Mitigation of Nonlinear Phase Noise in Single-channel Coherent 16-QAM Systems Employing Logistic Regression

Romulo de Paula

Universidade Estadual Paulista: Universidade Estadual Paulista Julio de Mesquita Filho

Lucas Marim

Universidade Estadual Paulista: Universidade Estadual Paulista Julio de Mesquita Filho

Rafael Penchel

Universidade Estadual Paulista: Universidade Estadual Paulista Julio de Mesquita Filho

Yessica Rumaldo Bustamante

CPQD

Marcelo Francisco Luis Abbade

Universidade Estadual Paulista: Universidade Estadual Paulista Julio de Mesquita Filho

Grethell Georgina Perez

Universidad Metropolitana

Ivan Aldaya (✉ ivan.a.aldaya@ieee.org)

Universidade Estadual de São Paulo <https://orcid.org/0000-0002-7969-3051>

Research Article

Keywords: Machine learning , Coherent systems , Nonlinearity compensation

Posted Date: June 4th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-499944/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Optical and Quantum Electronics on August 14th, 2021. See the published version at <https://doi.org/10.1007/s11082-021-03149-7>.

Mitigation of nonlinear phase noise in single-channel coherent 16-QAM systems employing logistic regression

Rômulo A. de Paula Jr. · Lucas Marim ·
Rafael A. Penchel · Yésica R. R.
Bustamante · Marcelo L. F. Abbade ·
Grethell Perez-Sanchez · Ivan Aldaya

Received: date / Accepted: date

Abstract We propose and analyze a classifier based on logistic regression (LR) to mitigate the impact of nonlinear phase noise (NPN) caused by Kerr-induced self-phase-modulation in digital coherent systems with single-channel unrepeatable links. Simulation results reveal that the proposed approach reduces the bit error ratio (BER) in a 100-km-long 16 quadrature amplitude modulation (16-QAM) system operating at 56-Gbps. Thus, the BER is reduced from $6.88 \cdot 10^{-4}$ when using maximum likelihood to $4.27 \cdot 10^{-4}$ after applying the LR-based classification, representing an increase of 0.36 dB in the effective Q-factor. This performance enhancement is achieved with only 624 operations per symbol, which can be easily parallelized into 16 lines of 39 operations.

Keywords Machine learning · Coherent systems · Nonlinearity compensation

1 Introduction

Nonlinear phase noise (NPN) caused by Kerr-induced self-phase modulation (SPM) is the dominant nonlinear distortion mechanism in single-channel unrepeatable links with coherent reception [1–3]. This is the case, for instance, of some coherent long-reach passive optical networks (LR-PONs) and high-capacity inter-datacenter links where linear impairments, i.e. chromatic dispersion (CD), polarization mode dispersion, and linear phase and frequency

Rômulo A. de Paula Jr, Lucas Marim, Rafael Abrantes Penchel, Marcelo L. F. Abbade, and Ivan Aldaya

Center for Advanced and Sustainable Technologies, São Paulo State University (UNESP),
São João da Boa Vista, SP, Brazil
E-mail: ivan.aldaya@unesp.br

Yesica R. R. Bustamante
CPqD Foundation, Campinas, Brazil.

Grethell Perez-Sanchez
Metropolitan Autonomous University, Mexico City, Mexico.

noises, are efficiently compensated by digital signal processing (DSP) schemes [4, 5]. The mitigation of SPM, nevertheless, still poses a significant challenge. SPM leads to a phase rotation in a differential length segment dL that depends on the power at position z and time t according to $d\phi_{NL}(z, t) = -\gamma P(z, t)dL$, being γ the nonlinear coefficient accounting for the influences of the nonlinear refractive index and the modal effective area [6]. In the absence of CD, the intersymbol interference can be neglected and, consequently, the phase rotation depends uniquely on the actual symbol power. In contrast, when CD is present, the instantaneous power depends not only on the actual symbol but also on the adjacent interfering symbols. This causes a word-dependent rotation that seems stochastic and hinders NPN compensation [7]. Since this nonlinear phase rotation can severely affect the system bit error ratio (BER), several machine learning techniques have been proposed to mitigate its impact. In [8] and [9], constellations affected by NPN are clustered using maximum expectation and histogram-based clustering, respectively. These clustering algorithms are interesting because they do not require any training sequences that reduce the data throughput. However, in highly static channels affected by impairments with a long coherence time, supervised algorithms can be employed with a small impact on the overall data transmission capacity. Several supervised approaches have been proposed to compensate the NPN. For instance, in [10], a k-nearest neighbors (KNN) algorithm is used to mitigate the effect of several impairments, including SPM. Another supervised approach denominated support vector machine (SVM) is employed in [11], [12], and [13]. In spite of their satisfactory results, both KNN and SVM suffer from high computational cost that makes their adoption in real-time implementations difficult. Several classification techniques, however, remain unexplored. One example is logistic regression (LR) that is extensively used in areas as geomorphology [14] and economy [15], but to the best of our knowledge, it has not been applied to nonlinear compensation.

In this work, we use multi-class LR to combat the NPN. The proposed solution presents a complexity as low as 624 floating point operations per symbol. In addition, its implementation is easily parallelizable, which further increases its potential for future real-time applications. The rest of the article is organized as follows: Section 2 describes the fundamentals of LR, including feature mapping to higher dimensional spaces and the extension to multi-class problems. In Section 3, the simulation setup is detailed, whereas in Section 4, performance and complexity results are presented. Finally, Section 5 concludes the paper.

2 LR for classification of constellations affected by NPN

Since this is, to the best of our knowledge, the first time LR is applied to the compensation of NPN, in this section we provide a detailed explanation of this approach [16, 17]. The fundamentals of binary LR for the single and multiple dimensional cases are firstly explained. Then, we discuss the case

where multiple classes are used. Figures illustrating examples for each LR classification process are used. The application of LR to NPN compensation problem is approached at the end of this section.

2.1 One-dimension binary logistic regression

Let us assume that we want to classify a set of symbols X using LR. We will first split these symbols in a training set $X_{tr} = \{x^{(i)}\}$ composed of m symbols with associated labels $Y = \{y^{(i)}\}$ and a test set $X_{test} = \{x^{(j)}\}$. The classification is then performed in two steps:

1) *Training stage*: During this stage, basic binary LR requires X_{tr} to be fitted by a sigmoid function (also known as logistic function), which is given by:

$$h_{\theta}(x^{(i)}) = \frac{1}{1 + \exp(-\theta x^{(i)})}, \quad (1)$$

Fig. 1 Data classification by logistic regression: (a) Classification of 1D data showing the fitted sigmoidal function and the threshold value. (b) 2D classification indicating the obtained boundary. (c) Complex boundary obtained after mapping to higher dimensional feature space. Solid line indicates optimum boundary obtained for $\lambda = 0.02$ and the gray and green discontinuous lines correspond to biased and overfitted cases, respectively. (d) Multi-class classification based on one-vs-all approach. Continuous lines represent the points where the hypothesis function of each binary LR equals 0.5 and each colored region represents the corresponding decision region.

where θ is the model parameter to be optimized and $h_\theta(\cdot)$ is the so-called hypothesis function. To find the best fitting parameter θ , a proper cost function must be adopted. Since the quadratic error is not convex in the case of binary LR, a different cost function needs to be considered. Typically, the cross-entropy function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left\{ \left(y^{(i)} - 1 \right) \ln(1 - h_\theta(x^{(i)})) - \left[y^{(i)} \ln(h_\theta(x^{(i)})) \right] \right\} \quad (2)$$

is employed [18]. In addition to its compact form, one of the advantages of the adoption of cross-entropy as the cost function is the simplified parameter optimization because it leads to the closed-form expression for updating θ . In this way, the value of the model parameter after the k -th iteration, $\theta[k]$, can be obtained from its previous value, $\theta[k-1]$, considering a learning rate α according to:

$$\theta[k] = \theta[k-1] - \frac{\alpha}{m} \sum_{i=1}^m [h(x^{(i)}) - y^{(i)}] x^{(i)}. \quad (3)$$

2) *Test stage*: Once the LR model has been properly trained, a new symbol $x^{(j)}$ can be classified by assigning the class $y = "1"$ if $h_\theta(x^{(j)}) > 0.5$ or to class $y = "0"$ if $h_\theta(x^{(j)}) \leq 0.5$, as can be seen in Fig. 1(a).

2.2 Two-dimension binary logistic regression

The LR classifier can be easily generalized to a higher dimensional input data space. In this case, θ becomes a vector composed of the weights of each feature (input variable). Figure 1(b) illustrates an example of the classification of a set of two-dimensional data employing LR. The resultant decision boundary is a straight line that corresponds to the line where the 2D hypothesis function equals 0.5. Although for certain problems straight boundaries are suitable, more complex problems cannot be accurately addressed employing such a simple model. In this case, the system is said to suffer from *biasing*. This issue can be overcome by mapping the original data to a higher dimensional space where classification can be more accurately performed. Among the different approaches to perform feature mapping, polynomial mapping can be applied. For instance, if the input features are given by $\{x_1, x_2\}$, then we can map them to a higher dimensional space according to Table 1.

By performing this mapping, the model considers a higher number of features that can fit much more complex systems. However, increasing the feature space can be detrimental because the model can fit disturbances of the training set. As a result, the ability to predict the class of new incoming symbols may be compromised. This effect, denominated overfitting, can be solved by introducing a penalization term in the cost function throughout a regularization

Table 1 Initial two-dimensional features and created higher order features for different polynomial degrees.

N	Initial features	Higher order features
2	$\{x_1, x_2\}$	$\{x_1, x_2, x_1^2, x_2^2, x_1x_2\}$
3	$\{x_1, x_2\}$	$\{x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1^2x_2, x_1x_2^2\}$

factor λ . The regularized cost function, $J'(\theta)$, then can be written in terms of the unregularized cost function as:

$$J'(\theta) = J(\theta) + \frac{\lambda}{2m} \sum_{l=1}^n \theta_l^2, \quad (4)$$

where θ_l is the l -th component of the θ parameter vector. The updating expressions now acquire a different form depending on the gradient component to be updated. In this way, the 0-th component is updated by

$$\theta_0[k] = \theta_0[k-1] - \frac{\alpha}{m} \sum_{i=1}^m [h(x^{(i)}) - y^{(i)}] x_0^{(i)} \quad (5)$$

whereas the rest of components are iteratively optimized according to

$$\theta_l[k] = \theta_l[k-1] \left(1 - \alpha \frac{\lambda}{m} \right) - \frac{\alpha}{m} \sum_{i=1}^m [h(x^{(i)}) - y^{(i)}] x_l^{(i)}. \quad (6)$$

Figure 1(c) shows the decision boundaries for an example of polynomial feature mapping for three different values of λ corresponding to: an unregularized case, $\lambda = 0$ where overfitting can be easily observed; an intermediate value of $\lambda = 0.02$ that results in an accurate classification; and an excessively high value of $\lambda = 0.04$, leading to biasing issues.

2.3 Multiclass Logistic Regression

LR can also be extended to multiple class problems, which can be addressed in different ways. One of the alternatives is to substitute the sigmoid function by a *softmax* function that considers multiple classes. Alternatively, the one-vs-all approach can be adopted, where for each class a binary LR is performed considering the data of the desired class as one class and all the points belonging to other classes as another class [19]. Once the parameters θ for each class are found, the receiver classifies a new received data by computing the values of the different hypothesis functions and assigning the label corresponding to the class resulting in the highest value of hypothesis function. As shown in Fig. 1(d), when employing feature mapping, this process leads to decision regions with curved boundaries that match nonlinearly distorted constellations better than straight lines.

Fig. 2 Constellations affected by SPM and receiver additive noise (a) in absence of ISI and (b) considering CD-induced ISI.

2.4 LR for classification of QAM constellations

LR can then be employed to classify QAM constellations in single-channel systems where SPM is the dominant nonlinear distortion mechanism. In order to gain some knowledge on the structure of the clusters, in Fig. 2(a) and (b), we show examples of constellations affected by SPM and receiver noise in absence and in presence of CD-induced inter-symbol interference (ISI), respectively. As can be seen, when no ISI is considered, the center of each cluster is rotated according to the amplitude of the symbol but it keeps its symmetrical two-dimensional Gaussian distribution. When ISI is present, on the other hand, in addition to an amplitude dependent rotation, the clusters acquire a more complex distribution. This distribution depends on the dispersion of link and on the relation between dispersion, received power, and receiver noise, making an analytical expression difficult to derive. As in many other cases, machine learning algorithms can assist to solve problems where obtaining an exact solution is not possible or it is too complex. Thus, for classifying the distorted constellations, we will test polynomial feature mapping of different orders alongside with regularization parameters to avoid overfitting.

3 System model

The simulation setup employed to evaluate the proposed classification technique in a system employing an unrepeatd single-channel link and 16-QAM modulation format is shown in Fig. 3. We used a Matlab/VPI Transmission Maker co-simulation where the QAM electrical modulation and demodulation, as well as the required digital signal processing, were carried in Matlab and the conversion to optical domain, fiber transmission, and signal detection were simulated using VPI Transmission Maker. In the transmitter side, a pseudo-random bit sequence (PRBS) was converted from serial to parallel in blocks of 4 bits and mapped into a 16-QAM constellation with Gray code. The resulting QAM symbols were decomposed into their real (in-phase) and imaginary

(quadrature) parts, oversampled, and filtered to emulate digital-to-analog conversion (DAC). These signals drove a dual-parallel Mach-Zehnder modulator (DP-MZM) that modulated the light coming from a continuous wave (CW) laser diode (LD₁) with an operating wavelength of 1550 nm. The modulated signal was amplified and its power was controlled by a variable optical attenuator (VOA) before being launched into an 80-km standard single mode fiber (SSMF). Afterwards, the propagated signal was attenuated by 18 dB, which may represent either a 1:64 splitting ratio or a power penalty due to other impairments. The signal was then transmitted over a second SSMF span of 20 km. At the receiver, a 90° hybrid network was used to combine the incoming signal with a second CW laser (LD₂), also operating at 1550 nm. The outputs of the hybrid network were detected by two pairs of balanced photodetectors (PDs) that generated two differential signals that were digitally processed. The first stage of the DSP was a frequency-domain chromatic dispersion compensation. It was followed by the time synchronization stage and by a downsampling process to get a single sample per symbol. The rotation of the constellation caused by the combined phase noise from LD₁ and LD₂ was compensated using a blind phase search algorithm. In our simulations we considered a bit rate of 56 Gbps and a number of simulated QAM symbols of 81,500. These symbols were split in a training set of up to 25,000 symbols and a fixed test set of 56,500 symbols. Due to the non-Gaussian distribution of the constellation points, the bit error rate (BER) was calculated through error counting and the effective Q factor was obtained from it according to

Fig. 3 Simulated coherent LR-PON system. S/P: serial-to-parallel conversion. DAC: digital to analog converter. LD: laser diode. DP-MZM: dual parallel-Mach-Zenhder modulator. EDFA: erbium-doped fiber amplifier. VOA: variable optical attenuator. SSMF: standard single mode fiber. Att: attenuator. LPF: low-pass filter. ADC: analog-to-digital converter. DSP: digital signal processing.

$Q_{eff} = \sqrt{2} \operatorname{erfc}^{-1}(BER)$ [20]. The most important simulation parameters are summarized in Table 2.

Table 2 Parameters used in the simulation.

System parameters		
Parameter	Value	Unit
LD ₁ linewidth	100	kHz
LD ₁ power	10	mW
LD ₂ linewidth	100	kHz
LD ₂ power	10	mW
MZM insertion loss (IL)	6	dB
SSMF ₁ length (L_2)	80	km
SSMF ₂ length (L_1)	20	km
Fiber attenuation α	0.2	dB/km
Fiber CD parameter (D)	16	ps/nm/km
Fiber PMD coefficient (D_{PMD})	3.16	ps/ $\sqrt{\text{km}}$
Nonlinear coefficient (γ)	1.3	$\text{W}^{-1} \cdot \text{km}^{-1}$
Fiber effective area	80	μm^2
Amplifier gain	20	dB
Amplifier NF	4	dB
Attenuator	18	dB
90°-hybrid insertion loss	10	dB
PD noise dens.	10	pA/Hz
PD responsivity	1	W/A
Electrical filter order	4	–
Electrical filter bandwidth	10.5	GHz
Signal parameters		
Parameter	Value	Unit
Modulation	16-QAM	–
Bit rate	56	Gbps
Simulation parameters		
Parameter	Value	Unit
PRBS length	81,500	–
Sampling rate (dig. dom.)	112	GSa/s
Sampling rate (opt. dom.)	896	GSa/s

4 Results

4.1 Performance analysis

Figure 4(a) shows the BER after detecting the signal using the widely adopted maximum likelihood (ML) method and LR with different configuration parameters. In particular, we considered combination of second and third order

polynomial mapping and regularization factors of 0 (unregularized), 0.02, and 0.04. As can be observed, at low power levels, where the additive noise is dominant, the BER curves corresponding to the LR configurations and maximum likelihood converge. For high launch optical power, on the other hand, LR outperforms ML, showing that indeed logistic regression is compensating the effect of SPM when this is more harmful. Comparing the different considered LR configurations, for both low and high launch optical power levels they present similar performance. Nevertheless, for intermediate values, a second-order polynomial mapping with a regularization factor of $\lambda = 0.02$ results in an optimum BER at a 6 mW of $4.27 \cdot 10^{-4}$, representing a 0.36-dB reduction in the effective Q-factor with respect to the commonly adopted ML approach. In addition, considering a forward error correction (FEC) limit of $2 \cdot 10^{-3}$, the use of LR leads to a power margin enhancement of ~ 2 mW,

Fig. 4 (a) BER in terms of the launch optical power for ML and different LR configurations. (b-d) Received constellations and the obtained decision regions using ML for launch optical powers of 2, 6 and 13 mW, respectively. (e-g) Constellations and decision regions employing second-order LR and $\lambda = 0.02$ for the aforementioned power levels.

Fig. 5 BER for each constellation point for both maximum likelihood and LR classifications: (a,b,c) are for maximum likelihood at launch optical power of 2, 6, and 10 mW, respectively. (d,e,f) are for the same power levels employing LR.

that is ~ 1 dB. In order to understand how LR classification reduces the BER in systems affected by SPM, in Fig. 4(b-c) we show the constellations for low (2 mW), intermediate (6 mW), and high (13 mW) power levels, alongside with the obtained decision regions when using conventional ML detection, whereas in Fig. 4(e-g), the decision regions obtained employing LR are shown. As can be seen, for 2 mW, when employing LR, the decision regions are limited by almost straight boundaries, which are very similar to the boundaries obtained using ML and, therefore, the obtained BER is similar for both approaches. For intermediate and high launch optical power levels, the boundaries found by LR are curved, being better adapted to the distorted constellation than those straight obtained ML.

The previous analysis considered the averaged BER. That is, the BER curves are for the overall classification error considering all the constellation points simultaneously but, looking at the constellations in Fig. 4(b-g) that at moderate and high power levels, points with larger amplitude are more prone to errors than those with lower amplitude. This effect is quantified in Fig. 5, where the BER for each symbol is graphically presented for both maximum likelihood (a-c) and LR (d-f) at three different power levels, i.e. 2, 6, and 10 mW. These values can be interpreted as the complimentary of accuracy matrices, as this metric better exemplifies the effect of nonlinear compensation [17]. Indeed, for the highest power level it can be seen that the great majority of the errors occurs only in four constellation points. This effect is common to maximum likelihood and LR but it can be clearly observed that LR is more efficient in compensating the BER in the most affected constellation points, thus resulting in an improved overall BER.

4.2 Comparison with other supervised classification techniques

In order to show the potential of the proposed LR-based nonlinear classifier, in 6, we show the BER performances of different classification methods. In particular, we compare LR with KNN, SVM artificial neural networks (ANNs), which operate in supervised mode and K-means that is an unsupervised classification method. Each algorithm was configured for optimum performance. Thus, for LR, $N = 2$ and $\lambda = 0.02$ were used. In the case of KNN, we adopted $K = 13$ neighbors, as this was the best configuration according to [21]. For SVM, γ and C were set to 0.1 and 20, respectively. Regarding the ANN, a single hidden layer composed of $s_2 = 16$ neurons. It is also worth noting that for K-means we employed K-means++ as initialization method as this reduces the amount of iterations and reduces the probability for erroneous classification [22]. As can be observed, at low power levels, all the classification methods converge to the BER values of ML but, as the power increases, the different machine learning algorithms outperform ML. Nevertheless, although KNN,

Fig. 6 BER curves for different classification methods: ML, LR with $N = 2$ and $\lambda = 0.02$, KNN with $K = 3$, and SVM with $\gamma = 0.1$ and $C = 20$, ANN with $s_2 = 16$, and K-means.

Table 3 Comparison between the performance of ML, LR, KNN, SVM, ANN, and K-means.

Technique	Parameters	LOP (mW)	BER	Q_{eff} (dB)
ML	-	5	$0.69 \cdot 10^{-4}$	10.10
LR	$N = 2, \lambda = 0.02$	6	$0.43 \cdot 10^{-4}$	10.46
KNN	$K = 13$	6	$0.58 \cdot 10^{-4}$	10.23
SVM	$\gamma = 0.1, C = 20$	6	$0.46 \cdot 10^{-4}$	10.41
ANN	$S_c = 16$	6	$0.47 \cdot 10^{-4}$	10.40
K-Means	-	5	$0.52 \cdot 10^{-4}$	10.32

SVM, ANN, K-means and LR can clearly mitigate NPN effects, the latter shows the best performance, as can be quantitatively observed in Table 3. The closest performance to that of LR is achieved with SVM, however, LR presents a lower computational complexity.

In order to show the potential of the proposed LR-based nonlinear classifier, in Fig. 6, we show the BER performances of different classification methods. In particular, we compare LR with KNN, SVM, artificial neural networks (ANNs), which operate in supervised mode and K-means that is an unsupervised classification method. Each algorithm was configured for optimum performance. Thus, for LR, $N = 2$ and $\lambda = 0.02$ were used. In the case of KNN, we adopted $K = 13$ neighbors, as this was the best configuration according to [21]. For SVM, γ and C were set to 0.1 and 20, respectively. Regarding the ANN, a single hidden layer composed of $s_2 = 16$ neurons. It is also worth noting that for K-means we employed K-means++ as initialization method as this reduces the amount of iterations and reduces the probability for erroneous classification [22]. As can be observed, at low power levels, all the classification methods converge to the BER values of ML but, as the power increases, the different machine learning algorithms outperform ML. KNN, SVM, ANN, K-means and LR are clearly able to mitigate NPN effects. However, LR outperforms the other methods, as can be quantitatively observed in Table 4.1. The closest performance to that of LR is achieved with SVM, however, LR presents a lower computational complexity.

4.3 Training set size and complexity analysis

Albeit the considered optical channel may be quite static and, therefore, training process should be carried out rarely, it is important to analyze the amount of data required to accurately train the LR-based receiver as this measures how prone the system is to eventual re-configurations. In Fig. 7, the BER of the test set in terms of the training set size for a second-order polynomial

Fig. 7 BER of the test set in terms of the training set size. For each set size we repeated the train 100 times.

mapping and a regularization factor of $\lambda = 0.02$. The performance of the LR-based classifier depends on the training set, which comes from a stochastic process. Therefore, for training set sizes ranging from 500 to 25,000, we performed the training 100 times by randomly selecting subsets of the training set while maintaining the test set fixed. In Fig. 7, dots represent the outcome of each training process and solid line corresponds to the ensemble average for each training set size. As expected, for small training sets, not only the BER variance but also its mean are relatively high, resulting in a poor classification performance. As the training set size increases, the mean BER converges to $\sim 5 \cdot 10^{-4}$ and the standard deviation stabilizes at $\sim 7 \cdot 10^{-5}$. From this figure, we can then conclude that we can use a training set of 10,000 symbols with a negligible performance penalty. Furthermore, the size of the training set can even be reduced down to 5,000 with an average increase in the BER of 27.21%.

Another important point to consider is the computational complexity of the algorithm, since an excessively high complexity would prevent its adoption in real-time systems. As mentioned, the system is relatively static and, therefore, we limit our study to the test stage, that is, after training process has been carried out. If the logistic function is implemented using a look-up table, then the only operations that should be performed are those required to calculate the activation parameter $\theta \cdot \mathbf{x}$, which will depend on the degree of the polynomial feature mapping. In addition, it is worth noting that since in-phase and quadrature components are independently considered, all the involved operations are real. We can split the total complexity into three terms: (i) the number of operation required to compute the values of the new features, (ii) the number of operations corresponding to multiplying each element of the feature vector by the corresponding weight, and (iii) the sum of all weighted terms. The number of operation to calculate the values of the feature is given by:

$$C_{(i)} = \frac{1}{3}N^3 + N^2 + \frac{2}{3}N, \quad \forall N \geq 1. \quad (7)$$

Regarding the number of product operation, this corresponds to the number of features, which is given by:

$$C_{(ii)} = \frac{1}{2}N^2 + \frac{3}{2}N + 1, \quad \forall N \geq 1. \quad (8)$$

And, finally, the number of adding operations can be calculated by subtracting one to the number of features:

$$C_{(iii)} = \frac{1}{2}N^2 + \frac{3}{2}N, \quad \forall N \geq 1. \quad (9)$$

Since we adopted a one-vs-all approach, these calculations have to be performed for each of the M classes, so the total number of operations is given by:

$$C_{total} = M \cdot \left[\frac{1}{3}N^3 + 2N^2 + \frac{11}{3}N + 1 \right]. \quad (10)$$

Thus, particularizing expression (10) to our parameters, $N = 2$ and $M = 16$, we obtain that the required number of operations per symbol is 624. It is worth noting that, since for each class the hypothesis function can be independently computed, parallelization is straightforward. Therefore, the latency of classification can be as low as 39 floating-point operations.

Comparing to KNN and SVM applied to the current scenario, the proposed approach presents a significantly lower complexity. In the case of KNN, its complexity depends on the particular implementation because, even if all implementations require the computation of the distances to all the training symbols, different approach to find the K nearest neighbors can be adopted. If, for instance, we consider sweeping K times over the whole distance vector to find the K nearest neighbors, it results in a total complexity of $(3 + K) \cdot N_{tr}$ operations, where N_{tr} is the length of the training set. Since for the system under test the best configuration for KNN is achieved for values of $K = 13$ and a training set length of 1,000, the total number of operation is 16,000. Regarding the SVM complexity, the number of required operations is given by $M \cdot (3 \cdot N_{SV} + N_{SV} - 1)$, where N_{SV} is the number of support vectors that ultimately depends on the distribution of symbols to classify. Therefore, the number of support vectors varies with the launch optical power. In particular, for the optimum power level, i.e. 6 mW, and a training set size of 10,000 symbols, 521 vectors were required, resulting in a total operation count of 33,328. Consequently, for optimum configurations of the different approaches, LR presents a complexity 25 and 53 times lower than those of KNN and SVM, respectively.

5 Conclusions

In this work, we have presented a LR-based classifier to reduce the BER in unrepeatable digital coherent systems employing 16-QAM affected by the combination of nonlinear phase and additive noise. Numerical simulations performed in co-simulated VPI Transmission Maker for Matlab show that the adoption of LR with a second-order polynomial mapping and a 0.02 regularization factor improves the system performance with respect to the traditional ML detection method, reducing the BER from $6.88 \cdot 10^{-4}$ to $4.27 \cdot 10^{-4}$ and offering a power margin of ~ 1 dB. Simulations also reveal that a training set of 10,000 symbols is sufficient to achieve optimum performance. In addition, LR-based classification presents low complexity (16 parallel executions of 39 floating point operations), which makes this approach promising for future implementations.

Funding

The authors thank the National Council for Scientific and Technological Development (CNPq, grant numbers 432303/2018-9 and 311035/2018-3) and the

São Paulo Research Foundation (FAPESP, grant numbers 2015/24517-8 and 2018/25339-4).

References

1. Gordon, J.P., Mollenauer, L.F.: Phase noise in photonic communications systems using linear amplifiers. *Optics Letters* **15**(23), 1351–1353 (1990)
2. Wang, D., Zhang, M., Cai, Z., Cui, Y., Li, Z., Han, H., Fu, M., Luo, B.: Combatting nonlinear phase noise in coherent optical systems with an optimized decision processor based on machine learning. *Optics Communications* **369**, 199–208 (2016)
3. Ho, K.P., Kahn, J.M.: Electronic compensation technique to mitigate nonlinear phase noise. *Journal of Lightwave Technology* **22**(3), 779 (2004)
4. Lavery, D., Maher, R., Millar, D.S., Thomsen, B.C., Bayvel, P., Savory, S.J.: Digital coherent receivers for long-reach optical access networks. *Journal of Lightwave Technology* **31**(4), 609–620 (2012)
5. Lavery, D., Thomsen, B.C., Bayvel, P., Savory, S.J.: Reduced complexity equalization for coherent long-reach passive optical networks. *Journal of Optical Communications and Networking* **7**(1), A16–A27 (2015)
6. Agrawal, G.P.: Nonlinear fiber optics. In: *Nonlinear Science at the Dawn of the 21st Century*, pp. 195–211. Springer (2000)
7. Chughtai, M.N.: Nonlinear phase noise in fiber optical communication. Ph.D. thesis, Chalmers University of Technology (2009)
8. Zibar, D., Winther, O., Franceschi, N., Borkowski, R., Caballero, A., Arlunno, V., Schmidt, M.N., Gonzales, N.G., Mao, B., Ye, Y., et al.: Nonlinear impairment compensation using expectation maximization for dispersion managed and unmanaged pdm 16-qam transmission. *Optics Express* **20**(26), B181–B196 (2012)
9. Aldaya, I., Giacumidis, E., de Oliveira, G., Wei, J., Pita, J., Marconi, J., Fagotto, E., Barry, L., Abbade, M.: Histogram based clustering for nonlinear compensation in long reach coherent passive optical networks. *Applied Sciences* **10**(1), 152 (2020)
10. Wang, D., Zhang, M., Fu, M., Cai, Z., Li, Z., Han, H., Cui, Y., Luo, B.: Nonlinearity mitigation using a machine learning detector based on k -nearest neighbors. *IEEE Photonics Technology Letters* **28**(19), 2102–2105 (2016)
11. Wang, D., Zhang, M., Li, Z., Cui, Y., Liu, J., Yang, Y., Wang, H.: Nonlinear decision boundary created by a machine learning-based classifier to mitigate nonlinear phase noise. In: *European Conference on Optical Communication (ECOC)*, pp. 1–3 (2015)
12. Li, M., Yu, S., Yang, J., Chen, Z., Han, Y., Gu, W.: Nonparameter nonlinear phase noise mitigation by using m-ary support vector machine for coherent optical systems. *IEEE Photonics Journal* **5**(6), 7800312–7800312 (2013)
13. Giacumidis, E., Mhatli, S., Nguyen, T., Le, S.T., Aldaya, I., McCarthy, M.E., Ellis, A.D., Eggleton, B.J.: Comparison of DSP-based nonlinear equalizers for intra-channel nonlinearity compensation in coherent optical OFDM. *Optics Letters* **41**(11), 2509–2512 (2016)
14. Ayalew, L., Yamagishi, H.: The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan. *Geomorphology* **65**(1-2), 15–31 (2005)
15. Laitinen, E.K., Laitinen, T.: Bankruptcy prediction: Application of the Taylor’s expansion in logistic regression. *International review of financial analysis* **9**(4), 327–349 (2000)
16. So, Y.: A tutorial on logistic regression. *SAS White Papers* (1995)
17. Friedman, J., Hastie, T., Tibshirani, R.: *The elements of statistical learning*, vol. 1. Springer series in statistics New York (2001)
18. Buja, A., Stuetzle, W., Shen, Y.: Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, November **3** (2005)
19. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. *Journal of machine learning research* **5**(Jan), 101–141 (2004)

20. Shafik, R.A., Rahman, M.S., Islam, A.R.: On the extended relationships among evm, ber and snr as performance metrics. In: 2006 International Conference on Electrical and Computer Engineering, pp. 408–411. IEEE (2006)
21. Paula, R., Borges, L., Abbade, M., Aldaya, I.: Mitigation of nonlinear phase noise in coherent 16-qam long-reach pons by k-nearest neighbors-based classification (2020). DOI 10.14209/SBRT.2020.1570649603
22. Borges, L., Costa, C., Paula, R., Abbade, M., Aldaya, I.: K-means clustering for mitigation of nonlinear phase noise in digital coherent optical systems using 16-qam modulation format (2020). DOI 10.14209/SBRT.2020.1570646347

Figures

Figure 1

Please see the Manuscript PDF file for the complete figure caption

Figure 2

Please see the Manuscript PDF file for the complete figure caption

Figure 3

Please see the Manuscript PDF file for the complete figure caption

Figure 4

Please see the Manuscript PDF file for the complete figure caption

Figure 5

Please see the Manuscript PDF file for the complete figure caption

Figure 6

Please see the Manuscript PDF file for the complete figure caption

Figure 7

Please see the Manuscript PDF file for the complete figure caption