Signal Processing 205 (2023) 108884

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Estimation of user activity prior for active user detection in massive machine type communications☆

Syed Ali Irtaza[a,*], Salma Riaz[a], Ali Nauman[b], Muhammad Ali Jamshed[c], Sung Won Kim[b]

[a] Institute of Space Technology, Islamabad, Pakistan
[b] Department of Information and Communication Engineering, Yeungnam University, Republic of Korea
[c] James Watt School of Engineering, University of Glasgow, Glasgow, United Kingdom

ARTICLE INFO

Article history:
Received 20 May 2022
Revised 15 November 2022
Accepted 6 December 2022
Available online 9 December 2022

Keywords:
Massive machine type communications
Active user detection
Prior user activity information
Compressed sensing
Grant free access

ABSTRACT

To enable massive-scale connectivity with low latency and efficient resource usage for massive machine type communications, the grant free access scheme has been proposed, which allows users to communicate with low signaling overhead and without the need for any prior resource allocation procedure. Due to the sporadic nature of packet transmission and non-orthogonal multiplexing, access points need to perform active user detection (AUD) to identify which users have sent the packets based on the received data. In this paper, we propose an enhanced AUD algorithm, which exploits each user's activity pattern for detecting active users. We assume that each user randomly sends its own packet with different probability distributions parameterized by the *user activity probability* (UAP). In our work, such UAPs are inferred from the trajectory of the measurements collected over a certain period of time. Then, the estimated UAPs are incorporated into the compressed sensing-based algorithm for the joint AUD and channel estimation. Using the expectation-maximization algorithm, our method can efficiently find the maximum likelihood estimate of the UAPs for all users. We also present an on-line algorithm that sequentially updates the UAP estimates for each measurement. Our numerical evaluation demonstrates the benefit of the UAP estimation for compressed sensing-based AUD methods.

1. Introduction

The present era demands wireless communications in various disciplines including public communications, high-speed data transfer, industrial automation, health monitoring, and transportation. In addition to the traditional human-central communications, next-generation wireless technologies are expected to support automated machine-type communications where hundreds of devices are connected to the internet, and information is exchanged to provide new services. To support machine-centric communications, international telecommunication union (ITU) has defined such new use-cases of communications as massive machine-type communication (MMTC) [1]. MMTC is distinct from conventional communication paradigms in that it pursues the multi-dimensional goal of massive connectivity, longer off-grid unmanned operations, and low latency and low power communications.

In order to support massive devices in the limited resource environments, grant-free access (GFA) has been proposed as a promising technique for implementing MMTC [2]. The users communicating in a GFA network can transmit packets to the access point (AP) at any time without user scheduling. This results in a significant reduction in the resources and latency required for interactive access protocols in conventional wireless systems. Owing to the potentially large number of users and limited coherence resources, non-orthogonal user identification codes are widely used for user multiplexing. However, this approach will induce a large amount of multi-user interference, limiting system performance. Fortunately, because of the sporadic nature of the packet transmissions in typical MMTC applications, only a fraction of users communicate in a particular time slot. To decode the data of these active users without a scheduling procedure, the AP needs to autonomously identify the users transmitting packets solely based on the received packets. This process is often called *active user detection* (AUD).

☆ This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A1A03039493) and in part by the NRF grant funded by the Korea government (MSIT) (NRF-2022R1A2C1004401).

* Corresponding author.
E-mail addresses: syed.ali@mail.ist.edu.pk (S.A. Irtaza), salma.riaz@mail.ist.edu.pk (S. Riaz), anauman@ynu.ac.kr (A. Nauman), muhammadali.jamshed@glasgow.ac.uk (M. Ali Jamshed), swon@yu.ac.kr (S. Won Kim).

https://doi.org/10.1016/j.sigpro.2022.108884
0165-1684/

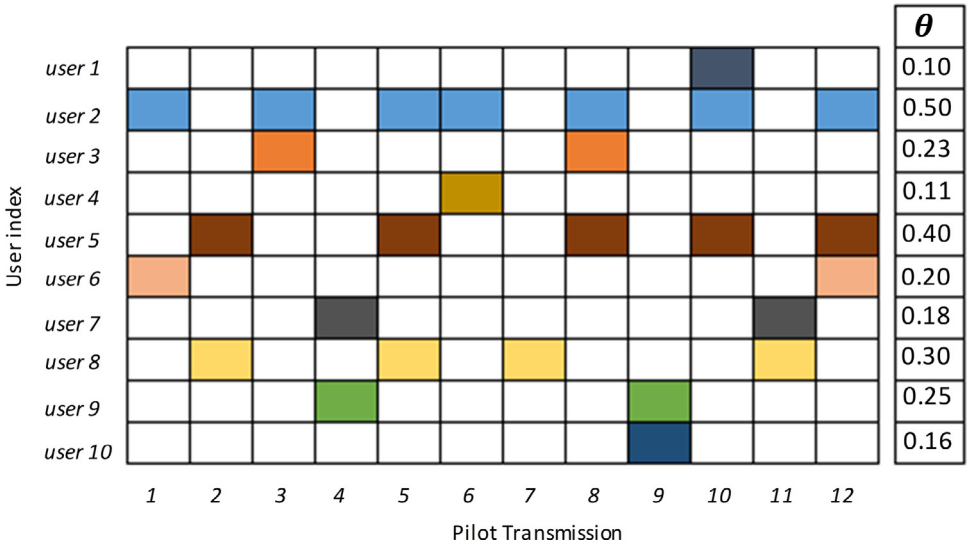

Fig. 1. Illustration of dynamic user activity samples for the given UAP $\boldsymbol{\theta}$.

Thus far, various AUD techniques have been proposed in the context of MMTC. Exploiting the sparse nature of user activity, compressed sensing (CS) techniques [3,4] have been widely used to detect active users. In [5–8,10–14], well-known CS recovery algorithms including basis pursuit denoising (BPDN) [18], orthogonal matching pursuit (OMP) [19], compressive sampling matching pursuit (CoSaMp) [20], subspace pursuit (SP) [21], and approximate message passing (AMP) [22] have been used to identify the support of the vector associated with the users' data symbols to be transmitted. In [5–9], user activity and data detection were jointly performed under the CS framework with the knowledge of channel information. In [10–13,15], by processing the pilot symbols contained in the data packet, joint AUD and channel estimation were performed. In [16,17], *data-aided* AUD algorithms, which exploit the information provided by both pilot and data symbols, were proposed. Recently, a considerable amount of effort has been made to enhance AUD performance using a particular structure of the underlying user sparsity. Under the assumption that the support of the user vectors is common over several packet transmissions, AUD methods recovering such joint sparsity were proposed in [23–26,32]. In [32], deep-neural network based architecture is proposed to exploit the block-sparse structure to perform joint AUD and channel estimation. In [27,28], a similar structure of the AUD algorithm recovering the joint sparsity was derived for multi-input multi-output (MIMO) setup, which recovered the common support over the measurements acquired from multiple antennas. In practical scenarios, user activity tends to vary slowly over consecutive packet transmissions; thus, partially common support models were adopted to develop the AUD algorithms [29,30]. Notable approaches handling the temporally correlated support include DCS-based MUD [29] and PIA-ASP [30].

In practical MMTC scenarios, various type of users exhibit different packet transmission behaviors and patterns. For example, some devices in the GFA network tend to transmit packets more frequently than others. If packet transmissions from a particular user can be viewed as a random event, each user will have a different probability of transmitting a packet at a particular instant. (see Fig. 1 for illustration.) From a Bayesian perspective, such prior user information can be used to enhance the performance of the AUD algorithm. There exist several works that have considered the use of such prior information in designing AUD algorithms [12,30,31,33,34]. In [12,33], the joint AUD and channel estimation was proposed, which used probabilistic information regarding user activity. In [31], the joint AUD, channel estimation and signal detection are performed using the prior information. In [30], prior user activity was obtained from previously estimated user activity using the temporal correlations of active user support sets. In [34], although not presented for AUD applications, two practical CS-recovery methods exploiting prior information on the sparsity pattern were proposed. It provides the theoretical foundation demonstrating the performance gain of prior information. Moreover, it is demonstrated that the performance gains are improved if the prior is sufficiently non-uniform as compared to uniform prior. While these approach might be applied to the AUD purpose, since the algorithm is developed under the assumption that prior user activity information is perfectly known, it cannot be directly used in practical scenarios. Methods to acquire such prior user activity information from the actual packet data have not yet been studied in depth.

In this work, we study the problem of analyzing user activity based on the sequence of measurements acquired during a certain period of time. Because the sequence of packet data reflects each users' footprints in packet transmission activity, we can learn the probability of the packet transmission from the sequential packet data. Such user activity probabilities (UAPs) are used as prior information in detecting active users in the AUD algorithm. Therefore, the proposed algorithm utilizes these packets for 1) UAPs estimation 2) joint active user detection (AUD) and the channel estimation. Specifically, we assume that the event of packet transmission for each user is the realization of a Bernoulli random variable with the parameter specified by the UAP. We aim to find the maximum likelihood (ML) estimate of the UAPs based on the sequence of measurements. In solving the problem, we use expectation-maximization algorithm to find the ML estimate of UAPs and use importance sampling to efficiently compute the log-likelihood function over a non-Gaussian distribution in a computationally efficient manner. We also present an online algorithm that sequentially updates the UAP estimates for a new measurement. Such online implementation enables our method to not store all measurements used for the UAP estimation. Finally, we present two CS-based AUD algorithms, called UAP-aware OMP and UAP-aware BPDN that exploit the prior user activity information for joint identification of active users and corresponding channel estimates. Our extensive simulations demonstrate that by using the

UAP estimate, the proposed scheme can achieve significant performance gains over conventional AUD methods. The main contributions of our paper are summarized as follows;

- We propose the algorithm that analyses the history of user activity over the sequence of received packets and infers the prior probability of packet transmission for each user. To our best knowledge, methods to infer user activity from the sequence of packet data in MMTC systems have not yet been proposed in the literature.
- Since the trajectory of user activity cannot be directly observed, it is not straightforward to derive the statistically optimal estimate of the UAP. With an aid of EM algorithm and importance sampling method, our algorithm can efficiently compute the UAP in an iterative fashion given the sequence of measurements.
- We present two CS-based AUD algorithms that can exploit the UAP information obtained by our method to jointly identify the active users and performs the channel estimation for the active users respectively. Our experiments confirm significant performance gains of the proposed AUD methods over the baseline methods.

The current work is applicable to massive antenna regime for the independently identically distributed (IID) user's activity pattern without any loss of generality. However, as a future research direction, it would be interesting to exploit the common support among the multiple antennas for UAP estimation.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the system model where the proposed AUD method is derived. In Section 3, we derive the EM algorithm for estimating UAPs based on the sequence of received packets. In Section 4, we present two CS-based AUD algorithms that exploit the UAP estimates as prior information. In Section 5, we present the simulation results and in Section 6, we conclude the paper.

2. System model

We consider a grant-free multiple access scenario for the MMTC uplink transmission, where a total of N users have a single antenna for establishing communication with the AP, which is also equipped with a single antenna. This communication setup can be easily extended to multiple antennas without any loss of generality. In the tth time slot, the kth user transmits its own packet containing the pilot symbol $p_{t,k} \in \mathbb{C}$ by spreading over the length-M pseudo-random unique user identification code $\boldsymbol{\phi}_k = [\phi_{k,1}, \ldots, \phi_{k,M}]^T$. We assume that among a total of N users, only $K(\leq N)$ active users transmit the data to the AP in one time slot, and the rest of the users remain silent during the whole transmission frame. The activity of the kth user at the tth time slot is represented by the binary random variable $s_{t,k}$, where $s_{t,k} = 1$ implies that the kth user is active and $s_{t,k} = 0$ implies that the kth user is inactive. The probability that $s_{t,k}$ equals one is specified by the parameter θ_k, called UAP. The higher value of θ_k indicates a higher likelihood that the kth user transmits the packet. The vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]^T$ shows the distribution of the UAPs of N users. The probabilistic density function of $s_{t,k}$ is given by the Bernoulli distribution

$$p(s_{t,k}; \theta_k) = (\theta_k)^{s_{t,k}}(1-\theta_k)^{1-s_{t,k}}. \tag{1}$$

Then, the measurement vector $\mathbf{y}_t$ received by the AP in the tth time slot can be expressed as

$$\mathbf{y}_t = \sum_{k=1}^{N} \boldsymbol{\phi}_k p_{t,k} s_{t,k} h_{t,k} + \mathbf{w}_t \tag{2}$$

$$= \boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\mathbf{h}_t + \mathbf{w}_t, \tag{3}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \ldots & \boldsymbol{\phi}_N \end{bmatrix} \tag{4}$$

$$\Lambda(\mathbf{s}_t) = \mathrm{diag}\left([s_{t,1}, \ldots, s_{t,N}]^T\right) \tag{5}$$

$$\mathbf{h}_t = [p_{t,1}h_{t,1}, \ldots, p_{t,N}h_{t,N}]^T \tag{6}$$

where $\mathbf{s}_t = [s_{t,1}, \ldots, s_{t,N}]^T$, $\mathbf{h}_t$ is the vector of the complex Gaussian channel gains following $\mathcal{CN}(\mathbf{0}, \sigma_h^2\mathbf{I})$ and $\mathbf{w}_t$ is the additive complex Gaussian noise vector $\mathcal{CN}(\mathbf{0}, \sigma_w^2\mathbf{I})$. Without loss of generality, we can assume that the pilot symbol is set to $p_{t,k} = 1$. We also assume that the matrix $\boldsymbol{\Phi}$ has a full row rank with $M < N$. Because only a small number of users transmit a packet for the given time slot, we assume that the structure of the vector $\mathbf{s}_t$ is sparse.

In summary, the AUD problem can be formulated as the joint estimation of the user activity $\mathbf{s}_t$ and channel $\mathbf{h}_t$ from the measurement vector $\mathbf{y}_t$. Such a joint estimation problem can be solved under the Bayesian framework for the given UAP $\boldsymbol{\theta}$. The key issue here is that such prior information given by the UAP is unknown in practice. In the next section, we discuss how to learn the UAP from the history of user activities observed in the received packets.

3. Proposed user activity prior estimation

In this section, we present the details of the proposed UAP estimation method.

3.1. Problem formulation

Since the history of the packet transmission activities for N users is captured in the sequence of l measurements, we aim to estimate the deterministic parameter $\boldsymbol{\theta}$ based on the l measurements $\mathbf{y}_{1:l} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_l\}$. The ML estimate of $\boldsymbol{\theta}$ for the given $\mathbf{y}_{1:l}$ is given by

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{y}_{1:l}; \boldsymbol{\theta}). \tag{7}$$

The sequence of binary user activity vectors $\mathbf{s}_{1:l} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_l\}$ contains the latent variables that cannot be directly observed. Thus, we can show that

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \int_{\mathbf{h}_{1:l}} \sum_{\mathbf{s}_{1:l}} p(\mathbf{y}_{1:l}, \mathbf{h}_{1:l}, \mathbf{s}_{1:l}; \boldsymbol{\theta}) d\mathbf{h}_{1:l}, \tag{8}$$

$$= \arg\max_{\boldsymbol{\theta}} \int_{\mathbf{h}_{1:l}} \sum_{\mathbf{s}_{1:l}} p(\mathbf{y}_{1:l}|\mathbf{h}_{1:l}, \mathbf{s}_{1:l}) p(\mathbf{h}_{1:l}) p(\mathbf{s}_{1:l}|\boldsymbol{\theta}) d\mathbf{h}_{1:l}, \tag{9}$$

where

$$p(\mathbf{y}_{1:l}|\mathbf{h}_{1:l}, \mathbf{s}_{1:l}) = \prod_{t=1}^{l} CN(\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\mathbf{h}_t, \sigma_w^2\mathbf{I}) \tag{10}$$

$$p(\mathbf{h}_{1:l}) = \prod_{t=1}^{l} CN(0, \sigma_h^2\mathbf{I}) \tag{11}$$

$$p(\mathbf{s}_{1:l}|\boldsymbol{\theta}) = \prod_{t=1}^{l}\prod_{k=1}^{N} (\theta_k)^{s_{t,k}}(1-\theta_k)^{1-s_{t,k}}. \tag{12}$$

We assumed that the channel gains $\mathbf{h}_{1:l}$ and user activity vectors $\mathbf{s}_{1:l}$ are temporally uncorrelated. Unfortunately, it is difficult to determine a tractable solution to (9) since the marginalization over the discrete-valued variable $\mathbf{s}_{1:l}$ requires 2^N configurations in total.

3.2. EM-based UAP estimation

The EM algorithm [35] seeks to determine the ML estimate of $\boldsymbol{\theta}$ in an iterative fashion by alternating the expectation step (E-step) and the maximization step (M-step) until convergence. The E-step creates a Q function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and the M-step computes parameters maximizing the expected log-likelihood found in the E-step.

3.2.1. E-Step

For the given measurements $\mathbf{y}_{1:l}$ and the estimate $\boldsymbol{\theta}^{(i-1)}$ obtained in the previous iteration, the E-step computes the function $\mathcal{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)})$ which is the log likelihood function of the complete data $\{\mathbf{y}_{1:l},\mathbf{h}_{1:l},\mathbf{s}_{1:l}\}$, i.e.,

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)}) = E\Big[\ln p(\mathbf{y}_{1:l},\mathbf{h}_{1:l},\mathbf{s}_{1:l};\boldsymbol{\theta})|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big], \tag{13}$$

$$\begin{aligned}&= \sum_{k=1}^{N}\sum_{t=1}^{l} E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big]\ln\theta_k \\ &\quad + \sum_{k=1}^{N}\left(l - \sum_{t=1}^{l} E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big]\right)\ln(1-\theta_k) + C,\end{aligned} \tag{14}$$

where C is the term independent of $\boldsymbol{\theta}$. The derivation of (14) is presented in Appendix B. The conditional expectation $E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big]$ is expressed as

$$\begin{aligned}E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big] &= 1\cdot p\Big(s_{t,k}=1|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big) \\ &\quad + 0\cdot p\Big(s_{t,k}=0|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)\end{aligned} \tag{15}$$

$$= p\Big(s_{t,k}=1|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big) \tag{16}$$

$$= \sum_{\{\mathbf{s}_t:s_{t,k}=1\}} p\Big(\mathbf{s}_t|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big) \tag{17}$$

$$= \sum_{\{\mathbf{s}_t:s_{t,k}=1\}} p\Big(\mathbf{s}_t|\mathbf{y}_t;\boldsymbol{\theta}^{(i-1)}\Big). \tag{18}$$

The expression of the distribution $p\big(\mathbf{s}_t|\mathbf{y}_t;\boldsymbol{\theta}\big)$ is provided in Appendix A. In typical MMTC scenarios, the number of potential users N is large so that the size of summation 2^{N-1} in (18) incurs infeasible computational complexity. In order to calculate $E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big]$ at low complexity, we use the importance sampling method [36], which calculates the conditional expectation using the samples generated by the proposal distribution, which is different from the posteriori distribution $p\Big(\mathbf{s}_t|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)$.

Suppose we have N_s sample vectors $\mathbf{s}_t^{(1)},\ldots,\mathbf{s}_t^{(Q)}$ drawn from the proposal distribution $q\Big(\mathbf{s}_t|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)$. Then, $E\Big[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big]$ can be approximated by

$$E[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}] \approx \sum_{j=1}^{N_s} w_t^{(j)} s_{t,k}^{(j)}, \tag{19}$$

where $s_{t,k}^{(j)}$ is the kth element of the jth sample $\mathbf{s}_t^{(j)}$ and the importance weight $w_t^{(j)}$ is given by

$$w_t^{(j)} = \frac{p\Big(\mathbf{s}_t^{(j)}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)}{q\Big(\mathbf{s}_t^{(j)}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)}. \tag{20}$$

The approximation error approaches zero as the number of samples N_s increases. It is difficult to generate the samples for the N elements of $\mathbf{s}_t$ from the *joint* posteriori distribution $p\Big(\mathbf{s}_t|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big)$ in (18). On the contrary, according to the prior distribution $p\Big(\mathbf{s}_t;\boldsymbol{\theta}^{(i-1)}\Big)$, the elements of $\mathbf{s}_t$ are independent and identically distributed. Thus, it is convenient to use the prior distribution as the proposal distribution, i.e.,

$$q\Big(\mathbf{s}_t|\mathbf{y}_{1:l},\boldsymbol{\theta}^{(i-1)}\Big) = p\Big(\mathbf{s}_t;\boldsymbol{\theta}^{(i-1)}\Big) \tag{21}$$

$$= \prod_{k=1}^{N}\big(\theta_k^{(i-1)}\big)^{s_{t,k}}\big(1-\theta_k^{(i-1)}\big)^{1-s_{t,k}}. \tag{22}$$

Owing to the statistical independence between the elements, we can easily generate samples according to $p\Big(\mathbf{s}_t;\boldsymbol{\theta}^{(i-1)}\Big)$. The importance weight $\tilde{w}_t^{(j)}$ without normalization can obtained from

$$\tilde{w}_t^{(j)} = \frac{p\Big(\mathbf{s}_t^{(j)}|\mathbf{y}_t;\boldsymbol{\theta}^{(i-1)}\Big)}{p\Big(\mathbf{s}_t^{(j)};\boldsymbol{\theta}^{(i-1)}\Big)} \tag{23}$$

$$= \frac{p\Big(\mathbf{y}_t|\mathbf{s}_t^{(j)};\boldsymbol{\theta}^{(i-1)}\Big)}{p\Big(\mathbf{y}_t;\boldsymbol{\theta}^{(i-1)}\Big)} \tag{24}$$

$$\propto \frac{|\sigma_w^2\Sigma_j^{-1}|}{(\pi\sigma_w^2)^M(\pi\sigma_h^2)^N}\exp \times\left(-\frac{1}{\sigma_w^2}\big(\|\mathbf{y}_t\|^2 - \mathbf{y}_t^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t^{(j)})\Sigma_j^{-1}\Lambda(\mathbf{s}_t^{(j)})^H\boldsymbol{\Phi}^H\mathbf{y}_t\big)\right) \tag{25}$$

$$\propto|\Sigma_j^{-1}|\exp\left(\frac{1}{\sigma_w^2}\mathbf{y}_t^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t^{(j)})\Sigma_j^{-1}\Lambda(\mathbf{s}_t^{(j)})^H\boldsymbol{\Phi}^H\mathbf{y}_t\right), \tag{26}$$

where $\Sigma_j = \Lambda(\mathbf{s}_t^{(j)})^H\boldsymbol{\Phi}^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t^{(j)}) + \big(\sigma_w^2/\sigma_h^2\big)\mathbf{I}$. The derivation of (25) is presented in Appendix C. The scaled weight $\tilde{w}_t^{(j)}$ in (26) is normalized to obtain the importance weight $w_t^{(j)}$

$$w_t^{(j)} = \left(\sum_{i=1}^{N_s}\tilde{w}_t^{(i)}\right)^{-1}\tilde{w}_t^{(j)}. \tag{27}$$

3.2.2. M-Step

In M-Step, we find $\boldsymbol{\theta}$ that maximizes $\mathcal{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)})$. From (14), the updated estimate $\boldsymbol{\theta}^{(i)}$ for the ith iteration is given by

$$\boldsymbol{\theta}^{(i)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)}), \tag{28}$$

$$= \frac{1}{l}\sum_{t=1}^{l} E\Big[\mathbf{s}_t|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\Big] \tag{29}$$

$$\approx \frac{1}{l}\sum_{t=1}^{l}\sum_{j=1}^{N_s} w_t^{(j)}\mathbf{s}_t^{(j)} \tag{30}$$

3.2.3. Algorithm summary

The procedure of the proposed UAP estimation algorithm is summarized in Algorithm 1.

Algorithm 1 Proposed UAP estimation algorithm.

Input : $\boldsymbol{\theta}_0$, $\{\mathbf{y}_1, \mathbf{y}_2 \ldots, \mathbf{y}_l\}$
Output : $\hat{\boldsymbol{\theta}}_{ML}$
Initialize $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}_0$.
for $i = 1$ to *Iter* **do**
  for $t = 1 : l$ **do**
    Generate the samples $\mathbf{s}_t^{(1)}, \ldots, \mathbf{s}_t^{(N_s)}$ according to $p(\mathbf{s}; \boldsymbol{\theta}^{(i-1)})$ in (22).
    Calculate the importance weights $\tilde{w}_t^{(1)}, \ldots, \tilde{w}_t^{(N_s)}$ according to (26).
    Normalize the importance weights : $w_t^{(j)} = \left[\sum_{i=1}^{N_s} \tilde{w}_t^{(i)}\right]^{-1} \tilde{w}_t^{(j)}$ for $i = 1, \cdots, N_s$.
    Calculate $E\left[\mathbf{s}_t | \mathbf{y}_{1:l}, \boldsymbol{\theta}^{(i-1)}\right] = \sum_{j=1}^{N_s} w_t^{(j)} \mathbf{s}_t^{(j)}$.
  end for
  Update $\boldsymbol{\theta}^{(i)} = \frac{1}{l} \sum_{t=1}^{l} E\left[\mathbf{s}_t | \mathbf{y}_{1:l}; \boldsymbol{\theta}^{(i-1)}\right]$.
end for
Obtain $\hat{\boldsymbol{\theta}}_{ML} = \boldsymbol{\theta}^{(Iter)}$

3.3. Practical implementation

In this subsection, we discuss the implementation of the UAP estimation algorithm in the practical MMTC scenarios.

3.3.1. Batch algorithm

One strategy to implement the proposed method is to calculate the estimate of UAP based on a block of L measurements collected periodically (called a measurement block). In the beginning, each element of the UAP $\boldsymbol{\theta}$ is initialized with a fixed value, e.g. 1/4. During the period of collecting the measurements, the AUD is performed using $\hat{\boldsymbol{\theta}}$ obtained from the previous measurement block. Once L measurements are collected, we update the UAP estimate by applying our UAP estimation algorithm described in the previous subsections. In EM iteration, $\boldsymbol{\theta}$ is initialized with the UAP estimate obtained from the previous measurement block. One disadvantage of this approach is that the entire sequence of measurements needs to be stored in the buffer because EM iterations need to be performed over the entire measurement block. Additionally, the size of the measurement block should be sufficiently large to analyze the user activity, resulting in considerable memory usage and the UAP estimate cannot be updated until the entire measurement block is received.

3.3.2. Online algorithm

In order to overcome the shortcomings of the batch algorithm, we devise the online algorithm. When processing large data sets, the online variants of the EM have been proposed to estimate the parameters of a latent data model without storing the data [37]. The online EM algorithm comprises two steps: a stochastic approximation version of the E-step incorporating the information brought by the newly available observation and the M-step maximizing the Q function as the M-step of the traditional EM algorithm. Consequently, our approach sequentially updates $\boldsymbol{\theta}$ only once for each new measurement. Denote $\boldsymbol{\theta}_t$ and Q_t as the parameter estimate and Q function updated at time step t, respectively. Then, the online EM algorithm can be described by

Stochastic E-STEP

$$Q_t = Q_{t-1} + \gamma_t \left(E\left[\ln p(\mathbf{y}_t, \mathbf{h}_t, \mathbf{s}_t; \boldsymbol{\theta}) | \mathbf{y}_t; \boldsymbol{\theta}_{t-1}\right] - Q_{t-1}\right), \tag{31}$$

$$= (1-\gamma_t) Q_{t-1} + \gamma_t E\left[\ln p(\mathbf{y}_t, \mathbf{h}_t, \mathbf{s}_t; \boldsymbol{\theta}) | \mathbf{y}_t; \boldsymbol{\theta}_{t-1}\right], \tag{32}$$

$$= (1-\gamma_t) Q_{t-1} + \gamma_t \left(\sum_{k=1}^{N} E\left[s_{t,k} | \mathbf{y}_t; \boldsymbol{\theta}_{t-1}\right] \ln \theta_{t-1,k} + \sum_{k=1}^{N} \left(1 - E\left[s_{t,k} | \mathbf{y}_t; \boldsymbol{\theta}_{t-1}\right]\right) \ln(1-\theta_{t-1,k}) \right) + C. \tag{33}$$

M-STEP

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}} Q_t \tag{34}$$

$$= (1-\gamma_t) \boldsymbol{\theta}_{t-1} + \gamma_t E\left[\mathbf{s}_t | \mathbf{y}_t; \boldsymbol{\theta}_{t-1}\right], \tag{35}$$

$$\approx (1-\gamma_t) \boldsymbol{\theta}_{t-1} + \gamma_t \left(\sum_{j=1}^{N_s} w_t^{(j)} \mathbf{s}_t^{(j)} \right). \tag{36}$$

Note that γ_t is the step size that scales the contribution of the present input in comparison with the past information. A higher γ leads to faster convergence of the parameter estimate at the expense of a more noisy estimate. Under some mild regularity conditions, the convergence of the online-EM algorithm is guaranteed [37].

4. UAP-aware active user detection

In this section, we propose an AUD algorithm that can adopt the UAPs as prior information for enhancing the massive machine type communications. The UAPs estimated by our algorithm are incorporated into two well-known CS-based AUD methods OMP and BPDN, yielding UAP-aware OMP and UAP-aware BPDN for joint AUD and channel estimation.

4.1. UAP-aware OMP

OMP is a greedy algorithm that identifies the strongest elements of the support set one at a time in an iterative fashion. In each iteration, OMP calculates the absolute inner product of the residual signal $\mathbf{r}_t$ with the columns of $\boldsymbol{\Phi}$. It selects the column from $\boldsymbol{\Phi}$ with the largest absolute inner product and adds it to the set Γ containing the selected columns. Then, the residual signal is calculated by projecting the measurement vector $\mathbf{y}_t$ on the orthogonal complement of the subspace spanned by the selected columns. We can show that the metric of absolute inner product used in the standard OMP algorithm can be derived from the posteriori probability of $\mathbf{s}_t$

$$\ln p(\mathbf{s}_t | \mathbf{y}_t) = \ln p(\mathbf{y}_t | \mathbf{s}_t) + \ln p(\mathbf{s}_t) \tag{37}$$

$$= \ln \frac{|\sigma_w^2 \Sigma^{-1}|}{(\pi \sigma_w^2)^M (\pi \sigma_h^2)^N} - \frac{1}{\sigma_w^2} \left(\|\mathbf{y}_t\|^2 - \mathbf{y}_t^H \boldsymbol{\Phi} \Lambda(\mathbf{s}_t) \Sigma^{-1} \Lambda(\mathbf{s}_t)^H \boldsymbol{\Phi}^H \mathbf{y}_t \right) + \sum_{k=1}^{N} s_{t,k} \ln(\theta_k) + (1 - s_{t,k}) \ln(1-\theta_k), \tag{38}$$

where $\Sigma = \Lambda(\mathbf{s}_t)^H \boldsymbol{\Phi}^H \boldsymbol{\Phi} \Lambda(\mathbf{s}_t) + \left(\sigma_w^2/\sigma_h^2\right)\mathbf{I}$. We observe that the absolute inner product metric can be obtained by plugging the N candidate vectors, $c_1 = [1, 0, \ldots, 0], c_2 = [0, 1, \ldots, 0], \ldots, c_N = [0, 0, \ldots, 1]$ in place of $\mathbf{s}_t$ in (38). Assuming that all columns of $\boldsymbol{\Phi}$ have the magnitude one, i.e., $\|\phi_k\|_2^2 = 1$ and with equal UAPs for

all uers, we can show that $\ln p(\mathbf{s}_t = c_j|\mathbf{y}_t) \propto \left|\mathbf{y}_t^H \mathbf{\Phi}\Lambda(c_j)\right|^2$, which equals the absolute inner product used in the standard OMP. In subsequent iterations, the absolute inner product metric $\psi(j)$ is obtained by replacing $\mathbf{y}_t$ with the residual signal $\mathbf{r}_t$

$$\psi(j) = \left|\mathbf{r}_t^H \mathbf{\Phi}\Lambda(c_j)\right|^2, \tag{39}$$

where $j \in [1, \ldots, N] \cup \Gamma^c$.

Similarly, when the UAP $\boldsymbol{\theta}$ is available, the new metric $\psi'(j)$ for the proposed AUD method can be obtained by plugging $c_1, \ldots, c_N$ in place of $\mathbf{s}_t$ in (38)

$$\psi'(j) = \frac{1}{\sigma_w^2}\left|\mathbf{r}_t^H \mathbf{\Phi}\Lambda(c_j)\right|^2 + \ln(\theta_j). \tag{40}$$

Except for the different metrics $\psi(j)$ and $\psi'(j)$ being used, the rest of the procedure is the same for the standard OMP and the proposed UAP-aware OMP.

In the proposed UAP-aware OMP, the sparsity level K can be determined by applying the stopping rule during the iterations. We can choose one of the well-known stopping rules including energy-based thresholding [16], cross-validation method [38] and rank estimation method [39]. Our method terminates the iteration if the following criterion is met:

$$\|\mathbf{r}_u\|_2 < \sqrt{M\sigma_w^2}. \tag{41}$$

The summary of the proposed UAP-aware OMP algorithm is presented in Algorithm 2.

Algorithm 2 UAP-aware OMP algorithm.

Input : $\hat{boldsymbol\theta}$ and $\mathbf{y}_t$
Output : $\hat{\mathbf{h}}_t$ and $\hat{\mathbf{s}}_t$

1: Initialize $\Gamma = \varnothing$ and $\mathbf{r}_t = \mathbf{y}_t$.
2: **while do**
3: $\gamma = \underset{j\in[1,\ldots,N]\cup\Gamma^c}{\arg\max}\ \psi'(j)$.
4: $\Gamma = \Gamma \cup \gamma$.
5: Update the residual signal using $\mathbf{r}_t = \mathbf{y}_t - (boldsymbol\Phi_\Lambda)^\dagger \mathbf{y}_t$.
6: Exit the loop if the stopping criterion is satisfied.
7: **end while**

4.2. UAP-Aware BPDN

Let $\mathbf{u}_t = \Lambda(\mathbf{s}_t)\mathbf{h}_t$, then $\mathbf{u}_t$ is also a sparse vector, and thus we can find the estimate of $\mathbf{u}_t$ using the standard BPDN

$$\hat{\mathbf{u}}_t = \arg\min_{\mathbf{u}_t} \|\mathbf{y}_t - \mathbf{\Phi}\mathbf{u}_t\|_2^2 + \lambda\|\mathbf{u}_t\|_1, \tag{42}$$

where $\|\mathbf{u}_t\|_1$ is the ℓ_1-norm regularization term promoting the sparsity of the solution and λ is the regularization parameter. In [36], it was shown that the UAP $\boldsymbol{\theta}$ can be incorporated into the BPDN as a prior information as

$$\hat{\mathbf{u}}_t = \arg\min_{\mathbf{u}_t} \|\mathbf{y}_t - \mathbf{\Phi}\mathbf{u}_t\|_2^2 + \lambda\sum_{k=1}^{N}(-\ln\theta_k)|u_{t,k}|, \tag{43}$$

where $\mathbf{u}_t = [u_{t,1}, \ldots, u_{t,N}]^T$. The solution to (43) can be found using the convex optimization tool [40].

5. Simulations

In this section, we evaluate the performance of the proposed AUD method under the IoT MMTC scenarios.

Table 1
UAPs assigned for each user group.

Group	Number of Users	Distribution
1	25	$U(0.02, 0.05)$
2	8	$U(0.05, 0.1)$
3	4	$U(0.1, 0.4)$
4	3	$U(0.4, 0.7)$

5.1. Simulation setup

We consider the MMTC scenario where there exist $N = 40$ users in total and the length of a user identification code is set to $M = 20$. The user identification codes are generated using Gaussian random number generator. We assume block fading channels where the channel gain is be fixed within each packet frame but randomly changes across the frames. Each packet consists of single pilot symbol and ten data symbols. The data symbols are modulated using quadrature phase shift keying (QPSK). The number of samples generated for the importance sampling is set to 500. The EM-iteration terminates when the power of the UAP difference is less than 10^{-4} between the two successive iterations or 10 iterations are performed in total. The UAP estimate is initialized to $\theta_k^{(0)} = 0.5$ for all users in the EM-iteration. For online algorithm, the step size $\gamma_t = t^{-0.9}$ is used, as suggested in [37]. We suppose that $N = 40$ users are divided into four groups that exhibit different packet transmission behaviors. The first group of 25 users send packets very occasionally so that they have relatively low UAPs uniformly distributed between 0.02 and 0.05. The next group of eight users has the slightly higher UAPs between 0.05 and 0.1 and another group of four users have even higher UAPs between 0.1 and 0.4. The final group of three users transmit packets very frequently with the UAPs between 0.4 and 0.7. The UAPs assigned for all four groups are summarized in Table 1.

We evaluate the accuracy of the UAP estimation using the normalized mean square error (NMSE) metric defined by

$$\mathrm{NMSE}(\boldsymbol{\theta}) = 10\log_{10}\left(\frac{E[||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}||^2]}{E[||\boldsymbol{\theta}||^2]}\right). \tag{44}$$

The performance of the AUD algorithms can be measured using the active user recovery rate and the net symbol error rate (NSER). The active user recovery rate implies the ratio of the number of packets over which the algorithm correctly detects all active users to the total number of packets transmitted. The NSER is measured by counting the errors occurring in decoding the data symbols for all users detected by the AUD algorithm. Since the channel estimate is also obtained as a byproduct, we also evaluate the channel estimation NMSE. The signal to noise ratio (SNR) is defined by

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sigma_h^2}{\sigma_w^2}\right). \tag{45}$$

5.2. Simulation results

In this section, we provide the simulation results. As a reference, we compare the proposed AUD methods with two baseline algorithms, the OMP and BPDN. In addition, we compare our methods with the ideal UAP-aware OMP and the ideal UAP-aware BPDN using the true UAP values. Note that these algorithms provide the performance bound that can be achieved by the proposed AUD methods, the UAP-aware OMP and the UAP-aware BPDN. Moreover, we have evaluated the existing algorithms [12,27] while utilizing the estimated UAP to reflect the utility of the proposed UAP estimation algorithm.

Fig. 2 provides a look at the performance of the online algorithm as compared to the batch algorithm. The SNR is set to

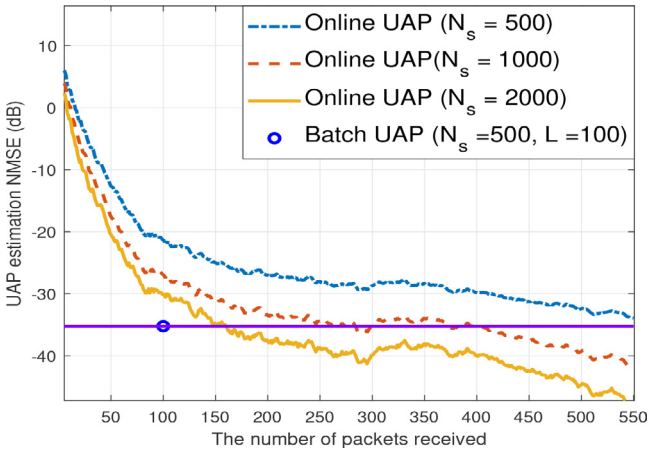

Fig. 2. Plot of comparison between online and batch variants of UAP estimation algorithms.

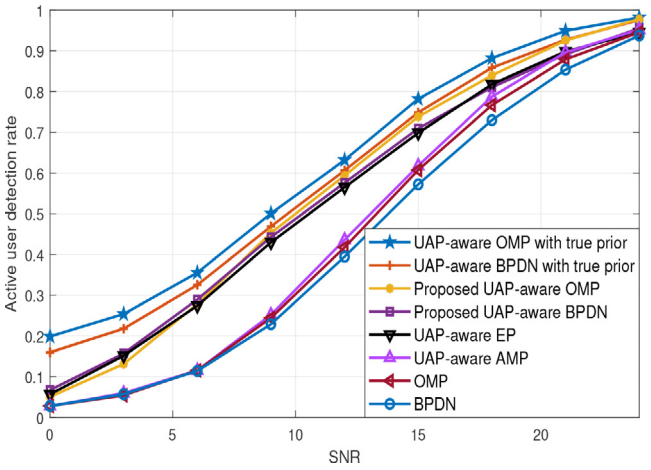

Fig. 4. Plot of active user recovery rate versus SNR for several AUD algorithms.

15 dB. It can be observed that the online version performs close to the batch version when large number of packets L are available for a fixed $N_s = 500$. However, if N_s is increased, it can be observed that the online version surpasses the batch version for a fairly reduced number of received packets L. Specifically, it can be observed that same performance can be observed for online versions when $L = 150, L = 250$, and $L = 550$ for $N_s = 2000, N_s = 1000$, and $N_s = 500$ respectively. It is also observed that for batch algorithm an estimate θ is obtained after L measurements, however for the online scenario, an estimate for θ is obtained immediately after each measurement is made available.

The computational complexity evaluation provides further insight about the performance comparison of the proposed UAP estimation algorithms. The computational complexity of the batch and online UAP estimation algorithms is of the order of $\mathcal{O}(IterLN_sMN)$ and $\mathcal{O}(LN_sMN)$ respectively for inferring an estimate of the UAP θ after L measurements.

Fig. 3 a provides insight into computational complexity for a fixed number of samples N_s while the measurement block size L is varied. As the batch UAP estimation algorithm takes multiple iterations before converging, the number of flop count required to infer θ increases much faster as compared to the online UAP approach. Fig. 3b analyze the computational complexity as the number of samples N_s are varied while the measurement block size is set to $L = 100$. It can be observed that even at significantly higher $N_s = 2000$, the flop count for the online UAP is much less as compared to the batch algorithm. In conclusion, the online UAP estimation algorithm provides compare able estimation accuracy at a fairly less computational complexity as compared to its counter part even if sufficiently large number of samples N_s or packets L are utilized for estimating the UAP.

Fig. 4 provides the plot of the active user recovery rate as a function of SNR. The batch algorithm is used for the proposed methods. We set the number of the packets used for UAP estimation to $L = 100$. We observe from Fig. 4 that the proposed UAP-aware AUD methods outperform the baselines by a significant margin. At the 15 dB SNR, the improvement of the active user recovery rate achieved by the proposed method over the baselines is over 10%. We also observe that the proposed AUD methods achieve the performance close to the ideal algorithms using the true UAP values. This shows that the proposed UAP estimator provides accurate UAP estimates so that the effect of the UAP information is kept large for the MMTC scenario considered. As the SNR increases, the quality of measurement improves and the role of prior information diminishes, reducing the performance gain over the baseline algorithms. Note that the largest performance gap is achieved at around 10dB SNR.

Fig. 5 shows the channel estimation NMSE for the active users detected by the proposed AUD methods. Similarly, the proposed methods achieve large performance gain over the baselines and the performance of our schemes is close to the performance bound achieved with the perfect knowledge of the UAP. Specifically, the proposed UAP-aware OMP and UAP-aware BPDN offer around 2dB gain over the baseline OMP and BPDN, respectively.

Next, we evaluate the NSER achieved by the active users as a function of SNR. We use the channel estimate obtained by our AUD

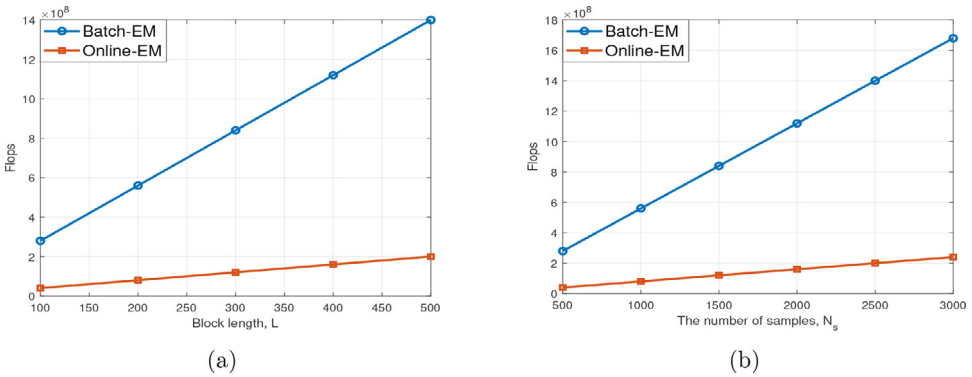

Fig. 3. Plot of computation complexity comparison of the proposed batch and online UAP estimation algorithms in terms of flops per iteration: (a) Flops versus block length L for $N_s = 500$, (b) Flops versus the number of samples N_s when $L = 100$.

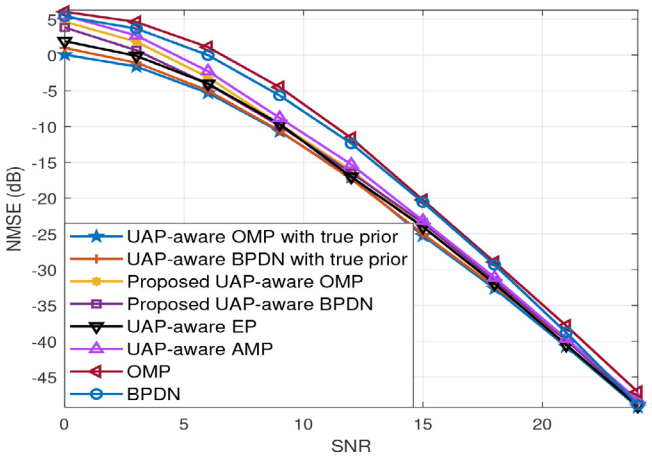

Fig. 5. Plot of channel NMSE versus SNR for several AUD algorithms.

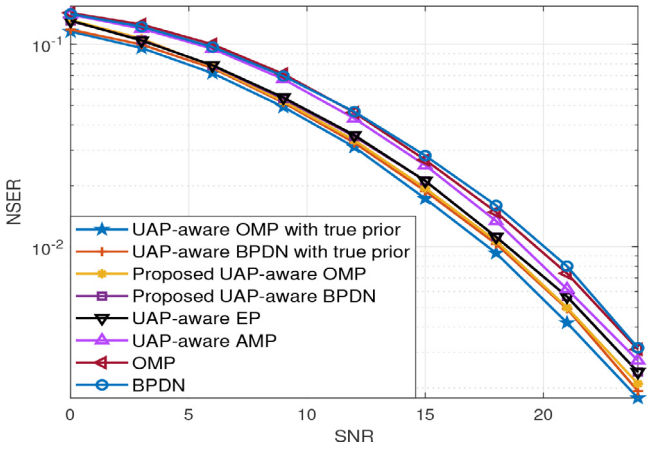

Fig. 6. Plot of NSER versus SNR for several AUD algorithms.

algorithm to demodulate the data symbols. Fig. 6 shows that the performance gain of the proposed methods is maintained in terms of the data detection performance. We observe that the proposed algorithms also achieve more than 2 dB gain over the baseline algorithms.

Next, we investigate how the performance of the proposed algorithm behaves as a function of the block length L, the number of the EM iterations *Iter*, and the number of samples N_s drawn for importance sampling. For all the cases, the SNR is set to 15 dB. Fig. 7a shows that the UAP estimation NMSE decreases with L. This is because when a long period of user activities are observed, better statistics can be obtained for UAP and the effect of noise averaging would be larger. Fig. 7b shows the convergence behaviour of EM iteration with $L = 100$. We observe that the EM algorithm converges faster in higher SNR range. We observe that for most cases, the EM algorithm used in our method converges within 10 iterations. Fig. 7c show the performance behavior as a function of the number of particles. We observe that the approximation error caused by the importance sampling decreases with the number of the particles but performance improvement diminishes over $N_s = 500$. Note that $N_s = 500$ was used for our simulations.

6. Conclusions

In this paper, we proposed the enhanced AUD algorithm for MMTC systems, which exploits the prior user activity information in identifying the active users. Based on the observation that each user can exhibit different packet transmission patterns, we designed the algorithm which can infer each user's probability of packet transmission based on the sequence of the data packets received over certain period of time. The proposed method efficiently calculated the estimate of UAP using the EM algorithm and its online variant was also presented for the efficient use of memory. We used the estimated UAP for two conventional AUD detectors OMP and BPDN as prior information and produced the new AUD

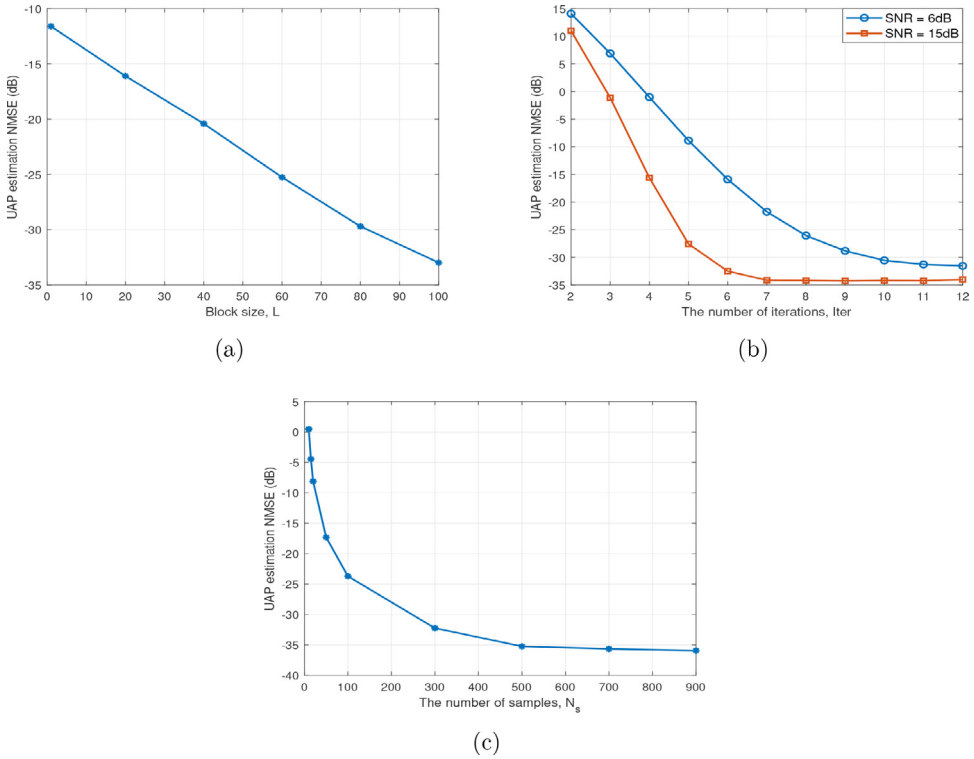

Fig. 7. Behavior of the proposed UAP-estimation algorithm : (a) UAP NMSE versus block length L, (b) UAP NMSE versus the number of EM iterations *Iter* (c) UAP NMSE versus the number of particles N_s for approxiating the desired density.

algorithms: UAP-aware OMP and UAP-ware BPDN. Our simulation results showed that the use of the UAP information can offer the significant performance gain over the baseline AUD detectors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Syed Ali Irtaza: Methodology, Software, Formal analysis, Writing – original draft. **Salma Riaz:** Validation, Writing – original draft, Writing – review & editing. **Ali Nauman:** Project administration, Writing – review & editing. **Muhammad Ali Jamshed:** Project administration, Writing – review & editing. **Sung Won Kim:** Validation.

Appendix A. Expression of posterior distribution in (18)

The posteriori distribution in (18) is expressed as

$$p\left(\mathbf{s}_t|\mathbf{y}_t;\boldsymbol{\theta}\right)=\frac{p(\mathbf{y}_t|\mathbf{s}_t)p\left(\mathbf{s}_t;\boldsymbol{\theta}\right)}{\sum_{\mathbf{s}_t}p(\mathbf{y}_t|\mathbf{s}_t)p\left(\mathbf{s}_t;\boldsymbol{\theta}\right)}, \tag{A.1}$$

where

$$p(\mathbf{y}_t|\mathbf{s}_t)=\frac{|\sigma_w^2\Sigma^{-1}|}{(\pi\sigma_w^2)^M(\pi\sigma_h^2)^N}\exp \times\left(-\frac{1}{\sigma_w^2}\left(\|\mathbf{y}_t\|^2-\mathbf{y}_t^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\Sigma^{-1}\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\mathbf{y}_t\right)\right), \tag{A.2}$$

$$p\left(\mathbf{s}_t;\boldsymbol{\theta}\right)=\prod_{k=1}^{N}(\theta_k)^{s_{t,k}}(1-\theta_k)^{1-s_{t,k}}. \tag{A.3}$$

Appendix B. Derivation of (14)

We can show that

$$\mathcal{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)})=E\left[\ln p(\mathbf{y}_{1:l},\mathbf{h}_{1:l},\mathbf{s}_{1:l};\boldsymbol{\theta})\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right], \tag{B.1}$$

$$=E\left[\ln p(\mathbf{y}_{1:l}|\mathbf{h}_{1:l},\mathbf{s}_{1:l})p(\mathbf{h}_{1:l})p(\mathbf{s}_{1:l};\boldsymbol{\theta})\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right], \tag{B.2}$$

$$=E\left[\ln p(\mathbf{s}_{1:l};\boldsymbol{\theta})+C\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right], \tag{B.3}$$

where C is the term independent of $\boldsymbol{\theta}$. Then,

$$\mathcal{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(i-1)})=E\left[\ln\prod_{t=1}^{l}p(\mathbf{s}_t;\boldsymbol{\theta})\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right]+C', \tag{B.4}$$

$$=E\left[\ln\prod_{t=1}^{l}\prod_{k=1}^{N}\theta_k^{s_{t,k}}(1-\theta_k)^{1-s_{t,k}}\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right]+C', \tag{B.5}$$

$$=E\left[\sum_{k=1}^{N}\sum_{t=1}^{l}s_{t,k}\ln\theta_k+\sum_{k=1}^{N}\left[l-\sum_{t=1}^{l}s_{t,k}\right]\ln(1-\theta_k)\middle|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right]+C', \tag{B.6}$$

$$=\sum_{k=1}^{N}\sum_{t=1}^{l}E\left[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right]\ln\theta_k +\sum_{k=1}^{N}\left(L-\sum_{t=1}^{l}E\left[s_{t,k}|\mathbf{y}_{1:l};\boldsymbol{\theta}^{(i-1)}\right]\right)\ln(1-\theta_k)+C'. \tag{B.7}$$

Appendix C. Derivation of (25)

The distribution $p(\mathbf{y}_t|\mathbf{s}_t)$ can be obtained from

$$p(\mathbf{y}_t|\mathbf{s}_t)=\int_{\mathbf{h}_t}p(\mathbf{y}_t|\mathbf{h}_t,\mathbf{s}_t)p(\mathbf{h}_t)d\mathbf{h}_t, \tag{C.1}$$

where $p(\mathbf{y}_t|\mathbf{h}_t,\mathbf{s}_t)\sim\mathcal{CN}(\boldsymbol{\Phi}\mathbf{S}_t\mathbf{h}_t,\sigma_w^2\mathbf{I})$ and $p(\mathbf{h}_t)\sim\mathcal{CN}(\mathbf{0},\sigma_h^2\mathbf{I})$. Then, we can easily show that

$$\int_{\mathbf{h}_t}p(\mathbf{y}_t|\mathbf{h}_t,\mathbf{s}_t)p(\mathbf{h}_t)d\mathbf{h}_t =\int_{\mathbf{h}_t}\frac{1}{(\pi\sigma_w^2)^M}\exp\left(-\frac{1}{\sigma_w^2}\|\mathbf{y}_t-\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\mathbf{h}_t\|^2\right)\frac{1}{(\pi\sigma_h^2)^N}\exp \times\left(-\frac{1}{\sigma_h^2}\|\mathbf{h}_t\|^2\right)d\mathbf{h}_t, \tag{C.2}$$

$$=\frac{1}{(\pi\sigma_w^2)^M(\pi\sigma_h^2)^N}\int_{\mathbf{h}_t}\exp \times\left(-\frac{1}{\sigma_w^2}\left(\mathbf{h}_t-\Sigma^{-1}\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\mathbf{y}_t\right)^H\Sigma\left(\mathbf{h}_t-\Sigma^{-1}\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\mathbf{y}_t\right)\right) \exp\left(-\frac{1}{\sigma_w^2}\left(\|\mathbf{y}_t\|^2-\mathbf{y}_t^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\Sigma^{-1}\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\mathbf{y}_t\right)\right)d\mathbf{h}_t, \tag{C.3}$$

$$=\frac{|\sigma_w^2\Sigma^{-1}|}{(\pi\sigma_w^2)^M(\pi\sigma_h^2)^N}\exp \times\left(-\frac{1}{\sigma_w^2}\left(\|\mathbf{y}_t\|^2-\mathbf{y}_t^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)\Sigma^{-1}\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\mathbf{y}_t\right)\right), \tag{C.4}$$

where $\Sigma=\Lambda(\mathbf{s}_t)^H\boldsymbol{\Phi}^H\boldsymbol{\Phi}\Lambda(\mathbf{s}_t)+\left(\frac{\sigma_w^2}{\sigma_h^2}\right)\mathbf{I}$.

References

[1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, A. Dekorsy, Massive machine-type communications in 5G: physical and MAC-layer solutions, IEEE Commun. Mag. 54 (9) (2016) 59–65.

[2] L. Liu, E.G. Larsson, W. Yu, P. Popovski, C. Stefanovic, E. Carvalho, Sparse signal processing for grant-free massive connectivity: a future paradigm for random access protocols in the internet of things, IEEE Signal Proc. Mag. 35 (5) (2018) 88–99.

[3] E.J. Candes, Compressive sampling, Proc. Int. Cong. Math. (2006) 1433–1452.

[4] J.W. Choi, B. Shim, Y. Ding, B. Rao, D.I. Kim, Compressed sensing for wireless communications: useful tips and tricks,, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1527–1550.

[5] H. Sasahara, K. Hayashi, M. Nagahara, Multiuser detection based on MAP estimation with sum-of-absolute-values relaxation, IEEE Trans. Sig. Proc. 65 (21) (2017) 5621–5634.

[6] H.F. Schepker, C. Bockelmann, A. Dekorsy, Efficient detection for joint compressed sensing detection and channel decoding, IEEE Trans. Commun. 63 (6) (2015) 2249–2260.

[7] B.C. Wang, L.L. Dai, Y.F. Yuan, Z.C. Wang, Compressive sensing based multi-user detection for uplink grant-free non-orthogonal multiple access, in: Proc. IEEE VTC, 2015, pp. 1–5.

[8] H. Zhu, G.B. Giannakis, Exploiting sparse user activity in multiuser detection, IEEE Trans. Commun. 59 (2) (2011) 454–465.

[9] Z. Zhang, Y. Li, C. Huang, Q. Guo, L. Liu, C. Yuen, Y.L. Guan, User activity detection and channel estimation for grant-free random access in LEO satellite-enabled internet of things, IEEE Inter. Thin. J. 7 (9) (2020) 8811–8825.

[10] G. Hannak, M. Mayer, A. Jung, G. Matz, N. Goertz, Joint channel estimation and activity detection for multiuser communication systems, in: IEEE ICCW, 2015, pp. 2086–2091.

[11] S. Park, H. Seo, H. Ji, B. Shim, Joint active user detection and channel estimation for massive machine-type communications, in: Proc. IEEE SPAWC, 2017, pp. 1–5.
[12] J. Ahn, B. Shim, K.B. Lee, EP-based joint active user detection and channel estimation for massive machine-type communications, IEEE Trans. Commun. 67 (7) (2019) 5178–5189.
[13] Z. Sun, Z. Wei, L. Yang, J. Yuan, X. Cheng, L. Wan, Exploiting transmission control for joint user identification and channel estimation in massive connectivity, IEEE Trans. Commun. 67 (9) (2019) 6311–6326.
[14] C. Wei, H. Liu, Z. Zhang, J. Dang, L. Wu, Approximate message passing-based joint user activity and data detection for NOMA, IEEE Commun. Lett. 21 (3) (2016) 640–643.
[15] R.B.D. Renna, R.C.D. Lamare, Adaptive activity-aware iterative detection for massive machine-type communication, IEEE Wire. Commun. Lett. 8 (6) (2019) 1631–1634.
[16] Y. Du, B. Dong, W. Zhu, P. Gao, Z. Chen, X. Wang, J. Fang, Joint channel estimation and multiuser detection for uplink grant-free NOMA, IEEE Wirel. Commun. Lett. 7 (4) (2018) 682–685.
[17] S.A. Irtaza, S.H. Lim, J.W. Choi, Greedy data-aided active user detection for massive machine type communications, IEEE Wirel. Commun. Lett. 8 (4) (2019) 1224–1227.
[18] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit„ SIAM J. Sci. Comput. 20 (1) (1998) 33–61.
[19] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inform. Theory 53 (12) (2007) 4655–4666.
[20] D. Needell, J.A. Tropp, CosaMP: iterative signal recovery from incomplete and inaccurate samples„ Appl. Comput. Harmon. Anal. 26 (2009) 301–321.
[21] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, IEEE Trans. Inf. Theory 55 (5) (2009) 2230–2249.
[22] M. Bayati, A. Montanari, The dynamics of message passing on dense graphs, with applications to compressed sensing, IEEE Trans. Inf. Theory 25 (2) (2015) 753–822.
[23] C. Wei, H. Liu, Z. Zhang, J. Dang, L. Wu, Approximate message passing-based joint user activity and data detection for NOMA, IEEE Commun. Letter 21 (3) (2017) 640–643.
[24] B. Wang, L. Dai, T. Mir, Z. Wang, Joint user activity and data detection based on structured compressive sensing for NOMA, IEEE Commun. Lett. 20 (7) (2016) 1473–1476.
[25] Z. Gao, L. Dai, Z. Wang, S. Chen, L. Hanzo, Compressive-sensing-based multiuser detector for the large-scale SM-MIMO uplink, IEEE Trans. Veh. Technol., 65 (10) (2016) 8725–8730.
[26] B.K. Jeong, B. Shim, K.B. Lee, MAP-based active user and data detection for massive machine-type communications, IEEE Trans. Veh. Tech. 67 (7) (2018) 8481–8494.
[27] Z. Chen, F. Sohrabi, W. Yu, Sparse activity detection for massive connectivity, IEEE Trans. Sig. Process. 66 (11) (2018) 1890–1904.
[28] L. Liu, W. Yu, Massive connectivity with massive MIMO - Part I: device activity detection and channel estimation, IEEE Trans. Sig. Process. 66 (11) (2018) 2933–2946.
[29] B. Wang, L. Dai, Y. Zhang, T. Mir, J. Li, Dynamic compressive sensing-based multi-user detection for uplink grant-free NOMA, IEEE Commun. Lett. 20 (11) (2016) 2320–2323.
[30] Y. Du, B. Dong, Z. Chen, X. Wang, Z. Liu, P. Gao, S. Li, Efficient multi-user detection for uplink grant-free NOMA: prior-information aided adaptive compressive sensing perspective, IEEE J. Sel. Are. Comm. 35 (12) (2017) 2812–2828.
[31] S. Jiang, X. Yuan, X. Wang, C. Xu, W. Yu, Joint user identification, channel estimation, and signal detection for grant-free NOMA„ IEEE Trans. Wire. Commun. 19 (10) (2020) 6960–6976.
[32] Z. Zhang, Y. Li, C. Huang, Q. Guo, C. Yuen, Y.L. Guan, DNN-aided block sparse Bayesian learning for user activity detection and channel estimation in grant-free non-orthogonal random access, IEEE Trans. Vehi. Tech. 68 (12) (2019) 12000–12012.
[33] J.C. Jiang, H.M. Wang, H.V. Poor, Performance analysis of joint active user detection and channel estimation for massive connectivity„ IEEE Trans. Sig. Proc. 70 (2022) 3647–3662.
[34] J. Scarlet, J.S. Evans, S. Dey, Compressed sensing with prior information: information-theoretic limits and practical decoders, IEEE Trans. Sig. Process. 61 (2) (2013) 427–439.
[35] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Statist. Soc. Ser. B (Methodolog.) 39 (1) (1977) 1–38.
[36] A. Doucet, J.F.G. de Freitas, N.J. Gordon, An Introduction to Sequential Monte Carlo Methods, Springer-Verlag, New York, 2001.
[37] O. Cappe, E. Moulines, On-line expectation maximization for latent data models, J. R. Statist. Soc. Ser. B (Methodolog.) 71 (3) (2009) 593–613.
[38] R. Ward, Compressed sensing with cross validation„ IEEE Trans. Inf. Theory 55 (12) (2009) 5773–5782.
[39] M.E. Lopes, Estimating unknown sparsity in compressed sensing, in: Proc. 30th Int. Conf. Mach. Learn., 2013, pp. 217–225.
[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011) 1–122.