

A Supervised-Learning Detector for Multihop Distributed Reception Systems

Seonho Kim and Song-Nam Hong

Abstract—We consider a multihop distributed uplink reception system in which K users transmit independent messages to one data center of $N_r \geq K$ receive antennas, with the aid of multihop intermediate relays. In particular, each antenna of the data center is equipped with one-bit analog-to-digital converters (ADCs) for the sake of power-efficiency. In this system, it is extremely challenging to develop a low-complexity detector due to the non-linearity of an end-to-end channel transfer function (created by relays' operations and one-bit ADCs). Furthermore, there is no efficient way to estimate such complex function with a limited number of training data. Motivated by this, we propose a supervised-learning (SL) detector by introducing a novel Bernoulli-like model in which training data is directly used to design a detector rather than estimating a channel transfer function. It is shown that the proposed SL detector outperforms the existing SL detectors based on Gaussian model for one-bit quantized (binary observation) systems. Furthermore, we significantly reduce the complexity of the proposed SL detector using the fast k NN algorithm. Simulation results demonstrate that the proposed SL detector can yield an attractive performance with a significantly lower complexity.

Index Terms—Multihop distributed reception system, data detection, classification, one-bit ADC.

I. INTRODUCTION

A distributed uplink reception system is a special case of a multi-source single-destination multihop relay network where multiple sources send independent messages to one destination of a large number of antennas with the help of multihop intermediate relays. In this system, numerous information-theoretical approaches have been proposed in [1]–[4], with the assumption that the destination perfectly knows all channel transfer functions (or at least end-to-end channel transfer function). A quantized-remap-and-forward (QMF) (extended in [2] where it is referred to as noisy network coding (NNC)) was presented in [1], which achieves the best-known performance. However, it is not practical as joint typical detector at the destination is prohibitive and the assumption of perfect channel state information is unrealistic. A more practical approach based on lattice code, named compute-and-forward (CoF), was presented in [5]–[7], which can significantly decrease the detection complexity, by converting the non-linear end-to-end channel transfer function into a linear one. However, its performance is not satisfactory for multihop relay networks

with realistic channels (e.g., Rayleigh fading), due to a severe non-integer penalty [6]. Therefore, it is still an open problem to develop a practical detection and channel estimation methods for a multihop communication system.

In a distributed uplink reception system, the use of a large number of receive antennas at data center is necessary to support multiple sources simultaneously. Unfortunately, it can highly increase the hardware cost and the radio-frequency (RF) circuit consumption [8]. Especially, a high-resolution analog-to-digital converter (ADC) is most problematic as the power consumption of an ADC is scaled exponentially with the number of quantization bits and linearly with the baseband bandwidth [9], [10]. To overcome this, the use of low-resolution ADCs (e.g., 1 \sim 3 bits) has received increasing attention for a large-scale multiple-input-multiple-output (MIMO) system [11]–[14]. The one-bit ADC is particularly attractive as it does not need an automatic gain controller [15]. In this sense, we consider a multihop distributed uplink reception system in which each receive antenna of the data center is equipped with one-bit ADCs. For this system, an end-to-end channel transfer function between K users and the data center is highly non-linear. Thus, it is extremely challenging to estimate such function with a limited number of one-bit quantized pilot signals. This motivates us to consider a data-driven supervised-learning (SL) detector in which the pilot signals (or training data) are exploited to directly learn a MIMO detector rather than estimating a complicated non-linear channel transfer function.

Very recently, SL detectors have been developed in [11] for MIMO systems with one-bit ADCs. It is remarkable that these methods are developed by assuming that data is generated from a Gaussian distribution. Although it is widely used, this model might not be suitable for binary data (e.g., one-bit quantized observations). In this paper, we propose a novel Bernoulli-like model which can be more suitable for binary random outputs. It is verified by showing that the proposed SL detector outperforms the existing SL detectors in [11]. Despite its superior performance, the complexity of the proposed SL detector (also, the existing SL detectors in [11]) is problematic as a search-space grows exponentially with the number of users K . This is the major drawback to be used in practice. We address this problem by presenting a low-complexity SL (LSL) detector using the *fast kNN* algorithm. The fast k NN algorithm, which can find a closest point in Hamming space fastly using an efficient data structure, enables to efficiently remove unnecessary elements in the search-space according to a given current observation. Thus, the LSL detector can perform over the significantly reduced search-

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Department of Electrical Engineering, Ajou University, Suwon, Korea (e-mail: {kimsh1005, snhong}@ajou.ac.kr).

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1702-00.

space. Simulation results demonstrate that the proposed LSL detector can yield an attractive performance with a practically manageable complexity.

This paper is organized as follows. In Section II, we describe a multihop distributed uplink reception system. In Section III, we briefly review the various existing SL detectors. In Section IV, we propose a novel SL detector based on a Bernoulli-like model. In Section V, we significantly reduce the complexity of the proposed SL detector by efficiently using the fast kNN algorithm. Section VI provides the simulation results to verify the superiority of the proposed SL detector. Finally, conclusion is provided in Section VII.

II. SYSTEM MODEL

We consider a multihop distributed uplink reception system where K sources transmit independent messages to one data center with the help of intermediate relays. In particular, the data center is equipped with $N_r \geq K$ receive antennas with one-bit ADCs. Let $w_k \in \{0, \dots, m-1\}$ denote the source k 's message for $k \in \{1, \dots, K\}$, each of which contains the $\log m$ information bits. We also denote m -ary constellation set by $S = \{s_0, \dots, s_{m-1}\}$ with power constraint $\frac{1}{m} \sum_{i=0}^{m-1} |s_i|^2 = P_t$. Let $\text{sign}(\cdot) : \mathbb{R} \rightarrow \{-1, 1\}$ represent the one-bit ADC quantizer function with $\text{sign}(u) = 1$ if $u \geq 0$ and $\text{sign}(u) = -1$, otherwise. Then, the transmitted symbol of the source k , \tilde{x}_k , is obtained by a modulation function $f : W \rightarrow S$ as $\tilde{x}_k = f(w_k) \in S$. Then, by converting a complex-valued scalar into an equivalent real-valued vector, the data center observes

$$\mathbf{r} = \text{sign}(\Phi(\tilde{\mathbf{x}}) + \tilde{\mathbf{z}}) \in \{-1, 1\}^N, \quad (1)$$

where $N = 2N_r$ and $\Phi(\cdot)$ represents a complex *non-linear* function (called end-to-end channel transfer function). Also, $\tilde{\mathbf{z}} = [\tilde{z}_1, \dots, \tilde{z}_N] \in \mathbb{R}^N$ denotes the noise vector whose elements are independent and identically distributed as circularly symmetric complex gaussian random variables with zero-mean and variance σ_z^2 , i.e., $\tilde{z}_i \sim \mathcal{CN}(0, \sigma_z^2)$.

It is remarkable that $\Phi(\cdot)$ can capture all the intermediate relays' operations and all the local wireless channels in the network. Although the proposed method in this paper can be applied to any relay's operation and local channel model, we assume that in our simulations, each relay with a single antenna performs an amplify-and-forward (AF) and each local channel is assumed as Rayleigh fading. Also, for the simplicity, it is assumed that each relay has the same power constraint with the sources as P_t and all the additive noises at the receivers in the network are circularly symmetric complex gaussian random variables with zero-mean and variance σ_z^2 . We define the signal-to-noise ratios (SNRs) as $\text{SNR} = P_t/\sigma_z^2$.

The proposed communication framework consists of training and data transmission phases (see Fig. 1). Note that during these phases, a wireless channel is assumed to be fixed.

- **Training phase:** In this phase, K sources transmits "known" sequences (i.e., pilot signals) so that the data center can learn a non-linear function $\Phi(\cdot)$. With machine-learning perspective, the data center collects the data and the corresponding labels. Let $\mathcal{M} = \{0, \dots, m-1\}^K$ denote the set of all possible messages of the K

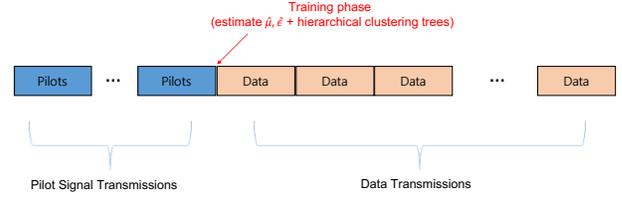


Fig. 1. Illustration of the training and data transmission phases within a coherence time.

sources. For each class $c \in \mathcal{M}$, the K sources transmit T pilot signals $\tilde{\mathbf{x}}_i^c$ for $i = 1, \dots, T$. From (1), the data center can collect the *labelled* data set

$$\mathcal{D} = \{\tilde{\mathbf{r}}_i^c \in \{-1, 1\}^N : c \in \mathcal{M}, i = 1, \dots, T\}. \quad (2)$$

- **Data transmission phase:** Given the \mathcal{D} and a new observation \mathbf{r} , the data center detects the class of \mathbf{r} (i.e., users' messages $\hat{\mathbf{w}} = (\hat{w}_1, \dots, \hat{w}_K)$) as

$$\Psi_{\mathcal{D}}(\mathbf{r}) = c \in \mathcal{M}, \quad (3)$$

which is what we will propose in this paper.

III. SL DETECTORS FOR BINARY DATA

In machine-learning perspective, the above detection problem (a.k.a., the supervised-learning problem) can be categorized into two approaches [16] as *non-parametric* and *parametric* learnings. A non-parametric learning does not require a priori knowledge on data set \mathcal{D} (e.g., a distribution of data) such as k -nearest neighbor (kNN), decision tree, and support vector machine (SVM). Whereas, in parametric learnings as logistic regression, naive bayes, and neural networks, data is assumed to be generated from a probabilistic model with some parameters (e.g., Gaussian model). Then, they are optimized from the given data set \mathcal{D} . Therefore, it is very important to choose a proper probabilistic model based on a priori knowledge (or domain knowledge) on the data set \mathcal{D} .

We briefly review the existing (parametric or non-parametric) SL detectors. It is noticeable that they can be immediately applied to a distributed reception system since the SL detector do not rely on system models.

- **Non-parametric learning:** In [11], empirical maximum-likelihood detector (eMLD) and minimum mean distance detector (MMD) have been presented. The eMLD can be viewed as kNN classifier where the k nearest data points from a new observation (or received signal) are identified and then, the majority voting is performed to find a class (e.g., users' messages). Also, the MMD is the special case of eMLD with $k = 1$ for the purpose of low-complexity.
- **Parametric learning:** In this approach, it is most important to seek a proper probabilistic model for a given data set \mathcal{D} . As in [11], [16], a Gaussian model is widely used where the data $\mathbf{r} \in \mathcal{D}$ is assumed to be generated from the probability distribution $P(\mathbf{r}|c, \theta_c) = \mathcal{N}(\boldsymbol{\mu}_c, \Sigma_c)$. Here, $c \in \mathcal{M}$ denotes the class (or message) of the K sources and θ_c represents the parameter vector for the class c . Using the given $\mathcal{D} = \{\tilde{\mathbf{r}}_i^c : i = 1, \dots, T\}$, we can optimize

$\theta_c = (\hat{\boldsymbol{\mu}}_c, \hat{\Sigma}_c)$ via maximum likelihood (ML) estimation as

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{T} \sum_{t=1}^T \tilde{\mathbf{r}}_t^c \quad (4)$$

$$\hat{\Sigma}_c = \frac{1}{T} \sum_{t=1}^T (\tilde{\mathbf{r}}_t^c - \hat{\boldsymbol{\mu}}_c)(\tilde{\mathbf{r}}_t^c - \hat{\boldsymbol{\mu}}_c)^\top, \quad (5)$$

where $\hat{\boldsymbol{\mu}}_c$ and $\hat{\Sigma}_c$ represent the mean and the covariance of the training data associated with the class c , respectively. When the training data is not sufficient, the covariance matrix tends to be rank-deficient and ill-conditioned. This problem can be resolved by shrinkage estimator [17]. Given the $\hat{\boldsymbol{\theta}}_c = (\hat{\boldsymbol{\mu}}_c, \hat{\Sigma}_c)$, the optimal ML detector is derived as

$$\Psi_{\mathcal{D}}(\mathbf{r}) = \operatorname{argmin}_{c \in \mathcal{M}} (\mathbf{r} - \hat{\boldsymbol{\mu}}_c)^\top \hat{\Sigma}_c^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_c). \quad (6)$$

In particular, the distance measure in the above is referred to as Mahalanobis distance, and the inverse matrix of $\hat{\Sigma}_c$ in (5) is called a precision matrix. When $\Sigma_c = \mathbf{I}$ for all c , as a special case, the resulting detector is equivalent to the Minimum-Centered-Distance (MCD) detector proposed in [11].

It was shown in [11] that, among the above SL detectors, MCD and eMLD detectors show the best performances. Since the complexity of eMLD is higher than MCD, the latter was highly recommended. However, one can argue that Gaussian model in (6) might not be suitable to model the distribution of binary data $\mathbf{r} \in \{1, -1\}^N$. This motivates us to propose a SL detector using a novel Bernoulli-like probabilistic model (see Section IV).

IV. THE PROPOSED (PARAMETRIC) SL DETECTOR

We propose a novel SL detector based on a Bernoulli-like model, where data is assumed to be generated from the following probability distribution:

$$P(\mathbf{r}|c, \boldsymbol{\theta}_c) = \prod_{i=1}^N \mathbf{1}_{\{r_i \neq \mu_{c,i}\}} (1 - \epsilon_{c,i}) \mathbf{1}_{\{r_i = \mu_{c,i}\}}, \quad (7)$$

where $\boldsymbol{\theta}_c = (\boldsymbol{\mu}_c, \boldsymbol{\epsilon}_c)$ for $c \in \mathcal{M}$, $\epsilon_{c,i} < 0.5$ for all i , and $\mathbf{1}_{\{\mathcal{A}\}}$ represents an indicator function with $\mathbf{1}_{\{\mathcal{A}\}} = 1$ if \mathcal{A} is true, and $\mathbf{1}_{\{\mathcal{A}\}} = 0$, otherwise. Given the training data for the class c (e.g., $\{\tilde{\mathbf{r}}_t^c : t = 1, \dots, T\}$), the parameter vector $\boldsymbol{\theta}_c$ is optimized using ML estimation as

$$(\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\epsilon}}_c) = \operatorname{argmax}_{(\boldsymbol{\mu}_c, \boldsymbol{\epsilon}_c)} \prod_{t=1}^T P(\tilde{\mathbf{r}}_t^c | \boldsymbol{\mu}_c, \boldsymbol{\epsilon}_c). \quad (8)$$

By plugging (7) into (8), the optimal parameters are obtained by taking the solutions of

$$(\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\epsilon}}_c) = \operatorname{argmax}_{(\boldsymbol{\mu}_c, \boldsymbol{\epsilon}_c)} \prod_{i=1}^N \prod_{t=1}^T \epsilon_{c,i}^{\mathbf{1}_{\{\tilde{r}_{t,i}^c \neq \mu_{c,i}\}}} (1 - \epsilon_{c,i})^{\mathbf{1}_{\{\tilde{r}_{t,i}^c = \mu_{c,i}\}}}.$$

For any $\epsilon_{c,i} < 0.5$, we can see that the above objective function is maximized by taking

$$\hat{\mu}_{c,i} = \operatorname{sign} \left(\sum_{t=1}^T \tilde{r}_{t,i}^c \right) \text{ for } i = 1, \dots, N, \quad (9)$$

independently from the choices of $\epsilon_{c,i}$'s. We let

$$N_d = \sum_{t=1}^T \mathbf{1}_{\{\tilde{r}_{k,i}^c \neq \hat{\mu}_{c,i}\}} \text{ and } N_s = \sum_{t=1}^T \mathbf{1}_{\{\tilde{r}_{k,i}^c = \hat{\mu}_{c,i}\}}. \quad (10)$$

Then, we can find an optimal $\epsilon_{c,i}$ independently from the other $\epsilon_{c,j}$'s with $i \neq j$ by taking the solution of $\operatorname{argmax}_{\epsilon_{c,i}} \epsilon_{c,i}^{N_d} (1 - \epsilon_{c,i})^{N_s}$. Taking $\frac{\partial(\epsilon_{c,i}^{N_d} (1 - \epsilon_{c,i})^{N_s})}{\partial \epsilon_{c,i}} = 0$, the optimal $\epsilon_{c,i}$ is obtained as

$$\hat{\epsilon}_{c,j} = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\{\hat{\mu}_{c,j} \neq \tilde{r}_{j,i}^c\}}. \quad (11)$$

With the parameter vector $\hat{\boldsymbol{\theta}}_c = (\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\epsilon}}_c)$ in (9) and (11), the optimal ML estimator (i.e., the proposed SL detector) is derived as

$$\Psi_{\mathcal{D}}(\mathbf{r}) = \operatorname{argmin}_{c \in \mathcal{M}} (\mathbf{r} - \hat{\boldsymbol{\mu}}_c)^\top \mathbf{diag}[-\log \hat{\epsilon}_{c,i}] (\mathbf{r} - \hat{\boldsymbol{\mu}}_c), \quad (12)$$

where $\mathbf{diag}[d_i]$ denotes the diagonal matrix with the i -th diagonal element d_i and its dimension is easily obtained from the context.

V. THE PROPOSED LSL DETECTOR

In the proposed SL detector in (12), the computational complexity is expensive as the size of search-space (e.g., $|\mathcal{M}| = m^K$) grows exponentially with K . To address this problem, we present a low-complexity SL (LSL) detector which is performed over the *reduced* search-space. The major contribution of this section is to build the reduced search-space by efficiently removing unnecessary candidates from the \mathcal{M} according to a current observation \mathbf{r} .

The proposed method to yield the reduced search-space can be outlined as follows (see Fig 1):

Training phase: From the training data $\mathcal{D} = \{\mathbf{r}_t^c : t = 1, \dots, T, c \in \mathcal{M}\}$, the parameters for the proposed SL detector are obtained from (9) and (11) as $\hat{\mathcal{U}} = \{\hat{\boldsymbol{\mu}}_c : c \in \mathcal{M}\}$ and $\hat{\mathcal{E}} = \{\hat{\boldsymbol{\epsilon}}_c : c \in \mathcal{M}\}$. Then, $\hat{\mathcal{U}}$ is decomposed using k -medoids clustering in [18], yielding a hierarchical clustering tree (see Algorithm 1). This algorithm starts with all the elements in \mathcal{D} and decomposes them into J clusters, where J is a parameter of the algorithm and called branching factor. The clusters are constructed by selecting J elements randomly as cluster centroids and then by assigning other elements to one of the clusters with the closest centroid. The algorithm is repeated recursively until the number of elements in each cluster is below the maximum leaf size J , where in this case, that node becomes a leaf node. In addition, Algorithm 1 is performed over W times to construct the W trees having possibly different decomposition structures, denoted by $\{\mathcal{T}_1, \dots, \mathcal{T}_W\}$. The use of the multiple trees can improve the quality of the resulting reduced search space.

Data transmission phase: Given a current observation \mathbf{r} , the search algorithm begins with traversing multiple trees in parallel. Note that W multiple trees share a single priority queue (\mathcal{Q}) where the nodes in the priority queue are arranged in the shortest Hamming distance order, with respect to the current observation \mathbf{r} . Then, it can efficiently produce the reduced search-space $\mathcal{S}(\mathbf{r}) \subseteq \mathcal{M}$ which only contains the

nearest $\hat{\mu}_c$'s to the \mathbf{r} . The detailed procedures are given in Algorithm 2. Then, the proposed SL detector is performed as

$$\Psi_{\mathcal{D}}(\mathbf{r}) = \underset{c \in \mathcal{S}(\mathbf{r})}{\operatorname{argmin}} (\mathbf{r} - \hat{\mu}_c)^{\top} \operatorname{diag} [-\log \hat{\epsilon}_{c,i}] (\mathbf{r} - \hat{\mu}_c). \quad (13)$$

Algorithm 1 Hierarchical clustering tree $h(\hat{\mathcal{U}}, J)$

Input: $\hat{\mathcal{U}} = \{\hat{\mu}_1, \dots, \hat{\mu}_{|\mathcal{M}|}\}$

Output: hierarchical clustering tree \mathcal{T}

Parameter: J (maximum leaf size and branching factor)

```

if  $|\hat{\mathcal{U}}| < J$  then
    create leaf node with the elements in  $\hat{\mathcal{U}}$ 
else
     $\mathcal{P} \leftarrow$  select  $J$  elements randomly from  $\hat{\mathcal{U}}$  (centroids)
     $\mathcal{C} \leftarrow$  cluster the elements in  $\hat{\mathcal{U}}$  with the centroids  $\mathcal{P}$ 
    for do each cluster  $C_i \in \mathcal{C} = \{C_1, \dots, C_J\}$ 
        create non-leaf node with centroid  $\mathcal{P}_i$ 
        recursively apply the algorithm  $h(C_i, J)$  (with the
        updated  $C_i$ )
    end for
end if

```

From now on, we will analyze the computational complexity of the proposed LSL detector which consists of construction and search complexities. Note that the construction complexity is taken only once during each coherence time and thus, it can be negligible when the coherence time is sufficiently large (i.e., a channel is slowly changed). Recall that N denotes the observation dimensionality (e.g., $N = 2N_r$) and m^K denote the number of all possible messages of K users.

Construction complexity: On the construction of k -medoids hierarchical trees, the complexity of distance-computation is equal to $\mathcal{O}(m^K N)$ with respect to a node selected as cluster centroid. Given the branch factor J , it constructs J distinctive clusters at each tree level and thus, the corresponding complexity is equal to $\mathcal{O}(m^K N J)$. Assuming that W multiple trees have balanced structures, the height of the tree will be $K \log_J m$. Then, the overall complexity for the forest construction is equal to $\mathcal{O}(m^K N J W K \log_J m)$.

Search complexity: First, it starts with traversing the forest $\{\mathcal{T}_1, \dots, \mathcal{T}_W\}$ simultaneously. Each node computes the distances with the J child nodes to find the closest node at each level. This computation repeats until it reaches to a leaf node at level $K \log_J m$ for each tree \mathcal{T}_i . The corresponding complexity is $\mathcal{O}(W J N K \log_J m)$. According to the Algorithm 2, it stops after examining L_{\max} elements. Except the first step, the remaining steps start with a node popped out from priority queue, which is likely to be located at a level between root and leaf. Assuming that every search returns J elements in a reached leaf node and the starting point is the root, the corresponding complexity is equal to $\mathcal{O}(W J N K \log_J m + (L_{\max} - W J) N K \log_J m)$. Note that this complexity is an upper bound and an actual complexity becomes lower both when leaf nodes of the trees return duplicate codes simultaneously and when trees would have skewed structures. Also, the search complexity for priority queue is negligible compared to the tree search complexity.

Algorithm 2 Searching parallel hierarchical clustering trees

Input: hierarchical clustering trees $\{\mathcal{T}_i : i = 1, \dots, W\}$ and a new observation \mathbf{r}

Output: $\mathcal{S}(\mathbf{r})$ (reduced search space associated with \mathbf{r})

Parameter: L_{\max} (the desired size of a reduced search space, e.g., $|\mathcal{S}(\mathbf{r})| = L_{\max}$)

```

1:  $\ell \leftarrow 0$  ( $\ell =$  number of points  $\hat{\mu}_c$  searched)
2:  $\mathcal{Q} \leftarrow$  empty priority queue
3:  $\mathcal{R} \leftarrow$  empty priority queue
4: for each tree  $\mathcal{T}_i$  do
5:     call  $\text{TraverseTree}(\mathcal{T}_i, \mathcal{Q}, \mathcal{R})$ 
6: end for
7: while  $|\mathcal{Q}| \neq 0$  and  $\ell < L_{\max}$  do
8:      $j \leftarrow$  top index of  $\mathcal{Q}$ 
9:     call  $\text{TraverseTree}(j, \mathcal{Q}, \mathcal{R})$ 
10: end while
11: return  $K$  top points from  $\mathcal{R}$ 

1: procedure  $\text{TRAVERSE TREE}(j, \mathcal{Q}, \mathcal{R})$ 
2:     if node  $j$  is a leaf node then
3:          $\mathcal{S} \triangleq$  {all the elements in the leaf node  $j$ }
4:          $\mathcal{R} = \mathcal{R} \cup \mathcal{S}$  and  $\ell \leftarrow \ell + |\mathcal{S}|$ 
5:     else
6:          $\mathcal{C} \leftarrow$  child nodes of node  $j$ 
7:          $i \leftarrow$  closest node of  $\mathcal{C}$  to the  $\mathbf{r}$ 
8:          $\mathcal{C}_p \leftarrow \mathcal{C} \setminus \{i\}$ 
9:         add all nodes in  $\mathcal{C}_p$  to  $\mathcal{Q}$ 
10:        call  $\text{TraverseTree}(i, \mathcal{Q}, \mathcal{R})$ 
11:     end if
12: end procedure

```

In sum up, the overall complexity can be well-approximated as

$$\mathcal{O}(N L_{\max} (K \log_J m + 1)), \quad (14)$$

where $N L_{\max}$ accounts for the detection complexity in (13). It is noticeable that the complexity of the proposed low-complexity SL detector grows linear with K while the other SL detectors grow exponentially with K . The approximated complexity in (14) will be used in Section VI to compute the complexity of the proposed LSL detector.

VI. NUMERICAL RESULTS

We evaluate the average bit-error-rate (BER) performances of the proposed SL detector over the existing SL detectors in [11]. Also, it is shown that the proposed low-complexity SL detector achieves the original performance with much lower complexity. For the simulations, QPSK modulation and Rayleigh fading are assumed. When a training overhead is small (e.g., T is small), an empirical error-probability (e.g., $\epsilon_{c,i}$) can be underestimated as zero although it is indeed not. Since this can cause severe error-floor problem, we assign a minimum value of $\hat{\epsilon}_{c,i}$ as 10^{-3} .

Fig. 2 shows the BER performances of the proposed SL detector and the existing one in [11]. Here, the number of training for each c is set by 15 (e.g., $T = 15$). We considered the two hop distributed reception network for the following

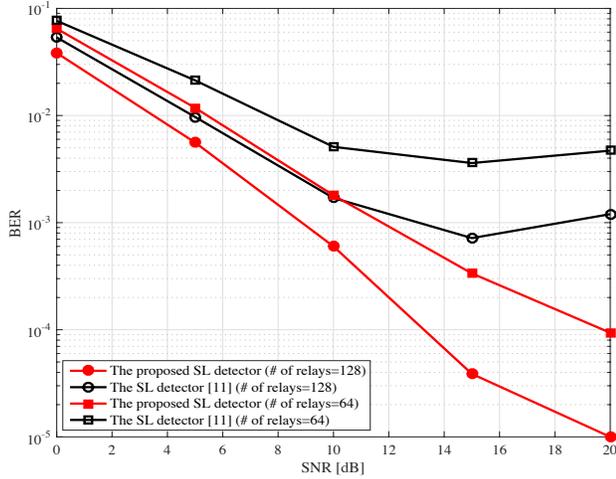


Fig. 2. $K = 8$, $N_r = 64$, and $T = 15$. Performance comparisons of the proposed SL detector and the existing SL detectors.

two scenarios: i) 64 intermediate relays; ii) 128 intermediate relays. From Fig. 2, we can observe that the proposed SL detector outperforms the existing SL detector, which implies that the proposed Bernoulli-like model is more suitable to binary data than Gaussian model.

Fig. 3 shows the BER performances of the proposed low-complexity detector according to L_{\max} in Algorithm 2. Also, we set $J = 32$ in Algorithm 1. In this simulation, the benefit of low-complexity detector stands out, since it can achieve the optimal performance perfectly with only 6% of original complexity. Thus, it is expected that the use of low-complexity technique is more beneficial for a large-scale distributed reception system (e.g., a large K).

VII. CONCLUSION

We proposed a supervised-learning (SL) detector by introducing a novel Bernoulli-like model as data probability distribution. Differently from a widely used Gaussian model, it can exploit the structure of binary data (e.g., one-bit quantized observation). We further developed the low-complexity SL detector with the aid of the fast kNN algorithm. Simulation results demonstrated that the proposed low-complexity detector almost achieves the original performance with a significantly lower complexity. Therefore, the proposed detector would be a good practical candidate for multihop distributed reception systems. We would like to emphasize that the proposed SL detector can be straightforwardly applied to any multihop relay network with a single destination. On going work, we are investigating to generalize the Bernoulli-like model by capturing the correlation of elements in an observed data. Also, it is an interesting future work to extend the proposed SL detector for the multihop communication systems with low-resolution ADCs (e.g., 2~3-bit ADCs).

REFERENCES

[1] S. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: a deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 1872-1905, Apr. 2011.

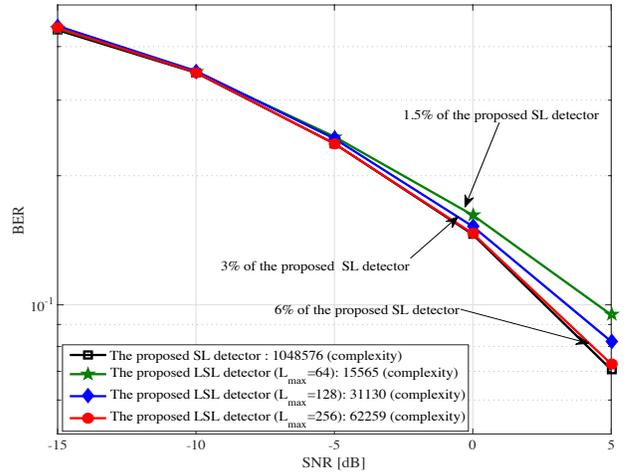


Fig. 3. $K = 14$, $N_r = 64$, and $T = 15$. Performances of the proposed low-complexity SL detector according to the size of reduced search space.

[2] S. Lim, Y. H. Kim, A. E. Gamal, and S. Chung, "Noisy network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 3132-3152, May 2011.

[3] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, "Robust and efficient distributed compression for cloud radio access networks," *IEEE Trans. Veh. Tech.*, vol. 62, no. 2, pp. 692-703, Feb. 2013.

[4] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, "Robust layered transmission and compression for distributed uplink reception in cloud radio access networks," *IEEE Trans. Veh. Tech.*, vol. 63, no. 1, pp. 204-216, Jan. 2014.

[5] B. Bobak and M. Gastpar, "Compute-and-forward: harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463-6486, Oct. 2011.

[6] S.-N. Hong and G. Caire, "Compute-and-forward strategies for cooperative distributed antenna systems," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5227-5243, Sep. 2013.

[7] S.-N. Hong, Y.-S. Jeon and N. Lee, "Distributed uplink reception in cloud radio access networks: a linear coding approach," *IEEE Trans. Veh. Tech.*, vol. 67, no. 2, pp. 1470-1481, Feb. 2018.

[8] Y. Hong and T. L. Marzetta, "Total energy efficiency of cellular large scale antenna system multiple access mobile networks," in *Proc. IEEE Conf. Green Commun. (GreenCom)*, pp. 29-31, Piscataway, NJ, Oct. 2013.

[9] B. Murmann, "ADC Performance Survey 1997-2015," [Online]. Available: <http://web.stanford.edu/~murmman/adcsurvey.html>.

[10] A. Mezghani and J. A. Nossek, "Modeling and minimization of transmitter power consumption in wireless networks," in *Proc. IEEE/ITG WSA*, pp. 1-8, Aachen, Germany, Feb. 2011.

[11] Y.-S. Jeon, S.-N. Hong and N. Lee, "Supervised-Learning-Aided Communication Framework for Massive MIMO Systems with Low-Resolution ADCs," *IEEE Trans. Veh. Tech.*, vol. 67, no. 8, pp. 7299-7313, Aug. 2018.

[12] S.-N. Hong, S. Kim, and N. Lee, "A weighted minimum distance decoding for uplink multiuser MIMO systems with low-resolution ADCs," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 1912-1924, May. 2018.

[13] S.-N. Hong and N. Lee, "Soft-Output Detector for Uplink MU-MIMO Systems with One-Bit ADCs," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 930 - 933, May. 2018.

[14] Y.-S. Jeon, N. Lee, S.-N. Hong, and R. W. Heath, Jr., "One-bit sphere decoding for uplink massive MIMO systems with one-bit ADCs," *IEEE Trans. Wireless. Commun.*, vol. 17, no. 7, pp. 4509-4521, Jul. 2018.

[15] S. Hoyos, B. M. Sadler and G. R. Arce, "Monobit digital receivers for ultrawideband communications," *IEEE Trans. Wireless. Commun.*, vol. 4, no. 4, pp. 1337-1344, Jul. 2005.

[16] K. V. Murphy, *Machine learning: a probabilistic perspective*, MIT Press, 2012.

[17] J. Schfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Stat. Appl. Genet. Mol. Biol.*, Nov. 2005.

[18] M. Marius and D. G. Lowe, "Fast matching of binary features," in *Proc. 2012 Ninth Conf. Comp. Robot Vision (CRV)*, pp. 404-410, May 2012.