УДК 621.391.(075.8)

ANALYSIS OF SUPPORT VECTOR MACHINE BASED DECODING ALGORITHM FOR BOSE-CHAUDHURI-HOCQUENGHEM CODES



***M.I. Galchenko*[1]**
System analyst at HiQo Solutions, LLS

***Y.Y. Ramanouski*[2]**
System analyst at HiQo Solutions, Ltd

***N.V. Spichekova*[2]**
System analyst at HiQo Solutions, Ltd

[1]HiQo Solutions, LLS, Russia, Saint-Petersburg
E-mail: maxim.galchenko@hiqo-solutions.com
[2]HiQo Solutions, Ltd, Republic of Belarus, Minsk
E-mail: yury.romanovskiy@hiqo-solutions.com
E-mail: natalia.spichekova@hiqo-solutions.com

M. I. Galchenko

Graduated from the St Petersburg University. A postgraduate student at the Saint-Petersburg State Agrarian University. Work as a system analyst at HiQo Solutions, LLS, Russia, Saint-Petersburg. Research interests include medical statistics, time series forecasting in electrical energy industry.

Y. Y. Ramanouski

Graduated from the Belarusian State University. Master of science. Work as a system analyst at HiQo Solutions, Ltd, Republic of Belarus, Minsk. Fields of research are data analysis, artificial neural networks, time series forecasting.

N.V. Spichekova

Graduated from the Belarusian State University. Doctor of Philosophy (Ph.D.). Work as a system analyst at HiQo Solutions, Ltd, Republic of Belarus, Minsk. Areas of focus are data analysis and machine learning.

Abstract. In modern communication systems data is exchanged over unreliable noisy channels that cause the data to be altered by errors. To combat the problem, specially designed error correcting codes are applied to transmit the message over the channel. Bose-Chaudhuri-Hocquenghem (BCH) is a class of codes that are widely applied in practice. Apart from the classical decoding algorithms developed in the latter half part of the 20th century, a number of alternative approaches have been proposed. Among other things, it has been suggested to apply Support Vector Machine (SVM), machine learning classification and regression technique, to decode BCH codes. In this paper characteristics of the SVM based decoding approach is analyzed. Properties of two its modifications are considered.

Keywords: BCH codes, syndrome norm, multi-class classification, Support Vector Machine.

*Introduction***.** Ongoing exchange of information is essential for the existence of the modern human society. Increase in data being transferred and constantly growing variety of communication systems require reliable communication over unreliable noisy channels. Digital TV and phone, mobile and space communication systems are based on synchronous correction of errors with error-correcting codes.BCH codes is a powerful class of error correcting cyclic block codes that are widely used in communication systems, storage devices, and consumer electronics. BCH codes have good error-correcting capability and hold a prominent place in both theory and practice of multiple-error correction.

There are many algorithms for decoding BCH codes. Peterson-Gorenstein-Zierler [1] and Berlekamp-Massey [2] algorithms are examples of traditional syndrome-based decoding algorithms. They rely on properties of the BCH code. Chase decoding algorithm [3] utilizes channel measurement information in addition to the conventional use of the algebraic properties of the code.

Over the last years, alternative approaches to decode BCH codes, in which the decoder is a machine learning algorithm, have emerged. In [4] Support Vector Machine (SVM), a classification technique, is used to decode convolutional codes. In [5] neural networks are applied to decode linear codes.

In this paper properties of SVM based approach to decode BCH codes proposed in [6] are analyzed. Number of support vectors is given, multiplicity of errors is calculated. Properties of two modifications of the initial SVM based decoding algorithm are considered.

BCH codes and channel model. In this paper a communication system that uses a binary BCH (n,k,d) code C_d to transmit a message over an additive white Gaussian noise (AWGN) channel is considered. Here n denotes the codeword length, k denotes number of bits in each message word, d denotes the minimum distance between codewords. Each message word $\bar{u}$ is a $k-$bit message $\bar{u} \in \{0,1\}^k$. There are 2^k distinct messages in total. The transmitter encodes a message word $\bar{u}$ by appending $n-k$ redundant check bits to $\bar{u}$. As a result an $n-$bit codeword $\bar{c}$ is obtained. The encoding is performed by multiplying $\bar{u}$ by a generator matrix $G \in \{0,1\}^{k\times n}$: $\bar{c} = \bar{u}G \in \{0,1\}^n$. Corresponding to the 2^k possible messages, there are 2^k distinct codewords. The codeword $\bar{c}$ is converted to a bipolar format $\bar{x} = 1-2\bar{c}$, $\bar{x} \in \{-1,1\}^n$, and transmitted over the channel. Because of the channel noise, the receiver receives a noisy signal $\bar{y} = \bar{x} + \bar{w} \in R^n$, where $\bar{w} \in R^n$ is a vector of AWGN channel noise having mean 0 and variance σ^2. The purpose of the decoder, that is a special component of the receiver, is to find a bipolar estimation $\hat{x}$ of $\bar{x}$ from $\bar{y}$. The decoded binary message $\hat{c}$ corresponding to $\hat{x}$ is equal to $\hat{c} = 0.5 - 0.5\hat{x}$. $\hat{c}$ can be presented as a sum modulo 2 of the initial codeword $\bar{c}$ and an error vector $\bar{e} \in \{0,1\}^n : \hat{c} = \bar{c} + \bar{e}$.

Instead of the generator matrix G a parity-check matrix $H \in \{0,1\}^{(n-k)\times n}$ can be used to define the code C_d. The vector $S(\hat{c}) = H\hat{c}^T$ is called the syndrome for the message $\hat{c}$. $S(\hat{c})$ is a zero vector if and only if $\hat{c}$ is a codeword. If $\hat{c}$ is not a codeword, its syndrome will be a non-zero vector. For $\hat{c} = \bar{c} + \bar{e}$ the syndrome $S(\hat{c})$ does not depend on the codeword $\bar{c}$. It is completely specified by the error vector $\bar{e}$. Computing the syndrome and comparing its with zero is the first step in the syndrome-based error-correcting procedure for BCH codes [7, sec 6.2].

In this paper a case of $d = 5$ and $n = 2^m - 1$ is under consideration. The parity-check matrix of the code C_5 can be presented [2] in the from $H = [\alpha^i, \alpha^{3i}]^T$, $0 \le i \le n-1$, where α is a root of a primitive irreducible over $GF(2)$ polynomial of a degree m, $m \ge 3$. The syndrome $S(\hat{c})$ has the form $S(\hat{c}) = (s_1, s_2)^T$, where syndrome components s_1, s_2 belong to the Galois field $GF(2^m)$ with 2^m elements. For the C_5 code $s_1 \ne 0$ for errors of weight 1 and 2, i.e. for errors that have correspondingly one or two non-zero coordinates. These non-zero coordinates equal to 1.

From the condition $d = 5$ it follows that syndromes for errors of weight 1 and 2 are pairwise distinct. Such errors can be corrected based on their syndrome values. Coordinates i and j of non-zero elements of the error vector $\bar{e}$ of weight 2 satisfy the following system of equations:

$$\begin{aligned} x + y &= s_1, \\ x^3 + y^3 &= s_2, \end{aligned} \qquad (1)$$

where $x = \alpha^i, y = \alpha^j$.

The decoder that uses syndrome-based decoding algorithms, such as the Peterson-Gorenstein-Zierler and Berlekamp-Massey algorithms, searches for the solution to the system (1). As soon as the solution is found, the error occurred is corrected by adding the error vector to the received word. But number of different syndromes that are to be distinguished by the decoder grows exponentially when the code length n increases. For example, for $n = 15, 31, 63, 127, 255, 1023$ there are correspondingly 120, 496, 2 016, 8 128, 32 640, 130 816, 523 776 different syndromes for errors of weight 1 and 2 for the code C_5 [8].

Theory of syndrome norms. Theory of syndrome norms allows to reduce number of syndromes selected by the decoder.

Let Γ be a group that includes all powers of the cyclic permutation σ that acts on each vector $\bar{x} = (x_1, x_2, \ldots, x_n)$ in the following way: $\sigma(\bar{x}) = (x_n, x_1, x_2, \ldots, x_{n-1})$. For any error vector $\bar{e} = (e_1, e_2, \ldots, e_n)$ its Γ-orbit is the set $< \bar{e} >_\Gamma = < \bar{e} > = \{\bar{e}, \sigma(\bar{e}), \ldots, \sigma^{\nu-1}(\bar{e})\}$, where ν is the least natural number such that $\sigma(\bar{e}) = \bar{e}$. ν possesses the property: $\nu = n$ or ν divides n. In the former case the Γ-orbit $< \bar{e} >$ has the maximum capacity and is called complete. In any cyclic code of length $n > 1$ there are n errors of weight 1 that form one complete Γ-orbit. In cyclic codes of length $n = 2\mu + 1$ there are $n(n-1)/2$ errors of weight 2 that are distributed among $(n-1)/2 = \mu$ complete Γ-orbits.

It can be easily checked that if

$$S(\bar{e}) = H\bar{e}^T = (s_1, s_2)^T,$$

then

$$S(\sigma(\bar{e})) = H\sigma(\bar{e})^T = (\alpha s_1, \alpha^3 s_2)^T.$$

The formula

$$N(S(\bar{e})) = \begin{cases} \dfrac{s_2}{s_1^3}, & s_1 \neq 0, s_2 \neq 0, \\ \infty, & s_1 = 0, s_2 \neq 0, \\ -, & s_1 = s_2 = 0, \end{cases} \qquad (2)$$

defines the norm of the syndrome $S(\bar{e})$ for the C_5 code.

As the syndrome norm is the same for all errors of a specific Γ-orbit, the syndrome norm is an invariant called the norm of the Γ-orbit.

Theorem 1 [8, p. 105; 9]. A BCH code C_d is able to decode any error vector from any set K of Γ-orbits of errors with pairwise distinct norms.

The theorem includes the set of errors that can be decoded by the conventional syndrome methods. This set is defined by the minimum distance d between codewords. But as capacity of syndrome norms is much higher, it is possible to add [8, sec. 4.6] additional Γ-orbits with distinct norms to this set.

The theory of syndrome norms is a basis for the norm-based approach to BCH codes decoding. Within the conventional syndrome decoding approach, the calculated syndrome of the error is searched for on the list of syndromes of errors that can be corrected. Within the norm-based approach, the calculated norm is searched for on the list of norms of Γ-orbits of errors that can be corrected. In comparison with the conventional approach, search in the norm-based approach is done in n times faster.

Theory of syndrome norms is presented in [8-10].

SVM-based decoding of BCH codes. The SVM [11] is a class of algorithms that can be used for both regression and classification problems. The SVM method conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimensional feature space through some non-linear mapping chosen a priori. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures high generalization ability of the algorithm.

Primarily the SVM method was constructed to solve the following binary problem given the training vectors $x_t \in R$, $t = 1,...,n$, and a vector $y \in \{1,-1\}^n$:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i,$$
$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, i = 1,...,n,$$

where ϕ is the kernel function used to map the training data to a higher dimensional space, C is the penalty parameter and ξ_i are the slack variables. The gaussian Radial Bias Function (RBF) $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|}$ is commonly used as a kernel function.

Minimizing the $\frac{1}{2} w^T w$ implies maximization of $\frac{2}{\|w\|}$, the margin between training data points belonging to different classes. The penalty term $C\sum_{i=1}^{n} \xi_i$ is necessary to reduce the number of training errors when the training data is not linearly separable.

Later the SVM method was generalized to be used for multi-class classification problem as well. One of the generalization technique is the one-vs-one method [12]. According to this method, $k(k-1)/2$ SVM models are trained for a multi-class problem with k classes. Each classifier is trained on data from two classes. Let (x_t, y_t), $t = 1,...,l$, where $x_t \in R$ and $y_t \in \{1,...,k\}$ is the class for x_t, be the training data. Then for training examples from the ith and jth classes the following binary classification problem is solved:

$$\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2} \left(w^{ij}\right)^T w^{ij} + C\sum_{t} \xi_t^{ij},$$
$$(w^{ij})^T \phi(x_i) + b^{ij} \geq 1 - \xi_t^{ij}, \quad if \quad y_t = i, \tag{3}$$
$$(w^{ij})^T \phi(x_i) + b^{ij} \leq -1 + \xi_t^{ij}, \quad if \quad y_t = j,$$
$$\xi_t^{ij} \geq 0.$$

After (3) is solved, $k(k-1)/2$ decision functions $(w^{ij})^T \phi(x) + b^{ij}$ are constructed. To define a class of a testing example x, the voting strategy can be applied: if $sign((w^{ij})^T \phi(x) + b^{ij})$ states that x in the ith class, then vote for the ith class is added by one. Otherwise, the jth is increased by one. Then it is said that x is in the class with the largest vote. If both classes have the same number of votes, x is said to belong to either ith or jth class.

Recently it has been proposed [6] to apply the SVM algorithm to decode BCH codes. The idea is the following. Each message word is considered as a class. There are $N = 2^k$ classes for a

BCH (n,k,d) code. Each coordinate of the received word is taken as a feature that defines the class y_i, $1 \le i \le N$, to which the received word belongs. The proposed SVM based decoding approach consists of two major phases: the training phase and the decoding phase.

During the training phase, suitable data is simulated to train the classifier. Each codeword is modulated and transmitted M times through an AWGN channel. For the training, the channel with SNR (Signal Noise Ratio) equal to 0 db is taken. The training data are used to construct one-vs-one SVM decoder. Within the decoder, each class of data is compared with another class of data and $N(N-1)/2$ classifiers are constructed. The soft margin constant C and kernel parameter γ of RBF are hyperparameters of the SVM algorithm. Cross-validation technique is used for parameter tuning. The idea of cross-validation is to divide training data into v equal subsets. The model is constructed using $v-1$ subsets as training data and tested on the one remaining subset. For each pair of (C,γ) the process is repeated v times. A pair of (C,γ) with the highest accuracy is taken as the optimal training parameter.

In the decoding phase, the received word is passed through the $N(N-1)/2$ classifiers. Output values of the decision functions determines the class to which the received word belongs.

The paper proves that BER (Bit Error Rate) for the SVM based decoder is significantly lower than for traditional decoding algorithms like Peterson-Gorenstein-Zierler, Berlekamp-Massey and Chase -2 algorithms.

Experimental results. To analyze the SVM based decoding algorithm described in the previous section and hereinafter called as the baseline SVM decoding algorithm, BCH (15, 7, 5) code has been considered. The same technique as in [6] has been used to prepare data and train classifiers. More specifically, all $N=128$ codewords of the code, each of length 15, have been generated. Then every codeword has been converted to the bipolar format and a received word has been generated on the basis of the AWGN channel model as a sum of the sent codeword c and channel noise *noise*. For a given SNR (specified in db), each coordinate of *noise* is a normally distributed random variable with 0 mean and standard deviation $\sigma = 10^{-\frac{SNR}{20}}$. The simulation has been performed using Python *numpy* and *pandas* libraries. *Scikit-learn*, a machine learning library for Python, has been used to train SVM classifiers with RBF kernal. One-vs-one schema has been applied to handle multiclass classification problem with 128 classes. To tune parameters of the SVM algorithm, a grid search over the prespecified set of values for C and γ parameters with 5-fold cross validation technique has been performed. Training has been done on received words simulated for $SNR=0$ db, each codeword repeated M times, $M=30, 60, 90.$ Testing has been performed on received words simulated for $SNR=1, 2, ..., 10,$ each codeword repeated $p=200$ times, 25 600 samples for each SNR. Table 1 summarizes number of support vectors for each class.

Table 1

Number of support vectors (SV).

M	Training set size	Baseline / Extended SVM decoding algorithm			HDD-SVM decoding algorithm		
		#SV for all classes	Min # SV for a class	Max #SV for a class	#SV for all classes	Min #SV for a class	Max #SV for a class
30	3840	3797	28	30	3840	15	46
60	7680	7678	59	60	7677	43	78
90	11520	11489	88	90	-	-	-

Form Table 1 it follows that total number of support vectors is very close to size of the training set. It means that almost all training samples are used when classification is performed for a new sample. As a result, the model has to store a large number of support vectors and coefficients. If codeword length n is rather big, it will result in a model of large size.

To compare the baseline SVM decoding algorithm with traditional Hard Decision Decoding (HDD) algorithms, a decoder based on theory of syndrome norms has been applied to the test set in the following way. First, for each of the received words $\hat{y}$ the estimated bipolar codeword $\hat{x}$ has been found by taking the hard decision $\hat{x} = sign(\hat{y})$. Then $\hat{x}$ has been converted to the binary format by the formula $\hat{c} = 0.5 - 0.5\hat{x}$ and the syndrome norm (2) has been calculated for $\hat{c}$. If the calculated value of syndrome norm is on the list L of syndrome norms of correctable errors , the error has been corrected. Only errors of weight 1 and 2 has been included in L, though L can be extended (see Theorem 1 and the following paragraph). Figure 1 illustrates that the performance of the baseline SVM decoding algorithm is higher than the performance of the HDD algorithm under consideration. It also shows that when M increases, BER decreases.

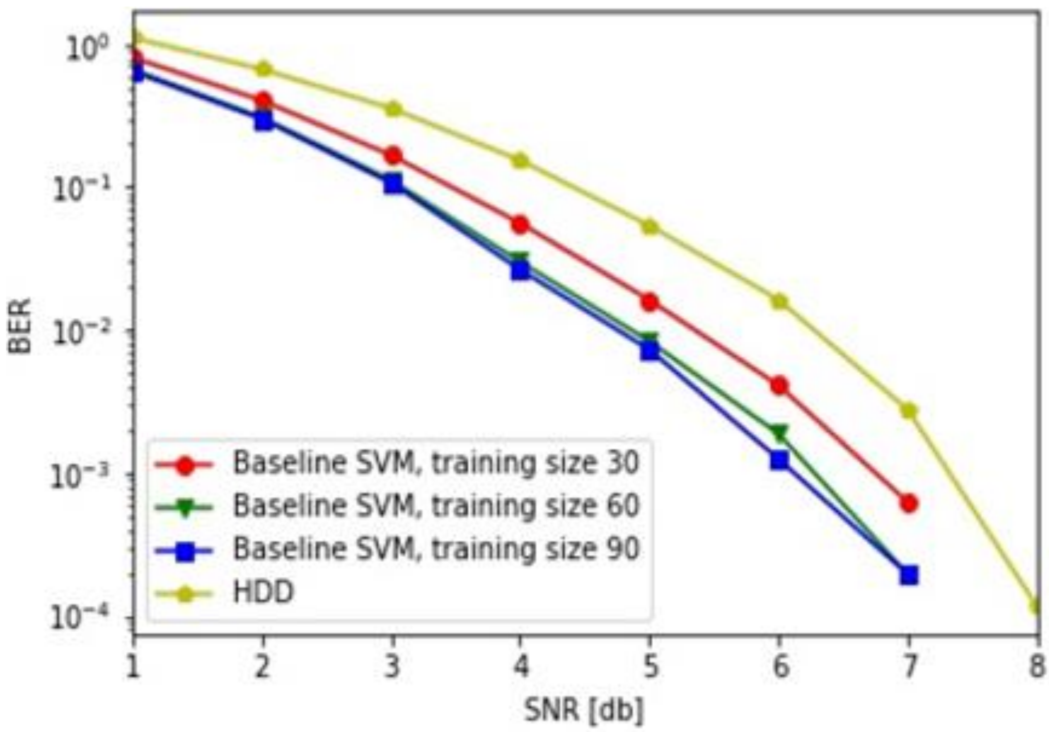

Figure 1: BER vs SNR plot of BCH (15, 7, 5) code.

For each vector from the test set, the sent and decoded messages have been compared bitwise and number of bits that were incorrectly decoded by the baseline SVM decoding algorithm (i.e. weight of an error) has been calculated. Minimum weight of an error is 5, maximum is 10. This is to say that errors committed by the baseline SVM decoding algorithm, have high weights. For a case of $M = 30, SNR = 1,$ number of received words decoded with an error of weight $w, w = 5, \ldots, 10,$ is presented in Table 2, section Baseline SVM algorithm.

Table 2

Weights of errors, $M = 30, SNR = 1$.

Error weight, w	# received words decoded with an error of weight w		
	Baseline SVM decoding algorithm	HDD-SVM decoding algorithm	Extended SVM decoding algorithm
5	2113	1631	2127
6	1306	1166	1301
7	228	183	225
8	52	49	53
9	25	23	23
10	4	3	4

The baseline SVM decoding algorithm doesn't have any knowledge about the code structure. As a result, the algorithm incorrectly decodes some of received words that are correctly decoded by the norm-based HDD algorithm described above. Number of such cases is tabulated in Table 3.

Table 3

Number of received words that are incorrectly decoded by the baseline / extended SVM algorithm and correctly decoded by the norm based HDD algorithm.

SNR	Baseline SVM decoding algorithm			Extended SVM decoding algorithm		
	$M = 30$	$M = 60$	$M = 90$	$M = 30$	$M = 60$	$M = 90$
1	700	385	353	692	383	347
2	437	244	231	433	240	224
3	244	113	105	248	113	102
4	114	40	42	111	39	41
5	32	11	11	32	11	10
6	10	1	0	7	1	0
7	2	0	0	2	0	0
8,9,10	0	0	0	0	0	0

To take advantage of BCH code structure knowledge, a two-step decoding algorithm, hereinafter called as the HDD-SVM decoding algorithm, has been tested. The decoding has been performed in the following way. First, the HDD decoder has been applied to each of the received words $\hat{y}$. Then the set S of words simulated for $SNR = 0$ db has been revised: a word $\hat{y}$ has been excluded from S if none of errors occurred or an error can be corrected by the HDD decoder, i.e. if the syndrome calculated for $\hat{y}$ by the HDD decoder is zero or on the list L of syndrome norms of errors that can be corrected by the HDD decoder. The remaining set has been used to train SVM classifiers. Training has been performed in the same manner as it has been done for the baseline SVM decoding algorithm except the fact that training samples have been taken from the revised set S . Two values for M has been considered: $M = 30$ and $M = 60$. The HDD-SVM algorithm has been tested on the same set as the baseline SVM algorithm.

Characteristics of the HDD-SVM algorithm are presented in Table 1, Table 2 and Figure 2.

From Table 1 it follows that there is no significant difference between number of support vectors for the baseline SVM and HDD-SVM algorithms. Like for the baseline SVM algorithm, minimum weight of an error committed by the HDD-SVM algorithm is 5, maximum is 10. Table 2 illustrates a decline in number of words decoded with an error of high weight. Figure 2 compares the performance of the HDD-SVM algorithm against the baseline SVM algorithm. At training size of 60, the baseline and HDD-SVM algorithms are found to have almost identical BER.

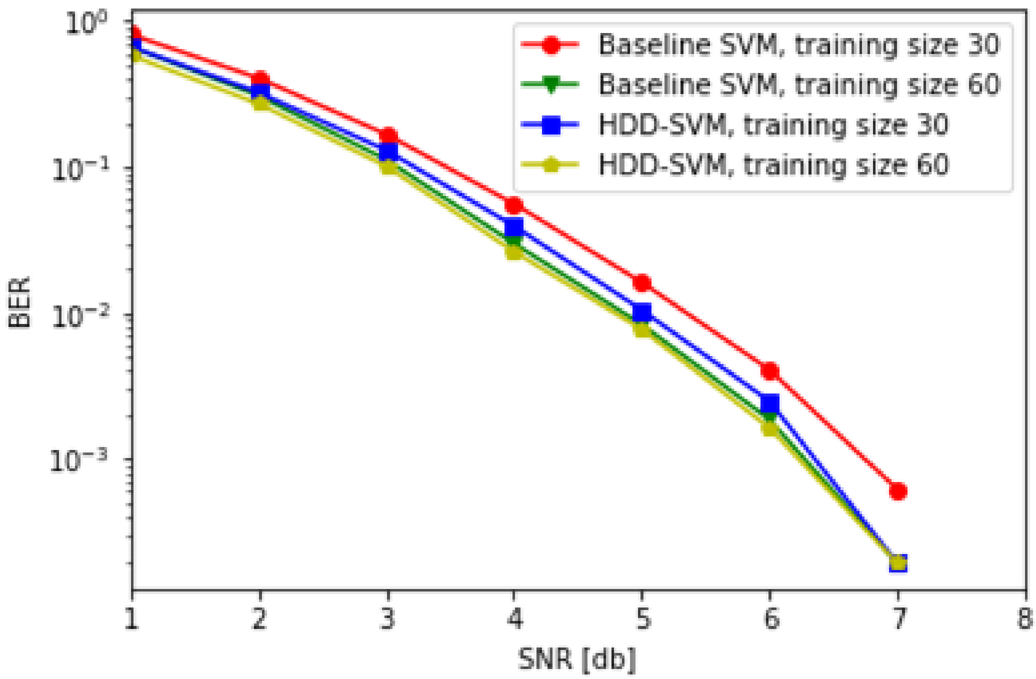

Figure 2: BER vs SNR plot of BCH (15, 7, 5) code.

To let the SVM algorithm “learn” the code structure, the syndrome norm has been added as an additional feature that defines the class to which the received word belongs. More specifically, for each received word $\hat{y}$ the syndrome norm has been calculated as in the HDD algorithm. Then one-hot encoding scheme has been applied to the received values. The one-hot encoding transforms any categorical attribute into binary features which can only contain 0 or 1. Number of resulting binary features depends on number of different values of the initial categorical attribute. After one-hot encoding has been done, the SVM classifiers has been trained on the extended set of features. Training has been performed in the same manner as it has been done for the baseline SVM decoding algorithm. Characteristics of the received algorithm, hereinafter called as the extended SVM decoding algorithm, are presented in Table 1, Table 2, Figure 3. It can be seen that though the extended SVM decoding algorithm uses additional features, its characteristics are almost identical to the characteristics of the baseline SVM decoding algorithm.

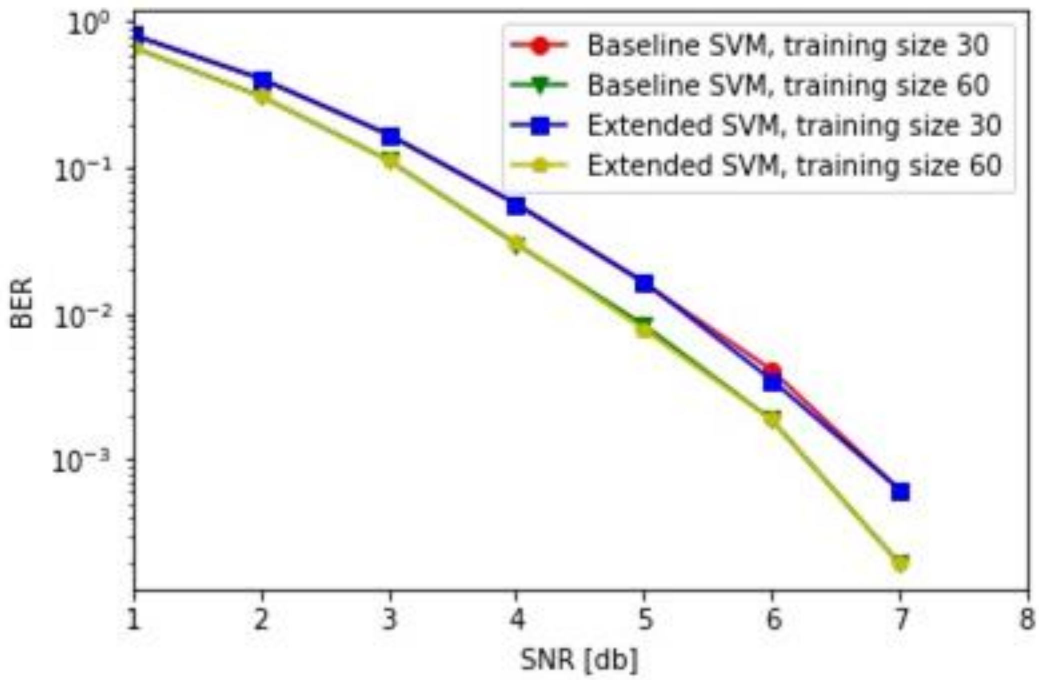

Figure 3: BER vs SNR plot of BCH (15, 7, 5) code.

Change in BER of SVM based decoding algorithms measured relative to BER of the HDD algorithm is presented in Table 4. If $SNR = 9$ or 10, BER is zero for all the algorithms.

Table 4

BER comparison (measured relative to BER of the HDD algorithm).

SNR	HDD	Baseline SVM, training size 30	Baseline SVM, training size 60	Baseline SVM, training size 90	Extended SVM, training size 30	Extended SVM, training size 60	ExtendedSVM, training size 90	HDD-SVM, training size 30	HDD-SVM, training size 60
1	1.111680	-27.34%	-40.1%	-41.84%	-27.31%	-40.02%	-41.73%	-40.05%	-47.72%
2	0.669063	-39.38%	-54.01%	-55.14%	-39.12%	-53.87%	-55.19%	-51.56%	-59.44%
3	0.357344	-53.09%	-68.57%	-70.03%	-52.77%	-68.69%	-69.81%	-63.57%	-71.67%
4	0.154453	-63.71%	-80.58%	-82.85%	-63.35%	-80.15%	-82.93%	-74.43%	-82.93%
5	0.053945	-69.88%	-84.5%	-86.46%	-69.44%	-85.66%	-86.82%	-80.52%	-85.66%
6	0.016328	-74.88%	-88.52%	-92.34%	-78.47%	-88.52%	-92.34%	-84.69%	-89.95%
7	0.002773	-77.46%	-92.96%	-92.96%	-77.46%	-92.96%	-92.96%	-92.96%	-92.96%
8	0.000117	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%

Conclusion. In this paper an analysis of the SVM based decoding algorithm for BCH code proposed in [6] has been performed. Two modifications of the initial algorithm has been considered. One of the modifications, the extended SVM decoding algorithm, uses the additional feature – the syndrome norm. Another one, the HDD-SVM based decoding algorithm, is a two-step decoding approach. At the first step, the HDD norm-based decoding algorithm is applied to correct errors of weight 1 and 2. At the second step the SVM based decoding algorithm is used. The SVM based decoding algorithm is capable of correcting errors of high weights that cannot be corrected by the conventional decoding algorithms. From Table 4 it follows that the HDD-SVM decoding algorithm demonstrates the biggest drop in BER measured relative to BER of the HDD algorithm. But this decrease in BER is archived by complication of the decoder.

References

[1]. Peterson, W.W. Error-correcting codes / W.W. Peterson, E.J. Walden. - MIT Press, 1972.

[2]. MacWilliams, F.J. The Theory of Error-Correcting Codes / F.J. MacWilliams, N.J.A Sloane. - New York, NY: North-Holland Publishing Company, 1977.

[3]. Chase, D. A Class of algorithms for decoding block codes with channel measurement information / D. Chase // IEEE Trans. Inform. Theory. - 1972. - Vol. 18, no. 1. - P. 170-182.

[4]. Kao, J. Error control coding based on support vector machine / J. Kao, S. Berber // Proc. 1st IAPR Worksh. Cogn. Inform. Process., Santorini, Greece. – 2008. – P. 182–187.

[5]. Nachmani, E. Deep learning methods for improved decoding of linear codes / E. Nachmani, E. Marciano, Loren Lugosch, W.J. Gross, D. Burshtein, Y. Be'ery [Electronic source] // https://arxiv.org/pdf/1706.07043.pdf - Accessed 15 January 2018.

[6] Sudharsan, V. Support vector machine based decoding algorithm for BCH codes / V. Sudharsan, B. Yamuna // Journal of Telecommunications and Information Technology. - 2016. - Vol. 2016. - P. 108-112.

[7]. Lin, S. Error control coding / S. Lin, D.J. Costello. - 2004. - 1272 p.

[8]. Lipnitski, V.A. Normennoe decodirovanie pomehoustoichivyh codov i algebraicheskie uravneniia [Norm-based decoding of error correcting codes and algebraic equations] / V.A. Lipnitski, V.K. Konopelko. [in Russian] - Minsk, 2007. - 239 p.

[9]. Konopelko, V.K. Teoriia norm sindromov v perestanovochnom dekodirovanii pomehoustoichivyh codov [The theory of syndrome norms in the permutation decoding action unjammable codes] / V.K. Konopelko, V.A. Lipnitski // Doklady BGUIR [in Russian].- 2004. - No. 2. - P. 146-157.

[10]. Lipnitski, V.A. Svoistva G-orbit dvoinyh oshibok i ih invariantov v BCH kodah [Properties of groups G of double errors and its invariants in BCH codes] / V.A. Lipnitski, A. Serada // System Analysis and Applied Information Science [in Russian]. - 2018. - No. 2. - P. 40-46.

[11]. Cortes, C. Support-vector netwoks / C. Cortes, V. Vapnik // Machine Learning. - 1995. - Vol. 20. - P. 273-297.

[12]. Hsu, C. W. A comparison of methods for multiclass support vector machines / C. W. Hsu, C. J. Lin // IEEE Trans. Neural Netw. - 2002. - Vol. 13. - No. 2. - P. 415–425.

АНАЛИЗ АЛГОРИТМА ДЕКОДИРОВАНИЯ БЧХ КОДОВ, ОСНОВАННОГО НА МЕТОДЕ ОПОРНЫХ ВЕКТОРОВ

М.И. ГАЛЬЧЕНКО[1]
Системный аналитик в ООО «ХайКво Солюшенс», Россия, Санкт-Петербург

Ю.Ю. РОМАНОВСКИЙ[2]
Системный аналитик в СООО «ХайКво Солюшенс», Республика Беларусь, Минск

Н.В. СПИЧЕКОВА[3]
Системный аналитик в СООО «ХайКво Солюшенс», Республика Беларусь, Минск

[1]ООО «ХайКво Солюшенс», Россия, Санкт-Петербург
E-mail: maxim.galchenko@hiqo-solutions.com
[2]СООО «ХайКво Солюшенс», Республика Беларусь, Минск
E-mail: yury.romanovskiy@hiqo-solutions.com
[3]СООО «ХайКво Солюшенс», Республика Беларусь. Минск
E-mail: natalia.spichekova@hiqo-solutions.com

Аннотация. В современных коммуникационных системах передача данных осуществляется по зашумленным каналам связи, что приводит к возникновению ошибок. Для борьбы с ошибками при передаче сообщений используются помехоустойчивые коды. Одним из классов кодов, которые нашли широкое применение на практике, являются коды Боуза-Чоудхури-Хоквингема (БЧХ-коды). Помимо классических декодирующих алгоритмов, разработанных во второй половине прошлого века, для декодирования БЧХ кодов был предложен ряд альтернативных подходов. Среди прочих – алгоритм декодирования, основанный на методе опорных векторов. В данной работе анализируются свойства алгоритма декодирования БЧХ-кодов, основанного на методе опорных векторов, а также двух предлагаемых в данной работе его модификаций.

Ключевые слова: Боуза-Чоудхури-Хоквингема коды, норма синдрома, многоклассовая классификация, метод опорных векторов.