Learning Vector Quantization-Aided Detection for MIMO Systems

Zheng Wang, *Member, IEEE*, Xiong Xu, Xiaodan Qiang, and Kezhi Li, *Member, IEEE*

***Abstract*— In this letter, the learning vector quantization (LVQ) from machine learning (ML) is adopted into the large-scale multiple-input multiple-output (MIMO) detection to improve the detection performance. Inspired by the decision region from lattice decoding, the random Gaussian noises are applied in the proposed learning vector quantization-aided detection (LVQD) algorithm for data generation. Then, based on the classification, supervised learning is activated to update the targeted prototype vector iteratively, so as to a better detection performance. Meanwhile, the decoding radius in lattices is also used to serve as a preprocessing for LVQD, which leads to an efficient detection without performance loss. Finally, simulation results confirm that considerable performance gain can be achieved by the proposed LVQD algorithm, which suits well for suboptimal detection schemes.**

***Index Terms*— Learning vector quantization, large-scale MIMO detection, lattice decoding, machine learning.**

I. Introduction

NOWADAYS, the large-scale multiple-input multiple-output (MIMO) system has become a promising extension of MIMO in 5G, which boosts the network capacity on a much greater scale without extra bandwidth [1]. However, the dramatically increased system size also places a pressing challenge on signal detection in MIMO communications. In theory, the problem in MIMO detection is known as the closest vector problem (CVP) in *lattice decoding* [2], and the technique of lattice reduction (LR) from number theory was introduced to improve the detection performance. Unfortunately, although the full receive diversity can be achieved by lattice-reduction-aided detection, the performance gap between the suboptimal detection schemes and the optimal ML detection is still substantial especially in high dimensional systems [3].

On the other hand, in pursuit of efficient detection, a number of advanced works have been made. Specifically, to avoid the computations invoked by matrix inversion in linear detection schemes (i.e. MMSE detection), low-complexity techniques like Neumann Series Expansion, Gauss-Seidel, Chebyshev iteration and conjugate gradient are introduced, which reduce the computational complexity of MMSE detection from $O(n^3)$ to $O(n^2)$ [4]. Besides, the low-complexity message passing (MP) detection scheme is also provided in [5], and a genetic-based detection scheme for MIMO systems can be found in [6]. In addition, sampling detection strategy is introduced for MIMO detection in [7], which obtains the near-optimal detection solution by probabilistic sampling.

Different from these detection schemes, we propose a supervised learning-based detection scheme named as learning vector quantization-aided detection (LVQD) algorithm, where the eligible labeled data are generated through the random Gaussian noises. By fully making use of these labeled data, the supervised learning, which is motivated by *learning vector quantization* (LVQ) from machine learning, is carried out to refine the target prototype vector. In order to ensure the reliable and efficient detection, the concepts of *decision region* and *decoding radius* from lattice decoding are introduced. Overall, this could be a good attempt for MIMO detection to exploit the detection potential in the way of supervised learning.

To summarize, this work advances the state of the art of MIMO detection in two fronts. On one hand, based on the learning vector quantization technique from machine learning, the proposed LVQD algorithm is driven by supervised learning, which could be significantly enhanced through deep neural networks [8]. On the other hand, in sharp contrast to the deep learning-based MIMO detection [9], [10], the proposed LVQD algorithm has no stringent request upon the excessive off-line data for training, making it easy to implement. Meanwhile, the mechanism of learning vector quantization also allows parallel implementation, which is beneficial for practical applications.

II. System Model

For notational simplicity, here we consider the detection of an $n \times n$ real-valued MIMO system, where the scenario extension to the $n \times m$, $n > m$ complex-valued one is straightforward [11]. Typically, let $\mathbf{x} \in \mathbb{Z}^n$ denote the transmitted signal. Then the received signal $\mathbf{y} \in \mathbb{R}^n$ is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{n}$ is the noise vector with zero mean and variance σ_n^2, $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \subset \mathbb{R}^{n \times n}$ is an $n \times n$ full column-rank matrix with Gaussian channel coefficients.

Given the MIMO system in (1), the optimal maximum likelihood (ML) detection reads

$$\widehat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{Z}^n} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{2}$$

Manuscript received November 2, 2020; accepted November 17, 2020. Date of publication November 20, 2020; date of current version March 10, 2021. This work is supported in part by the National Natural Science Foundation of China under Grant 61801216, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20180420. The associate editor coordinating the review of this letter and approving it for publication was J. Choi. *(Corresponding author: Zheng Wang.)*

Zheng Wang is with the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, CEMEE, Luoyang 471003, China, and also with the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Nanjing University of Aeronautics and Astronautics, Ministry of Industry and Information Technology, Nanjing 211106, China (e-mail: z.wang@ieee.org).

Xiong Xu is with the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, CEMEE, Luoyang 471003, China.

Xiaodan Qiang is with MCC Huatian Engineering and Technology Corporation, Nanjing 210000, China.

Kezhi Li is with the Institute of Health Informatics, University College London, London WC1E 6BT, U.K.

Digital Object Identifier 10.1109/LCOMM.2020.3039528

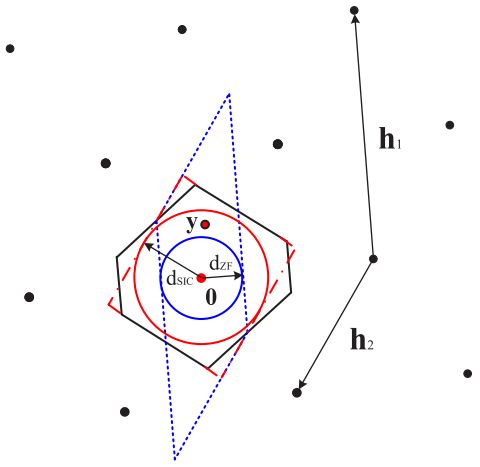

Fig. 1. Decision regions of ZF (blue dotted line), SIC (red dash-dotted line) and ML (black solid line) detections in a 2-D lattice $\mathbf{Hx}$ with $\mathbf{H} = [\mathbf{h}_1\mathbf{h}_2]$ and $\mathbf{x} \in \mathbb{Z}^2$. The blue and red circles denote the sphere region determined by decoding radii d_{ZF} and d_{SIC}.

which corresponds to solving the closest vector problem (CVP) in lattice decoding [2]. Typically, in lattice theory, the n-dimensional lattice generated by $\mathbf{H}$ is defined by

$$\mathcal{L}(\mathbf{H}) = \{\mathbf{Hx} : \mathbf{x} \in \mathbb{Z}^n\}, \tag{3}$$

where $\mathbf{H}$ is referred to as the lattice basis. In other words, the ML detection in large-scale MIMO systems corresponds to finding the closest lattice point $\mathbf{Hx}$ to $\mathbf{y}$ in the space.

In order to perform analytic diagnosis about detection schemes, the concept of *decision region* was proposed in [2]. Without loss of generality, let the transmitted signal $\mathbf{x}$ be $\mathbf{0}$, for example, the decision region of ML detection, i.e., $\mathcal{R}_{\text{ML}}$, can be described as

$$\mathcal{R}_{\text{ML}} = \{\mathbf{y} : \|\mathbf{y} - \mathbf{Hx}\| \geq \|\mathbf{y}\|, \forall \mathbf{x} \in \mathbb{Z}^n\}, \tag{4}$$

where $\mathcal{R}_{\text{ML}}$ is also known as the *voronoi cell* of a lattice. Theoretically, if the given query point $\mathbf{y}$ locates in the decision region of ML (i.e., $\mathbf{y} \in \mathcal{R}_{\text{ML}}$), then it will be correctly detected by ML detection. Besides ML detection, each suboptimal detection also accounts for an own decision region as well while correct detection can be achieved if $\mathbf{y}$ locates in the overlap between decision regions of ML and itself, i.e., $\mathbf{y} \in \mathcal{R}_{\text{detection scheme}} \cap \mathcal{R}_{\text{ML}}$. To make it clear, the decision regions of suboptimal detections like zero forcing (ZF) and successive interference cancelation (SIC) are also depicted in Fig. 1 for a fixed but arbitrary 2-dimensional lattice (i.e., $n = 2$) [2]. Typically, the decision region of ZF detection is the fundamental parallelogram of lattice $\mathcal{L}(\mathbf{H})$ while the decision region of SIC is a rectangle specified by the Gram-Schmidt (GS) vectors as $\{\widehat{\mathbf{H}}\mathbf{a}, \mathbf{a} \in \mathbb{R}^n, |a_i| \leq 1/2\}$. Since the signal space is geometrically uniform in lattices, all these decision regions are symmetric with respect to the origin.

Based on decision regions, the *minimum decoding distance* (also known as *decoding radius*), i.e., denoted by d, was defined to serve as a performance measurement for various detection schemes. In particular, it indicates the Euclidean distance from the original point $\mathbf{0}$ to the closest facet of the decision region $\mathcal{R}$. Specifically, the decoding radii of ZF, SIC and ML are derived as [2]

$$d_{\text{ZF}} = \frac{1}{2} \min_i \|\mathbf{h}_i\| \sin\theta_i, \tag{5}$$

$$d_{\text{SIC}} = \frac{1}{2} \min_i \|\widehat{\mathbf{h}}_i\| = \frac{1}{2} \min_i \|\mathbf{h}_i\| \sin\phi_i, \tag{6}$$

$$d_{\text{ML}} = \frac{1}{2}\lambda(\mathbf{H}). \tag{7}$$

Here, θ_i denotes the angle between $\mathbf{h}_i$ and the linear space spanned by the other $n-1$ basis vectors, $\widehat{\mathbf{h}}_i$'s are the Gram-Schmidt vectors of the matrix $\mathbf{H}$ according to Gram-Schmidt orthogonalization (GSO), ϕ_i is defined as the angle between $\mathbf{h}_i$ and the hyperplane spanned by $\mathbf{h}_1, \ldots, \mathbf{h}_{i-1}$, and $\lambda(\mathbf{H})$ represents the shortest nonzero vector of lattice $\mathcal{L}(\mathbf{H})$. Based on the decoding radius, the related sphere region for each detector can be obtained. Theoretically, once the received query point $\mathbf{y}$ locates within the sphere, it would be correctly detected for sure, which means a larger decoding radius naturally corresponds to a better detection performance.[1]

III. Learning Vector Quantization-Aided Detection

According to the concept of decision region, the locations of the given query point $\mathbf{y}$ can be summarized in the following four cases (take ZF detection as an example), which are illustrated in Fig. 2 in detail:

1) within the sphere built by decoding radius $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\| \leq d_{\text{ZF}}$ (depicted by the blue circle);
2) outside the sphere $\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\| > d_{\text{ZF}}$ but in the decision region $\mathbf{y} \in \mathcal{R}_{\text{ZF}} \cap \mathcal{R}_{\text{ML}}$ (depicted by solid lines);
3) outside the decision region of ZF $\mathbf{y} \notin \mathcal{R}_{\text{ZF}}$ but still in the decision region of ML detector $\mathbf{y} \in \mathcal{R}_{\text{ML}}$ (depicted by dash-dotted lines);
4) outside the decision region of ML detection $\mathbf{y} \notin \mathcal{R}_{\text{ML}}$ (other regions).

Undoubtedly, $\mathbf{y}$ in the first two 1) and 2) cases can be correctly detected due to $\mathbf{y} \in \mathcal{R}_{\text{ZF}} \cap \mathcal{R}_{\text{ML}}$. Here, the correct decision region in 1) can be found out by the derived decoding radius while the correct decision region in 2) is generally hard to know in most cases of interest. Different from cases 1) and 2), in case 4) because $\mathbf{y}$ locates outside the decision region of ML criterion, it would never be correctly detected even with ML detection.

Apart from cases 1), 2) and 4), significant detection potential can be exploited with respect to $\mathbf{y}$ in case 3), where ZF detection yields a wrong result but ML detection issues a correct answer. Therefore, it is possible to optimize the location of $\mathbf{y}$ in case 3) to make it gradually evolve into the case 2) or 1), which leads to a better detection performance. To this end, the learning vector quantization (LVQ) technique from machine learning is introduced to dynamically update $\mathbf{y}$ to a better location.

[1]Since $\widehat{\mathbf{h}}_i$ only needs to be orthogonal to $\mathbf{h}_1, \ldots, \mathbf{h}_{i-1}$, after careful ordering ϕ_i is at least more than θ_i and therefore $d_{\text{ZF}} \leq d_{\text{SIC}}$.

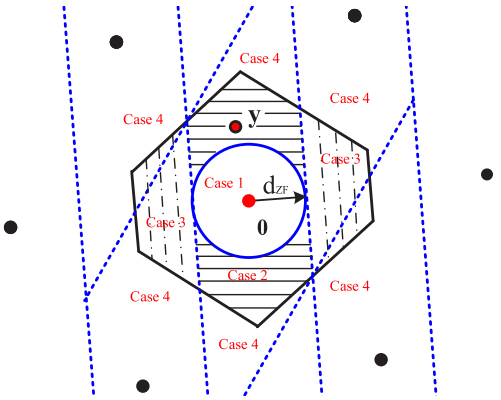

Fig. 2. Illustration of the decision regions of ZF detection with cases 1 (the region within the sphere), 2 (the region with solid lines), 3 (the region with dash-dotted lines) and 4 (the other regions), where $\mathbf{y}$ locates in the region of case 2.

As a special case of artificial neural networks, learning vector quantization is a prototype-based supervised classification algorithm [12], which applies a winner-take-all Hebbian learning-based approach. Since LVQ was introduced by Kohonen, it has been used in a variety of academic and commercial applications, which is rather appealing for easy implementation and controllable complexity. Most importantly, different from other classification schemes like feedforward networks or support vector machine (SVM), LVQ is straightforward to interpret due to the explicit class assignment via the closest prototype, making it attractive in various research fields.

According to the refined prototypes, LVQ performs the classification over the labeled data, where each prototype accounts for a class region. Since MIMO detection only pays attention on the correct detection results, the proposed LVQD algorithm only concerns the prototype indicating the classification with the best detection results while other prototypes are ignored during the detection process. Along with the iterations of the target prototype vector $\mathbf{y}$, the final detection results $\widehat{\mathbf{x}}$ is outputted from it. Specifically, the proposed LVQD algorithm can be described by the following 5 steps.

- **Preprocessing**: The proposed LVQD algorithm aims to improve the performance of suboptimal detection schemes through supervised learning. However, the optimal detection results also can be obtained by suboptimal detection schemes if the effect of noises is mild (i.e., the cases 1) and 2) about the location of $\mathbf{y}$). In this condition, there is no need to recall the proposed LVQD. Motivated by this, here we propose to set a threshold for the activation judgement, which is based on the decoding radius as

$$\|\mathbf{y}-\mathbf{H}\widehat{\mathbf{x}}\| \leq \alpha \cdot d_{\text{detection scheme}}. \tag{8}$$

  Here, $\widehat{\mathbf{x}}$ and $d_{\text{detection scheme}}$ are the detection result and the decoding radius of the applied suboptimal detection scheme respectively, and $\alpha \geq 1$ serves as a coefficient to adjust the threshold. Intuitively, with $\alpha = 1$, if (8) is satisfied by $\widehat{\mathbf{x}}$ from the suboptimal detection scheme, then $\widehat{\mathbf{x}}$ could be directly output to save computational complexity without any performance loss. Otherwise, the following procedures of LVQD will be invoked to improve the detection performance. Note that such a preprocessing can be omitted if the decoding radius is unknown, and the proposed LVQD algorithm shown below still works.
- **Initialization**: The data for supervised learning are generated through the random Gaussian noises as follows

$$\mathbf{y}_j = \mathbf{y} + \mathbf{n}_j \in \mathbb{R}^n, \tag{9}$$

  where $1 \leq j \leq K$ indicates the index number, K is data size and $\mathbf{n}_j \in \mathbb{R}^n$ follows zero mean with variance $\beta \cdot \sigma_n^2$. Here, coefficient $\beta > 0$ is used to adjust the impact of the added noises. It is clear to see with $\beta = 1$ the noise effect upon the given query point $\mathbf{y}$ can be offset in the ideal case (i.e., $\mathbf{n}_j = -\mathbf{n}$). In general, the generated data $\mathbf{y}_j$'s are randomly distributed with center locating at $\mathbf{y}$.
- **Classification**: Given the generated data $\mathbf{y}_j$'s, suboptimal detection is applied to recover $\mathbf{x} \in \mathbb{Z}^n$, where the number of classification, i.e., T, is determined by the number of different recovered vectors $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_T$. Note that the suboptimal detections over $\mathbf{y}_j$'s can be implemented in parallel, making it more efficient After this, the labeled data $\mathcal{D}$ can be obtained as $\mathcal{D} = \{(\mathbf{y}_1, \widehat{\mathbf{x}}_1), (\mathbf{y}_2, \widehat{\mathbf{x}}_2), \ldots, (\mathbf{y}_K, \widehat{\mathbf{x}}_T)\}$ and could be further expressed by subsets as

$$\mathcal{D} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_T \tag{10}$$

  with $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, 1 \leq i \neq j \leq T$, where the subset $\mathcal{S}_t = \{\mathbf{y}_j's \in \mathbb{R}^n | \widehat{\mathbf{x}}_t \in \mathbb{Z}^n\}, 1 \leq t \leq T$ only contains the data with the same classification. Given the labels $\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_T$, the optimal classification can be found according to the fitness function

$$\widehat{\mathbf{x}} = \underset{\mathbf{x} \in \{\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_T\}}{\arg\min} \|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2 \tag{11}$$

  so as to determine the positive subset $\mathcal{S}_{\text{positive}}$. Accordingly, the subsets marked by other labels are referred to as negative subsets. Note that according to (11) there could be numerous negative subsets but only one positive subset.
- **Update**: Based on the results of classification, the target prototype vector $\mathbf{y}$ is updated by

$$\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} + \eta_p \cdot (\overline{\mathbf{y}}_{\text{positive}} - |\mathcal{S}_{\text{positive}}| \cdot \mathbf{y}^{(i-1)}) \tag{12}$$

  and

$$\mathbf{y}^{(i)} = \mathbf{y}^{(i-1)} - \eta_n \cdot (\overline{\mathbf{y}}_{\text{negative}} - |\mathcal{S}_{\text{negative}}| \cdot \mathbf{y}^{(i-1)}) \tag{13}$$

  respectively with

$$\overline{\mathbf{y}}_{\text{positive}} = \sum_{i \in \mathcal{S}_{\text{positive}}} \mathbf{y}_i \text{ and } \overline{\mathbf{y}}_{\text{negative}} = \sum_{j \in \mathcal{S}_{\text{negative}}} \mathbf{y}_j, \tag{14}$$

  where the superscript i is the iteration index of the supervised learning with $\mathbf{y}^{(0)} = \mathbf{y}$, $\eta_p, \eta_n \in (0, 1)$ are the learning rates, $|\mathcal{S}|$ denotes the set size of $\mathcal{S}$. Subsequently, with respect to the updated $\mathbf{y}^{(i)}$, the supervised learning steps into the next iteration including initialization, classification and update for a better target prototype vector.

- **Stopping Criteria**: The target prototype vector $\mathbf{y}^{(i)}$ is updated along the iterations shown above, and the number of iterations, i.e., N, can be used as the stopping trigger. The more number of iterations, the better detection performance. Then, the target prototype vector $\mathbf{y}^{(i)}$ is detected to output the final solution. Note that at the end of each iteration the judgement in (8) can be recalled, and the detection may terminate if the target prototype vector has been well developed.

According to the supervised learning, it is clear to see that the target prototype vector $\mathbf{y}$ is gradually approaching to the direction composed by $\overline{\mathbf{y}}_{\text{positive}}$ while is getting far away from the direction made up by $\overline{\mathbf{y}}_{\text{negative}}$'s, where the convergence speed is controlled by the learning rates η_p and η_n. For the consideration of detection efficiency, it is preferred to choose a negligible learning rate η_n for the update from negative subsets while maintains a regular learning rate η_p for the update from the positive subset. By doing this, the update of the target prototype vector $\mathbf{y}$ is mainly based on the supervised learning from the positive subset, and the updates driven by negative subsets are ignored so that the related calculations about those negative subsets can be avoided, which leads to a better detection efficiency.

In particular, the cost function of the proposed LVQD algorithm can be described as

$$\widehat{\mathbf{y}}^{(i)} = \underset{\mathbf{y}^{(i)}=\mu(\mathbf{y},\mathcal{D})}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathcal{Q}(\mathbf{y}^{(i)})\|^2, \tag{15}$$

where $\mathcal{Q}(\cdot)$ denotes the detection operations of the applied suboptimal detector and μ indicates the nonlinear operations to update $\mathbf{y}$ at each iteration given the labeled data $\mathcal{D}$. This is different from the original LVQ method as it does not have an associated cost function to ensure the convergence. For this reason, the original LVQ method in machine learning is deemed as heuristic, and such a convergence problem is addressed in the generalized LVQ (GLVQ) through the setup of cost function so that the learning rule is derived via the steepest descent [13]. Most importantly, the movement of the target prototype $\mathbf{y}$ is actually determined by the fitness function in (11), which is measured by the Euclidean distance $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|$.

For a better understanding, the illustration of the proposed LVQD algorithm in a 2-D lattice is presented in Fig. 3. Clearly, the target prototype vector $\mathbf{y}$ goes through 2 iterations and reaches into the sphere built by decoding radius, which results in a correct detection thereafter. For a clear presentation, the generated data at the second iteration are not shown in Fig. 3.

Given the decision region of ML detection $\mathcal{R}_{\text{ML}}$, an effective way to strengthen the proposed LVQD algorithm is to optimize the decision region and improve the decoding radius. Therefore, powerful suboptimal detection schemes are encouraged. Besides, Lenstra-Lenstra-Lovász (LLL) reduction is also recommended as it not only refines the decision region but also enlarges the decoding radius by introducing a more orthogonal lattice basis [2]. In a word, our work about LVQ-based neural networks is just a beginning and there are several open questions worthy being further considered for a better detection trade-off, e.g., the learning rates need to be set properly to guarantee the performance improvement. It is also possible to further simplify the supervised learning for a more efficient detection. To summarize, the operations of the proposed LVQD algorithm is outlined in Algorithm 1. Another point should be emphasized is that although decision regions are used here to serve for the algorithm description, there is no need to find them out in the proposed LVQD algorithm because the detection results about $\mathbf{x}$ can be obtained directly without recalling them.

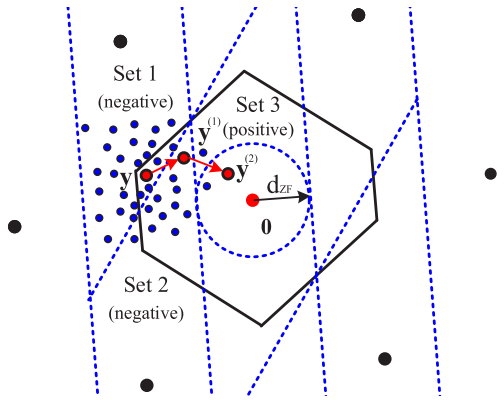

Fig. 3. Illustration of the proposed LVQD algorithm with ZF detection in a 2-D lattice. The blue points are the generated data centered at the query point $\mathbf{y}$ while the red arrows denote the update by supervised learning.

Algorithm 1 Learning Vector Quantization-based Detection

Input: $\mathbf{H}, \mathbf{y}, K, N, \alpha, \beta, \eta_p, \eta_n$
Output: $\widehat{\mathbf{x}}$

perform sub-optimal detection to obtain $\widehat{\mathbf{x}}$
if $\|\mathbf{y} - \mathbf{H}\widehat{\mathbf{x}}\| \leq \alpha \cdot d_{\text{detection scheme}}$ **then**
  output $\widehat{\mathbf{x}}$ as the detection result directly
else
  for $i = 1, \ldots, N$ **do**
    given $\mathbf{y}^{(i-1)}$ generate K data $\mathbf{y}_j$'s by (9)
    get the labeled data $\mathcal{D} = \{(\mathbf{y}_1, \widehat{\mathbf{x}}_1), \ldots, (\mathbf{y}_K, \widehat{\mathbf{x}}_T)\}$
    perform the classification $\mathcal{D} = \mathcal{S}_{\text{positive}} \cup \mathcal{S}'_{\text{negative}}s$
    update the target prototype $\mathbf{y}^{(i)}$ by (12) and (13)
    given updated $\mathbf{y}^{(i)}$ output the detection result $\widehat{\mathbf{x}}$
    if $\|\mathbf{y} - \mathbf{H}\widehat{\mathbf{x}}\| \leq \alpha \cdot d_{\text{detection scheme}}$ **then**
      break output $\widehat{\mathbf{x}}$ as the detection result
    end if
  end for
  given the updated $\mathbf{y}^{(N)}$ output the detection result $\widehat{\mathbf{x}}$
end if

IV. Simulation Results

In this section, the performance of the proposed LVQD algorithm is evaluated through MIMO detection. Note that the proposed LVQD algorithm is designed to enhance the suboptimal detection schemes, and MMSE detection is applied here as an example. Typically, the default configurations of the LVQD algorithm are as follows: $\alpha = 1$, $\beta = 1$, $\eta_p = 0.6$ and $\eta_n = 0.001$ which could be further exploited for a better detection performance.

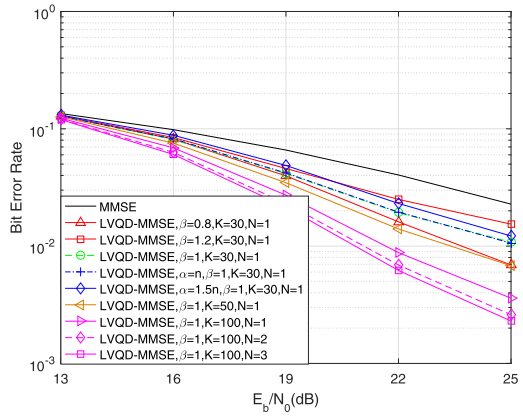

Fig. 4. Bit error rate versus E_b/N_0 for the uncoded 10×10 MIMO system using 64-QAM.

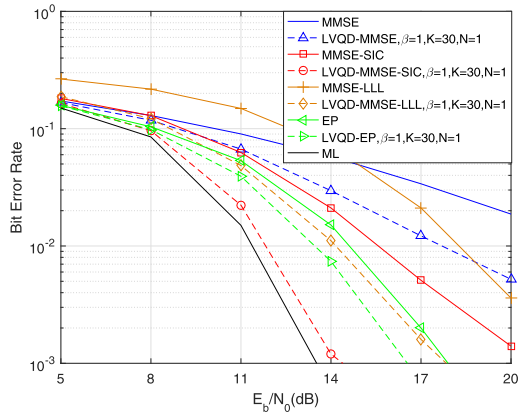

Fig. 5. Bit error rate versus E_b/N_0 for the uncoded 8×8 MIMO system using 16-QAM.

As can be seen from Fig. 4, the proposed learning vector quantization-aided MMSE detection achieves a better detection performance than the standard MMSE. In particular, it is clear to see that with the increment of data size $K = 30, 50, 100$, the detection performance increases gradually. Meanwhile, given the fixed data size $K = 100$, the detection performance also can be further improved along the number of iterations $N = 1, 2, 3$. This is accordance with the update of the target prototype vector $\mathbf{y}^{(i)}$ as it is optimized via the supervised learning. On the other hand, as for the coefficients setup, two detection results with $\beta = 0.8$ and $\beta = 1.2$ are shown as fair comparisons. It is interesting to observe that the case with $\beta = 0.8$ is even better than the default case with $\beta = 1$. This is a little counterintuitive and is worthy being further investigated. Moreover, with $\alpha = n$, the detection suffers negligible performance loss but the a certain of detection complexity can be saved through the decoding radius judgement. By letting $\alpha = 1.5n$, the judgement becomes more loose so that more detection through LVQD are avoided but the corresponding performance loss is also introduced. Moreover, in Fig. 5, the bit error rates (BERs) comparison of the proposed LVQD algorithms over various suboptimal detectors (i.e., MMSE-SIC detection, MMSE-LLL detection and EP detection in [14]) are given in a 8×8 uncoded MIMO system with 16-QAM, where the performance gain induced by LVQD can be verified clearly.

TABLE I

PERCENTAGE OF DIRECT DETECTION AT THE PREPROCESSING STAGE

	$E_b/N_0 = 16$	$E_b/N_0 = 19$	$E_b/N_0 = 22$	$E_b/N_0 = 25$
$\alpha = n$	0.1197	0.1297	0.1387	0.1451
$\alpha = 1.5n$	0.4010	0.3977	0.3914	0.3885

As a complement to illustrate the computational reduction, the percentages of direct detection by MMSE are also given in Table I. In particular, the designed coefficient $\alpha \geq 1$ controls the threshold of the preprocessing stage. The larger α corresponds to the less recalls of LVQD, and direct detection by suboptimal detector itself will be applied more frequently. As can be seen from Table I and Fig. 4, for $\alpha = n$ with $E_b/N_0 = 25$, about 14.5% recalls of LVQD can be saved by direct MMSE detection but with negligible performance loss, which leads to a better detection trade-off. With the increment of $\alpha = 1.5n$, more recalls of LVQD would be avoided but the performance loss also increases accordingly. Therefore, a well chosen α is highly desired.

REFERENCES

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[2] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2795–2808, Jun. 2011.

[3] Z. Wang, Y. Huang, and S. Lyu, "Lattice-reduction-aided Gibbs algorithm for lattice Gaussian sampling: Convergence enhancement and decoding optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4342–4356, Aug. 2019.

[4] G. Peng, L. Liu, P. Zhang, S. Yin, and S. Wei, "Low-computing-load, high-parallelism detection method based on Chebyshev iteration for massive MIMO systems with VLSI architecture," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3775–3788, Jul. 2017.

[5] J. Zeng, J. Lin, and Z. Wang, "Low complexity message passing detection algorithm for large-scale MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 708–711, Oct. 2018.

[6] Y. Wang, Z. Wang, F. Shen, and Q. Shi, "Efficient genetic-based detection algorithm for large-scale MIMO systems," in *Proc. IEEE 2nd Int. Conf. Electron. Inf. Commun. Technol. (ICEICT)*, Jan. 2019, pp. 79–83.

[7] Z. Wang and C. Ling, "Lattice Gaussian sampling by Markov chain Monte Carlo: Bounded distance decoding and trapdoor sampling," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3630–3645, Jun. 2019.

[8] H. de Vries, R. Memisevic, and A. Courville, "Deep learning vector quantization," in *Proc. Eur. Symp. Artif. Neural Netw.*, Apr. 2016, pp. 503–508.

[9] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 1805–1809.

[10] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[11] S. Liu, C. Ling, and D. Stehle, "Decoding by sampling: A randomized lattice algorithm for bounded distance decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5933–5945, Sep. 2011.

[12] T. Kohonen, *Self-Organizing Maps*. New York, NY, USA: Springer-Verlag, 1997.

[13] D. Nova and P. A. Estévez, "A review of learning vector quantization classifiers," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 511–524, Sep. 2014.

[14] J. Cespedes, P. M. Olmos, M. Sanchez-Fernandez, and F. Perez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2840–2849, Aug. 2014.