On the Optimality of Support Vector Machines for Channel Decoding

Gastón De Boni Rovella[†‡] , Meryem Benammar[†]
[†] *ISAE-SUPAERO, University of Toulouse*
[‡] *TéSA Research Lab*, Toulouse, France
{gaston.de-boni-rovella,meryem.benammar}@isae-supaero.fr

Hugo Méric
CNES,
Toulouse, France
hugo.meric@cnes.fr

Tarik Benaddi
Thales Alenia Space,
Toulouse, France
tarik.benaddi@thalesaleniaspace.fr

***Abstract*—In this work, we investigate the construction of channel decoders based on machine learning solutions, and more specifically, Support Vector Machines (SVM). The channel decoding problem being a high-dimensional multiclass classification problem, previous attempts were made in the literature to construct SVM-based channel decoders. However, existing solutions suffer from a dimensionality curse, both in the number of SVMs involved –which are exponential in the block length– and in the training dataset size. In this work, we revisit SVM-based channel decoders by alleviating these limitations and prove that the suggested SVM construction can achieve optimal Bit Error Probability (BEP) by attaining the performance of the bit-Maximum A Posteriori (MAP) decoder in the Additive White Gaussian Noise (AWGN) channel.**

***Index Terms*—SVM, channel coding, Maximum A Posteriori decoding, machine learning, communication systems.**

I. Introduction

Since the deployment of 5G, there has been an increasing interest in the beyond-5G and 6G communication technologies, with a particular emphasis on the usage of machine learning both at the physical layer and at the above layers of the communication protocol stack [1]–[3]. The introduction of machine learning aims at handling complex and time-consuming communication problems in a data-based approach, as opposed to the traditional model-based approach [4]. This alleviates the limitations of sub-optimal mathematical modeling and relegates the traditional online complexity to an offline, possibly complex, training procedure.

Of particular interest in this work is the physical layer Forward Error Correction (FEC), and more specifically, the decoding operation at the receiver which can be described as a high-dimensional classification problem. The design of low complexity and low latency decoding solutions for a given error correction code based on machine learning dates back to the 80s [5]–[7]. However, with recent advances in computer science and computing power, their interest has increased dramatically ever since but is often directed towards deep learning solutions (neural networks [8]–[10], attention-based networks [11], etc.). In this work, we investigate an alternative to neural networks for high-dimensional classification, namely, Support Vector Machines (SVM).

SVMs were introduced in the 90s in [12], [13], and present three main advantages with respect to deep neural network solutions: (i) they are maximum margin classifiers and are thus more robust to mismatches between training and application channel conditions [14]; (ii) the optimization algorithm to be solved is convex and therefore converges to a global minimum and; (iii) due to the nature of the optimization problem, there is very little risk of overfitting. This motivated the application of SVMs to channel decoding as was undertaken in [15]–[17]. However, the therein *one vs. one* and *one vs. rest* suggested approaches produce a number of SVMs exponential in the block length, and hence, become quickly intractable when the latter increases to more than a few bits. This is because these methods need to produce at least one decision function per valid codeword, and the number of possible codewords increases exponentially with the size of the code.

In this work, we revisit SVM-based channel decoding by means of (i) suggesting a bit-wise approach that reduces the number of SVMs necessary for decoding from exponential to linear in the number of information bits; (ii) reducing the complexity of the optimization process by reducing the size of the training dataset to a single codeword per class and; (iii) proving that the suggested SVM bit-wise classifier can attain the optimal bit-wise Maximum A Posteriori (MAP) decoding over an Additive White Gaussian Noise (AWGN) channel.

The remainder of this work is organized as follows. Section II describes the system model and the channel decoding problem. Section III gives an overview on the use of SVMs for binary classification and methods for non-linear separability of the data, along with previous works on SVM decoding. In section IV we introduce our solution and its advantages compared to previous works, and investigate the optimality of SVM for channel decoding. In section V, we illustrate our results with numerical simulations and analyze the complexity and robustness of the system. Conclusions and future directions are drawn in section VI.

Notations: Upper-case Roman and bold letters (e.g. X and $\boldsymbol{X}$) will denote random variables and vectors, and lower-case (e.g. x and $\boldsymbol{x}$) their realizations. $\mathbb{P}(\cdot)$ represents the event probability, $\mathbb{1}(\cdot)$ the Boolean indicator operator and i.i.d. means independent and identically distributed. $\mathcal{R}(y)$ and $\mathcal{I}(y)$ denote respectively the real and imaginary parts of $y \in \mathbb{C}$; $[\boldsymbol{a}, \boldsymbol{b}]$ denotes the concatenation of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, while $\mathbf{I}_n$ is the identity matrix of size n. $|\mathcal{S}|$ denotes the cardinality

Identify applicable funding agency here. If none, delete this.

of the set $\mathcal{S}$ while $\{1...k\}$ denotes the integers from 1 to k.

II. Problem Statement

A. The channel coding problem

Let us consider a coded and modulated transmission over an AWGN channel as depicted in Figure 1.

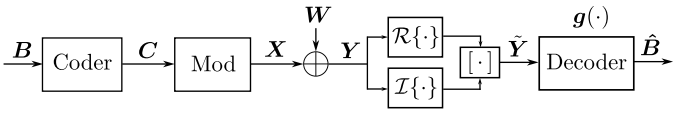

Fig. 1. The system model.

In such a setting, a random binary input message $\boldsymbol{B} \in \{0,1\}^k$ consisting in k independent bits uniformly distributed, is mapped through an (n,k) linear block error correction code into a codeword $\boldsymbol{C} \in \{0,1\}^n$ of n bits. The obtained codeword is mapped through a linear modulation scheme (e.g. PSK or QAM) of order q, thus yielding a complex vector $\boldsymbol{X} \in \mathbb{C}^{n'}$, where $n' = n/q$. The channel output $\boldsymbol{Y}$ results from the transmission of the input $\boldsymbol{X}$ through a memoryless AWGN channel, i.e., $\boldsymbol{Y} \triangleq \boldsymbol{X} + \boldsymbol{W}$ where $\boldsymbol{W} \overset{\text{i.i.d}}{\sim} \mathcal{CN}(0, \sigma^2 \mathbf{I}_{n'})$.

In order to feed real-valued vectors to the decoder, we first perform a preprocessing stage of the complex-valued channel output $\boldsymbol{Y}$,

$$\tilde{\boldsymbol{Y}} \triangleq [\,\mathcal{R}(\boldsymbol{Y}),\, \mathcal{I}(\boldsymbol{Y})\,], \tag{1}$$

which yields a real-valued signal $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{2n'}$. This signal is then fed to a decision rule (decoder) $\boldsymbol{g}(\cdot)$ which produces an estimate of the original message of k bits $\hat{\boldsymbol{B}} \triangleq \boldsymbol{g}(\tilde{\boldsymbol{Y}}) \in \{0,1\}^k$.

For a fixed (n,k) linear block code, *the channel decoding problem* consists in developing a decoding rule $\boldsymbol{g}(\cdot)$ that minimizes the average Bit Error Probability (BEP) defined by

$$P_e^b \triangleq \frac{1}{k} \sum_{j=1}^{k} \mathbb{P}(\hat{B}_j \neq B_j), \tag{2}$$

where B_j and $\hat{B}_j$ are binary random variables corresponding to the jth element of the message $\boldsymbol{B}$ and estimated message $\hat{\boldsymbol{B}}$, respectively.

B. The bit-MAP decoding rule

The theoretically optimal decoding rule $\boldsymbol{g}(\cdot)$ from a BEP point of view is the so-called bit-Maximum A Posteriori (bit-MAP) decoding rule which we state in the following lemma.

Lemma 1 (Bit-MAP decoding rule)**.**
The optimal decoding rule for the BEP defined in (2) *is given by the concatenation of k bit-MAP decoding rules $\{g^{(j)}(\cdot)\}_{1\leq j\leq k}$ where, for all $j \in \{1...k\}$*

$$g^{(j)}(\boldsymbol{y}) \triangleq \underset{b\in\{0,1\}}{\textit{argmax}}\ \mathbb{P}(B_j = b|\boldsymbol{Y} = \boldsymbol{y}). \tag{3}$$

Proof. The proof follows from classical tools from detection theory and is omitted here due to space limitation. □

The bit-MAP decoding rule can be further simplified under the assumption of i.i.d. and uniformly distributed input bits.

Lemma 2 (Bit-MAP simplification)**.**
Under the assumption of i.i.d. and uniformly distributed bits $\{B_j\}_{1\leq j\leq k}$, the bit-MAP decoding rule can be simplified as

$$g^{(j)}(\boldsymbol{y}) = \mathbb{1}\left\{ \sum_{\substack{\boldsymbol{b}/\\ b_j=1}} P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}(\boldsymbol{b})) > \sum_{\substack{\boldsymbol{b}/\\ b_j=0}} P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}(\boldsymbol{b})) \right\}, \tag{4}$$

where $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})$ denotes the channel conditional probability density function (p.d.f.) and $\boldsymbol{x}(\boldsymbol{b})$ denotes the modulated codeword associated with the message $\boldsymbol{b}$.

Proof. Let $\boldsymbol{y}$ be a received sequence Let us assume $\boldsymbol{y}$ has a non-zero marginal p.d.f., i.e., $P_{\boldsymbol{Y}}(\boldsymbol{y}) \neq 0$. Then, we have by definition for all $j \in \{1...k\}$,

$$g^{(j)}(\boldsymbol{y}) = \begin{cases} 1 & \text{if} \quad \dfrac{\mathbb{P}(B_j = 0|\boldsymbol{Y} = \boldsymbol{y})}{\mathbb{P}(B_j = 1|\boldsymbol{Y} = \boldsymbol{y})} < 1 \\ 0 & \text{else} \end{cases}. \tag{5}$$

Next, by observing that for $b \in \{0,1\}$,

$$\mathbb{P}(B_j = b|\boldsymbol{Y} = \boldsymbol{y}) = \sum_{\boldsymbol{b}/b_j=b} \mathbb{P}(\boldsymbol{B} = \boldsymbol{b}|\boldsymbol{Y} = \boldsymbol{y}) \tag{6}$$

$$= \frac{2^{-k}}{P_{\boldsymbol{Y}}(\boldsymbol{y})} \sum_{\boldsymbol{b}/b_j=b} P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}(\boldsymbol{b})) \tag{7}$$

where we have used Bayes' identity along with the assumption of i.i.d. uniformly distributed input bits $\mathbb{P}(\boldsymbol{B} = \boldsymbol{b}) = 2^{-k}$, and the fact that the encoding and modulations are one-to-one mappings, i.e., $P_{\boldsymbol{Y}|\boldsymbol{B}}(\boldsymbol{y}|\boldsymbol{b}) = P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}(\boldsymbol{b}))$. The proof of the lemma follows from simplifying the likelihood ratio in (5) and rewriting it using an indicator operator. □

Under the formulation in (4), the summations imply a marginalization over binary sequences of length $k-1$, rendering the complexity of the bit-MAP decoder exponential in k. Hence, for generic codes for which the marginalization cannot be done on graphs or trellises, the bit-MAP remains an impractical algorithm.

In this work, we will investigate the construction of SVM-based decoders and study their connection to the MAP decoding rule in the case of AWGN channels.

III. Preliminaries on SVM

A. Linearly separable data

Let us consider a labeled dataset $\{\tilde{\boldsymbol{y}}_i, l_i\}_{1\leq i\leq N}$ which consists of N vectors $\tilde{\boldsymbol{y}}_i \in \mathbb{R}^{2n'}$ with their respective labels $l_i \in \{-1,+1\}$. Let the class $\mathcal{C}_0$ (resp. $\mathcal{C}_1$) be the set of vectors $\tilde{\boldsymbol{y}}_i$ for which $l_i = -1$ (resp. $l_i = +1$). The dataset is said to be linearly separable if there exists $\boldsymbol{\xi} \in \mathbb{R}^{2n'}$ and $\nu \in \mathbb{R}$ that define a hyperplane $\mathcal{P} \in \mathbb{R}^{2n'}$

$$\mathcal{P} : \{\tilde{\boldsymbol{y}} \in \mathbb{R}^{2n'} \mid f(\tilde{\boldsymbol{y}}) = 0\}, \tag{8}$$

where $f(\tilde{\boldsymbol{y}}) = \boldsymbol{\xi}^T \tilde{\boldsymbol{y}} + \nu$, such that $f(\tilde{\boldsymbol{y}}) < 0$ for all $\tilde{\boldsymbol{y}} \in \mathcal{C}_0$ and $f(\tilde{\boldsymbol{y}}) > 0$ for all $\tilde{\boldsymbol{y}} \in \mathcal{C}_1$. The SVM principle consists in producing a hyperplane $\mathcal{P}$ that satisfies the *maximum margin property*, i.e., being at an equal and maximum distance from

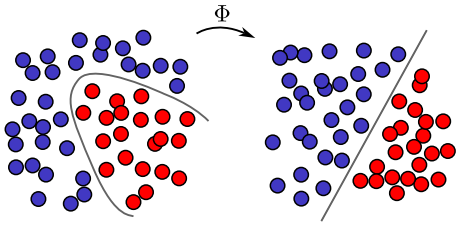

Fig. 2. Non-linear transformation of the data via a function Φ.

the nearest points of each class (the so-called *support vectors*). This hyperplane is given by the solutions $\boldsymbol{\xi}$ and ν of the following optimization problem,

$$\underset{\boldsymbol{\xi}}{\text{argmax}} \frac{1}{||\boldsymbol{\xi}||} \tag{9}$$

$$\text{s.t.} \min_{i=1,..,N} |\boldsymbol{\xi}^T \tilde{\boldsymbol{y}}_i + \nu| = 1. \tag{10}$$

Finally, $\tilde{\boldsymbol{y}} \in \mathcal{C}_1$ if $f(\tilde{\boldsymbol{y}}) > 0$, and $\tilde{\boldsymbol{y}} \in \mathcal{C}_0$ otherwise. See [18] for complete proof of this result. In sum, learning a classifier from the labeled dataset amounts to learning a *decision function* $f(\cdot)$ such that for all $\tilde{\boldsymbol{y}} \in \mathbb{R}^{2n'}$,

$$\begin{cases} f(\tilde{\boldsymbol{y}}) > 0 \implies \tilde{\boldsymbol{y}} \in \mathcal{C}_1 \\ f(\tilde{\boldsymbol{y}}) \leq 0 \implies \tilde{\boldsymbol{y}} \in \mathcal{C}_0. \end{cases} \tag{11}$$

B. Linearly non-separable data

If the two classes are not linearly separable, we need to resort to *kernel methods* in which the data points $\tilde{\boldsymbol{y}}$ are projected into a high-dimensional space via a non-linear function $\Phi(\cdot)$ (see Figure 2), where they become linearly separable. Given this mapping function Φ, its associated kernel function is given by $K(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{y}}') = \langle \Phi(\tilde{\boldsymbol{y}}), \Phi(\tilde{\boldsymbol{y}}') \rangle$, where $\langle \, , \rangle$ denotes the inner product in the space defined by $\{\Phi(\tilde{\boldsymbol{y}}) : \tilde{\boldsymbol{y}} \in \mathbb{R}^{2n'}\}$.

Considering the kernel function $K(\cdot, \cdot)$ and the dataset $\{\tilde{\boldsymbol{y}}_i, l_i\}_{1 \leq i \leq N}$, SVM training consists in first solving the following optimization problem in $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)$:

$$\begin{aligned} &\underset{(\alpha_1,...,\alpha_N)}{\text{argmin}} \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j l_i l_j K(\tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_j) - \sum_{i=1}^{N} \alpha_i \\ &\text{subject to: } 0 \leqslant \alpha_i \leqslant C \text{ and } \textstyle\sum_{i=1}^{N} l_i \alpha_i = 0, \end{aligned} \tag{12}$$

where C is a relaxing hyperparameter to be chosen[1]. The obtained optimal parameters $\boldsymbol{\alpha}$ are then used to define the decision function $f(\cdot)$ in (11) as:

$$f(\tilde{\boldsymbol{y}}) = \sum_{i=1}^{N} l_i \alpha_i K(\tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}) + \nu, \tag{13}$$

where ν is computed directly using any of the *support vectors*, which are determined by the indices i for which $\alpha_i > 0$. In this work, we will select $C = +\infty$ –meaning no relaxation– and as a kernel, the well-known Radial Basis Function (RBF)

$$K(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{y}}') \triangleq e^{-\gamma ||\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{y}}'||^2}, \tag{14}$$

where γ determines the slope of the exponential and is a hyperparameter to be tuned.

[1]Readers interested in the full deduction are referred to [18].

C. SVM for decoding: previous works

Regardless of whether the dataset is linearly separable or not, SVMs are by nature binary classifiers. To build a decoding algorithm for a (n, k) linear block code, we need to resort to multiclass classification. Previous solutions [15]–[17] for SVM joint demodulation and decoding employ the so-called *one vs rest* and *one vs one* approaches [19].

The *one vs. rest* method is based on producing 2^k binary SVM decision functions $f^{(j)}$ for $j \in \{1, ..., 2^k\}$, each one isolating one class (i.e., one codeword) against all the others. All the SVMs are then applied to the received signal $\tilde{\boldsymbol{y}}$, and the selected class $\mathcal{C}_{j^\star}$ (codeword) is such that:

$$j^\star = \underset{j \in \{1,...,2^k\}}{\text{argmax}} \; f^{(j)}(\tilde{\boldsymbol{y}}). \tag{15}$$

This corresponds to the class that has the largest distance between the point $\tilde{\boldsymbol{y}}$ and the separating hyperplane. If no decision function $f^{(j)}(\tilde{\boldsymbol{y}})$ gives a positive outcome, then the nearest class is selected, i.e., $f^{(j)}(\tilde{\boldsymbol{y}})$ with the negative value closest to 0.

The *one vs. one* approach consists in producing an SVM decision function for each possible pair of classes –i.e. valid codewords– resulting in a total of $C_2^{2^k} = 2^{k-1}(2^k - 1)$ binary classifiers. For the final decision, a voting system is implemented, where each decision function decides between two classes (codewords), and the class that gets the most votes at the end is selected. Ties can be resolved either randomly or by considering the mean of the votes' reliabilities, i.e., the distance to the separating hyperplanes. More detailed descriptions of these methods are given in [19].

IV. Proposed solution

In the following, we describe our suggested solution, which combines bit-wise SVM with noiseless optimization.

A. Bit-wise SVM

The previous approaches present the main constraint of a complexity that increases exponentially in the message length, with at least 2^k SVMs for a code of size (n, k). To alleviate this constraint, we suggest a novel bit-wise approach that resorts to only k SVM classifiers for an (n, k) linear block code (see Figure 3). This method transforms the multiclass

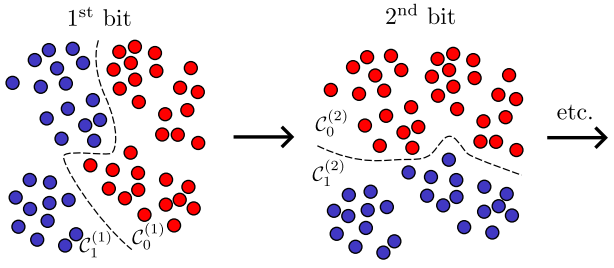

Fig. 3. Visual representation of the proposed bit-wise approach.

problem generating one SVM per valid codeword, into a series of k binary classifications necessitating one SVM per bit

position. To this end, for all $j \in \{1...k\}$, we divide the dataset $\tilde{\boldsymbol{y}} \in \{\tilde{\boldsymbol{y}}_1, ..., \tilde{\boldsymbol{y}}_N\}$ into two non-intersecting classes:

- $\mathcal{C}_1^{(j)}$ corresponding to the vectors for which the transmitted message $\boldsymbol{b}$ satisfies $b_j = 1$;
- $\mathcal{C}_0^{(j)}$ corresponding to the vectors for which $b_j = 0$.

Consequently, each decision function $f^{(j)}(\cdot)$, for $j \in \{1...k\}$, will decide whether the jth bit of the estimated message $\hat{\boldsymbol{b}}$ is a 0 or a 1. The suggested bit-wise SVM not only reduces the number of SVMs necessary from 2^k to k, but can be implemented in parallel in order to reduce latency.

B. Proposed training: noiseless optimization

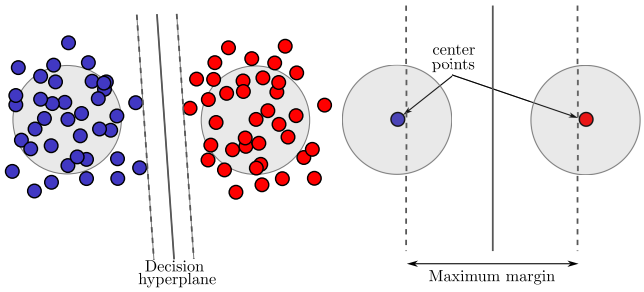

Fig. 4. Visual demonstration of noiseless optimization.

To further reduce the complexity with respect to the SVM-based solutions in [15]–[17] we make use of a particularly appealing feature of SVMs, namely, their maximum margin property, which yields separating hyperplanes that are equidistant from both dataset classes. As such, when investigating symmetric channel models like the AWGN, this is equivalent to a maximum margin classifier between *only* the original noiseless codewords (i.e. the classes' centroids). Consequently, rather than the traditional training approach which considers a dataset with randomly generated noisy codewords, it would suffice to optimize –or *train*– the suggested bit-wise SVM on only noiseless modulated codewords as shown in Figure 4.

The suggested noiseless optimization not only drastically reduces the size of the training dataset but also allows to be robust to possible mismatches between the training and actual channel conditions (SNR for instance).

C. Proposed optimization problem

Combining the two suggested elements (bit-wise SVM and noiseless optimization), the training dataset will be composed of the 2^k valid modulated codewords $\{\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_{2^k}\}$ with $2n'$ elements each –where the same preprocessing (1) has been applied to $\boldsymbol{x}$–, and k binary classifiers will be produced following the bit-wise approach of section IV-A. The classification of an unlabeled vector $\tilde{\boldsymbol{y}}$ consists of k classifiers $f^{(j)}(\tilde{\boldsymbol{y}})$, each determining the value of the jth bit of the estimated message $\hat{\boldsymbol{b}}$, and given by

$$f^{(j)}(\tilde{\boldsymbol{y}}) = \sum_{i=1}^{2^k} l_i^{(j)} \alpha_i^{(j)} K(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}) + \nu^{(j)}, \tag{16}$$

where $l_i^{(j)} = +1$ if the jth information bit of the modulated codeword $\boldsymbol{x}_i$ is 1, and $l_i^{(j)} = -1$ otherwise. Lastly, $\boldsymbol{\alpha}^{(j)}$ constitutes the solution to the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{argmin}} \ & \frac{1}{2} \sum_{i,m=1}^{2^k} \alpha_i \alpha_m l_i^{(j)} l_m^{(j)} K(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_m) - \sum_{i=1}^{2^k} \alpha_i \\ & \text{subject to: } \alpha_i \geqslant 0 \text{ and } \textstyle\sum_{i=1}^{2^k} l_i^{(j)} \alpha_i = 0. \end{aligned} \tag{17}$$

Each of the k bit-wise SVM optimization problems given in (17) can be written as a quadratic programming problem with linear constraints given by

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{argmin}} \ & \frac{1}{2} \boldsymbol{\alpha}^T Q^{(j)} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ & \text{subject to: } \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \boldsymbol{\alpha}^T \boldsymbol{l}^{(j)} = 0, \end{aligned} \tag{18}$$

where $Q^{(j)}$ is a matrix such that $Q_{i,m}^{(j)} = l_i^{(j)} l_m^{(j)} K(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_m)$. Since all $Q^{(j)}$ are definite positive matrices, these optimization problems are all convex.

D. Optimality analysis and interpretation

Although obtaining closed-form solutions of (17) for generic choices of the (n, k) linear block code, the constellation, and the parameter γ might be challenging, we show that under certain assumptions, the resulting SVM decision rule can be obtained in closed-form and related to the Bit-MAP decision rule.

Theorem 1 (Optimal solution and equivalence to bit-MAP)**.**

i) *For $\gamma \gg 1$, the optimal solution to* (17) *for all $j \in \{1...k\}$ is given by $\boldsymbol{\alpha}^* = (1, 1, ..., 1)$, and $\nu^* = 0$.*

ii) *Furthermore, if $\gamma = 1/\sigma^2$, this solution yields decision functions $f^{(j)}(\tilde{\boldsymbol{y}})$ equal to the bit-MAP decision rule $g^{(j)}(\tilde{\boldsymbol{y}})$ of Lemma 2.*

Proof. To prove i), let us notice that if $\gamma \gg 1$, then one can show that for all $i \neq m$, $K(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_m) \approx 0$. Hence, one can rewrite the objective function (18) for all $j \in \{1...k\}$ as

$$\frac{1}{2} \sum_{i=1}^{2^k} \alpha_i^2 - \sum_{i=1}^{2^k} \alpha_i = \frac{1}{2} \sum_{i=1}^{2^k} (\alpha_i^2 - 1)^2 - 2^{k-1}. \tag{19}$$

One can then easily show that the solution $\boldsymbol{\alpha}^* = (1, 1, ..., 1)$ yields a lower bound to the objective function, since $\sum_{i=1}^{2^k} (\alpha_i^2 - 1)^2 \geq 0$, and verifies the inequality constraints $\alpha_i \geq 0 \ \forall i \in \{1...2^k\}$. The equality constraint is also verified, since for all $j \in \{1...k\}$, each of the two classes $\mathcal{C}_1^{(j)}$ and $\mathcal{C}_0^{(j)}$ consist in 2^{k-1} sequences $\tilde{\boldsymbol{x}}_i$ and hence,

$$\sum_{i=1}^{2^k} l_i^{(j)} = |\mathcal{C}_0^{(j)}| - |\mathcal{C}_1^{(j)}| = 0 \quad \forall j \in \{1...k\}. \tag{20}$$

Thus, for all $j \in \{1...k\}$, $\boldsymbol{\alpha}^* = (1, 1, ..., 1)$ is the solution to the optimization problem in (17). As for $\nu^{(j)}$, note that by evaluating (16) for any $\tilde{\boldsymbol{x}}_m$ using $\gamma \gg 1$, we obtain

$$f^{(j)}(\tilde{\boldsymbol{x}}_m) = \sum_{i=1}^{2^k} l_i^{(j)} e^{-\gamma ||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_m||^2} + \nu^{(j)} = l_m^{(j)} + \nu^{(j)}, \tag{21}$$

which yields $\nu^{(j)} = 0$.

To prove ii), replacing the proposed solution $\{\boldsymbol{\alpha}^* = (1, 1, ..., 1), \nu^* = 0\}$ in (13) yields

$$f^{(j)}(\tilde{\boldsymbol{y}}) = \sum_{i=1}^{2^k} l_i^{(j)} e^{-\gamma||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{y}}||^2}. \tag{22}$$

Let us divide the summation argument into the two classes $\mathcal{C}_1^{(j)}$ and $\mathcal{C}_0^{(j)}$. Hence, we can rewrite the decision function as

$$f^{(j)}(\tilde{\boldsymbol{y}}) = \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{C}_1^{(j)}} e^{-\gamma||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{y}}||^2} - \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{C}_0^{(j)}} e^{-\gamma||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{y}}||^2}. \tag{23}$$

From (23), it is easy to deduce the value of the jth bit as

$$g^{(j)}(\tilde{\boldsymbol{y}}) = \mathbb{1}\left\{ \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{C}_1^{(j)}} e^{-\gamma||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{y}}||^2} > \sum_{\tilde{\boldsymbol{x}}_i \in \mathcal{C}_0^{(j)}} e^{-\gamma||\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{y}}||^2} \right\} \tag{24}$$

Selecting $\gamma = 1/\sigma^2$ in (24) yields the bit-MAP rule in (4) for an AWGN channel with noise power σ^2 by noticing that

$$P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}(\boldsymbol{b})) = \frac{1}{(\pi\sigma^2)^{n'}} e^{-\frac{||\boldsymbol{x}(\boldsymbol{b}) - \boldsymbol{y}||^2}{\sigma^2}}. \tag{25}$$

□

In the following, we will show that the assumption of $\gamma \gg 1$ in Theorem 1, is valid even for moderate SNR values. Besides, we will analyze the effect of relaxing the constraint $\gamma = 1/\sigma^2$ on the obtained results with respect to the bit-MAP.

V. Numerical results and analysis

A. Effect of the hyperparameter γ

The suggested bit-wise SVM was implemented for both an extended $(32, 11)$ BCH code and a $(32, 11)$ polar code [20], each under a 16-QAM modulation scheme and an AWGN channel with an $E_b/N_0 = \frac{n}{k.q}\frac{1}{\sigma^2}$. Let us define γ_s the value of the exponential's slope:

$$\gamma_s = 1/\sigma_s^2, \tag{26}$$

where σ_s^2 is the noise power such that $E_b/N_0 = s\,\text{dB}$. This is what is referred to as a value of γ *adapted* to a normalized signal-to-noise ratio of $E_b/N_0 = s\,\text{dB}$. In the following, we will distinguish two training scenarios. In the first scenario, the choice of γ in the RBF kernel is adapted to each E_b/N_0, i.e., $\gamma = 1/\sigma^2$, where σ^2 is the noise power corresponding to each E_b/N_0 ratio. In the second scenario, s is set to 0, i.e. $\gamma = \gamma_0$, for all values of E_b/N_0.

The Bit-Error Rate (BER) curves of both the suggested solution (bit-wise SVM) and the optimal solution (bit-MAP) are given in Figure 5. We observe that, for the first scenario, by adapting the value of γ to the corresponding E_b/N_0, the SVM decoder is matched to the bit-MAP decision rule, and so their performances coincide as per Theorem 1. However, for the second scenario in which $\gamma = \gamma_0$, the resulting SVM curve degrades in the high E_b/N_0 (low-noise) regime.

To assess the effect of the choice of the parameter γ in the second scenario, Figure 6 shows the E_b/N_0 corresponding to a BER of 10^{-3} as a function of $s \in \{-2, ..., 15\}$ for both

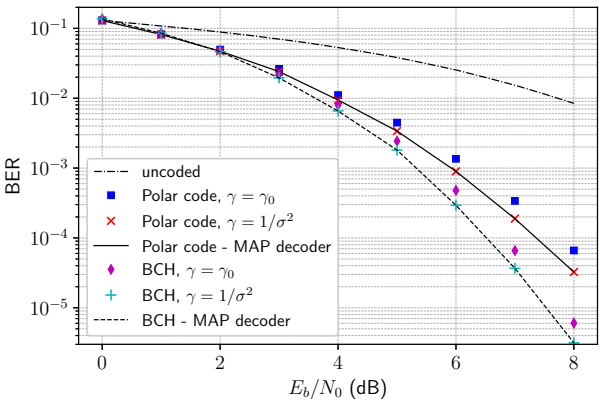

Fig. 5. BER of the suggested decoder for both a BCH and a polar code of size $(32, 11)$, with a 16-QAM modulation and under an AWGN channel.

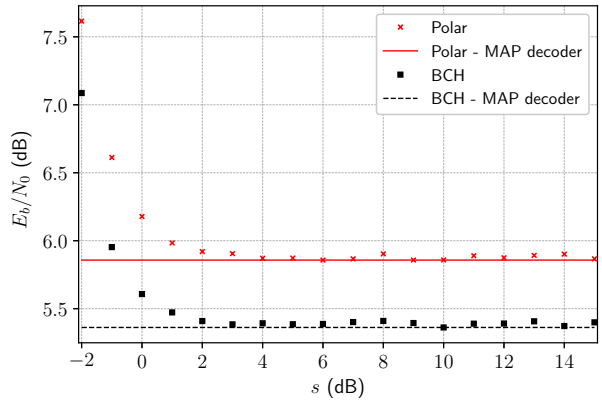

Fig. 6. E_b/N_0 corresponding to a BER of 10^{-3} as a function of s.

the $(32, 11)$ BCH code and the $(32, 11)$ polar code. One can observe that very low values of s –and thus their corresponding values of γ_s–, display poorer performances in terms of BER. However, above a threshold value of around 2dB, the SVM achieves the objective BER of 10^{-3} at essentially the same E_b/N_0 as the MAP decoding solution. This phenomenon suggests that training with large values of γ relaxes the need to adapt γ to the current E_b/N_0, generalizing in this way the result of Theorem 1, ii).

Moreover, as previously discussed, the result of Theorem 1, i) is valid for γ corresponding to even moderate values of E_b/N_0. Figure 7 shows the solution to the optimization problem (12) as a function of $s \in \{-2, ..., 15\}$. We observe that, indeed, the optimal solution to (17) is given by $\boldsymbol{\alpha}^* = (1, 1, ..., 1)$ and $\nu^* = 0$ for all $s > 2\text{dB}$, which corresponds to relatively moderate values of E_b/N_0.

B. Complexity analysis

Table I summarizes the decoding complexity of our method and those in the literature. As we can observe, the bit-wise approach is the first to enable a linearly growing number

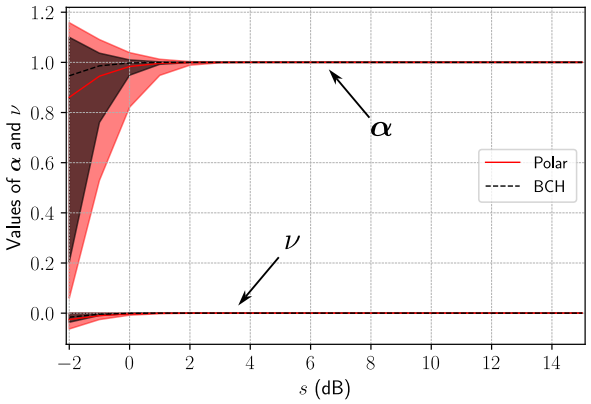

Fig. 7. Optimal values of $\boldsymbol{\alpha}$ and ν –i.e. solutions to the optimization problem (12)– as a function of the adapted γ.

of SVMs, which is more easily scalable than exponentially growing methods. The same goes for the dataset: for the SVM to learn a decision rule between two classes, it has to *see* at least one element of each class. With our noiseless optimization, the dataset size has been reduced to its minimum $N = 2^k$.

Nevertheless, complexity is not only based on the number of SVM classifiers but also on the number of operations required to perform each one of these classifications. Even with our method with reduced complexity, the size of the dataset is 2^k, with one element per valid codeword. This implies exponential growth, as the size of the dataset determines the number of terms in (13).

TABLE I
COMPLEXITY COMPARISON BETWEEN METHODS

	bit-wise	One vs rest	One vs one
# of SVM classifiers	k	2^k	$2^{k-1}(2^k-1)$
# of terms in (13)	N	N	$\approx \frac{N}{2^{k-1}}$
# of terms in (13) with noiseless opt.	2^k	2^k	2

VI. CONCLUSION

In this work, we investigated the design of SVM-based channel decoders for modulated communications over noisy channels. To this end, a novel bit-wise channel decoder was introduced, which significantly reduces complexity compared to previous solutions [15]–[17], both in the number of SVMs to be generated and the size of the dataset. Then, it was shown that the bit-wise SVM is equivalent to the bit-MAP decoder under the assumption of an RBF kernel and an AWGN channel model. The main limitation of the suggested solution is that being equivalent to the bit-MAP channel decoder, its complexity remains intractable for large codes. This is because all the points in the dataset are support vectors ($\alpha_i = 1 \; \forall i \in \{1, ..., 2^k\}$). Future works may explore a way of exploiting the structure of the particular code to reduce this complexity, or even its application to more complex channels that result in fewer support vectors to be evaluated.

REFERENCES

[1] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[2] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017.

[3] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[4] C. She, R. Dong, Z. Gu, Z. Hou, Y. Li, W. Hardjawana, C. Yang, L. Song, and B. Vucetic, "Deep Learning for Ultra-Reliable and Low-Latency Communications in 6G Networks," *IEEE Network*, vol. 34, no. 5, pp. 219–225, 2020.

[5] G. Zeng, D. Hush, and N. Ahmed, "An Application of Neural Net in Decoding Error-Correcting Codes," in *IEEE International Symposium on Circuits and Systems*, 1989.

[6] J. Bruck and M. Blaum, "Neural Networks, Error-Correcting Codes, and Polynomials over the Binary n-Cube," *IEEE Transactions on Information Theory*, vol. 35, no. 5, pp. 976–987, 1989.

[7] J. Yuan, V. Bhargava, and Q. Wang, "An Error Correcting Neural Network," in *Conference Proceeding IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1989.

[8] T. Gruber, S. Cammerer, J. Hoydis, and S. Brink, "On Deep Learning-based Channel Decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 2017, pp. 1–6.

[9] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to Decode Linear Codes using Deep Learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 341–346.

[10] A. Bennatan, Y. Choukroun, and P. Kisilev, "Deep Learning for Decoding of Linear Codes - A Syndrome-Based Approach," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, June 2018.

[11] Y. Choukroun and L. Wolf, "Error Correction Code Transformer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 695–38 705, 2022.

[12] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, jul 1992.

[13] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995.

[14] M.-G. Garcia, J. Rojo-Alvarez, F. Alonso-Atienza, and M. Martinez-Ramon, "Support Vector Machines for Robust Channel Estimation in OFDM," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 397–400, 2006.

[15] S. Akın, M. Penner, and J. Peissig, "Joint Channel Estimation and Data Decoding using SVM-based Receivers," *ArXiv:2012.02523*, 2020.

[16] J. W. H. Kao and S. M. Berber, "Error Control Coding based on Support Vector Machine," *Proc. 1st IAPR Worksh. Cogn. Inform. Process.*, pp. 182–187, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:3002188

[17] V. Sudharsan and B. Yamuna, "Support Vector Machine Based Decoding Algorithm for BCH Codes," *Journal of Telecommunications and Information Technology*, 2016.

[18] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H. Lin, *"Learning from data"*. AMLBook New York, 2012, vol. 4.

[19] C. Hsu and C. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, March 2002.

[20] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.