# MU-MIMO precoding using Machine Learning Tools

G.Yashwanth Naik, P.Aashrith
*Indian Institute of Technology, Hyderabad*

***Abstract*—In a MU-MIMO setup where there are multiple users and base station antennas, we find the best set of users to transmit information such that the throughput of the entire system is maximized. We first implemented the basic algorithm which finds user set through exhaustive bruetforce search. However, such a algorithm cannot be used when the number of users are large. Then we implemented SUS algorithm, a low-complexity suboptimal user selection algorithm which uses orthogonality of users to find the best user set. We also designed a unsupervised learning model inspired form [1]. Further work include implementing the designed model.**

## I. Introduction

Multiple-input/Multiple-output(MIMO) systems have great potential to achieve high throughput in wireless systems. To improve throughput of these sytems we opt multiuser diversity, which is is a form of selection diversity among users; when the number of users $K$ is large, the base station can schedule its transmission to those users with favorable channel fading conditions. So, for a given channel matrix $H_{K\times M}$ (where $M$ is number of base station antennas), our challenge is to find the best subset of users for large $K$ and $M << K$.

## II. System Model

Consider a single-cell MIMO BC with a single base station supporting data traffic to user terminal.The base station is equipped with $M$ transmit antennas and the $k$th user terminal with $(N_k = 1)$ receive antenna. Therefore the signal received will be

$$y = Hx + z \tag{1}$$

where $x \in \mathbb{C}^{M\times 1}$ is the transmitted symbol from base station antennas, $H \in \mathbb{C}^{K\times M}$ is the channel matrix, $z \in \mathbb{C}^{M\times 1}$ is AWGN noise. Hence based on the given $H$ we try to select subset of users using various algorithms. In our work we don't focus on how $H$ is obtained, rather we assume $H$ is given and use that $H$ to find the best subset of users. Let $\mathcal{S}$ be the subset obtained from an algorithm then we define rate R as,

$$R = \log_2\left(|\mathbf{I} + W(\mathcal{S}) \cdot W(\mathcal{S})'|\right) \tag{2}$$

where,

$$W(\mathcal{S}) = H(\mathcal{S}) \cdot H(\mathcal{S})^{\dagger}$$

Here, $H(\mathcal{S})$ is the channel matrix of subset $\mathcal{S}$. Suppose if the cardinaity of set $\mathcal{S}$ is $L$ where $L \leq M$, then $H$ has dimensions $L \times M$.

## III. Approach

We approached this problem of selecting subset of users using two methods. First one is the basic bruteforce approach which has exponential time complexity. Hence for a large $K$ this algorithm is very slow. Second one is SUS(Suboptimal User Selection) algorithm which has lower time complexity and uses orthogonality of users for user selection.

### A. The Bruteforce Approach

This approach basically iterates over all the possible subsets and chooses the best one. In this approach we construct optimal set $\mathcal{S}$ as follows.

**Step 1)** Generate all the possible subsets of lengths varying from $1, 2, ....., M$.
**Step 2)** In this step we check whether the subset of users are orthogonal or not. For this we check whether $H(\mathcal{S}) \cdot H(\mathcal{S})'$ is equal to $I$ or not. If $H(\mathcal{S}) \cdot H(\mathcal{S})' = I$, then the subset of users are orthogonal else they are not orthogonal.
**Step 3)** Exclude the non-orthogonal sets and collect the orthogonal sets.
**Step 4)** Calculate the rates of the orthogonal sets using (2).
**Step 5)** Our final optimal set is the one with highest rate.

### B. The SUS Approach

We construct a suboptimal user group $\mathcal{S}$ using a semiorthogonal user selection (SUS) algorithm as follows.
**Step 1)** Initialize,

$$\begin{aligned} \mathcal{T}_1 &= \{1, \ldots, K\} \\ i &= 1 \\ \mathcal{S}_0 &= \phi \end{aligned} \tag{3}$$

where,
$\mathcal{T}_i$ - set of user indices at $i^{th}$ iteration
$S_i$ - set of selected users at $i^{th}$ iteration
$i$ - counter variable
**Step 2)** For each user $k \in \mathcal{T}_i$, calculate $\mathrm{g}_k$, the component of $\mathbf{h}_k$ orthogonal to the subspace spanned by $\left\{\mathbf{g}_{(1)}, \ldots, \mathbf{g}_{(i-1)}\right\}$

The component of $h_k$ projected onto the subspace spanned by $\left\{\mathbf{g}_{(1)}, \ldots, \mathbf{g}_{(i-1)}\right\}$ is $\sum_{j=1}^{i-1} \frac{\mathbf{h}_k \mathbf{g}_{(j)}^*}{\left\|\mathbf{g}_{(j)}\right\|^2} \mathbf{g}_{(j)}$ .

$$\begin{aligned} \mathbf{g}_k &= \mathbf{h}_k - \sum_{j=1}^{i-1} \frac{\mathbf{h}_k \mathbf{g}_{(j)}^*}{\left\|\mathbf{g}_{(j)}\right\|^2} \mathbf{g}_{(j)} \\ &= \mathbf{h}_k \left( \mathbf{I} - \sum_{j=1}^{i-1} \frac{\mathbf{g}_{(j)}^* \mathbf{g}_{(j)}}{\left\|\mathbf{g}_{(j)}\right\|^2} \right) \end{aligned}$$

When $i = 1 \implies \mathbf{g}_k = \mathbf{h}_k$.

**Step 3)** We select $i^{th}$ user as,

$$\begin{aligned} \pi(i) &= \arg\max_{k \in \mathcal{T}_i} \|\mathbf{g}_k\| \\ \mathcal{S}_0 &\leftarrow \mathcal{S}_0 \cup \{\pi(i)\}, \text{ (include the user)} \\ \mathbf{h}_{(i)} &= \mathbf{h}_{\pi(i)} \\ \mathbf{g}_{(i)} &= \mathbf{g}_{\pi(i)}. \end{aligned} \tag{4}$$

where,

$\pi(i)$ - index of user with max $\|\mathbf{g}_k\|$ at $i^{th}$ iteration

**Step 4)** If $|\mathcal{S}_0| < M$, then calculate $\mathcal{T}_{i+1}$, the set of users semiorthogonal to $\mathrm{g}_{(i)}$ in $\mathcal{T}_i$

$$\mathcal{T}_{i+1} = \left\{ k \in \mathcal{T}_i, k \neq \pi(i) \mid \frac{\left|\mathbf{h}_k \mathbf{g}_{(i)}^*\right|}{\|\mathbf{h}_k\ \|\| \left\|\mathbf{g}_{(i)}\right\|} < \alpha \right\} \tag{5}$$

$$i \leftarrow i + 1$$

where , $\alpha$ is a small positive constant.

**if** $\mathcal{T}_{i+1} \neq \phi$ and $|S_0| < M$ **then**
  go to step2
**else**
  break
**end if**

So, by using the above 4 steps of SUS algorithm we obtain the best subset of users.

## IV. Simulations

The MATLAB implementations of both the algorithms can be found here .

We tested these algorithms on the data with (10000 channel vectors, 16 users, 64x1 for each user on one subcarrier) provided by Prof. Sai Dhiraj Amuru.

## V. Unsupervised Learning Approach

We developed the Unsupervised Learning pipeline as shown in Fig.1 . This Unsupervised Approach is mainly inspired from [1].

The above blocks in Fig.1 are represented by (A),(B),(C),(D). Below is the detailed description of these blocks.

### A. Block A - Input

The input for the pipeline is the channel matrix $\mathcal{H}$ with dimensions $K \times M$, where $K$ is the number of users and $M$ is number of base-station antennas.

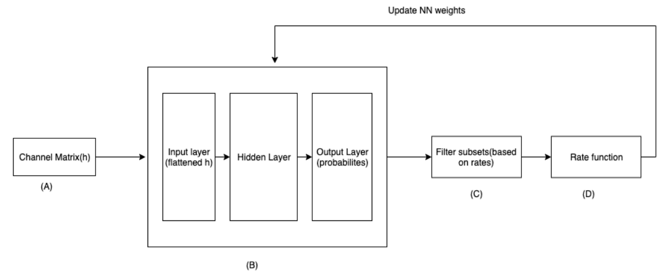


Fig. 1. Pipeline

### B. Block B - Neural Network

(B) has 3 layers. They are Input, Hidden and Output Layers.

**Input Layer :** The input layer for the Neural Network is the flattened $\mathcal{H}$ matrix with dimensions $K \cdot M \times 1$.

**Hidden Layer :** The hidden layer dimensions are yet to be decided based on the implementation.

**Output Layer :** The output layer will contain probabilities for a certain user to be in the final set. For example if K = 5 then the output layer may look like $\begin{bmatrix} 0.7 & 0.15 & 0.12 & 0.02 & 0.01 \end{bmatrix}_{5\times1}^T$. The dimensions for this layer will be $K \times 1$

### C. Block C - Filter Subsets

Given the probabilities for a particular user to be in the set. This block does the following:

- Sort the vector based on the probabilities.
- Compute the rate of top i users where i = 1 to M. For example if K = 5, M = 3 then we compute rates for (1),(1,2),(1,2,3).Here we need to perform M iterations every time.
- Output the best rate subset from the above computed rates.

### D. Block D - Cost Function(Rate)

Given the subset as input to the rate function, the cost function(here rate) directs us to move in the direction of best rate. It Generates gradients for backward propagation and updates the NN weights $\implies$ probabilities will be updated. After training for an input we can obtain the final subset using the final probabilities and the filter set block (C).

**NOTE** : This model is tentatively proposed and simulations are yet to be done.

## VI. Results

The observations and results are summarized in the following points.

1) In bruteforce approach multiple subsets have highest cardinality and rate. So in order to choose a single subset we can give priority orders for users so that we can choose the set with largest sum of priorities of users in the subset.

2) The optimal subset obtained from SUS Algorithm depends on $\alpha$. If $\alpha$ is large $\implies$ suboptimality of sets obtained is very low and vice versa.
3) For random and dependent vectors of $H$ as input, the the rate of optimal user set was higher than the rate of SUS user set.
4) We can also input the priorities of users to Unsupervised Learning model so that single user set is obtained at the end.

## References


[1] R. Nikbakht, A. Jonsson and A. Lozano, "Unsupervised Learning for Cellular Power Control," in IEEE Communications Letters, vol. 25, no. 3, pp. 682-686, March 2021, doi: 10.1109/LCOMM.2020.3027994.

[2] Taesang Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," in IEEE Journal on Selected Areas in Communications, vol. 24, no. 3, pp. 528-541, March 2006, doi: 10.1109/JSAC.2005.862421.