Pilot-Less One-Shot Sparse Coding for Short Packet-Based Machine-Type Communications

Jiao Wu, Wonjun Kim, and Byonghyo Shim

Abstract—This paper presents a novel transmission scheme to support massive machine-type communications (MTC) devices sending very short packets for Internet-of-Things (IoT) applications. The proposed scheme, termed as pilot-less one-shot (PLOS) transmission, does not require the pilot signaling. The key idea behind PLOS is to encode information into the inter-block nonzero positions and intra-block nonzero positions of a sparse vector. In the receiver, we propose a deep neural network-based scheme, referred to as deep learning-based PLOS (DL-PLOS) to recover the nonzero positions of the sparse vector. From the simulations results, we demonstrate that PLOS is effective in the short packet transmission and DL-PLOS outperforms the conventional greedy algorithms.

Index Terms—Machine-type communications (MTC), short packet transmission, deep neural network (DNN)-based decoding.

I. INTRODUCTION

In the 5th generation (5G) networks and beyond, new types of communication technologies are needed to support reliable massive access for massive machines and devices [1]–[3]. Machine-type communications (MTC) are emerging to support a large variety of services and applications, such as vehicle-to-vehicle (V2V), vehicle-to-everything (V2X), Industry 4.0 and many more [4]. Distinctive feature of MTC over the long-standing human-type-communications (HTC) is that the machine-type devices, such as drones, robots, and sensors, sporadically transmit small-sized packets to deliver the control/command information (e.g., start/stop, move, shift, rotate) or sensing data (e.g., power consumption, temperature, gas density) [5]. One well-known issue in the short packet transmission is that the pilot occupies a large portion in the packet. This issue has not been a serious concern in the past since the amount of payload (transmit information) is much larger than the amount of pilot signal so that the pilot overhead can be easily ignored in the conventional systems. However, when the packet size is tiny, which is true for most massive MTC (mMTC) scenarios, pilot overhead is no longer negligible so that it is important to come up with a machine-type transmission scheme with negligible pilot overhead.

Over the years, various studies have been made to reduce the pilot overhead in the short packet transmission [6]–[10]. In [6], reliable symbols are exploited as virtual pilots to perform the channel re-estimation. In [7], an algorithm to minimize the pilot overhead under the block length and block error probability constraints has been proposed. In [8], an upper bound on the packet error probability obtainable at a given block length and the code rate has been obtained. In [9], a machine learning framework called label-assisted transmission was proposed to reduce the overhead caused by pilot-assisted transmission. In [10], an approach to send the short packet in a form of the sparse vector has been proposed. The main idea of this scheme, called sparse vector coding (SVC), is to map the information into nonzero positions of a sparse vector and then transmit after the random spreading.

In this paper, we propose a novel transmission scheme for MTC scenarios to encode control and data information into one packet and then transmit without pilot signaling. The key idea of the proposed scheme, referred to as Pilot-Less One-Shot (PLOS) transmission, is to map the control information into the nonzero blocks of a sparse vector and then map the sensing data into the nonzero positions of the chosen blocks (see Fig. 1). The pseudo-random spreading is then performed after the sparse mapping. Since the packet size is very small, it is safe to assume that undergoing channel remains unchanged for the packet duration. This means that the composite of such a fading channel and the input sparse vector (channel-scaled input vector) shares the same nonzero positions with the sparse input vector. Under this formulation, the pseudo-random spreading matrix and the channel-scaled input vector serve as the system matrix and the input vector, respectively, and thus the PLOS decoding problem can be cast into the sparse signal recovery (more accurately support identification) problem in the compressed sensing (CS) [12]. Since what we want is the identification of nonzero positions, not the actual nonzero value, the PLOS decoding can be done without the channel information, meaning that the pilot signals to acquire channel estimation is unnecessary.

In solving the support identification problem, the greedy algorithms have been widely used [13]. In this work, we put forth an entirely different approach based on a deep neural network (DNN) which has been successful to solve complicated nonlinear problems [14]. In the proposed deep learning scheme, called deep learning-based PLOS (DL-PLOS), support identification problem can be modeled as a multi-label classification problem. In the DL-PLOS decoding, the network learns the complicated and nonlinear mapping between the received vector $y$ and the support of the sparse vector via the training process. We demonstrate from the simulations that the DL-PLOS decoding outperforms the conventional CS-based techniques, achieving about 4 dB and more than 10 dB gain in the inter-block and intra-block decoding performances, respectively.

The rest of this paper is organized as follows. In Section II, we introduce the SVC scheme and present the PLOS transmission. In

---

Manuscript received December 26, 2019; revised April 7, 2020; accepted May 11, 2020. Date of publication May 19, 2020; date of current version August 13, 2020. This work was supported in part by the National Research Foundation of Korea (NRF), in part by the Korea government (MSIP) under Grant 2014R1A5A1014178, in part by Samsung Research Funding and Incubation Center for Future Technology of Samsung Electronics under project SRFC-TF1901-17. The review of this article was coordinated by Prof. G. Gui.

(Corresponding author: Byonghyo Shim.)

The authors are with the Institute of New Media and Communications and the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: jiaowu@islab.snu.ac.kr; wjkim@islab.snu.ac.kr; bshim@snu.ac.kr).

Digital Object Identifier 10.1109/TVT.2020.2995840

---

Fig. 1. Example of PLOS encoding.
Section III, we perform the PLOS decoding via the conventional greedy algorithms, and propose a novel decoding scheme based on the deep learning method. In Section IV, we present the simulation results and conclude the paper in Section V.

II. PILOT-LESS ONE-SHOT TRANSMISSION

A. SVC Processing

We consider the short packet transmission in the uplink MTC scenarios where a user uses $N_s$ physical resources to transmit $b$-bit of information. The SVC transmit vector is generated by two steps: sparse mapping and pseudo-random spreading. First, an information vector $w$ is mapped to the positions of a sparse vector $x$. When we try to put $k$ nonzero elements into the $N_s$-dimensional sparse vector, we have $\binom{N_s}{k}$ choices in total and thus $b = \lfloor \log_2 \binom{N_s}{k} \rfloor$ bits of information can be encoded by the sparse mapping.

After the sparse mapping, we spread the sparse vector $x$ into $N_B$ resources using the pseudo-random codebook $C = [c_1, c_2 \cdots c_{N_B}]$. For example, if $x_2$ and $x_3$ are nonzero positions of $x$, then the spread vector $s$ can be expressed as

$$s = Cx = c_2x_2 + c_3x_3,$$

where $c_i = [c_{i1}, c_{i2} \cdots c_{iN_s}]^T$ ($i = 1, 2, \ldots, N_s$) is the $i$-th codeword (spreading sequence). The resulting underdetermined sparse system is given by

$$y = HCx + v = HS + v,$$

where $H = \text{diag}(h_1, \ldots, h_{N_B})$ is the channel matrix, $h_i$ is the channel corresponding to the $i$-th subcarrier, $s = Cx$, and $v \sim CN(0, \sigma_v^2 I)$ is the additive white Gaussian noise (AWGN) vector.

Since the SVC decoding is done by the support identification of $x$, not the recovery of the nonzero values, any sparse recovery algorithm, such as the orthogonal matching pursuit (OMP) [13], can be employed.

B. Pilot-Less One-Shot Transmission

Fig. 2 describes the block diagram of the proposed PLOS scheme. Encoding process of PLOS transmission is divided into two steps: inter-block encoding and intra-block encoding. First, in the inter-block encoding, $x$ is divided into $N_B$ blocks from which $K_B$ blocks are chosen. Since in this case we have $\binom{N_B}{K_B}$ choices in total, $b_c = \lfloor \log_2 \binom{N_B}{K_B} \rfloor$ bits of (control) information can be mapped to the inter-block support.\(^2\)

Second, in the intra-block encoding, the data information (e.g., sensing data or control/command) is mapped to the intra-block support\(^3\) (i.e., nonzero positions of the chosen blocks). As shown in Fig. 1, when $N_s = 16$, $N_B = 4$, and $K_B = 2$, $b_c = 2$ bits of control information can be delivered. Also, since $K_B_1 = K_B_2 = 2$, $b_c = 4$ bits of data can be transmitted via the intra-block support. For simplicity, we assume that $K_B_1 (i = 1, 2, \ldots, K_B)$ is a constant in this work.

After the transform of a transmit information into the sparse vector, we perform the pseudo-random spreading using the block-diagonal codebook matrix $C \in \mathbb{C}^{N_s \times N_s}$. The relationship between the received vector $y$ and the input sparse vector $x$ is expressed as

$$y = HCx + v.$$

Since the packet length is generally much smaller than the channel coherence time,\(^4\) the channel response is assumed to be a constant during the packet transmission period, i.e., $H = \text{diag}(h, \ldots, h)$. Thus, the channel-scaled input vector $Hx$ shares the same nonzero positions with $x$. Recalling that the goal of the PLOS packet decoding is to recover the nonzero positions of $x$, not the nonzero value, the decoding process can be done by the support identification of $x$. In view of this, the received vector $y$ of PLOS transmission can be rewritten as

$$y = HCx + v = CHx + v = C\tilde{x} + v,$$

where $\tilde{x} = Hx$. Thus,

$$y = C_1v_1 + C_2v_2 + \cdots + C_{N_B}v_{N_B} = \begin{bmatrix} C_1 & C_2 & \cdots & C_{N_B} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_B} \end{bmatrix} = (C\tilde{x})_B.$$ 

Since our goal is to find out the inter-block and intra-block supports of $\tilde{x}$, the channel estimation is unnecessary, which implies that we can save the resources for the pilot signals and pilot transmission power, resulting in a good relief to power hungry battery-powered devices (e.g., energy harvesting IoT devices).

III. DECODING OF THE PLOS PACKET

A. Decoding Using Conventional Algorithms

Since the control information and data information are separately encoded into inter-block and intra-block supports, the decoding process is divided into two parts. First, referring to the system model (4), the inter-block decoding problem can be formulated as

$$\Omega_B = \arg \min_{|\Omega_B| = K_B} \frac{1}{2} ||y - C_{\Omega_B} x^{\Omega_B}||^2_2.$$ 

Well-known approach to solve this problem is the block orthogonal matching pursuit (BOMP) algorithm [15]. Second, in recovering the intra-block support of $x$, one can use a greedy algorithm such as the OMP algorithm.

\(^2\)Inter-block support is the index set of nonzero blocks of a sparse vector. In Fig. 1, $x_1 = [1 \ 0 \ 0 \ 1], x_2 = [0 \ 0 \ 0 \ 0], x_3 = [0 \ 0 \ 0 \ 0], x_4 = [0 \ 1 \ 0 \ 1]$, so that the inter-block support of $x$ is $\Omega_B = \{1, 4\}$.

\(^3\)If $x = [x_1 \ x_2 \ x_3 \ x_4] = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]$, then the intra-block support is $\Omega_C = \{1, 4, 14, 16\}$.

\(^4\)When the carrier frequency $f_c = 3.5$ GHz and the mobile speed is $v = 15$ km/h, the channel coherence time $T_c = \frac{1500}{f_c} = 1.52$ ms is larger than the slot length $T_s = 0.5$ ms. We use LTE symbol length, i.e., 14 symbols in 1 subframe (1 ms).
We note that the greedy algorithm is not so effective in the practical mMTC scenarios since the performance of BOMP/OMP depends heavily on the column correlation of the codebook \( \mathbf{C} \). In many studies, a level of column correlation in \( \mathbf{C} \) is expressed as a mutual coherence \( \mu \) (i.e., \( \mu = \max_{i \neq j} \langle \mathbf{c}_i, \mathbf{c}_j \rangle \)). When we try to support the massive connectivity with relatively small amount of resources (i.e., \( N_x \gg N_z \)), the underdetermined ratio \( N_x/N_z \) would be large so that \( \mu \) will also be large. Thus, conventional sparse recovery algorithms cannot properly handle the column (codeword) correlation of multiple users.

B. Deep Learning Based PLOS

As mentioned, main goal of PLOS decoding is to find out the nonzero positions of \( \mathbf{x} \), not the nonzero values. To handle this task, we use the deep learning technique that learns the nonlinear mapping between the input (i.e., received signal \( y \)) and the inter-block and intra-block supports of \( \mathbf{x} \). The inter-block support identification problem can be expressed as

\[
\hat{\Omega}_B = g_1(y, \Theta),
\]

where \( g_1 \) is the mapping between the input vector \( y \) and the inter-block support of \( \mathbf{x} \) and \( \Theta \) is the set of weights and biases of DL-PLOS network [16]. Similarly, the intra-block support identification problem can be expressed as

\[
\hat{\Omega}_C = g_2(y, \Theta),
\]

where \( g_2 \) represents the mapping between \( y \) and the intra-block support of \( \mathbf{x} \). Instead of solving two problems separately, DL-PLOS jointly solves two problems by learning the nonlinear mappings \( g_1 \) and \( g_2 \) simultaneously.

Fig. 3 depicts the structure of the proposed DL-PLOS technique. The DL-PLOS consists of multiple building blocks including the convolutional layers, rectified linear unit (ReLU) layers, flatten layer, fully-connected (FC) layer, and softmax layer with the batch normalization. In the training process, we use \( P \) training data \( y^{(1)}, \ldots, y^{(P)} \) in each training iteration. The output vector \( \mathbf{z}_0^{(p)} \) of the convolution layer can be expressed as

\[
\mathbf{z}_0^{(p)} = \mathbf{W}^{(0)}y^{(p)} + \mathbf{b}^{(0)}, \quad p = 1, \ldots, P
\]

where \( \mathbf{W}^{(0)} \) and \( \mathbf{b}^{(0)} \) are the initial weight and bias, respectively. After the convolution layer, \( P \) output vectors are stacked to the mini-batch \( \mathbf{B} = [\mathbf{z}_0^{(1)}, \ldots, \mathbf{z}_0^{(P)}]^T \) and then normalized to enforce zero mean and unit variance (this process is referred to as the batch normalization [17]).

After the batch normalization, the output vector passes through the residual blocks (RBs). Each RB consists of the batch normalization layers, ReLU layers, convolution layers with a residual connection. The output of the \( l \)-th RB is given by

\[
s_l^{(p)} = \mathbf{z}_l^{(p)} + \mathcal{A}(\mathbf{z}_l^{(p)}, \Delta), \quad l = 1, \ldots, L
\]

where \( \mathbf{z}_l^{(p)} \) is the input of the \( l \)-th RB, and \( \mathcal{A}(\mathbf{z}_l^{(p)}, \Delta) \) is the residual function affected by \( \mathbf{z}_l^{(p)} \) and parameters \( \Delta \) of the RB module. As shown in Fig. 3, RB includes the residual connection that puts the identity (shortcut) connection between the stacked hidden layers. Since the signal can be directly propagated from one to another through the residual connection, one can accelerate the training speed and also handle the vanishing gradient problem [18]. After that, a nonlinear activation function is applied to \( s_l^{(p)} \) to determine whether the information generated by the hidden unit is activated (delivered to next layer) [19]. The output \( \mathbf{x}_{l+1}^{(p)} \) of the activation function is \( \mathbf{x}_{l+1}^{(p)} = f_{\text{ReLU}}(s_l^{(p)}), f_{\text{ReLU}}(x) = \max(0, x) \).

After passing through the \( L \) RBs, the FC layer produces \( N_x \) output values whose dimension is matched with the size of the sparse vector \( \mathbf{x} \). In the softmax layer, output values are mapped to the probabilities \( \hat{p}_i, \ldots, \hat{p}_{N_x} \) which represent the likelihood of being the support element. The \( q \)-th probability \( \hat{p}_q \) is given by

\[
\hat{p}_q = \frac{e^{s_{\text{softmax}q}}}{\sum_{j=1}^{N_x} e^{s_{\text{softmax}j}}}, \quad q = 1, \ldots, N_x.
\]

Among the output values of the softmax layer, the largest value for the \( i \)-th block \( \hat{p}_{B_i} \) are selected to obtain the vector \( \mathbf{p}_B = [\hat{p}_{B_1}, \ldots, \hat{p}_{B_{N_B}}] \), where \( \hat{p}_{B_i} = \max_{j \in I_{B_i}} \hat{p}_j \), and \( I_{B_i} \) is the index set of the \( i \)-th block. An estimate of the inter-block support is then obtained by taking indices of the \( K_B \) largest values in \( \mathbf{p}_B \):

\[
\hat{\Omega}_B = \arg\max_{|\Omega_B|=K_B} \sum_{B_i \in \Omega_B} \hat{p}_{B_i}.
\]

Finally, an estimate of the intra-block support is obtained by taking indices of the \( K_B K_p \) largest output values of the softmax layer:

\[
\hat{\Omega}_C = \arg\max_{|\Omega_C|=K_B K_p} \sum_{q \in \Omega_C} \hat{p}_q.
\]
In our simulations, we consider the short packet transmission in the uplink mMTC scenarios. The inter-block sparsity $K_B$ and intra-block sparsity $K_{B_i}$ are known at both the transmitter and receiver. As a spreading codebook, we employ the random Bernoulli sequences ($c_{ij} \in \{+1, -1\}$) and set the dimension of the codebook to $128 \times 256$. In the inter-block decoding, we use the BOMP algorithm where the stopping criteria is described as $k \leq K_B$ ($k$ is the iteration index). In the intra-block decoding, OMP stops the iteration if the $K_{B_i} K_{B_i}$ support elements are chosen. For the DL-PLOS network, we set the learning rate to 0.0001, batch size to 100, and the number of RBs to 6 ($L = 6$). As a performance measure, we use the success probability of support identification $P_{\text{succ}}$, which is defined as the percentage of the accurately decoded support elements among the original support.

In Fig. 4, we evaluate $P_{\text{succ}}$ of inter-block support decoding of DL-PLOS and conventional BOMP algorithm. We observe that DL-PLOS outperforms the BOMP algorithm for all SNR regime. For example, when $K_B = 2$, the DL-PLOS scheme achieves around 4 dB gain over BOMP at $P_{\text{succ}} = 0.9$.

In Fig. 5, we plot $P_{\text{succ}}$ of the intra-block support decoding of DL-PLOS and OMP. We observe that DL-PLOS outperforms OMP by a large margin, achieving more than 10 dB gain over OMP at $P_{\text{succ}} = 0.9$. Also, we can see that DL-PLOS maintains its robustness even when $K_{B_i}$ increases. For instance, when $K_{B_i}$ increases from 2 to 4, performance of DL-PLOS does not change in the high SNR regime but the performance of OMP algorithm is degraded sharply from 0.88 to 0.34.

V. Conclusion

In this paper, we proposed a novel pilot-less one-shot transmission scheme suitable for the short packet transmission. The key idea behind the proposed scheme is to encode control information to the inter-block support and data information to the intra-block support of a sparse vector and then to decode the packet using the deep learning-based multi-label classification technique. The DL-PLOS transmission scheme does not require the pilot signaling, thereby saving the transmission power, receiver processing time and cost for the channel estimation, and also can be easily extended to the single-input multiple-output (SIMO) and multiple-input multiple-output (MIMO) configurations.