A compressive sensing-based active user and symbol detection for massive machine-type communications

April 18, 2018

Byeong Kook Jeong

Mobile Communication Lab.

Seoul National University, Korea

0. Outline

- System model
- Conventional approach
- Proposed approach
- Simulation result
- Conclusion & further work

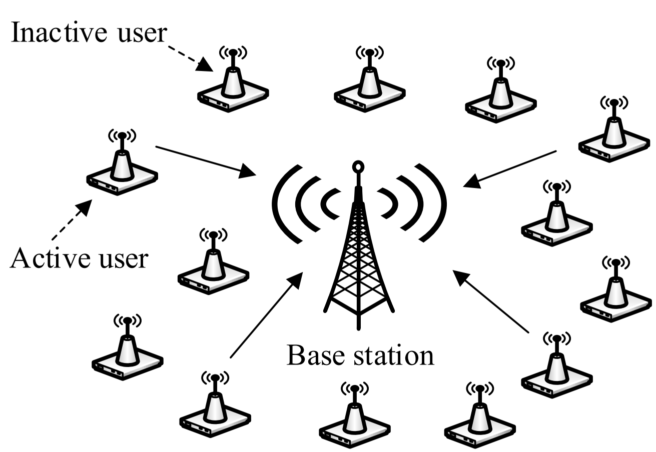
1. System Model (1/3)

❖ Massive machine-type communication (mMTC)

- Sporadic uplink transmission of short packets with low data rates
- Only small number of machine-type devices (i.e., users) are active at a time.
  - Activity probability of a user n is p_n and $p_n \ll 1$.

❖ Goal

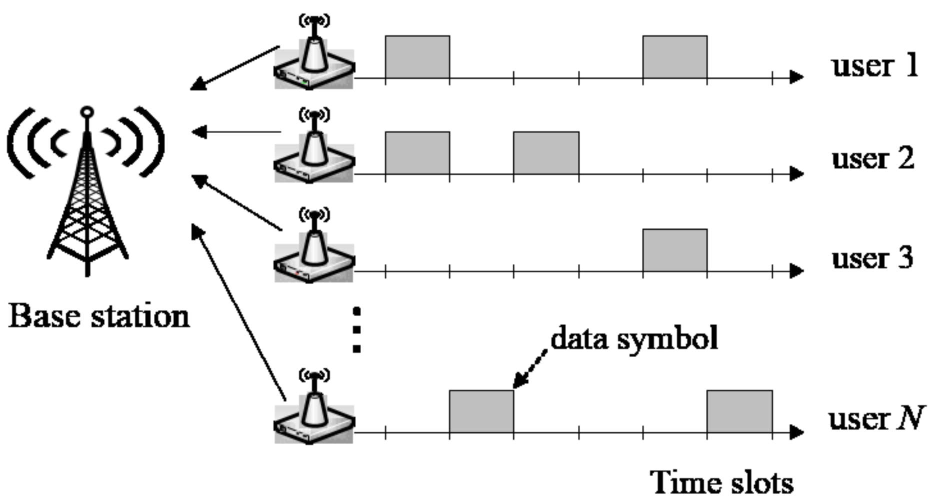
- To simultaneously detect the active user and its data (i.e., symbol).



1. System Model (2/3)

❖ Assumption

- Synchronous transmission
- Small inter-symbol-interference (ISI): Symbol length ≫ Multipath profile
  - Each symbol is spread by a user-specific spreading sequence.
- The base station (BS) has the perfect knowledge of wireless channels between users and the BS.
- User activity probability (p_n) is known to the BS.
  - Considering use-cases (e.g., health care, factory automation, environment sensing), the mMTC data is generally periodic so that p_n can be estimated by statistics.



1. System Model (3/3)

❖ System model

- $\mathbf{y} = \sum_{n=1}^{N}(\mathbf{h}_n * \mathbf{s}_n)x_n + \mathbf{v} = \mathbf{A}\mathbf{x} + \mathbf{v}$
  - ➢ The received vector $\mathbf{y} \in \boldsymbol{\mathcal{C}}^M$, the channel impulse response (CIR) between the user n and the BS $\mathbf{h}_n \in \mathbb{C}^\tau$, and the user-specific spreading sequence of the user n $\mathbf{s}_n \in \mathbb{C}^M$.
  - ➢ N: number of all active and inactive users, M: spreading factor
    - $N \gg M$, i.e., the system is underdetermined.
- The (known) channel matrix $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \dots \mathbf{a}_N] \in \mathbb{C}^{M \times N}$ where $\mathbf{a}_n = \mathbf{h}_n * \mathbf{s}_n \in \mathbb{C}^M$
  - ➢ Ignore the tail of the convolution based on the negligible ISI assumption (i.e., $\tau \ll M$).
- The data of a user n, i.e., $x_n \in \boldsymbol{\mathcal{A}}$ (finite alphabet, e.g., BPSK, QPSK, and QAM)

❖ Sparse signal recovery problem

- This is because $\mathbf{x}$ is a sparse vector, meaning that most entries are 0.
- Compressive sensing-based multi-user detection (CS-MUD) shows the better performance than conventional MUD (e.g., linear MMSE).

2. Conventional algorithm

❖ Convex optimization-based approach

- $\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0$ where λ is a sparsity-promoting term.
- After LASSO relaxation, $\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ (convex function)
- It can be solved by linear programming methods.
  - Powerful but computationally complex

❖ Greedy approach

- Iterative algorithm
  - Identification → augmentation → projection/detection → cancellation
- Identification: $n^* = \arg\max_{n} |\mathbf{a}_n^H \mathbf{r}|$ where initially $\mathbf{r} = \mathbf{y}$.
- Augmentation: S (support, i.e., index set of nonzero entries) $\leftarrow S \cup \{n^*\}$
- Projection/detection: $\hat{\mathbf{x}}_S = \mathbf{A}_S^\dagger \mathbf{y}$
- Cancellation: $\mathbf{r} = \mathbf{y} - \mathbf{A}_S \hat{\mathbf{x}}_S$
- It is simple and effective so that it has been popularly used to CS-MUD.

3. Proposed algorithm (1/5)

❖ Modified greedy algorithm

- Exploits *a priori* information on p_n and a finite alphabet ($\mathcal{A}$) constraint.
  - C.f. a sparsity-aware sphere detection (SA-SD): complex combinatorial list search
- Maximum *a posteriori* probability (MAP)-based active user and symbol detection

❖ Preliminary

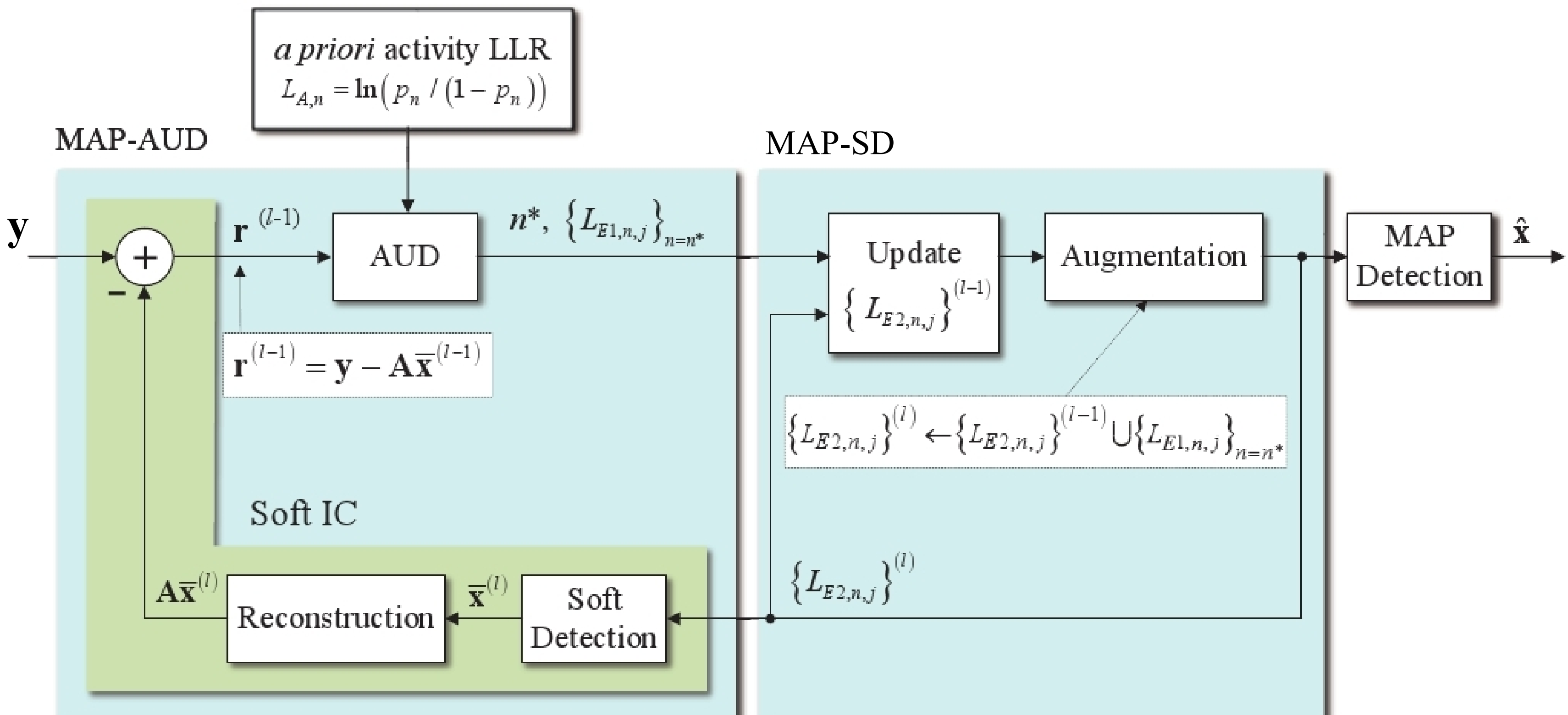
- ***A posteriori*** user activity log-likelihood ratio (LLR)
  - $L_n(\mathbf{y}) = \ln \frac{P(x_n \in \mathcal{A} \mid \mathbf{y})}{P(x_n = 0 \mid \mathbf{y})} = \ln \frac{P(\mathbf{y} \mid x_n \in \mathcal{A})}{P(\mathbf{y} \mid x_n = 0)} + \ln \frac{P(x_n \in \mathcal{A})}{P(x_n = 0)}$

  where $L_{E,n}(\mathbf{y}) = \ln \frac{P(\mathbf{y} | x_n \in \mathcal{A})}{P(\mathbf{y} | x_n = 0)}$ is **extrinsic** LLR and $L_{A,n} = \ln \frac{P(x_n \in \mathcal{A})}{P(x_n = 0)}$ is ***a priori*** LLR.
- ***A posteriori*** symbol-element activity LLR
  - $L_{n,j}(y) = \ln \frac{P(x_n = \mathcal{A}_j \mid \mathbf{y})}{P(x_n = 0 \mid \mathbf{y})} = \ln \frac{P(\mathbf{y} \mid x_n = \mathcal{A}_j)}{P(\mathbf{y} \mid x_n = 0)} + \ln \frac{P(x_n = \mathcal{A}_j)}{P(x_n = 0)}$
  - A set of symbol-element activity LLR can be converted into user activity LLR by log-sum-exp.
- Soft symbol information on x_n (transmitted by a user n)
  - $\{ P(x_n = 0) \} \cup \{ P(x_n = \mathcal{A}_j) \}_{j=1,\ldots,|\mathcal{A}|} \Leftrightarrow \{ L_{A,n,j} \}_{j=1,\ldots,|\mathcal{A}|}$

3. Proposed algorithm (2/5)

❖ **Serial concatenation of MAP-based active user detector (MAP-AUD) and MAP-based symbol detector (MAP-SD)**

- MAP-AUD and MAP-SD exchanges extrinsic LLR which serves *a priori* LLR to each other.



The subscripts, 1 and 2, mean it is generated by MAP-AUD and MAP-SD, respectively.

3. Proposed algorithm (3/5)

❖ MAP-AUD

- Finds a user n^* with maximum *a posteriori* activity probability.
  - $n^* = \arg\max_{n \in S^c} L_n(\mathbf{y})$

    $= \arg\max_{n \in S^c} \left(L_{E1,n}(\mathbf{y}) + L_{A,n} \right)$

    $= \arg\max_{n \in S^c} \left(\ln \sum_j \exp\left(L_{E1,n,j}(\mathbf{y}) \right) + L_{A,n} \right)$

    $\approx \arg\max_{n \in S^c} \left(\max_j L_{E1,n,j}(\mathbf{y}) + L_{A,n} \right)$, where S is the support of the previous iteration.
  - $L_{A,n} = \ln p_n/(1-p_n)$ and $L_{E1,n,j}(\mathbf{y}) = \mathcal{R}\left\{ \left(2\mathcal{A}_j \mathbf{r} - \left|\mathcal{A}_j\right|^2 \mathbf{a}_n \right)^H \mathbf{C}_n^{-1} \mathbf{a}_n \right\}$ (modified correlation between modified residual and column vector),

    where $\mathbf{r}$ (residual) $= \mathbf{y} - \sum_{i \in S} E[x_i]\mathbf{a}_i$ and $\mathbf{C}_n = \mathrm{cov}\left(\sum_{i \in S^c, i \neq n} x_i \mathbf{a}_i + \mathbf{v} \right)$ (Gaussian approx.)
  - Extrinsic information from MAP-SD is used to compute $E[x_i]$ ($i \in S$).
    - $E[x_i] = \sum_j P\left(x_i = \mathcal{A}_j\right)\mathcal{A}_j = \frac{\sum_j \exp\left(L_{E2,i,j}\right)\mathcal{A}_j}{1 + \sum_j \exp\left(L_{E2,i,j}\right)} \approx \frac{\sum_j \exp\left(L_{E2,i,j}\right)\mathcal{A}_j}{\sum_j \exp\left(L_{E2,i,j}\right)}$
- Delivers extrinsic soft symbol information on x_{n^*} (i.e.,$\{L_{E1,n^*,j}\}$) to MAP-SD.

3. Proposed algorithm (4/5)

❖ MAP-SD

- The user signal x_{n^*} detected by MAP-AUD is not an interference any more.
- Using extrinsic information $\{L_{E1,n^*,j}\}$ from MAP-AUD, MAP-SD generates (updates) $\{L_{E2,n,j}\}$ of all users $n \in S$ (i.e., users who are detected in the previous iteration)
- Similar to MAP-AUD, for all $n \in S$,
  - $L_{E2,n,j}(\mathbf{y}) = \mathcal{R}\left\{\left(2\mathcal{A}_j\mathbf{r}_n' - |\mathcal{A}_j|^2\mathbf{a}_n\right)^H \boldsymbol{\Gamma}^{-1}\mathbf{a}_n\right\}$ (see paper for details)

    where $\mathbf{r}_n' = \mathbf{y} - \sum_{T_n} E[x_i]\mathbf{a}_i$ (where $T_n = S \cup \{n^*\} - \{n\}$) and $\boldsymbol{\Gamma} = \mathrm{cov}\left(\sum_{i\in(S\cup\{n^*\})^c} x_i\mathbf{a}_i + \mathbf{v}\right)$
  - Extrinsic information from MAP-AUD is used to compute $E[x_i]$ $(i \in T_n)$

$$E[x_i] = \sum_j P\left(x_i = \mathcal{A}_j\right)\mathcal{A}_j \approx \frac{\sum_j \exp\left(\tilde{L}_{E1,i,j}\right)\mathcal{A}_j}{\sum_j \exp\left(\tilde{L}_{E1,i,j}\right)}$$

where $\tilde{L}_{E1,i,j} = L_{E1,i,j}$ (from MAP-AUD) if $i = n^*$, and $L_{E2,i,j}$ (from pervious iteration), otherwise.

- Augments the support (i.e., $S \leftarrow S \cup \{n^*\}$) and the soft symbol information (i.e., $\{L_{E2,n,j}\} \leftarrow \{L_{E2,n,j}\} \cup \{L_{E1,n^*,j}\}$)
- Finally, feeds the extrinsic information $\{L_{E2,n,j}\}$ back to MAP-AUD for next iteration.

3. Proposed algorithm (5/5)

❖ **Final step**

- If the power of the residual becomes small enough, the iteration is terminated.
- Finally, we have the indices of active users (S) and their soft symbol information (L_{E2}).
  - Hard decision on the symbol of the user n: $\hat{x}_n = \boldsymbol{\mathcal{A}}_j$ where $j = \arg\max_j L_{E2,n,j}$.

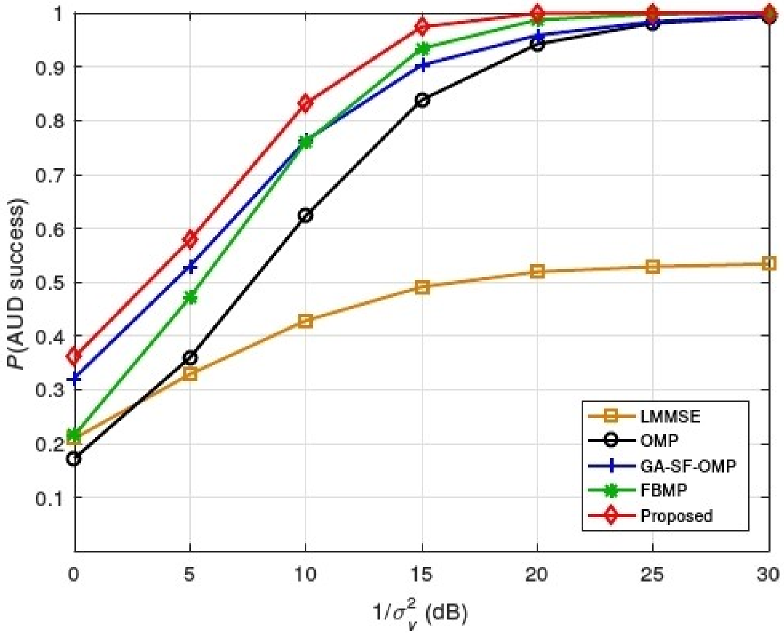
❖ **Inversion of the interference covariance matrix, $\boldsymbol{C_n}$ and $\boldsymbol{\Gamma}$**

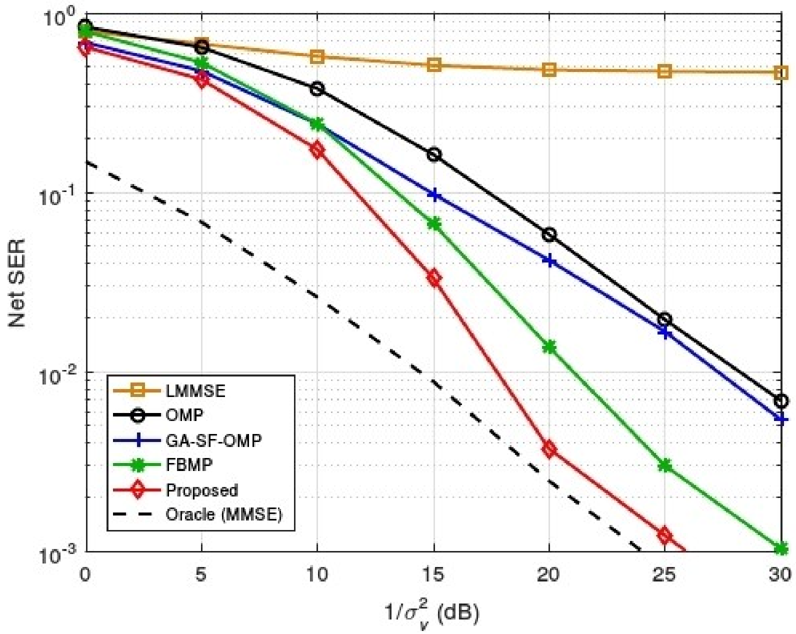
- It is burdensome since the inversion should be performed per user and iteration.
- There are two methods to compute the inversion with low complexity.
  - M1: We can recursively obtain the covariance matrices in each iteration by exploiting matrix inversion lemma.
  - M2: We can approximate the covariance matrix to be a diagonal matrix based on the assumption that the user-specific spreading sequence $\mathbf{s}_n$ is randomly generated.
    - Refer to the paper for details.

4. Simulation result

❖ **Simulation setup**

- N (# of users) = 128, M (spreading factor) = 64, Modulation: BPSK
- User activities are set to the same value (i.e., $p_n = p = 0.05$) for simplicity.
- The iteration is terminated when $\|\mathbf{r}\| < 10^{-4}$.
- Metric: Probability of successful active user detection (AUD), Net symbol error rate (SER)
  - ➢ Net SER = $1 - P(\text{Successful AUD} \cup \text{ Successful SD})$





OMP: Orthogonal matching pursuit, GA-SF-OMP: Genie-aided soft feedback OMP, FBMP: fast Bayesian matching pursuit
Oracle detector has the perfect knowledge of user activity.

5. Conclusion and further work

❖ Conclusion

- We proposed a CS-MUD algorithm exploiting the *a priori* information on user activity and the finite alphabet constraint.
- Specifically, we proposed MAP-based detection of the active user and its symbol.
  - The performance improvement by exchanging the extrinsic information between MAP-AUD/SD is similar to the well-known Turbo principle.

❖ Further work

- To extend this work to the case involved in multiple received vectors with common activity.
  - The data packet is composed of multiple symbols, not a single symbol.
  - By exploiting the common activity, the performance can be dramatically improved.

Thank You !

(Q&A)