RFF Based Detection for Massive MIMO

Project Report

by

VARUN CHHANGANI

DISCIPLINE OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
INDORE, MADHYA PRADESH (INDIA)

July, 2019

RFF Based Detection for Massive MIMO

A PROJECT REPORT

by

VARUN CHHANGANI

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE, MADHYA PRADESH (INDIA)

July, 2019

INDIAN INSTITUTE OF TECHNOLOGY INDORE,
MADHYA PRADESH (INDIA)

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled "**RFF Based Detection for Massive MIMO**" in the partial fulfillment of the requirements for the award of the certificate of internship and submitted in the DISCIPLINE OF ELECTRICAL ENGINEERING, INDIAN INSTITUTE OF TECHNOLOGY INDORE, MADHYA PRADESH (INDIA) is an authentic record of my own work carried out during the time period from 10^{th} May 2019 to 10^{th} July 2019 under the supervision of Prof. Vimal Bhatia.

The matter presented in this project report has not been submitted by me for the award of any other degree of this or any other institute.

10/07/2019
Signature of the student with date
(VARUN CHHANGANI)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of project Supervisor with date
(Prof. Vimal Bhatia)

Contents

1 Use of Online Learning Method with KLMS in RKHS

In this chapter, we introduce the reader to the problem of non-linear function approximation over Reproducing Kernel Hilbert Spaces(RKHS), and formulate online solutions like the Kernel Least Mean Squares(KLMS) algorithm toward solving problems worked upon in [1]–[5].

First we introduce the terminology followed herein. Let x denote the given sample and y be the output where the output $y = f(x)$ where $f(\cdot)$ is a non-linear function. The value y is approximated using linear regression using gradient descent [6]–[8] on the given samples for multiple iterations. Let $(\cdot)_i$ denote value of $(\cdot)$ at iteration i; and f_i denote the estimate of function f at i^{th} iteration. Thus, for learning f_i, the samples $x_1, x_2, \ldots, x_i$ along with the known outputs for the respective inputs as $y_1, y_2, \ldots, y_n$.

Due to the inherently non-linear relation between y and x, a kernel function k is used.

This k, in RKHS can be said to be reproducible by function ϕ as $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. This will be further studied in the next chapter.

For KLMS, the instantaneous mean squared objective is considered. This objective is minimized and thus the function is approximation is performed via stochastic gradient algorithm. Let $\mathcal{L}$ denote the loss function. So,

$$\mathcal{L}(f) = (y - f(x))^2$$

Thus using online learning,

$$f_n = f_{n-1} + \mu \epsilon_n k(x_n, \cdot)$$

here, $\epsilon = y - f(x)$ and μ is the step size

So, after $n-1$ iterations, output function appears to be (when the inital function is $f_0 = \mathbf{0^T} \cdot$):

$$f_{n-1} = \sum_{i=1}^{n-1} \alpha_i k(\cdot, x_i)$$

(Here $\alpha_i = \mu \epsilon_i$)

2 Kernel Approximation through Random Fourier Features

In this chapter, the theory of Random Fourier Feature space(RFFs) is reviewed for approximation of feature map ϕ. The approximation of the function $\phi(\cdot)$ is denoted by $z_\Omega(\cdot)$.

Kernel-based learning methods involve a large number of kernel evaluations between training samples. In the batch mode of operation, for example, this means that a large kernel matrix called the Gram Matrix has to be computed, increasing the computational cost of the method significantly [9]. Hence, to alleviate the computational burden, one common approach is to use an approximation to the kernel evaluation.

The most common methods are Nyström method and random Fourier Feature approach [3] that are compared in [10].

Random Fourier Features is one of the viable methods for approximation of feature map where the input data is mapped to an approximate RKHS. The input data space vectors x are mapped to an approximate RKHS vectors $z_\Omega(x) \in \mathbb{R}^D$ with dimension higher than that of the input space (but less than $\mathcal{H}$) using a randomized feature map.

$$z_\Omega : \mathbb{R}^d \to \mathbb{R}^D$$

$$k(x_n, x_m) = \langle z_\Omega(x_n), z_\Omega(x_m) \rangle$$

Theorem 1 (Bochner[11]) *A continuous function $k(x, y) = k(x - y)$ on $\mathbb{R}^d$ is positive definite if and only if $k(\delta)$ is the Fourier Transform of a non-negative measure.*

Theorem 2 (Rahimi and Recht[3]) *Consider a shift invariant positive definite kernel $k(x - y)$ defined on $\mathbb{R}^d$ and its Fourier transform*

$$p(\omega) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} k(\delta) e^{-i\omega\delta} d\delta$$

which (according to Bochner's Theorem) it can be regarded as a probability density function. Then, defining $z_{\omega,b} = \sqrt{2}cos(w^T x + b)$ it turns out that $k(x-y) = E_{\omega,b}[z_{\omega,b}(x)' z_{\omega,b}(y)]$ where ω is drawn from p and b from the uniform distribution on [0,2π].

Thus for D dimensions, approximating k using D Fourier features drawn from probability density function p $(w_1, w_2, \ldots, w_D)$ and D random numbers $(b_1, b_2, \ldots, b_n)$ from uniform distribution gives

$$k(x_n, x_m) \approx \frac{1}{D} \sum_{i=1}^{D} (z_{w_i,b_i}(x_n) z_{w_i,b_i}(x_m))$$

For Gaussian kernel, the Fourier transform is $p(w) = \left(\sigma/\sqrt{2\pi}\right)^D e^{-\frac{\sigma^2||w||^2}{2}}$ which evaulates to a multivariate Gaussian function with covariate matrix $\frac{\mathbf{I_D}}{\sigma^2}$ and mean $\mathbf{0_D}$

Thus $z_\Omega(\cdot)$ can be represented as:

$$z_\Omega(u) = \sqrt{\frac{2}{D}} \begin{pmatrix} cos(w_1^T u + b_1) \\ cos(w_2^T u + b_2) \\ \vdots \\ cos(w_D^T u + b_D) \end{pmatrix}; \Omega = \begin{pmatrix} w_1 w_2 \dots w_D \\ b_1 b_2 \dots b_D \end{pmatrix}_{(d+1)\times D}$$

Using this RFF based approximation of an kernel, the regression model follows as:

$$f_{n-1}(x_n) \approx \left(\sum_{i=1}^{n-1} \alpha_i z_\Omega(x_i)\right)^T z_\Omega(x_n)$$

which can be written as $f(x_n) \approx \theta^T z_\Omega(x_n)$ Thus, the function approximation can be performed by a stochastic gradient update on θ as follows:

$$\theta_n = \theta_{n-1} + \mu \epsilon_n z_\Omega(x_n)$$

The approximation for the Guassian kernel $\kappa(\triangle) = e^{-\gamma||\triangle||^2}$ is approximated using 1000 Random Fourier Features as shown in the following graphs as in Figure **??**.

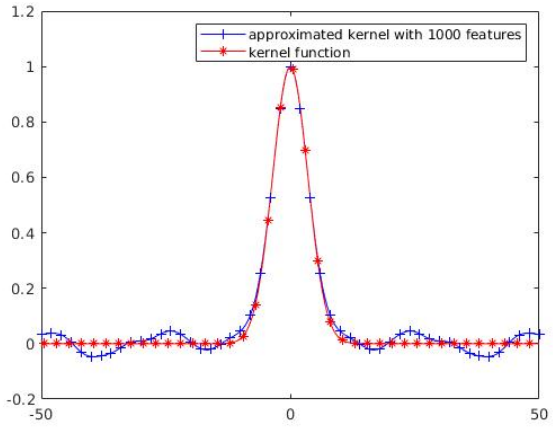

Figure 1: Kernel Approximation

3 Review of Adaptation Strategies for Distributed Learning over Networks

In this chapter, we review various adaptation strategies for learning over networks. We first present the network model, then the data model is presented, and finally some distributed adaptation algorithms are reviewed.

3.1 Network model

In this section, the network model is presented which is adopted from [2].

We consider a network with a set of nodes given by $\mathcal{N}$, such that $|\mathcal{N}|$=N and $\mathcal{N}_k$ denote the neighbourhood of node k (including k).

The convex combination scalars $\{a_{lk}\}$ are chosen such that they satisfy

$$\begin{gathered} a_{lk} \geq 0 \\ \sum_{l \in \mathcal{N}_k} a_{lk} = 1 \\ a_{lk} = 0 \forall l \notin \mathcal{N}_k \end{gathered}$$

The combination matrix is denoted by $A = \{a_{lk}\}_{N \times N}$ where $a_{lk} \in [0, 1]$ which which is the measure of confidence of node l on node k's regression vector w. Thus, the A can be adjusted according to the noise at the node k. The matrix A varies according to the combination policy. Possible choices for combination matrix A are Metropolis, Laplacian and the nearest neighbors in case of non-adaptive combination matrix [12]–[14]. An adaptive combination matrix is modelled using Hasting's rule or relative-variance rule wherein the variance of noise is taken into consideration [2]. In current model, Metropolis combination matrix is chosen.

3.2 Data model

Considering the global cost function for the network, then w^o (network goal) is

$$w^o = \arg\min_w \sum_{k=1}^{N} J_k(w)$$

Considering $J(w)$ to be strongly convex that is $\nabla_w^2 J(w) > 0$

Also,

$$\min_w J(w) \equiv \min_w \mathbb{E} Q(w; \mathbf{x})$$

$$w_i \leftarrow w_{i-1} - \mu \cdot \nabla_w Q(w_{i-1}; \mathbf{x_i})$$

The conditions of step size $\mu(i)$ can be outlined by [2], [4].

$$\sum_{i=0}^{\infty} \mu(i) = \infty$$

$$\sum_{i=0}^{\infty} \mu^2(i) < \infty$$

$$\mu(i) = \frac{c}{i^p}, p \in \left(\frac{1}{2}, 1\right]$$

This introduces gradient noise $s(w)$ as we must be using cost function or risk function $J(w)$ instead, we are using loss function $Q(w)$

$$s(w_{i-1}) \triangleq \nabla_w Q(w_{i-1}) - \nabla_w J(w_{i-1})$$

Where $J_k(w)$ is the loss calculated at node k for parameter vector w. N is the number of Nodes in the Network $\mathcal{N}$. The neighborhood of k is denoted by $\mathcal{N}_k$

3.3 Basic strategies to update w

In this section, some distributed adaptive strategies are reviewed for enhancing the understanding of the reader while formulating RFF based detector for massive MIMO.

3.3.1 Non cooperative strategies

In this case, each agent applies its own LMS and thus, has updation formula:

$$w_{k,i} = w_{k,i-1} + \mu u_{k,i}^* [d_{k,i} - u_{k,i} w_{k,i-1}]$$

Here $\text{MSD}_k \triangleq \lim_{i\to\infty} E||\widetilde{w}_{k,i}||^2$

For sufficiently small step size, it is shown in [4] that

$$\text{MSD}_{\text{ncop,k}} \approx \frac{\mu M}{2} \sigma_{v,k}^2$$

the rate towards convergence here is:

$$r \approx 1 - 2\mu \cdot \lambda_{min}(R_u)$$

That is dependent on smallest eigenvaue of R_u matrix

For overall network [2] shows that

$$\text{MSD}_{\text{ncop}}^{\text{network}} \approx \frac{\mu M}{2} \left(\frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 \right)$$

3.3.2 Centralized Fusion-Based Solution

For applying LMS at a central processor, the input and the response will tapped at the agents and will be sent to a central fusion processor which gives the updated estimator for w^o from w_{i-1} to w_i for the iteration i.

The updation strategy for the estimator as shown in [2] is followed as

$$w_i = w_{i-1} + \mu \left(\frac{1}{N} \sum_{k=1}^{N} (u_{k,i}^* [d_{k,i} - u_{k,i} w_{k,i-1}] \right)$$

It is thus deduced that

$$\text{MSD}_{\text{cent}} \approx \frac{\mu M}{2} \frac{1}{N} \left(\frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 \right)$$

Thus, an N fold increase is observed as compared to $\text{MSD}_{\text{ncop}}^{\text{network}}$.

3.3.3 Diffusion Strategies

Diffusion strategies allow the solution to be found in a distributed and adaptive manner. Compared to the noncooperative solution, these strategies introduce a useful aggregation step that helps incorporate into the adaptation mechanism information collected from the local neighborhoods. Diffusion strategies as compared to consensus have relatively more stable convergence.

Combine-Then-Adapt (CTA) In this diffusion strategy, the estimators of the neighbors are aggregated using the combination matrix and then the estimator is updated according to the LMS strategy.

Thus, we find an intermediate value of the estimator $\psi_{k,i-1}$ for the agent k by aggregating the estimator's values at the neighbors of the agent k after the iteration $i-1$.

Thus the updation strategy is

$$\begin{cases} \psi_{k,i-1} = \sum_{l \in \mathcal{N}_k} a_{lk} w_{l,i-1} \\ w_{k,i} = \psi_{k,i-1} + \mu u_{k,i}^* [d_{k,i} - u_{k,i} \psi_{k,i-1}] \end{cases}$$

OR

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} w_{l,i-1} + \mu u_{k,i}^* \left[d_{k,i} - u_{k,i} \sum_{l \in \mathcal{N}_k} a_{lk} w_{l,i-1} \right]$$

Adapt-Then-Combine (ATC) In this diffusion strategy, the estimators adapted according to the behaviour of unknown model observed by the agents as the intermediate value of estimator for the iteration i. The values of the intermidiate estimators of the neighboring agents are then aggregated according to the combination policy to get the updated value of estimator.

The updation policy is thus specified mathematically as

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu u_{k,i}^* [d_{k,i} - u_{k,i} w_{k,i-1}] \\ w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} \psi_{l,i} \end{cases}$$

OR

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} (w_{k,i-1} + \mu u_{k,i}^* [d_{k,i} - u_{k,i} w_{k,i-1}])$$

3.3.4 Consensus

In consensus based distributed adaptive filtering, the stochastic adaptation over node k is performed upon the combination of the estimator functions of $\mathcal{N}_k$ and is subject to error in detection of correct symbol in the previous iteration by node k which makes this strategy relatively unstable. This has been shown in [2].

For two time scale approach, the updation strategy becomes

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} w_{l,i-1} + \mu u_{k,i}^* \left[d_{k,i} - u_{k,i} w_{k,i-1} \right]$$

The simulation was thus conducted for different strategies and the result is shown in Figure 2.

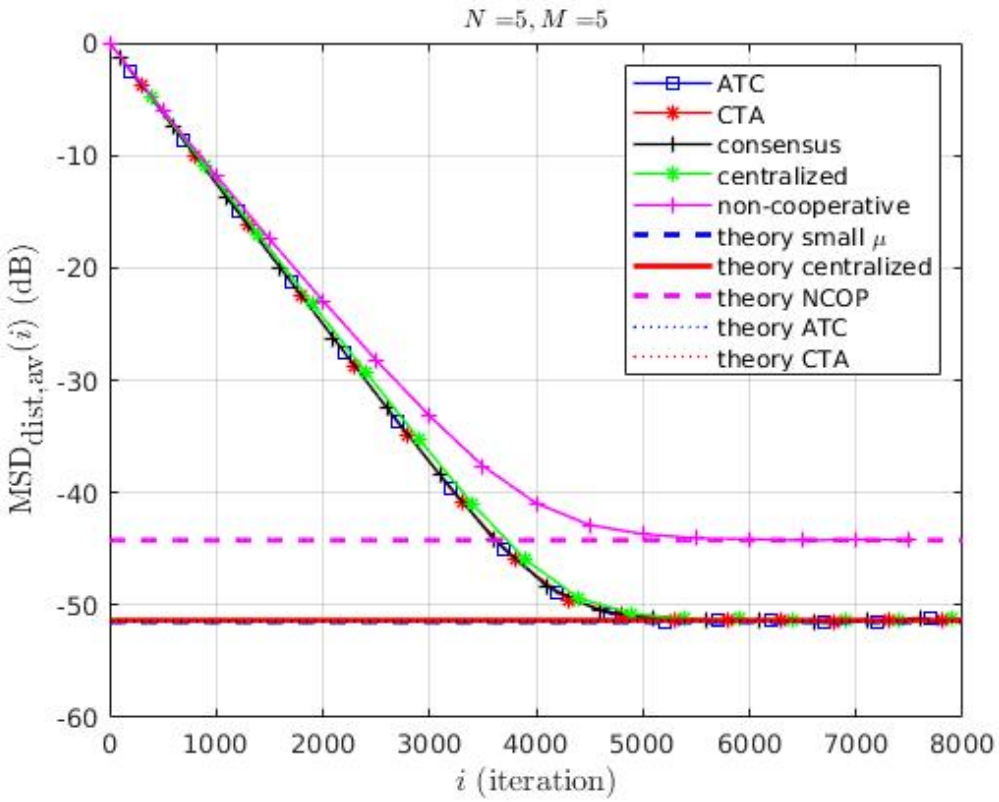

Figure 2: Learning over network

4 Review of Classical MIMO Recievers

4.1 MIMO

MIMO stands for multiple input multiple output. In this type of transmission, the diversity and the spatial multiplexing is used to harness the benefits of multiple antennas to give better signal reception and higher data rates

4.2 Methods used for simulation

- Zero Forcing (ZF) recievers
- Minimum Mean Squared Error (MMSE) estimator reciever
- ZF with SIC
- MMSE with SIC

4.3 System Model

Consider N_t transmitters and N_r receivers, with the transmitter signal x and received signal y represented as:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_t} \end{bmatrix}; y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_r} \end{bmatrix}$$

The received signal follows as:

$$y = Hx + n$$

Where $H_{N_r \times N_t}$ is the channel matrix and $n_{N_r \times 1}$ denotes additive white Gaussian noise (AWGN) vector.

4.4 Condition Number

Condition number is a ratio of biggest and smallest singular values and quantifies correlation and noise. Condition number is a measure of how sensitive the function is to errors in input. As the condition number increases, the noise is amplified as the ratio between the spectral value and the smallest singular value increases. The ideal value of condition number for MIMO is 1 (or 0dB) and values below 10dB or 10 are desirable.

The condition number of moment matrix in linear regression can be used as diagonastic for collinearity. A moment matrix with a low condition number is called well-conditioned and otherwise is called ill-conditioned.

Condition number of matrix, where $\kappa(H)$ denotes the condition number of matrix H is:

$$\kappa(H) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

4.5 Zero Forcing

As $y = Hx + n$

We here want to minimize $||\hat{x} - x||^2$

Thus we take

$$\hat{x} = H^{\dagger}y$$

This is known as zero forcing receiver. Here we are taking the inverse of the channel matrix H. For a non invertible channel matrix, its pseudo-inverse $H^{\dagger} = (H^H H)^{-1}H^H$ is used.

$$\hat{x} = (H^H H)^{-1}H^H y$$

4.6 Drawbacks of ZF

As

$$\hat{x} = (H^H H)^{-1}H^H y$$

Thus,

$$\hat{x} = H^{\dagger}(Hx + n) = x + H^{\dagger}n$$

Notably, if the channel matrix is ill-conditioned the variance of the noise is amplified due to pseudo-inverse operation.

The condition number is specified by

$$\kappa(H) = \frac{\sigma_{max}}{\sigma_{min}}$$

4.7 Minimum Mean Square Error estimation

$E||\hat{x} - x||^2$ has to be minimized here.

Suppose $\hat{x} = c^T y$. Thus, we want to minimize $E||c^T y - x||^2$:

$F = E||c^T y - x||^2 = E(c^T y - x)(c^T y - x)^T$

$F = E(c^T y y^T c - x y^T c - c^T y x^T + x x^T)$

$F = c^T R_{yy} c - R_{xy} c - c^T R_{yx} + R_{xx}$

$F = c^T R_{yy} c - 2c^T R_{yx} + R_{xx}$

$\nabla_c F = 0 = 2R_{yy}c - R_{yx}$

$R_{yy}c = R_{yx}$

$c = R_{yy}^{-1} R_{yx}$

$\hat{x} = c^H y = R_{xy} R_{yy}^{-1} y$

$\hat{x} = (H^H H + \sigma^2 I)^{-1} H^H y$

4.8 Theoretical Comparison between ZF and MMSE Receivers

In MMSE, the condition number is given by

$$\kappa(H) = \frac{\sigma_{max} + \sigma^2}{\sigma_{min} + \sigma^2}$$

Thus for high SNR, the performance of MMSE based receiver is equivalent of ZF; however at low SNR, MMSE can be viewed as a matched filter.

Also, it can be found that as $\sigma^2 >> \sigma_{max}, \sigma_{min}$, $\kappa(H) \approx 1$ which circumvents the problem of noise amplification.

4.9 SIC Techniques to Improve Nulling estimators

The estimate of x can be further improved by successively cancelling the interference.

$$y_1 = h_{1,1}x_1 + h_{1,2}x_2 + n_1$$
$$y_2 = h_{2,1}x_1 + h_{2,2}x_2 + n_2$$

Taking $P_{x_1} = |h_{1,1}|^2 + |h_{2,1}|^2$ $P_{x_2} = |h_{1,2}|^2 + |h_{2,2}|^2$, $\hat{x_1}$ and $\hat{x_2}$ can be found with techniques like ZF, MMSE, etc. and then re-estimate the signals with lesser power after subtracting the higher power terms from the observed signal.

4.10 ZF SIC

First ZF is applied

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{N_t} \end{bmatrix} = H^{\dagger}y$$

The power is calculated as

$$P_{x_1} = |h_{1,1}|^2 + |h_{2,1}|^2 + \cdots + |h_{N_r,1}|^2$$
$$P_{x_2} = |h_{1,2}|^2 + |h_{2,2}|^2 + \cdots + |h_{N_r,2}|^2$$
$$\vdots$$
$$P_{x_{N_t}} = |h_{1,N_t}|^2 + |h_{2,N_t}|^2 + \cdots + |h_{N_r,N_t}|^2$$

according to the ordering of power,

The $\hat{x}$ with the highest power is assumed as it is and after that, that symbol is removed from the rest of the data.

Thereafter $\hat{x}$ is recalculated using ZF technique.

4.11 MMSE SIC

The MMSE SIC based receiver is similar to ZF except the calculation of $\hat{x}$ is done using the MMSE technique instead of ZF. That is

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{N_t} \end{bmatrix} = ((H^H H + \sigma^2 \mathrm{I})^{-1} H^H y$$

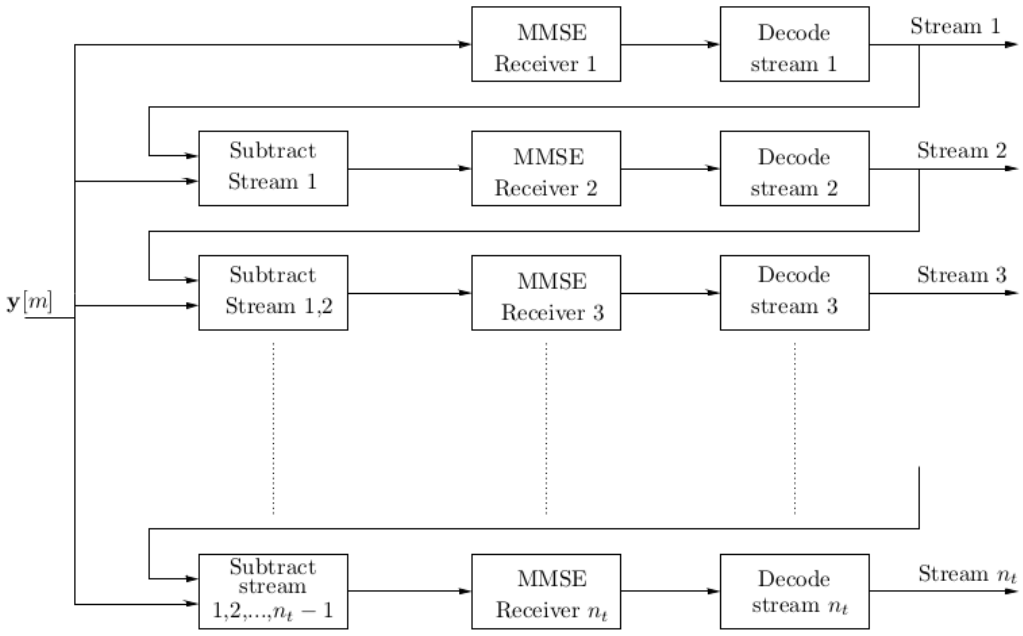

Figure 3: Successive Interference Cancellation (adapted from [15])

This is followed by finding the sorted order of power and thereafter is removed from the signal and the process is repeated to find values of all $\hat{x_i}$

4.12 Theoretical Performance

As the SIC technique differentiates between near user and far user, it has a lower BER than the non SIC techniques. Also, due to the problems with ZF discussed earlier, the MMSE technique has a lower BER than ZF for low SNR but approaches ZF at high SNR.

Therefore,

$$\text{BER}_{\text{ZF}} > \text{BER}_{\text{MMSE}}$$

$$\text{BER}_{\text{ZF}} > \text{BER}_{\text{ZF\_SIC}}$$

$$\text{BER}_{\text{MMSE}} > \text{BER}_{\text{MMSE\_SIC}}$$

$$\text{BER}_{\text{ZF\_SIC}} > \text{BER}_{\text{MMSE\_SIC}}$$

4.13 Simulation Results

It is thus observed that all the theoretical statements hold here and observe that at less than 34 dB for the given Rayleigh Channel, the MMSE SIC achieved close to 0 BER which happened at less than 36 for ZF-SIC, and less than 42 for ZF and MMSE.

Also it is observed that MMSE approaches to ZF at close to 38 dB SNR.

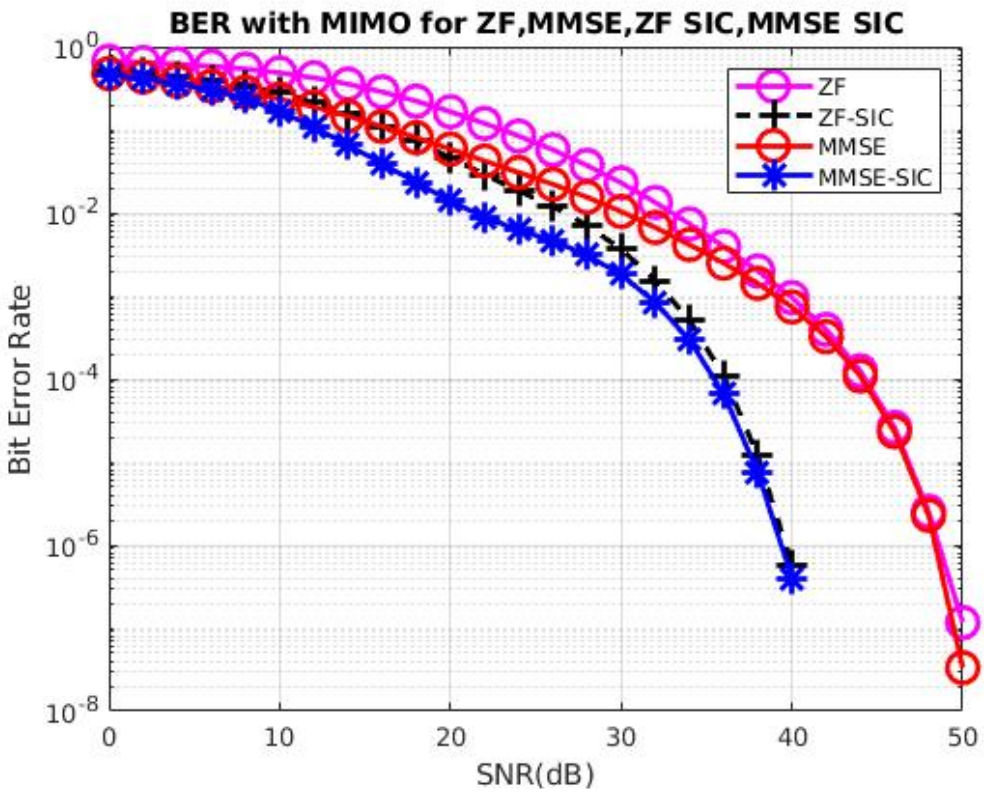

Figure 4

5 Contribution: KLMS based Parallel Multiuser Detector for Massive-MIMO

5.1 Motivation

The paper on RKHS based online detection [1] discusses detection of transmitted signal by using a Gaussian Kernel and learning inputs and detecting output from them using learning via dictionary. Implementation of Parallel MU Massive MIMO detector by learning using Kernel Approximation and learning using Linear Regression was studied in this problem.

5.2 System Model

In this section, the system model of the considered uplink-MU massive-MIMO system is presented.

As considered in [1], the uplink system model of M-users each with a single transmitter antenna, describing a frequency selective Rayleigh fading channel [16] under the assumption of transmit side power-ampilifier nonlinearity (as given in [17]).

The Rayleigh fading channel denoted by $\mathbf{H}_s$ is given by $\mathbf{H}_s \forall s \in [0, T-1]$ where T is the memory of the channel, the correlation factorρ is considered in the fading channel as in [18], $\mathbf{H}_s = \mathbf{R}_1^{\frac{1}{2}} \mathbf{H}'_s \mathbf{R}_2^{\frac{1}{2}}$ where

$$\mathbf{R}_1 = \text{Toeplitz}([1, \rho^2, \rho^4, \ldots, \rho^{2N-2}])$$

$$\mathbf{R}_2 = \text{Toeplitz}([1, \rho^2, \rho^4, \ldots, \rho^{2M-2}])$$

$$\mathbf{H}'_s \in \mathcal{CN}(0, 1)$$

Thus the recieved signal can be written as:

$$\mathbf{y}_i = \Sigma_{s=0}^{T-1} \mathbf{H}_s f(\mathbf{x}_{i-s}) + \mathbf{n}_i \tag{1}$$

where $\mathbf{y}_i \in \mathbb{C}^N$ is the received signal. For $\mathbf{H}_{s,s=0} \in \mathbb{C}^{NM}$ we have the LOS path with the fixed rice factor as given in [16]. The $\mathbf{H}_{s,s\neq 0} \in \mathbb{C}^{NM}$ is a complex Gaussian matrix with zero mean and unit variance denoted as $\mathbf{H} \sim \mathcal{CN}(0, 1)$. Where $N(>> M)$ is the number of antennas, subscript $(\cdot)_i$ denotes the value of quantity at the ith time instant. For a MU massive MIMO, $N \geq 10M$ is chosen. The $f(\cdot)$ is the transmit side power amplifier which in general is a non-linear function modelled by Rapp model as given in [17] that is used in [1]. The vector x_i denotes the transmitted symbols of all the M-users at the ith time instant, is the transmitted vector. The noise vector $\mathbf{n}_i \in \mathbb{C}^N$ is additive white Gaussian noise for the ith instant and can be denoted as $\mathbf{n}_i \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$, where σ^2 denotes the variance of the random Gaussian variable.

5.3 Proposed Technique

In this chapter, a technique is proposed to solve for $\mathbf{x}_i$ from given $\mathbf{y}_i$ as given in equation (1).

The method is divided in three parts namely, a) Block mapping to RKHS as in [1], b) Kernel approximation using RFF , and c) KLMS-Regression based detector

Block mapping to RKHS As given in [1], for the lth block of the channel matrix, $\mathbf{H}^l$ is defined as:

$$\mathbf{H} = [\mathbf{H}^{1^T}\mathbf{H}^{2^T}\dots\mathbf{H}^{L^T}]^T$$

Thus, size of $\mathbf{H}^L$ is $\frac{N}{L} \times M$. The L is taken in the bounds that $\frac{N}{L} >> M$

Also, the disjoint blocks of $\mathbf{y}_i$, $\mathbf{y}_i^l$ at time instant i can be given as:

$$\mathbf{y}_i = [\mathbf{y}_i^{1^T}\mathbf{y}_i^{2^T}\dots\mathbf{y}_i^{L^T}]^T$$

Thus the relation for the disjoint blocks is:

$$\mathbf{y}_i^l = \sum_{s=0}^{T-1} \mathbf{H^l}_s f(\mathbf{x}_{i-s}) + \mathbf{n}_i^l \tag{2}$$

where $\mathbf{n}_i^l$ is the lth block of the noise vector $\mathbf{n}_i$

Kernel approximation using RFF As RFF can be applied to Kernels which satisfy the conditions given in Theorem **??**, the kernel chosen is:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma||[\mathcal{R}(\mathbf{x}_i)^T, \mathcal{I}(\mathbf{x}_i)^T]^T - [\mathcal{R}(\mathbf{x}_j)^T, \mathcal{I}(\mathbf{x}_j)^T]^T||^2\right)$$

where $\mathcal{R}(\cdot)$ returns the real part of the input and $\mathcal{I}(\cdot)$ returns the imaginary part of the input.

As the kernel function is Gaussian Ω can be found using RFF (Theorem **??**) as given in chapter **??** such that $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle z_\Omega(\mathbf{x}_i), z_\Omega(\mathbf{x}_j)\rangle$.

After

$$\hat{\mathbf{x}}_i = W z_\Omega(\mathbf{y}_i)$$

Where W is the weight vector used for regression, as discussed in the next paragraph.

For parallelization, we can define Ω^l as $\Omega = [\Omega^1, \Omega^2, \dots, \Omega^L]$ and Ω^l is of size $d + 1 \times \frac{D}{L}$, where D is the number of random Fourier features used.

Thus,

$$z_{\Omega^l(x)} = \sqrt{\frac{2}{D}} \begin{pmatrix} \cos(\omega_{(L*(l-1)+1)}^T x + b_{(L*(l-1)+1)}) \\ \cos(\omega_{(L*(l-1)+2)}^T x + b_{(L*(l-1)+2)}) \\ \vdots \\ \cos(\omega_{(L*l)}^T x + b_{(L*l)}) \end{pmatrix}$$

$$\hat{\mathbf{x}}_i^l = W^l z_{\Omega^l}(\mathbf{y}_i^l)$$

Where W^l is defined using W as $W = [W^1, W^2, \ldots, W^L]$ Where W^l is of size $N \times \frac{D}{L}$

KLMS-Regression based detector Let the predicted transmitted symbol be $\hat{\mathbf{x}}$. Thus, we want to find W such that, $\hat{\mathbf{x}}_i = W^T \mathbf{y}_i$. For l^{th} block, $\hat{\mathbf{x}}_i^l = {W^l}^T \mathbf{y}_i$.

Thereafter, $\hat{\mathbf{x}}_i$ can be written as the $\hat{\mathbf{x}}_i^l$ which appears most number of times for $l \in [1, L]$.

The proposed method involves training and testing of the weight vector W_i where i denotes the iteration till which the weights are trained.

Initialize step-size η; $W^l = 0_{N \times \frac{D}{L}} \forall l \in [1, L]$; $\mathbf{y}_i^l \ \forall i, \forall l \in [1, L]$
for l from 1 to L **do**
  for i from 1 to Num_training **do**
    $\hat{\mathbf{x}}_i^l = \text{qamdemod}(W^l \cdot z_{\Omega^l}(\mathbf{y}_i^l))$
    $W^l = W^l + \eta(\mathbf{x}_\text{i} - W^l z_{\Omega^l}(\mathbf{y}_i^l)) z_{\Omega^l}^T(\mathbf{y}_i^l)$
  end
  Here we can calculate the BER per iteration during training
end
while $i > 0$ **do**
  for l from 1 to L **do**
    $\hat{\mathbf{x}}_i^l = \text{qamdemod}(W^l \cdot z_{\Omega^l}(\mathbf{y}_i^l))$
  end
  for u from 1 to M **do**
    $\hat{\mathbf{x}}_i(u) = \text{mode}\ \{\, \hat{\mathbf{x}}_i^l(u) \quad \forall l \,\}$
  end
end

Algorithm 1: Training Algorithm for Proposed KLMS based detector with low space complexity

After the training, we have used the trained model to determine recieved signal.

5.4 Simulations

The problem of RKHS based parallel non-linear channel detection of massive-MIMO was extended as an equivalent approximate parallel linear channel detection for massive-MIMO using RFFs.

For the given input from the reciever antennas, the input vector was divided into L blocks of size $\frac{N}{L} \times 1$. Each node provides a soft estimate of symobls upon convergence, which are fused by majority voting after demodulation.

Additionally, kernel function was changed to a more stable and rotation independent function because the kernel approximation using RFF is defined on positive definite and shift invariant kernels, however, the initial kernel was rotation dependent, thus the new kernel is easily approximated using the method proposed in [3].

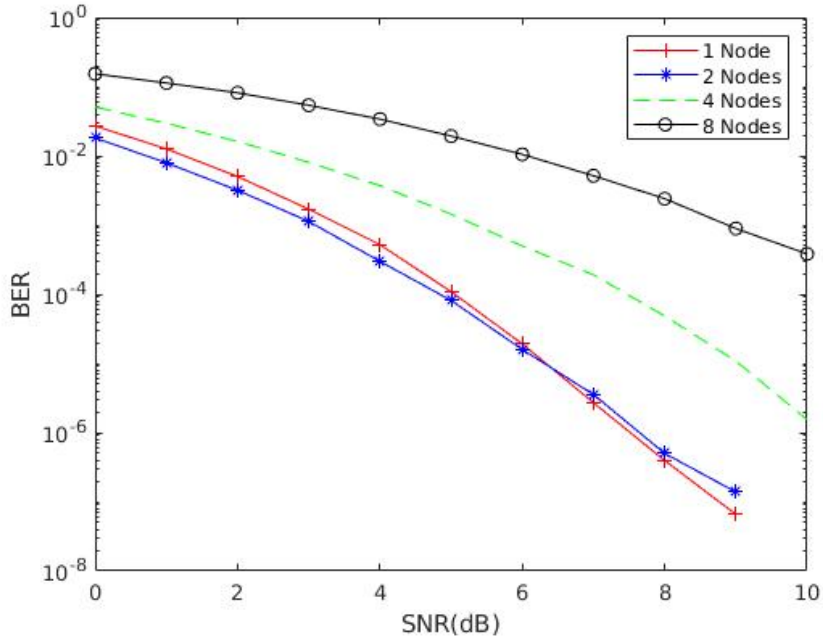

Figure 5: Y axis: BER, X axis: SNR(in dB)
Number of RFF D = 200

The number of Random Fourier Features used in the experiment were 200 for a kernel width of 1. The algorithm was trained for 5000 samples and then tested for 10^5 samples. The BER vs SNR is plotted in Figure **??** for L=1,2,4,8 and it can be observed that the proposed detector can achieve reasonable BER performance without a dictionary using RFFs for memoryless channel.

In figure **??** the BER vs SNR is plotted. The simulation was ran for 4 users and 160 antennas and for a memoryless channel. Four different scenarios

considered are: a)$L = 1$, b)$L = 2$, c)$L = 4$ and d)$L = 8$. The computational cost comes out to be $O(D)$, where D is the number of RFFs used; as there is no need to evaluate Gram Matrix which takes $M \times M$ which is avoided by using RFF approximation of the kernel function. Also, the space complexity of the algorithm is $O(D)$ as the dictionary is not used, which makes the proposed methodology practically viable.

It is also observed that for lower number of nodes, $L << N$, the BER is considerably low as compared to when $\frac{N}{L} = 20$.

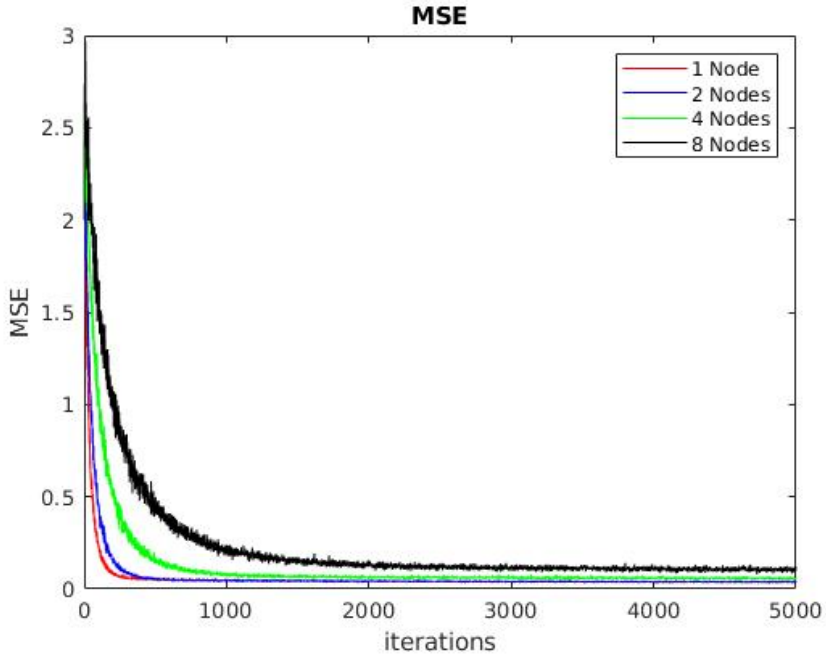

Figure 6: Y axis: MSE, X axis: Iterations
Number of RFF D = 200

In figure **??**, the MSE over time for L=1,2,4,8 are plotted. It is observed that for higher number of nodes, computational complexity per node reduces; however the BER and MSE increases due to reduction in input - regression dimensions available to each node.

A Appendix A

In this chapter, Reproducing Kernel Spaces will be briefed from ground up as done in [19]

A.1 Vector space

Defined over **Ordered field** F as V as $(V,F,\oplus,\otimes,0_v)$. The vectors in vector space may be added together and multiplied by a number called scalar. The operations are defined as:

- Addition($\oplus$) is defined as $\oplus:\mathbb{V}\times\mathbb{V}\to\mathbb{V}$ wherein the resultant vector is called sum of vectors and represented by $v\oplus w$ where v and w are the input vectors.
- Scalar Multiplication($\otimes$) is defined as $\otimes:\mathbb{C}\times\mathbb{V}\to\mathbb{V}$ wherein the resultant vector is represented by $a\otimes v$ where a is a complex scalar and v is a vector.

The operations must satisfy following axioms:

- Associativity of addition
  $u+(v+w)=(u+v)+w$
- Commutativity of addition
  $u+v=v+u$
- Identity element of addition
  $\mathbf{0}\in\mathbb{V}$ called a zero vector such that $v+\mathbf{0}=v$
- Inverse elements of addition
  For every $v\in\mathbb{V}$ there exist a vector $-v\in\mathbb{V}$ such that $v+(-v)=\mathbf{0}$
- Compatibility of scalar multiplication with field multiplication
  $a(bv)=(ab)v \quad v\in\mathbb{V},\ a,b\in\mathbb{C}$
- Identity element of scalar multiplication
  $1\otimes v=v$ where 1 represents multiplicative identity in $\mathbb{V}$
- Distributivity of scalar multiplication with respect to vector addition
  $a(u+v)=au+av$
- Distributivity of scalar multiplication with respect to field addition
  $(a+b)v=av+bv$

$\oplus$ for sum of two vectors $\ominus$ for additive inverse $\otimes$ for scalar multiplication. Here, $u,v,w\in\mathbb{V}$ and $a,b\in\mathbb{C}$

A.2 Banach Space

Norm is defined as $||X||_{(.)}$ where $(.)$ denotes the corresponding Banach space.

Every normed space $\mathbb{X}$ can be isometrically embedded in a Banach space. More precisely, for every normed space $\mathbb{X}$, there exist a Banach space $\mathbb{Y}$ and a mapping $T:\mathbb{X}\to\mathbb{Y}$ such that T is an isometric mapping and $T(\mathbb{X})$ is dense in $\mathbb{Y}$. If $\mathbb{Z}$ is another Banach space such that there is an isometric isomorphism from $\mathbb{X}$ onto a dense subset of $\mathbb{Z}$, then $\mathbb{Z}$ is isometrically isomorphic to $\mathbb{Y}$.

$$||\langle x_i \rangle_{i=1}^{n}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

For infinite sequences, the norm must not diverge for given p,
Thus, a space l_p (pronounced *"little ell p"*) as:

$$l_p = \left\{ \langle x_i \rangle_{i=0}^{\infty} : \sum_{i=0}^{\infty} |x_i|^p < \infty \right\}$$

So, we define norm on l_p as:

$$||\langle x_i \rangle_{i=0}^{\infty}||_{l_p} = \left(\sum_{i=0}^{\infty} |x_i|^p \right)^{\frac{1}{p}}$$

For norm on function spaces,

$$||f||_{sup} = \sup_{x \in X} |f(x)|$$

This is called natural norm or uniform norm or sup norm. i.e. this is the highest (supremum) value that f takes on all of X. this is analogous to the ∞ norm defined for sequences.

Further, we define the notion of an L_p ("ell p") iver function from $\mathbb{R}^n$ to $\mathbb{R}$.

$$L_p = \left\{ (f : \mathbb{R}^n \to \mathbb{R}) : \int_{-\infty}^{\infty} |f^p(x)| dx < \infty \right\}$$

we define a norm on L_p by:

$$||f||_{L_p} = \left(\int_{-\infty}^{\infty} |f^p(x)| dx \right)^{\frac{1}{p}}$$

A.3 Hilbert Space

A Banach space with inner product $\langle x, y \rangle_{\mathcal{H}}$ or $\langle x, y \rangle$ also represented as $x.y$

$$x.y = \sum_i x_i y_i$$

For infinite dimensional space, $\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x)dx$
Properties:

- Symmetry- commutative over inner product (or say conjugate inverse commutative over inner product)
- Linear over first argument and conjugate linear over second argument $\langle x, ay_1 + by_2 \rangle = \bar{a}\langle x, y_1 \rangle + \bar{b}\langle x, y_2 \rangle$ and $\langle ax_1 + bx_2, y \rangle = a\langle x_1, y \rangle + b\langle x_2, y \rangle$

$$\langle x, ay_1 + by_2 \rangle = \bar{a}\langle x, y_1 \rangle + \bar{b}\langle x, y_2 \rangle$$

and

$$\langle ax_1 + bx_2, y \rangle = a\langle x_1, y \rangle + b\langle x_2, y \rangle$$

- Positive definiteness $\langle u, u \rangle \geq 0$

$$\langle u, u \rangle \geq 0$$

  here equality holds only when $u = 0$

Given complete vector space $\mathbb{V}$ with dot product $\langle \cdot, \cdot \rangle_{\mathbb{V}}$, we can easily define a norm on $\mathbb{V}$ by $||u||_{\mathbb{V}} = \sqrt{\langle u, u \rangle}$

B Appendix B Transient Analysis of Diffusion

In this section, the transient analysis of Diffusion Techniques will be reviewed as done in [20].

B.1 Mean transient Analysis LMS

$$\widetilde{w}^i \triangleq w^{(o)} - w^i$$

Now as $Gw^{(o)} = w^{(o)}$, we get

$$\widetilde{w}^i = Gw^{(o)} - Gw^{i-1} - DU_i^* \left(U_i w^{(o)} + v_i - U_i G w^{i-1} \right) = (I_{NM} - DU_i^* U_i) G\widetilde{w}^{i-1} - DU_i^* v_i$$

Assuming noise to gaussian with 0 mean,

$$E\widetilde{w}^i = (I_{NM} - DR_u) G E\widetilde{w}^{i-1}$$

where $R_u = diag\{R_{u,1}, \ldots, R_{u,N}\}$ is a block diagonal and $R_{u,k} = Eu_{k,i}^* u_{k,i}$

So, for stablity in mean, $|\lambda(BG)| < 1$ where $B = (I_{NM} - DR_u)$

Now considering Non cooperation,

$$E\widetilde{w}^i = BE\widetilde{w}^{i-1} \therefore ||BG||_2 \leq ||B||_2 \cdot ||G||_2$$

And as R_u is B is Hermitian.

And as $G = A \otimes I_M$,

$$\therefore |\lambda_{max}(BG)| \leq ||A||_2 \cdot |\lambda_{max}(B)|$$

As $||A||_2 \leq 1$ (1 for double stochastic matrix)

$$\therefore |\lambda(BG)| \leq |\lambda(B)|$$

B.2 Mean Square Transient Analysis

B.2.1 Weighted Energy and Variance Relations

$$e_k(i) = d_k(i) - u_{k,i}\psi_k^{i-1}$$
$$e_i = d_i - U_i G w^{i-1} = U_i G\widetilde{w}^{i-1} + v_i \triangleq e_{a,i}^G + v_i$$
where
$$e_{a,i}^G = U_i G\widetilde{w}^{i-1}$$
$$\text{and}\widetilde{w}^{i} = G\widetilde{w}^{i-1} - DU_i^* e_i$$

Taking different cases for ariori and posteriori weighted estimation errors:

$$e_{a,i}^{D\Sigma G} \triangleq U_i D\Sigma G\widetilde{w}^{i-1}\text{and}e_{p,i}^{D\Sigma} \triangleq U_i D\Sigma\widetilde{w}^{i}$$

For some arbitrary matrix $\Sigma_{NM\times NM} \geq 0$. The freedom in selecting Σ will enable us later to characterize the MSD and EMSE performance of the network.

$$E||\widetilde{w}^{i}||_{\Sigma}^2 = E||\widetilde{w}^{i-1}||_{G^*\Sigma G}^2 - E(e_{a,i}^{D\Sigma G})^* e_{a,i}^G - E(e_{a,i}^G)^* e_{a,i}^{D\Sigma G} + E(e_{a,i}^G)^* U_i D\Sigma DU_i^* e_{a,i}^G + Ev_i^* U_i D\Sigma DU_i^* v_i$$
$$E||\widetilde{w}^{i}||_{\mathbf{\Sigma}'}^2 = E||\widetilde{w}^{i-1}||_{G^*\mathbf{\Sigma}'\mathbf{G}}^2 + Ev_i^* U_i D\Sigma DU_i^* v_i$$

here $\mathbf{\Sigma}'$ is:

$$\mathbf{\Sigma}' = G^*\Sigma G - G^*\Sigma DU_i^* U_i G - G^* U_i^* U_i D\Sigma G + G^* U_i^* U_i D\Sigma DU_i^* U_i G$$

As the weighting matrix $\mathbf{\Sigma}'$ is data dependent however, as such it is a random quantity. This makes analysis challenging so we make the assumption that U_i is independent of $\widetilde{w}^{i-1}$. In this way, the random weighting matrix $\mathbf{\Sigma}'$ can be replace by its mean value $\mathbf{\Sigma}' = E\Sigma'$

$$E||\widetilde{w}^{i}||_{\Sigma'}^2 = E||\widetilde{w}^{i-1}||_{G^*\Sigma' G}^2 + Ev_i^* U_i D\Sigma DU_i^* v_i$$
$$\Sigma' = G^*\Sigma G - G^*\Sigma DE(U_i^* U_i)G - G^* E(U_i^* U_i)D\Sigma G + G^* E(U_i^* U_i D\Sigma DU_i^* U_i)G$$

B.3 The Case of Gaussian Regressors

Doing the eigendecomposition of $R_u = T\Lambda T^*$, where $\Lambda = diag\{\Lambda_1, \ldots, \Lambda_N\}, \Lambda_k > 0$ are diagonal.

$$\overline{w}^i = T^*\widetilde{w}^{i}, \overline{U}_i = U_i T, \overline{G} = T^* G T$$
$$\overline{\Sigma} = T^*\Sigma T, \overline{\Sigma}' = T^*\Sigma' T, \overline{D} = T^* D T = D$$

Then,

$$E||\overline{w}^i||_{\overline{\Sigma}'}^2 = E||\overline{w}^{i-1}||_{G^*\overline{\Sigma}' G}^2 + Ev_i^*\overline{U_i} D\overline{\Sigma} D\overline{U_i}^* v_i$$
$$\overline{\Sigma}' = \overline{G}^*\overline{\Sigma G} - \overline{G}^*\overline{\Sigma} DE(\overline{U_i}^*\overline{U_i})\overline{G} - \overline{G}^* E(\overline{U_i}^*\overline{U_i})D\overline{\Sigma G} + \overline{G}^* E(\overline{U_i}^*\overline{U_i} D\overline{\Sigma} D\overline{U_i}^*\overline{U_i})\overline{G}$$

As several quantities have a block diagonal structure, we can exploit it. We will employ operation `bvec{}`, which converts a block matrix Σ into a single column vector σ in two steps as follows. Let Σ be an $NM \times NM$ block matrix $\Sigma = [\Sigma_{k,l}]$ where $\Sigma_{k,l}$ is a $M \times M$ matrix. First the block columns are stacked on top of each other to give a $N^2M \times M$ matrix

$$\begin{gathered}\Sigma_l = \text{col}\{\Sigma_{1l}, \Sigma_{2l}, \ldots, \Sigma_{Nl}\}, l = 1, 2, \ldots, N \\ \Sigma^c = \text{col}\{\Sigma_1, \Sigma_2, \ldots, \Sigma_N\} \\ \sigma_l = \text{col}\{\sigma_{1l}, \sigma_{2l}, \ldots, \sigma_{Nl}\} \text{ with } \sigma_{kl} = \text{vec}\{\Sigma_{kl}\} \\ \sigma = \text{col}\{\sigma_1, \sigma_2, \ldots, \sigma_N\}\end{gathered}$$

We write $\sigma = \text{bvec}\{\Sigma\}$ to denote conversion of Σ into a single column. We also write $\Sigma = \text{bvec}\{\sigma\}$ to recover the original block matrix form of the column vector σ.

$$\text{A}\Sigma\text{B} = (B \odot A^T)\sigma$$

$$\overline{\Sigma}' = \overline{G}^* \overline{\Sigma} \overline{G} - \overline{G}^* \overline{\Sigma} D E(\overline{U_i}^* \overline{U_i}) \overline{G} - \overline{G}^* E(\overline{U_i}^* \overline{U_i}) D \overline{\Sigma} \overline{G} + \overline{G}^* E(\overline{U_i}^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* \overline{U_i}) \overline{G}$$

$$E\overline{U_i}^* \overline{U_i} = \Lambda$$

$$\text{bvec}\{\overline{G}^* \overline{\Sigma} D \Lambda \overline{G}\} = \left(\overline{G} \odot \overline{(G)}^*\right) \text{bvec}\{I_{NM} \overline{\Sigma} D \Lambda\} = \left(\overline{G} \odot \overline{G}^*\right) (I_{NM} \odot D\Lambda)\overline{\sigma}$$

$$\text{bvec}\{\overline{G}^* \Lambda D \overline{\Sigma} \overline{G}\} = \left(\overline{G} \odot \overline{G}^*\right) (D\Lambda \odot I_{NM})\overline{\sigma}$$

$$E v_i^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* v_i = \text{Tr}\left\{\Lambda_v E \overline{U_i} D \overline{\Sigma} D \overline{U_i}^*\right\}$$

Where $\Lambda_v > 0$ is a diagonal matrix given by

$$\Lambda_v = \text{diag}\{\sigma_{v,1}^2, \sigma_{v,2}^2, \ldots, \sigma_{v,N}^2\}$$

The entries of $E\overline{U_i} D \overline{\Sigma} D \overline{U_i}^*$ is given by

$$E\overline{U_i} D \overline{\Sigma} D \overline{U_i}^* = \text{diag}\left\{\mu_k^2 \text{Tr}\left(\Lambda_k \overline{\Sigma}_{kk}\right)\right\} = \text{diag}\{\mu_k^2 \lambda_k^T \overline{\sigma}_{kk}\}$$

where $\lambda_k = \text{vec}\{\Lambda_k\}$ and $\overline{\sigma}_k l = \text{vec}\{\overline{\Sigma}\}$

$$E v_i^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* v_i = b^T \overline{\sigma}$$

with $b = \text{bvec}\{R_v D^2 \Lambda\}$, $R_v = \Lambda_v \odot I_m$

$$\text{bvec}\left\{\overline{G}^* E(\overline{U_i}^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* \overline{U_i}) \overline{G}\right\} = \left(\overline{G} \odot \overline{G}^{*T}\right) \cdot \text{bvec}\left\{E(\overline{U_i}^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* \overline{U_i})\right\}$$

Now as both $\overline{U_i}^* \overline{U_i}$ and D are block diagonal, so that

$$E(\overline{U_i}^* \overline{U_i} D \overline{\Sigma} D \overline{U_i}^* \overline{U_i}) = D E(\overline{U_i}^* \overline{U_i \Sigma U_i}^* \overline{U_i}) D$$

Which gives:

$$\text{bvec}\left\{\overline{G}^{*}E(\overline{U_i}^{*}\overline{U_i}D\overline{\Sigma}D\overline{U_i}^{*}\overline{U_i})\overline{G}\right\} = \left(\overline{G}\odot\overline{G}^{*^T}\right)(D\odot D)\text{bvec}\{A\}$$

where $A \triangleq E(\overline{U_i}^{*}\overline{U_i\Sigma U_i}^{*}\overline{U_i})$

The $M\times M$ kl-block of A is given by

$$A_{kl} = E\overline{u_{k,i}}^{*}\overline{u_{k,i}}\overline{\Sigma}\overline{u_{l,i}}^{*}\overline{u_{l,i}} \tag{3}$$

$$= \begin{cases}\Lambda_k\text{Tr}(\Lambda_k\overline{\Sigma_{kk}}) + \gamma\Lambda_k\overline{\Sigma_{kk}}\Lambda_k, k=l\\ \Lambda_k\overline{\Sigma_{kl}}\Lambda_l, k\neq l\end{cases} \tag{4}$$

$$\tag{5}$$

where $\gamma = 1$ for complex data and $\gamma = 2$ for real data. Now express A as

$$A = [A_1A_2\dots A_N]\text{and,}a \triangleq \text{bvec}\{A\} = \mathcal{A}\overline{\sigma}$$

where $\mathcal{A}_{\updownarrow}$ is a diagonal matrix

Thus, we can summarize the results as

$$E||\overline{w}^i||^2_{\overline{\sigma}} = E||\overline{w}^{i-1}||^2_{\overline{F}\overline{\sigma}} + b^t\overline{\sigma}$$
$$\overline{F} = (\overline{G}\odot\overline{G}^{*^T})[I_{N^2M^2} - (I_{NM}\odot\Lambda D) - (\Lambda D\odot I_{NM}) + (D\odot D)\mathcal{A}]\overline{\sigma}$$

We can find the learning rate by the recursion

$$E||\overline{w}^i||^2_{\overline{\sigma}} = E||\overline{w}^{i-1}||^2_{\overline{\sigma}} + b^t\overline{F}^i\overline{\sigma} - ||w^{(o)}||^2_{\overline{F}^i(I-\overline{F})\overline{\sigma}}$$

$$a_{kl} = \text{vec}\{A_{kl}\} = \begin{cases}(\lambda_k\lambda_k^T + \gamma\Lambda_k\otimes\Lambda_k)\overline{\sigma}_{kk}, k=l\\ (\Lambda_k\otimes\Lambda_l)\overline{\sigma}_{kl}, k\neq l\end{cases}$$

$$a_l = \text{col}\{(\Lambda_1\otimes\Lambda_l)\overline{\sigma}_{1l},(\Lambda_2\otimes\Lambda_l)\overline{\sigma}_{2l},\dots,(\lambda_l\lambda_l^T+\gamma\Lambda_l\otimes\Lambda_l)\overline{\sigma}_{ll},\dots,(\Lambda_N\otimes\Lambda_l)\overline{\sigma}_{Nl}\} \triangleq \mathcal{A}_k\overline{\sigma}_l$$

where

$$\mathcal{A}_l = \text{diag}\{(\Lambda_1\otimes\Lambda_l),(\Lambda_2\otimes\Lambda_l),\dots,(\lambda_l\lambda_l^T+\gamma\Lambda_l\otimes\Lambda_l),\dots,(\Lambda_N\otimes\Lambda_l)\}$$
$$\text{bvec}\{A\} = \text{col}\{\mathcal{A}_1\overline{\sigma}_1,\mathcal{A}_2\overline{\sigma}_2,\dots,\mathcal{A}_N\overline{\sigma}_N\} = \mathcal{A}\overline{\sigma}$$
$$\mathcal{A} = \text{diag}\{\mathcal{A}_1,\mathcal{A}_2,\dots,\mathcal{A}_N\}$$

References

[1] R. Mitra and V. Bhatia, "Kernel-based parallel multi-user detector for massive-MIMO," *Computers & Electrical Engineering*, vol. 65, pp. 543–553, 2018.

[2] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[3] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in neural information processing systems*, 2008, pp. 1177–1184.

[4] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2009.

[5] G. Azarnia and M. A. Tinati, "Steady-state analysis of the deficient length incremental lms adaptive networks with noisy links," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 1, pp. 153–162, 2015.

[6] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[7] J. Kiefer, J. Wolfowitz, *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[8] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[9] P. Drineas and M. W. Mahoney, "Approximating a gram matrix for improved kernel-based learning," in *International Conference on Computational Learning Theory*, Springer, 2005, pp. 323–337.

[10] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, "Nyström method vs random fourier features: A theoretical and empirical comparison," in *Advances in neural information processing systems*, 2012, pp. 476–484.

[11] W. Rudin, *Fourier analysis on groups*. Wiley Online Library, 1962, vol. 121967.

[12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[14] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Departmental Papers (ESE)*, p. 29, 2003.

[15] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005, ISBN: 0-5218-4527-0.

[16] Z. Gao, C. Hu, L. Dai, and Z. Wang, "Channel estimation for millimeter-wave massive MIMO with hybrid precoding over frequency-selective fading channels," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1259–1262, 2016.

[17] K. M. Gharaibeh, *Nonlinear distortion in wireless systems: Modeling and simulation with MATLAB*. John Wiley & Sons, 2011.

[18] S. L. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *IEEE Communications letters*, vol. 5, no. 9, pp. 369–371, 2001.

[19] H. Daumé III, "From zero to reproducing kernel hilbert spaces in twelve pages or less," *University of Maryland*, 2004.

[20] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.