian reasoning principles. Possessing a bell-shaped probability density function with mean μ and standard deviation σ

$$f(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}(\frac{(x-\mu)^2}{\sigma^2})}$$

the normal distribution, $N(\mu,\sigma)$, is central to BS-TANB. Essentially, BS-TANB uses the normal distribution for two purposes. First of all, it is used to provide a *Bayesian estimate* of the reward expectation associated with each of the available bandit arms. Secondly, a pertinent feature of BS-TANB is that it uses the normal distribution as the basis for a *randomized arm selection mechanism*. The following algorithm contains the essence of BS-TANB (see [9] for further details).

Algorithm: BS-TANB
Input: Observation noise σ^2_{ob}.
Initialization: $\mu_0[1]=\mu_1[1]=A;\ \sigma_0[1]=\sigma_1[1]=B$; *# Typically, A can be set to* 0*, with B being sufficiently large.*
Method:
For $t=1,2,\ldots$ **Do**

1. For each *Arm*, $j\in\{0,1\}$, draw a value x_j randomly from the associated *normal* distribution, $N(\mu_j[t],\sigma_j[t])$.
2. Pull the *Arm* i whose drawn value x_i is the largest one:

$$\alpha[t]=i=\arg\max_{j\in\{0,1\}} x_j.$$

3. Receive a reward $\tilde{r}_i$ from pulling *Arm* i, and update parameters as follows:

- *Arm* i:

$$\mu_i[t+1]=\frac{\sigma_i^2[t]\cdot\tilde{r}_i+\sigma^2_{\text{ob}}\cdot\mu_i[t]}{\sigma_i^2[t]+\sigma^2_{\text{ob}}}$$

$$\sigma_i^2[t+1]=\frac{\sigma_i^2[t]\sigma^2_{\text{ob}}}{\sigma_i^2[t]+\sigma^2_{\text{ob}}}$$

- *Arm* $j\neq i$:

$$\mu_j[t+1]=\mu_j[t]$$
$$\sigma_j^2[t+1]=\sigma_j^2[t]$$

End

As seen from the above BS-TANB algorithm, t is a discrete time index and the parameters $\phi^t=\langle(\mu_0[t],\sigma_0[t]),(\mu_1[t],\sigma_1[t])\rangle$ form an infinite 4-dimensional continuous state space, with each pair $(\mu_i[t],\sigma_i[t])$ giving the prior distribution of the unknown reward r_i associated with *Arm* i. Within Φ the BS-TANB navigates by transforming each prior distribution into a posterior distribution, based on the rewards $\tilde{r}_i$ obtained from selecting *Arm* i, $\alpha[t]=i$, as well as the observation noise, σ^2_{ob}, given as an input parameter to the algorithm. Essentially, the algorithm uses observation noise, σ^2_{ob}, to determine how much emphasis to put on the reward $\tilde{r}_i$, which is a crucial property that we will now take advantage of.

In the interest of notational simplicity, let *Arm* 1, $\alpha[t]=1$, be the arm under investigation. Then, for any parameter configuration $\phi^t\in\Phi$ we can state, using a generic notation,[1] that the probability of selecting *Arm* 1, $\alpha[t]=1$, is equal to the probability $P(X_1>X_0|\phi^t)$—the probability that a randomly drawn value $x_1\in X_1$ is greater than the other randomly drawn value $x_0\in X_0$ at time step t. Since the associated stochastic variables X_0 and X_1 are normally distributed, with parameters $(\mu_0[t],\sigma_0[t])$ and $(\mu_1[t],\sigma_1[t])$, respectively, we have that:

$$\begin{aligned}P\big(\alpha[t]=1\big)&=P\big(X_1\geq X_0|\phi^t\big)\\&=\int_{-\infty}^{0}f\Big(x;\mu_0[t]-\mu_1[t],\sqrt{\sigma_0^2[t]+\sigma_1^2[t]}\Big)\end{aligned}\tag{1}$$

In the following, we will let $p[t]$ denote this latter probability.

3.2 BS-TANB based decentralized decision making

The overall decentralized decision making scheme that we propose is illustrated in Fig. 1. On each round t, the n decision makers $V_q\in\{V_1,\ldots,V_n\}$ choose one of two arms, $\alpha_q[t]=i\in\{0,1\}$, simultaneously and independently (they do not see each other), with $\alpha_q[t]=0$ referring to a "No"-vote and $\alpha_q[t]=1$ referring to a "Yes"-vote.

Let $p_q[t]=P(\alpha_q[t]=1)$ be the probability that decision maker V_q casts a "Yes" vote on round t. Then $1-p_q[t]$ is the probability that V_q casts a "No" vote, and each voting $\alpha_q[t]$ can be seen as a Bernoulli trial in which a "Yes" vote is a success and a "No" vote is a failure. Note that the concrete instantiation of the arm selection probability $p_q[t]$ is governed by the learning scheme applied, which in our case is BS-TANB.

Definition 1 (Arm Selection Variance) In a two-armed bandit problem where the current arm selection probability is p, we define *Arm Selection Variance*, σ^2, to be the variance,

[1] By this we mean that P is not a fixed function. Rather, it denotes the probability function for a random variable, given as an argument to P.

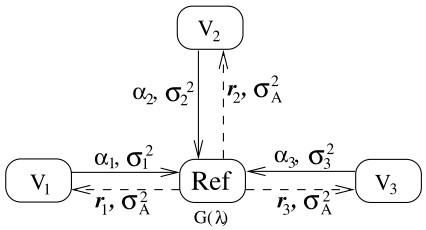

Fig. 1 Decentralized decision making with accelerated learning

$p(1-p)$, of the outcome of the corresponding Bernoulli trial.

As seen in Fig. 1, in addition to casting a vote $\alpha_q[t]$, each decision maker V_q also submits its present *Arm Selection Variance*, $\sigma_q^2[t]$, in order to signal its level of exploration. Thus, as in the traditional Goore Game setup, a Referee calculates the fraction, $\lambda[t]$, of "Yes" votes. In addition, it now also calculates the variance $\sigma_A^2[t]$ of the total number of "Yes" votes, which simply is the sum of the variances of the independently cast votes (cf. Bienayme formula): $\sigma_A^2[t] = \sum_{q=1}^{n} \sigma_q^2[t]$. Note that in practice, such as in QoS control in wireless sensor networks [15], this operation is conducted by the so-called base station of the network.

The Referee has a uni-modal *normally distributed* performance criterion $G(\lambda[t]; \mu_G, \sigma_G)$, where μ_G is the mean and σ_G^2 is the variance, which is thus optimized when the fraction of "Yes" votes is exactly μ_G, $\lambda[t] = \mu_G$. The current voting round ends with the Referee awarding a reward $\tilde{r}_i$ to each voter, with the reward being of magnitude $G(\lambda[t]; \mu_G, \sigma_G)$. Additionally, white noise $N(0, \sigma_W)$ is independently added to the reward received by each voter.

On the basis of their individual gains, the voters then decide, again independently how to cast their votes on the next round.

3.3 Measuring fluctuating observation noise in Goore Games

In order to develop a decentralized BS-TANB based scheme for solving the above problem, whose accuracy does not rely merely on conservative learning, it is crucial that we are able to determine the observation noise, σ_{ob}^2, needed by BS-TANB for its Bayesian computations.

From the perspective of voter V_q, let $Y_q = \sum_{r \neq q} \alpha_r[t]$ be the total number of "Yes" votes found among the $n-1$ votes cast by the other voters ($r \neq q$). According to our Bayesian bandit scheme, each voter V_q, at any given iteration t of the game, cast its vote according to a Bernoulli distribution with success probability $p_q[t] = P(\alpha_q[t] = 1) = P(X_1 > X_0 | \phi_q^t)$ — the probability of voting "Yes". Furthermore, initially, all voters vote "Yes" with probability $p_q[1] = 0.5$, and based on Bayesian computations, gradually shift their probability of voting "Yes" towards either 0 or 1, as learning proceeds. This leads us to design a solution for the case where Y_q is a sum of independent random variables of similar magnitude, in other words, where Y_q is approximately normally distributed for large n, $Y_q \sim N(\mu_F^q, \sigma_F^q)$. Since each term in the summation is Bernoulli distributed, the mean of the sum becomes $\mu_F^q = \sum_{r \neq q} p_r[t]$ while the variance becomes $\sigma_F^{q\,2} = \sum_{r \neq q} p_r[t](1 - p_r[t])$. The above entails that each voter, V_q, essentially decides whether to add an additional "Yes" vote or not to a random sum of yes votes, $Y_q \sim N(\mu_F^q, \sigma_F^q)$. That is, the reward that voter V_q receives when he votes either "Yes" ($\alpha_q[t] = 1$) or "No" ($\alpha_q[t] = 0$), becomes a function $G(\frac{Y_q + \alpha_q[t]}{n})$ governed by the random variable $Y_q \sim N(\mu_F^q, \sigma_F^q)$ as well as the decision α_q of voter V_q.

Thus $E[G(\frac{Y_q + \alpha_q[t]}{n})]$ is the expected reward received by voter V_q when pulling arm $\alpha_q[t]$ and $\mathrm{Var}[G(\frac{Y_q + \alpha_q[t]}{n})]$ is the variance of the reward, which we will refer to as observation noise, σ_{ob}.

Lemma 1 *Let X be a normally distributed random variable, $X \sim N(\mu_F, \sigma_F)$. The expected value $E[G(X)]$ of a deterministic function $G(X) \sim N(\mu_G, \sigma_G)$ of X then becomes*:

$$E\big[G(X)\big] = \frac{1}{\sqrt{2\pi(\sigma_G^2 + \sigma_F^2)}} e^{-\frac{(\mu_G - \mu_F)^2}{2(\sigma_G^2 + \sigma_F^2)}} \tag{2}$$

Proof

$$\begin{aligned} E\big[G(X)\big] &= \int_{-\infty}^{\infty} G(x) f(x; \mu_F, \sigma_F) dx \\ &= \int \frac{1}{2\pi\sigma_f\sigma_g} e^{-(\frac{(x-\mu_F)^2}{2\sigma_F^2} + \frac{(x-\mu_G)^2}{2\sigma_G^2})} dx \\ &= \int_{-\infty}^{\infty} \frac{e^{\gamma}}{2\pi\sigma_f\sigma_g} e^{-(\frac{(x-\mu_{FG})^2}{2\sigma_{FG}^2})} dx \end{aligned} \tag{3}$$

with

$$\gamma = -\frac{(\mu_G - \mu_F)^2}{2(\sigma_G^2 + \sigma_F^2)} \tag{4}$$

$$\mu_{FG} = \frac{\mu_F\sigma_G^2 + \mu_G\sigma_F^2}{\sigma_F^2 + \sigma_G^2} \tag{5}$$

$$\sigma_{FG} = \sqrt{\frac{\sigma_F^2\sigma_G^2}{\sigma_F^2 + \sigma_G^2}} \tag{6}$$

The integral of the resulting Gaussian then becomes:

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{e^{\gamma}}{2\pi\sigma_F\sigma_G} e^{-\left(\frac{(x-\mu_{FG})^2}{2\sigma_{FG}^2}\right)} dx \\
&= \frac{e^{\gamma}}{\sqrt{2\pi}\sigma_F\sigma_G}\sqrt{\frac{\sigma_F^2\sigma_G^2}{\sigma_F^2+\sigma_G^2}} \\
&= \frac{1}{\sqrt{2\pi(\sigma_F^2+\sigma_G^2)}} e^{-\frac{(\mu_G-\mu_F)^2}{2(\sigma_G^2+\sigma_F^2)}} \qquad (7)
\end{aligned}$$

□

Lemma 2 *A deterministic function $G(X) \sim N(\mu_G, \sigma_G)$ of a normally distributed random variable, $X \sim N(\mu_F, \sigma_F)$, has the variance*:

$$\mathrm{Var}\big[G(X)\big] = \frac{e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+2\sigma_F^2}}}{2\pi\sigma_G\sqrt{\sigma_G^2+2\sigma_F^2}} - \frac{e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+\sigma_F^2}}}{2\pi(\sigma_G^2+\sigma_F^2)} \qquad (8)$$

Proof

$$\mathrm{Var}\big[G(X)\big] = E\big[G(X)^2\big] - \big(E\big[G(X)\big]\big)^2 \qquad (9)$$

$$\big(E\big[G(X)\big]\big)^2 = \frac{e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+\sigma_F^2}}}{2\pi(\sigma_G^2+\sigma_F^2)} \qquad (10)$$

$$\begin{aligned}
&E\big[G(X)^2\big] \\
&= \int_{-\infty}^{\infty} G(x)^2 f(x)\, dx \\
&= \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2\pi}\sigma_G} e^{-\frac{1}{2}\left(\frac{(x-\mu_G)^2}{\sigma_G^2}\right)}\right)^2 \frac{1}{\sqrt{2\pi}\sigma_F} e^{-\frac{1}{2}\left(\frac{(x-\mu_F)^2}{\sigma_F^2}\right)} dx \\
&= \int_{-\infty}^{\infty} \frac{e^{\gamma}}{2^{\frac{3}{2}}\pi^{\frac{3}{2}}\sigma_F\sigma_G^2} e^{-\left(\frac{(x-\mu_{FG^2})^2}{2\sigma_{FG^2}^2}\right)} dx \qquad (11)
\end{aligned}$$

with

$$\gamma = -\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+2\sigma_F^2} \qquad (12)$$

$$\mu_{FG^2} = \frac{\mu_F\sigma_G^2 + 2\mu_G\sigma_F^2}{\sigma_G^2+2\sigma_F^2} \qquad (13)$$

$$\sigma_{FG^2} = \sqrt{\frac{\sigma_F^2\sigma_G^2}{\sigma_G^2+2\sigma_F^2}} \qquad (14)$$

The integral of the resulting Gaussian then becomes:

$$\begin{aligned}
&\int_{-\infty}^{\infty} \frac{e^{\gamma}}{2^{\frac{3}{2}}\pi^{\frac{3}{2}}\sigma_F\sigma_G^2} e^{-\left(\frac{(x-\mu_{FG})^2}{2\sigma_{FG}^2}\right)} dx \\
&= \frac{e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+2\sigma_F^2}}}{2\pi\sigma_G\sqrt{\sigma_G^2+2\sigma_F^2}} \qquad (15)
\end{aligned}$$

□

Figure 2 depicts the intricate behavior of $\mathrm{Var}[G(X)]$ when seen as a function of μ_F and σ_F, and when $\mu_G = 0.4$ and $\sigma_G = 0.2$. Since both the mean of G, μ_G, and the mean of F, μ_F, generally are unknown, the latter equation cannot be used directly to guide the bandit based learning. Instead, we consider the maxima of $\mathrm{Var}[G(X)]$ with $\mu_F \in (0, 1)$ being the free variable. By considering the maxima, learning accuracy is prioritized, at the potential cost of reduced learning speed. In the following, we will see that the maximization eliminates both μ_F and μ_G from the equation.

Theorem 1 *The maximum of the variance* $\mathrm{Var}[G(X)]$ *with respect to $\mu_F \in (0, 1)$, for the function $G(X) \sim N(\mu_G, \sigma_G)$, with $X \sim N(\mu_F, \sigma_F)$, is*:

$$\begin{aligned}
&\max_{\mu_F\in(0,1)} \mathrm{Var}\big[G(X)\big] \\
&= \sigma_F^2\big(\sigma_G^2+\sigma_F^2\big)^2 \\
&\quad\times \frac{e^{\frac{\sigma_G^2(\log(\sigma_F(\sigma_G^4+2\sigma_F^2\sigma_G^2+\sigma_F^4)\sigma_G)-\log(\sigma_F\sigma_G^2(\sigma_G^2+2\sigma_F^2)^{\frac{3}{2}}))}{\sigma_F^2}}}{2\pi\sigma_G^2(\sigma_G^2+2\sigma_F^2)^3} \qquad (16)
\end{aligned}$$

Proof We find maxima and minima for $\mathrm{Var}[G(X)]$ with respect to μ_F by solving the following equation:

$$\frac{\partial\,\mathrm{Var}[G(X)]}{\partial\mu_F} = 0 \iff \qquad (17)$$

$$\frac{(\mu_G-\mu_F)e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+2\sigma_F^2}}}{\pi\sigma_G(\sigma_G^2+2\sigma_F^2)^{\frac{3}{2}}} - \frac{(\mu_G-\mu_F)e^{-\frac{(\mu_G-\mu_F)^2}{\sigma_G^2+\sigma_F^2}}}{\pi(\sigma_G^2+\sigma_F^2)^2} = 0 \qquad (18)$$

The above equation has four solutions, with two symmetric maxima in the region of interest $\mu_F \in (0, 1)$, as illustrated in Fig. 2. The first maximum is:

$$\begin{aligned}
\mu_F = \Bigg\{ &\sqrt{\sigma_G^4+3\sigma_F^2\sigma_G^2+2\sigma_F^4} \\
&\times \sqrt{\log\left(\frac{\sqrt{\sigma_G^2+2\sigma_F^2}\,(\sigma_G^3+2\sigma_F^2\sigma_G)}{\sigma_G^4+2\sigma_F^2\sigma_G^2+\sigma_F^4}\right)} \\
&+ \mu_G\sigma_F \Bigg\} \sigma_F^{-1} \qquad (19)
\end{aligned}$$

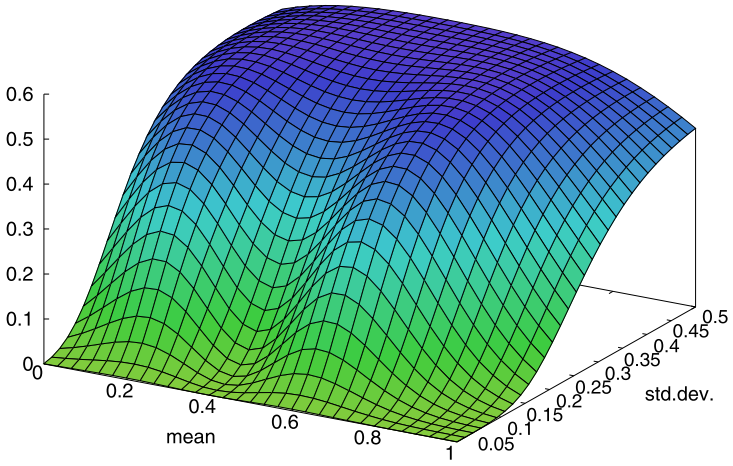

Fig. 2 The variance of $G(X)$, $\mathrm{Var}[G(X)]$, for $\mu_G = 0.4$ and $\sigma_G = 0.2$. The axes depicts the mean μ_F and the standard deviation σ_F, respectively

while the second maximum is:

$$\mu_F = \left\{ \mu_G \sigma_F - \sqrt{\sigma_G^4 + 3\sigma_F^2\sigma_G^2 + 2\sigma_F^4} \times \sqrt{\log\left(\frac{\sqrt{\sigma_G^2 + 2\sigma_F^2}\,(\sigma_G^3 + 2\sigma_F^2\sigma_G)}{\sigma_G^4 + 2\sigma_F^2\sigma_G^2 + \sigma_F^4} \right)} \right\} \sigma_F^{-1} \quad (20)$$

By substituting either Eq. (19) or Eq. (20) into Eq. (8) and simplifying, we see that both μ_F and μ_G have been eliminated from the equation, which completes the proof:

$$\max_{\mu_F \in (0,1)} \mathrm{Var}\big[G(X)\big] = \sigma_F^2 \left(\sigma_G^2 + \sigma_F^2\right)^2 \times \frac{e^{\frac{\sigma_G^2 (\log(\sigma_G^4 + 2\sigma_F^2\sigma_G^2 + \sigma_F^4)) - \log(\sigma_G (\sigma_G^2 + 2\sigma_F^2)^{\frac{3}{2}})}{\sigma_F^2}}}{2\pi\sigma_G^2 (\sigma_G^2 + 2\sigma_F^2)^3} \quad (21)$$

□

A crucial consequence of the results presented in this section is that since σ_F in the above equation can be approximated based on the feedback σ_A from the Referee (see Fig. 1), we can find the worst case observation noise based on Theorem 1. *Thus, we have found a closed form formula for worst case observation noise, σ_{ob}, that each voter can apply adaptively in its Bayesian computations!*

4 Empirical results

In this section we evaluate the ADL-TAB scheme by comparing it with the currently best performing algorithm—the family of *Bayesian* techniques reported in [8]. Based on our comparison with these "reference" algorithms, it should be quite straightforward to also relate the ADL-TAB performance results to the performance of other similar algorithms. We first use artificial data and then data from a simulated sensor network.

4.1 Artificial data

We have conducted numerous experiments using various reward distributions, including a wide range of $G(\lambda)$-functions and a wide range of voters, under varying degrees of observation noise. The full range of empirical results all show the same trend, however, we here report performance on a representative subset of the experiment configurations, involving the 3, 5, and 10 player Goore Game. Performance is measured in terms of *Regret—the difference between the sum of rewards expected after N successive rounds of the GG, and what would have been obtained by always casting the optimal number of "Yes" votes*.

For these experiment configurations, an ensemble of 1000 independent replications with different random number streams was performed to minimize the variance of the reported results. In order to investigate the performance of the schemes under a broad spectrum of environments, we test the schemes using three different representative $G(\lambda)$ functions—one sloped, with optimum close to $\lambda = 0.5$, $G \sim N(0.35, 0.2)$, another one also sloped, but with optimum farther from $\lambda = 0.5$, $G \sim N(0.125, 0.2)$, and finally, one peaked reward function, also with optimum far from $\lambda = 0.5$ (thus, being the most challenging one). In Table 1, Regret is reported after 10, 100, 1000, and 10 000 iterations for both the new *accelerating* scheme and the traditional *static* scheme.

As seen from the table, for all reported configurations, our ADL-TAB scheme not only learns faster initially, but also attains the best regret in the long run. Note that for the two bottom configurations, we use an augmented σ_F, $\widehat{\sigma_F} = c \cdot \sigma_F$, with $c = 1.5$, when the final observation noise σ_{ob} is calculated. Indeed, the constant c can be used to handle

Table 1 Regret after 10, 100, 1000, and 10 000 iterations for 3, 5, and 10 players

Scheme	#Players	Function	10	100	1000	10 000
Accelerating	3	$G \sim N(0.125, 0.1)$	11.56	26.72	30.96	33.17
Static	3	$G \sim N(0.125, 0.1)$	11.63	27.27	34.88	47.20
Accelerating	3	$G \sim N(0.125, 0.2)$	5.26	8.47	10.35	11.09
Static	3	$G \sim N(0.125, 0.2)$	5.28	9.53	15.15	25.15
Accelerating	3	$G \sim N(0.375, 0.2)$	6.62	10.86	11.99	12.63
Static	3	$G \sim N(0.375, 0.2)$	6.73	12.15	14.36	17.72
Accelerating	5	$G \sim N(0.125, 0.1)$	18.37	41.60	51.78	61.65
Static	5	$G \sim N(0.125, 0.1)$	18.28	44.94	58.52	99.49
Accelerating	5	$G \sim N(0.125, 0.2)$	6.92	12.94	22.99	60.80
Static	5	$G \sim N(0.125, 0.2)$	7.01	15.39	32.94	69.86
Accelerating	5	$G \sim N(0.375, 0.2)$	6.12	20.47	22.70	25.75
Static	5	$G \sim N(0.375, 0.2)$	6.16	24.24	30.11	35.74
Accelerating	10	$G \sim N(0.125, 0.1)$	32.81	93.82	133.65	443.7
Static	10	$G \sim N(0.125, 0.1)$	32.84	99.27	143.67	549.8
Accelerating	10	$G \sim N(0.125, 0.2)$	10.19	19.57	39.58	110.53
Static	10	$G \sim N(0.125, 0.2)$	10.21	22.57	56.91	167.09
Accelerating	10	$G \sim N(0.375, 0.2)$	4.40	31.20	113.42	116.65
Static	10	$G \sim N(0.375, 0.2)$	4.41	32.03	163.40	197.31

Table 2 Performance with σ_F augmented, $\widehat{\sigma_F} = c \cdot \sigma_F$ (10 players, $G \sim N(0.1, 0.1)$, $\sigma_W = 0.1$)

Scheme/c	1.0	1.25	1.5	1.75
Accelerating	1030.0	684.7	444.7	408.7
Static	965.6	624.3	550.4	414.2

the non-stationarity arising as the number of voters grows, as demonstrated in Table 2.

Since ADL-TAB applies the standard deviation σ_G of the reward function $G(\lambda)$ to find overall observation variance, it is interesting to see how robust the scheme is to distortion of σ_G. As summarized in Table 3, setting σ_G too low is better than setting it too high in the present setting. Indeed, performance improves slightly with a lower σ_G.

Note that the above reported performance gap is reduced with the level of white noise added to G, as shown in Table 4. As the variance of the white noise raises to extreme values, the white noise dominates the overall observation noise, rendering the variance introduced by the voters insignificant. However, for realistic degrees of white noise, as also seen from the table, ADL-TAB clearly outperforms the static BS-TANB scheme.

Thus, based on our empirical results on artificial data, we conclude that ADL-TAB is the superior choice for the GG, both when σ_G is known or slightly distorted, providing significantly better performance in all experiment configurations.

4.2 QoS control in wireless sensor networks

As mentioned in the introduction, the GG can be used for QoS control in wireless sensor networks. A scenario of particular interest is randomly deployed networks, whose applications include environmental monitoring and battlefield surveillance & reconnaissance [3]. An additional complexity for QoS control under such settings is sensor break down. Due to the random deployment of sensors, and due to the nature of typical applications, it is often infeasible to track down and repair broken sensors. Instead, batches of new sensors are deployed to replace broken ones, e.g., by air drop. As a result, the population of sensors are dynamically changing over time, in a stochastic manner.

To stress the ADL-TAB scheme under particularly challenging conditions, we have thus simulated the latter kind of dynamic environments. In brief, the simulated environment starts out with ten randomly deployed sensors. As in [15], both the lifetime of each sensor and the rate of deployment is governed by the same exponential distribution. Therefore the total number of operative sensors will fluctuate, but remain constant on average.

The QoS function used here is $G \sim N(0.375, 0.2)$, which means that optimally 37.5 % of the sensors should be active at any given instant. A white noise of $\sigma_w = 0.3$ is applied to feedback. This configuration was executed for 10 000 hours, with the average regret for 2000 sample runs summarized in Table 5 for different birth/death rates.

From the table it is clear that ADL-TAB obtains significantly lower regret than the static BS-TANB scheme in these

Table 3 Performance with distorted $\widehat{\sigma}_G$ given to ADL-TAB (10 players, $G \sim N(0.125, 0.2)$, $\sigma_W = 0.1$)

$\widehat{\sigma}_G$	$0.85 \cdot \sigma_G$	$0.90 \cdot \sigma_G$	$0.95 \cdot \sigma_G$	$1.0 \cdot \sigma_G$	$1.05 \cdot \sigma_G$	$1.10 \cdot \sigma_G$	$1.15 \cdot \sigma_G$
Regret	74.4	75.7	90.9	123.5	162.9	194.4	237.5

Table 4 Performance with varying degrees of white noise $N(0, \sigma_W)$ (10 players, $G \sim N(0.375, 0.2)$)

Scheme/σ_W	0.01	0.05	0.1	0.5	1.0	5.0
Accelerating	56.6	54.8	61.1	123.4	315.8	2012.0
Static	120.5	121.4	121.6	184.4	371.7	2013.3

Table 5 Regret after 10, 100, 1000, and 10 000 hours

Scheme	Birth/Death Rate	10	100	1000	10 000
Accelerating	100	4.47	31.50	180.43	1055.23
Static	100	4.48	31.45	194.14	3003.87
Accelerating	50	4.47	31.65	195.84	1268.32
Static	50	4.45	31.58	203.19	4734.49
Accelerating	12.5	4.49	32.71	219.34	1270.76
Static	12.5	4.39	32.04	357.60	9589.84

Table 6 Standard deviation of regret after 10 000 hours

Scheme	Birth/Death Rate = 100	Birth/Death Rate = 50	Birth/Death Rate = 12.5
Accelerating	$\sigma_{100} = 254.6$	$\sigma_{50} = 229.0$	$\sigma_{12.5} = 206.5$
Static	$\sigma_{100} = 1561.5$	$\sigma_{50} = 1628.9$	$\sigma_{12.5} = 1393.7$

scenarios too. Indeed, the performance benefit is increasing with faster replacement of sensors. This can be explained by the ability of a newly placed sensor, that is governed by the ADL-TAB scheme, to perceive any learning stability already established among the operating sensors, which in turn will allow the sensor to accelerate its transition from exploration to exploitation.

Another effect worth noting is the stability of ADL-TAB. That is, the standard deviation of the sample runs for ADL-TAB after 10 000 iterations is much lower than for the static scheme, as can be seen from Table 6. This performance robustness provided by ADL-TAB can be explained by the acceleration that takes place to reduce exploration.

5 Conclusion and further work

In this paper we proposed a novel scheme, ADL-TAB, for decentralized decision making based on the Goore Game. Theoretical results concerning the variance of the observations made by each individual decision maker enabled us to accelerate learning as exploration turns into exploitation. Indeed, our empirical results demonstrated that the accelerated learning improves both learning accuracy and speed, outperforming state-of-the-art Goore Game solution schemes, both when using artificial data and when using data from a wireless sensor network simulation.

As further work, we intend to study how the Kalman filter can be incorporated into ADL-TAB, so that non-stationary behavior can be modeled and addressed in a principled manner. We are also currently investigating how the present result can be extended to other classes of decentralized decision making problems. Finally, we believe this avenue of research can lead to enhancements in application areas such as decentralized task scheduling, processing pipeline optimization, and resource allocation.

References

1. Bouhmala N, Granmo O-C (2010) Combining finite learning automata with GSAT for the satisfiability problem. Eng Appl Artif Intell 23:715–726
2. Cao YU, Fukunaga AS, Kahng A (1997) Cooperative mobile robotics: antecedents and directions. Auton Robots 4(1):7–27
3. Chen D, Varshney PK (2004) QoS support in wireless sensor networks: a survey. In: The 2004 international conference on wireless networks (ICWN 2004)
4. Dimitrakakis C (2006) Nearly optimal exploration-exploitation decision thresholds. In: Proceedings of the 16th international conference on artificial neural networks (ICANN 2006). Lecture notes in computer science. Springer, Berlin, pp 850–859
5. Gelly S, Wang Y (2006) Exploration exploitation in go: UCT for Monte-Carlo go. In: Proceedings of NIPS-2006, NIPS
6. Google. Google website optimizer: http://www.google.com/websiteoptimizer. Retrieved November 2011

7. Graepel T, Candela JQ, Borchert T, Herbrich R (2010) Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's bing search engine. In: Proceedings of the twenty-seventh international conference on machine learning (ICML-10), p 1320
8. Granmo O-C (2010) Solving two-armed Bernoulli bandit problems using a Bayesian learning automaton. Int J Intell Comput Cybern 3(2):207–234
9. Granmo O-C, Berg S (2010) Solving non-stationary bandit problems by random sampling from Sibling Kalman filters. In: Proceedings of the twenty third international conference on industrial, engineering, and other applications of applied intelligent systems (IEA-AIE 2010). Springer, Berlin, pp 199–208
10. Granmo O-C, Bouhmala N (2007) Solving the satisfiability problem using finite learning automata. Int J Comput Sci Appl 4(3):15–29
11. Granmo O-C, Oommen BJ, Myrer SA, Olsen MG (2007) Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation. IEEE Trans Syst Man Cybern, Part B, Cybern 37(1):166–175
12. Granmo O-C, Oommen BJ (2010) Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. IEEE Trans Comput 59(4):545–560
13. Gupta N, Granmo O-C, Agrawala A (2011) Successive reduction of arms in multi-armed bandits. In: Proceedings of the thirty-first SGAI international conference on artificial intelligence (SGAI 2011). Springer, Berlin
14. Gupta N, Granmo O-C, Agrawala A (2011) Thompson sampling for dynamic multi-armed bandits. In: Proceedings of the tenth international conference on machine learning and applications (ICMLA'11). IEEE, New York
15. Iyer R, Kleinrock L (2003) QoS control for sensor networks. In: IEEE international conference on communications, vol 1, pp 517–521
16. May BC, Korda N, Lee A, Leslie DS (2011) Optimistic Bayesian sampling in contextual-bandit problems. Technical report, Statistics Group, Department of Mathematics, University of Bristol
17. Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice Hall, New York
18. Oommen BJ, Granmo O-C (2008) Learning automata-based solutions to the Goore game and its applications. In: Game theory: strategies, equilibria, and theorems. Nova Science Publishers, New York
19. Oommen BJ, Misra S, Granmo O-C (2007) Routing bandwidth guaranteed paths in MPLS traffic engineering: a multiple race track learning approach. IEEE Trans Comput 56(7):959–976
20. Scott SL (2010) A modern Bayesian look at the multi-armed bandit. Applied Stochastic Models in Business and Industry (26):639–658
21. Thompson WR (1933) On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25:285–294
22. Tsetlin ML (1973) Automaton theory and modeling of biological systems. Academic Press, San Diego
23. Tung B, Kleinrock L (1996) Using finite state automata to produce self-optimization and self-control. IEEE Trans Parallel Distrib Syst 7(4):47–61
24. Wang T, Lizotte D, Bowling M, Scuurmans D (2005) Bayesian sparse sampling for on-line reward optimization. In: Proceedings of the 22nd international conference on machine learning, pp 956–963

Ole-Christoffer Granmo was born in Porsgrunn, Norway. He obtained his M.Sc. in 1999 and the Ph.D. degree in 2004, both from the University of Oslo, Norway. He is currently a Professor in the Department of ICT, University of Agder, Norway. His research interests include Intelligent Systems, Stochastic Modelling and Inference, Machine Learning, Pattern Recognition, Learning Automata, Distributed Computing, and Surveillance and Monitoring. He is the author of more than 65 refereed journal and conference publications.

Sondre Glimsdal was born in Oslo, Norway. He is currently pursuing his Ph.D. degree at the Department of ICT, University of Agder, Norway. His research interests include Machine Learning, Stochastic Modeling and Inference, and Computational Linguistics.