How Expert Confidence Can Improve Collective Decision-Making in Contextual Multi-Armed Bandit Problems

Axel Abels[1,2][0000−0003−2784−8653], Tom Lenaerts[1,2][0000−0003−3645−1455], Vito Trianni[3][0000−0002−9114−8486], and Ann Nowé[2][0000−0001−6346−4564]

[1] Machine Learning Group, Université Libre de Bruxelles, Brussels, Belgium
[2] AI lab, Vrije Universiteit Brussel, Brussels, Belgium
[3] Laboratory of Autonomous Robotics and Artificial Life, Instituto di Scienze e Tecnologie della Cognizione, Rome, Italy

Abstract. In collective decision-making (CDM) a group of experts with a shared set of values and a common goal must combine their knowledge to make a collectively optimal decision. Whereas existing research on CDM primarily focuses on making binary decisions, we focus here on CDM applied to solving contextual multi-armed bandit (CMAB) problems, where the goal is to exploit contextual information to select the best arm among a set. To address the limiting assumptions of prior work, we introduce confidence estimates and propose a novel approach to deciding with expert advice which can take advantage of these estimates. We further show that, when confidence estimates are imperfect, the proposed approach is more robust than the classical confidence-weighted majority vote.

Keywords: Deciding with expert advice; Contextual Bandits; Confidence; Noise; Collective Decision-Making

1 Introduction

In CDM, a group (e.g. experts) aims to find collectively the best solution among a set of alternatives for a given problem [4,3,12]. Example applications are peer review processes, wherein the decisions of multiple expert reviewers are combined to ensure that the qualitatively best works are selected from a pool of submissions [13], and medical diagnostics, wherein a group of medical experts must decide on the best treatments for patients [14].

The quality of the decision produced in CDM depends strongly on the expertise of each person involved in the process. Participants in the CDM process should thus have the opportunity to provide an estimate of the confidence they have in their advice, as is now mandatory in many paper reviewing procedures, for instance. When confidence information is available, it should be considered by a CDM system to enhance decision accuracy based on expert advice [2].

The Exponential-weight Algorithm for Exploration and Exploitation using Expert advice (EXP4 for short [2]) is a state-of-the-art CDM solver that learns how to integrate the advice of a set of stationary experts, which can either be trained predictors or human experts. Although it is considered to be one of the key approaches to automatically infer collective decisions based on expert advice, it does not consider expert confidence in the learning process. This is a limiting assumption, as experts are likely to have expertise only for the region of the problem on which they were trained. In general, it is safe to assume experts will perform relatively well in settings for which they have prior experience. Querying human experts for (honest) confidence estimates about their given advice has been shown to improve performance on visual perception and value estimation tasks [3,12]. The dominant approach in these tasks is to use confidence-weighted majority votes [12]. The higher an expert's confidence, the higher its opinion (i.e., advice) will be weighted when aggregating.

We hypothesize that EXP4's performance can be improved by including such confidence information. We show how EXP4 can be adapted and under which conditions this hypothesis holds, considering two types of confidence, i.e., a global/non-contextual and a contextual one. Yet, as one cannot assume that an expert always provides correct confidence estimates, we also consider how imperfect confidence estimates affect CDM, revealing thus the robustness of the approach to noise.

Our analysis is performed in the framework of contextual multi-armed bandits (CMAB) [11,21], where each arm is identified by a combination of contextual features, associated in turn to a context-dependent reward retrieved from an a-priori unknown function. Most problems that can be solved through CDM naturally lend themselves to a formalization through CMABs. In medical decision-making [19] for example, the set of possible patient-treatment pairs is the set of arms and the contexts are pairwise patient-treatment characteristics (e.g. patient symptoms, the results of medical tests, treatment properties). The aim of every medical expert is to select the most appropriate treatment (i.e., the best arm) given the set of contexts and their associated information. Similarly, in a review process the best submission(s) (i.e., the best arm(s)) must be selected based on their context. Different from classic CMAB solving, CDM requires all experts to solve the same problem given their knowledge and combine their proposals/opinions in a way that maximizes expected outcome (e.g., maximizing the overall quality of accepted submissions by taking into account multiple reviewers). As the exact performance of a set of experts can be difficult to estimate a-priori, it is useful to learn to exploit their knowledge purposefully. EXP4 performs exactly this task.

Prior to reporting the methods used to generate the results of this paper, the next section provides first of all the background knowledge on CMAB and making decisions based on expert advice. The final section provides a discussion on the generated results and the conclusions that can be drawn. Note that within this work, we do not yet consider human experts. Experts are modelled as known

stochastic contextual bandit algorithms [21], having different degrees of expertise on the CMAB problems for which they need to provide advice.

2 Background

2.1 Contextual Multi-Armed Bandits

Formally, a multi-armed bandit (MAB) is characterized by a set of K arms identified by numbers $1, ..., K$ and a function f that maps each arm to a reward distribution. In MABs, the aim of a learner is typically to identify the best arm $k^* = \arg\max_{k \in K} \mathbb{E}[f(k)]$. Because f is unknown, exploration is required to identify the best arm(s). State-of-the-art MAB algorithms balance exploration of uncertain arms and exploitation of likely optimal arms to minimize regret [1].

While MABs are useful as models of fundamental repeated decision-making, they are limited in their applicability as they assume that arms are only characterized by their identifier. In real world problems, decisions are typically made based on additional information about the problem or the arms. The addition of such information transforms a MAB into a contextual MAB (or CMAB). We follow here the formalism for stochastic contextual bandits specified in [21], which associates to each individual arm a context vector. At time t, the set of all arms in a CMAB is characterized by K time-dependent d-dimensional context vectors $\{\vec{x}_{1,t}, ..., \vec{x}_{K,t}\} \in \mathbb{R}^{K \times d}$. A CMAB also possesses a fixed but unknown scalarization function f which maps each arm context to a reward, $f : \mathbb{R}^d \to [0, 1]$. A policy $\pi : \mathbb{R}^{K \times d} \to [0, 1]^K$ maps all K arm contexts to a probability distribution according to which the learner chooses an arm, i.e., $k_t \sim \pi_t(\mathbf{x}_t)$. In CMABs, the regret over T rounds of a learner pulling arm $k_t \sim \pi_t(\mathbf{x}_t)$ by following its policy π_t at each time-step t is the sum of reward differences between the pulled arm given the context at time t, i.e., $f(\vec{x}_{k_t \sim \pi_t, t})$, and the best arm at each time step t, i.e., $\max_{k=1}^{K} f(\vec{x}_{k,t})$;

$$R_T^\pi = \sum_{t=1}^{T} \big(\max_{k=1}^{K} f(\vec{x}_{k,t}) - f(\vec{x}_{k_t \sim \pi_t, t}) \big) \tag{1}$$

The aim of any learner is to minimize the regret R_T^π.

Most research on tackling CMABs focuses on approximating the underlying distribution function f by balancing exploration and exploitation [6,18]. The complexity of f can however be such that it either cannot be learnt by a dedicated CMAB algorithm, or it would be prohibitively expensive to do so from scratch [16]. Alternatively, f can be relatively straightforward to approximate, but the contexts cannot be meaningfully converted into datapoints. In natural language processing for example, meaningfully capturing context can be hard to accomplish by artificial systems [9]. In such complex cases, experts can provide the knowledge required to select the appropriate arms. Indeed, experts can reduce the complex contextual information into an advice about each arm that can be exploited for the arm choice. To this end, methods such as EXP4 – described in the following section – learn to identify the best performing experts.

2.2 EXP4: deciding with expert advice

When deciding with expert advice [2], a learner can query a set of N stationary experts for their advice on which arm to select, with each expert having access to every context vector $\vec{x}_{k,t}$ associated with those arms. Each expert n has some expertise about these contexts which it uses to express advice $\xi^n_{k,t}$ for each arm k (i.e., the probability of pulling arm k). Such experts can be complex, time-intensive algorithms trained in advance on similar data as well as human experts that are able to infer relations beyond the reach of current algorithmic approaches (some human intuitions for example are hard to translate into an algorithm), or a mix of both. In this setting, the learner has no access to the information about the best arm, and should rely on the experts. Informally, a learner must identify which experts' advice to follow. Although one can again use regret minimization in this setting, the regret is now relative to the best expert and not the best arm, hence the new formula for regret ($R^{'\pi}_T$) is now:

$$R^{'\pi}_T = \max_{n=1}^{N} \sum_{t=1}^{T} \left(f(\vec{x}_{k_t \sim \vec{\xi}^{\,n}_t, t}) - f(\vec{x}_{k_t \sim \pi_t, t}) \right) \tag{2}$$

By combining the knowledge of multiple experts, algorithms can surpass the performance of the best single expert, generating a negative regret value.

Algorithm 1 outlines the problem of deciding with expert advice. Solvers for this problem provide a concrete implementation of (i) the policy which maps advice to selected arms (Line 5), and (ii) how the policy is updated based on the observed reward (Line 6). EXP4 [2] performs these tasks by maintaining a weight w^n_t for each expert which it uses to compute a weighted average of expert advice as follows:

$$p_{k,t} = \sum_{n \in N} \frac{exp(w^n_t)}{\sum_{n' \in N} exp(w^{n'}_t)} \xi^n_{k,t} \tag{3}$$

Based on this weighted average, the learner pulls an arm k_t and collects a reward $r_t \sim f(\vec{x}_{k_t,t})$. Weights are updated based on the collected reward, the expert's

Algorithm 1 Deciding with expert advice

Require: N experts, contextual bandit with distribution function f and K arms of dimensionality d, learner with policy $\pi : [0,1]^{K \times N} \rightarrow [0,1]^K$ which maps advice to a probability distribution over the arms

1: Each expert n has experience on P^n contexts sampled from its expertise region.
2: **for** $t = 1, 2, ..., T$ **do**
3: Observe context matrix $\mathbf{x}_t = \{\vec{x}_{1,t}, ..., \vec{x}_{K,t}\}$ and share it with experts
4: Get expert advice vectors $\boldsymbol{\xi}_t = \{\vec{\xi}^{\,1}_t, ..., \vec{\xi}^{\,N}_t\}$
5: Pull arm $k_t \sim \pi_t(\boldsymbol{\xi}_t)$ and collect resulting reward r_t
6: $\pi_{t+1} = update(\pi_t, \boldsymbol{\xi}_t, k_t, r_t)$ ▷ Learner updates its policy based on the received advice, the pulled arm and the observed reward

advice and the aggregated probability, as follows:

$$w_{t+1}^n = w_t^n + \gamma r_t \xi_{k_t,t}^n \frac{1}{p_{k_t,t}}, \tag{4}$$

where γ is the learning rate. The factor $\frac{1}{p_{k_t,t}}$ is included to un-bias the estimator by increasing the weight of arms that were unlikely to be pulled. However, because of the factor's high variance, EXP4 is prone to instability [5]. EXP4.P [5] (see Algorithm 2), a later improvement on EXP4, reduces this instability by including an additional term in the weight update:

$$\hat{v}_t^n = \sum_{k=1}^{K} \xi_{k,t}^n / p_{k,t} \tag{5}$$

$$w_{t+1}^n = w_t^n + \frac{\gamma}{2}\left(r_t \xi_{k_t,t}^n \frac{1}{p_{k_t,t}} + \hat{v}_t^n \sqrt{\frac{\ln(N/\delta)}{KT}} \right) \tag{6}$$

Intuitively, the term in (5) measures how much each expert disagrees with the aggregated probabilities. For any given expert n this term will be large when there is an arm k such that $\xi_{k,t}^n >> p_{k,t}$ (in other words, if expert n disagrees with the aggregated probability $p_{k,t}$). The factor $\sqrt{\frac{\ln(N/\delta)}{KT}}$ weighs this additional term in function of the number of experts (N), the number of arms (K), the number of time-steps (T) and the parameter δ.

Neither EXP4 nor EXP4.P make use of contextual information when updating weights. As a consequence, the weight of an expert is uniform over the complete context space, which limits the usefulness of EXP4.P when expertise is localized (e.g., when experts provide good advice for subsets of the context-space, but do not show significant differences in performance when the whole context-space is considered).

2.3 Weighted Majority Vote

To evaluate the results obtained for our adaptations of EXP4, we consider as a baseline a straightforward aggregation method consisting in computing a weighted average of all advices and acting greedily on this average, i.e., the Weighted Majority Vote (WMV) algorithm [12]. The weights used in the WMV can for example be based on experts' expressed confidence, with higher confidence resulting in a higher impact on the weighted aggregation. In binary decision-making the usage of confidence-based weights is optimal [12]. Building on that result we propose a rudimentary extensions of the weighted majority vote for the n-ary case by computing the weighted average of the advice vectors. If the confidence $c_{k,t}^n$ of an expert n at time t about context $\vec{x}_k$ is expressed in the range $[0, 1]$ wherein confidences of $1, 0.5$, and 0 correspond respectively to a perfect expert, a random expert, and the worst possible expert, we can weigh advice as follows:

$$\sum_{n \in N} \ln\left(\frac{c_{k,t}^n}{1 - c_{k,t}^n}\right) \xi_{k,t}^n \tag{7}$$

Algorithm 2 Description of the EXP4.P algorithm

Require: $\delta > 0$

1: Define $\gamma = \sqrt{\frac{\ln N}{KT}}$, set $w_{1,i} = 1$ for $i = 1, ..., N$.
2: **for** $t = 1, 2, ...T$ **do**
3: Get expert advice vectors $\boldsymbol{\xi}_t = \{\vec{\xi}^1_t, ..., \vec{\xi}^N_t\}$, each vector is of size K.
4: **for** $k = 1, 2, ..., K$ **do** ▷ compute weighted average

$$p_{k,t} = \sum_{n \in N} \frac{exp(w^n_t)}{\sum_{n' \in N} exp(w^{n'}_t)} \xi^n_{k,t}$$

5: Draw arm k_t according to $\vec{p}_t$, and receive reward r_t.
6: **for** $n = 1, ..., N$ **do** ▷ Update weights

$$\hat{y}^n_t = \xi^n_{t,k_t} \cdot r_t / p_{k_t,t}$$

$$\hat{v}^n_t = \sum_{k=1}^{K} \xi^n_{k,t} / p_{k,t}$$

$$w^n_{t+1} = w^n_t + \frac{\gamma}{2} \left(\hat{y}^n_t + \hat{v}^n_t \sqrt{\frac{\ln(N/\delta)}{KT}} \right)$$

Given the weighted value, WMV greedily selects the arm with the highest resulting value. The term $\ln\left(\frac{c^n_{k,t}}{1-c^n_{k,t}}\right)$ provides optimal weights for the binary case [12], because experts with worse-than-random confidence ($c^n_{k,t} < 0.5$) are weighted negatively, and experts with random confidence ($c^n_{k,t} = 0.5$) are ignored. A further discussion on how the confidence estimates can be obtained for CMABs is given in Section 3.2. It should nevertheless be clear that this method heavily relies on accurate confidence estimates. To address this drawback we propose an expansion of EXP4.P in Section 3.4 which can take advantage of accurate confidence estimates but is robust to inaccurate values.

3 Implementation

In this section we introduce our extensions to EXP4.P as well as the two forms of confidence that will be used, i.e., contextual and non-contextual confidence, and how they intuitively can be incorporated in the algorithm. We also hypothesize that by using as advice the expected value obtained from pulling an arm as opposed to the probability distribution over the arms an additional boost in decision-making performance can be obtained. As already mentioned in the introduction, experts are considered here as instances of known stochastic contextual bandit algorithms [21].

3.1 Value Advice

When considering localized expertise in a CMAB problem, experts can be knowledgeable about a subset of the active contexts but agnostic about the remaining contexts. To provide probability advice, experts must make assumptions about unknown arms which affect the probability distribution over all the arms. Previous work on deciding with expert advice has been limited to advice in the form of probability distributions (see Section 2.2). In contrast, if advice consists of one value estimate per arm, the uncertainty about some contexts does not affect the given advice for the known arms. This is the main motivation behind our first contribution: the introduction of value advice and a straightforward extension of EXP4.P to this setting.

Concretely, in the case of value advice, if $\tilde{f}_t^n$ is expert n's approximation of f at time t, then its advice for context vector $\vec{x}_{k,t}^n$ at time t is: $\xi_{k,t}^n = \tilde{f}_t^n(\vec{x}_{k,t}^n)$

In the original algorithm, when using probability advice, EXP4.P computes the following unbiased gain for each expert which is used to increment expert weights: $\hat{y}_t^n = \xi_{k,t}^n r_t / p_{k,t}$, with $p_{k,t}$ the probability of pulling arm k at time t. When dealing with value advice we hypothesize that an expert with low prediction errors will have low regret and use the negation of (unbiased) squared error between the expert's predicted value and the outcome: $\hat{y}_t^n = -(\xi_{k,t}^n - r_t)^2 / p_{k,t}$. This value iteratively increases the relative weight of the experts with the lowest mean square error. While value advice prevents the spread of uncertainty to all arms, the expression of confidence as we explore in the following section can further help the CDM algorithm in its decision-making.

3.2 Confidence in Contextual Multi-Armed Bandits

In what follows we propose two measures of confidence, i.e., non-contextual confidence, which is analogous to the accuracy measure used for weighted majority votes in binary classification [12], and contextual confidence, in which experts provide confidence for active contexts.

Non-contextual confidence The *non-contextual confidence* takes inspiration from the accuracy of experts in binary classification problems [12], wherein a perfect expert has confidence 1, a random expert has confidence 0.5 and the worst possible expert has confidence 0. Given an expert's confidence c^n (the probability that an expert n's advice is the right one), their advice can be optimally weighted by $ln(\frac{c^n}{1-c^n})$. The goal is to derive a similar measure for CMABS in CDM.

To this end one needs to derive a confidence for n-ary decisions similar to the binary classification accuracy, with similar mappings as for the binary classification, i.e., a measure of 1 for perfect performance or optimal policy π^*, a measure of 0 for the worst possible performance or worst policy π^-, and 0.5 for the performance of a random agent or uniform policy $\pi^{\mathcal{U}}$). Because rewards are not all-or-nothing as they are in binary classification, we define confidence in terms of regret. Recall that R_T^π (or $R_T^{'\pi}$) is the regret of policy π over T steps (see Equation 1 and Equation 2). Given that policy, we derive its confidence

over T steps by normalizing its regret with regards to the worst possible regret ($R_T^{\pi^-}$) and random regret ($R_T^{\pi^{\mathcal{U}}}$) as:

$$c_T^\pi = \left(\frac{R_T^{\pi^-} - R_T^\pi}{R_T^{\pi^-}} \right)^\rho \tag{8}$$

where $\rho = log(0.5)/log(\frac{R_T^{\pi^-} - R_T^{\pi^{\mathcal{U}}}}{R_T^{\pi^-}})$ scales the regret such that a random policy is assigned a confidence of 0.5. Analogously to the binary classification setting, this confidence measure has the following properties: (i) $c_T^\pi \in [0, 1]$ for every policy π, (ii) $c_T^\pi < c_T^{\pi'} \Leftrightarrow R_T^\pi > R_T^{\pi'}$, (iii) $c_T^{\pi^{\mathcal{U}}} = 0.5$, (iv) $c_T^{\pi^*} = 1$ and v) $c_T^{\pi^-} = 0$.

Note that, while determining the exact confidence of an expert a priori would be impossible, a reasonable assumption is that a confidence estimate is available based on prior experiences. Such (approximate) confidence measure reflects how confident a participant can be about its advice. This provides the aggregating algorithm with information on how to weigh expert advice. Such confidence is however limited in that it only captures a general trend rather than decision-specific confidence. A more appropriate form of confidence, would depend on the decision that needs to be made rather than on a sequence of decisions.

Contextual confidence It is reasonable to assume that expertise is not uniformly distributed over the context-space. Global confidence measures like the one discussed earlier fail to capture such a heterogeneous expertise distribution. When confidence can be considered on a case-by-case basis, i.e., based on the contexts for which a decision must be made, we refer to it as *contextual confidence*. Concretely, every time an expert n gives an advice $\xi_{k,t}^n$ for an arm k, she can also express a confidence measure $c_{k,t}^n$ related to that advice. We assume that this confidence is on average correlated to the expert's performance.

Confidences are expressed for the advice on the current context $\mathbf{x}_{k,t}$, which is, intuitively, how likely it is that following the expert's advice for that arm will help the learner pick the best arms. An expert's lack of confidence might reflect that the expert has spent little time solving problems in that region of the context space (e.g., a patient showing symptoms the doctor is not familiar with might reduce the doctor's confidence in which treatment is appropriate). Contextual confidence provides a convenient way of modelling for example a general practitioner as an expert whose prior experiences are spread out over most of the context-space as opposed to a specialist (e.g., an ophthalmologist) whose prior experience is focused on a small region of the context-space. The former will have a moderate confidence over most of the context space, the latter will have high confidence for that specific region on which she was trained.

Note that confidence is not always accurate. Humans also have a tendency to overestimate their confidence [7], and over-fitting is a well-known problem in algorithmic prediction wherein performance on training data (on which one might base confidence) does not translate to performance on the test data [15,20]. The noise model presented in Section 3.3 partially addresses this drawback.

Although human experts can readily provide contextual confidence our experimental results focus on CDM with CMAB-based algorithmic experts. the next section discusses how contextual confidence can be derived for some of the more common CMAB algorithms [21].

Deriving confidence from artificial experts CMAB algorithms make (explicit) use of an additional term that drives exploration. At time t, these algorithms generally select the arm k that maximizes $\tilde{f}_t(\vec{x}_{k,t}) + \alpha\sigma_{k,t}$, where $\tilde{f}_t$ is the learner's current approximation of f, $\sigma_{k,t}$ is the uncertainty around context $\vec{x}_{k,t}$ and α weighs this exploratory term. Thus, the higher the uncertainty about a context, the higher the exploratory drive. This measure of uncertainty is naturally linked to a lack of confidence as one can consider that low uncertainty is correlated with a high accuracy and can be used as a proxy for perfect confidence.

Hence, since experts here correspond to CMAB algorithms (and more specifically KernelUCB [18]), we can exploit this uncertainty to obtain confidence estimates. In essence, when the uncertainty $\sigma_{k,t}$ around a context $\vec{x}_{k,t}$ is large, the expert's confidence should be low. Conversely, a small uncertainty should result in a high confidence. A high level of uncertainty for all contexts indicates an overall lack of knowledge. In such cases, performance is equivalent to a random policy and a confidence of 0.5 is appropriate. In all other cases more information than random is available, and the resulting confidence should reflect this by being superior to 0.5.

Taking this into account, if $\sigma^n_{k,t} \in [0, 1]$ is expert n's uncertainty for context vector $\vec{x}_{k,t}$, the expert's contextual confidence is defined as $c^n_{k,t} = 0.5 + \frac{1}{2}(1 - \sigma^n_{k,t})$.

In the case of probability advice, the probabilities are expressed in function of multiple contexts. As a result, the expressed confidence should be a function of the confidences of these multiple contexts. When we need to combine multiple confidences we will use the geometric mean of individual confidences.

3.3 Noise Model

Experimental results have shown that humans have a tendency to show bias in self-reported confidence estimates [7]. Whether it is because of past experience which is no longer relevant or simply a tendency to over or under-estimate one's confidence, an estimated confidence which diverges from the expert's actual confidence can be counter-productive. It is therefore desirable to have CDM methods that are robust to the presence of noise in confidence estimates. To simulate the presence of imperfect confidence we propose the following noise model parametrized by a noise level η. Given an expert with true confidence c^n_T, we sample her noisy confidence from the Beta distribution $\beta(1+c^n_T/\eta, 1+(1-c^n_T)/\eta)$, which ensures one remains in the $[0, 1]$ interval. As Figure 1 (left panel) illustrates, the lower the noise level, the more likely it is that the expert's sampled confidence equals its true confidence.

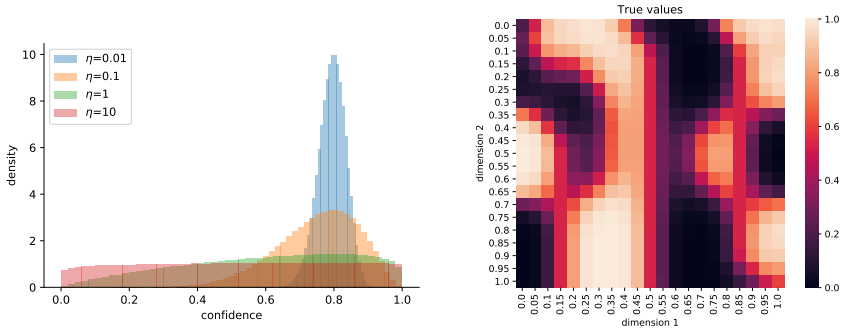

Fig. 1. Confidence distributions and reward values in 2-dimensional contextual bandit. (left) Confidence distribution in function of noise (η) levels for $c_T^e = 0.8$. Given an expert with true confidence c_T^n, the expert's noisy confidence is sampled from the beta distribution $\beta(1 + c_T^n/\eta, 1 + (1 - c_T^n)/\eta)$. As $\eta \to 0$ the sampled values converge to c_T^n. As $\eta \to \infty$, the confidence distribution converges to a uniform distribution over $[0, 1]$. (right) Example of the truth values of a 2-dimensional contextual bandit. Brighter values have a higher expected reward. This landscape is generated using the Perlin noise procedure described in Section 4.1.

3.4 Confidence-weighted, value-based EXP4.P

Our implementation starts from EXP4.P [5], detailed in Algorithm 2, which builds on the assumption that expertise is equal over the complete context-space. To deal with more localized expertise with the help of confidence estimates we propose EXP4.P+CON.

In its original description, EXP4.P assumes no prior knowledge about the performance of experts. Hence, weights are initialized uniformly. However, if a confidence estimate $c_{k,t}^n$ is available, we modify Line 4 in Algorithm 2 to integrate at each time-step the confidence estimate in the aggregation rule (the denominator ensures the weights add up to 1):

$$\sum_{n \in N} \frac{exp(w_{n,t}) c_{k,t}^n / (1 - c_{k,t}^n)}{\sum_{n' \in N} exp(w_{n',t}) c_{k,t}^{n'} / (1 - c_{k,t}^{n'})} \xi_{k,t}^n \tag{9}$$

4 Experiments

In what follows the experimental settings are defined. First, the performance of the different advice types, i.e., probability advice and value advice, is evaluated. Second, the effect of (non-)contextual confidence estimates on performance are tested against the scenario without confidence. Third, as accurate confidence estimates are not always available, the impact of noisy confidence estimates is tested. In all cases, EXP4.P+CON is compared to the WMV algorithm described earlier[4].

[4] Code to reproduce these results will be made available upon publication of this paper.

4.1 Setting

Defining the contextual bandits While human expert CDM datasets exist, they are typically limited in either the number of arms (typically binary), the number of samples, the consistency in which experts participate in the CDM, the absence of confidence, or, finally, the absence of value estimates. To allow us to exhaustively test our methods we use artificial experts which solve an artificial CMAB. We consider a context space of $[0, 1]^d$ with $d = 2$. The value landscape is generated following Perlin noise [10], as visualized in Figure 1 (right panel). Values generated in this manner have an average reward of 0.5 and range from 0 to 1. When pulling an arm with context $\vec{x}$ in this space, the reward is sampled from a binomial distribution with probability of success $p(r = 1; \vec{x}) = f(\vec{x})$, where $f : [0, 1]^d \to [0, 1]$ is the function mapping the context to its value in the value landscape.

Prior expert information We simulate prior knowledge by introducing each expert n to P^n experiences on contexts sampled from within its hyper-rectangle with origin $\vec{o}^n \in [0, 1]^d$ and side lengths $\vec{s}^n \in [0, 1]^d$ with $\vec{o}^n_d + \vec{s}^n_d \leq 1 \; \forall \; d$. This allows us to model the case wherein experts come into play with (in)accurate prior knowledge. Increasing P^n improves the expert's performance, increasing side lengths increases the region of context-space for which the expert can provide relevant advice.

Expert implementation A pool of N experts with differences in prior expertise is used here. Each individual expert is implemented as a KernelUCB algorithm, as mentioned earlier, and exposed to a different subset of the context space. We fix $P^n = 100$, $\vec{s}^d = \overrightarrow{0.5}$ and randomly sample origins from $[0, 0.5]^d$, meaning every expert covers (a possibly overlapping) 25% of the context space. The noisy confidence values are sampled following the noisy model discussed in Section 3.3.

Collective Decision Making A trial or run is defined as one iteration of Algorithm 1. Within each trial, after each expert acquires experiences in its expertise range, collective decision-making occurs with $T = 1000$. The results discussed in the following sections are aggregated over 1000 runs. At the start of each run, a new CMAB instance and pool of experts is generated. For any given trial however, all algorithms considered here solve the same CMAB instance with the same pool of experts.

4.2 Results and Discussion

Due to space limitations the results presented here are limited to two extreme bandit-arms/expert combinations, i.e., many arms ($K = 32$) few experts ($N = 2$), and few arms($K = 2$) many experts ($N = 32$).

How does value advice alter probability advice results? The left-most column of Figure 2 compares the performance of the different algorithms when no confidence is provided. It appears that EXP4.P with value advice performs better than probability advice when $K << N$, a tendency which inverses this condition is no longer met.

When the number of arms is high but the number of experts is low, it is easier for an expert's overestimation of a sub-optimal arm to affect the final decision. In contrast, when the number of experts outnumbers the number of arms, the collective variance is reduced (similarly to ensemble methods [17]). It's notable that EXP4.P's improvement on a non-adaptive weighted majority vote is not as large as one might expect. In part, this is due to the absence of worse than random experts, which is one of the well known conditions for effective majority votes [8].

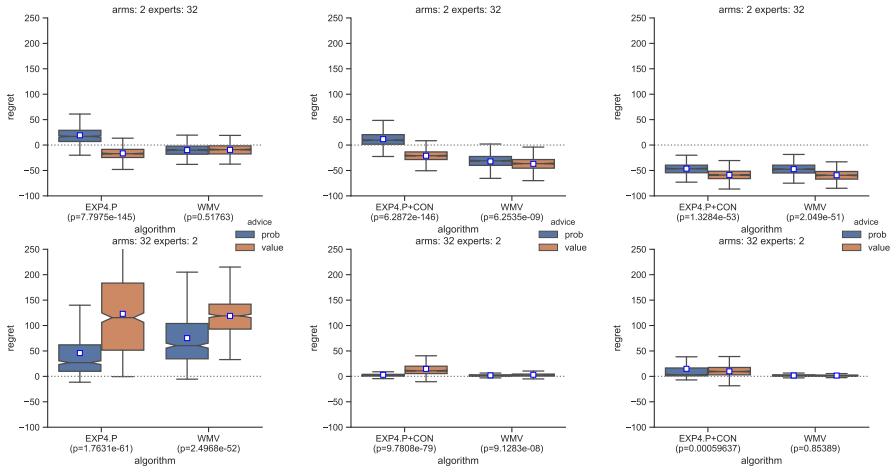

Fig. 2. Performance per advice and confidence type. Regret of different aggregation algorithms in function of advice and confidence type. A value of 0 means the algorithm performs as well as the best expert. The white square marks the mean. The given p-value results from a Wilcoxon test on the results for probability and value advice. This plot presents performance when experts outnumber arms (top) and arms outnumber experts (bottom). (left) Performance without confidence, (middle) performance with non-contextual confidence, and (right) performance with contextual confidence.

How does non-contextual confidence influence EXP4.P and WMV? Results using non-contextual confidence estimates are given in the central column of Figure 2. Comparing these results with those obtained without confidence, a significant (Wilcoxon rank-sum test with confidence level 5%) increase

in performance can be observed for both EXP4.P (with an exception for probability advice in the $N >> K$ case) and the (weighted) Majority vote. We further note that the methods appear to be limited by the performance of the best expert when $K >> N$.

While both methods can improve with the use of non-contextual confidence, the improvements for EXP4.P are less pronounced. This is in part due to how the confidence measures are exploited. In the case of EXP4.P, confidence is essentially used as a prior on the weights, and as such the EXP4.P algorithm can learn to diverge from the given confidence estimates. While this can be useful if confidence is inaccurate, it also reduces the performance benefits when confidence estimates are appropriate. Similar to results obtained in binary classification [8], increasing the number of experts increases the relative performance of the collective, as the lower regret when $N >> K$ suggests.

How does Contextual confidence influence the outcome The results using contextual confidence estimates are given in the right-most column of Figure 2. Comparing these results with those obtained without confidence or with non-contextual confidence, a significant improvement in performance can be observed for both EXP4.P and the (weighted) Majority vote when $N >> K$. Similarly to non-contextual confidence, the methods seem to be bound by the performance of the best expert when N is low.

By providing contextual confidence, experts can be weighted by their (estimated) performance for the given contexts. This allows the methods to improve beyond the performance of non-contextual confidence, which only give a general trend. This seems to especially be the case for EXP4.P+CON. We suggest that the change in priors can purposefully drive exploration in the early stages of the learning process and prevent the early convergence of EXP4.P+CON.

How does noisy confidence estimation affect both methods? Plots in function of different noise levels are given in Figure 3. As the noise levels increase, the performance of EXP4.P+CON and the Majority vote degrades. Furthermore, while the performance of the majority significantly degrades with large noise, the performance of EXP4.P+CON is less affected.

These results strongly suggest that, while EXP4 benefits less from accurate confidence, it is also more robust to noisy confidence estimates than the majority vote is. From this, a rule of thumb for the selection of the appropriate algorithm can be derived. If noisy confidence is expected, one should prefer EXP4.P. What's more, this confirms the intuition that if confidence is known to be extremely noisy, it should be ignored when making decisions.

5 Conclusion

To reduce the influence of uncertainty, this paper proposed an alternative take on advice in the deciding with expert advice setting. More specifically, we introduced value advice and proposed an extension to EXP4.P to integrate such

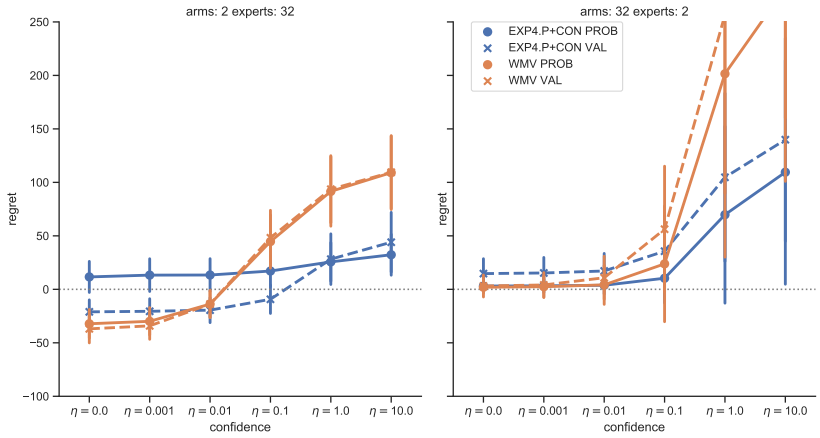

Fig. 3. Influence of noise on algorithm performance. For each noise level (η) confidence is sampled from the beta distribution $\beta(1 + a/\eta, 1 + (1 - a)/\eta)$. A value of 0 means the algorithm performs as well as the best expert. Dashed lines use value advice, full lines use probability advice. This plot presents performance when experts outnumber arms (left) and arms outnumber experts (right).

value advice as opposed to probability advice. Our results show such value advice can significantly improve performance when the number of experts is sufficiently bigger than the number of arms. What's more, to handle the problem of localized expertise, we proposed the addition of confidence estimates in the deciding with expert advice. By using these confidences as priors on EXP4.P's weights we obtain a method that can benefit from confidence and is more robust than the classical weighted majority vote when confidence is noisy. We also find that contextual confidence, which is straightforward to derive from existing CMAB experts can further improve the performance when compared to non-contextual confidence. As the latter only provides information on overall performance it is ineffective at determining optimal per-context weights. Confidence with high noise remains a problem however, suggesting that a method which purposefully identifies when confidence is noisy might provide further improvements. This lays the foundation for future work in which we aim to further explore the influence of noise (in the form of bias) on confidence and how it can be counteracted.

Acknowledgment

Abels A. (FRIA Research Fellow) is financially supported by the Fondation de la Recherche Scientifique (F.R.S.-FNRS) of the Walloon-Brussels Federation.

References

1. Agrawal, R.: Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. Advances in Applied Probability **27**(4), 1054–1078 (1995)
2. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.: The nonstochastic multiarmed bandit problem. SIAM Journal on Computing **32**(1), 48–77 (2002). https://doi.org/10.1137/S0097539701398375, https://doi.org/10.1137/S0097539701398375
3. Bang, D., Aitchison, L., Moran, R., Castanon, S., Rafiee, B., Mahmoodi, A., Lau, J., Latham, P., Bahrami, B., Summerfield, C.: Confidence matching in group decision-making. Nature Human Behaviour **1** (05 2017). https://doi.org/10.1038/s41562-017-0117
4. Bang, D., Frith, C.D.: Making better decisions in groups. Royal Society Open Science **4**(8), 170193 (2017)
5. Beygelzimer, A., Langford, J., Li, L., Reyzin, L., Schapire, R.E.: An optimal high probability algorithm for the contextual bandit problem. CoRR **abs/1002.4058** (2010), http://arxiv.org/abs/1002.4058
6. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 208–214 (2011)
7. Dunning, D.: The dunning–kruger effect: On being ignorant of one's own ignorance. In: Advances in experimental social psychology, vol. 44, pp. 247–296. Elsevier (2011)
8. Grofman, B., Owen, G., Feld, S.L.: Thirteen theorems in search of the truth. Theory and Decision **15**(3), 261–278 (Sep 1983). https://doi.org/10.1007/BF00125672, https://doi.org/10.1007/BF00125672
9. Khashabi, D., Azer, E.S., Khot, T., Sabharwal, A., Roth, D.: On the capabilities and limitations of reasoning for natural language understanding. CoRR **abs/1901.02522** (2019), http://arxiv.org/abs/1901.02522
10. Lagae, A., Lefebvre, S., Cook, R., DeRose, T., Drettakis, G., Ebert, D.S., Lewis, J.P., Perlin, K., Zwicker, M.: A survey of procedural noise functions. In: Computer Graphics Forum. vol. 29, pp. 2579–2600. Wiley Online Library (2010)
11. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670. ACM (2010)
12. Marshall, J.A., Brown, G., Radford, A.N.: Individual confidence-weighting and group decision-making. Trends in Ecology & Evolution **32**(9), 636 – 645 (2017). https://doi.org/https://doi.org/10.1016/j.tree.2017.06.004, http://www.sciencedirect.com/science/article/pii/S0169534717301520
13. Reily, K., Finnerty, P.L., Terveen, L.: Two peers are better than one: aggregating peer reviews for computing assignments is surprisingly accurate. In: Proceedings of the ACM 2009 international conference on Supporting group work. pp. 115–124. ACM (2009)
14. Robson, N., Rew, D.: Collective wisdom and decision making in surgical oncology. European Journal of Surgical Oncology (EJSO) **36**(3), 230 – 236 (2010). https://doi.org/https://doi.org/10.1016/j.ejso.2010.01.002, http://www.sciencedirect.com/science/article/pii/S074879831000003X
15. Scholkopf, B., Smola, A.J.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2001)

16. Shamir, O.: Fundamental limits of online and distributed algorithms for statistical learning and estimation. CoRR **abs/1311.3494** (2013), http://arxiv.org/abs/1311.3494
17. Ueda, N., Nakano, R.: Generalization error of ensemble estimators. In: Proceedings of International Conference on Neural Networks (ICNN'96). vol. 1, pp. 90–95. IEEE (1996)
18. Valko, M., Korda, N., Munos, R., Flaounas, I.N., Cristianini, N.: Finite-time analysis of kernelised contextual bandits. CoRR **abs/1309.6869** (2013), http://arxiv.org/abs/1309.6869
19. Villar, S., Bowden, J., Wason, J.: Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. Statistical Science **30**, 199–215 (05 2015). https://doi.org/10.1214/14-STS504
20. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
21. Zhou, L.: A survey on contextual multi-armed bandits. CoRR **abs/1508.03326** (2015), http://arxiv.org/abs/1508.03326