Progress report: Learning for Link Adaptation

Oluwasanmi Koyejo

CS395T - Data Mining

Abstract

The aim of this report is to provide motivation and document current progress towards the design of a learning based link adaptation algorithm for wireless channels that exploits all available channel information and can adapt to specific user channels. I seek to design a sparse algorithm to minimize the computational cost associated with channel classification. I accomplish this is in three stages using various clustering and classification techniques. I also examine the choice of *distance* metric and its effect on complexity and algorithm performance.

I. Introduction

In a wireless communication system, *Link Adaptation* (LA) is the technique of modifying the transmission scheme according to the measured wireless channel and/or its statistics [1]. In general, this is a very difficult problem because the cost function is often difficult to design and solve analytically. For this reason, link adaptation decisions are often made using expensive brute force searches over the relevant parameter space or heuristics based on domain knowledge. The effect of user specific channels can be captured using training data. Unfortunately, many link adaptation techniques proposed do not utilize a training phase and make decisions based only on models of the wireless channel [1]–[4].

The algorithm I propose is designed using standard assumptions about the channel model. However, minimal assumptions are made about the correlation structure of the channel. The training information is used to augment this domain knowledge: this leads to improved performance by utilizing user specific training information before link adaptation decisions are made. The algorithm also utilizes all available channel information for making link adaptation decisions. This is in contrast to LA algorithms that use a pre-processed feature space such as the condition number of the channel matrix; leading to some loss of potentially discriminative information.

This report is organized as follows. In section II, I present the instantaneous and stochastic multiple input, multiple output (MIMO) channel model. Next, I discuss the proposed algorithm (section III) which is composed of three sub-algorithms: unsupervised clustering (section III-C), optimization of MIMO scheme for clusters (section III-D) and eventually sparse

classification (section III-E). I discuss preliminary simulation results in section IV and discuss proposed analytical and simulation work (section V). I conclude the report in section VI.

A. Notation

I denote $\mathcal{CN}_p(\mu, \mathbf{\Phi})$ as a p-variate complex Gaussian distribution with mean $\mu \in \mathbb{C}^{p\times 1}$ and covariance matrix $\mathbf{\Phi} \in \mathbb{C}^{p\times p} > 0$. The matrix $\mathbf{X} \sim \mathcal{CN}_{p,q}(\mathbf{\Upsilon}, \mathbf{\Psi}, \mathbf{\Phi})$ is the complex matrix Gaussian random variable in $\mathbb{C}^{p\times q}$. It is equivalent in distribution to $\text{vec}(\mathbf{X}^\dagger) \sim \mathcal{CN}_{pq}(\text{vec}(\mathbf{\Upsilon}^\dagger), \mathbf{\Phi} \otimes \mathbf{\Psi})$ [5]. $(\cdot)^*$ denotes conjugation, $(\cdot)^T$ denotes the transpose and $(\cdot)^\dagger$ denotes the hermitian transpose. $|\cdot|$ denotes the determinant and $\|\cdot\|_F$ denotes the Frobenius norm. $\|\cdot\|_p$ denotes the p norm and $tr(\cdot)$ denotes the trace of a matrix. $(\cdot)^{\frac{1}{2}}$ denotes the hermitian square root. Finally, $\mathbf{I}_p$ is the $p \times p$ identity matrix and $\mathbf{0}_p$ is the $p \times p$ matrix of all 0's.

II. System Model

In a narrow-band wireless MIMO system with N_t transmit antennas and N_r receive antennas, the system can be modeled by:

$$\mathbf{y} = \sqrt{\frac{E_s}{N_o}}\mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{y} \in \mathbb{C}^{N_r\times 1}$ is the received signal vector, $\mathbf{x} \in \mathbb{C}^{N_t\times 1}$ is the transmitted signal vector with the power constraint $\mathcal{E}\{\|\mathbf{x}\|_2^2\} = N_t$. $\mathbf{H} \in \mathbb{C}^{N_r\times N_t}$ is the channel matrix where its (i,j)th entry contains the complex fading parameter between the jth transmit antenna and the ith receive antenna. The signal to noise ratio (SNR) is $\gamma_o = \frac{E_s}{N_o}$ and $\mathbf{n} \in \mathbb{C}^{N_r\times 1}$ is zero mean additive Gaussian noise vector with covariance matrix $\mathcal{E}\{\mathbf{n}\mathbf{n}^\dagger\} = N_o\mathbf{I}_{N_r}$.

In the instantaneous channel case, the channel can be decomposed using the singular value decomposition (SVD) as:

$$\mathbf{H} = \mathbf{U}_h\mathbf{\Sigma}_h\mathbf{V}^\dagger$$

$$\mathbf{H}\mathbf{H}^\dagger = \mathbf{U}_h\mathbf{\Lambda}_h\mathbf{U}_h^\dagger \tag{2}$$

$$\mathcal{H} = \gamma_o\mathbf{H}\mathbf{H}^\dagger \tag{3}$$

where $\mathbf{\Lambda}_h$ are the eigenvalues of $\mathbf{H}\mathbf{H}^\dagger$, $\mathbf{\Lambda}_h = \mathbf{\Sigma}_h^2$ and $\mathcal{H}$ is the combined channel and SNR (for joint adaptation).

Each row of $\mathbf{H}$ is correlated with correlation matrix matrix $\mathbf{R_t}$ (transmit antenna correlation) and each column of H is correlated with correlation matrix $\mathbf{R_r}$ (receive antenna correlation). This model has been motivated rigorously in the literature (see [1] and references therein). In this scenario, the the channel matrix $\mathbf{H}$ can be decomposed as:

$$\mathbf{H} = \sqrt{a}\mathbf{M} + \sqrt{b}\mathbf{R_r}^{\frac{1}{2}}\mathbf{H}_w\mathbf{R_t}^{\frac{1}{2}}$$

$$\sim \mathcal{CN}_{N_r,N_t}(\sqrt{a}\mathbf{M}, b\mathbf{R_r} \otimes \mathbf{R_t}) \tag{4}$$

where a and b denote the appropriate power normalization factors[1]. This formulation implies that the correlation matrices are normalized ($\mathcal{E}(\|\mathbf{H}-\mathbf{M}\|_F^2) = N_r N_t$). I can further decompose the correlation matrices as:

$$\mathbf{R_t} = \mathbf{U}_t \mathbf{\Lambda}_t \mathbf{U}_t^\dagger$$

$$\mathbf{R_r} = \mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^\dagger \tag{5}$$

I assume the presence of an error-free feedback channel so the transmitter receives updates of either the instantaneous channel $\mathbf{H}$ or the long term channel statistics $\{\mathbf{M}, \mathbf{R_t}, \mathbf{R_r}\}$. In order to make link adaptation decisions[2].

III. Learning the optimal MIMO scheme

As mentioned in the introduction, the motivation for this algorithm is sparsity. Link adaptation decisions have to be made on the basis of channel realizations. However, in all but the most trivial cases, finding the optimal transmission scheme given a channel realization can be very computationally expensive (see section III-D). In order to reduce system complexity, I aim to design a sparse solution by first clustering the channel realizations to reduce the number of brute force searches needed for training, then designing sparse classifiers to eliminate the need for new searches for new channel realizations.

The total number of training objects is n. k is the number of clusters and m is the number of classes (transmission schemes available). Since our aim is to utilize the entire feature space for classification, the algorithm should adapt to observed realizations (or distributions) of $\mathbf{H}$. This can be done by defining an appropriate distance measure that captures all the information in $\mathbf{H}$.

A. Distance measures for instantaneous channel

In the instantaneous channel, the distance measure most commonly used is simply the euclidean distance between the values of $\mathrm{vec}(\mathcal{H})$. I define this distance as

$$\mathbf{D}_{vecnorm}(\mathcal{H}_1, \mathcal{H}_2) = \|\mathrm{vec}(\mathcal{H}_1) - \mathrm{vec}(\mathcal{H}_2)\|_F^2 \tag{6}$$

This norm is the product of two d^2 dimensional vectors and so the computational complexity of computing this distance is $\mathcal{O}(d^2)$.

Research into the performance of wireless channels has shown that capacity can be achieved by optimizing transmission on the eigenvalues of $\mathcal{H}$ [1]. For this reason, the measure of similarity used should be based on the distance between the channel

[1]In the ricean channel model, $a = \frac{K}{K+1}$ and $b = \frac{1}{K+1}\frac{\gamma_o}{N_t}$ with a specified K factor

[2]The formulation also allows for decision making at the receiver where only the MCS scheme index is fed back to the transmitter.

eigenvalues. One distance measure that takes this into account is the Burg matrix divergence[3] [7] which can be defined as:

$$\begin{aligned}\mathbf{D}_{burg}(\mathcal{H}_1, \mathcal{H}_2) &= tr(\mathcal{H}_1\mathcal{H}_2^{-1}) - log|\mathcal{H}_1\mathcal{H}_2^{-1}| - d \\ &= \sum_i \sum_j \frac{\lambda_i}{\gamma_j}(\mathbf{v}_i^\dagger \mathbf{w}_j)^2 - \sum_i log\frac{\lambda_i}{\gamma_i} - d \end{aligned} \tag{7}$$

where $d \times d$ is the dimension of $\mathcal{H}$(here $d = N_t$), $\{\lambda_i, \gamma_i\}$ are the eigenvalues of $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively, and $\{\mathbf{v}_i, \mathbf{w}_i\}$ are the corresponding eigenvectors. Note that this distortion is non-symmetric. The computation of this distance measure is dominated by the inverse and the determinant (or the eigenvalue decomposition) and so is of $\mathcal{O}(d^3)$.

The complexity of this computation can reduced while retaining the essential information (effects of eigenvalues) by using only the trace measure which has a complexity of $\mathcal{O}(d^2)$ as shown in (8).

$$\begin{aligned}\mathbf{D}_{tr}(\mathcal{H}_1, \mathcal{H}_2) &= tr(\mathcal{H}_1\mathcal{H}_2) \\ &= \sum_i \sum_j \mathcal{H}_1^{(1,j)}\mathcal{H}_2^{(1,j)} \end{aligned} \tag{8}$$

where $\mathbf{X}^{(i,j)}$ denotes the (i, j) the entry of $\mathbf{X}$. In this report, $\mathbf{D}_{tr}$ is only used for classification, but it will be implemented for clustering in the final report (see section V).

B. Distance measures for the stochastic channel

The MIMO wireless channel is well modeled by the matrix Gaussian distribution (as given by equation (4)). This introduces a natural question: What is the best notion of similarity between Gaussian distributed random variables? For Gaussian random vectors, the Kullback Leibler divergence (KL) is a well established distance measure. The authors in [8] also show that it can be expressed as a sum of Bregman divergences leading to a *natural* distance measure between Gaussian random vectors. The matrix Gaussian distribution can also be expressed in an equivalent vector Gaussian form (for details see section I-A and [5]). Therefore, the KL divergence can be used directly on matrix Gaussians. The KL divergence between two multivariate Gaussian $\mathbf{x}$ and $\mathbf{y}$ can be defined given their means: $\{\mathbf{m}, \mu\}$ and covariances: $\{\mathbf{S}, \mathbf{\Sigma}\}$ as:

$$\begin{aligned}\mathbf{D}_{KL}(p(\mathbf{x}|\mathbf{m}, \mathbf{S})\|p(\mathbf{x}|\mu, \mathbf{\Sigma})) &= \frac{1}{2}(tr(\mathbf{S}\mathbf{\Sigma}^{-1}) - log|\mathbf{S}\mathbf{\Sigma}^{-1}| - d) + \frac{1}{2}(\mathbf{m} - \mu)^\dagger \mathbf{\Sigma}^{-1}(\mathbf{m} - \mu) \\ &= \frac{1}{2}D_{burg}(\mathbf{S}, \mathbf{\Sigma}) + \frac{1}{2}\mathbf{M}_{\mathbf{\Sigma}^{-1}}(\mathbf{m}, \mu) \end{aligned} \tag{9}$$

where $\mathbf{M}_A$ is the Mahalanobis distance parametrized by the matrix $\mathbf{A}$ and defined as

$$\mathbf{M}_{\mathbf{A}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\dagger \mathbf{A}(\mathbf{x} - \mathbf{y}) \tag{10}$$

Note that the KL divergence is a non-symmetric measure.

[3]See [6] for a description of Bregman divergences and some applications to clustering

In the simple case considered in this report, I also compute distances between covariances using the $\mathbf{D}_{tr}$ distance. More details can be found in section IV.

C. Unsupervised Clustering

The aim of the unsupervised clustering algorithm is to reduce the search space required for the optimal classification of channels. Given a collection of objects (channels) and a distance metric between these objects, a clustering algorithm can be designed based on the k-means algorithm [8].

On initialization, the algorithm randomly assigns objects to clusters. where π_i denotes the assigned cluster of object i. Next the mean of each cluster is computed using a suitable distance measure. For instance, the distance measure in the Gaussian stochastic case is chosen to be the KL divergence. After the means are updated, π is recalculated by computing the distance of each object to each cluster center and re-assigning each object to its closest center. This process of computing centers and reassigning members is repeated until none of the cluster assignments change or a specified objective function (such as the sum of distances between objects and centers) does not change significantly.

In general clustering is a NP-hard problem with no closed form solutions. However, k-means has been shown to converge within a few iterations even with high dimensional data. Suppose each distance calculation takes $\mathcal{O}(\alpha(\mathbf{D}))$ complexity, each iteration can be shown to be of $\mathcal{O}(\alpha(\mathbf{D})nk)$ complexity, where n, k are the number of objects and number of clusters respectively. If the clustering takes place in τ steps, the computation of the full algorithm will be of $\mathcal{O}(\alpha(\mathbf{D})nk\tau)$.

In the instantaneous channel, the objects are the channel matrices $\mathcal{H}$. these are clustered using both $\mathbf{D}_{Burg}(\cdot,\cdot)$ and $\mathbf{D}_{vecnorm}(\cdot,\cdot)$. For the stochastic case, we refer to the work in [8]. In this case, the channel correlations are clustered using $\mathbf{D}_{KL}(\cdot,\cdot)$. the authors in [8] show that this measure can lead to very good representative objects (as measured by normalized mutual information). Further details on the k-means algorithm and various optimizations can be found in [9]–[11] and references therein.

If unsupervised clustering based training is not desired, training data can still be utilized to create sparsity in the channel representation. This can be done by defining *typical* channels based on domain knowledge. Given n such channels and training data, we can find the subset of k channels that best represent the user space. This can be done using a k-nearest neighbor (k-nn) type solution. Using training data, we can compute the distance between each training channel and the set of optimal channel representatives. We can then discard the $(n-k)$ representative channels least describe the training data; we can discard the representative channels that are farthest away from the training data, or equivalently, we can assign each training channel to its closest representative channel, and discard the representative channels with the fewest assignments.

D. Brute Force Search: Choosing the optimal MIMO transmission scheme

The most critical sub-algorithm for our link adaptation formulation is computing the optimal transmission scheme given a channel realization (or channel statistics). This is the fundamental link adaptation problem. This can be achieved in several ways. In this section, I outline a few of these methods and discuss come advantages and disadvantages.

The fundamental measure of wireless system performance is the rate of transmission. this can be generalized to the notion of capacity. The channel capacity can be defined as the maximum error free rate that can be supported by a given wireless channel. It is given by the maximum mutual information between the input and output of the channel. In general, it can be expressed as:

$$\begin{aligned}\mathcal{C} &= \max_{Q:tr(Q)<E_s} \mathcal{I}(\mathbf{X};\mathbf{Y}) \\ &= \max_{Q:tr(Q)<E_s} logdet(I_{N_r} + \frac{\mathbf{HQH}^\dagger}{N_o}) = \max_{Q:tr(Q)<E_s} logdet(I_{N_r} + \frac{\mathbf{QH}^\dagger\mathbf{H}}{N_o})\end{aligned} \tag{11}$$

where $\mathbf{Q} = \mathcal{E}\{\mathbf{x}\mathbf{x}^\dagger\}$ is the correlation between the transmitted symbols. Note that the maximization can also be carried out over $\hat{\mathbf{Q}} = \mathbf{U}_\mathcal{H}\mathbf{\Lambda}_\mathcal{H}\mathbf{U}_\mathcal{H}^\dagger$ because $tr(\hat{\mathbf{Q}}) = tr(\mathbf{Q})$

It has has been shown that the optimal $\hat{\mathbf{Q}}$ is diagonal [1] and its corresponding capacity can be achieved by water-filling on the eigenvalues of $\mathbf{HH}^\dagger$. This is an optimization problem that must be solved numerically. A sub-optimal solution can be found by specifying transmission schemes that define various $\mathbf{Q}$ matrices. For instance, In beam-forming, $\hat{\mathbf{Q}}$ is a $\mathbf{0}_{N_t}$ matrix with E_s at index $(1,1)$ i.e. data is transmitted on the leading eigenvalue of $\mathbf{H}^\dagger\mathbf{H}$. In spatial multiplexing without channel knowledge, equal power allocation has been shown to be optimal; here $\mathbf{Q} = \frac{E_s}{N_t}\mathbf{I}_{N_t}$.

In the stochastic case, the notion of capacity is usually the average capacity with is defined as $\mathcal{C}_{ergodic} = \mathcal{E}_\mathbf{H}(C)$ i.e. the average capacity of the channel over the channel distribution. The ergodic capacity of double stochastic channels (the matrix Gaussian channel) is still an open problem. However, there are known results for the capacity in the zero mean single correlated channel case ($\mathbf{M} = \mathbf{0}, \mathbf{R_r} = \mathbf{I}_{N_r}, \mathbf{R_t}$ is arbitrary). It has been shown that this capacity can be achieved by water filling on the eigenvalues of the long term correlation matrix $\mathbf{R_t}$ [12]. There are also results for the channel capacity using beamforming and spatial multiplexing. current results for the capacity of stochastic wireless channels can be found in [5].

The use of capacity for link adaptation in a practical system adaptation is faced with many challenges. The most important consideration is that capacity is defined using infinite block lengths of coding. This impractical for a real systems. The capacity formulation does not explicitly deal with the notion of bit error as it assumes all transmissions are error free. In a practical system, error considerations are often relaxed in order to increase the combined data rate.

A more realistic optimization problem is to find the transmit configuration that leads to maximizing rate while minimizing

bit error rate. this problem can be solved analytically in specific cases. For instance, the problem was solved using double space time transmit diversity (D-STTD) with linear receivers in [3]. The analytic solution depends on the channel model chosen and is very sensitive to real world impairments. For instance, a real transmitter has to contend with non-linearity in the radio frequency (RF) front end, channel estimation errors and other impairments. Even with current analysis tools, rate and BER equations are not available for every MIMO scheme. Note that a brute force search is still required in this case to find the scheme that maximized the rate while minimizing BER. If there are only a few schemes, switching point can be found based on analysis of the above equations [1]. These switching points are often implemented using indicators of spatial selectivity (essentially a compressed feature space) such as channel condition number [3].

If there is a sufficiently detailed system model implemented, transmission decisions can be made by simulation. The rate and BER of a channel realization $\{\mathbf{H}, \lambda_o\}$, can be computed by streaming randomly generated bits through this channel with a simulated transmitter and receiver. This can be done for all possible transmission schemes. The scheme that maximizes the rate while maintaining a but error rate threshold is chosen. In a stochastic system, this will have to be repeated multiple times with multiple channel realizations in order to guarantee performance. The major advantage of this method is its generality. for instance, channel coding algorithms can be implemented on the transmitted data and form part of the switching criteria without a full analysis of the effect of coding on transmission[4].

Using any of the above methods, link adaptation using brute force search is very expensive. It is for this reason that I seek sparse representations of the channel space.

E. Classification

We discuss two ways to design a sparse classifier. First we can design a simple k-nn classifier using the appropriate distance measures. The k-nn classifier partitions the input space into Voronoi regions corresponding to each training object. For each new test object, the distances to each training object are computed. The new object is classified by majority decision between it's k nearest training points. This algorithm has been shown to perform well in various scenarios [10]. I can increase the sparsity of this classifier by making decisions based on the k clusters instead of all n training points. I intend to show that there is a negligible loss of performance in this case.

Another method for creating sparsity is the use of a support vector machine (SVM). SVM's encourage sparsity by using only objects that are close to the decision boundary to define the optimum separation of classes. In this way, they avoid computation using the entire training set to make decisions about a new test object (calculations only need to be done using the support vectors, see [10] for details). This problem can also be solved using an appropriately defined kernel matrix. The kernel matrix

[4]This is a difficult problem to solve analytically because coding adds another dimension of variation and different coding schemes behave differently [1].

$K(\cdot,\cdot)$ defines the distance between data points in a higher dimensional space. It allows classification in the higher dimensional space. However, computation is still done in the lower dimension - reducing computational complexity. The kernel matrix is constrained to be symmetric and positive definite. In our case, this can be achieved using $\mathbf{D}_{vecnorm}$ and $\mathbf{D}_{tr}$ but the Burg and KL divergence will have to be modified [13]. One common modification of the KL divergence that makes it symmetric and positive definite is shown in equation (12).

$$\mathbf{D}_{symKL}(\mathbf{X},\mathbf{Y}) = exp(\frac{1}{\rho}(KL(\mathbf{X}\|\mathbf{Y}) + KL(\mathbf{y}\|\mathbf{X}))) \tag{12}$$

Using the above measures, I intend to provide a solution to the SVM using the Trace and symmetric-KL kernels and apply the resulting support vector machines to the classification of wireless channels. the sparse nature of these classifiers should reduce overall classification complexity.

IV. Simulation Results

In my initial work, I chose to focus on a simple case of the wireless channel (for tractability). In this simple case, the channel has zero mean and transmit correlation only. This means $\mathbf{H} \sim \mathcal{CN}_{N_r,N_t}(\mathbf{0}_{N_r,N_t}, \mathbf{I}_{N_r,N_r} \otimes \mathbf{R_t})$. I implemented the clustering algorithm as described in section III. The algorithms were tested using synthetic channel correlations ($\mathbf{R_t}$). For each of k channel distributions, a channel correlation matrix was generated where the eigenvalues of $\mathbf{HH}^\dagger$ were chosen uniformly between 0 and 1 (and normalized). γ_o was chosen uniformly between 0 and γ_{max}.

I generated $n = 100$ realizations of k channel distributions by picking each distribution randomly and generating the relevant channel realization. For the stochastic case, I generated the k channel distributions by averaging over $30N_t$ channel realizations. In order to study the effect of cluster size, number of antennas and SNR, I generated channel values with $k \in \{5, 15, 25, 35\}, N_r = N_t \in \{2, 5, 8\}, \gamma_{max} \in \{-8, 0, 8\}dB$. The entire algorithm was repeated 60 times for each combination of channel values. The n channels were first clustered into k objects. These k objects were then classified into one of $m = 2$ classes using the capacity formulation [1], [3] i.e. the mode that maximized capacity was chosen (but the actual transmission rate/constellation adaptation was not implemented). Next $u = 20$ new channel objects were created randomly from the same k initial distributions. I then used a 1-nn (single closest neighbor) to find the optimum transmission schemes for the new channel objects.

In the instantaneous channel case, the clustering was done using the Burg divergence and the vecnorm distance. Both of these measures and the trace distance were used for classification. In the stochastic channel case, the clustering was done using the KL divergence. The performance was calculated by computing the percentage of time the k cluster representatives fell into different classes from the n channel objects. The performance of the final classification was also computed similarly using the

percentage of time classification led to different classes from brute force search.

The number of steps required for clustering is plotted for all scenarios in Figs. 1, 5. I saw that for larger number of antennas, clustering converged pretty quickly. However, for two antennas, the clustering converged very slowly and even failed to converge for $k \in \{25, 35\}$. This suggests that the number of clusters should scale with the number of antennas. It is likely that the results will be similar even in the realistic correlation cases to be simulated for the final report [14].

Using the burg measure Fig. 2, the percentage of misclassified training points was about 40% on average. This meant that many of the training points were classified incorrectly especially at high SNR's. I attributed this to the Burg measure attempting to cluster using both the eigenvalues and eigenvectors of the channel realization, even though the capacity (BFS method) depends only on the eigenvalues. I intend to solve this problem by using another distance measure that only considers the channel eigenvalues for clustering. The largest training error was found for $Nt = 8, \gamma_{max} = 8dB$. The plateau of error for $N_t = 2$ was due to the clustering not converging for those values of k. I found similar results using the vecnorm error Fig. 6, though the rise with k was more pronounced. The test error did not fare much better with the best performance at low SNR's and the worst at high SNR's. once again, the $Nt = 8, \gamma_{max} = 8dB$ case performed badly.

The stochastic Channel was clustered using the KL divergence measure (Figs. 9-12). As in the instantaneous case, the number of iterations required for clustering convergence increased with number of clusters; Fig. 9. The algorithm was also unable to converge in less than 50 iterations for high values of k. The training error was low and fairly constant for most situations. In fact training errors of 0% were observed in may cases. I noticed a marked increase in error for $N_t = 2, \gamma_{max} = 8dB$. I need to investigate this occurrence further in order to explain the comparatively bad performance in this case The stochastic channel was clustered using the KL measure and the trace measure. Both algorithms performed well except in the $N_t = 2, \gamma_{max} = 8dB$ case. On average, the trace distance was able to outperform the KL distortion, but the performance increase was within the margin of error. This further motivates using the trace distance or another less computationally expensive distance measure to cluster and classify wireless channels.

V. Proposed and Future Work

There is still extensive work left to be done for this project. In this section we list the work that I aim to complete before the end of the semester

- I intend to show analytically that the clustering does not significantly affect capacity if the correct distance measure is used. I intend to show this by providing loose bounds on lost capacity based on clustering. This is similar to work done in [15] for the effect of imperfect channel estimates.
- I intend to implement the brute force search using BER/rate and using system simulations as discussed in section III-D.

- The channel correlation simulated was a very simple case. I intend to simulate a more realistic channel model in order to better quantify performance. In a related problem, I intend to show the performance of the algorithm in the general matrix Gaussian channel.
- I expect that there are a few correlation scenarios where it will be advantageous cluster and classify channels based on the full channel matrix instead of computing the channel condition number [3]. This problem is likely more pronounced in high dimensions. I intend to simulate other adaptation algorithms and show the scenarios where the proposed algorithm has superior performance.
- The KL divergence works well for stochastic adaptation. However, it is quite expensive to compute. The burg distance did not work well for the instantaneous channel classification case. I intend to explore the use of different distance measures to find one with that utilizes the matrix eigenvalues, but is relatively simple to compute.
- I intend to solve the SVM problem using the symmetric KL divergence and the trace metric. I will compare the performance of this algorithm to the performance of the k-nn classifier.

There are also extensions that I am interested in but will be unable to pursue this semester. they include:

- The k-means algorithm has a rich library of optimizations for speed [9]–[11]. I would like to explore the use of some of these heuristics to speed up the general k-means algorithm with non-euclidean distance.
- I would like to extend this work to MIMO-OFDM. the major issue there is getting good estimates of the channel correlations given very few data samples. However the channel is relatively well understood and I can use Bayesian priors to improve the estimates of channel statistics.
- The algorithm as described seems to have a strong connection to vector/channel quantization and code-book design. I would like to explore these extensions further and explore opportunities for limited feedback algorithms based on the clustering and classification.

VI. Conclusions

In this report, I have motivated and outlined an algorithm for link adaptation based on learning methods. I have also provided some preliminary results. the preliminary results seem promising especially in the stochastic channel case. This also motivate further work in order to understand the effect of the distance measure and other choices on algorithm performance.

References

[1] R. W. Heath, Jr. and Others, *Untitled MIMO Book (in Development)*, 2008.

[2] D. Gesbert, M. Shafi, D. shan Shiu, P. J. Smith, and A. Naguib, "From theory to practice: an overview of MIMO space-time coded wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 281–302, Apr. 2003.

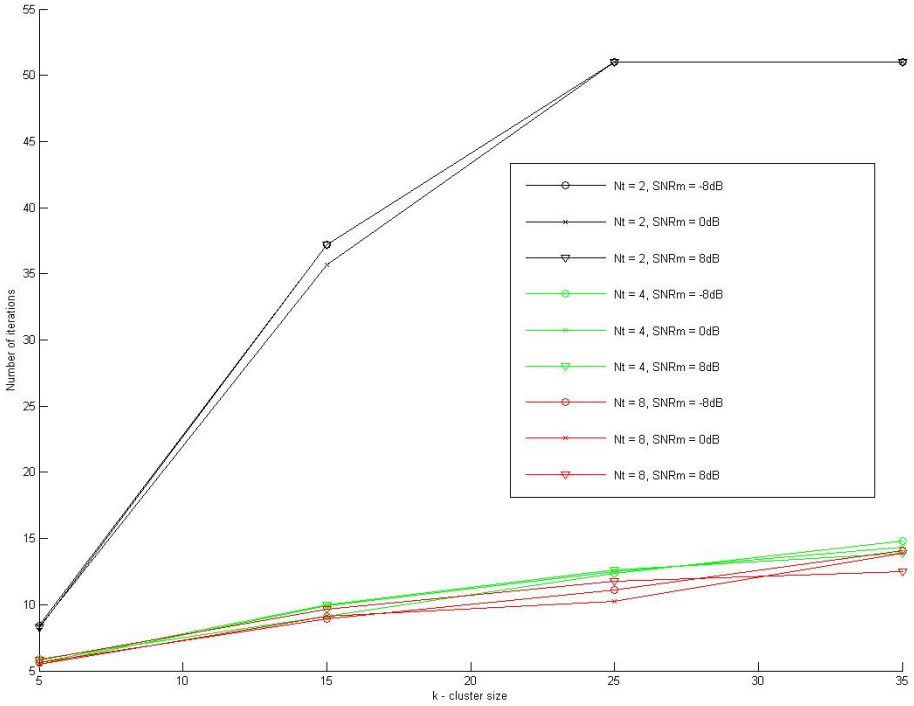

Fig. 1. Instantaneous channel with Burg distortion - number of steps for convergence

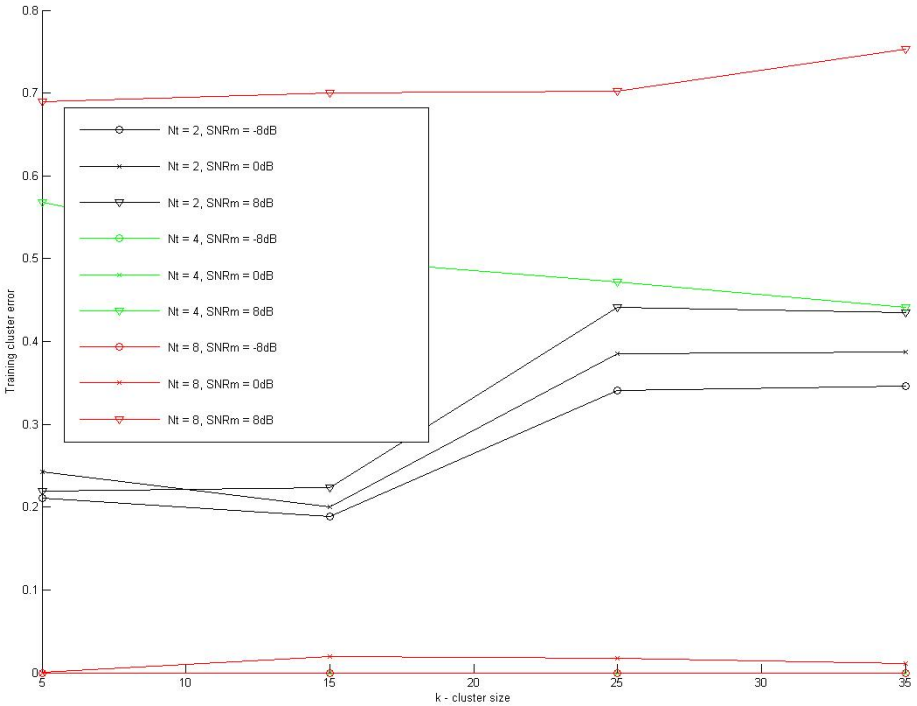

Fig. 2. Instantaneous channel with Burg distortion - training error

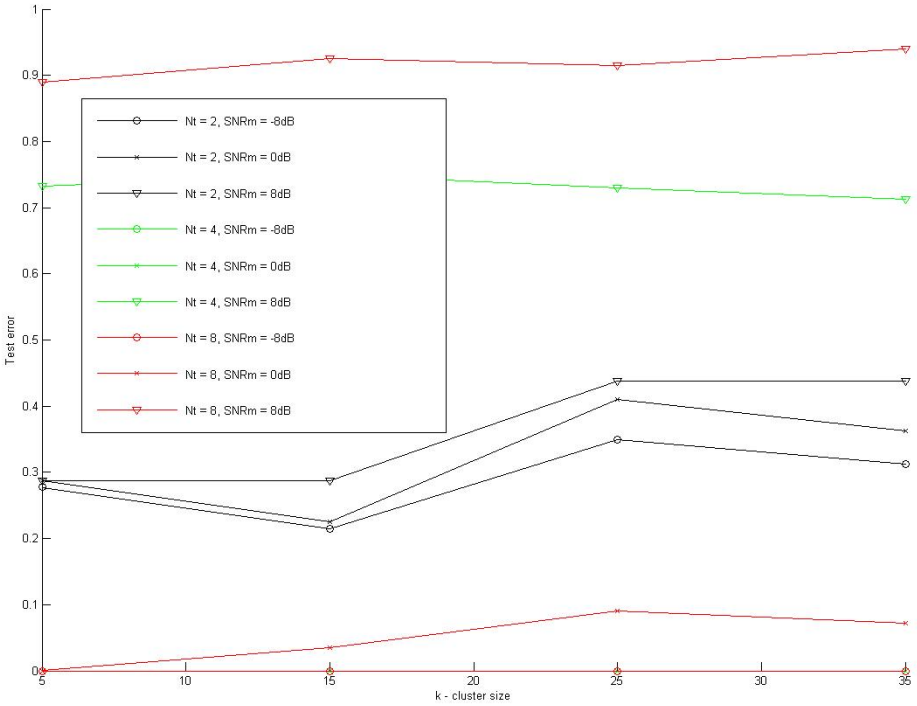

Fig. 3. Instantaneous channel with Burg distortion - Test error with burg distance

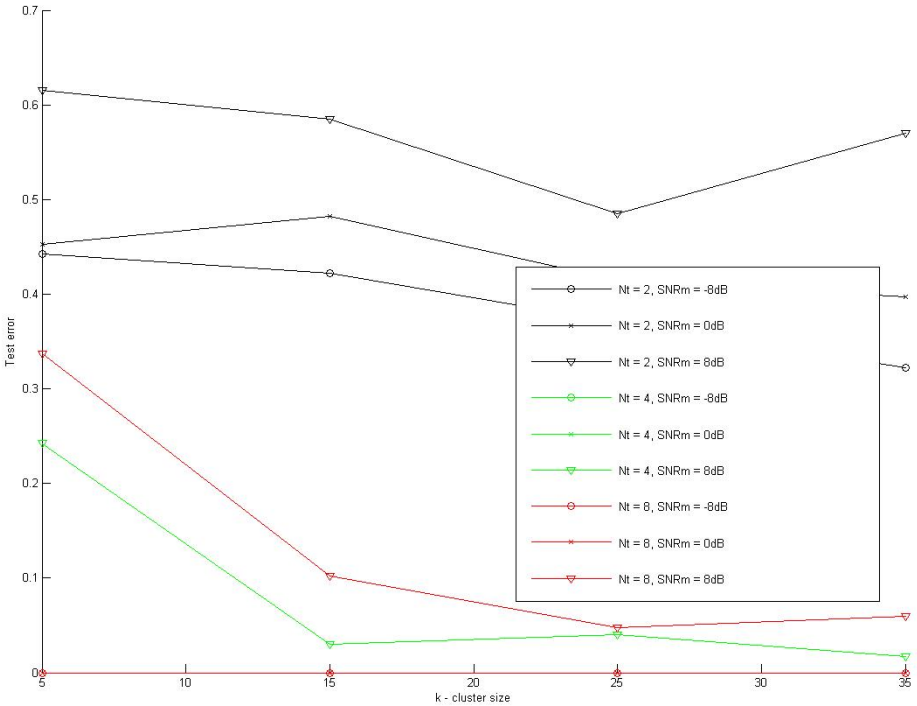

Fig. 4. Instantaneous channel with Burg distortion - test error with trace distance

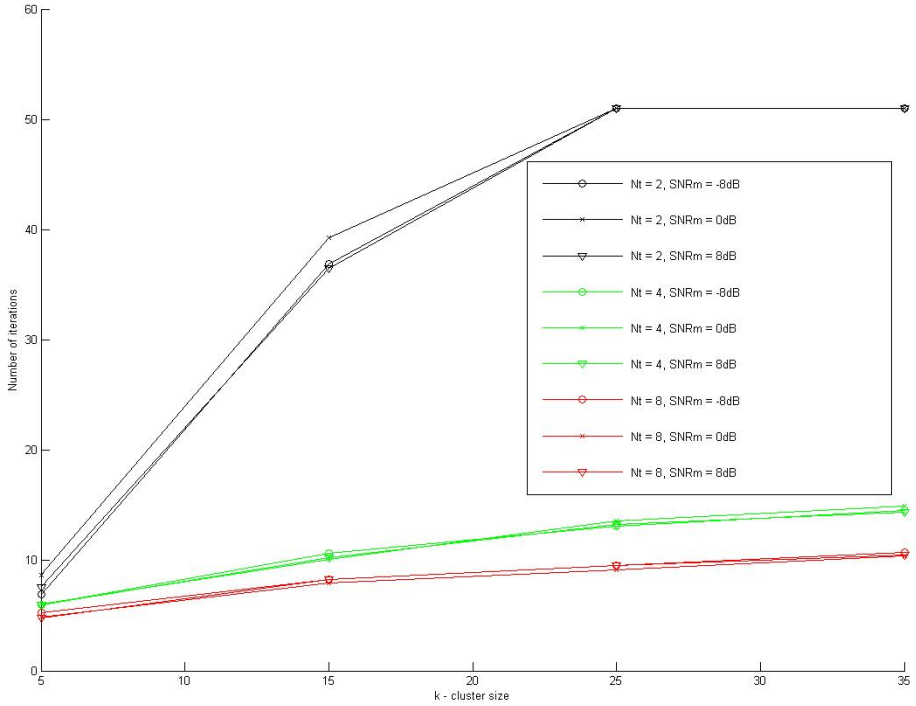

Fig. 5. Instantaneous channel with normvec distortion - number of steps for convergence

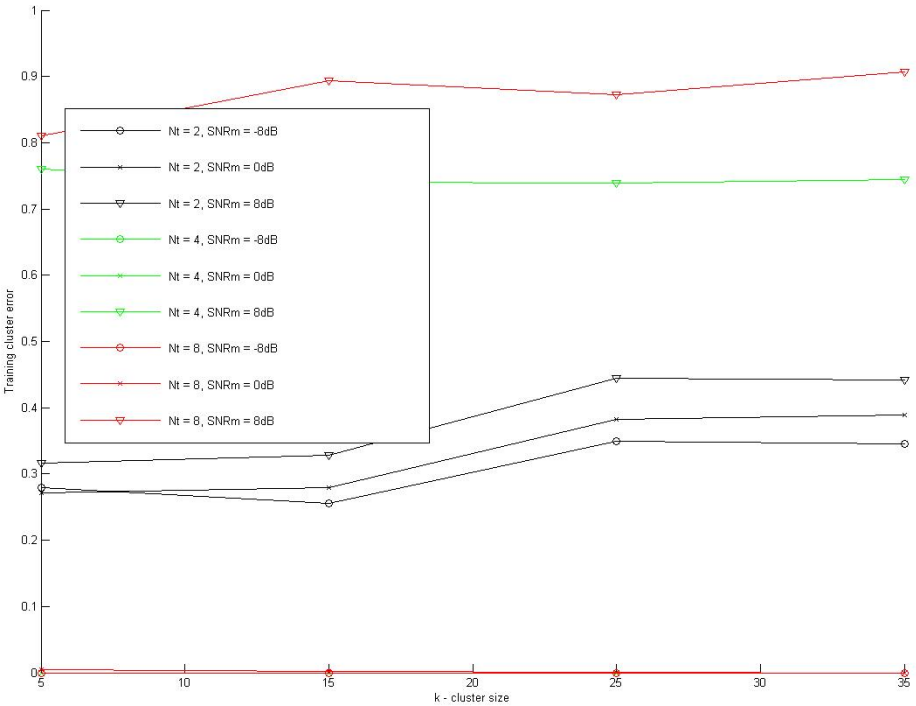

Fig. 6. Instantaneous channel with normvec distortion - training error

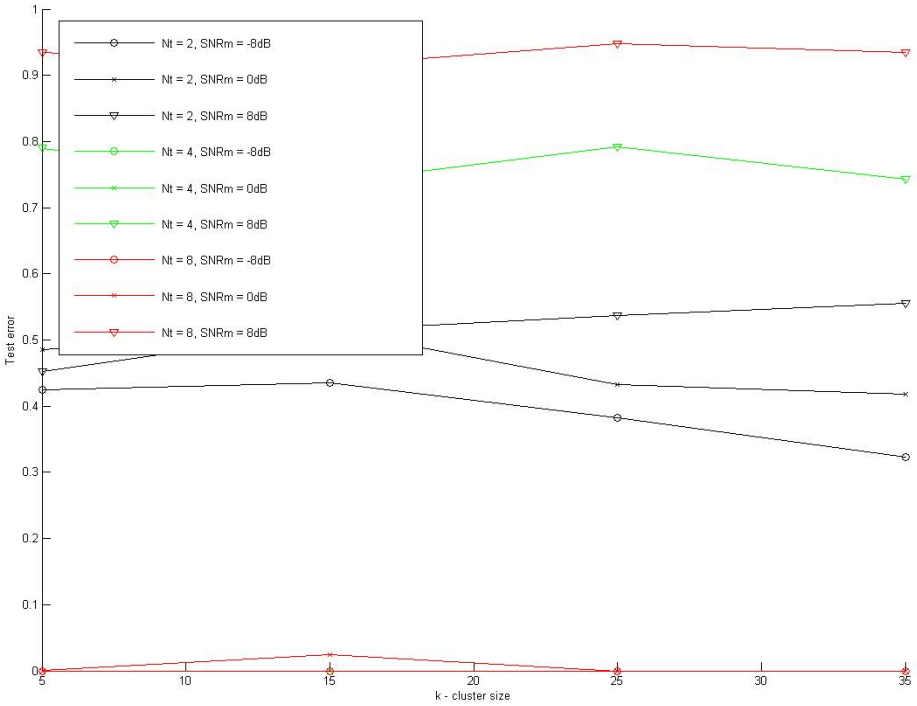

Fig. 7. Instantaneous channel with Burg distortion - Test error with burg distance

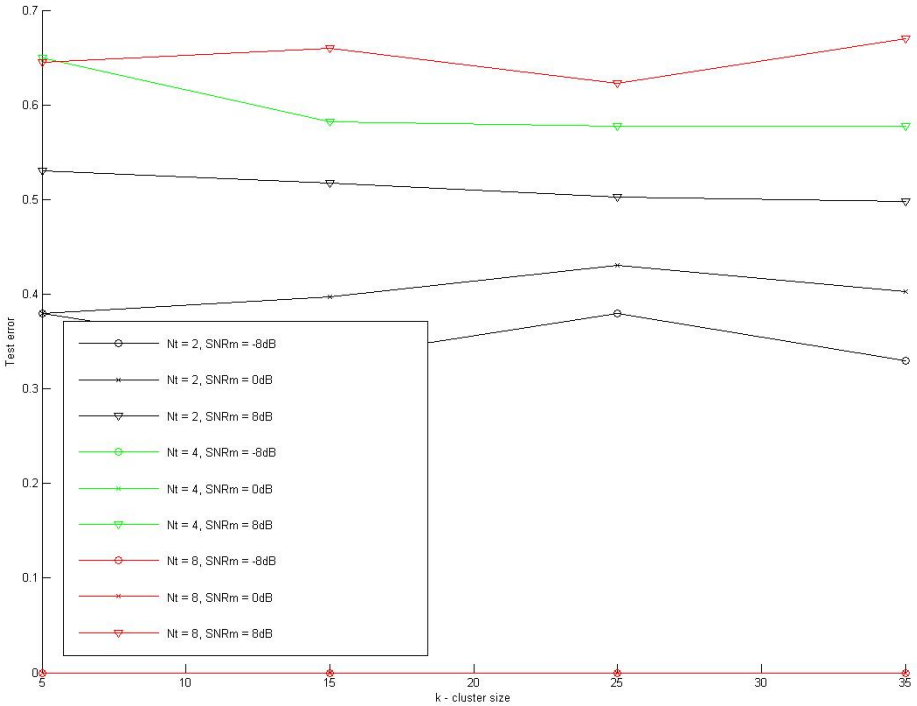

Fig. 8. Instantaneous channel with Burg distortion - test error with trace distance

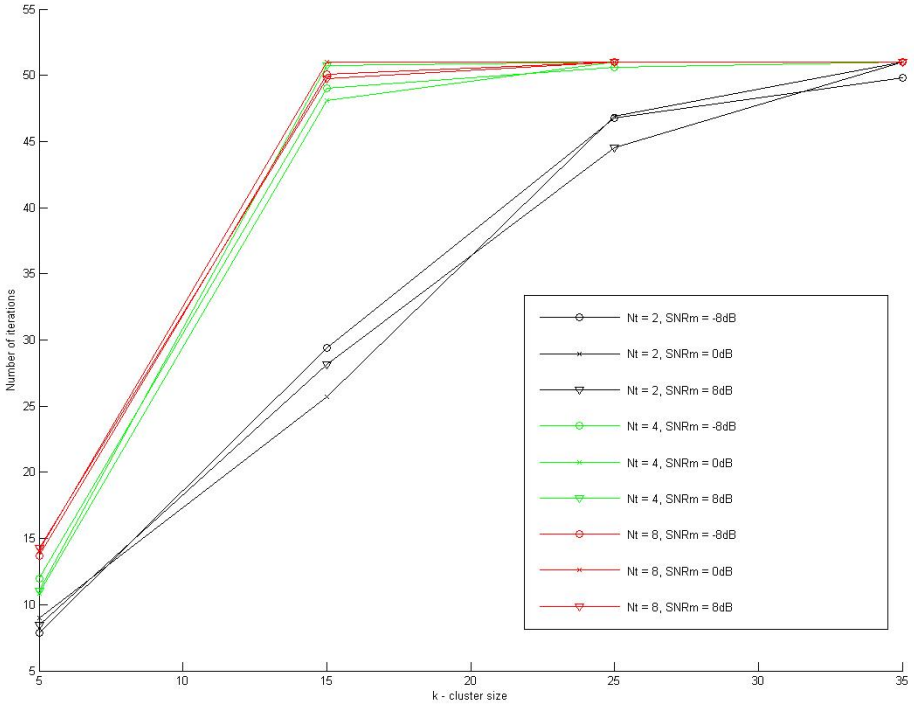

Fig. 9. Stochastic channel with normvec distortion - number of steps for convergence

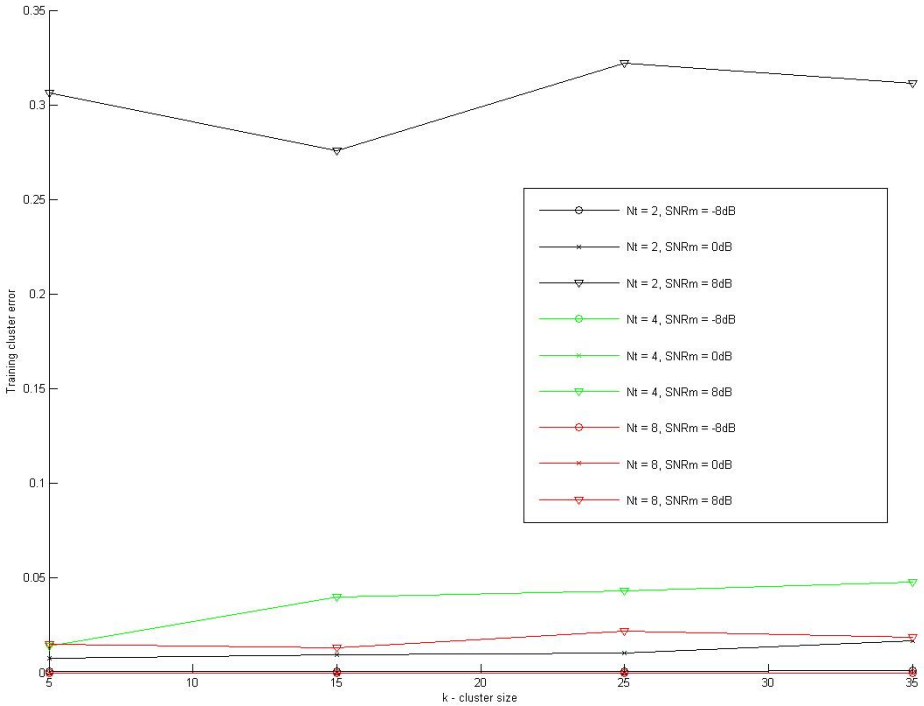

Fig. 10. Stochastic channel with KL distortion - training error

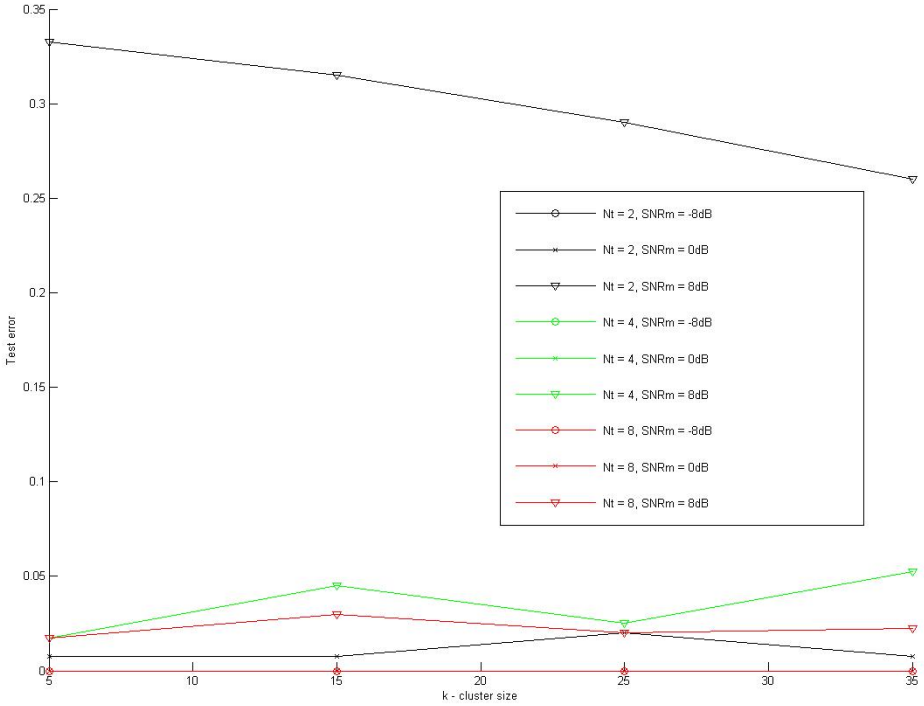

Fig. 11. Stochastic channel with KL distortion - test error

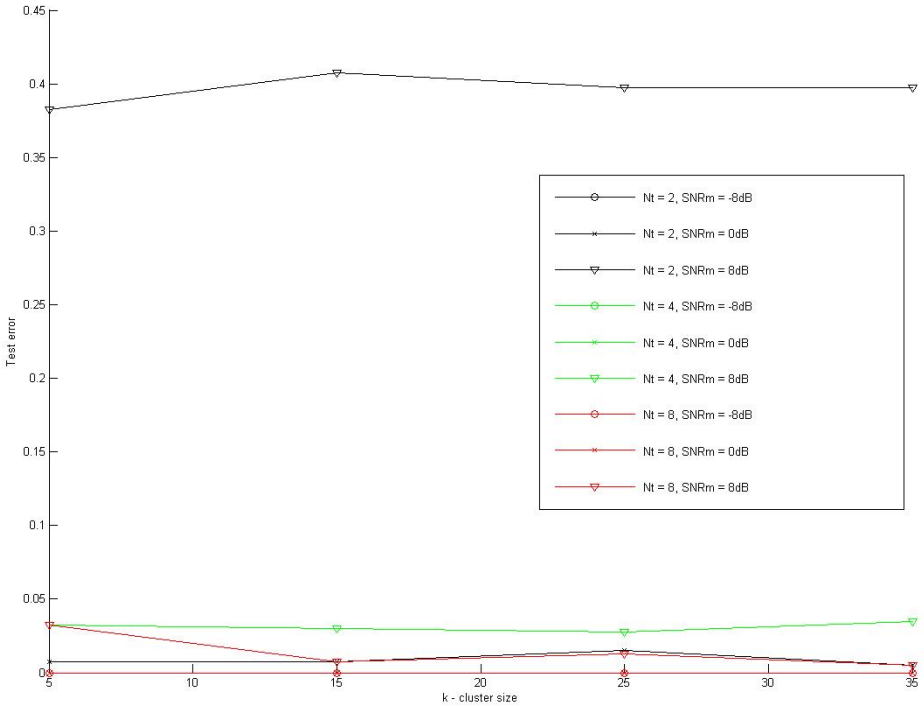

Fig. 12. Stochastic channel with Trace distortion - test error

[3] A. Forenza, M. R. McKay, A. Pandharipande, R. W. Heath, and I. B. Collings, "Adaptive MIMO transmission for exploiting the capacity of spatially correlated channels," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 619–630, Mar. 2007.

[4] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless LAN via linkadaptation," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 7, Helsinki, Finland, 2001, pp. 1995–2000.

[5] M. R. McKay and I. B. Collings, "General capacity bounds for spatially correlated rician MIMO channels," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3121–3145, Sep. 2005.

[6] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with bregman divergences," 2004. [Online]. Available: citeseer.ist.psu.edu/article/banerjee04clustering.html

[7] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *ICML '06: Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA: ACM, 2006, pp. 505–512.

[8] J. V. Davis and I. Dhillon, "Differential entropic clustering of multivariate gaussians," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 337–344.

[9] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[11] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[12] E. A. Jorswieck and H. Boche, "Channel capacity and capacity-range of beamforming in MIMO wireless systems under correlated fading with covariance feedback," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1543–1553, Sep. 2004.

[13] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A kullback-leibler divergence based kernel for svm classification in multimedia applications," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[14] A. Forenza, D. J. Love, and R. W. Heath, "Simplified spatial correlation models for clustered MIMO channels with different array configurations," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 1924–1934, Jul. 2007.

[15] P. Kyritsi, R. A. Valenzuela, and D. C. Cox, "Channel and capacity estimation errors," *IEEE Communications Letters*, vol. 6, no. 12, pp. 517–519, Dec. 2002.