Cloud-Assisted On-Sensor Observation Classification in Latency-Impeded IoT Systems

Igor Burago and Marco Levorato
Department of Computer Science, University of California, Irvine
{iburago, levorato}@uci.edu

***Abstract*—The combination of computation and communication constraints within the Internet of Things systems require intelligent allocation of decision making and learning processes across a network of sensing and computing devices. In this paper, we present the problem of observation selection for reactive on-sensor decision-making, where the most accurate decision rule cannot be used unaided neither at the sensor (due to limited computing power), nor in the cloud (due to high communication latency). To make time-sensitive adaptation possible in these conditions, we consider learning a decision rule that is computationally viable for on-sensor use and is continuously adjusted by the cloud using the optimal decision rule for supervision. We pose a constrained stochastic optimization problem for online learning of such instrumental on-sensor classifier, propose an algorithm for updating its parameters, and establish the conditions under which convergence to a local extremum is guaranteed, at least for samples of independent observations.**

I. Introduction

The rise of the Internet of Things (IoT) and Machine-to-Machine (M2M) communications [1] has been the leading force for the introduction of distributed forms of intelligence in interconnected devices. Rather than function just as collectors of raw signals, sensing devices have been tasked with an increasing amount of data processing. The advances in machine learning techniques of the recent years further boosted this trend, making even simple devices capable of extracting high-level information from both exogenous and endogenous signals. Exemplar applications exist in a multitude of technological areas, including home automation systems [2], video surveillance systems [3], smart cities infrastructure [4], and transportation systems [5].

The main impediment in such systems originates from the mobile nature of most IoT devices, limiting their computation power, energy availability, and data storage capacity. In existing research, two main approaches are generally proposed to address this issue: (i) reducing the complexity of models and processing algorithms (e.g., classifiers or predictors) used at the sensors [6], and (ii) offloading computations to remote interconnected devices, such as cloud or edge servers [7], [8]. The former approach preserves the locality of signal processing, but typically results either in a decreased generality of the models, which are trained for specific contexts or under specific assumptions, or in a degraded accuracy of the outcome. The latter approach preserves generality and accuracy of signal processing, but inevitably introduces a delay, which often is stochastic in nature due to the inherent variability of the state of the links connecting the mobile devices to the data centers. The edge computing paradigm mitigates this issue by placing compute-capable devices at the network edge connected to the mobile devices through a one-hop wireless link. However, the quality of wireless links can suffer rapid and considerable fluctuations due to variables that are not in full control of the system, such as interference, distance between the transmitter and the receiver, existence of a line-of-sight propagation path, etc.

The consequences of these two approaches are clear: we either accept a reduced on-device analysis capability (via local computing), or accept longer and/or uncertain capture-to-classification outcome delays (via cloud or edge computing). Recent contributions [9] propose to split the workload between the sensors and the cloud to reduce the bandwidth usage necessary to involve the cloud in the classification process. However, this class of approaches still incurs the round-trip delay induced by transferring data from a sensor to the cloud server and the outcome back to the sensor on the reverse path.

The approach and results presented in this paper attempt to bridge these two extremes in order to achieve local analysis at the mobile devices that attains both high accuracy and low complexity. The core idea is to establish an advanced form of collaboration between the local low-complexity decision rule and the classification function of higher complexity at the cloud, where the latter is used to dynamically adjust the former based on observation history. Importantly, our framework realizes a continuous training process, where parameters are repeatedly tuned to match recent observations whose distribution may be influenced by the local time-varying context. Thus, the main question at the center of our work is:

> *Given a structure of an auxiliary low-complexity decision rule to be run at a sensor, how can the system train and update it under the supervision of a high-complexity reference decision process run at the cloud?*

In the rest of the paper, we construct an algorithm for the cloud-assisted adaptation of the parameters governing the local classifier based on a history of observations. Using a stochastic Lyapunov function method and martingale theory, we prove that the proposed stochastic optimization procedure converges and allows for on-the-fly tuning of the local classifier's parameters. Although we envision promising application

This work was supported in part by UCOP under Grant LFR-18-548175 and National Science Foundation under Grant IIS-1724331 and ECCS-1611349.

opportunities for the proposed framework in the areas of mobile health care and autonomous systems, herein, we focus on providing the technical foundations of the approach in general, leaving specific use-cases to future work.

II. Cloud-Assisted Decision-Making

Consider a sensor continuously acquiring a stream of observations $z_t \in Z$ at some discretized schedule of time slots $t = 1, 2, \ldots$. Each observation z_t is to be classified locally to enable fast reaction of the sensor to the characteristics of its signal. For instance, consider an autonomous unmanned aerial vehicle (UAV) using an onboard classifier to detect and classify objects to make navigation decisions. The maximum capture-to-decision time needed to achieve highly responsive flight dynamics may constrain the complexity of the local classifier, whereas it is desirable to have the UAV be able to function in any context and environment. The temporal constraint on decision, together with the need for adaptability, exclude the use of a simplified classifier or offloading to the cloud. Similarly, in mobile healthcare applications, the sensor may use the outcome of local classification to promptly tune signal acquisition parameters, e.g., by increasing the sampling rate, eschewing the need to wait for a response from the cloud.

Below, we concentrate on the *binary classification problem* as the one governing decisions in many monitoring tasks, where an observation z_t is assigned to one of the two classes: *positives* ($z_t \in Z_1$) and *negatives* ($z_t \in Z_0$). One such classifier of a sufficiently high quality is assumed to be available to the system at the cloud processor in the form of a decision rule $\delta\colon Z \mapsto \{0, 1\}$, where $\delta(z_t) = i$ implies $z_t \in Z_i$. Unfortunately, high accuracy typically implies high complexity, so this *reference decision rule* δ cannot be implemented at the sensor due to the limited computing power, while offloading it to the cloud may incur too long of a delay due to the need to transport the observation to the cloud and the outcome to the sensor. Therefore, the sensor is in the need of a simplified *auxiliary decision rule* $\hat{\delta}_t\colon X \mapsto \{0, 1\}$ that it can use at every time slot t to approximate the reference decision $\delta(z_t)$ with $\hat{\delta}_t(x_t)$, based on the feature vector $x_t \in X$ of the observation $z_t \in Z$. In order to be practical, the function $\hat{\delta}_t$ has to be chosen from a class of limited-complexity functions that can be computed by the sensor for any observation in the span of a single time slot.

With the introduction of the auxiliary decision rule, the sensor, then, faces the problem of learning the function $\hat{\delta}_t$ that produces decisions to be on average as close as possible to the decisions output by the reference decision rule δ for the same observations. Due to the difference in computational complexity between the two decision functions, in general, we have to accept that $\hat{\delta}_t$ will always be a lossy representation of δ. Since this is a supervised learning problem, in order to train $\hat{\delta}_t$ for its use at the sensor, the system has to have the values of $\delta(z_t)$ available as a supervisory input for training.

We pose the problem of training an auxiliary decision rule as an online learning problem, for the following reasons. (i) The auxiliary rule $\hat{\delta}_t$, commonly, cannot be pre-trained, as it would implicate a period of low-quality classification output at the sensor for a significant period at the start of its work. (ii) Although the same reference decision rule δ may be used for supervising each sensor, different sensors may have different auxiliary decision rules that are adapted to the distribution of the data they specifically see. (iii) The auxiliary decision rule $\hat{\delta}_t$ may be able to adapt to a temporal drift of the observational distribution seen by the sensor, if the change rate is lower than the rate of convergence of the learning algorithm.

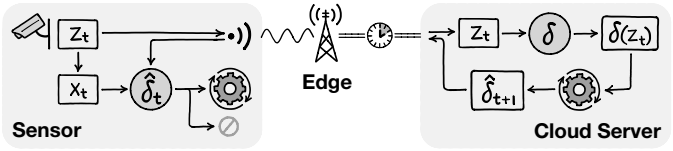

Figure 1. Principal organization of the system implementing cloud-assisted observation selection for on-sensor decision-making. The gear symbols in the Sensor and Cloud Server blocks depict the post-processing induced by an observation being attributed to the class of positives by the auxiliary decision rule $\hat{\delta}_t$, and the processing required from the cloud to update it, respectively.

Because of the memory and computation constraints of the sensor node, we will assume that it is undesirable for the sensor to run the learning algorithm. Therefore, the information necessary for each successive update of the sensor's decision rule, from $\hat{\delta}_t$ to $\hat{\delta}_{t+1}$, either will be used right away at the cloud, which will be already computing $\delta(z_{t+1})$, or will be gathered at the edge, which will do the update instead. In any case, afterwards, the newly updated decision rule $\hat{\delta}_{t+1}$ will be transmitted back to the sensor for deployment. Schematically, the sensor-cloud information exchange is shown in Figure 1.

To make the amount of data necessary for these updates contained within the available bandwidth, the class of functions considered for the auxiliary decision rule should be elided of the functions that cannot be parameterized thriftily. However, we assume that the bandwidth available for the sensor is sufficient for reliably sending all of its observations to the cloud. Importantly, variations in the available bandwidth, for instance due to network congestion, will only affect the rate of adaptation of the local classifier, without impairing the ability of the sensor to make timely decisions.

III. Decision-Rule Learning Problem

According to the reference decision rule at the cloud, the abstract set of all possible observations Z is partitioned into subsets $Z_1 \subseteq Z$ and $Z_0 = Z \backslash Z_1$. Locally, the sensor operates by its own approximation of that partition within some feature space $X \subseteq \mathbb{R}^d$, where the mapping $x_t \triangleq \chi(z_t)$ from observations $z_t \in Z$ to feature vectors $x_t \in X$ is assumed to be fixed by some deterministic feature-extraction procedure χ.

The auxiliary decision rule $\hat{\delta}_t(x)$ is to be found in a parametric form of a separating surface $\{x \in X : f(x, \theta) = \mu\}$ deterministically dividing the two classes of observations in X:

$$\widehat{\delta}(x)\colon\ f(x,\theta) \underset{x \in \widehat{X}_0}{\overset{x \in \widehat{X}_1}{\gtrless}} \mu, \tag{1}$$

where, ideally, the boundaries describable by f allow for $\widehat{X}_0 \cap X_1 = \widehat{X}_1 \cap X_0 = \varnothing$. In general, this goal is unfeasible,

as it requires both (i) the features to be chosen so that the two classes are fundamentally separable in X, i.e., $X_0 \cap X_1 = \varnothing$; and (ii) the sensor's computational constraints to not rule out functions f sufficiently complex to separate X_0 and X_1.

Hence, it is only natural to fit the parameters $\tau = \mathrm{vec}[\theta, \mu]$ of the decision rule (1) to minimize a loss defined via the corresponding error probabilities, instead. From here on, we focus on the following general risk function:

$$V(\tau) \triangleq \mathbb{E}\big[I_{\mathrm{E}}(z,\tau)|u_\theta - \mu|\big], \tag{2}$$

$$I_{\mathrm{E}}(z,\tau) \triangleq I_0(z)\hat{I}_\mu^{(1)}(u_\theta) + I_1(z)\hat{I}_\mu^{(0)}(u_\theta), \tag{3}$$

where $u_\theta \triangleq f(x,\theta)$, $\hat{I}_\mu^{(1)}(u) \triangleq \mathbb{1}[u > \mu]$, $\hat{I}_\mu^{(0)}(u) \triangleq 1 - \hat{I}_\mu^{(1)}(u)$, and $I_i(z) \triangleq \mathbb{1}[z \in Z_i]$ for $i \in \{0,1\}$. The indicator $I_{\mathrm{E}}(z,\tau)$ signals an event of misclassification, i.e., the mismatch between the supervisory decision $I_1(z)$ produced by δ, and its approximation $\hat{I}_\mu^{(1)}(u_\theta)$ produced by $\hat{\delta}$. In other words, $V(\tau)$ is defined to be an average risk of classification error weighted by the "distance" $|u_\theta - \mu|$ to the separating surface whose shape is given by the parameters θ and threshold μ.

In addition to the loss function measuring misclassification, we also consider a cost associated with positive classification, which may be used, for instance, to limit the number of positive samples a sensor can store in its memory bank, or constrain the load imposed by local post-processing of selected observations before they are presented to the user. For simplicity, we assume this cost to be the same regardless of the observation, with its expectation being the probability

$$\varphi(\tau) \triangleq \mathbb{E}\big[\hat{I}_\mu^{(1)}(u_\theta)\big]. \tag{4}$$

Thus, to find the optimal on-sensor decision rule (1), we need to solve the problem of minimizing the expected risk (2) under the constrained cost (4):

$$V(\tau) \longrightarrow \min_\tau, \quad \text{s.t.} \quad \varphi(\tau) \le \varphi_*, \tag{5}$$

where φ_* stands for some threshold following from the sensor's constraints, which, we assume, are such that *potential positive classificability* holds in the sense that φ_* does not prevent the sensor from the possibility of correctly classifying all true positive cases from Z_1 in principle, i.e.,

$$\varphi_* > \pi_1 \triangleq \mathbb{P}[z \in Z_1]. \tag{6}$$

IV. Decision-Rule Update Strategy

To implement online learning of the on-sensor decision rule, we propose the stochastic optimization procedure given in Algorithm 1. The necessary conditions for which it converges to a solution of the problem (5) are established in Theorem 1.

Algorithm 1. *Given the current parameters $\tau_t = \mathrm{vec}[\theta_t, \mu_t]$ after t time slots, obtain a new observation z_{t+1} and compute:*

$$\tau_{t+1} = \mathcal{H}\big(\tau_t - \gamma_t \widehat{G}(z_{t+1}, \tau_t)\big), \tag{7}$$

with learning rate $\gamma_t > 0$, where, for $\tau = \mathrm{vec}[\theta, \mu]$,

$$\widehat{G}(z,\tau) \triangleq \mathrm{vec}\big[J(z,\tau)\nabla_\theta f(\chi(z),\theta),\ -J(z,\tau)\big], \tag{8}$$

$$J(z,\tau) \triangleq \hat{I}_\mu^{(1)}\big(f(\chi(z),\theta)\big) - I_1(z), \tag{9}$$

and $\mathcal{H}(\tau)$ is a risk-preserving transformation encapsulating the a priori knowledge on parameter localization, if it is available, or the identity transform $\mathcal{H}(\tau) \equiv \tau$, otherwise.

Theorem 1. *Given*

1) *a sample of i.i.d. observations $\{z_t\}$;*
2) *a continuously differentiable decision function $f(x,\theta)$, such that, for all feature vectors x, there exist constants $C_f(x)$ and $C_{\nabla f}(x)$, so that for any θ', θ'',*

$$|f(x,\theta'') - f(x,\theta')| \le C_f(x)\|\theta'' - \theta'\|, \tag{10}$$

$$|\nabla_\theta f(x,\theta'') - \nabla_\theta f(x,\theta')| \le C_{\nabla f}(x)\|\theta'' - \theta'\|, \tag{11}$$

where $\mathbb{E}\big[(C_f(x))^\kappa\big] \le C_{f,\kappa}$ and $\mathbb{E}[C_{\nabla f}(x)] \le C_{\nabla f}$, for some $\kappa > 1$ and constants $C_{f,\kappa}$ and $C_{\nabla f}$;
3) *a feature-extraction function $\chi(z)$, such that the c.d.f. F_{u_θ} of the random variable $u_\theta \triangleq f(\chi(z),\theta)$ satisfies*

$$|F_{u_\theta}(u'') - F_{u_\theta}(u')| \le C_F|u'' - u'|, \tag{12}$$

for all θ, u', u'', where C_F is some constant;
4) *a mapping $\mathcal{H}(\tau)$ in the parameter space, such that $V(\mathcal{H}(\tau)) \le V(\tau)$ for all parameter vectors τ;*
5) *a sequence of learning rates $\gamma_t > 0$, such that*

$$\textstyle\sum_{t=1}^\infty \gamma_t = \infty, \qquad \sum_{t=1}^\infty \gamma_t^{2\lambda} < \infty, \tag{13}$$

where $\lambda \triangleq \frac{\kappa}{\kappa+1}$ for κ from condition 2) above;

it holds that Algorithm 1

1) *exhibits the criterion convergence, i.e.,*

$$\lim_{t\to\infty} V(\tau_t) \overset{\text{a.s.}}{=} V_*, \quad \textit{for} \quad \mathbb{E}V_* < \infty; \tag{14}$$

2) *achieves the necessary condition of extremum for $V(\tau)$ on a subsequence, i.e.,*

$$\liminf_{t\to\infty} \|G(\tau_t)\| \overset{\text{a.s.}}{=} 0, \tag{15}$$

where

$$G(\tau) \triangleq \mathrm{vec}\big[\mathbb{E}[J(z,\tau)\nabla_\theta f(x,\theta)],\ -\mathbb{E}[J(z,\tau)]\big]. \tag{16}$$

Proof. To establish the convergence of the algorithm, in the following we employ the stochastic Lyapunov function method and martingale theory [10]–[12]. For the Lyapunov function we use the criterion (2) itself.

1. Consider the variation in the value of the criterion V between two arbitrarily fixed parameter vectors $\tau = \mathrm{vec}[\theta,\mu]$ and $\tau' = \mathrm{vec}[\theta',\mu'] = \tau + \delta\tau$ for some $\delta\tau = \mathrm{vec}[\delta\theta, \delta\mu]$.

By the definitions (2) and (9),

$$\begin{aligned} V(\tau') - V(\tau) &= \mathbb{E}[J(z,\tau')(u_{\theta'} - \mu')] - V(\tau) \\ &= U(\tau,\tau') + W(\tau,\tau'), \end{aligned} \tag{17}$$

where, for $\delta\hat{I}(\tau,\tau') \triangleq \hat{I}_{\mu'}^{(1)}(u_{\theta'}) - \hat{I}_\mu^{(1)}(u_\theta)$ and $\delta u \triangleq u_{\theta'} - u_\theta$,

$$U(\tau,\tau') \triangleq \mathbb{E}\big[\delta\hat{I}(\tau,\tau')(u_{\theta'} - \mu')\big], \tag{18}$$

$$W(\tau,\tau') \triangleq \mathbb{E}\big[J(z,\tau)(\delta u - \delta\mu)\big], \tag{19}$$

2. The expectation (18) also can be trivially split into two by separating the contributions of τ and $\delta\tau$:

$$U(\tau,\tau') = S(\tau,\tau') + T(\tau,\tau'), \tag{20}$$

$$S(\tau,\tau') \triangleq \mathbb{E}\big[\delta\hat{I}(\tau,\tau')(u_\theta-\mu)\big], \quad (21)$$
$$T(\tau,\tau') \triangleq \mathbb{E}\big[\delta\hat{I}(\tau,\tau')(\delta u-\delta\mu)\big]. \quad (22)$$

Due to the concordance between the indicator difference

$$\begin{aligned}\delta\hat{I}(\tau,\tau') &= \mathbb{1}[u_{\theta'}>\mu'] - \mathbb{1}[u_\theta>\mu]\\ &= \mathbb{1}[\mu'-\delta u<u_\theta\le\mu] - \mathbb{1}[\mu<u_\theta\le\mu'-\delta u], \quad (23)\end{aligned}$$

and the term $(u_\theta-\mu)$ in (21), it is obvious that $S(\tau,\tau')\le 0$.

As to the other expectation (22), its magnitude can be tied to the variation in the parameters $\delta\tau$. Indeed, thanks to condition 2), Hölder's inequality, and equation (23),

$$\begin{aligned}|T(\tau,\tau')| &\le \mathbb{E}\big[|\delta\hat{I}(\tau,\tau')|\big(|\delta\mu|+C_f(x)\|\delta\theta\|\big)\big]\\ &\le \big(\mathbb{E}\big[\big(|\delta\mu|+C_f(x)\|\delta\theta\|\big)^\kappa\big]\big)^{\frac{1}{\kappa}}\big(P(\tau,\tau')\big)^{\frac{\kappa-1}{\kappa}}, \quad (24)\end{aligned}$$

where $P(\tau,\tau') \triangleq \mathbb{P}[\mu'-\delta u<u_\theta\le\mu] + \mathbb{P}[\mu<u_\theta\le\mu'-\delta u]$.

3. For the first factor in (24), we have an immediate bound:

$$\begin{aligned}\mathbb{E}\big[\big(|\delta\mu|+C_f(x)\|\delta\theta\|\big)^\kappa\big] &\le \|\delta\tau\|^\kappa\,\mathbb{E}\big[(\max\{1,C_f(x)\})^\kappa\big]\\ &\le \|\delta\tau\|^\kappa\,\mathbb{E}\big[1+C_f(x)^\kappa\big] \le \|\delta\tau\|^\kappa\,(1+C_{f,\kappa}). \quad (25)\end{aligned}$$

For the other factor, we can use the property (10) once again:

$$\begin{aligned}P(\tau,\tau') &\le \mathbb{P}\big[\mu'-C_f(x)\|\delta\theta\|<u_\theta\le\mu\big]\\ &\quad+\mathbb{P}\big[\mu<u_\theta\le\mu'+C_f(x)\|\delta\theta\|\big]\\ &\le \mathbb{P}\big[|u_\theta-\mu'|\le\max\{|\delta\mu|,\,C_f(x)\|\delta\theta\|\}\big]\\ &\le \mathbb{P}\big[|u_\theta-\mu'|\le\max\{1,C_f(x)\}\|\delta\tau\|\big]. \quad (26)\end{aligned}$$

In order to bound the probability in (26), let us introduce the stochastic event $A \triangleq \{|u_\theta-\mu'|\le\|\delta\tau\|^\lambda\}$ with λ from condition 5). By condition 3), the property (11) implies that

$$\mathbb{P}[A] = \mathbb{P}\big[\mu'-\|\delta\tau\|^\lambda\le u_\theta\le\mu'+\|\delta\tau\|^\lambda\big] \le 2C_F\|\delta\tau\|^\lambda. \quad (27)$$

Further, by Chebyshev's inequality,

$$\begin{aligned}Q(\tau,\tau') &\triangleq \mathbb{P}\big[|u_\theta-\mu'|\le\max\{1,C_f(x)\}\|\delta\tau\|\,\big|\,\bar{A}\big]\\ &\le \mathbb{P}\big[\max\{1,C_f(x)\}\ge\|\delta\tau\|^{\lambda-1}\,\big|\,\bar{A}\big]\\ &\le \|\delta\tau\|^{(1-\lambda)\kappa}\,\mathbb{E}\big[1+(C_f(x))^\kappa\,\big|\,\bar{A}\big]. \quad (28)\end{aligned}$$

Then, from (27) and (28) combined, we have the probability

$$\begin{aligned}P(\tau,\tau') &\le \mathbb{P}[A]+\mathbb{P}[\bar{A}]\,Q(\tau,\tau')\\ &\le 2C_F\|\delta\tau\|^\lambda + \|\delta\tau\|^{(1-\lambda)\kappa}\,\mathbb{E}\big[1+(C_f(x))^\kappa\big]\\ &\le (2C_F+C_{f,\kappa}+1)\|\delta\tau\|^\lambda. \quad (29)\end{aligned}$$

Substituting the bounds (25) and (29) into (24) and then (20), while dropping the nonpositive $S(\tau,\tau')$, we finally get:

$$\begin{aligned}U(\tau,\tau') &\le T(\tau,\tau')\\ &\le (C_{f,\kappa}+1)^{\frac{1}{\kappa}}(2C_F+C_{f,\kappa}+1)^{\frac{\kappa-1}{\kappa}}\|\delta\tau\|^{1+\lambda\frac{\kappa-1}{\kappa}}\\ &\le (2C_F+C_{f,\kappa}+1)\|\delta\tau\|^{2\lambda}. \quad (30)\end{aligned}$$

4. Due to the property (11) from condition 2), for all x there must exist $C'_f(x)$, such that $\mathbb{E}[C'_f(x)]\le C'_f<\infty$ and

$$f(x,\theta') \le f(x,\theta) + \langle\delta\theta,\,\nabla_\theta f(x,\theta)\rangle + C'_f(x)\|\delta\theta\|^2. \quad (31)$$

Using the expansion (31) for δu in the definition (19), and recalling (30), (17), and (16), we obtain that, for a constant C,

$$\begin{aligned}V(\tau')-V(\tau)-\langle\delta\tau,\,G(\tau)\rangle &\le U(\tau,\tau') + \mathbb{E}[C'_f(x)]\,\|\delta\theta\|^2\\ &\le U(\tau,\tau') + C'_f\|\delta\theta\|^2 \le C\max\big\{\|\delta\tau\|^{2\lambda},\,\|\delta\tau\|^2\big\}. \quad (32)\end{aligned}$$

In other words, since $U(\tau,\tau')\ge 0$, the function $G(\tau)$ serves the role of a *quasi-gradient* of the Lyapunov function $V(\tau)$.

5. Now, consider the parameters $\tau_t \triangleq \mathrm{vec}[\theta_t,\mu_t]$, obtained after using some t steps of Algorithm 1, and their successors $\tau_{t+1} \triangleq \mathrm{vec}[\theta_{t+1},\mu_{t+1}]$. Applying condition 4) to the adjusted $\tilde{\tau}_{t+1} \triangleq \tau_t - \gamma_t\widehat{G}(z_{t+1},\tau_t)$; using the upper bound (32) on the variation of V after a single step of the algorithm; taking the conditional expectation for a fixed history of observations $z_{1:t} \triangleq (z_1,z_2,\ldots,z_t)$; taking advantage of condition 1); and, finally, noticing that, by its definition (8), the estimator $\widehat{G}(z,\tau)$ is a stochastic expectation of the quasi-gradient $G(\tau)$; we end up with the following bound:

$$\begin{aligned}\mathbb{E}[V(\tau_{t+1})\mid z_{1:t}] &\le \mathbb{E}[V(\tilde{\tau}_{t+1})\mid z_{1:t}]\\ &\le V(\tau_t) - \gamma_t\|G(\tau_t)\|^2 + C\gamma_t^{2\lambda} \le V(\tau_t) + C\gamma_t^{2\lambda}. \quad (33)\end{aligned}$$

Hence, up to the $O(\gamma_t^{2\lambda})$ term, the sequence of $V(\tau_t)$ forms a supermartingale. It is known that such sequences converge, so there must exist a random variable V_* that affords (14). Appropriate results can be derived as corollaries from Doob's theorems on martingales, and are extensively present in the stochastic optimization literature (e.g., see [10], [11]).

Applying the expectation operator to both sides of the inequality (33) and summing it up for $t=1$ through some T, we end up with the following telescoping sum:

$$\begin{aligned}\textstyle\sum_{t=1}^T\big(\mathbb{E}[V(\tau_{t+1})]-\mathbb{E}[V(\tau_t)]\big) &= \mathbb{E}[V(\tau_{T+1})]-\mathbb{E}[V(\tau_1)]\\ &\le -\textstyle\sum_{t=1}^T\gamma_t\mathbb{E}\big[\|G(\tau_t)\|^2\big] + C\sum_{t=1}^T\gamma_t^{2\lambda}. \quad (34)\end{aligned}$$

Passing to the limit $T\to\infty$, and taking into account that, by condition 5), the series of $\gamma_t^{2\lambda}$ is convergent, we must conclude that, with probability one, $\sum_{t=1}^\infty\gamma_t\|G(\tau_t)\|^2<\infty$. At the same time, the series of γ_t alone is divergent, hence necessitating the remaining conclusion (15). □

V. Discussion

A. Feature Space and Decision Function

Within the proposed learning framework, the substantial knowledge about the nature of the on-sensor observation classification problem is localized within three functions: the feature-extraction function $\chi(z)$, the decision function $f(x,\theta)$, and the parameter transformation $\mathcal{H}(\tau)$.

The choice of the features $\chi(z)$ and separator $f(x,\theta)$ is to be guided by the trade-off between their computational complexity (bounded from above by the sensor's constraints) and the resulting separability potential (bounded from below by the minimal necessary decision quality). The mapping $\mathcal{H}(\tau)$ complements the decision functions $f(x,\theta)$ that either: (i) exhibit some parametric redundancy that may slow down learning, in which case $\mathcal{H}$ is to eliminate that redundancy; or (ii) are meaningfully defined only on a (convex) subset D reflecting

the a priori knowledge about the solution, in which case $\mathcal{H}$ projects parameters onto D when they escape it. (In the latter case, existence of $\tau \in \operatorname{int}(D)$ with $\|G(\tau)\| = 0$ is assumed.)

Specific recommendations for these selections are necessarily application-dependent and are out of scope of this paper. Here we simply assume that for the problem at hand one can choose $\chi(z)$, $f(x,\theta)$, and $\mathcal{H}(\tau)$ that satisfy computational constraints, allow for *weak separability* in the sense of

$$\mathbb{P}[u_\theta > u \wedge z \in Z_0] \le \mathbb{P}[u_\theta > u \wedge z \in Z_1], \qquad (35)$$

and meet conditions 2), 3), and 4) of Theorem 1.

A simple yet practical choice for the separating surface is a hyperplane, with $f(x,\theta) \triangleq \langle \theta, x\rangle / \|\theta\|$, $\mathcal{H}(\tau) \triangleq \operatorname{vec}\big[\theta/\|\theta\|, \mu\big]$, and Algorithm 1 being specialized into Algorithm 2 below.

Algorithm 2. *In the notation of Algorithm 1,*

$$\tilde{\theta}_{t+1} = \theta_t - \gamma_t J(z_{t+1}, \tau_t)(\mathbf{I} - \theta_t \theta_t^{\mathsf{T}}) x_{t+1}, \qquad (36)$$

$$\mu_{t+1} = \mu_t + \gamma_t J(z_{t+1}, \tau_t), \qquad (37)$$

$$\theta_{t+1} = \tilde{\theta}_{t+1} / \|\tilde{\theta}_{t+1}\|, \qquad (38)$$

where $J(z,\tau) = \hat{I}_\mu^{(1)}\big(\langle \theta,\, \chi(z)\rangle\big) - I_1(z)$ *for* $\tau = \operatorname{vec}[\theta, \mu]$.

B. Decision Rule and Criterion Structure

From the very beginning, we fix the step-wise nature of the decision rule (1). It might seem interesting to consider a generalization of that rule with an explicit uncertainty zone where the classification decisions are made randomly according to some smooth accepting probability function $\tilde{I}^{(1)}_{\eta_0,\eta_1}$, such that: (i) $\tilde{I}^{(1)}_{\eta_0,\eta_1}(u) = 0$ for $u \le \eta_0$; (ii) $\tilde{I}^{(1)}_{\eta_0,\eta_1}(u) = 1$ for $u > \eta_1$; and (iii) $\tilde{I}^{(1)}_{\eta_0,\eta_1}(u) \le \tilde{I}^{(1)}_{\eta_0,\eta_1}(u')$ for all $u < u'$. However, it can be shown that such an extension is entirely redundant.

Theorem 2. *If the error probabilities for the decision rule* (1) *are continuous functions of* μ*, there exists* $\mu \in [\eta_0, \eta_1]$ *so that*

$$V(\theta,\mu) \le \widetilde{V}(\theta,\eta_0,\eta_1) \;\; \textit{and} \;\; \varphi(\theta,\mu) \le \widetilde{\varphi}(\theta,\eta_0,\eta_1), \qquad (39)$$

where V, φ *denote the original criterion* (2) *and constrained probability* (4)*, while* $\widetilde{V}$, $\widetilde{\varphi}$ *stand for their counterparts with the indicators* $\hat{I}^{(1)}_\mu(u_\theta)$ *and* $\hat{I}^{(0)}_\mu(u_\theta)$ *changed to the probabilities* $\tilde{I}^{(1)}_{\eta_0,\eta_1}(u_\theta)$ *and* $\tilde{I}^{(0)}_{\eta_0,\eta_1}(u_\theta) \triangleq 1 - \tilde{I}^{(1)}_{\eta_0,\eta_1}(u_\theta)$*, respectively.*

The criterion (2) has an important structural property following from the concordance of the two factors that are involved in it. Indeed, it is easy to notice (as we do in (17)) that $u_\theta - \mu > 0$ when $I_{\mathrm{E}}(z,\tau) = 1$, and $u_\theta - \mu \le 0$ otherwise. This composition of the expected risk becomes significant for the key element in construction of Algorithm 1, as it allows for a stochastic quasi-gradient (8) that can be efficiently computed on iterations, making efficient *online update* of the on-sensor decision rule possible.

C. Convergence Conditions and Implications

For a conventional choice of learning rate $\gamma_t = Ct^{-a}$ with $C, a > 0$, from condition 5) in Theorem 1 it follows that, for the requirements (13) to hold, the exponent a must satisfy $\frac{1}{2\lambda} = \frac{\kappa+1}{2\kappa} < a \le 1$ for κ from condition 2). If all moments of the random variable $C_f(x)$ defined in that condition 2) are known to be finite, so that $\kappa = \infty$, then $\lambda = 1$, $\frac{1}{2} < a \le 1$, and (13) turns into a rather standard combination of requirements on the series of γ_t and γ_t^2.

The conclusion 2) of Theorem 1 guarantees convergence to parameters satisfying the necessary part of the condition of extremum, which is rather standard for multi-extremal optimization problems. Since the last component of the quasi-gradient $G(\tau)$ has the magnitude of $|\varphi(\tau) - \pi_1|$, the limit (14) implies that, at least on a subsequence, the probability $\varphi(\tau_t)$ converges to the true positive probability (6), thus implicitly satisfying the constraint imposed in problem (5), i.e.,

$$\liminf_{t\to\infty} \varphi(\tau_t) \overset{\text{a.s.}}{=} \pi_1 < \varphi_*. \qquad (40)$$

When further a priori knowledge is available for the specifics of the application in question, the kind of convergence in (40) may be strengthened. For instance, the following holds.

Theorem 3. *In the context of Theorem 1, if condition 2) holds for* $\kappa = \infty$*, and* $\mathcal{H}(\tau)$ *is such that (i)* $\|\mathcal{H}(\tau)\| \le C < \infty \;\; \forall \tau$*, and (ii)* $\mathbb{E}\|\tau_{t+1} - \tau_t\| \le C_1\gamma_t\|G(\tau_t)\| + C_2\gamma_t^2$ *for some* C_1, C_2*, where* τ_{t+1} *is defined by the iteration step* (7) *of Algorithm 1, then* $\varphi(\tau_t)$ *converges to* π_1 *in mean square and w.p.* 1.

Unlike the implication (40) from Theorem 1, Theorem 3 guarantees that, w.p. 1, the sequence $\{\tau_t\}$ will indeed have $\varphi(\tau_t) < \varphi_*$, starting with some time slot t, as desired.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, "IoT based smart security and home automation system," in *International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2016, pp. 1286–1289.

[3] T. Zhang, A. Chowdhery, V. Bahl, K. Jamieson, and S. Banerjee, "The design and implementation of a wireless video surveillance system," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 426–438.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[5] C. T. Barba, M. A. Mateos, P. R. Soto, A. M. Mezher, and M. A. Igartua, "Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 902–907.

[6] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *CoRR*, vol. abs/1511.06530, 2015.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[9] J. Hauswald, T. Manville, Q. Zheng, R. Dreslinski, C. Chakrabarti, and T. Mudge, "A hybrid approach to offloading mobile image classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8375–8379.

[10] B. T. Polyak, *Introduction to Optimization*, ser. Translations Series in Mathematics and Engineering. New York: Optimization Software, 1987.

[11] H.-F. Chen, *Stochastic Approximation and Its Applications*, ser. Nonconvex Optimization and Its Applications. Dordrecht: Kluwer Academic Publishers, 2002, vol. 64.

[12] A. N. Shiryaev, *Probability*, 2nd ed., ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1996, vol. 95.