Machine Learning Techniques in Non-Linear Receivers for Interference Mitigation

A Project Report

submitted by

MILIND RAO

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

June 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **Machine Learning Techniques in Non-Linear Receivers for Interference Mitigation**, submitted by **Milind Rao**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. K. Giridhar
Research Guide,
Dept. of Electrical Engineering,
IIT-Madras, 600 036.

Place: Chennai

Date: 6th June 2013

ACKNOWLEDGEMENTS

I would first like to place on record my immense gratitude to my project advisor Prof. Giridhar. Apart from aiding me at key points in the project, he gave me tremendous flexibility and freedom in my work. The perspectives he lent always kept the bigger picture in sight for me.

I would like to thank Prof. B. Ravindran for his advice on clustering algorithms and also both the Vaishnavis who worked on similar problem statements for lending their insight. The faculty and support staff have been very helpful with every request I had of them.

My family and friends have been very supportive and understanding throughout this project and I cannot be more grateful to them. Both for the encouragement I received while working on the project and for making these last four years very memorable.

ABSTRACT

KEYWORDS: Interference Mitigation, Gaussian Mixture Models, Expectation-Maximization, Hierarchical clustering, Density based Clustering, Spectral Clustering, MIMO-OFDM

With universal frequency reuse systems being ubiquitously adapted to allow more users per unit area, co-channel interference mitigation has emerged as real necessity. The interference profile and signal to interference ratio impose strong constraints on performance. In this thesis, we focus on receiver techniques that are interference aware.

Linear Minimum Mean Square Error (LMMSE) is popularly deployed but is not effective in scenarios with large interference as the interference plus noise profile is not Gaussian. Other linear approaches such as the Minimum Bit Error Rate approaches outperform LMMSE with non-Gaussianity but performance is seen to floor with heterogeneous interferers. The interference and noise can be modelled as a Gaussian Mixture Model (GMM). The algorithm that is used to initialize a GMM requires the number of components parameter which we can estimate only through clustering approaches.

Clustering or finding groups of points which are 'closer' to each other may be achieved through a variety of approaches. Three methods have been explored in this thesis - Hierarchical clustering algorithms with clustering metrics, density based clustering approaches and Spectral clustering. Alternate less computationally expensive initializations of the GMM from the clustering are studied.

Finally, the non-linear receiver is extended to the multiple antenna case and different methods of combining data from the two antennas or combining the results from two clustering approaches are seen significantly improving the performance of MIMO-OFDM systems.

TABLE OF CONTENTS

LIST OF FIGURES

List of Algorithms

ABBREVIATIONS

3GPP	Third Generation Partnership Project
CCI	Co-channel Interference
SNR	Signal to Noise Ratio
SINR	Signal to Interference plus Noise Ratio
MRC	Maximal Ratio Combining
LMMSE	Linear Minimum Mean Square Error
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MSE	Mean Square Error
CSI	Channel State Information
OFDM	Orthogonal Frequency Division Multiplexing
QPSK	Quadrature Phase Shift keying
QAM	Quadrature Amplitude Modulation
KDE	Kernel Density Estimate
GMM	Gaussian Mixture Model
EM	Expectation Maximization
BER	Bit Error Rate
MBER	Minimum Bit Error Rate
JMD	Joint Minimum Distance Detection
SSE	Sum of Squared Error
PCCA	Perron Cluster Cluster Analysis
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
OPTICS	Ordering Points To Identify the Clustering Structure

NOTATION

The following are the notations used in the thesis. Lower case alphabets with normal font indicate scalars while old indicates vectors. Bold face alphabets in upper case denote Matrices.

x_i	ith element of vector $\mathbf{x}$
$X_{i,j}$	ith row, jth column of matrix $\mathbf{X}$
$\mathbf{x}^\intercal$	Transpose of the vector
$\mathbf{X}^{-1}$	Inverse of the matrix
$\|S\|$	Number of elements in set S
$\|\mathbf{x}\|_2$	Euclidean 2-norm
$\Re(.)$	Real part
$\Im(.)$	Imaginary part
$\mathbb{E}[.]$	Expectation operator
$\mathbf{I}_n$	Identity matrix of size $n \times n$
$\mathcal{CN}(\mu, \sigma^2)$	Circular symmetric Complex Gaussian with mean μ and variance $\sigma^(2)$
C_i	ith cluster
$C(i)$	Cluster containing the ith datapoint
$d^2(x_i, x_j)$	Squared Euclidean distance between ith and jth datapoints
$N_\epsilon(p)$	Points in the ϵ neighbourhood of point p

CHAPTER 1

Introduction

There is an ever increasing need in wireless communication standards to fit in more users per unit area in the system. Hence, emerging technologies deploy universal frequency reuse. As more users demand higher data rates, it is becomes imperative that multiple links sharing common resources in frequency and time are active simultaneously. This leads to the problem of *co-channel interference* (CCI) and the signal to interference ratio becomes a more limiting factor in achieving better throughput performance or lower *Bit Error Rates* (BER) than simply the signal to noise ratio. Increasing transmit power will not reduce the *signal to interference and noise ratio* (SINR) and this necessitates the need for better interference mitigation techniques and architectures. These techniques have the potential to significantly improve average throughput at various levels of the entire system.

1.1 The System Model

We model a scenario where there are several single antenna transmitters $T_1, \ldots, T_n$ where n is not known and a receiver comprising of l antennas $R_1, \ldots, R_l$. This is shown in Fig. 1.1. T_1 transmits the desired message and the other transmitters are interferers with an unknown modulation alphabet.

The communication system employed is a $10MHz$ Orthogonal Frequency Division Multiplexing (OFDM) multi-carrier system (Cho *et al.*, 2010). The guard interval is adequate to eliminate out-of-band-radiation. The cyclic prefix is modelled to be larger than the channel delay spread mitigating intersymbol interference (ISI). There are 1024 sub-carriers in the OFDM system out of which 600 are usable, the rest comprising the guard interval. A single *Physical Resource Block* (PRB) is 12 sub-carriers wide in frequency and 14 symbols long in time. Doppler effect is not included in the model. Thus, a scenario where the channel is static in time over several resource blocks is modelled. If $X_{i,l}[k]$, $Y_l[k]$, $H_{i,l}[k]$ and $Z_l[k]$ denote the kth subcarrier frequency components of the

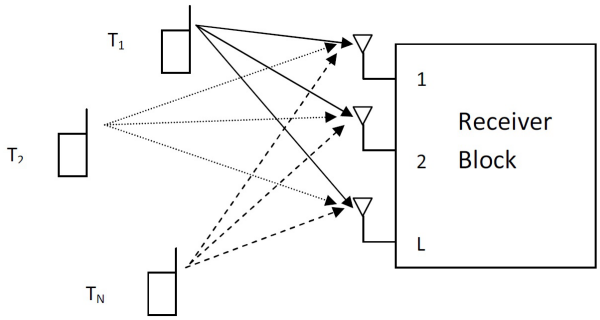

Figure 1.1: The system model - one desired transmitter T_1, other interferers $T_2, \ldots, T_n$ transmitting to one receiver with l antennas (Vaishnavi, 2012).

lth transmitted symbol , received symbol of the ith user, channel frequency response and noise in the frequency domain,

$$Y_l[k] = H_{1,l}[k]X_{1,l}[k] + \sum_{i=2}^{n} H_{i,l}[k]X_{2,l}[k] + Z_l[k], \quad (1.1)$$

$$= H_{1,l}[k]X_{1,l}[k] + Z'_l[k]. \quad (1.2)$$

Here the noise is modelled to be Gaussian circular noise with zero mean. $Z'_l[k]$ is the net interference plus noise.

The channel is modelled based on the Power Delay Profiles (PDP) of the ITU-R Pedestrian outdoor channel A (Ped A), Pedestrian outdoor channel B (Ped B) and vehicular test environment channel A (Veh A) (Jain, 2007). Fig. 1.2 shows a typical channel realization in all three models. The transmitters use a single PRB or a few sub-carriers.

1.2 Interference Mitigation Techniques

CCI interference mitigation techniques are mainly of two types depending on whether the interference cancellation takes place at the transmitter or receiver (Vaishnavi, 2012). Although this thesis is concerned with receiver based techniques, a short introduction to transmitter based techniques is given.

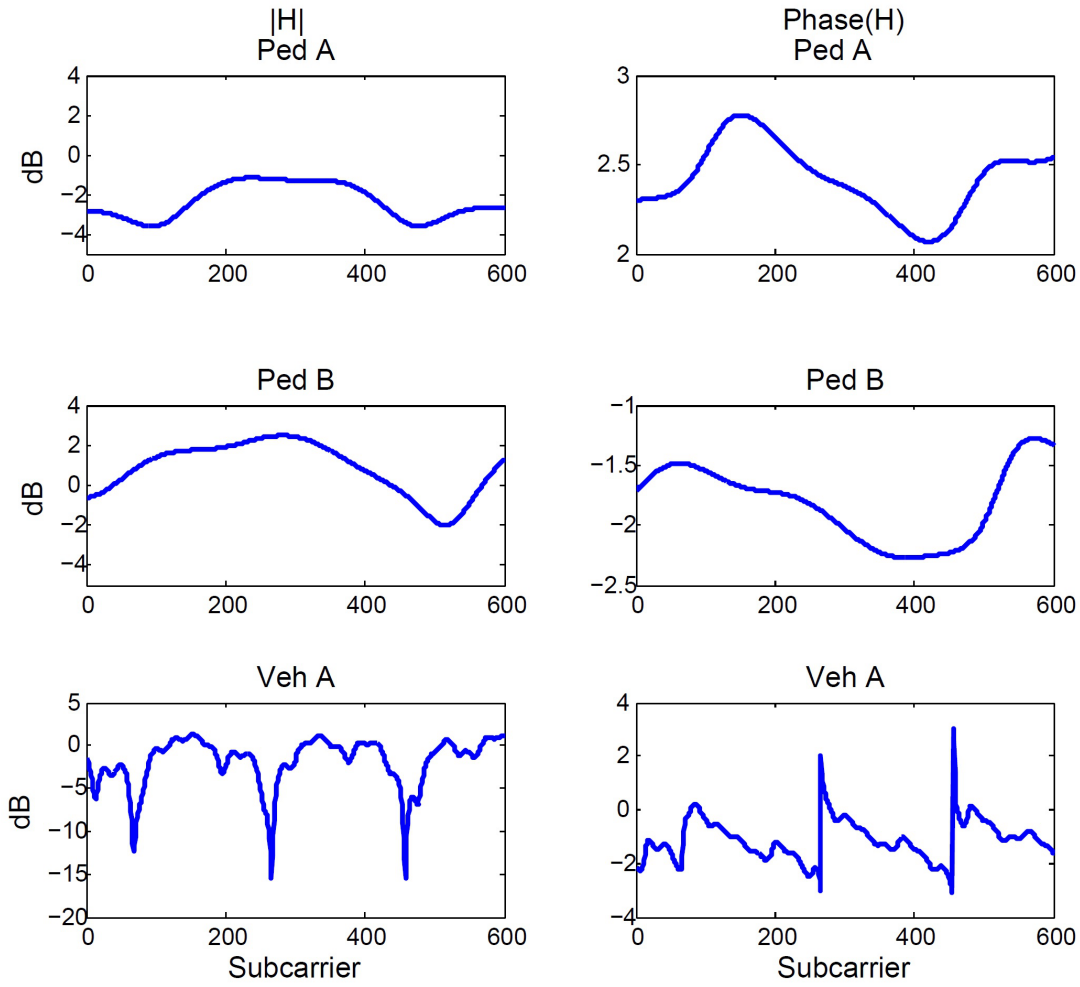

Figure 1.2: Channels modelled - Ped A, Ped B and Veh A.

1.2.1 Transmitter Techniques

Beamforming or precoding are the signal processes involved in interference mitigation at the transmitter (Tse and Viswanath, 2005). The transmit signals are aligned and added in phase to maximize received SNR by allotting more power to the transmit antenna with better gain. Maximal Ratio Transmission and Equal Gain Transmission are such techniques. In Multiple Input Multiple Output (MIMO) systems, interference alignment can use the spacial dimension such that users coordinate and have the interference signal lie in a reduced dimensional sub-space (Cadambe and Jafar, 2008). Other techniques involve using the position of the users for Dynamic Channel Allocation (Cho *et al.*, 2010). These techniques require the complete knowledge of the Channel State Information (CSI) at the transmitter which presents a large overhead in transmission of these values from the receiver to the transmitter for beamforming.

1.2.2 Receiver Techniques

The traditional techniques used to detect the signal in the presence of interference are now described:

1. **Maximal Ratio Combining (MRC)**: Diversity combining technique in which the signal at the receive antenna is weighted proportional to the amplitude of the desired signal. This ensures that the SNR at the receiver is maximum but is ineffective with interference. It is not an interference cancellation technique.
2. **Decorrelation receiver**: With l antennae in this method, $l - 1$ interferers can be cancelled. The signal is projected onto the subspace perpendicular to the interferer's channels. This method leads to loss in SNR and is optimal only at high SNR.
3. **Linear Minimum Mean Square Error receiver (LMMSE)** : The signal at the receive antenna are weighted such that the Mean Square Error (MSE) between the actual symbol and the estimate is minimized. Effectively, MRC is done when the desired signal is of low SNR and zero-forcing or decorrelation reception done at high SNR.

These methods require complete CSI from all interferers which is not feasible. Minimizing MSE minimizes BER only when the interference plus noise signal space is Gaussian which is not the case with strong interferers. Vaishnavi (2012) showed that Linear Minimum BER (LMBER) methods significantly outperforms LMMSE in the

presence of non-Gaussian noise. LMBER involves estimating the probability distribution of the interference plus noise using Kernel Density Estimates and choosing the beamforming weights such that the BER is minimized. It was shown that LMBER techniques do not work as well with heterogeneous interferers.

1.3 Gaussian Mixture Models and Expectation Maximization

1.3.1 Introduction

In this section we show that the conditional pdf of the interference and noise is not Gaussian and may be modelled by a mixture of Gaussians. We then introduce the Maximum Likelihood non-linear detector that maximizes the probability of a correct decision. When the conditional pdf is not Gaussian, the minimum distance measure does not translate to minimum error rate. The expectation maximization and the k-means algorithms are then introduced and the problems and parameters required by the EM algorithm are explained.

1.3.2 Gaussian Mixture Models

In this thesis, we mainly use the conditional pdf of the interference and noise. This data can be obtained from pilot signals where the desired signal is known after estimating the CSI of the desired user. It may also be obtained in a bootstrapping procedure. i.e. by estimating the desired signal using methods described earlier such as LMMSE to generate the pdf of the interference and noise.

Consider a system with one receive antenna and an interferer. Let the parameter vector $\theta = [h_1, h_2, \chi_1, \chi_2]$ be known to the receiver where h_1 and h_2 are the channels of the user and interferer and χ_1 and χ_2 be the modulating alphabet of the user and interferer.

$$\begin{aligned} y &= h_1 x_1 + h_2 x_2 + n, \\ &= h_1 x_1 + \tilde{n}. \end{aligned}$$

where n is the noise and $\tilde{n}$ is the interference and noise. By jointly detecting x_1 and x_2, we get,

$$(\hat{x}_1, \hat{x}_2) = \arg\max_{\chi_1 \times \chi_2} p_{y/\theta}(y/\theta).$$

The conditional pdf $p_{y/\theta}(y/\theta)$ is given by,

$$p_{y/\theta}(y/\theta) = \frac{1}{\pi\sigma_n^2} \exp\left[\frac{-\left(y - \sum_{i=1}^{2} h_i x_i\right)^2}{\sigma_n^2}\right].$$

We can now obtain the estimate for x_1 by,

$$\hat{x}_1 = \arg\max_{x_1 \in \chi_1} \sum_{x_2 \in \chi_2} p_{y/\theta}(y/\theta) p_{x_2}(x_2). \tag{1.3}$$

The conditional pdf of the interference and noise is seen to be a Gaussian Mixture Model by convolving the $p_n(n)$ and $p_{x_2}(x_2)$ as,

$$p(\tilde{n}) = \frac{1}{|\chi_2|\pi\sigma_n^2} \sum_{x_2 \in \chi_2} \exp\left[\frac{-\left(y - \sum_{i=1}^{2} h_i x_i\right)^2}{\sigma_n^2}\right].$$

Let $\theta_1 = [h_1, \chi_1]$ be the parameters of the desired user. Clearly $p_{y/\theta_1}(y/\theta_1) = p(\tilde{n})$. The estimate of x_1 by the Maximum-Likelihood rule is given by,

$$\begin{aligned} \hat{x}_1 &= \arg\max_{x_1 \in \chi_1} p_{y/\theta_1}(y/\theta_1) \\ &= \arg\max_{x_1 \in \chi_1} \sum_{x_2 \in \chi_2} p_{y/\theta}(y/\theta) p_{x_2}(x_2). \end{aligned} \tag{1.4}$$

This can be extended to the case of multiple interferers. Thus the pdf of the interference and noise is a GMM. ML detection on GMM is equivalent to performing joint detection given that we know the parameters as is evident from equations (1.3) and (1.4).

1.3.3 Expectation Maximization Algorithm

The EM algorithm is an iterative procedure for finding ML (or maximum a posteriori) estimates of parameters in models. The EM algorithm can be used to estimate the parameters (mean, variance and fraction) of the individual components of the GMM (Bishop, 2006). The EM algorithm alternates between performing an expectation (E) step which computes the expectation of the log-likelihood evaluated with the current parameters and the maximization step (M) which recomputes the parameters to maximize the log-likelihood.

The GMM is given by:

$$p(\mathbf{x}) = \sum_{i=1}^{k} \omega_i \mathcal{CN}(\mathbf{x}|\mu_i, \mathbf{R}_i),$$

where ω_i, μ_i and $\mathbf{R}_i$ are the probabilities of component i and its mean and variance and k is the number of clusters.

The steps of the EM algorithm are:

1. **E-Step**: Let $\gamma_{n,i}$ represent the posterior probability that a point $\mathbf{x}_n$ came from component i. It also represent the the responsibility of component k is explaining the data point.

$$\gamma_{n,i} = \frac{\omega_i \mathcal{CN}(\mathbf{x}_n|\mu_i, \mathbf{R}_i)}{\sum_j \omega_j \mathcal{CN}(\mathbf{x}_n|\mu_j, \mathbf{R}_j)}.$$

2. **M-Step**: The parameters of the GMM model are re-evaluated from maximizing the log-likelihood which is not shown here.

$$\begin{aligned} N_i &= \sum_n \gamma_{n,i} \\ \mu_i^* &= \frac{1}{N_i} \sum_n \gamma_{n,i} \mathbf{x}_n \\ \mathbf{R}_i^* &= \frac{1}{N_i} \sum_n \gamma_{n,i} (\mathbf{x}_n - \mu_i^*)^\intercal (\mathbf{x}_n - \mu_i^*) \\ \omega_i &= \frac{N_i}{N}, \end{aligned}$$

   where N is the total number of points.

We stop iterating if the log-likelihood $\ln p(\mathbf{X}|\mu, \mathbf{R}, \omega)$ is within a tolerance limit.

K-means algorithm

The EM algorithm is slow to converge and converges to local maxima. To speed up computation, it is often preceded by the k-means algorithm, an EM algorithm that performs clustering. The E step in the k-means algorithm assigns points to clusters such that the intra-cluster Sum of Squares Error (SSE) is minimized. This just translates to assigning it to the mean closest to it. The M step is the re-calculation of means. Iterations are stopped when the SSE is within a tolerance limit. The SSE is explained in more detail in chapter 2. The k-means algorithm itself can be speeded up with proper initializations of the means. The means may be randomly chosen amongst the points or by the k-means++ algorithm (Arthur and Vassilvitskii, 2007). In this algorithm:

1. A seed is first chosen at random and the distances of the other points from it are computed.
2. The next point is chosen from a distribution that is proportional to this distance.
3. These steps are now repeated till k seeds are obtained.

Thus, the kmeans++ algorithm attempts to seed the kmeans algorithm with points that are more distant from each other.

Both the kmeans algorithm and EM algorithm are very sensitive to initialized parameters.

Parameter initialization

The EM algorithm that fits a GMM on the data requires the number of components k. In cases where the interference channels are low, $k = 1$ is an optimal value. However if the interferers are transmitting larger modulation alphabets (eg. 16 QAM), we need to extract cluster-order from the data to determine the number of components. In this thesis we explore methods of detecting the number of clusters in a data set. Several of these methods also provide clustering and the initialization of the EM algorithm can be done with parameters obtained from this clustering. Fig. 1.3 shows the clustering with $k = 16$ done on a data set of 500 points from 3 QPSK interferers of magnitude 0db, -3dB and -6dB on the Ped A channel with an SNR of 10. This is henceforth referred to as the Ped A dataset. Accompanying the clustering is the pdf obtained by running the EM algorithm initialized by this run of k-means.

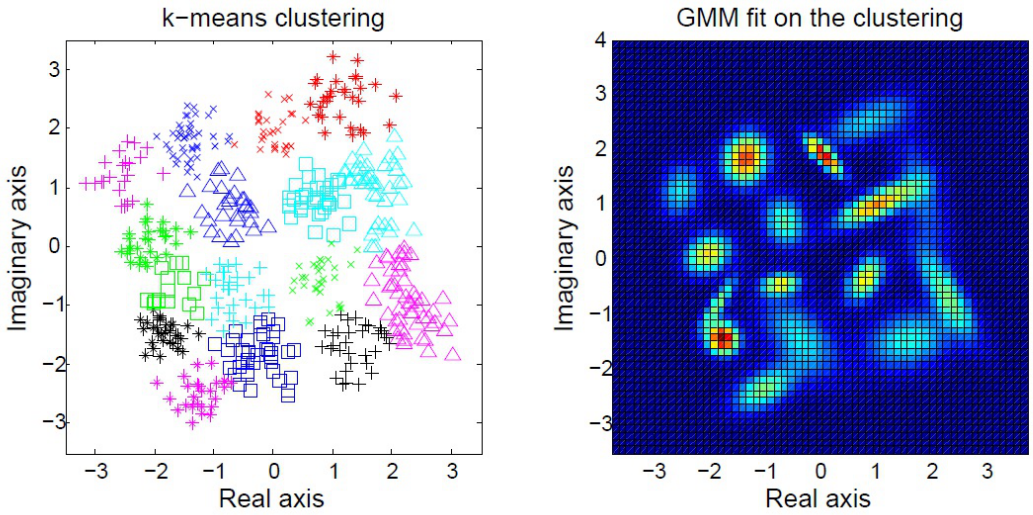

Figure 1.3: k-means clustering and EM to fit a GMM run on the Ped A dataset with $k = 16$.

1.4 Scope of the Thesis

LMMSE based interference mitigation techniques assume that the pdf of the interference plus noise is Gaussian which is usually not the case. There is interest in developing more effective receivers. We have introduced the non-linear Maximum Likelihood (ML) which uses the Gaussian Mixture Model (GMM) to model the interference and noise. The Expectation Maximization (EM) algorithm which is used to fit the GMM on the data is explored and shortcomings have been listed. The EM algorithm requires a parameter estimating the number of Gaussian mixtures. Chapters 2-4 look at techniques to estimate the cluster-order and also initialize the EM algorithm with the means and variances of individual clustering to hasten convergence. Chapter 2 looks at Hierarchical clustering and quality of clustering metrics. Chapter 3 introduces Density based clustering techniques - DBSCAN and OPTICS while chapter 4 explores an emerging method of clustering which is elegant and easy to implement, the Spectral Clustering algorithms. Chapter 5 considers results of using the preceding algorithms in the design of a single receive antenna and also introduces methods of extending the clustering algorithm methods to the case of multiple receive antennas.

CHAPTER 2

Clustering Metrics and Hierarchical Clustering

2.1 Introduction

Determining the PDF of the the signal space can be done as we have seen through the EM algorithm that fits a GMM on the data. However, the algorithm like other approaches to clustering like Hierarchical clustering and k-means require an estimate of the number of clusters in the data. One approach is the OPTICS algorithm or using the Spectral properties of the data which are explored in later sections. In this chapter, we focus on clustering metrics and methods which select the number of clusters based on the quality of clustering.

2.2 Clustering Metrics

Clustering metrics are measures of 'quality' or goodness of fit of a partition of clusters. They can be used to validate the clustering. Measuring the appropriateness of an instance of clustering is very dependent on the metrics chosen. In this section, some commonly used metrics are presented, analysed and their performance shown. The clustering metrics can therefore be used to choose the number of clusters in the data.

These measures of cluster validity are classified into two main types -

- **External index** - These measures compare how well cluster labels match externally supplied ones. These indices are not useful in the problem at hand because we do not have externally classified labels.
- **Internal index** - They are used to measure goodness of fit without labels being externally provided. The metrics presented in this chapter fall under this category. They can be further divided into global and local methods. The former evaluating the measure over the entire data set and finding the optimum with respect to the number of clusters and the latter considering pairs of clusters and evaluating need for amalgamation (Gordon, 1999).

2.2.1 Hartigan Index

Most clustering metrics compute and contrast Cluster Cohesion with Cluster Separation. Cluster cohesion is a measure of how similar or close objects in a particular cluster are while cluster separation measures how far apart the clusters are.

A proposed within cluster dissimilarity metric is Sum of Squares Error (SSE) defined as the sum of distance square of all points from the centroid of its cluster,

$$W(k) = \sum_{j=1...k} \sum_{i \in C_j} d^2(x_i, \bar{x}_j), \tag{2.1}$$

where k is the number of clusters. This metric results in a low value when the clustering is good, i.e. for an appropriate number of clusters. However, it is monotonically non-increasing as k increases by construction. Hartigan (1975) proposed a correction factor which considers relative improvement in the SSE metric weighted by a factor depending on the number of clusters as follows -

$$\begin{aligned} H(k) &= \gamma(k) \frac{W(k) - W(k+1)}{W(k+1)}, \\ \gamma(k) &= (n-k-1)^{-1}, \end{aligned} \tag{2.2}$$

where $\gamma(k)$ is the Hartigan correction factor and n is the total number of points. Hartigan proposed that the number of clusters be increased if the measure was greater than 10 and hence the ideal number of clusters was the smallest k such that

$$H(k) \leq 10. \tag{2.3}$$

2.2.2 Silhouette Index

The Silhouette statistic as defined by Kaufman and Rousseeuw (1990) can use any dissimilarity measure like the euclidean distance used in SSE (2.1). The statistic is defined as follows

$$a_i = \frac{1}{|C(i)|} \sum_{j \in C(i)} d^2(x_j, x_i),$$

$$b_i = \min_{j,\, C_j \neq C(i)} \frac{1}{|C_j|} \sum_{l \in C_j} d^2(x_l, x_i),$$

$$Sil_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{2.4}$$

$$Sil(k) = \frac{1}{n} \sum_{i=1\ldots n} Sil_i. \tag{2.5}$$

Here a_i represents the cluster cohesion of a point, b_i is a measure of cluster separation - it is the least dissimilarity measure between a point and points in another cluster. Sil_i represents the statistic for each point. It is normalized and a maximum value of 1 represents good clustering and a minimum value of -1 indicates that the point may be better placed in another cluster. $Sil(k)$ represents the average silhouette statistic of the data set. The optimal number of clusters k is chosen if $Sil(k)$ is maximized, this happens if the data is tightly grouped.

2.2.3 Modularity

Correlation is a metric which works out the correlation between two matrices - the proximity or similarity matrix $\mathbf{S}$ and an 'Incidence' matrix $\mathbf{A}$. Both matrices have a row and column for every data point. Both $\mathbf{S}$ and $\mathbf{A}$ are symmetric matrices. There are a variety of similarity measures, one being the Gaussian kernel as defined below,

$$S_{i,j} = \begin{cases} e^{-\sigma d^2(x_i, x_j)} & i \neq j, \\ 0 & i = j. \end{cases} \tag{2.6}$$

where σ is a parameter that weights the distance measure - a larger σ would imply that only points which are very close by are similar. $S_{i,j}$ can be regarded as a link between data point (node) i and j. The incidence matrix $\mathbf{A}$ have elements defined as,

$$A_{i,j} = \begin{cases} 1 & C(i) = C(j),\ i \neq j \\ 0 & C(i) \neq C(j)\ \vee i = j. \end{cases}$$

A high correlation between these two matrices is a crude indicator of how well the clustering is. Another measure that uses matrix $\mathbf{S}$ is the Modularity measure which compares the clustering to a random network or distribution of data points. The degree of a data point or node k_i is defined as the sum of link weights from node i to all other nodes or,

$$\begin{aligned} k_i &= \sum_{j=1\ldots n} S_{i,j}, \\ l_n &= \sum_{i=1\ldots n} k_i \\ &= 2m, \end{aligned} \tag{2.7}$$

where l_n is the sum of degrees of all points and m is the average degree of a node. If we were to generate a random network of the same degree distribution as our dataset, the expected link-weight between node i and j would be $\frac{k_i k_j}{2m}$. Modularity Q is a measure of how much the link weight between nodes in a community is more than the expected link weight in a random network with the same degree distribution and is defined as,

$$Q = \frac{1}{2m}\sum_{i,j}\left[S_{i,j} - \frac{k_i \times k_j}{2m}\right]\delta\left(C(i), C(j)\right). \tag{2.8}$$

$$\begin{aligned} \delta\left(C(i), C(j)\right) &= \sum_r A_{i,r}A_{j,r}, \\ B_{i,j} &= S_{i,j} - \frac{k_i \times k_j}{2m}. \end{aligned}$$

Hence,

$$Q = \frac{1}{2m}\sum_{i,j}\sum_r B_{i,j}A_{i,r}A_{j,r} = \frac{1}{2m}\mathrm{Tr}(\mathbf{A}^\intercal\mathbf{B}\mathbf{A}). \tag{2.9}$$

The higher the modularity, the more pronounced the community structure is in the dataset. Clustering may also be done through maximization of Q which involves a process that can be written in terms of its eigenspectrum similar to the Spectral Clustering approach described in chapter 4 (Newman, 2006). The modularity measure suffers from a bias which can fail to resolve well defined small clusters is the network is large (Fortu-

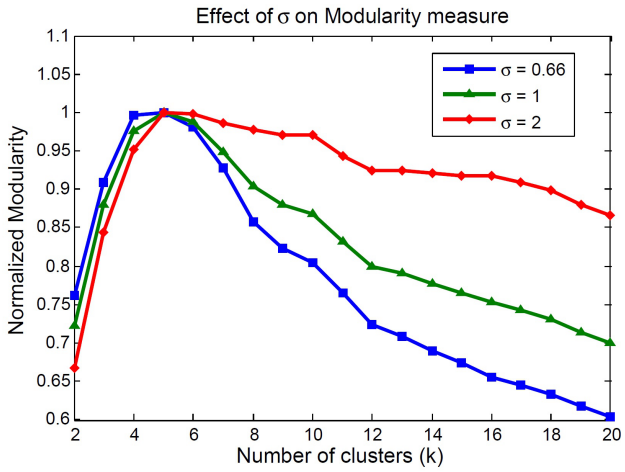

Figure 2.1: Normalized modularity measure on a dataset obtained from the Ped A channel. The parameter σ is seen to scale the value but not the choice of the right cluster order. 16 means Hierarchical clustering performed.

nato and Barthélemy, 2007). Fig. 2.1 shows the effect of the varying the σ parameter in the Proximity matrix $\mathbf{S}$. The normalized modularity appears to be scaled but the choice of the best cluster number is the same in every case. Hence there is a wide choice of parameter σ. This simulation was run on the Ped A dataset obtained from the Ped A channel with 3 QPSK interferers at an SNR of 10.

2.2.4 Gap Statistic

Another clustering metric approach which uses a reference random scenario (or network) is the Gap statistic proposed by Tibshirani *et al.* (2001). It also makes use of the SSE metric in (2.1) and compares it to a reference random distribution. b Monte Carlo samples $X_1, \ldots, X_b$ with n points each are drawn from the reference uniform distribution over the signal space. We can also perform Singular Value Decomposition on the data set X to determine the principal components and generate the Monte Carlo samples in this rotated frame.

We then perform clustering as was done with the dataset X and compute expected value $\mathbb{E}\left[\log\left(W^*(k)\right)\right]$ by taking the average of the quality metrics of each sample. Additionally the standard deviation of the measure is also calculated. The Gap statistic is

defined as,

$$Gap(k) = \frac{1}{b}\sum_{i=1}^{b} \log(W_i^*(k)) - \log(W(k)). \tag{2.10}$$

The ideal number of clusters $\hat{k}$ is computed as the smallest k such that,

$$Gap(k) \geq Gap(k+1) - sd_{k+1}, \tag{2.11}$$

where sd_{k+1} is the computed standard deviation of the quality measure among all samples. This measure is more computationally intensive than the earlier metrics because multiple realizations of the reference uniform distribution are required for reliable results with Tibshirani *et al.* (2001) using $b = 50$ in simulations.

2.2.5 Other approaches

There are several other cluster metric approaches which use sum of squared distances between clusters or within a cluster. A notable one is the metric proposed by Calinski and Harabasz (1974) :

$$CH(k) = \frac{B(k)}{k-1} \times \frac{n-k}{W(k)},$$

where $W(k)$ is defined as in (Hartigan Index) and $B(k)$ is the between cluster sum of squares.

Another approach is the internal use of external metrics. Ben-Hur *et al.* (2002) proposes a stability approach that clusters the dataset X and a modified dataset X^* with a fraction of the points deleted. The similarities between the two resulting methods is compared to give a measure of the quality of clustering.

2.3 Hierarchical Clustering

Hierarchical clustering is a connectivity based clustering approach. These algorithms seek to group objects that are closer based on a distance measure. At every step of these

algorithms, different clusters form which can be represented on a dendrogram shown in Fig. 2.2. Algorithms mainly differ on the linkage criterion used and the metric for distance.

Hierarchical algorithms are of two general types -

1. **Agglomerative** - Each observation starts in a different cluster and at each step pairs of clusters merge based on a greedy linkage criterion.
2. **Divisive** - All data points are initially formed into one cluster and at every step, recursive splits are performed. An exhaustive search in divisive clustering is $O(2^n)$ which makes this method less favoured in our model.

Various metrics for distance measure include the euclidean distance, squared euclidean distance, Manhattan distance which is the absolute difference between data points, Mahanolobis distance which weights various dimensions, Hamming distance or cosine similarity. The data points for our usage are only in two dimensions and euclidean distance is an appropriate metric.

Linkage criterion determines the distance between a pair of clusters as a function of pairwise distances between observations. Commonly used criterion include:

1. **Complete Linkage clustering**- The distance between two clusters is given by,

$$d(C_i, C_j) = \max \{d(x_a, x_b) : \ a \in C_i, \ b \in C_j\} . \tag{2.12}$$

   An efficient implementation in $O(n^2)$ steps is presented in Defays (1977). Everitt *et al.* (2009) showed that complete linkage clustering tends to find compact clusters of nearly equal diameters.

2. **Single Linkage clustering**- The distance between two clusters is given by,

$$d(C_i, C_j) = \min \{d(x_a, x_b) : \ a \in C_i, \ b \in C_j\} .$$

   Single linkage suffers from the problem of chaining where clusters may be merged together due to single elements in the two clusters being close to one another.

3. More computationally complex and general linkage criterion are described such as centroid distance between clusters or minimum energy clustering which cannot be done better than $O(n^3)$.

Dasgupta and Long (2005) have shown that it is possible to construct hierarchical clustering with a performance guarantee that the maximum radius of the resulting clusters is at most eight times that of the optimal k clustering.

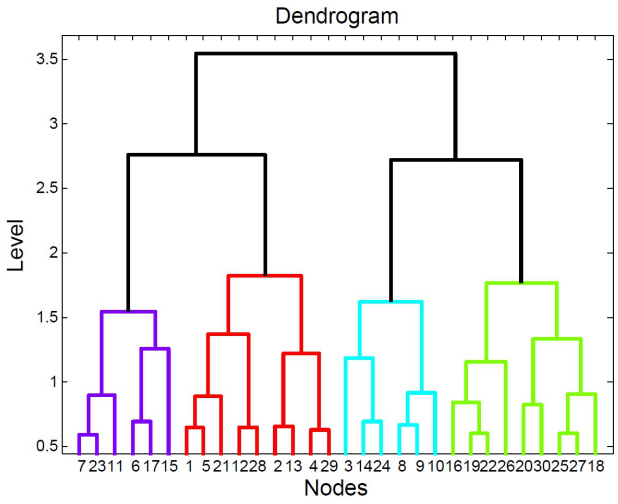

Figure 2.2: Dendrogram from performing agglomerative complete-link hierarchical clustering on the dataset from the Ped A channel.

Hierarchical clustering algorithm for cluster order estimation

The essential idea is that we insert a clustering metric (order $O(n^2)$) in every stage of the agglomerative complete link hierarchical algorithm. When the pre-set minimum number of clusters is reached, the clustering id corresponding to the highest quality measure is returned. The steps are as highlighted in Algorithm 1. This algorithm is of order $O(n^3)$.

Algorithm 1 Hierarchical clustering algorithm for model order estimation.

$n \times n$ distance matrix $\mathbf{D}$ with $D_{i,j} = d(x_i, x_j)$ is given
ClusterID = $1, 2, \ldots n$ assigned.
Loop till number of clusters = 1

1. Find least dissimilar pair of clusters $a,\ b$ such that $d(a,b) = \min_{i,j} d(i,j)$.
2. Update ClusterID such that the id of all points in $b = a$.
3. Update matrix $\mathbf{D}$ by deleting rows and columns corresponding to a and b with distance of other clusters from the merged cluster given as $d\left(k,(a,b)\right) = \max\left[d(k,a),\ d(k,b)\right]$.
4. Calculate the quality of the cluster and if it is the best till now, ClusterIdOptimal = ClusterId.

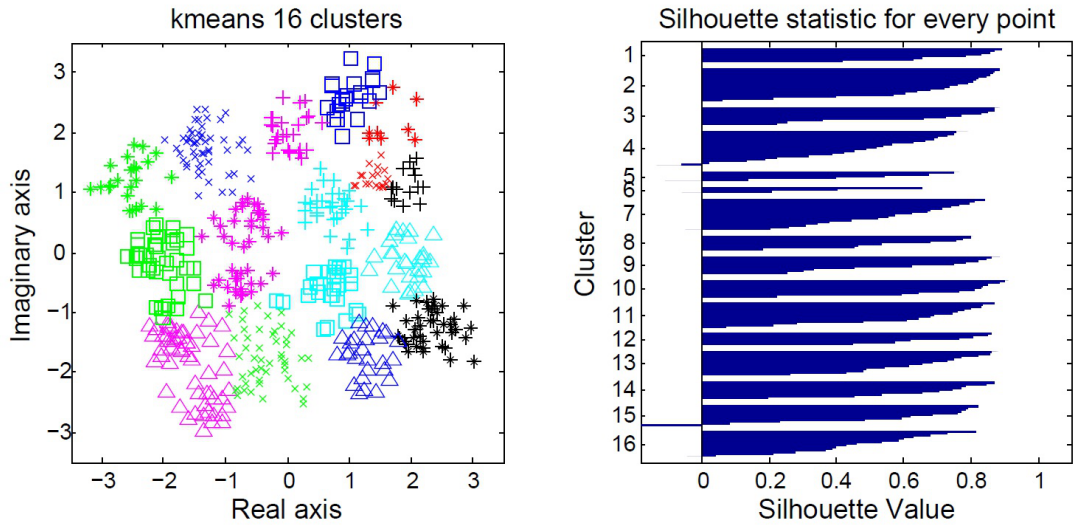

Figure 2.3: Silhouette statistic on kmeans clustered dataset generated from Ped A channel, 3 QPSK inteferers (0, -3, -6 dB) with SNR 10

2.4 Results and discussion

2.4.1 What The Metrics Measure

Fig. 2.3 shows the silhouette statistic applied to all 16 clusters that a kmeans algorithm has recognized. Although, the algorithm has not performed clustering very well (as can be evidenced in the bottom left portion where two distinct clusters have been merged into one), it receives a high silhouette score as kmeans seeks to minimize SSE, which is what the statistic measures. This may not necessarily translate into a better quality clustering. The performance evaluation measures are very sensitive to what we seek to measure and are not entirely indicative of the quality of clustering. These measures are suitable for only a particular choice of data. If we anticipate the presence of non-convex shaped clusters or situations where one cluster envelops another, all the clustering indices will give a poor score which is not truly representative of the scenario. Clustering cohesion vs. separation is not the best comparison approach for all cases. Fig. 2.4 contrasts silhouette performance from the same dataset on kmeans clustering as well as Hierarchical clustering. Although, silhouette scores for kmeans are higher and imply that clustering is better, kmeans is unable to identify the ideal number of clusters as 4 appears to be the best solution. Although, Hierarchical clustering receives lower silhouette scores, 16 is correctly identified as the number of clusters.

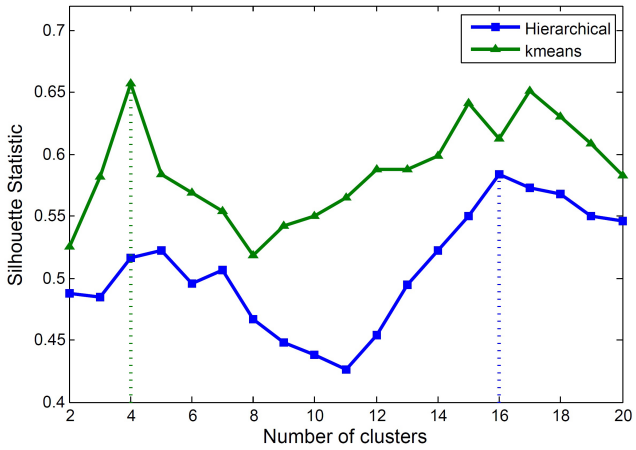

Figure 2.4: Silhouette statistic on kmeans and Hierarchical algorithms. Although, the latter gets a lower evaluation, the right number of clusters is properly evaluated as 16 unlike kmeans.

2.4.2 Comparison of the Clustering Metrics

The Hierarchical algorithm for model order estimation was run on typical datasets that arise from our channel. The first as described in the preceding section is a dataset obtained from 3 QPSK interferers of 0, -3 and -6 dB power and an SNR of 10. The other is obtained from 2 QPSK interferers (0 and -6 dB) on the Ped B channel with an SNR of 12. It can be seen from Fig. 2.5 and 2.6 that the Silhouette and the Gap statistic outperform Modularity and Hartigan index. The Hartigan index results in very poor results from the proposed cluster order heuristic mentioned in (2.3). The knee of the graph or the portion of the graph where the Hartigan index decreases at a much lower rate is the optimal clustering order and it is observed to match the true order in both cases but automated order extraction is a challenge. The Gap statistic while yielding similar results as the Silhouette statistic is much more memory and time intensive as it relies on multiple realizations from a reference distribution and hence, we use the Silhouette statistic as a less intensive metric in the Hierarchical cluster order extraction algorithm. Table 2.1 lists what the heuristics from the metrics propose as the cluster order. Also, from Fig. 2.5, we see that the Hierarchical clustering algorithm has identified the cluster means more similarly to what we observe and expect than the kmeans model seen in Fig. 2.3.

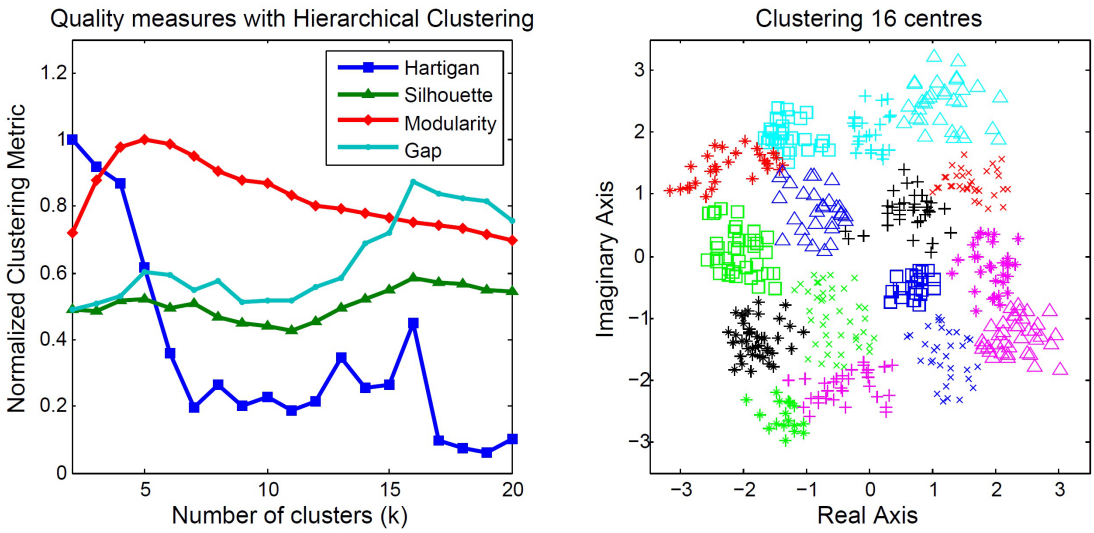

Figure 2.5: Comparing the cluster validation metrics on the dataset from the Ped A channel.

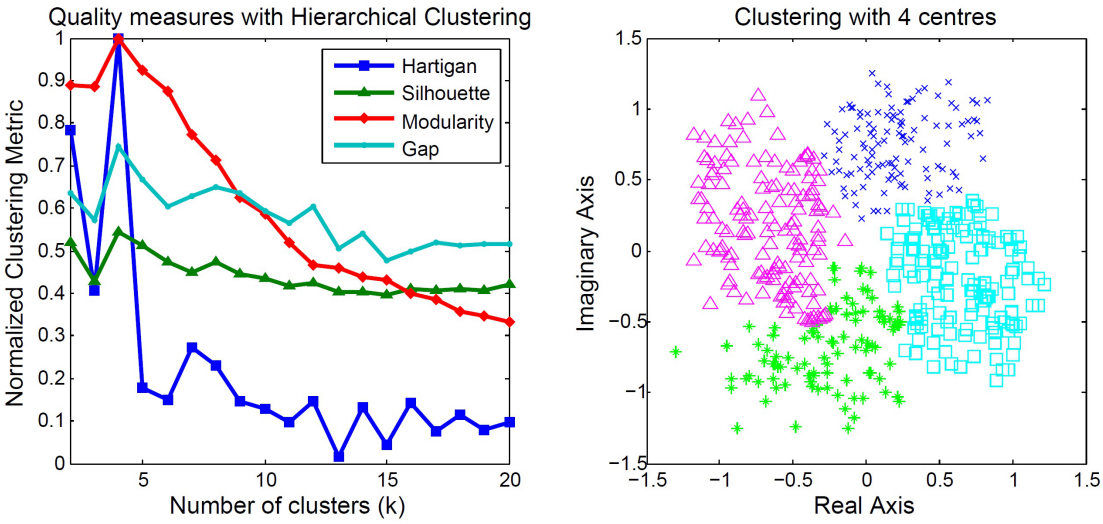

Figure 2.6: Comparing the cluster validation metrics on the dataset from the Ped B channel

Clustering Metric	Ped A channel Fig. 2.4	Ped B channel Fig. 2.5
Hartigan Index	16*	4*
Silhouette	16	4
Modularity	5	4
Gap Statistic	16	4

Table 2.1: Comparison of the model order estimated by using various clustering metrics. Hartigan index value is observed and not as predicted by the heuristic in (2.3).

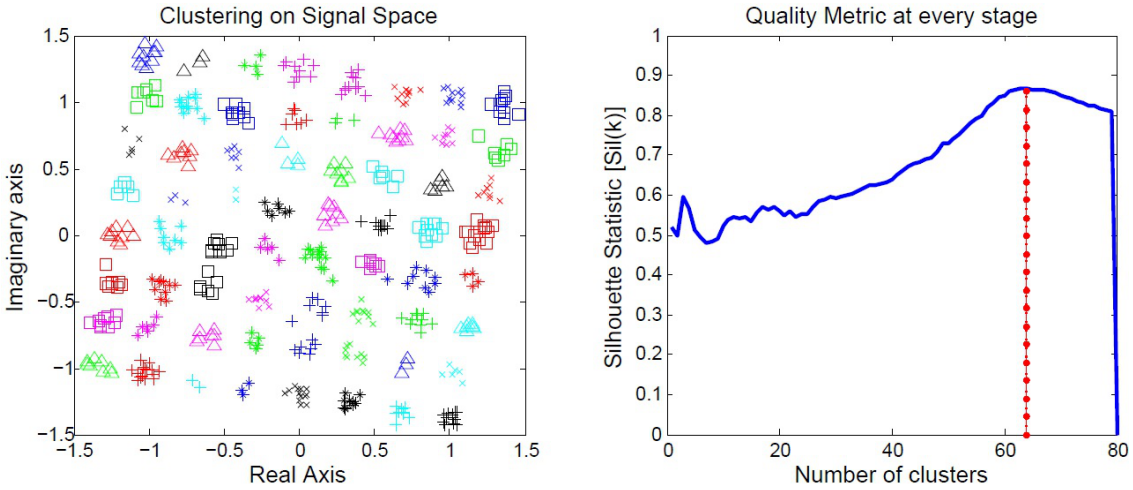

Figure 2.7: Hierarchical clustering algorithm correctly estimating model order and clustering configuration of a dataset from Ped A channel.

2.4.3 The Hierarchical Model Extraction Algorithm

Fig. 2.6 shows the algorithm performing on a more artificial dataset of a single 64 QAM interferer with an SNR of 22. The algorithm with the silhouette statistic built-in has chosen 64 correctly as the cluster order.

CHAPTER 3

Density Based Clustering

3.1 Introduction

We have in the preceding chapters encountered connectivity models, centroid models and distribution models of clustering. In this chapter, we explore density based clustering approaches OPTICS and DBSCAN for estimating cluster model order and for modelling the PDF of the interference and noise in the signal space.

Density based clustering models use a 'density based notion' of a cluster and are capable of detecting clusters of arbitrary shape which none of the earlier clustering techniques are capable of. The density-notion of clustering implies that the density (Number of points in a given radius) of a point in a cluster is more than a certain threshold.

Algorithms which perform density based clustering are introduced in the subsequent section followed by a results and discussion section.

3.2 DBSCAN

Ester *et al.* (1996) presented DBSCAN and formalized the notion of density based clusters and noise.

3.2.1 Definitions

An ϵ neighbourhood of a point p, $N_\epsilon(p)$ is defined as,

$$N_\epsilon(p) = \{q \in D \mid d(p,q) < \epsilon\}, \tag{3.1}$$

where $d(p,q)$ can be any distance metric between the points, euclidean distance is chosen in our system. A *core point* is one in which $|N_\epsilon(p)| > MinPts$, with $MinPts$

being the minimum number of points for such a classification. ϵ and $MinPts$ are the parameters fed to the algorithm. A core point p is *directly density-reachable* from point q if the latter is a core point and $p \in N_\epsilon(q)$. A point p is *density-reachable* from point q if there is a sequence of points $p = p_1,\ p_2, \ldots,\ p_n = q$ such that p_i is directly density reachable from p_{i+1}. Points p and q are *density-connected* if there exists point o such that p and q are density-reachable from point o.

A cluster C is defined on database D as a non-empty subset of D such that:

1. $\forall p,\ q$ if $q \in C$ and p is density-reachable from q, then $q \in C$. This is the maximality condition
2. All points in a cluster are pairwise density connected. This is the connectivity condition.

Noise points are those points which have not been classified into a cluster. Clearly, every cluster has to have at least $MinPts$ points as there needs to be a core point for the members to be density-connected. Also, every point in the cluster is density-reachable from a core point in the cluster.

3.2.2 Algorithm

DBSCAN starts with an arbitrary point p and finds $N_\epsilon(p)$. If it is a core point, it is assigned to a new cluster and all the density-reachable points from p are found and added to the cluster. If p is not a core point, it may be a *border point* in the cluster or a noise point and another arbitrary point is chosen and the procedure repeated. This is detailed in algorithm 2.

3.2.3 Comments

The clustering is very sensitive to parameters ϵ and $MinPts$. In our system, ϵ can be a function of the noise variance as each cluster can be regarded as a Gaussian PDF with variance as the noise variance. In the subsequent section, a method to estimate ϵ from the reachability plot is presented. As we have 500 points per dataset and do not expect the number of clusters in most cases to exceed 30-50, $MinPts$ of 10 can be chosen. The algorithm presents the cluster order but requires two difficult to establish parameters.

Algorithm 2 DBSCAN algorithm

Loop p through all unvisited points in database D

1. if $|N_\epsilon(p)| > MinPts$, mark point as visited and assign to new cluster. Start Expand Cluster routine passing $N_\epsilon(p)$ detailed below.
2. Else mark as noise and continue

Expand Cluster:

List= $N_\epsilon(p)$

Loop q through unvisited points in List

1. If q is a core point, add $N_\epsilon(q)$ to List. Mark as visited and assign to $C(p)$.
2. Else q is a border point. Mark as visited and assign to $C(p)$.

The parameters are also global parameters for all clusters and hence clusters of varying densities cannot be detected. A low $MinPts$ value has the same chaining problem of single-linkage Hierarchical clustering introduced in the previous chapter. A high $MinPts$ value will require a higher ϵ value else more points will be categorized as noise but larger ϵ values limit the resolution of the clustering. Fig. 3.1 shows the effect of the $MinPts$ parameter with ϵ chosen such that the maximum fraction of noise points is the same. The clustering resulting from lower $MinPts$ value has more clusters of smaller size which would otherwise be categorized as noise.

The complexity of the DBSCAN algorithm is limited mainly by the routine to find the ϵ neighbourhood of a point. Without indexing, it takes place in $O(n)$ steps. Since every point is scanned only once for its neighbourhood, the algorithm runs at $O(n^2)$ and can run at $O(n \log n)$ if there is an indexing structure for the neighbourhood query.

Another advantage of DBSCAN is that the clustering is relatively inert to the order in which points are selected. Two border points in adjoining clusters may get swapped if points are evaluated in a different order. This can be seen in Fig. 3.2.

3.3 OPTICS

Ankerst *et al.* (1999) presented the OPTICS algorithm as an extension to the DBSCAN algorithm which had two main failings - estimation of the ϵ parameter and detection of clusters of varying densities.

The OPTICS algorithm unlike the DBSCAN algorithm is not a clustering algorithm

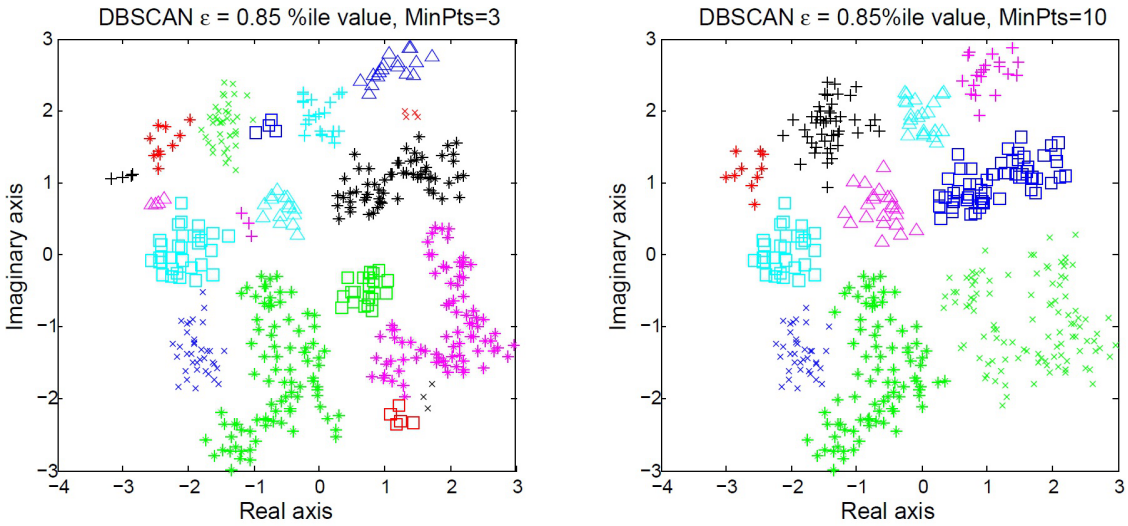

Figure 3.1: Effect of parameter $MinPts$ on DBSCAN. Ped A dataset is used. The ϵ value used is such that the maximum fraction of noise points is 15%.

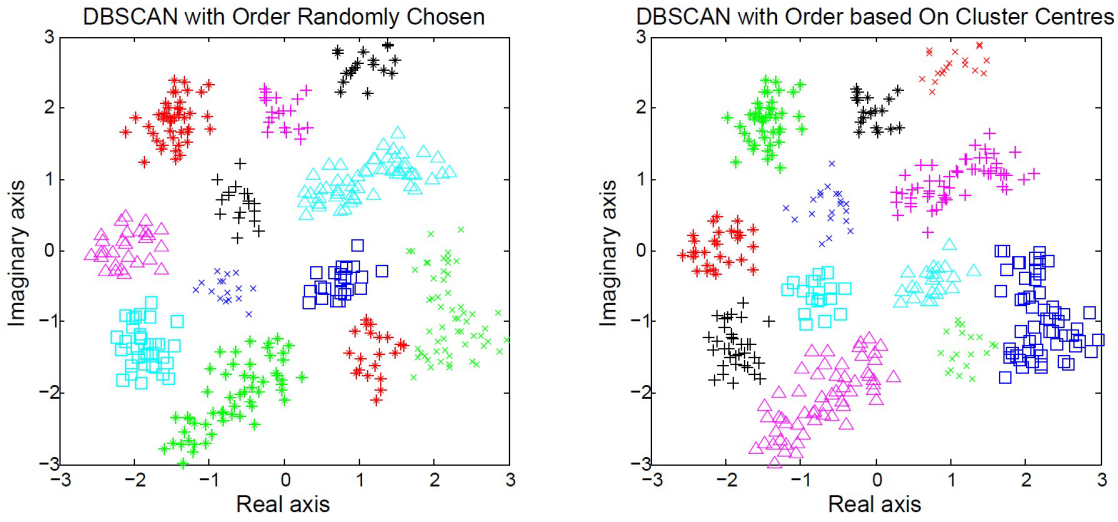

Figure 3.2: Effect of order of points on DBSCAN algorithms. In the case on the right, after a cluster was chosen, the next point was not chosen randomly but a point near another cluster centroid that was externally supplied.

in the sense that it does not assign cluster labels to datapoints but the clustering can be easily extracted from the reachability plot which the algorithm generates.

3.3.1 Definitions

The OPTICS algorithm also requires two parameters : ϵ, which is the maximum radius to consider and $MinPts$, the minimum number of points that can form a cluster. The *core-distance* of a point is defined only for points which have more than $MinPts$ points in its ϵ neighbourhood. For such points, it is the distance to the $MinPts$-th closest point.

Reachability distance of a point p from q is defined only if q is a core point with respect to ϵ and $MinPts$. It is the maximum of the distance from point p to q and the core distance of q. It cannot be smaller than the core distance of q because it would then imply that q is not a core point.

For a given value of $MinPts$, and $\epsilon_1 < \epsilon_2$, DBSCAN produces clusters in the first case which are completely contained in the second case. OPTICS extends this idea to any $\epsilon^* < \epsilon$, the generating distance. We can extract any DBSCAN based clustering provided the points are correctly ordered. Or the points that are density-reachable for the lowest ϵ^* values are processed first while expanding the cluster. The OPTICS algorithm provides this order.

3.3.2 Algorithm

The OPTICS algorithm places those objects which are density-reachable for the smallest values of ϵ^* such that they are processed first after a core point is visited. The algorithm chooses an unexplored point and if it is a core point with respect to ϵ, all density-reachable points are explored. Those with the lowest reachability-distance are placed in order first. This is described in more detail in algorithm 3.

3.3.3 Extracting Clusters

DBSCAN clusters of $MinPts$ and any $\epsilon^* < \epsilon$ can be easily obtained from the *reachability-plot*. The reachability plot is the plot of reachability-distances of the points in the or-

Algorithm 3 OPTICS algorithm

Loop p over all unvisited points in database D

1. If $|N_\epsilon(p)| > MinPts$, add p is written in the ordered file, mark as visited, core-distance is calculated and reachability-distance is left undefined. Expand Cluster routine is called.

2. Else, continue and choose another point p

Expand Cluster:

List= $N_\epsilon(p)$

Reachability distance of $d \in N_\epsilon(p)$ calculated.

Loop q through unvisited points in List with lowest reachability-distance

1. If q is a core point, its core distance is calculated. $N_\epsilon(q)$ is added to list and reachability distance of these neighbouring points are calculated and updated.

2. q is marked as visited and added to the ordered file.

dered file. We scan through the points in this order. If the reachability distance of the point is undefined or more than ϵ^*, and the core-distance is less than ϵ^*, it is a core point and added to a new cluster. If the core distance is more, it implies that it is a noise point. Points whose reachability distances are less than ϵ^* are added to the cluster. As mentioned in the preceding section, the reachability distances of the points offers a heuristic to estimating ϵ for DBSCAN. If we choose an ϵ value that is the 85 percentile value of the reachability distance, not more than 15% of the datapoints will be classified as noise. This method of cluster extraction suffers from the same problem as DBSCAN - being unable to extract clusters of different densities. Fig. 3.3 displays the reachability-plot from the Ped A dataset. The red line indicates the ϵ value that can be used in DBSCAN as described above. The valleys in the reachability-plot indicate that these points belong to one cluster. These valleys typically start with a steep downward region followed by points with low reachability points and end with a steep upward region which is indicative of the next cluster.

A more sophisticated approach to automated cluster extraction from the reachability plot is to recognize spikes or steep regions as they correspond to core points of new clusters. A ξ-*steep upward point* is one whose reachability distance is $\xi\%$ lower than its successor. The ξ-*steep downward point* is similarly defined. A ξ-*steep region* $I = [s,\ e]$ is defined when the following hold:

1. $s,\ e$ are ξ-steep upward points

2. The reachability-distances of points are non-decreasing from s to e.

3. I does not contain more than $MinPts$ consecutive non-steep points. Else, this region could be a separate cluster and should not be part of the steep region.
4. I is the maximal such interval.

Conceptually, a cluster starts at the last point with a high reachability value and ends at the last point with a low reachability value. Using the notion of steep regions, a ξ-*cluster* $C = [s,\ e]$ can be defined as follows:

1. There exists a down-steep region containing s and an upward steep region containing e.
2. $e - s \geq MinPts$ to satisfy the minimum number of points in a cluster constraint.
3. Reachability value of all points between the steep regions should be $\xi\%$ lower than the starting point of the downward steep region and the last point of the upward steep region.
4. If the terminal points of the steep regions are more than $\xi\%$ away from one another, a point in the steep region with the higher terminal point is chosen to be the start or end of the cluster as the case may be.

Algorithm 4 solves the problem of extracting clusters by recognizing steep regions.

Algorithm 4 Extracting clusters from the reachability plot

Scan $index$ through points in the ordered file

1. If a steep downward region starts at $index$, add to list of steep downward regions L. Update $index$ to last point in the steep region
2. If a steep upward slope starts at $index$, then:
   (a) For every downward steep region D in list L, verify if the intervening points match the cluster requirements. If it does, then add points to a new cluster after finding the start and end points.
   (b) Update $index$ to the last point in the steep region

3.3.4 Comments

The OPTICS algorithm is of the same complexity as the DBSCAN algorithm and can run in $O(n \log n)$ steps if neighbourhood queries are indexed and $O(n^2)$ otherwise. The cluster extraction algorithm performs a single scan through all the datapoints in the order as given by the OPTICS algorithm.

The algorithm is not as sensitive as DBSCAN to parameters ϵ and $MinPts$. For lower values of $MinPts$, the reachability plot is seen to be more jittery but the basic

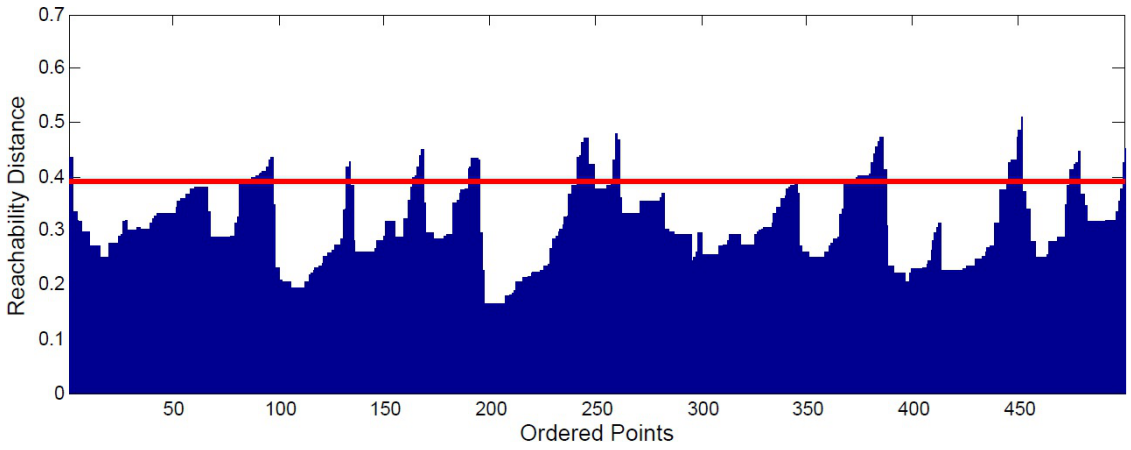

Figure 3.3: Reachability Plot from the Ped A dataset. The red line indicates the ϵ value that is used in DBSCAN that results in less than 15% of the points being classified as noise. The valleys are points in one cluster.

clustering structure can still be ascertained. ϵ should be chosen such that most points are reachable from any point p. An ϵ value which is chosen thus will place most points in one cluster and information of all the clustering levels can be extracted from it. This can be seen in Fig. 3.4. Alternatively, the expected $MinPts$-th neighbour distance when datapoints are randomly distributed can be used as a heuristic for ϵ. A low value of ϵ does not allow for high reachability values and the reachability plot looks cut off. A wide range of parameter values are empirically seen to offer the same results. As the ξ parameter in the cluster extraction algorithm decreases, the number of clusters detected is seen to increase. $\xi = 1$ is empirically seen to identify the correct number of clusters in a number of cases although a wide range of ξ can suffice. Density clustering algorithms require a change in density to detect a cluster which may not always be the case.

3.4 Results

3.4.1 DBSCAN based Order of Clustering

As can be seen from Fig. 3.5 where the parameter ϵ was chosen as varying percentiles the reachability-distances of the points, the resulting number of clusters and quality of clustering is very dependent on the parameters. Estimating these parameters is a challenge.

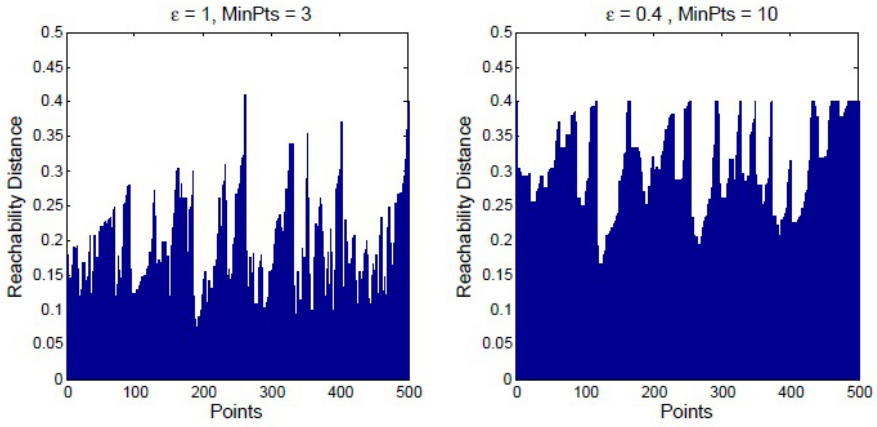

Figure 3.4: Impact of parameters ϵ and $MinPts$ on the reachability-plot. Ped A dataset used.

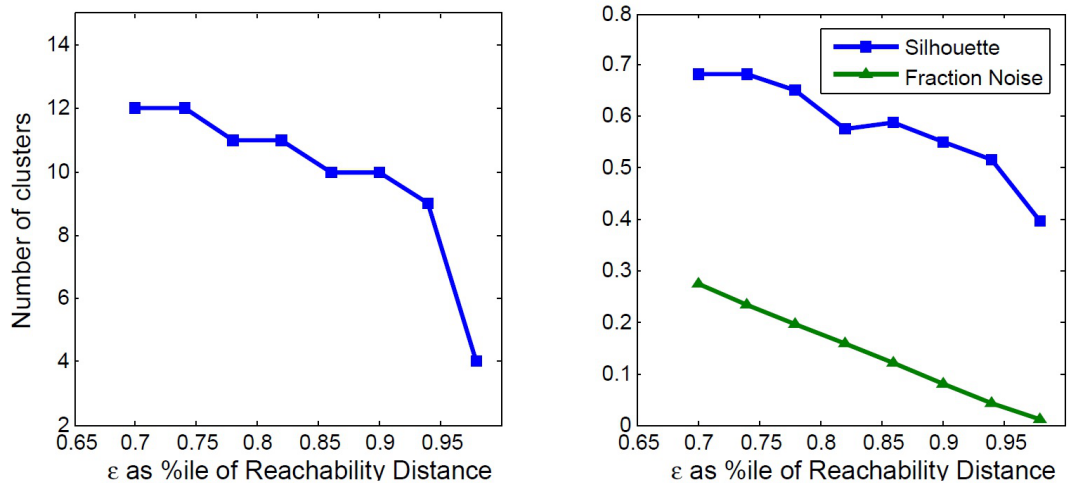

Figure 3.5: Estimating the number of clusters with DBSCAN. Percentage of points that are classified as noise and quality is also recorded. The Ped A dataset is used.

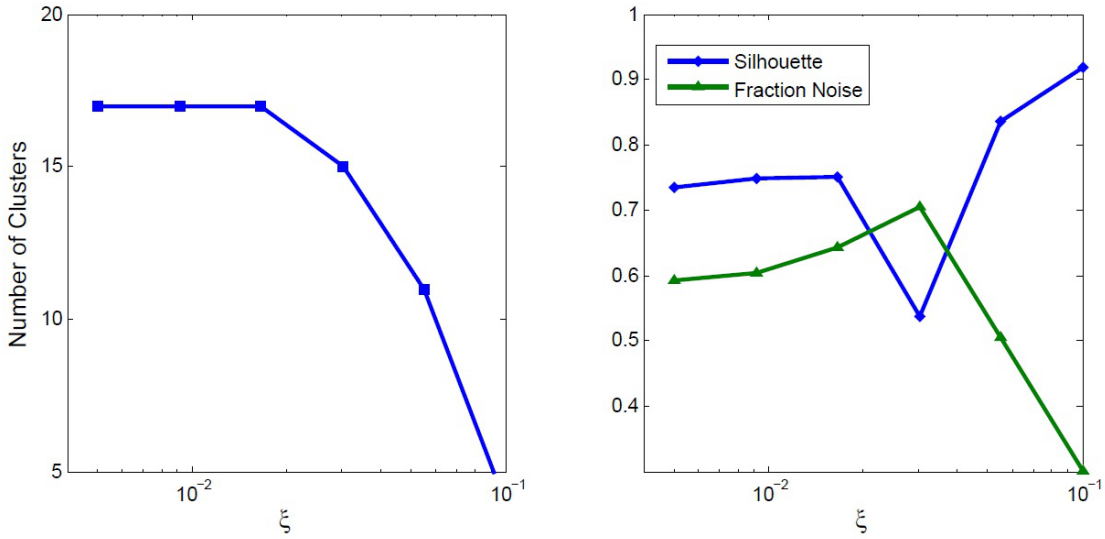

Figure 3.6: Impact of the ξ parameter on the extract clusters from OPTICS algorithm. There is larger flexibility in choosing ξ as long as it is sufficiently small. Larger ξ results in more points being categorized as noise. Ped A dataset used.

3.4.2 OPTICS based Cluster-Extraction

Fig. 3.6 illustrates the impact of the ξ parameter in the extraction of clusters algorithm. Increasing ξ decreases the number of resulting clusters and keeping it near 10^{-2} results in the optimum number. Although more clusters may be detected from smaller values of ξ, this will not impact the setting up of a Gaussian Mixture Model on this clustering as clusters are not merged and stray points are not categorized as independent clusters. Fig. 3.7 and Fig. 3.8 show the resulting clustering on the Ped A and Ped B datasets that were described in the previous chapter.

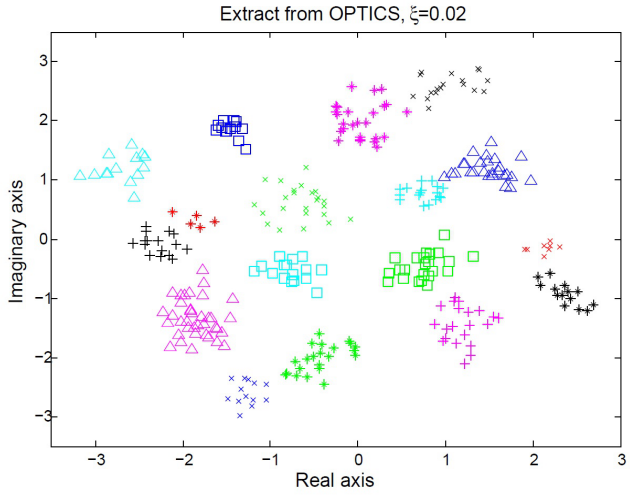

Figure 3.7: OPTICS cluster extraction with $\xi = 2\%$. Noise points are not shown and 18 clusters are identified. Ped A dataset was used.

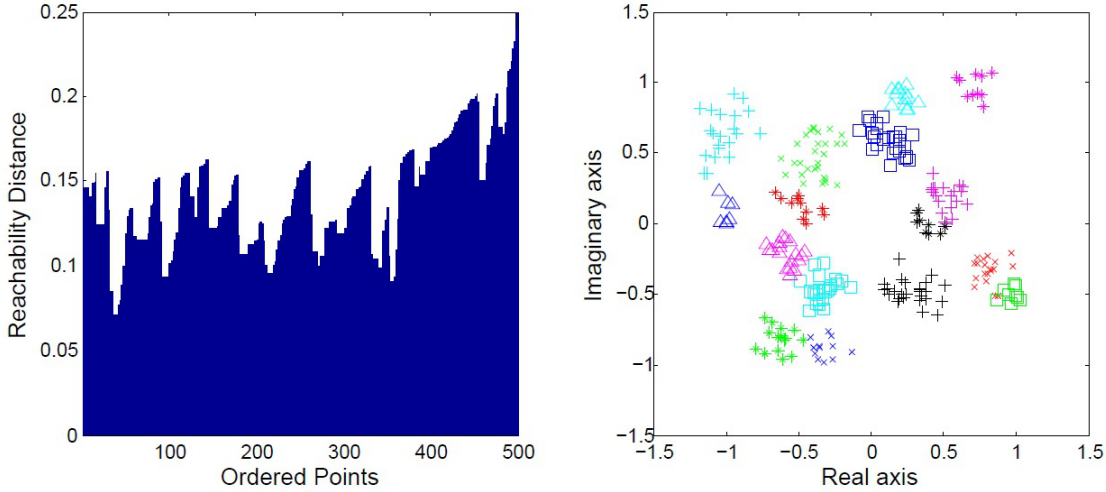

Figure 3.8: OPTICS cluster extraction performance and reachability plot for Ped B dataset. 16 clusters have been identified.

CHAPTER 4

Spectral Clustering Methods

4.1 Introduction

Spectral clustering algorithms are a family of algorithms which have emerged to become widely used because of ease of implementation and performance benefits over traditional clustering algorithms (Von Luxburg, 2007).

Spectral clustering approaches employ the spectrum of the *Laplacian* matrix constructed from the similarity or proximity matrix $\mathbf{S}$ introduced in the section on the Modularity Metric. The spectrum is used to perform dimensionality reduction in $\mathbf{S}$ and clustering is this in this reduced dimension space.

The subsequent sections introduce familiar spectral clustering algorithms Normalized Cut Algorithm and PCCA followed by an analysis.

4.2 Normalized Cut Algorithm

The mathematics behind the Normalized Cut algorithm is first introduced followed by the description of the algorithm and notes on its implementation

4.2.1 Laplacians

From the dataset V, the proximity matrix $\mathbf{S}$ can be constructed from (2.6). Other approaches are detailed in the notes section. The *degree* k_i of node i in V is defined in (2.7). For two subsets of nodes $A,\ B \subset V$, $W(A, B)$ is defined as,

$$W(A, B) = \sum_{i \in A, j \in B} S_{i,j}, \tag{4.1}$$

which is indicative of the strength of links between set of vertices (or clusters) A and B. The *volume* of subset A, an indicator of the size of the subset is given by,

$$\text{vol}(A) = \sum_{i \in A} k_i.$$

Another indicator of size is the *size* of the cluster simply given by $|A|$. The *degree matrix* $\mathbf{D}$ is a diagonal matrix where $D_{i,i} = k_i$. The *Unnormalized Graph Laplacian* $\mathbf{L}$ is defined as,

$$\mathbf{L} = (\mathbf{D} - \mathbf{S}). \tag{4.2}$$

For a vector $\mathbf{f} \in \mathbb{R}^n$,

$$\mathbf{f}^\intercal \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,\ j \in V} S_{i,j} (f_i - f_j)^2. \tag{4.3}$$

This proves that $\mathbf{L}$ is a symmetric positive semi-definite matrix. The eigenvector corresponding to the zero eigenvalue is the constant vector where $f_i = f_j\ \forall i,\ j \in V$. Similarly the *Normalized Graph Laplacian* $\mathbf{L}_{sym}$ is defined as,

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}. \tag{4.4}$$

Just as in (4.3), $\mathbf{L}_{sym}$satisfy,

$$\mathbf{f}^\intercal \mathbf{L}_{sym} \mathbf{f} = \frac{1}{2} \sum_{i,\ j \in V} S_{i,j} \left(\frac{f_i}{\sqrt{k_i}} - \frac{f_j}{\sqrt{k_j}} \right)^2.$$

$\mathbf{L}_{sym}$ is also a symmetric positive semi-definite matrix.

4.2.2 Graph Cut

The graph cut view of clustering seeks to cluster all the vertices in groups A_i such that the sum of edge weights between clusters is minimized - meaning points from different clusters are dissimilar. The *cut* of a partition of V into $A_1, \ldots,\ A_k$ is given by,

$$\text{cut}(A_1, \ldots,\ A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i), \tag{4.5}$$

where $\bar{A}_i$ is the set of all points in V not in A_i. Minimizing cut leads to solutions where the clusters are of dissimilar sizes as separating one vertex from the rest minimizing it. Hence, objective functions such as *RatioCut* (Hagen and Kahng, 1992) and normalized cut *Ncut* (Shi and Malik, 2000) have been defined to penalize dissimilarly sized configurations,

$$\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}, \tag{4.6}$$

$$\text{Ncut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}. \tag{4.7}$$

Spectral clustering is a means to solve the relaxed versions of the minimization of the above values.

Consider a partition $A,\ \bar{A}$ in V. Define indicator vector $\mathbf{f} \in \mathbb{R}^n$ as,

$$f_i = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}} & v_i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}} & v_i \notin A. \end{cases} \tag{4.8}$$

From (4.3), we get,

$$\begin{aligned} \mathbf{f}^\intercal \mathbf{L} \mathbf{f} &= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} S_{i,j} \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in A, j \in A} S_{i,j} \left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 \\ &= |V| \times \text{RatioCut}(A, \bar{A}). \end{aligned} \tag{4.9}$$

Also,

$$\mathbf{f}^\intercal \mathbf{f} = n, \tag{4.10}$$

$$\sum f_i = 0. \tag{4.11}$$

The last equation implying that $\mathbf{f}$ is perpendicular to the constant vector, another eigenvector of $\mathbf{L}$. The minimization of RatioCut in (4.6) with f_i as defined in (4.8) subject to constraints (4.10) and (4.11) is an NP hard problem. It can be relaxed by allowing $\mathbf{f}$ to

take any real value. The relaxed problem is,

$$\min_{\mathbf{f}\in\mathbb{R}^n} \mathbf{f}^\intercal \mathbf{L}\mathbf{f} \text{ subject to } \sum_i f_i = 0,\ \mathbf{f}^\intercal\mathbf{f} = n. \tag{4.12}$$

The above equation by the Rayleigh-Ritz theorem has the solution of $\mathbf{f}$ as the second smallest eigenvector of $\mathbf{L}$. Cluster labels can be assigned by seeing the sign of f_i or more generally by performing k-means clustering initialized with two cluster centres.

The problem can be extended to case with an arbitrary number of clusters k by defining indicator matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ with ,

$$H_{i,j} = \begin{cases} \frac{1}{\sqrt{|A_j|}} & v_i \in A_j \\ 0 & \text{otherwise.} \end{cases}$$

Also,

$$\text{RatioCut}(A_1, \ldots,\ A_k) = \text{Tr}(\mathbf{H}^\intercal\mathbf{L}\mathbf{H}).$$

The relaxed problem becomes:

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \text{Tr}(\mathbf{H}^\intercal\mathbf{L}\mathbf{H}) \text{ subject to } \mathbf{H}^\intercal\mathbf{H} = \mathbf{I}.$$

By the Rayleigh-Ritz theorem, the $\mathbf{H}$ is formed from the smallest k eigenvectors of $\mathbf{L}$. To obtain clustering, k-means clustering is done on $\mathbf{H}$.

Similar to the analysis above, the normalized cut is found to be,

$$\text{Ncut}(A_1, \ldots,\ A_k) = \text{Tr}(\mathbf{T}^\intercal\mathbf{L}_{sym}\mathbf{T}),$$

where $\mathbf{T} = \mathbf{D}^{-1/2}\mathbf{H}$. The relaxation of the problem of finding the minimum cut is as given below:

$$\min_{\mathbf{T}\in\mathbb{R}^{n\times k}} \text{Tr}(\mathbf{T}^\intercal\mathbf{L}_{sym}\mathbf{T}) \text{ subject to } \mathbf{T}^\intercal\mathbf{T} = \mathbf{I}.$$

The solution for $\mathbf{T}$ is the first k eigenvectors of $\mathbf{L}_{sym}$ and transforming, the solution for

$\mathbf{H}$ is the first k eigenvectors of another Laplacian $\mathbf{L}_{rw}$defined as,

$$\mathbf{L}_{rw} = \mathbf{D}^{-1}\mathbf{L}. \tag{4.13}$$

Meila and Shi (2001) established the equivalence of of the transition probability on a random walk and the normalized cut. Minimizing normalized cut is equivalent to finding a cut to minimizing the probability of transition from one cluster to another when taking a random walk on the graph. RatioCut minimization is achieved by maximizing the size of the clusters and an emphasis on within cluster similarity is not placed. Hence, the normalized spectral clustering approach is preferred as it achieves the twin objectives of minimizing cut$(A, \bar{A})$ as well as maximizing within-cluster similarities $W(A, A)$ and $W(\bar{A}, \bar{A})$.

4.2.3 Normalized Cut Algorithm

The algorithm for normalized spectral clustering involves constructing a similarity matrix from the data. The first k eigenvectors of the Laplacian $\mathbf{L}_{rw}$ are found. Finally k-means clustering is done on the n rows of the matrix containing the k eigenvectors. This is described in algorithm 5.

Algorithm 5 Normalized Cut Algorithm

One approach to spectral clustering:

1. Proximity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is constructed from the dataset.
2. Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is found.
3. The smallest k generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ found from the general eigenvector problem: $\mathbf{Lu} = \lambda\mathbf{Du}$. This is equivalent to finding the eigenvectors of $\mathbf{L}_{rw}$.
4. Matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$ having $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns constructed k-means clustering is done on the rows of $\mathbf{U}$.

The construction of the similarity matrix from the data can be done through different methods. Apart from simply a different choice of similarity measure, points could be connected if they lie in an ϵ neighbourhood, are k closest neighbours or only if both points are mutually among the k closest neighbours of each other. Similarity matrices constructed from the ϵ neighbourhood approach tend to connect only points of a higher density while the k nearest neighbour approach can connect points on different scales

(Von Luxburg, 2007). Both parameters are difficult to establish while the completely connected graph which has been implemented in (2.6) is not a sparse matrix increasing the complexity of algorithms finding the eigenvectors of the Laplacians. One method of estimating the parameter σ is to have the mean radius of the kernel envelop $k \sim \log n$ points.

4.2.4 Extracting the number of clusters

The eigengap heuristic is proposed in Von Luxburg (2007) which is applicable for all Laplacians. Number k is chosen such that eigenvalues $\lambda_1, \ldots, \lambda_k$ is small and λ_{k+1} is relatively larger. The justifications presented are from perturbation theory based on the observation that k completely disconnected clusters, eigenvalue 0 has a multiplicity of k and the $k+1$-th eigenvalue is much larger. Such heuristics return non-ambiguous results only when the clustering structure is very pronounced. Weber (2004) suggests choosing λ_{k+1} as the first eigenvalue above a minimum bound, another difficult parameter in practice to evaluate.

4.3 PCCA Algorithm

The Perron Cluster Analysis algorithm was introduced and analyzed by Weber *et al.* (2004). Extension of the problem to general similarity matrices was carried out by Kumar *et al.* (2013).

The PCCA algorithm is very similar to the other algorithm in terms of construction of the graph Laplacian from the similarity matrix but later assigns a soft clustering to the points that exploits the structural properties of the objects even in the spectral sub-space instead of simply performing k-means. A cluster here is a vertex of a simplex in the spectral sub-space and clustering as the membership of data to these vertices.

4.3.1 Simplex

The transition probability matrix for a markov process defined on the graph or for a random walk on the graph is,

$$\mathbf{T} = \mathbf{D}^{-1}\mathbf{S} = \mathbf{I} - \mathbf{L}_{rw},$$

where the symbols are as defined earlier. If there were disjoint clusters, the multiplicity of eigenvalue $\lambda = 0$ would be the number of clusters k with eigenvectors being $\mathbf{f}_1, \ldots, \mathbf{f}_k$ with $\mathbf{f}_i \in \mathbb{R}^n$ and,

$$f_{ij} = \begin{cases} 1 & v_j \in C(v_i) \\ 0 & \text{otherwise.} \end{cases}$$

This can be easily visualized if we change the order of rows and columns of $\mathbf{T}$ such that points in the same cluster have rows (and columns) that are adjacent to one other. This can be seen in Fig. 4.1 This transformed matrix will be a diagonal matrix. If the rows of this matrix $\mathbf{t}_1, \ldots \mathbf{t}_n$ can be written as a linear combination of k representative rows as,

$$\mathbf{t}_i = \sum_{j=1}^{k} \alpha_{i,j} \mathbf{t}_{\pi_j},$$

We can verify easily the following because the sum of each row of $\mathbf{T}$ is 1,

$$\sum_{j=1}^{k} \alpha_{i,j} = 1.$$

The linear combination factors are *convex combination factors*. The convex combination of the representative rows of the eigenvector matrix also happens to be the same. Let $\mathbf{X}$ be the eigenvector matrix with rows $\mathbf{x}_1, \ldots \mathbf{x}_n$ and columns $\mathbf{u}_1, \ldots \mathbf{u}_n$ as,

$$\begin{aligned} X_{i,l} &= \lambda_i^{-1} \mathbf{t}_i^{\intercal} \mathbf{u}_l \\ &= \lambda_i^{-1} \left[\sum_{j=1}^{k} \alpha_{i,j} \mathbf{t}_{\pi_j} \right]^{\intercal} \mathbf{u}_l \\ \Rightarrow \mathbf{x}_i &= \sum_{j=1}^{k} \alpha_{i,j} \mathbf{x}_{\pi_j}. \end{aligned} \tag{4.14}$$

Thus, we see in (4.14) that the eigenvectors lie in a simplex. Weber (2004) shows that pure-state transition matrices when perturbed approximately fall in a simplex and

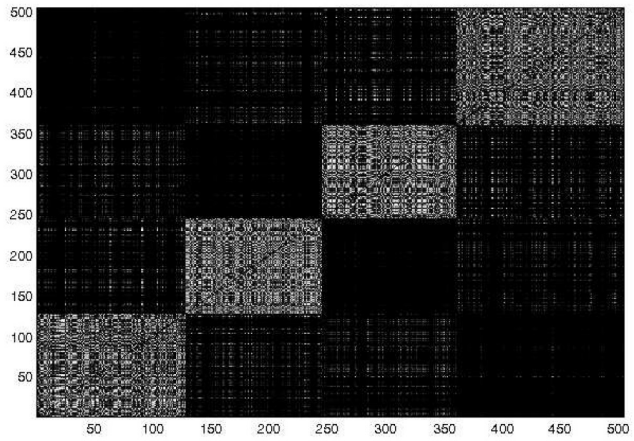

Figure 4.1: Laplacian or transition matrix when aligned such that adjacent points belong to the same cluster. Spectral clustering done with $k = 4$ on Ped A dataset. Lighter areas indicate higher values.

also suggests a Gram-Schmidt normalization procedure to find the furthest points or the vertices of the simplex in the sub-space. Further, if only the first k eigenvectors are considered, the final cluster membership mapping is a linear one defined as follows,

$$\begin{aligned} \mathbf{A}^{-1} &= \begin{pmatrix} X_{\pi_1,1} & \dots & X_{\pi_1,k} \\ \vdots & & \vdots \\ X_{\pi_k,1} & \dots & X_{\pi_k,k} \end{pmatrix}, \\ \chi &= \mathbf{XA}. \end{aligned} \tag{4.15}$$

Here $\chi_{i,j}$ indicates the probability of membership of point i to cluster j. An example of a simplex can be seen in Fig. 4.2 where clustering has been done on the Ped A dataset with $k = 4$. The resulting eigenspace is projected onto its top three principal components for visualization.

4.3.2 PCCA Algorithm

As in the case of the Normalized Cut Algorithm, the similarity matrix $\mathbf{S}$ is constructed from the data and Laplacian is constructed. The first k eigenvectors are computed.Kumar *et al.* (2013) proposes that the gap heuristic be used to determine the number of clusters.

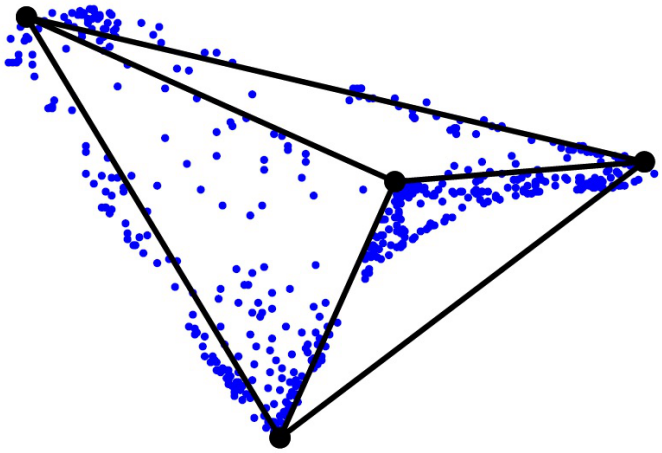

Figure 4.2: Simplex resulting from plotting eigenspace on top 3 principal components. The Ped A dataset has been clustered with $k = 4$.

The heuristic is that we choose the first k eigenvalues such that,

$$\frac{e_{k+1} - e_k}{1 - e_k} > t_c, \tag{4.16}$$

where e_i is the i-th eigenvalue and t_c is the *Spectral Gap Threshold.* k points of the data in $\mathbb{R}^k$ eigenspace now have to be chosen as vertices of a $k-1$ simplex. i.e. a complex hull in which all the data points lie. This is done by first choosing the point furthest away from the origin $\mathbf{x}_{\pi_1}$. Now we choose a point $\mathbf{x}_{\pi_2}$ furthest away from $\mathbf{x}_{\pi_1}$ and subsequent points are chosen if they are the farthest from the hyperplane formed by the joining the preceding points. Algorithm 6 details the procedure.

Algorithm 6 PCCA algorithm

Similarity matrix computed as earlier.

1. Construct $\mathbf{L}$ from similarity matrix $\mathbf{S}$ as: $\mathbf{L} = \mathbf{D}^{-1}\mathbf{S}$.
2. Compute the first k eigenvectors $\mathbf{u}_1, \ldots \mathbf{u}_k$ of $\mathbf{L}$ for which the gap of the largest k eigenvalues exceeds the Spectral Gap Threshold as given in 4.16. Form $\mathbf{X}$ from the eigenvectors as columns.
3. Define π_1 as the index for which $\|\mathbf{x}_{\pi_1}\|_2$ is maximum. Define $\gamma_1 = \text{span}(\mathbf{x}_{\pi_1})$.
4. For $i = 2, \ldots, k$: π_i is the index for which $\|\mathbf{x}_{\pi_i} - \gamma_{i-1}\|_2$ is maximum. $\gamma_i = \text{span}(\mathbf{x}_1, \ldots \mathbf{x}_i)$.

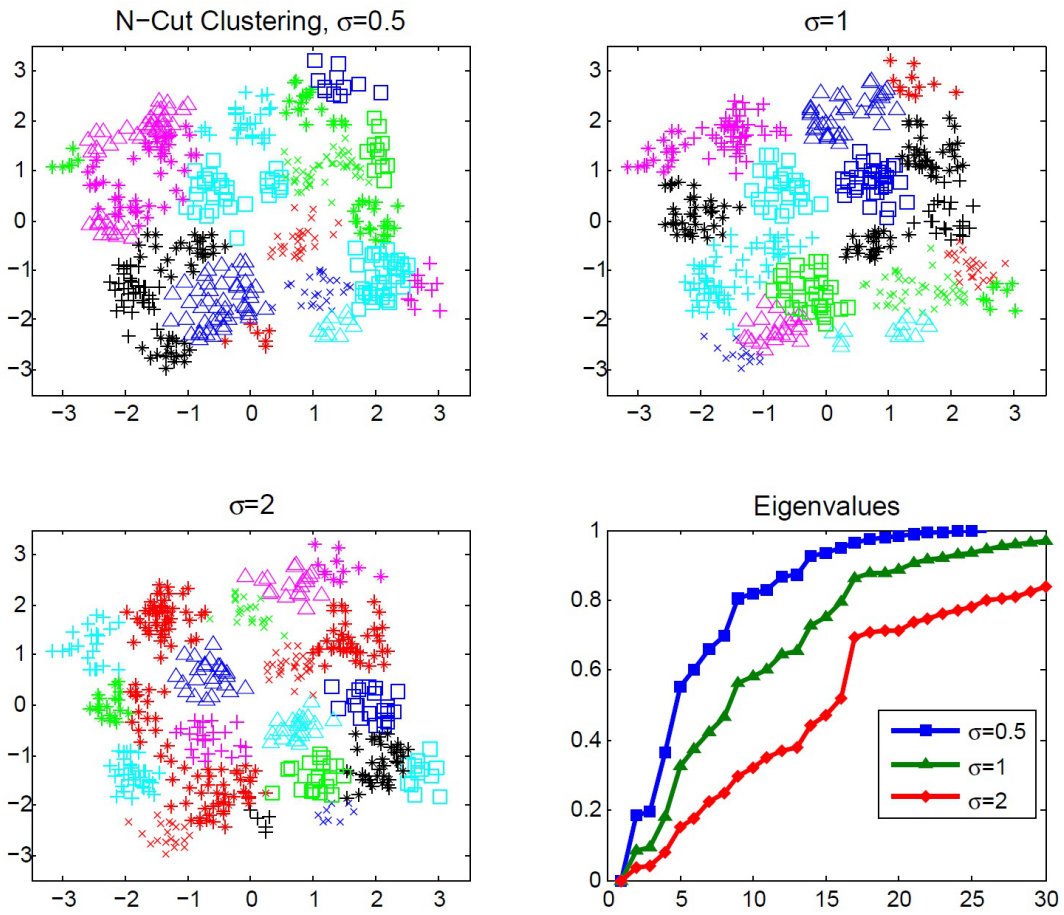

Figure 4.3: Normalized Cut clustering with $k = 16$ on the Ped A dataset for various values of σ. Its effect on the first k eigenvalues is seen in the graph at the bottom-right.

4.4 Results

4.4.1 Spectral Clustering

Fig. 4.3 shows the results of Normalized Cut spectral clustering for various various of σ used to construct the $\mathbf{S}$. As σ gets larger, only very close points get non-zero similarity measures. This may lead to a situation where the similarity measure cannot differentiate between a point in the adjacent cluster and another further away and thus multiple disjoint clusters may get mapped as one. Smaller σ leads to merging of neighbouring clusters.

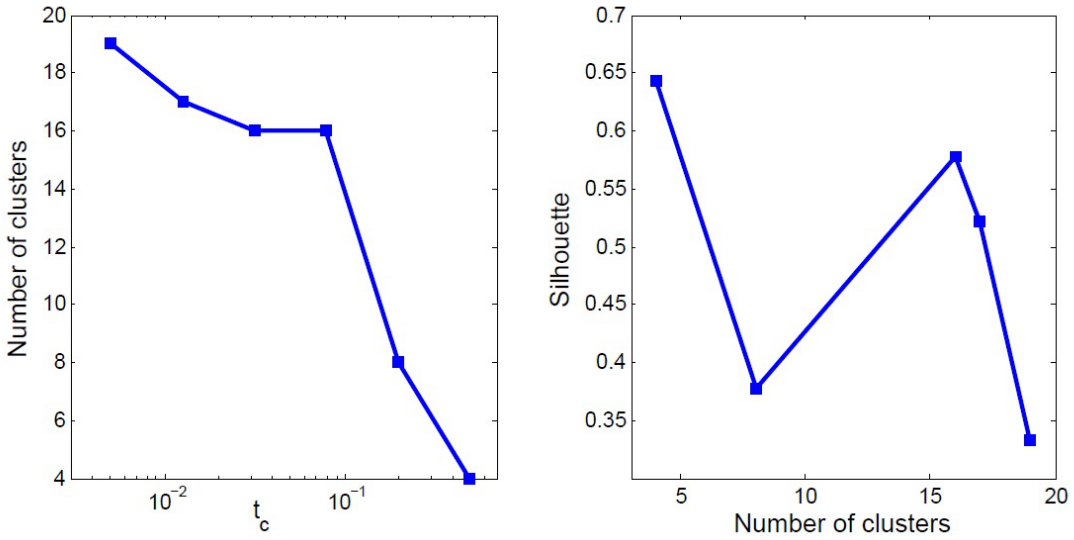

Figure 4.4: Cluster-order extraction of the PCCA algorithm using the eigengap heuristic with a quality metric. Ped A dataset was used.

4.4.2 PCCA algorithm

The sensitivity of the number of clusters chosen on the basis of the heuristic mentioned in 4.16 is shown in Fig. 4.4 along with a quality measure. The automatic extraction of the cluster-order from the eigenvalues is not a trivial procedure. Fig. 4.5 and Fig. 4.6 show the performance of the PCCA algorithm on the Ped A and Ped B datasets. Clearly, this algorithm outperforms Normalized Cut spectral clustering. This is because clustering in the eigenspace takes into consideration the structural properties of the spectral sub-space. The plot of the eigenvalues serves to again highlight the challenge in choosing k, the knee point after which the eigenvalues floor.

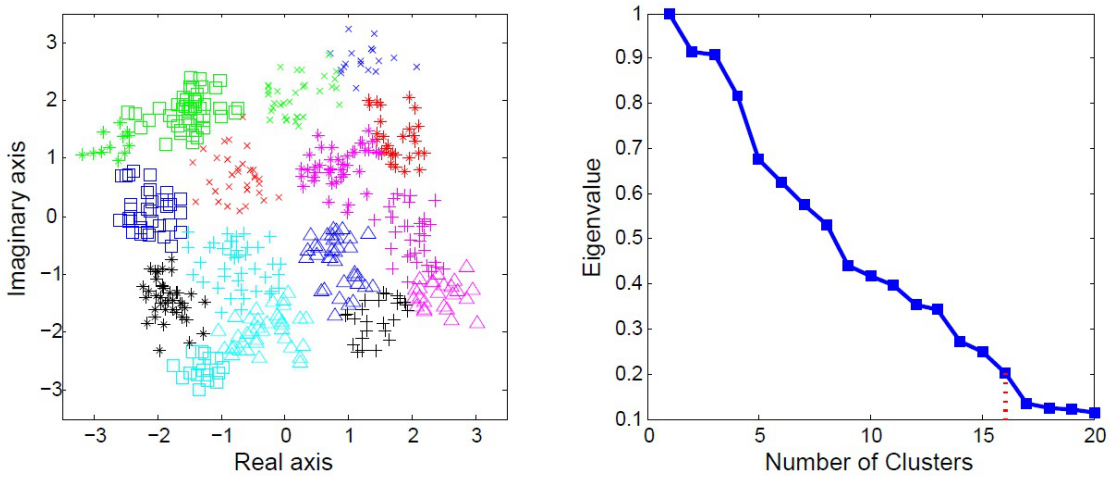

Figure 4.5: PCCA clustering with $k = 16$ on the Ped A dataset with a plot of the eigenvalues. The red line indicates the cluster order chosen after which the eigenvalues floor.

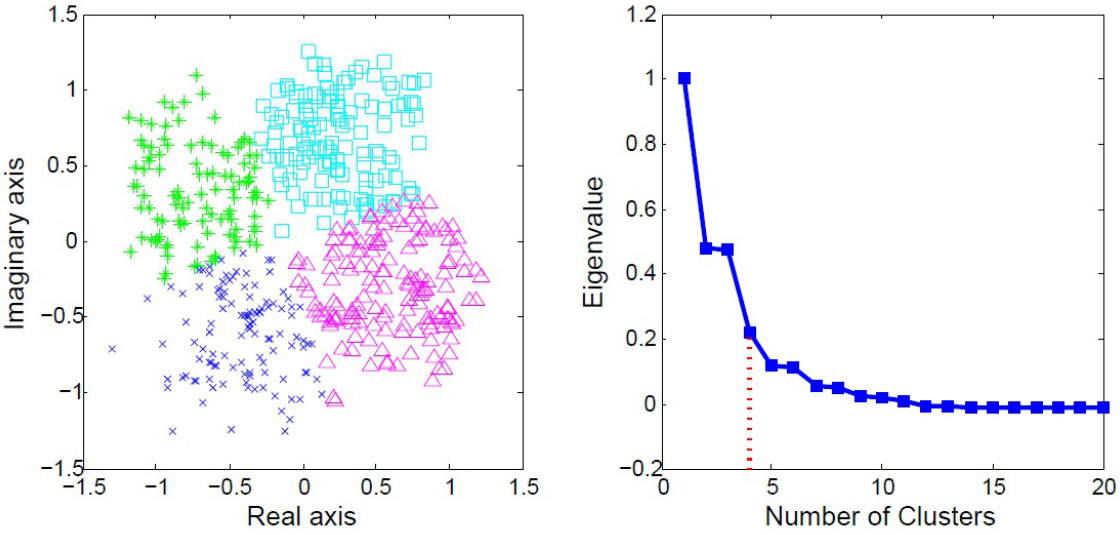

Figure 4.6: PCCA clustering on the Ped B dataset with $k = 4$ with a plot of the top eigenvalues.

CHAPTER 5

Optimal Non-Linear Receivers

5.1 Decoding procedure

The datapoints are first passed through a cluster-order extraction algorithm and are clustered based on this order. A GMM is initialized with this clustering through two approaches - initializing the EM algorithm with the means and variances of points in each cluster with the probability of a component being proportional to the number of points or directly initializing a GMM with the above parameters. After the pdf of the interference plus noise is obtained, the ML estimate for the desired signal is obtained by finding the maximum value on the pdf of the points as given below,

$$\hat{x} = \arg\max_{x} p(y - h_1 x).$$

In the simulations, multiple channel realizations (~ 100) are used to generate the average BER plots.

5.2 Single Antenna Case

Fig. 5.1 presents the BER vs. SNR plots when various clustering algorithms are utilized. The system in this case involves 2 QPSK interferers at 0dB and -3dB transmitting on the Ped A channel. A number of observations can be made:

1. For all the clustering algorithms where the GMM could be initialized in both ways as mentioned in the previous section, there is not a significant difference in performance between the computationally expensive procedure of initializing the EM algorithm or directly instantiating a GMM. This can be seen in the BER vs. SNR plots of the Hierarchical clustering algorithms.

2. The cluster-order that is extracted plays a significant role in the determination of performance. Cluster-extraction through the eigengap metric is not yielding the right number of clusters as can be seen in the graph on the bottom right and this

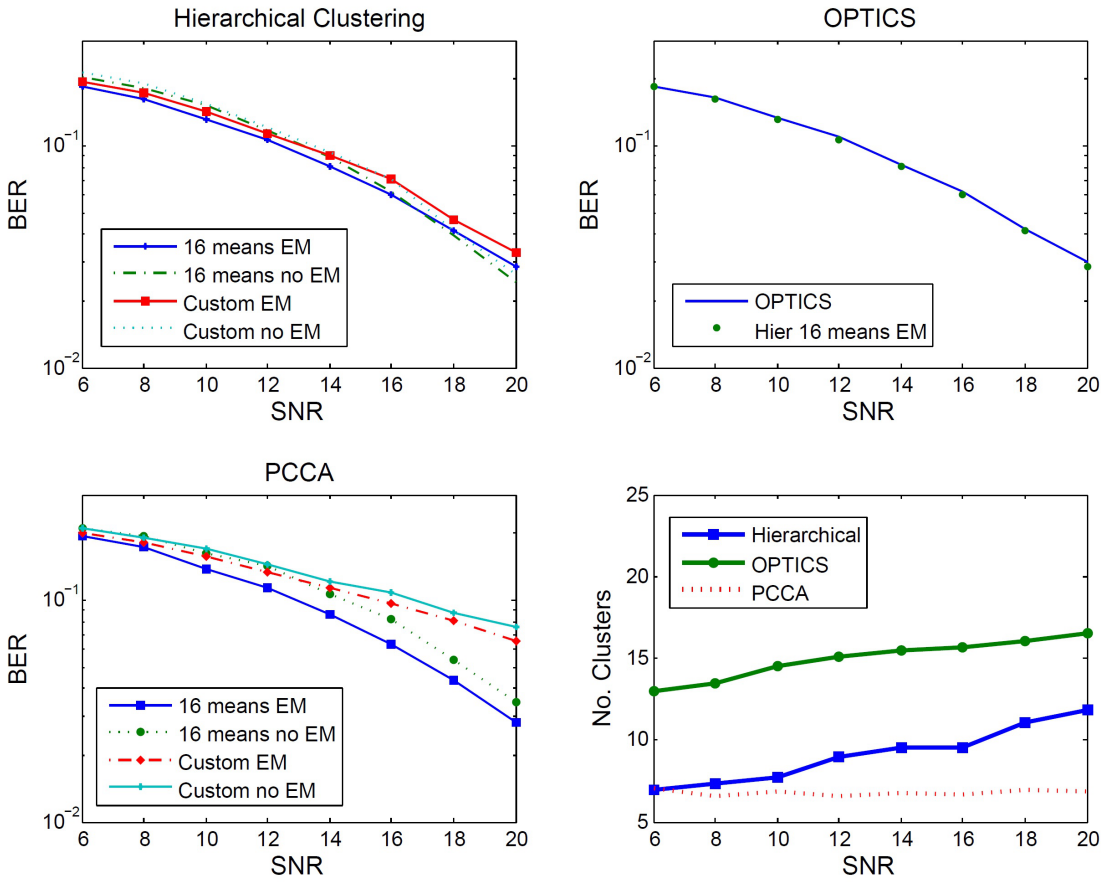

Figure 5.1: BER vs SNR plots of various clustering approaches used in the decoding algorithm for a Ped A channel with 2 QPSK interferers at 0dB and -3 dB.

flattens the performance of the decoder based on PCCA. Also, the performance of the approach of directly instantiating a GMM is slightly worse than initializing the EM algorithm as the clusters identified by the PCCA can be of non-convex shape. Estimating the parameter σ is again difficult, impacting the final clustering and performance here.

3. The OPTICS algorithm is seen to match the performance of Hierarchical Clustering given that there are 16 cluster centres, the ideal case. Thus the problem of cluster extraction seems to be appropriately handled.

4. Hierarchical clustering even if it extracts the wrong cluster order is seen to perform just as well as the ideal case of 16 clusters. This shows that there is a wide tolerance band of the cluster order in terms of performance.

Fig. 5.2 compares the above system (Case 1) with a similar system but without a distinctive cluster structure. The second system has 2 interferers - one from a 16 QAM alphabet and the other from QPSK with powers 0db and -3 dB respectively transmitting on the Ped A channel. Clearly, the receiver which in this case uses the Hierarchical clustering algorithm floors as the main bottleneck here is the interference profile.

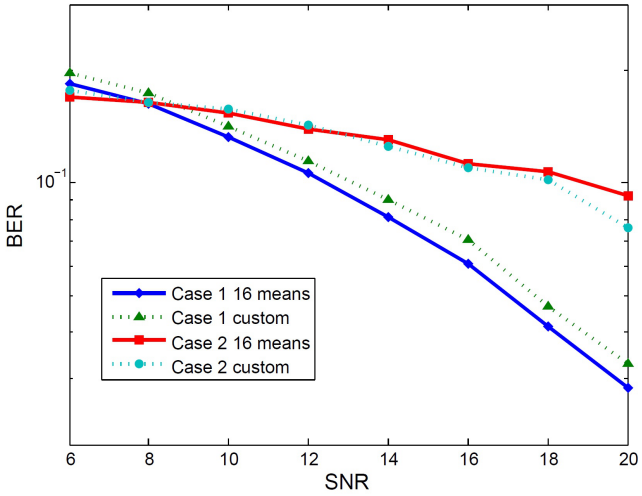

Figure 5.2: Performance comparison of the single antenna decoder with hierarchical clustering on case 1 with a distinct clustering structure and case 2, a similar system but with a 16 QAM interferer lacking any distinctive cluster structure.

5.3 Multiple Antenna Case

In the system model it is assumed that the receiver gets independent channels at each of its antennae. There are several ways of using receive diversity in decoding, a few are listed below:

1. Running the clustering algorithms separately to determine the cluster-order. Exchange this information and run the algorithm on both with maximum of the cluster-orders. The soft decoded outputs can be combined in series - by taking the average assigning equal importance to both cases, a weighted average assigning more importance to the clustering with higher quality or by taking the product. Taking the product is akin to finding the probability that both the data streams are independently clustered to be the same.

2. Doubling the dimension and running the clustering algorithm on this larger space. This method is quicker than the previous one as the clustering algorithm is run only once.

Fig. 5.3 compares the performance of approaches 1 and 2. In both cases, the system (Case 1) described in the previous section is used. On the left side, as more antennas are used, the soft outputs are averaged equally and this offsets poor clustering and GMM fitting at say one of the antennae. Predictably, the performance improves as more antennas are added. However, it is very clear that doubling the dimension is far more optimal and yields performance that is orders of magnitude better as the number of antennae are

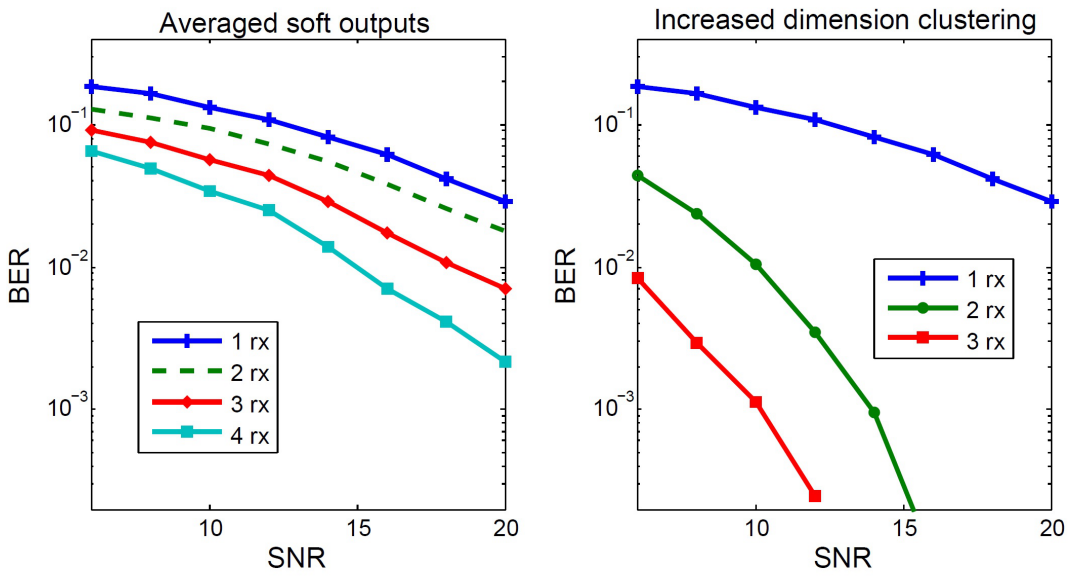

Figure 5.3: Multiple antenna approaches for the heirarchical clustering algorithm - averaging the outputs and doubling dimension. The system used in Fig. 5.1 is used.

increased. This could be because the distance between 16 clusters spread across four dimensions is much larger improving cluster separation and cohesion. Fig. 5.4 shows the performance of the case whether the hierarchical clustering algorithm extracts the cluster order, shares the information with the other antennas and then averages the soft-outputs after clustering with higher k. Clearly, as the number of antennas increase, the maximum of the cluster-order estimated by the various antennas approaches the optimal figure of 16. The performance of this method is not significantly worse than the method used in Fig. 5.3 where we had supplied the cluster-order of 16.

Fig. 5.5 corroborates the above observations even with the OPTICS algorithm. It is far more optimal to double the dimension and perform clustering rather than combining the soft-outputs after clustering in each antennae.

Finally in Fig. 5.6, all the multi-antenna approaches have been simulated with the hierarchical clustering algorithm. EM algorithm has not been used to initialize the GMM. The following are some observations:

1. The interferer data $\mathbf{x} \in \chi_1 \times \chi_2 \times \ldots \chi_n$ for each point determines the clustering label in both antennas and both should be similar.

2. Averaging the soft outputs or performing a weighted average based on the quality of clustering (Silhouette metric) does not yield any difference.

3. Reducing the dimension of the combined signal space by the principal component

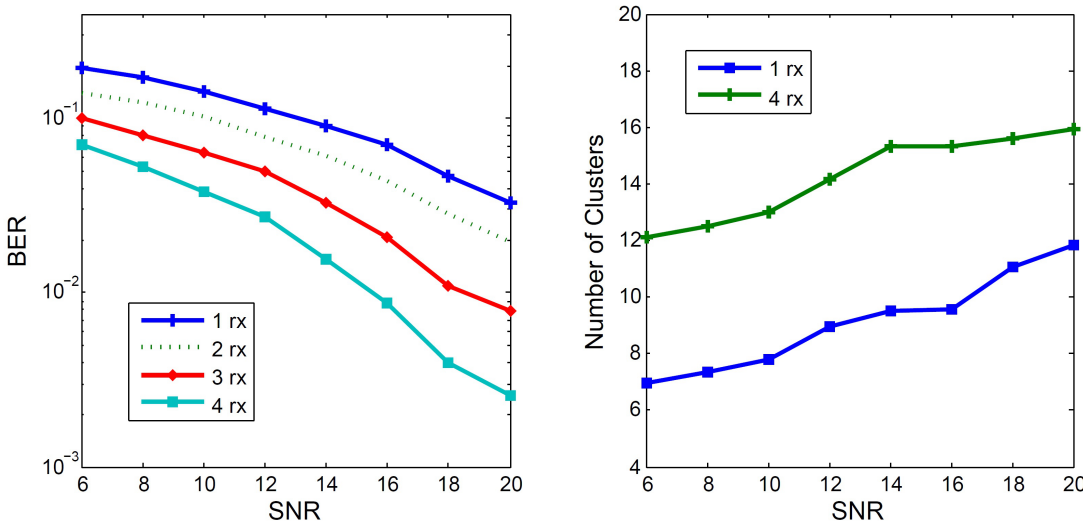

Figure 5.4: Hierarchical clustering used for cluster extraction from multiple antennas. The order is shared among the antennas and the maximum is chosen and clustering done again. Finally the soft-outputs are added.

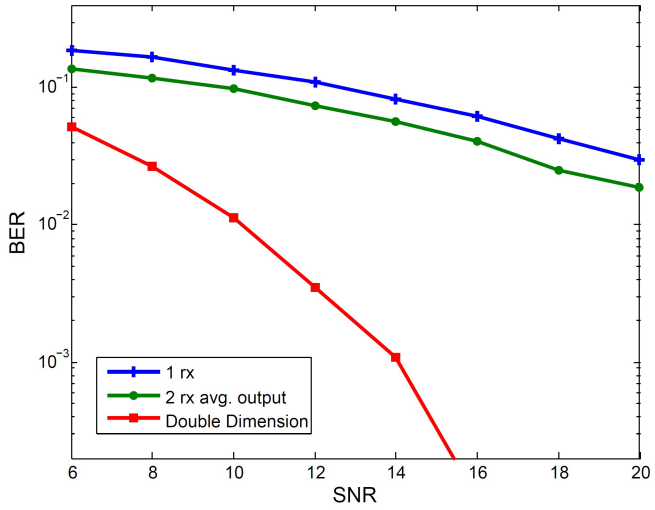

Figure 5.5: Performance of multi-antenna approaches using the OPTICS algorithm. The Ped A system with 2 QPSK interferers is used.

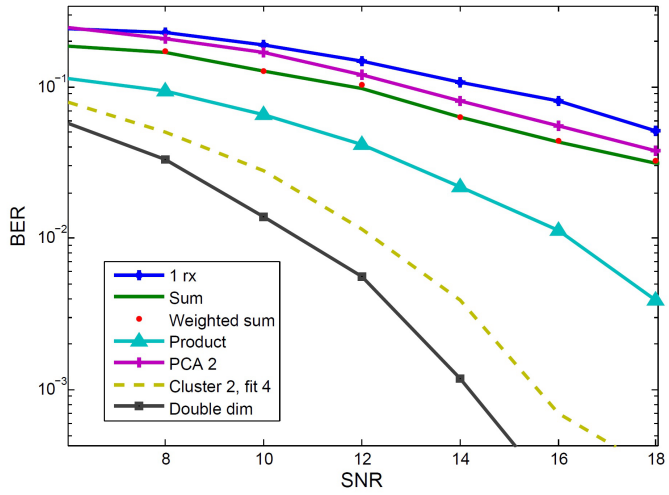

Figure 5.6: Various multi-antenna approaches with hierarchical clustering done with $k = 16$.

method brings only a small improvement over the single antenna case. In both cases a GMM is fit over a two dimensional subspace.

4. Taking the product of the soft-outputs has a performance that falls in between dimension doubling methods and methods averaging individual clustering performance on the clusters. Taking the product ensures that only points which have been wrongly clustered in both antennas will be incorrectly clustered overall.

5. Performing clustering on a single antenna and then doubling the dimension and fitting a pdf on the four dimensional point based on the single antenna cluster labels does not carry a significant penalty in performance from clustering directly on the doubled dimension. Some saving is achieved in the distance computation to generate the similarity matrix here.

CHAPTER 6

Conclusions

6.1 Summary

The non-Gaussian nature of the interference plus noise rendered popular techniques such as LMMSE ineffective and necessitated the need for more sophisticated interference aware mitigation techniques. The Expectation Maximization algorithm was studied but a problem in implementation was estimating the number of components.

Clustering approaches were studied as a means to determine the cluster-order in the system. The Hierarchical algorithm coupled with a quality of clustering metric were first explored. Quality of clustering metrics largely depend on the data from the system and for wireless systems with convex clusters, the Silhouette approach was observed to be more accurate and computationally efficient than the others. The OPTICS clustering algorithm, an implementation of density based clustering, was seen to the most accurate in terms of cluster-order detection. While being computationally cheaper than the other approaches, it classifies a lot of points as noise and hence can only be used to initialize the EM algorithm as approached to the faster method of initializing a GMM directly on the data. Finally, the eigenspace of the Similarity matrix was studied and approaches such as Normalized Cut and Perron Cluster Analysis was studied but the estimation of cluster order was seen to be a challenge.

The performance of this non-linear ML receiver was seen not to floor only in cases with a clear clustering structure. Certain interference profiles such as two dissimilar QPSK interferers was seen to have better error performance than two QPSK interferers of the same magnitude because it has a more apparent clustering structure. Notably, the performance of the much cheaper approach of initializing a GMM directly from the clusters is seen to not differ significantly from the method of initializing the parameters of the EM algorithm such as the means and variances when the clusters indicated are convex in shape. The optimal approach in multiple-antennas receivers is to double the dimension of the data by merging data-streams. With all clustering approaches, this

method is not only more efficient but the increased separation between clusters due to the fact that the dimension is increased results in enhanced accuracy.

6.2 Future Work

In this thesis, the channel was considered to be static in time and only a few sub-carriers wide in frequency. Exploring how the results from the previous clustering run can be used to either speed up computations in this run or improve accuracy can be studied. Means to track the slowly varying PDF of the interference and noise can be useful and save calculation. Only the performance of the uncoded data is studied. How the performance of the channel varies with coding will be useful to study.

In the thesis a clustering run in performed on around 500 points, a number chosen based on the time period in which the Doppler does not play a significant role and keeping in mind the expected cluster structure. However, the performance of the clustering and the ML detector as a function on the volume of data is critical as it allows for intelligent design of pilot signals and bootstrapping methods. In the multiple antenna approach, the channels are considered to be statistically independent which may not be accurate in all cases. Performance of various clustering approaches and combining them when the channels are correlated will be useful to study.

REFERENCES

1. **Ankerst, M.**, **M. M. Breunig**, **H.-P. Kriegel**, and **J. Sander**, Optics: ordering points to identify the clustering structure. *In Proceedings of the 1999 ACM SIGMOD international conference on Management of data*. ACM, 1999.

2. **Arthur, D.** and **S. Vassilvitskii**, k-means++: the advantages of careful seeding. *In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2007.

3. **Ben-Hur, A.**, **A. Elisseeff**, **I. Guyon**, *et al.* (2002). A stability based method for discovering structure in clustered data. *Pacific symposium on biocomputing*, **7**(6), 6–17.

4. **Bishop, C.**, *Pattern Recognition and Machine Learning*. Springer, 2006.

5. **Cadambe, V.** and **S. Jafar**, Interference alignment and spatial degrees of freedom for the k user interference channel. *In Communications, 2008. ICC '08. IEEE International Conference on*. 2008.

6. **Calinski, R. B.** and **J. Harabasz** (1974). A dendrite method for cluster analysis. *Communications in Statistics*, **3**, 1–27.

7. **Cho, Y. S.**, **J. Kim**, **W. Y. Yang**, and **C.-G. Kang**, *MIMO-OFDM Wireless Communications with MATLAB*. John Wiley & Sons, 2010.

8. **Dasgupta, S.** and **P. M. Long** (2005). Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.*, **70**(4), 555–569.

9. **Defays, D.** (1977). An efficient algorithm for a complete link method. *The Computer Journal*, **20**(4), 364–366.

10. **Ester, M.**, **H. P. Kriegel**, **J. Sander**, and **X. Xu**, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *In Second International Conference on Knowledge Discovery and Data Mining*. 1996.

11. **Everitt, B. S.**, **S. Landau**, and **M. Leese**, *Cluster Analysis*. Wiley Publishing, 2009, 4th edition.

12. **Fortunato, S.** and **M. Barthélemy** (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, **104**(1), 36–41.

13. **Gordon, A.**, *Classification, 2nd edn*. Chapman and Hall-CRC, 1999.

14. **Hagen, L.** and **A. B. Kahng** (1992). New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **11**(9), 1074–1085.

15. **Hartigan, J.**, *Clustering Algorithms*. Wiley, 1975.

16. **Jain, R.** (2007). Channel models a tutorial. URL `www.cse.wustl.edu/~jain/cse574-08/ftp/channel_model_tutorial.pdfâĂŐ`.

17. **Kaufman, L.** and **P. Rousseeuw**, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

18. **Kumar, P.**, **L. Niveditha**, and **B. Ravindran** (2013). Spectral Clustering as Mapping to a Simplex.

19. **Meila, M.** and **J. Shi**, A random walks view of spectral segmentation. 2001.

20. **Newman, M. E. J.** (2006). Finding community structure in networks using the eigenvectors of matrices. phys. *Rev. E*, 036104.

21. **Shi, J.** and **J. Malik** (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(8), 888–905.

22. **Tibshirani, R.**, **G. Walther**, and **T. Hastie** (2001). Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **63**, 411–423.

23. **Tse, D.** and **P. Viswanath**, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

24. **Vaishnavi, J.** (2012). *Advanced Receiver Techniques based on Density Estimation for Inter-cell Interference Mitigation*. Master's thesis, Department of Electrical Engineering, IIT-Madras.

25. **Von Luxburg, U.** (2007). A tutorial on spectral clustering. *Statistics and computing*, **17**(4), 395–416.

26. **Weber, M.** (2004). Clustering by using a simplex structure. Technical Report ZR-04-03, Konrad-Zuse-Zentrum Berlin.

27. **Weber, M.**, **W. Rungsarityotin**, and **A. Schliep** (2004). Perron Cluster Analysis and its Connection to Graph Partitioning for Noisy Data. *ZIB Report*, **04-39**.