2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Smartphone Battery Saving by Bit–Based Hypothesis Spaces and Local Rademacher Complexities

Davide Anguita, *Senior Member, IEEE*, Alessandro Ghio, *Member, IEEE*,
Luca Oneto, *Student Member, IEEE*, Sandro Ridella, *Member, IEEE*

***Abstract*— Smartphones emerge from the incorporation of new services and features into mobile phones, allowing to implement advanced functionalities for the final users. The implementation of Machine Learning (ML) algorithms on the smartphone itself, without resorting to remote computing systems, allow to achieve such goals without expensive data transmission. However, smartphones are resource-limited devices and, as such, suffer from many issues, which are typical of stand–alone devices, such as limited battery capacity and processing power. We show in this paper how to build a thrifty classifier by exploiting bit–based hypothesis spaces and local Rademacher Complexities. The resulting classifier is tested on a real–world Human Activity Recognition application, implemented on a Samsung Galaxy S II smartphone.**

I. Introduction

SMARTPHONES are nowadays playing an important role in the exploration of novel alternatives for the retrieval of information directly from the users. It is foreseen that these devices will be able to monitor and learn from our actions effectively and unobtrusively [1], thus allowing to implement personalized approaches to healthcare and remote assistance, entertainment, or marketing. Such results can be achieved thanks to the opportunities, offered by smartphones, of identifying the actions carried out by one or more subjects through the gathering and understanding of context information about the user state, behavior and habits and its surrounding environment. This can be done by the exploitation of environmental and on-body sensors, and distributed computing resources. Accelerometry is one of the mechanisms used for the retrieval of information [2], while peripherals like Bluetooth and data collection mechanisms like cloud storages allow to implement effective knowledge sharing and context depicting.

Current–generation smartphones embed accelerometers, gyroscopes and magnetometers. The purpose of these sensors was initially to allow implementing appealing and enriched graphical user interfaces and more pervasive gaming options, but they also provide information about the actions performed by the device owner, and allow to extract high–level behavioral patterns through Machine Learning (ML) algorithms [2]. ML algorithms on such mobile devices could target a wide range of new applications that benefit from the device processing and opportunistic sensing capabilities.

Davide Anguita, Alessandro Ghio, Luca Oneto and Sandro Ridella are with the University of Genoa, Via Opera Pia 11A, I-16145, Genoa, Italy (email: {Davide.Anguita, Alessandro.Ghio, Luca.Oneto, Sandro.Ridella}@unige.it).

We focus here on the problem of learning effective multi–class classifiers to be implemented on smartphones for Human Activity Recognition (HAR). We target mainly energy efficiency, which is currently the main limitation of these mass-marketed devices, and propose a novel approach that requires fewer system resources for its operation, if compared to a straightforward implementation.

For this purpose, we will consider the conventional supervised learning framework, where the learning process consists in selecting an appropriate hypothesis space and, then, in choosing the most suitable model in it [3]. It is a well–known fact that these two steps lead to an *approximation* error, depending on the (non-optimal) choice of the hypothesis space and an *estimation* error, due to the finite number of available observations for learning [4]. The approximation and estimation errors have been widely investigated throughout the last decades, and effective theoretical approaches and practical procedures have been designed to deal with them.

However, when resorting to actual computing systems, further constraints arise, related to implementation restrictions. In large scale learning, for example, data do not usually fit in memory and computational capabilities are not sufficient for learning the whole dataset, which compels to the searching for approximate solutions [5], [6].

On another side, resource–limited devices give rise to other types of constraints. In particular, power consumption and thermal dissipation lead to preferring fixed-point computations over floating-point ones to increase computation efficiency and improve battery saving [7], [8]. In addition to new challenges in designing the learning process, such constraints introduce an *implementation* error, related to the restrictions of realizations on real-world devices. The investigation of these aspects from both a theoretical and a practical point of view is thus necessary to design procedures allowing to cope with these problems and to target our objectives.

The problem of finding models to be effectively implemented on resource-limited devices, has been usually tackled by adapting models to fit computational constraints in an *a-posteriori* fashion (namely, after the learning process concluded) [9]. In this paper, instead, we reverse the perspective by acting on the whole learning process, starting from the definition of the hypothesis space. For such purposes, we mostly rely on two main pillars, i.e. two ideas that emerged in recent literature.

The first result [10], [11], [12] allows proving that enhancements can be obtained if the hypothesis space is *local*:

namely, it consists of only those functions, that are most likely to be chosen by the learning procedure.

The second pillar result [13] shows that exploiting a limited representation when defining an hypothesis space is equivalent to introducing a regularization in the learning process: in fact, using few bits allows reducing noise as well as the number of functions included in the hypothesis space [14], [15].

In this work we propose a novel learning procedure: by exploiting a representation relying on few bits, it allows to describe a limited number of functions, chosen so to foster locality. The derived advantages are threefold:

- The learning process is implicitly regularized by the use of a limited representation;
- Model performance is improved thank to the locality of the hypothesis space;
- Physical/computational constraints hold, as only few bits are exploited.

The paper is organized as follows. Section II briefly recalls the standard supervised learning framework, in particular by focussing on how it must be modified to deal with implementation on resource–limited devices. Section III moves from the theoretical introductory considerations and results of Section II to practice, by presenting an actual algorithm to train effective classifiers and by contextualizing it to the particular case of the exploitation of a well–known solver tool. Section IV shows some preliminary results on a Human Activity Recognition on Smartphones (HARoS) dataset, publicly available on a widely known online file repository. Finally, Section V proposes some concluding remarks.

II. The Learning Framework: Structural Risk Minimization and Local Rademacher Complexity

In this section, we recall the standard supervised learning framework [10] and, then, we properly modify it so to introduce the implementation constraints, typical of small scale learning with resource-limited devices, such as smartphones.

Our goal is to approximate the relationship between inputs from a set $\mathcal{X}$ and outputs from a set $\mathcal{Y}$, which is encoded by a fixed, but unknown, probability distribution μ over $\mathcal{X} \times \mathcal{Y}$. The learning algorithm maps a set of labeled samples $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ to $h \in \mathcal{H}$ ($X \in \mathcal{X}$ and $Y \in \mathcal{Y}$). The accuracy in representing the hidden relationship μ is measured with reference to a loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1] . \tag{1}$$

The generalization error is defined as

$$L(h) = \mathbb{E}_{(X,Y)} \ell(h(X), Y), \tag{2}$$

where we assume that each labelled sample is independently generated according to μ. Since μ is unknown, we can only compute its empirical estimate, i.e. the empirical error

$$\hat{L}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(X_i), Y_i). \tag{3}$$

It is possible to prove that we can bound $L(h)$ with probability $(1 - \delta)$, $\forall h \in \mathcal{H}$, where $\mathcal{H}$ is the hypothesis space, by using empirical quantities only [10]:

$$\begin{aligned} L(h) \le & c_1 \hat{L}_n(h) + c_2 r^* + \phi_1(\delta, n) \\ & r^* : r = c_3 \hat{\mathcal{R}}_n \left\{ \mathcal{H}^{(*,h_0)} :\ \hat{L}_n(h) \le 2r \right\} + \phi_2(\delta, n) \end{aligned} \tag{4}$$

where c_1, c_2, c_3 are computable constants, while ϕ_1, ϕ_2 are functions that depend only on the level of confidence [10]. Moreover, $\hat{\mathcal{R}}_n \left\{ \mathcal{H}^{(*,h_0)} :\ \hat{L}_n(h) \le 2r \right\}$ is the Local Rademacher Complexity (LRC) term, that can be computed as follows:

$$\begin{aligned} & \hat{\mathcal{R}}_n \left\{ \mathcal{H}^{(*,h_0)} :\ \hat{L}_n(h) \le 2r \right\} = \\ & \qquad \mathbb{E}_{\sigma_1,\dots,\sigma_n} \sup_{h \in \mathcal{H}^{(*,h_0)}, \hat{L}_n(h) \le 2r} \frac{2}{n} \sum_{i=1}^{n} \sigma_i \ell(h(X_i), Y_i) \end{aligned} \tag{5}$$

where $\sigma_1, \dots, \sigma_n$ are independent uniform $\{\pm 1\}$-valued random variables. $\mathcal{H}^{(*,h_0)}$ is the hypothesis space, derived by star–shaping $\mathcal{H}$ around h_0:

$$\mathcal{H}^{(*,h_0)} = \{ h + \alpha(h - h_0) :\ h \in \mathcal{H}, \alpha \in [0, 1] \} , \tag{6}$$

$$\mathcal{H}^* = \mathcal{H}^{(*,0)} = \{ \alpha h :\ h \in \mathcal{H}, \alpha \in [0, 1] \} . \tag{7}$$

The bound of Eq. (5) contemplates only those functions in $\mathcal{H}$ that will be likely chosen by the learning process to assess the performance of a model. Consequently, according to the Structural Risk Minimization (SRM) principle [3], we can define an infinite series of hypothesis spaces of increasing size $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ and select:

$$\begin{aligned} h^{\text{opt}} :\ & \arg \min_{h \in \mathcal{H}_i \in \{ \mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \}} L(h) \\ & L(h) \le c_1 \hat{L}_n(h) + c_2 r^* + \phi_1(\delta, n) \\ & r^* : r = c_3 \hat{\mathcal{R}}_n \left\{ \mathcal{H}_i^{(*,h_0)} :\ \hat{L}_n(h) \le 2r \right\} + \phi_2(\delta, n) \end{aligned} \tag{8}$$

As a last step, we devote the last part of this section to the introduction of implementation constraints of real-world applications on resource-limited devices, as discussed in the introduction. For this purpose, we exploit the results in [13], by limiting our analysis to the binary classification case: $\mathcal{X} \in \mathbb{R}^d$, so that $X = \boldsymbol{x}$, and $\mathcal{Y} \in \{\pm 1\}$, so that $Y = y$. $\mathcal{H}$ consists of functions, which are parametrized as follows:

$$h(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}),\ \boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^D \text{ and } \boldsymbol{w} \in \mathbb{R}^D. \tag{9}$$

Moreover, the ramp loss [16]

$$\ell(h(\boldsymbol{x}), y) = \frac{1}{2} \min[2, \max[0, 1 - y h(\boldsymbol{x})]] \tag{10}$$

is used as loss function, which allows to perform regularization in the learning process. Moreover with the ramp loss function the Rademacher Complexity defined in Eq. (5) becomes:

$$\begin{aligned} & \hat{\mathcal{R}}_n \left\{ \mathcal{H}^{(*,h_0)} :\ \hat{L}_n(h) \le 2r \right\} = \\ & \qquad \mathbb{E}_{\sigma_1,\dots,\sigma_n} \sup_{h \in \mathcal{H}^{(*,h_0)}, \hat{L}_n(h) \le 2r} \frac{2}{n} \sum_{i=1}^{n} \sigma_i \ell(h(X_i), Y_i) = \end{aligned}$$

$$\mathbb{E}_{\sigma_1,\ldots,\sigma_n} \sup_{h\in\mathcal{H}^{(*,h_0)},\hat{L}_n(h)\le 2r} \frac{2}{n}\sum_{i=1}^{n}\ell(h(X_i),\sigma_i) \tag{11}$$

According to the SRM principle, we start by defining the infinite series of hypothesis spaces as $\mathcal{H} = \{\boldsymbol{w} : \|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2\}$, where w_{MAX}^2 is the hyperparameter that adjusts the size of the class of functions. It is worth underlining that $\mathcal{H}$ is star–shaped around zero by definition, i.e. $\mathcal{H} = \mathcal{H}^{(*,0)} = \mathcal{H}^*$. Given the restrictions of several real-world applications and implementations on resource-limited devices (like smartphones), it is often convenient to resort to fixed-point arithmetics, in order to reduce the computational burden of Machine Learning approaches. A (possibly limited) number of bits should be then used in order to describe the hypothesis space and the functions it includes, for example by defining the following class of models:

$$\mathcal{H} = \left\{ \boldsymbol{w} : \left[\begin{array}{l} \|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2 \\ w_j \in \frac{w_{\text{MAX}}}{2^\kappa - 1}\{-2^\kappa+1,\ldots,2^\kappa-1\} \\ \sum_{i=1}^{D}[w_j \ne 0] \le D\zeta \end{array} \right] \right\} \tag{12}$$

where κ is the number of exploited bits. Moreover, a further constraint has been introduced to select a subset of features (ζ controls the maximum percentage of inputs to consider): in resource-limited applications, this is of importance to limit the computational burden by neglecting features (e.g. sensors signals) that are of scarce influence on the overall classification process. It is clear noting that the size of the hypothesis space increases with the hyperparameters w_{MAX}, κ, and ζ.

As, unfortunately, the class of functions of Eq. (12) is not star–shaped, a further step is needed, i.e. we can impose star-shaping around zero:

$$\mathcal{H}^* = \{\alpha\boldsymbol{w} : \boldsymbol{w} \in \mathcal{H}, \alpha \in [0,1]\}. \tag{13}$$

III. From Theory to Practice

We have to reformulate the SRM optimization Problem (8) in order to exploit $\mathcal{H}^*$, as defined in the last part of the preceding section. We need to define some further quantities:

$$\Phi = [\boldsymbol{\phi}(\boldsymbol{x}_1)|\ldots|\boldsymbol{\phi}(\boldsymbol{x}_n)]^T \tag{14}$$
$$Y = \text{diag}[y_1|\ldots|y_n] \tag{15}$$
$$\Theta = \text{diag}[\sigma_1|\ldots|\sigma_n] \tag{16}$$
$$\boldsymbol{a}_n = [a_1|\ldots|a_n]^T \tag{17}$$
$$\boldsymbol{y} = [y_1|\ldots|y_n]^T \tag{18}$$
$$\boldsymbol{\sigma} = [\sigma_1|\ldots|\sigma_n]^T. \tag{19}$$

As we have to compute the two quantities of interest, namely $\min_{h\in\mathcal{H}} \hat{L}_n(h)$ and $\hat{\mathcal{R}}_n\left\{\mathcal{H}^* : \hat{L}_n(h) \le 2r\right\}$, the corresponding minimization problem can be formulated as follows:

$$\min_{h\in\mathcal{S},\frac{1}{n}\sum_{i=1}^{n}\ell_T(h(\boldsymbol{x}_i),y_i)\le 2r} \sum_{i=1}^{n}\ell(h(\boldsymbol{x}_i),\sigma_i). \tag{20}$$

In order to compute $\hat{\mathcal{R}}_n\left\{\mathcal{H}^* : \hat{L}_n(h) \le 2r\right\}$, we have $\mathcal{S} = \mathcal{H}^*$; instead, $\mathcal{S} = \mathcal{H}$ and $\sigma_i = y_i,\ \forall i \in \{1,\ldots,n\}$ when we are interested in deriving $\min_{h\in\mathcal{H}} \hat{L}_n(h)$. Consequently, the (overall) optimization problem can be formulated as:

$$\min_{\boldsymbol{w},\alpha,\boldsymbol{\xi^y},\boldsymbol{\xi^\sigma}} \min[\boldsymbol{2}_n,\boldsymbol{\xi^\sigma}], \tag{21}$$
$$\text{s.t.} \begin{cases} \|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2 \\ \boldsymbol{\xi^y} \ge \max[\boldsymbol{0}_n, \boldsymbol{1}_n - \alpha Y\Phi\boldsymbol{w}] \\ \boldsymbol{\xi^y} \ge \boldsymbol{0}_n \\ \boldsymbol{\xi^\sigma} \ge \max[\boldsymbol{0}_n, \boldsymbol{1}_n - \alpha\Theta\Phi\boldsymbol{w}] \\ \boldsymbol{\xi^\sigma} \ge \boldsymbol{0}_n \\ \boldsymbol{w} \in \frac{w_{\text{MAX}}}{2^\kappa-1}\{-\boldsymbol{2}_D^\kappa+\boldsymbol{1}_D,\ldots,\boldsymbol{2}_D^\kappa-\boldsymbol{1}_D\} \\ \boldsymbol{1}_D^T[\boldsymbol{w} \ne \boldsymbol{0}_D] \le D\zeta \\ \boldsymbol{1}_n^T \min[\boldsymbol{2}_n,\boldsymbol{\xi}]^{\boldsymbol{y}} \le 4nr \end{cases}$$

where $\alpha = 1$ when searching for $\min_{h\in\mathcal{H}} \hat{L}_n(h)$.

Problem (21) is an NP-problem and, thus, we introduce a further error source, related to the impossibility of finding its global minimum. Nevertheless, the hypothesis space is designed so to consist of a limited number of functions, and thus it can be effectively explored [10].

Then, a good solution can be generally found by exploiting general purpose optimization tools, such as CPLEX [17], the method proposed in [13], or a mixture of these last two techniques. With reference to this last option, we first define an alternative formulation of Problem (21), where a relaxation of the combinatorial constraints is implemented:

$$\min_{\boldsymbol{w},\boldsymbol{\xi^y},\boldsymbol{\xi^\sigma}} \min[\boldsymbol{2}_n,\boldsymbol{\xi^\sigma}], \tag{22}$$
$$\text{s.t.} \begin{cases} \|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2 \\ \boldsymbol{\xi^y} \ge \max[\boldsymbol{0}_n, \boldsymbol{1}_n - Y\Phi\boldsymbol{w}] \\ \boldsymbol{\xi^y} \ge \boldsymbol{0}_n \\ \boldsymbol{\xi^\sigma} \ge \max[\boldsymbol{0}_n, \boldsymbol{1}_n - \Theta\Phi\boldsymbol{w}] \\ \boldsymbol{\xi^\sigma} \ge \boldsymbol{0}_n \\ \boldsymbol{1}_n^T \min[\boldsymbol{2}_n,\boldsymbol{\xi^y}] \le 4nr \end{cases}$$

Note that α does not appear in the previous formulation since $\alpha\boldsymbol{w}$, with $\alpha \in [0,1]$, already belongs to the class $\|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2$.

We can solve Problem (22) using CPLEX [17]. For this purpose we have to reformulate Problem (22) as follows:

$$\min_{\boldsymbol{w},\boldsymbol{\xi^y},\boldsymbol{\xi^\sigma},\boldsymbol{\eta^y},\boldsymbol{\eta^\sigma}} \boldsymbol{\xi^\sigma} + 2\boldsymbol{\eta^\sigma}, \tag{23}$$
$$\text{s.t.} \begin{cases} \|\boldsymbol{w}\|_2^2 \le w_{\text{MAX}}^2 \\ \boldsymbol{\xi^y} \ge \boldsymbol{1}_n - Y\Phi\boldsymbol{w} - M\boldsymbol{\eta^y} \\ \boldsymbol{\xi^y} \ge \boldsymbol{0}_n \\ \boldsymbol{\xi^\sigma} \ge \boldsymbol{1}_n - \Theta\Phi\boldsymbol{w} - M\boldsymbol{\eta^\sigma} \\ \boldsymbol{\xi^\sigma} \ge \boldsymbol{0}_n \\ \boldsymbol{1}_n^T(\boldsymbol{\xi^y} + 2\boldsymbol{\eta^y}) \le 4nr \\ \boldsymbol{\eta^y} \in \{0,1\} \\ \boldsymbol{\eta^\sigma} \in \{0,1\} \end{cases} \tag{24}$$

The previous problem is equivalent, for large values of M, to Problem (22), as also underlined in [18]. Problem (23) is a standard Mixed Integer Quadratic Programming (MIQP).

As a last step, we project the "floating-point" solution to the nearest (with respect to the Euclidean distance) solution, which satisfies the two combinatorial constraints:

$$\boldsymbol{w} \in \frac{w_{\text{MAX}}}{2^\kappa - 1}\{-\boldsymbol{2}_D^\kappa+\boldsymbol{1}_D,\ldots,\boldsymbol{2}_D^\kappa-\boldsymbol{1}_D\} \tag{25}$$

$$\mathbf{1}_D^T \left[\boldsymbol{w} \neq \mathbf{0}_D \right] \leq D\zeta. \quad (26)$$

It is worth underlining that Problem (21) is anyhow more complex to solve than other learning procedures (e.g. models training with the Support Vector Machine algorithm [3]): in other words, we move the computational burden to the learning phase, so to keep as light as possible the feed-forward running phase. This is not unusual in learning, where a trade-off between learning and model complexity is implemented and properly balanced (e.g. refer to [19], [20], [21], [22]).

IV. RESULTS & DISCUSSION

We propose some preliminary results, obtained by applying the previously presented approach to the Human Activity Recognition on Smartphones (HARoS) [8], [23] dataset. A set of trials with volunteers was required to create and develop the Human Activity Recognition (HAR) dataset. In total, 30 people with ages from 19 to 48 years participated in this research and performed a set of motion sequences comprising the 6 proposed Activities of Daily Living:

1) walking
2) walking upstairs
3) walking downstairs
4) sitting
5) standing
6) laying.

Each subject performed the experiment protocol twice, and each activity was at least performed two times on each trial to simulate repeatability (refer to [8] for further details). Also, a timeout of 5 seconds in which people remained still was arranged between each activity in order to separate each task and simplify the data labeling process.

The obtained database was partitioned into training and test sets in a proportion of 70% to 30%. The partition was randomized but assuring that no samples were from the same user in both subsets. The Samsung I9100 Galaxy S II smartphone was the device utilized for the experiments. The manual labeling process was done by selecting the videos recorded from the experiments as the ground truth and comparing them with the log files of the inertial signals. Since the dataset consists of six classes and in this work we deal with two-class problems, a One-vs-One (OvO) approach was applied.

Model selection was performed by varying w_{MAX} in the range $[10^{-3}, 10^2]$ among 10 values, equally spaced in a logarithmic scale; we also tested different values for $\kappa = \{8, 16, 32\}$ and $\zeta = \{0.1, 1\}$, for which we computed the class complexity r^* accordingly. Table I reports the empirical error on the training set $\hat{L}(h)$ and on the test set $\hat{L}_T(h)$, in addition to r^*.

Table I also presents results showing how performance, in terms of predictions per second and battery life (in hours), changes as κ and ζ are varied: these results are averaged over the different OvO binary classification problems. These values were obtained by simulating the HARoS process on a Samsung Galaxy S II smartphone[1], where we implemented both fixed-point (8 and 16 bits) and floating-point (32 bits) procedures, in accordance with the explored values of κ. The code was written in Java for the user interface and in C for implementing the most expensive operations, such as signal processing and classification algorithms, more efficiently: the use of C on Android was possible thanks to the Native Development Kit (NDK), which allows embedding native code components into Android OS applications. Most of the phone services (e.g. Wi-Fi and 3G Network) as well as the phone screen were switched-off during tests, as they are in general the most energy consuming phone devices. The idea was to isolate the classification process as much as possible to obtain an approximate estimation of the battery consumption of the realized app.

The time of the activity recognition process from the sensor reading to determining the output was measured to estimate the average prediction rate for each approach. Table I shows the obtained results. It is worth highlighting the large difference between the rates obtained using the fixed-point representation and the results achieved by floating-point arithmetic, and the proportional relationship between the number of bits used and the processing time.

An additional test was carried out aimed to measure battery consumption with the floating-point and fixed-point representations. The experiment consisted of continuously running the HAR smartphone application and measuring the battery discharging time from a fully charged state down to a minimum level of 10%. We found that the average battery life is increased up to 100% when a fixed-point 8-bit application is running instead of a floating-point one. These results are highly dependent on the exploited hardware and operating system, but they show the improvements that can be achieved thanks to the proposed approach.

For obtaining a more reliable measure of the relationship between the battery savings and processing time, more experimental tests with different devices and operational conditions would be required. In current scenarios, even small savings in battery consumption make a big difference in deciding whether or not to use a mobile app: this is the case where HAR applications are required to deliver activity information to other higher-level decision applications (e.g. phone apps for maintaining a healthy lifestyle through HAR [24]), thus implying sharing system resources. A general aim is to build a device able to operate at least during a full day, so that the battery recharges can occur during the night time. These results are a good indicator of the benefits that this method can offer for saving battery life and the possibility of being integrated into devices for everyday life.

V. FINAL REMARKS

We showed in this paper how reducing the number of bits emerges as an appealing procedure to decrease the

[1]The smartphone is equipped with a Li-Ion 1650 mAh battery, with up to 610 hours of stand-by operation time, and mounts Android Gingerbread v2.3.4 operating system.

TABLE I

EXPERIMENTAL RESULTS ON THE HAROS DATASET.

	$\zeta = 0.1$									$\zeta = 1$								
	$\kappa = 8$			$\kappa = 16$			$\kappa = 32$			$\kappa = 8$			$\kappa = 16$			$\kappa = 32$		
Empirical error $\hat{L}(h)$, complexity of the class r^*, error on the test set $\hat{L}_T(h)$																		
OvO	$\hat{L}$	r^*	$\hat{L}_T$	$\hat{L}$	r^*	$\hat{L}_T$	$\hat{L}$	r^*	$\hat{L}_T$	$\hat{L}$	r^*	$\hat{L}_T$	$\hat{L}$	r^*	$\hat{L}_T$	$\hat{L}$	r^*	$\hat{L}_T$
1 vs 2	4.06	1.41	9.32	4.06	1.57	9.32	3.90	2.39	9.14	3.90	1.58	9.14	3.90	1.75	9.18	3.90	2.75	9.18
1 vs 3	3.08	3.18	7.24	3.08	3.45	7.24	2.90	3.52	7.60	2.90	3.63	7.60	2.90	3.63	7.60	2.90	3.64	7.60
1 vs 4	0.00	2.22	0.29	0.00	2.99	0.29	0.00	2.99	0.35	0.00	2.75	0.35	0.00	3.16	0.35	0.00	3.75	0.35
1 vs 5	6.98	3.04	9.55	0.02	3.46	0.31	32.29	4.72	34.50	0.00	4.45	0.31	31.95	4.60	33.92	0.00	4.75	0.31
1 vs 6	0.00	4.13	0.00	0.00	5.56	0.00	0.00	8.36	0.00	0.00	7.50	0.00	0.00	8.35	0.00	0.00	8.50	0.00
2 vs 3	4.98	0.49	7.53	4.98	1.40	7.53	4.99	1.40	7.55	4.99	1.38	7.55	4.99	1.42	7.55	4.99	1.58	7.55
2 vs 4	0.01	0.49	0.35	0.01	1.94	0.35	0.00	4.75	0.32	0.00	1.94	0.32	0.00	4.55	0.32	0.00	4.57	0.32
2 vs 5	0.01	0.74	0.12	0.01	2.60	0.12	0.00	5.70	0.13	0.00	0.84	0.13	0.00	4.74	0.13	0.00	5.74	0.13
2 vs 6	0.00	2.29	0.00	0.00	3.50	0.00	0.00	5.69	0.00	0.00	2.29	0.00	0.00	5.62	0.00	0.00	5.64	0.00
3 vs 4	0.01	0.24	0.03	0.01	0.87	0.03	0.00	2.49	0.05	0.00	0.88	0.05	0.00	2.58	0.05	0.00	2.59	0.05
3 vs 5	0.00	0.44	0.00	0.00	0.97	0.00	0.00	3.63	0.00	0.00	1.03	0.00	0.00	4.25	0.00	0.00	4.27	0.00
3 vs 6	0.01	2.51	0.06	0.01	3.80	0.06	0.00	5.10	0.05	0.00	2.45	0.05	0.00	5.22	0.05	0.00	5.23	0.05
4 vs 5	13.40	0.71	17.42	13.40	0.71	17.42	12.81	0.80	16.37	12.81	0.80	16.37	12.80	0.80	16.35	12.80	0.80	16.35
4 vs 6	0.17	3.33	0.00	0.17	3.63	0.00	0.01	6.54	0.00	0.01	3.30	0.00	0.00	6.53	0.00	0.00	6.53	0.00
5 vs 6	0.00	1.63	0.00	0.00	1.83	0.00	0.00	3.60	0.00	0.00	1.48	0.00	0.00	3.82	0.00	0.00	3.82	0.00
Average prediction per seconds and battery life in hours.																		
Predictions	4100			2700			330			715			442			67		
Battery Life	250			180			120			210			140			90		

complexity of the hypothesis space while, however, maintaining the capability of the trained models to generalize well on new and previously unseen data. The use of Local Rademacher Complexity plays a central role, as it allows to exclude from the hypothesis space those functions that will not be chosen by the learning procedure as the final model. The possibility, offered by the proposed approach, of more accurately shaping and designing the class of functions leads to remarkable positive outcomes when dealing with model accuracy, prediction rate, and battery lifespan in resource-limited devices like smartphones.

REFERENCES

[1] D. J. Cook and S. K. Das, "Pervasive computing at scale: Transforming the state of the art," *Pervasive and Mobile Computing*, vol. 8, pp. 22–35, 2012.

[2] R. A. Felicity, A. Eliathamby, H. L. Nigel, and G. C. Branko, "Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models," *Physiological Measurement*, vol. 27, no. 10, p. 935, 2006.

[3] V. N. Vapnik, *Statistical learning theory*. Wiley–Interscience, 1998.

[4] E. De Vito, L. Rosasco, A. Caponnetto, U. D. Giovannini, and F. Odone, "Learning from examples as an inverse problem," *Journal of Machine Learning Research*, vol. 6, pp. 883–904, 2005.

[5] S. Shalev-Shwartz and N. Srebro, "Svm optimization: inverse dependence on training set size," in *International Conference on Machine learning*, 2008.

[6] L. Bottou and O. Bousquet, "The tradeoffs of large-scale learning," *Optimization for Machine Learning*, pp. 351—368, 2011.

[7] D. J. Cook and S. K. Das, "Pervasive computing at scale: Transforming the state of the art," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 22–35, 2012.

[8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *Journal of Universal Computer Science*, vol. 19, no. 6, pp. 1295–1314, 2013.

[9] D. Anguita, A. Ghio, S. Pischiutta, and S. Ridella, "A support vector machine with integer parameters," *Neurocomputing*, vol. 72, no. 1, pp. 480–489, 2008.

[10] P. L. Bartlett, O. Bousquet, and S. Mendelson, "Local rademacher complexities," *The Annals of Statistics*, vol. 33, no. 4, pp. 1497–1537, 2005.

[11] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1926–1940, 1998.

[12] D. Anguita, A. Ghio, L. Oneto, and S. Ridella, "The impact of unlabeled patterns in rademacher complexity theory for kernel classifiers," in *Advances in Neural Information Processing Systems*, 2011.

[13] ——, "A support vector machine classifier from a bit-constrained, sparse and localized hypothesis space," in *IEEE International Joint Conference on Neural Networks*, 2013.

[14] B. Lesser, M. Mücke, and W. N. Gansterer, "Effects of reduced precision on floating-point svm classification accuracy," *Procedia Computer Science*, vol. 4, pp. 508–517, 2011.

[15] T. Luo, L. O. Hall, D. B. Goldgof, and A. Remsen, "Bit reduction support vector machine," in *IEEE International Conference on Data Mining*, 2005.

[16] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Trading convexity for scalability," in *International conference on Machine learning*, 2006.

[17] . CPLEX, "Ibm software group," *User-Manual CPLEX*, 2013.

[18] C. Orsenigo and C. Vercellis, "Discrete support vector decision trees via tabu search," *Computational statistics & data analysis*, vol. 47, no. 2, pp. 311–322, 2004.

[19] E. Alba, D. Anguita, A. Ghio, and S. Ridella, "Using variable neighborhood search to improve the support vector machine performance in embedded automotive applications," in *IEEE International Joint Conference on Neural Networks*, 2008.

[20] J. Manikandan, B. Venkataramani, and V. Avanthi, "Fpga implementation of support vector machine based isolated digit recognition system," in *IEEE International Conference on VLSI Design*, 2009.

[21] K. Irick, M. DeBole, V. Narayanan, and A. Gayasen, "A hardware efficient support vector machine architecture for fpga," in *IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2008.

[22] O. Pina-Ramfrez, R. Valdes-Cristerna, and O. Yanez-Suarez, "An fpga implementation of linear kernel support vector machines," in *IEEE International Conference on Reconfigurable Computing and FPGA's*, 2006.

[23] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[24] N. D. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "Bewell: A smartphone application to monitor, model and promote wellbeing," in *International Conference on Pervasive Computing Technologies for Healthcare*, 2011.