11-28-2016

Unsupervised Shift-invariant Feature Learning from Time-series Data

Masoumeh Heidari Kapourchali

Unsupervised Shift-invariant Feature Learning from Time-series Data

by

Masoumeh Heidari Kapourchali

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Electrical and Computer Engineering

The University of Memphis

December 2016

To my parents.

Acknowledgments

I would like to take this opportunity to express my profound appreciation to my advisor, Dr. Bonny Banerjee for his constant guidance, support and insightful comments. His commitment, diligence and enthusiasm of solving fundamental problems not only were deeply influential in this research, but the experience as his student has led to my vision of how to benefit the world through academic research.

I would also like to extent my thanks to my thesis committee members, Dr. Dale Bowman, Dr. Eddie Jacobs and Dr. Madhusudhanan Balasubramanian for all their help and guidance.

ABSTRACT

Unsupervised feature learning is one of the key components of machine learning and artificial intelligence. Learning features from high dimensional streaming data is an important and difficult problem which is incorporated with number of challenges. Moreover, feature learning algorithms need to be evaluated and generalized for time series with different patterns and components. A detailed study is needed to clarify when simple algorithms fail to learn features and whether we need more complicated methods.

In this thesis, we show that the systematic way to learn meaningful features from time-series is by using convolutional or shift-invariant versions of unsupervised feature learning. We experimentally compare the shift-invariant versions of clustering, sparse coding and non-negative matrix factorization algorithms for: reconstruction, noise separation, prediction, classification and simulating auditory filters from acoustic signals. The results show that the most efficient and highly scalable clustering algorithm with a simple modification in inference and learning phase is able to produce meaningful results. Clustering features are also comparable with sparse coding and non-negative matrix factorization in most of the tasks (e.g. classification) and even more successful in some (e.g. prediction). Shift invariant sparse coding is also used on a novel application, inferring hearing loss from speech signal and produced promising results.

Performance of algorithms with regard to some important factors such as: time series components, number of features and size of receptive field is also analyzed. The results show that there is a significant positive correlation between performance of clustering with degree of trend, frequency skewness, frequency kurtosis and serial correlation of data, whereas, the correlation is negative in the case of dataset average bandwidth. Performance of shift invariant sparse coding is affected by frequency skewness, frequency kurtosis and serial correlation of data. Non-Negative matrix factorization is influenced

by data characteristics same as clustering.

Table of Contents

List of Figures

Figure Page

Figure Page

List of Tables

Table Page

1. Introduction

Feature learning is a very important problem in today's world inundated with data. Three classes of algorithms have been found to be very effective for unsupervised feature learning: sparse coding that minimizes the reconstruction error subject to sparsity constraints, non-negative matrix factorization (NMF) which is also called non-negative sparse coding in some cases as well as clustering that captures the data distribution. Coates et al. [CN11, CLN10] analyzed the performance of several feature learning algorithms, such as, sparse autoencoders, sparse RBMs, k-means clustering, and Gaussian mixture model, for the task of classification in images. The simplest and computationally efficient k-means clustering emerged as the best performer on the CIFAR-10 and NORB datasets.

In the case of time series, however, Keogh and Lin [KL05] made objections to existing methods of clustering and found it meaningless in some cases.The standard way to deal with time-series is to sample it using a shifting window; the data distribution within a window is assumed to be stationary. It was found, if the overlap between consecutive windows is high, the features learned using clustering is independent of the data and hence, were deemed meaningless. On the other hand, overlap between two consecutive windows is necessary because without overlap, choice of offset for the first window would become a critical parameter and choices that differ by just one point can give arbitrarily different results.

Since 2005, the challenges of time-series clustering have been well-studied. The problem with the previous works is proposing heuristic solutions without analyzing the underlying issues, complexity, missing parts of information, lack of real world high dimensional data validation and failing to find desired patterns. To the best of our knowledge none of the previous works focused on comparing clustering feature learning approach with other feature learning algorithms. Furthermore, even though sparse coding has been widely used in different applications and specially as a subroutine in deep learning, there is a lack of evaluation method to study quality of learned features [HCSH15].

Our goals are two-fold: 1) to analyze when clustering of time series is meaningless, how this problem can be solved using shift-invariant algorithms and how meaningful features can be learned from time-series using shift-invariant (Convolutional) clustering algorithms, and 2) to evaluate the performance of these three clustering, sparse coding and non-negative matrix factorization for unsupervised feature learning from different benchmark time-series datasets for applications of reconstruction, noise separation, prediction, classification and simulating auditory filters . Furthermore, we report our findings on inferring hearing loss from an individual's speech using a novel line of investigation.

Structural analysis of data along with evaluating algorithms with regard to number of features and size of receptive field provides a prescription of choosing the best algorithm and parameters for a given dataset.

1.1 Contributions

- Proposing a systematic way of learning meaningful features from time series data using clustering algorithm.
- Evaluating results of three unsupervised feature learning algorithms on five different tasks.
- Analyzing the factors that can significantly affect performance of algorithms on representing data.
- Providing a prescription of choosing the best feature learning algorithm and parameters for a given dataset.

- Applying shift invariant sparse coding to infer hearing loss of hearing impaired individuals from their speech.

1.2 Outline

This thesis will proceed as follows:

Chapter 2: Literature Review. Chapter 2 will cover the previous works on unsupervised feature learning, its applications and evaluation methods. Since, unsupervised feature learning concerns with a wide area of research, we will focus on the three algorithms that have been used in this thesis.

Chapter 3: Models and Methods. In this section the three learning algorithms, the applications that the learned features are applied on, including reconstruction, noise separation, prediction, classification and simulating encoding signals as well as the evaluation metrics are introduced. The approach that has been used for inferring hearing loss of hearing impaired individuals' from their speech is also introduced in this section.

Chapter 4: Experimental Results. The experimental results and evaluation of results are presented in chapter 4. In this section, we will also analyze the results, take the factors that may influence performance of algorithms into account, and provide a prescription on selecting the feature learning algorithms on different data and situations.

2. Literature Review

In this section literature review of the relevant works on the three unsupervised feature learning algorithms, the five applications and evaluation methods is presented.

2.1 Subsequence time series clustering

Clustering is one of the important tasks of data mining that can also be seen as an unsupervised feature learning algorithm [CLN10]. K-means clustering had been applied by Das et al. [DLM+98] on timeseries subsequences in 1998. A time series can be defined as an ordered sequence of real valued numbers which are uniformly sampled measurements of an event or quantity. Many signals of interest such as speech, stock price, and physiological signals are time series. Many other works have used subsequence clustering as a subroutine [TSD00, FCNL01, HDT02, JLS02, MU01] in different tasks such as prediction, abnormality detection and prediction. In 2005, Keogh and Lin using some experiments showed subsequence of time series clustering (STS clustering) is meaningless [KL05]. Since 2005 researchers have tried to figure out the challenges of time series clustering and find a solution [Che05, Che07, DBD09, MKBS09, RKLE12, CHKB13, MSRR13, RNR12, SYCC+15]. Authors in [ZAT14] have reviewed subsequence time series clustering, found three main research proposing solutions. The problem with the previous works is proposing heuristic solutions without analyzing the underlying issues, complexity, missing parts of information, lack of real world high dimensional data validation, and failing to find desired patterns.

2.2 Shift-invariant sparse coding

Sparse coding is a signal representation method which adopts a dictionary of features [AEB06, MBPS09]. Concentration on shift invariant versions of sparse coding has increased in the applications of audio and image signals during last decade [SL06, MLG+08, CPR13, BL14, Woh14, Woh16]. Sparse coding has many applications in the pipeline of unsupervised feature learning such as audio classification[LPLN09], cognitive science [AL01], deionising and reconstruction of audio signals [HCSH15] and prediction [FRG14]. Motivations of using shift-invariant version of sparse coding is different based on the area of study. In the case of audio which is a type of time series, authors in [SL06] refer to disadvantages of blocking and role of choosing the offset of first window.

2.3 Non-negative matrix factorization

Non-negative matrix factorization [PT94] is also a method of finding a suitable representation of data. There are different types of non-negative matrix factorization which use different cost functions. In the case of using mean squared cost function and L1 regularization, this algorithm is very similar to sparse coding [HD11], so it can be called non-negative sparse coding [Hoy02, TN12]. There are some works to make NMF algorithm shift-invariant [Beh03, PPC08].

3. Models and Methods

This section introduces the algorithms and methods that have been used to make clustering of time series meaningful and evaluate the algorithms. A brief description of each application is also included.

3.1 Algorithms

This section discusses unsupervised shift-invariant (or convolutional) feature learning algorithms for time series data. Choice of sliding window with maximum overlap between two consecutive windows is to include all possible phases of each pattern in the learning stage. So shift-invariant learning approaches should be able to find the patterns even if they are distributed in different phases. Stochastic gradient descent which is an incremental method is used for optimizing the objective function.

3.1.1 Shift-invariant spherical clustering

In case of high-dimensional data, such as time-series, the direction of a data vector is more important than its magnitude [SGM00] which is captured by cosine similarity in spherical clustering. Shift-invariant spherical clustering learns nonorthogonal and shift-invariant features that partitions the input space on the surface of a d-dimensional hypersphere of unit radius. The algorithm captures the density of the data in an unsupervised manner by maximizing the following objective on convergence:

$$\ell(\mathcal{X}, \mathcal{D}) = \sum_{i=1}^{k} \sum_{x_j \in \mathcal{N}(i)} (x_j * D_i) \tag{3.1}$$

where $\mathcal{X} = \{x_1, x_2, ...x_t\}$ is the set of n-dimensional data points, $\mathcal{D} = \{D_1, D_2, ...D_k\}$ is the set of d-dimensional features (or cluster centers), $d \geq n$, $\mathcal{N}(i)$ is the set of data points in the neighborhood of D_i, and $*$ is the convolution operator. The performance of shift-invariant clustering algorithm is evaluated using frequency analysis and statistical metrics, such as: meaningfulness, entropy and F-measure.

3.1.2 Shift-invariant sparse coding

Sparse coding may be construed as a generalization of the winner-take-all spherical clustering [AEB06]. The algorithm consists of two steps: encoding which is often computed using a matching pursuit-like [MZ93] algorithm, and learning by minimizing the following objective function:

$$\varepsilon(\mathcal{X}, \mathcal{D}) = \frac{1}{2}||\mathcal{X} - \sum_{i=1}^{k} \mathcal{D}_i * \alpha_i||_2^2 + \lambda||\alpha||_1 \tag{3.2}$$

where α is the coefficient vector and λ is a parameter governing the tradeoff between accurate reconstruction of the data points and the regularization.

3.1.3 Shift-invariant non-negative matrix factorization

Non-negative matrix factorization (NMF) is an unsupervised feature learning algorithm and different objective functions are proposed. In the case of using mean squared cost function and L1 regularization, it is very similar to shift invariant sparse coding [Beh03, PPC08]. The only difference is that both data and dictionary have to be non -negative. The objective function is as follows:

$$\varepsilon(\mathcal{X}, \mathcal{D}) = \frac{1}{2}||\mathcal{X} - \sum_{i=1}^{k} \mathcal{D}_i * \alpha_i||_2^2 + \lambda \sum f(\alpha) \tag{3.3}$$

The difference between sparse coding and NMF is the sparsity constraints. The form of f defines a measurement for trade off between reconstruction accuracy and sparsity level. The typical choice of f in NMF is $f(\alpha) = |\alpha|$. The convolution is used instead of dot product to learn shift invariant features.

3.2 Applications

In this section the applications which have been used for evaluation of algorithms are introduces.

3.2.1 Reconstruction

Reconstruction of a signal can be done using a matching pursuit algorithm and reconstruction signal-to-noise ratio (SNR) is a widely used measure for evaluating the ability of a dictionary of features with respect to rate fidelity or ability of reconstruction [MLG+08, SL06]. As the number of coefficients increases, SNR also increases which shows improvement of reconstruction. However, a larger number of coefficients need more computational cost because of a larger number of iterations in matching pursuit algorithm. SNR can be calculated in decibels (dB) using the following equation:

$$SNR_{dB} = 20 \log_{10} \frac{A_{signal}}{A_{residual}} \tag{3.4}$$

A_{signal} is amplitude of signal (a timeseries window) and $A_{residual}$ is amplitude of reconstruction error.

3.2.2 Noise separation

Four separation oriented measures are introduced in [HCSH15] to evaluate ability of dictionaries in noise separation. ϵ-error noise to speech separability (ϵ_{NSS}) is one of them. Given a speech dictionary $\mathcal{D}_s = \{D_1, D_2, ..., D_I\}$ and a noise dictionary $\mathcal{D}_v = \{D_1, D_2, ..., D_J\}$, a speech evaluation dataset $\mathcal{X}_s$, and a noise evaluation dataset $\mathcal{X}_v$,

$$\epsilon_{NSS} = \epsilon_{ASD(\mathcal{D}_s,\mathcal{X}_v)} - \epsilon_{ASD(\mathcal{D}_s,\mathcal{X}_s)} \tag{3.5}$$

where ϵ_{ASD} is average sparseness degree of representation when the error tolerance is fixed as ϵ. A matching pursuit algorithm was used for sparse decomposition in encoding (testing) step. In this algorithm each feature can be used multiple times which is an applicable property for shift-invariant algorithms because they need less number of features to represent the data, the dictionaries do not have to be over complete even in sparse coding.

3.2.3 Prediction

Three methods of online prediction with capturing temporal correlations have been introduced in [FRG14]. An exponentially decaying window technique is used for prediction task using the three shift invariant algorithms. Considering observations $x_\tau \in \Re^n$ and dictionary of features $\mathcal{D}$ which is learned on data $\{x_\tau\}_{\tau=1}^{t-1}$, the vector of coefficients α_t corresponding to the tth measurement x_t is found as:

$$\alpha_t = \arg\min_{\alpha} \mathcal{L}_\gamma^t(\alpha, \mathcal{D}) \tag{3.6}$$

where

$$\mathcal{L}_\gamma^t(\alpha, \mathcal{D}) = \frac{1}{2}\sum_{\tau=1}^{t} \gamma_{t,\tau}||x_\tau - \mathcal{D}\alpha||_2^2 + \lambda_1||\alpha||_1 \tag{3.7}$$

where the forgetting factor $\gamma_{t,\tau} = \gamma^{t-\tau}, \gamma \in (0,1]$ allows to capture temporal correlation in α while down-weighs influence of old measurements.

The performance of algorithms are measured using two metrics, Mean Absolute Percentage Error (MAPE) [HK06] and relative Root Mean Square Error (RMSE).

$$MAPE = \frac{100}{n}\sum_{i=1}^{n} |\frac{x_t - \hat{x_t}}{x_t}| \tag{3.8}$$

where $\hat{x_t}$ is the forecasted value for observation at time t.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_t - \hat{x_t})^2} \tag{3.9}$$

3.2.4 Classification

A simple classification algorithm (k-NN) is used to classify the time series using learned features [Alt92]. After unsupervised feature learning step, a mapping between a new signal and a feature vector is needed to train the classifier. These features can be selected arbitrarily [Coa12]. This step is called feature encoding or inference. The features feed k-NN as an input. The classification error rate is used to evaluate quality of learned features in classification. K-NN works as follows: when a sample data arrives, k-NN finds the k nearest neighbors in the labeled training set based on some distance measures (e.g. Euclidean distance) and the classification is done using a voting approach.

3.2.5 Simulating auditory filters

Authors in [SL06] have used the convolutional sparse coding to simulate auditory filters. They have shown the acoustic waveforms can be represented efficiently using a non-linear model based on a population spike code. The spikes are able to encode temporal positions and magnitudes of acoustic features precisely. They have also shown that there is striking similarities between learned features from speech and auditory nerve-filters.

Using box plots they have shown similarities of learned features with cat auditory nerve filters and gammatones filter bank which is a mathematical approximation of cochlear filters.

3.2.6 Inferring hearing loss from speech

Shift-invariant sparse coding is used to learn kernels (features) from speech. Then, unimportant and harmful features were detected and removed using the method introduced in [HCSH15]. The kernel functions are analyzed using the a set of neuro-physiological metrics and statistical metrics to infer hearing loss of hearing impaired individual.

Tuning Curve (TC)

A frequency TC is used to display the auditory threshold at various frequencies for a single auditory neuron. Each nerve fiber has a *characteristic frequency* (CF) where it responds at threshold. TCs should be symmetric at frequencies below 1000 Hz, whereas, at higher frequencies the curves become increasingly asymmetric with a very sensitive, frequency-selective tip and long, broadly-tuned tail. Hair cell damage is a leading cause of hearing loss. Flattened tip of the TC happens due to damage of outer hair cells results and loss of sensitivity. However, loss of inner hair cells allows the TC to maintain its overall shape but there is a loss of sensitivity. Loss of both inner and outer hair cells result in a major loss of sensitivity which causes a much broader shape to the TC. The distribution of CFs will highlight the frequency regions within the audible range where hair cells are damaged or missing; such regions should be larger or more frequent in individuals with severe-to-profound hearing loss than normal ones. The shape of the TC is captured by its bandwidth and Q_{10} value.

Q_{10} value

The sharpness of a TC is determined by the width of the V-shape of the curve relative to the CF which is commonly expressed in terms of the quality (Q) factor. The Q_{10} is typically used; it refers to the point that is 10 dB below the peak. Formally, $Q_{10} = f_C/BW$ where f_C is the CF and BW is the bandwidth. The half-power points are the usual cutoff values which are used to define a bandwidth. Since it is difficult to determine the half-power points of TCs, the points on the curve that are 10 dB up from the minimum point of the TC are used. The bandwidth of a TC provides important information about its frequency selectivity; as bandwidth increases, frequency selectivity decreases. Thus, hearing-impaired individuals ought to have greater bandwidth than their normal counterparts which can be captured by the mean bandwidth of all TCs across the spectrum. For a particular CF, narrower the bandwidth, larger is the Q_{10} dB value. Due to greater bandwidth, the slope of Q_{10} values increases slower with frequency for hearing-impaired individuals as compared to normal-hearing ones.

Perception Measurements

Three measurements were considered to show the level of hearing loss in our subjects.

Hearing measurement.

Each subject's hearing was quantified by calculating the pure tone average (PTA) which provides the average of the hearing threshold levels at 500, 1000, and 2000 Hz. This frequency region is commonly referred to as the speech frequency region of the audiogram. The PTA is a decibel level that quantifies the degree of hearing loss for each ear.

Perception measurement.

All subjects' speech perception ability was evaluated using the AzBio sentences [SDL+12]. The AzBio sentences are recorded by both male and female talkers and are routinely used to evaluate the speech perception capabilities of hearing-impaired subjects. All subjects listened to three 20-sentence AzBio lists, one in quiet and two in noise, and listeners were required to repeat the sentences heard. Listener responses were scored as percent correct based on the number of words repeated correctly across all sentences in a list.

Hearing loss age of onset.

Hearing loss age of onset is the age which hearing loss was started. This feature is considered because it affects ability of speaking, and there should be a significant difference between a person who have never had a hearing ability and a person who lost his hearing ability when he was older.

3.2.7 Structural analysis of time series

In this section, performance of algorithms with regard to structural analysis of data is analyzed. There are some classical and advanced statistical features which describe global characteristics of time series [WSH06]. Trend, seasonality, skewness, kurtosis, frequency skewness, frequency kurtosis, serial correlation, and average bandwidth are eight of the quantified descriptors that are used to be checked in this thesis. A normalized metric to [-1,1] shows degree of presence of a feature.

Trend and seasonality

Trend and seasonality are common features of time series. Traditionally, every time series can be decomposed to trend, cyclic, seasonal, and irregular components. Seasonal-Trend decomposition procedure based on Loess (STL) [CCMT90] which is a filtering procedure for decomposing a time series into trend, seasonal, and reminder components is used to decompose time series. Let Y be the original time series, X be de-trended time series which is computed using $X = Y - T$, Z be de-seasonalized signal $Z = Y - S$, and reminder series is defined as $Y' = Y - T - S$. Then, measures of trend and seasonality are as follows [WSH06]: degree of trend= $1 - \frac{Var(Y')}{Var(Z)}$ and degree of seasonality $= 1 - \frac{Var(Y')}{Var(X)}$.

Skewness

Skewness is defined as a measure of symmetry or lack of symmetry. A distribution of data is considered symmetric if left and right of its center point look the same. Degree of asymmetry of values around the mean value for a univariate data Y_t can be calculated using the skewness coefficient $S = \frac{1}{n\sigma^3} \sum_{i=1}^{n} (Y_t - \bar{Y_t})^3$, where, $\bar{Y_t}$ is the mean and σ is the standard deviation and n is the number of data points. Skewness of normal distribution is zero. Negative values of skewness indicate the data are skewed left and positive values indicate the data is skewed right. Skewness is actually a measure to show if the data distribution is heavy tail. Frequency skewness is calculated in the same manner except the data is transformed to frequency domain.

Kurtosis

Kurtosis is a measure to show if distribution of data is peaked or flat comparing with a normal distribution. Kurtosis for a univariate time series Y_t can be calculated as follows: $K = \frac{1}{n\sigma^4} \sum_{i=1}^{n} (Y_t - \bar{Y_t})^4$. Since, the kurtosis for a standard normal distribution is 3, the excess kurtosis is $K - 3$. The standard normal distribution has a kurtosis of zero while positive kurtosis indicates a peaked distribution and negative kurtosis indicates a flat distribution.

Serial correlation

Serial correlation is another important properties of time series [WSH06]. To extract a measure which shows the degree of serial correlation of a dataset, autocorrelation function can be used. Autocorrelation, at a single time is, $r_k = Corr(Y_t, Y_{t-k})$ where k is time lag. The average autocorrelation is considered as degree of serial correlation.

Average bandwidth

Bandwidth is measure to show difference between lower and upper frequencies and is typically measured in hertz. Sometimes it is considered as the difference between the upper and lower cutoff frequencies (e.g. -3 dB). It is considered to show if frequency spectrum contains a wide range and is thick or not.

4. Experimental Results

Experimental results for evaluation of clustering algorithms and comparison with two other feature learning methods are presented in this section.

4.1 Shift-invariant clustering is meaningful

To understand Keogh's report, his experiment was replicated using Cylinder-Bell-Funnel time series in the UCR datasets [CKH+15]. For each pattern, 30 normalized instances were concatenated together with each instance having a length of 128. Then, k-means clustering (k = 3) was applied to the subsequences of the time series using a sliding window technique, with w = 128 and s = 1 (w and s represent window length and slide length). While we expected the features capture the three patterns, they were closely similar to sine waves, as shown in Figure 4.1. Figure 4.1 illustrates one sample of each cylinder, bell, and funnel patterns and their frequency spectrums while Figures 4.2 and 4.3 contain results of shift-variant and shift invariant algorithms on the CBF dataset respectively. To figure out what these features captured, we plotted power spectrum of each pattern that can be calculated using Discrete Fourier Transform, and

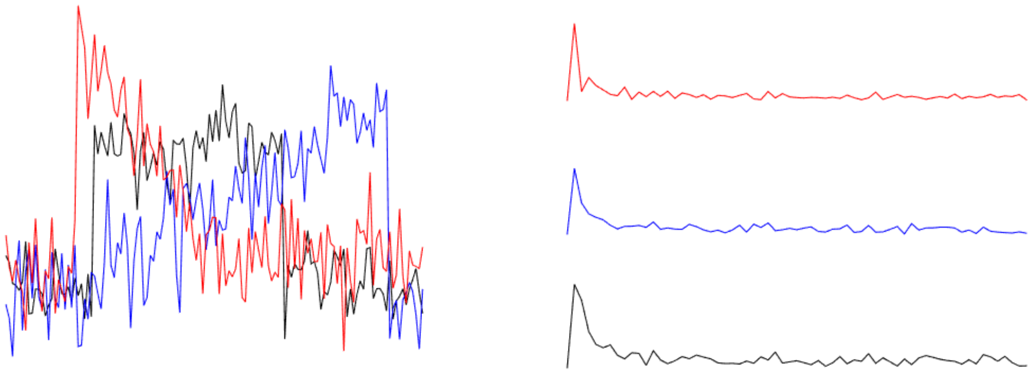

Fig. 4.1: a) patterns of CBF dataset, and b) their frequency spectrum.

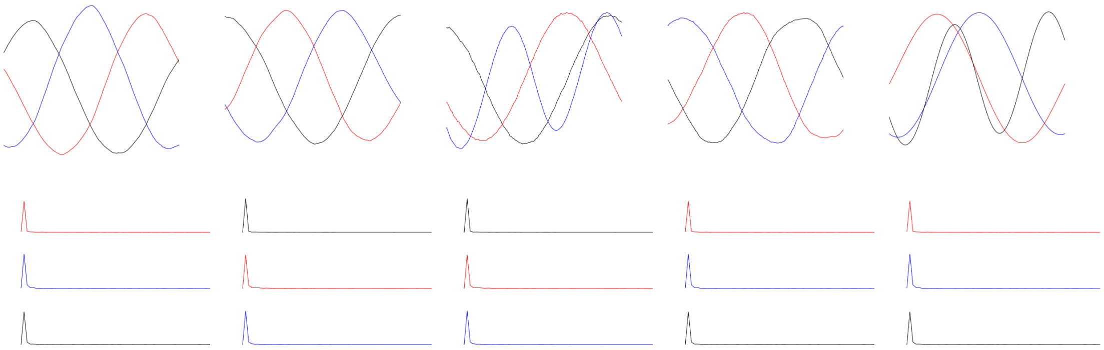

Fig. 4.2: Results of applying different shift-variant algorithms on Cylinder, Bell, Funnel dataset [KL05]. Top rows shows the learned kernels using a) k-means, b) spherical clustering, c) sparse coding, d) non-negative matrix factorization (NMF), and e) principal component analysis (PCA). The bottom row shows the corresponding power spectra.

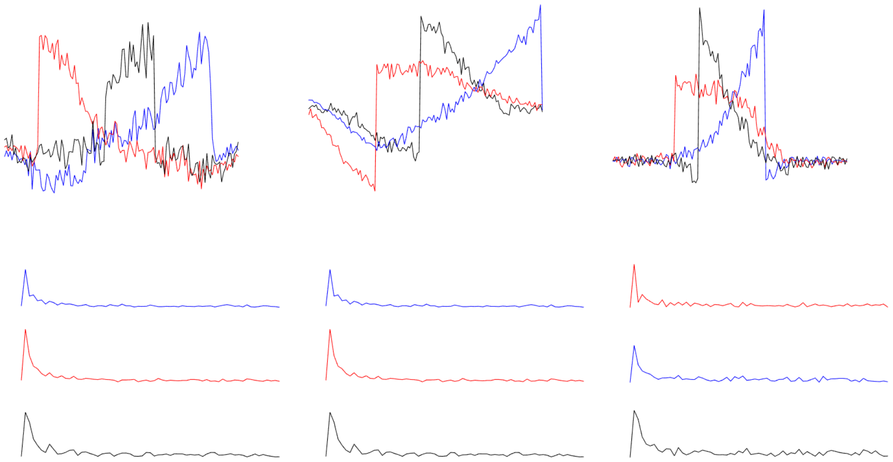

Fig. 4.3: Results of applying shift invariant versions of a) clustering, b) sparse coding, and c) non-negative matrix factorization on CBF dataset [KL05].

found all three patterns have strong peaks in the same frequency, however the same patterns in the dataset had different phases. Figure 4.1 shows power spectrum of a sample of each signal. As the plots show, the features capture the frequency of data.

To validate this claim, we generated a set of five pure tones with frequencies of 100, 500, 1000, 2000, and 3000 Hz. We concatenated the tones in three different orders and then concatenated the three signals to get one long signal. Then the signal was broken to the subsequences with maximum overlap. K-means clustering could find the frequencies, but the cluster centers were a combination of almost all frequencies. K-means algorithm with Euclidean distance as similarity measure is not able to separate different patterns when they are distributed in different phases. The learned features and their power spectrum are shown in figure 4.4.

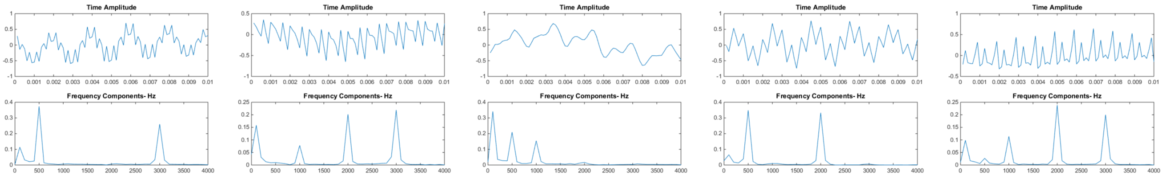

Fig. 4.4: Results of applying k-means on pure tone dataset. k-means is not able to separate the patterns in the dataset and the features capture multiple frequencies that are available in dataset.

Since, k-means clustering failed in our experiment, we applied four different algorithms to our synthesized dataset and summarized the results in table 4.1 where sph, omp1, shift inv. sph, and shift inv. omp1 stands for online clustering with cosine similarity measure, and shift-invariant online clustering. RMSE is mean of root square error between a pure tone and its best match cluster center, CF-Error is difference of center frequencies (location of strong peak in power spectrum) between set of pure tones and set of cluster centers, whereas Distance of frequency distribution is difference between frequency distribution of pure tone signals and the cluster centers which calculated by fast Fourier transform.

Table 4.1: Results of different clustering algorithms on pure tone dataset

	RMSE	**CF-Error**	**Dist. of frequency**
k-means	0.0124	3500	18.9475
Sph	0.0140	2300	29.5232
Shift inv. Sph.	**9.4247e-05**	**0**	**0.7861**

Shift-invariant algorithms were able to capture the frequencies and separate the patterns. Figure 4.5 shows the features learned by shift-invariant clustering.

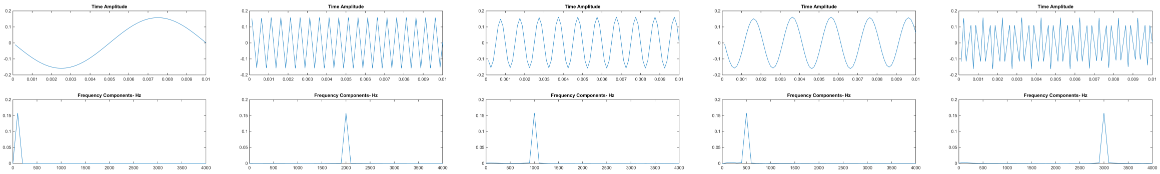

Fig. 4.5: Results of applying shift-invariant spherical clustering on pure tone dataset. Shift-invariant spherical clustering is able to separate the patterns in the dataset and each feature captures frequency of one pattern.

We also applied the shift-invariant algorithms on CBF dataset and got the patterns back. The results are shown in figure 4.2. Performance of shift-invariant spherical clustering algorithm, were also evaluated on UCR datasets based on entropy, F-measure as well as ratio of average within-cluster-distance and average between-cluster-distance. Eight datasets were selected to be comparable with results that were reported in [JJO11]. Figure 4.6 shows performance of our algorithm in comparison with clustering algorithm with different similarity measures with respect to entropy and F-measure.

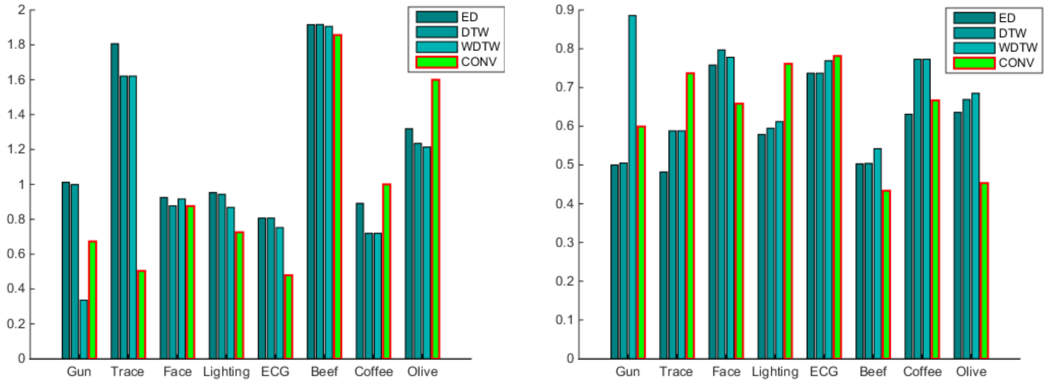

Fig. 4.6: Comparing performance of shift-invariant spherical clustering with other clustering algorithm which were presented in [JJO11]. ED: Euclidean distance, DTW: dynamic time warping, WDTW: weighted dynamic time warping and Conv: shift-invariant clustering.

Figure 4.7 shows performance of our algorithm in comparison with clustering algorithm with different similarity measures with respect to meaningfulness.

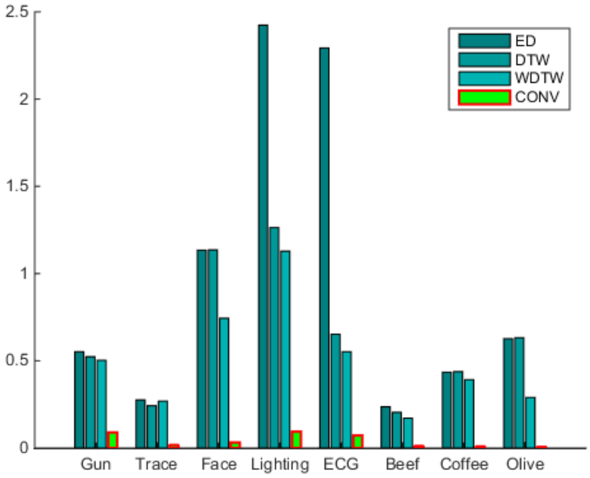

Fig. 4.7: Comparing performance of shift-invariant spherical clustering with other clustering algorithm which were presented in [JJO11]. ED: Euclidean distance, DTW: dynamic time warping, WDTW: weighted dynamic time warping and Conv: shift-invariant clustering.

The ratio of average within-cluster-distance and average between-cluster-distance is known to show meaningfulness of clustering. Nearest to zero is the best. The results show that in all of datasets shift-invariant clustering was able to learn meaningful features, however, performance of clustering is shown in entropy and F- measure plots. The lower the value of entropy, the higher the clustering quality, on the contrary, the higher the value of F-measure, the better the clustering quality. The results show that in order to get the best performance in clustering, choice of similarity measure plays an important role. For example if in a dataset, the position of each pattern is important, then shift-invariant approaches cannot be the best. We can also use a penalty measure for large lags same as the method which is used in [JJO11] for DTW.

4.2 Comparing shift-invariant unsupervised feature learning methods

To understand Keogh's report, his experiment was replicated using Cylinder-Bell-Funnel time series in the UCR datasets [CKH+15]. For each pattern, 30 normalized instances were concatenated together, with each instance having a length of 128. Then, k-means clustering (k = 3) was applied to the subsequences of the time series using a sliding window technique, with w = 128 and s = 1 (w and s represent window length and slide length). While we expected the features capture the three patterns, they were closely similar to sine waves, as shown in Figure 4.2.

To figure out what these features captured, we plotted power spectrum of each pattern that can be calculated using Discrete Fourier Transform and found that all three patterns have strong peaks in the same frequency; however the same patterns in the dataset had different phases.

4.2.1 Reconstruction

In this section the algorithms are evaluated with respect to signal reconstruction. Three sets of features were learned on TIMIT training set [ZSG90] and Stock price dataset (Standard and poor 500 closed price). Their reconstruction SNR is shown in figure 4.8. Reconstruction SNR using a dictionary of random Gaussian noise is also calculated to show the effect of feature learning in reconstruction. The results show ability of reconstruction depends on dataset. In the case of speech data, sparse coding learns the best feature, however, shift invariant clustering is more successful in stock price data.

4.2.2 Noise separation

In order to evaluate the algorithms in the noise separation application, a random subset of 1000 seconds of speech from TIMIT dataset were chosen for testing

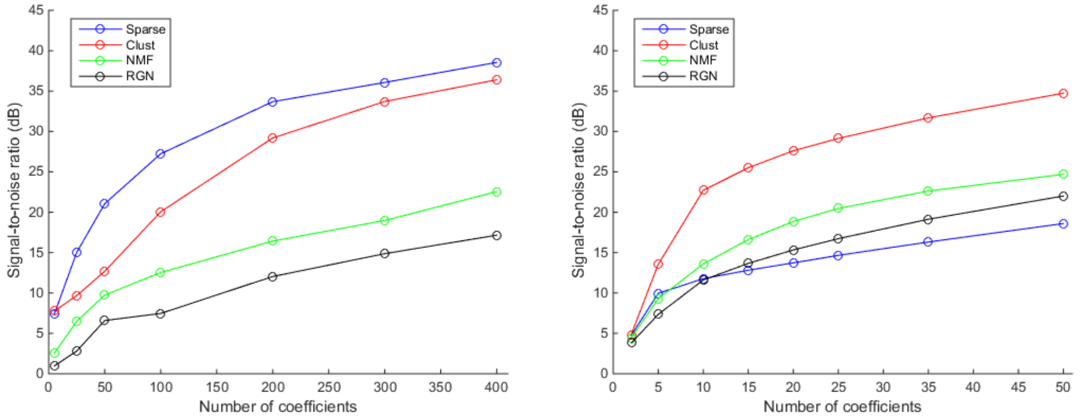

Fig. 4.8: Reconstruction SNR for TIMIT and Stock price

performance of two shift-invariant algorithms. Three kinds of noise including babble, pink, and white noise were down sampled to sampling frequency of 16 KHz.118 seconds of each were chosen for training and rest of 117 sec were excluded for testing. Length of window for all the experiments was 20 ms. Dictionaries of 50 atoms were learned using each algorithm on the four datasets. A Matching pursuit algorithm was used for sparse decomposition in encoding step. Slope of linear regression line for the last 5 points was used as a stopping criteria in learning section with 0.001 as a threshold. The largest value of ϵ-NSS is better. Figure 4.9 shows performance of algorithms for noise separation. The three algorithms are able to separate white noise from speech. However, separating babble noise and pink noise is more complicated. Babble noise is a mixture of speech and is difficult to be removed. Pink noises are different from speech; however, they have overlapped spectrum with speech and do not satisfy the noise assumption of sparse coding. In these experiments sparse coding performed better. Even though, if error tolerance is low, clustering also performed as well as sparse coding. Performance of non-negative matrix factorization is between sparse coding and clustering.

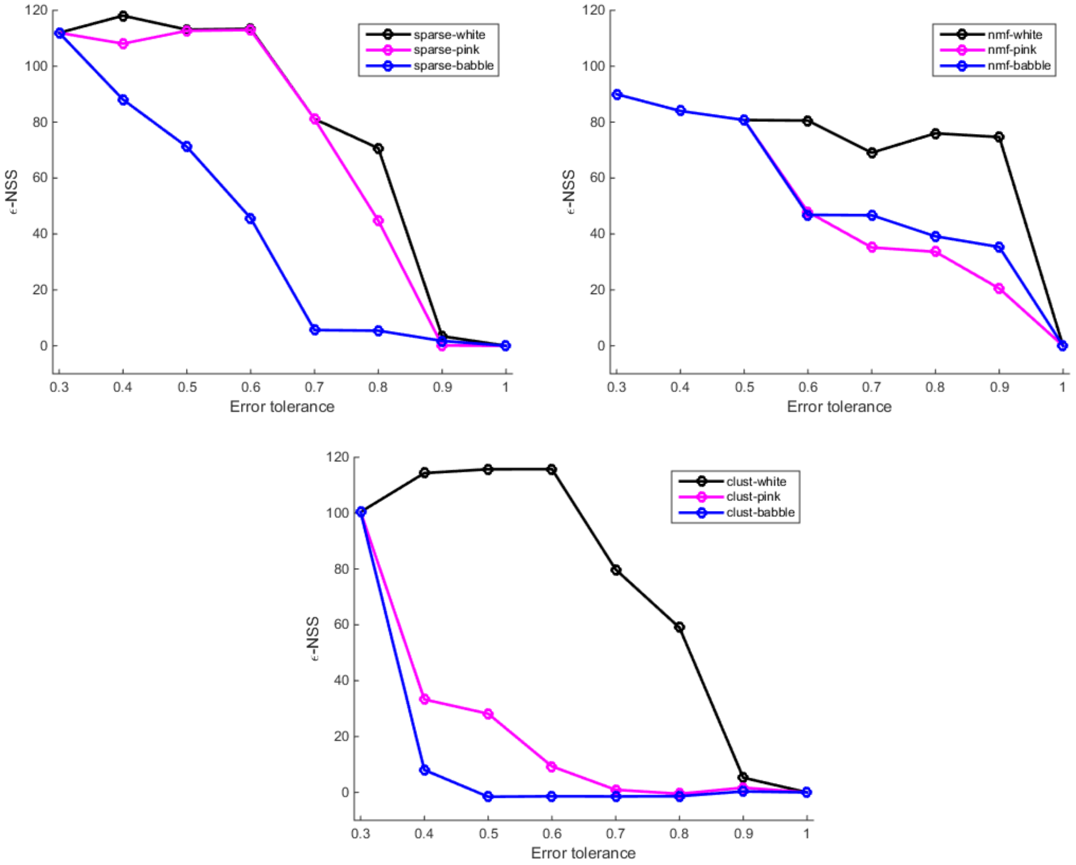

Fig. 4.9: Comparisons of e-NSS. Three noises, namely white, pink and babble noises are used.

4.2.3 Prediction

Three datasets are used to compare performance of algorithms in prediction task: 1) Darwin sea level pressures (SLP), which contains monthly values of the Darwin Sea Level Pressure series from 1882 to 1998, 2) Electricity demand (ELD), 15 minutes averaged values of power demand in the full year 1997, and 3) Standard and Poor 500 (SandP) daily stock price from 1960 till 2016. The results of one step prediction are shown in table 4.2 and table 4.3

Table 4.2: Prediction MAPE for the three datasets. Clust, Sparse and NMF refer to shift invariant versions of clustering, sparse coding and non-negative matrix factorization.RNN refers to recurrent neural network. Number of hidden units in RNN is considered equal to the number of features in the feature learning algorithms.

Algorithms	Clust	Sparse	NMF	RNN
SLP	12.7087	17.8825	16.0027	7.76
ElD	5.0779	34.8392	5.1932	1.899
SandP	1.4805	2.3810	1.7271	0.68

Table 4.3: Prediction RMSE for the three datasets. Clust, Sparse and NMF refer to shift invariant versions of clustering, sparse coding and non-negative matrix factorization. RNN refers to recurrent neural network. Number of hidden units in RNN is considered equal to the number of features in the feature learning algorithms.

Algorithms	Clust	Sparse	NMF	RNN
SLP	0.1724	0.2286	0.2196	0.1138
ElD	0.0640	0.3583	0.0683	0.02716
SandP	0.0196	0.0287	0.0246	0.0121

Figure 4.10 shows parts of predicted signals using the features learned from three algorithms.

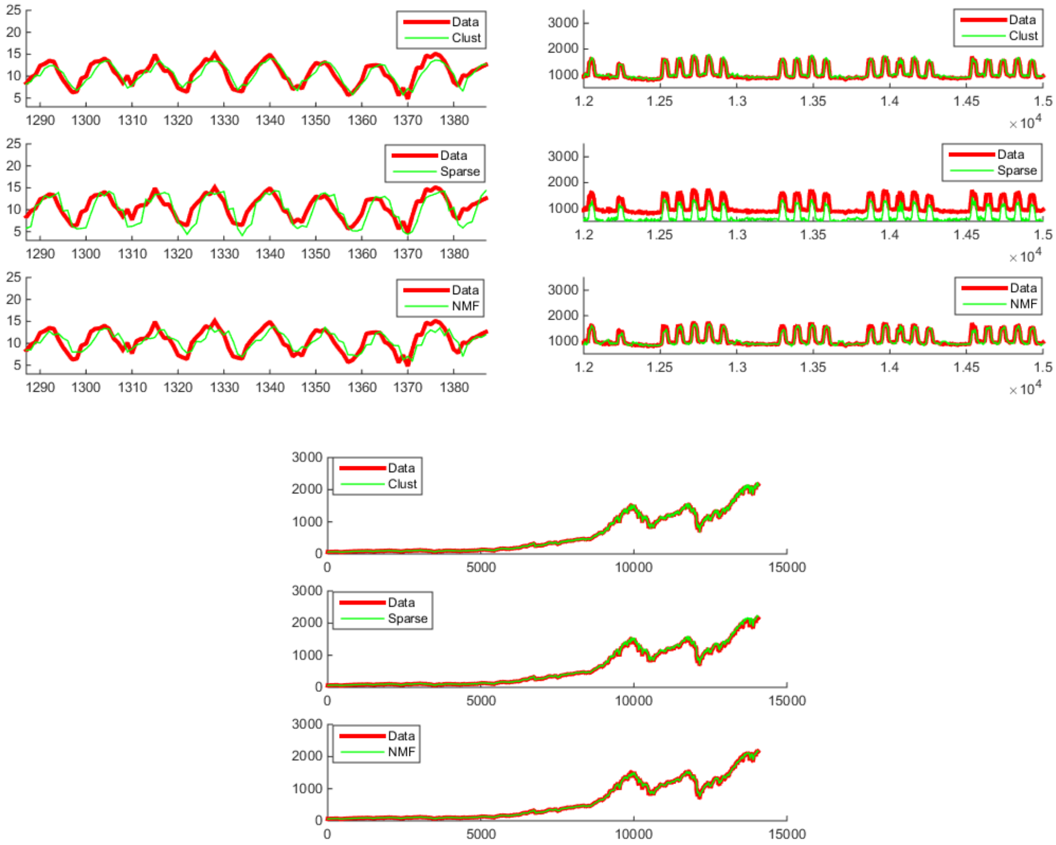

Fig. 4.10: Prediction results for SLP, ELD and SandP datasets

4.2.4 Classification

Clustering and sparse coding are widely used to learn features from data and classify the patterns using deep structures to get high accuracy. However, our goal is not finding the best accuracy but comparing three groups of unsupervised feature learning algorithms in time series classification. For this reason a simple classification algorithm (k-NN) has been chosen to classify the time series using learned features and in all of the experiments k is set to one. In learning part, maximum overlap was considered between consecutive windows. In inference part also two simple methods were chosen: 1) dot product of features with the time series and 2) results of cross correlation and their lags. These two inferences were used to feed 1-NN algorithm. For each dataset the number of learned features for clustering and sparse coding is the same.

Table 4.4: Information about datasets

Datasets	**Train-size**	**Test-size**	**length**	**# of classes**	**Type**
Beef	30	30	470	5	SPECTRO
BeetleFly	20	20	512	2	IMAGE
BirdChicken	20	20	512	2	IMAGE
CBF	30	900	128	3	SIMULATED
DistalPhalanx-OutlineAgeGroup	400	139	80	3	IMAGE
Earthquakes	322	139	512	2	SENSOR
ECG 200	100	100	96	2	ECG
ECG 5000	500	4500	140	5	ECG
ElectricDevices	8926	7711	96	7	DEVICE
Face (four)	24	85	350	4	IMAGE
Face (all)	560	1690	131	14	IMAGE
FacesUCR	200	2050	131	14	IMAGE
Fish	175	175	463	7	IMAGE
FordB	3636	810	500	2	SENSOR
Ham	109	105	431	2	SPECTRO
Strawberry	613	370	235	2	SPECTRO
Trace	100	100	275	4	SENSOR
TwoLeadECG	23	1139	82	2	ECG
Wine	57	54	234	2	SPECTRO
WordSynonyms	267	638	270	25	IMAGE
Worms	181	77	900	5	MOTION
WormsTwoClass	181	77	900	2	MOTION

Table 4.5: Classification Error Rate

Datasets	**clust-dot**	**clust-xcorr**	**sparse-dot**	**sparse-xcorr**	**nmf-dot**	**nmf-xcorr**	**Raw data**
Beef	**0.3667**	0.5333	0.4000	0.6333	0.4000	0.5333	0.333
BeetleFly	0.3500	0.6000	0.3000	0.5000	**0.2000**	0.5000	0.250
BirdChicken	0.5000	0.4000	**0.3500**	0.5500	0.4000	0.4500	0.450
CBF	0.1244	**0.0256**	0.1444	0.1978	0.1400	0.2400	0.148
DistalPhalanxOutlineAgeGroup	0.2800	0.2575	0.2575	0.2625	**0.2475**	0.3125	0.218
Earthquakes	0.3199	0.3602	0.3043	0.3571	**0.2646**	0.3665	0.326
ECG 200	0.2000	0.1600	0.1900	0.1900	**0.1000**	0.2100	0.120
ECG 5000	**0.0807**	0.1036	0.0829	0.829	0.0842	0.1004	0.075
ElectricDevices	0.4775	0.5676	0.4551	0.5656	**0.4462**	0.5274	0.450
Face (four)	**0.2614**	0.3977	0.2955	0.4659	0.3864	0.5455	0.216
Face (all)	**0.4473**	0.5633	0.4888	0.5787	0.6503	0.5811	0.286
FacesUCR	0.5776	0.5137	**0.4615**	0.5824	0.7288	0.6273	0.231
Fish	0.5943	0.5943	**0.3657**	0.5771	0.5314	0.5886	0.217
FordB	0.4607	**0.4587**	0.4849	0.4788	0.4970	0.4956	0.442
Ham	0.4095	0.4762	**0.3333**	0.4476	0.4970	0.4476	0.400
Strawberry	0.1207	0.1240	**0.0930**	0.1207	0.1354	0.1289	0.062
Trace	0.3000	0.3000	**0.1100**	0.2700	0.2800	0.1900	0.240
TwoLeadECG	0.3626	0.3433	**0.2133**	0.2968	0.4390	0.3863	0.253
Wine	**0.2407**	0.5556	0.4074	0.5370	**0.2407**	0.4444	0.389
WordSynonyms	0.4337	0.5096	0.4984	**0.3809**	0.4310	0.7132	0.382
Worms	0.6133	0.6575	0.6188	**0.5912**	0.6409	0.6630	0.635
WormsTwoClass	0.4144	0.4530	**0.3702**	0.4144	0.4586	0.4972	0.414

Features were learned in one layer and maximum sparsity level in shift-invariant sparse coding was equal to number of features. The results were compared with 1-NN with Euclidean distance in the raw data obtained from [CKH+15]. Table 4.4 contains information about 22 of UCR time series datasets [CKH+15]. Table 4.5 illustrates error rate of Classification.

Classification for the above datasets were also done using recurrent neural network (RNN). Number of hidden units are considered equal to the number of features that were learned in feature learning algorithms. Results are shown in table 4.6. Since there were fluctuations in error rate, we ran the experiments five times and the results are average of the five errors.

Table 4.6: Classification Error Rate using recurrent neural network (RNN)

Datasets	**RNN**
Beef	0.18002
BeetleFly	0.15
BirdChicken	0.26
CBF	0.19416
DistalPhalanxOutlineAgeGroup	0.2008
Earthquakes	0.15098
ECG 200	0.2986
ECG 5000	0.0596
ElectricDevices	0.3785
Face (four)	0.1892
Face (all)	0.1696
FacesUCR	0.3525
Fish	0.1491
FordB	0.092
Ham	0.458
Strawberry	0.026
Trace	0.253
TwoLeadECG	0.1029
Wine	0.2376
WordSynonyms	0.6092
Worms	0.5399
WormsTwoClass	0.4031

4.2.5 Simulating auditory signals

A quantitative comparison between three sets of features which have been learned using three algorithms is needed. Three different codes were optimized to represent speech (sph), environmental sound (Env), and vocalization (Voc) using the three algorithms. For each learned kernel function in the given code, the best matching recover filter was found from the Gamma chirp functions which is a parameterized model of cochlear filters.

Figure 4.11 shows the distribution of correlation coefficients of active kernel functions where the red line is the median of the coefficients values for that code. The 25^{th} and 75^{th} quartiles are shown by the lower and upper edges of the box while the whiskers indicate the 5^{th} and 95^{th} percentiles. Outliers are shown with red pluses.

Efficient codes for speech is significantly better predictors of the cochlear code approaching the fitted gammatone model in accuracy consistent with results in the literature. In all of algorithms environmental sound has higher median in correlation coefficients in comparison with animal vocalization. Correlation coefficients with Gaussian white noise are also included to be compared with learned features and illustrate effect of learning. Shift invariant sparse coding outperformed the other two feature learning algorithms. Shift invariant clustering has the same median but there some outliers in the correlation results. In all of algorithms

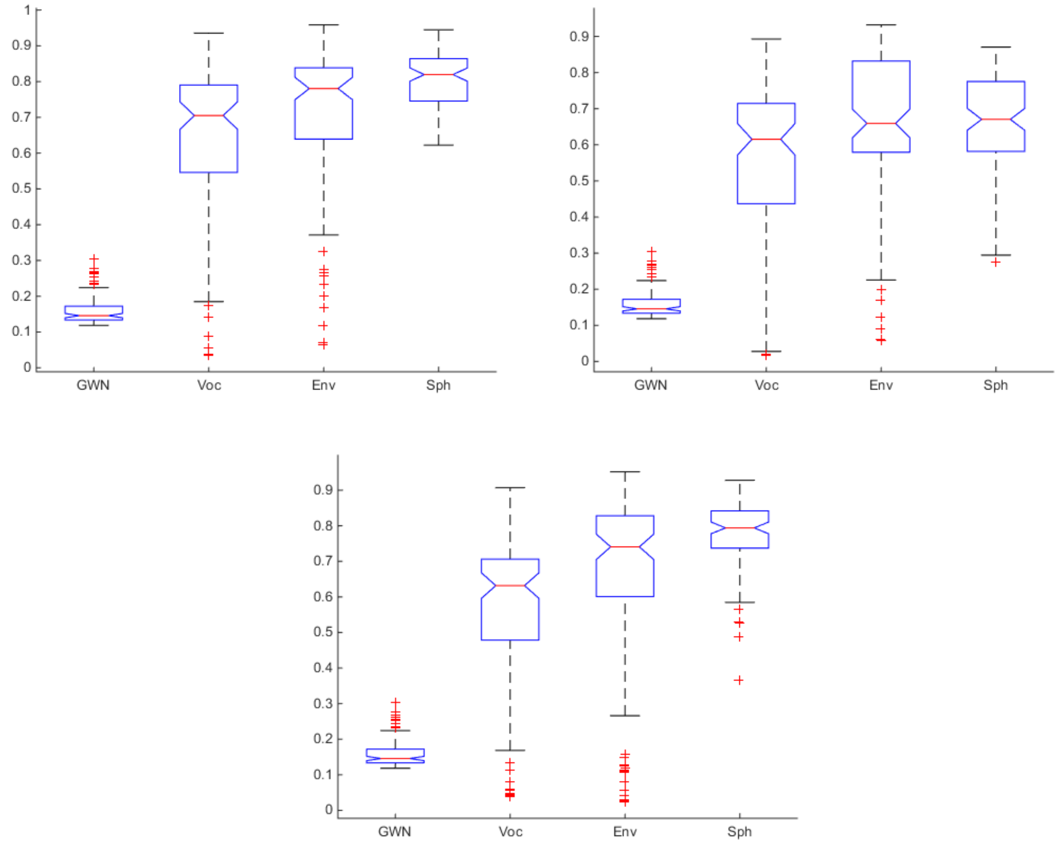

Fig. 4.11: Distribution of correlation coefficients of active kernel functions. a) sparse coding, b) Non-negative matrix factorization and, c) Clustering. GWN, Voc, Env and Sph refer to Gaussian white noise, vocalization, environmental sound and speech.

4.3 Effect of time series components

In this section we want to figure out how time series structural characteristics affect performance of feature learning algorithms. First, eight features from raw time series, called degrees of trend, seasonality, skewness, frequency skewness, kurtosis, frequency kurtosis, serial correlation, and frequency bandwidth have been calculated from 23 datasets of time series. Then coefficient of variation ($c_v = \sigma/\mu$) of three feature learning algorithms is calculated for all datasets. Table 4.7 shows the results for structural characteristics, while, tables 4.8 contains C_v results for shift invariant feature learning algorithms.

Table 4.7: Structural characteristics of dataset: trend, Seas: seasonality, Skew: Skewness, F.Skew: Frequency skewness, Kurt: Kurtosis, F.Kurt: Frequency Kurtosis and Scorr: serial correlation, along with C_v coefficient of variation which is degree of clustering successfulness

Datasets	*Trend*	*Seas*	*Skew*	*F.Skew*	*Kurt*	*F.Kurt*	*Scorr*
Beef	.9988	-.0014	1e-4	.0082	.028	.2527	.1489
BeetleFly	.9994	6e-4	5e-4	.0046	.0257	.2076	.1345
BirdChicken	.999	2e-4	1e-4	.0037	.0376	.4259	.2655
CBF	.8350	.0711	-.0062	.0193	.0999	.5422	.2042
DistalPhalanxAgeGroup	.9929	.0157	.0028	.0218	.0767	.2857	.1521
Earthquakes	.4451	.0029	.0071	.0294	.0027	.0126	.0219
*ECG*200	.9834	.0039	-.0032	.0234	.1008	.5832	.2402
*ECG*5000	.7999	3e-4	-.0125	.0665	.0463	.2205	.1580
ElectricDevices	.3918	-2e-16	.0624	.4	.0011	.0451	.0258
Face(four)	.9341	.0012	6e-4	.0099	.0158	.0625	.0673
Face(all)	.9692	.0014	3e-4	.0228	.028	.0669	.0767
FacesUCR	.992	-.0056	-.0061	.0342	.0382	.1207	.1080
Fish	.9994	.0013	3e-4	.0039	.0624	.9324	.3116
FordB	.9040	5e-4	1e-4	.0046	.0181	.0950	.0838
Ham	.9886	.0027	-4e-4	.0046	.0183	.0950	.0919
Phoneme	.2988	.0278	2e-4	.0030	.0066	.0343	.0482
Strawberry	.9978	-2e-4	.006	.0198	.0506	.3558	.2069
Trace	.9974	-6e-5	-.0012	.015	.0414	.321	.1682
TwoLeadECG	.9902	.0085	-.023	.0668	.0761	.3076	.1932
Wine	.997	2e-5	.005	.0198	.0342	.1701	.1385
WordSynonyms	.987	6e-6	.0066	.018	.0302	.148	.1286
Worms	.999	9e-4	5e-4	.0029	.024	.3418	.1819
WormsTwoClass	.999	-7e-4	5e-4	.0029	.0241	.3418	.1819

Table 4.8: Coefficient of variation C_v for the three feature learning algorithms on the 23 datasets.

Datasets	*Clust*	*Sparse*	*NMF*
Beef	16.74	8.4259	7.03
BeetleFly	9.7565	8.7096	6.46
BirdChicken	12.2675	8.4241	9.54
CBF	15.3797	17.4542	2.73
DistalPhalanxAgeGroup	34.2150	19.2665	9.81
Earthquakes	3.2724	6.7169	1.9
*ECG*200	19.1786	23.7957	2.95
*ECG*5000	12.3258	19.3985	2.54
ElectricDevices	3.6254	5.67	2.76
Face(four)	7.5113	4.2001	20.03
Face(all)	5.4362	9.0601	1.91
FacesUCR	5.6230	6.9284	3.76
Fish	47.9050	49.3436	2.58
FordB	5.7346	2.7673	6.31
Ham	19.3397	3.5637	6.51
Phoneme	3.3590	12.5585	5.54
Strawberry	23.6463	15.3873	4.81
Trace	14.1564	9.6931	5.34
TwoLeadECG	27.9643	6.1959	5.41
Wine	24.6635	4.5112	6.51
WordSynonyms	7.3467	7.3776	2.72
Worms	6.1156	5.3354	1.9
WormsTwoClass	19.1786	5.3554	4.77

Correlation coefficients of C_v with different structural features, show significant relationship between clustering features performance and trend, frequency skewness, frequency kurtosis, same similarity and frequency bandwidth. Table 4.9 shows the Pearson correlation and spearman correlation coefficients and their significance level. The results show that shift invariant clustering algorithms perform better if the trend degree of time series is higher, frequency distribution is asymmetric and has a stronger peak (dense) with heavy tail and same similarity of data is high, whereas the performance decreases when the frequency bandwidth increases.

Table 4.9: Correlation of the structural features with shift invariant clustering performance. * indicates p-value $<$ 0.05 and ** shows p-value $<$0.01

Features	Trend	F.Skew	F.Kurtosis	SSim	BW$_{avg}$
Pearson	0.42*	0.622**	0.696**	0.655**	-0.387*
Spearman	0.459*	0.748**	0.645**	0.677**	-0.393*

However, sparse coding has positive correlation with frequency skewness, frequency kurtosis and same similarity of data which is consistent with our finding in the experimental results. The results are shown in table 4.10.

Table 4.10: Correlation of the structural features with shift invariant sparse coding performance. * indicates p-value $<$ 0.05 and ** shows p-value $<$0.01

Features	F.Skew	F.Kurtosis	SSim
Pearson	0.598**	0.815**	0.641**
Spearman	0.735**	0.516*	0.495*

Shift invariant NMF shows similarities to both clustering and sparse coding as expected. It has positive correlation with trend, frequency skewness, frequency kurtosis and same similarity while the correlation with frequncy bandwidth is negative. Table

4.11 contains the results.

Table 4.11: Correlation of the structural features with shift invariant NMF performance. * indicates p-value < 0.05 and ** shows p-value <0.01

Features	Trend	F.Skew	F.Kurtosis	SSim	BW_{avg}
Pearson	0.352*	0.227	0.765**	0.716**	-0.433*
Spearman	0.697**	0.566**	0.664**	0.674**	-0.601**

4.4 Number of features

Number of features play an important role in performance and efficiency of algorithms. In this section we show that number of features that should be learned depends on size of dataset and structure of data. All other factors remain the same during the experiments. Three different datasets are used to determine how number of features affect performance of algorithms, namely standard and poor stock price, ECG dataset and German emotional speech dataset. Part of datasets ($20\% of dataset$) is used for test. The learned features are used with the same objective function as learning procedure but not updated. Activity of features are counted and the features with activities of more than 10% of the median of all activations are considered as useful. The activation rate of dictionaries is calculated as ratio of useful features over number of features in the dictionary. The results are shown in Figure 4.12.

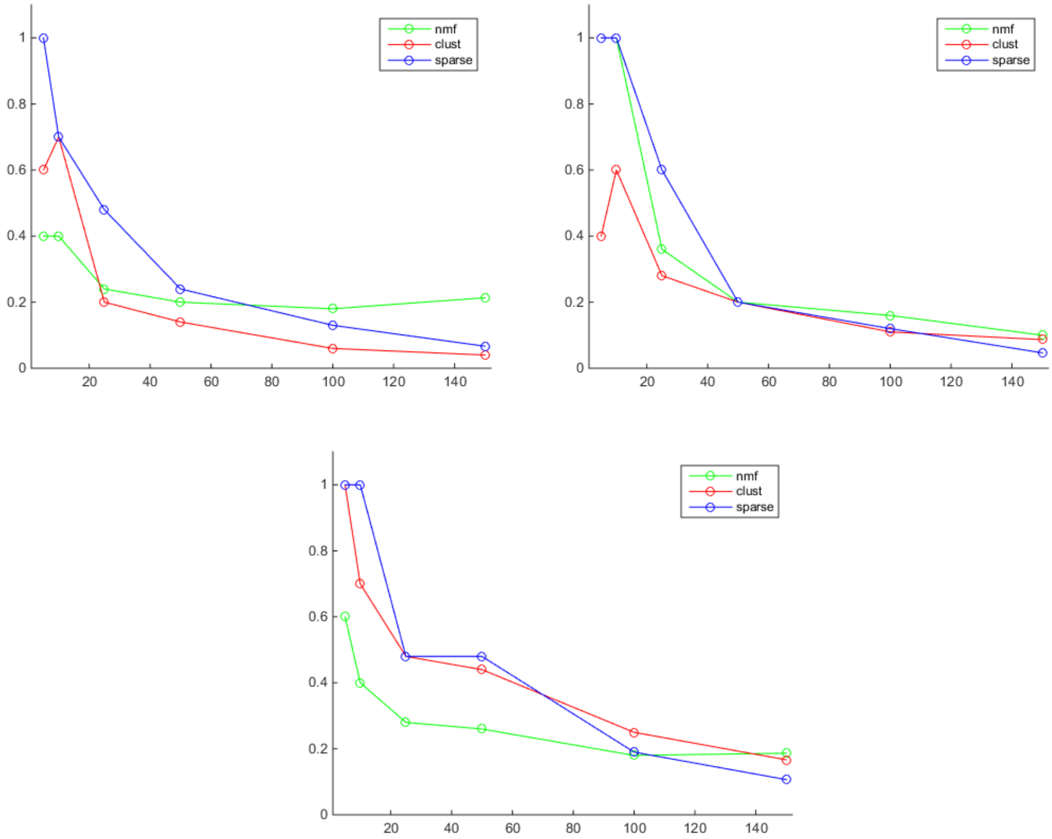

Fig. 4.12: Activity rate of dictionaries of 5, 10, 25, 50, 100 and 150 features learned using the three algorithms on three datasets: a. Stock price, b. ECG, and c. German emotional speech.

If a threshold of 0.2 is chosen to characterise a dictionary as active, from figures it is shown that for stock price, clustering and NMF need to be initialized by 25 features but sparse coding needs 50 features. For ECG all dictionaries should be initialized with 50 features and for German emotional speech we need dictionaries of 100 features. Size of dataset is 14000 for stock price, 19000 for ECG, and 380160 for speech.

There is a strong relationship between size of dataset and activation of dictionary. Furthermore, there is an evidence of relationship between dictionary activation and global characteristics of dataset since for a dataset with high degree of trend (stock price), clustering and NMF needs less number of features than sparse coding.

4.5 Size of receptive field

Size of receptive field is another factor that affects performance of feature learning algorithms. In this section dictionary of features using the three algorithms are leaned on different length receptive fields. The same datasets in the last section are used and number of features are also selected based on results of last section. Quality of representation is measured with the same objective function that is used in learning phase for each algorithm so results of different algorithm should not be compared with other algorithms. Sparsity level for sparse coding is fixed on 15% of features.

Encoding phase in sparse coding and non-negative matrix factorization is done with matching pursuit. Sparsity constraints of each algorithm is used. Since, matching pursuit is a strong encoder and try to reconstruct data with every kinds of features, the features that are not touched or touched only one time are ignored to remove effect of noise (We have seen normal Gaussian noise performed better than sparse coding features in reconstructing stock price.) The results are illustrated in Table 4.13

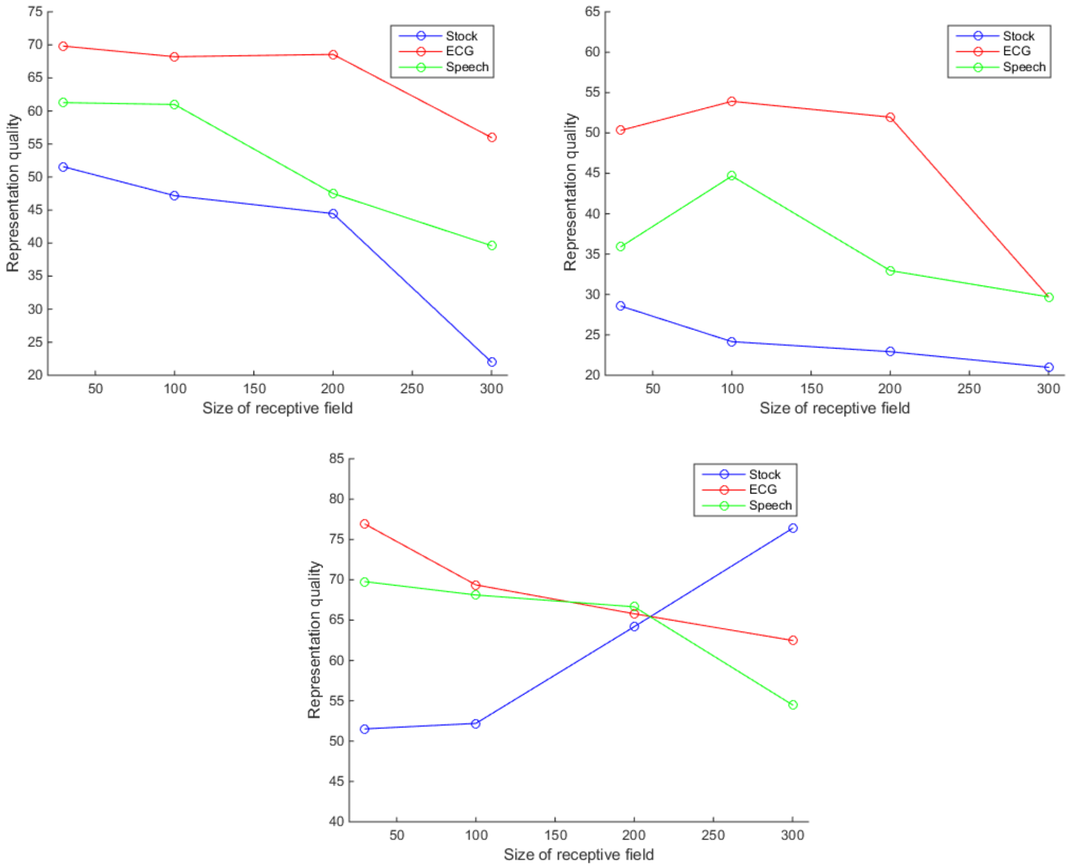

Fig. 4.13: Quality of representation against size of receptive field using the three algorithms on stock price, ECG, and German emotional speech datasets. a) sparse coding b) NMF, and c) clustering

The results show that quality of features decrease as size of receptive field increase in sparse coding. In non-negative matrix factorization the pattern is not as regular than sparse coding but the performance decrease with increasing size of receptive field in general. Clustering has different behaviour in different datasets. In a stock price which contains time series with high degree of trend, increasing receptive field size, increase quality of representation, however, in the two other datasets it is inverse.

4.6 Inferring hearing loss from speech

All recorded data was downsampled from 44.1 to 16 KHz. The kernels of length 320 samples (20ms) are learned from normalized time-amplitude speech windows of 200 ms duration. Hence the learned kernels are also time-amplitude signals; they resemble the gammatone filters. The frequency components of a kernel determine its tuning properties, with the most dominant component being its CF. Unimportant and harmful features were removed from all sets of features.

The kernels learned from each of our subjects were evaluated based on neurophysiological metrics. In order to show degree of loss of characteristic frequencies, distance between distribution of CFs from each subject's features with respect to distribution of CFs from TIMIT dataset features was found using Kolmogorov-Smirnov statistic. Since, TIMIT is a dataset of many people's speech, all possible ranges of CFs are existed in its set of features. Slope of the linear regression from the Q_{10} vs. CF plot were also found. Pearson correlation of these three features with result of AzBio test, PTA and hearing loss age of onset were found.

The features calculated from speech kernels are identified from the literature as salient features that clearly discriminate between normal and hearing-impaired individuals based on their tuning properties in the peripheral auditory pathway and the three features came from data are the features that show the factors which might affect speech of hearing impaired subjects. The results are shown in table 4.12.

Table 4.12: Correlation Analysis Results ($* = p < 0.05, ** = p < 0.01, *** = p < 0.005$)

	$LossCF$	$Slope_{Q_{10}}$
$AzBio$	-0.417 **	0.596 ***
PTA	0.40 *	-0.51 ***
$Ageofonset$	-0.56 ***	0.3393

As we expected, loss of characteristic frequency has a significant correlation with all three features. It has a positive correlation with PTA and negative correlation with $AzBio$. It means as we move from normal hearing people to hearing impaired with higher level of hearing loss, $LossCF$ increases. It has a negative correlation with hearing loss age of onset which means subjects who lost their hearing ability at birth or early ages have more $LossCF$, whereas people who lost their ability of hearing in older ages, have a similar characteristic frequency distribution to normal hearing people.

Consistent with previous findings, it shows people cannot produce frequency spectrums which have not heard. $Slope_{Q_{10}}$ has a very significant correlation with Azbio test results and PTA which shows $Slope_{Q_{10}}$ decreases as we move from normal hearing subjects to hearing impaired with severe hearing loss.

Then, the features were used for spike coding. For comparison, a 200 ms window of speech with a wide range of frequencies from TIMIT dataset is chosen to show significant differences in auditory representation of a typical speaker and a hearing impaired speaker using the predicted cochlea filters. Figure 4.14 shows the time amplitude signal, its spectrogram which is a visual representation of the spectrum of frequencies and two spikegrams, one from a normal hearing subject and one from hearing impaired subject.

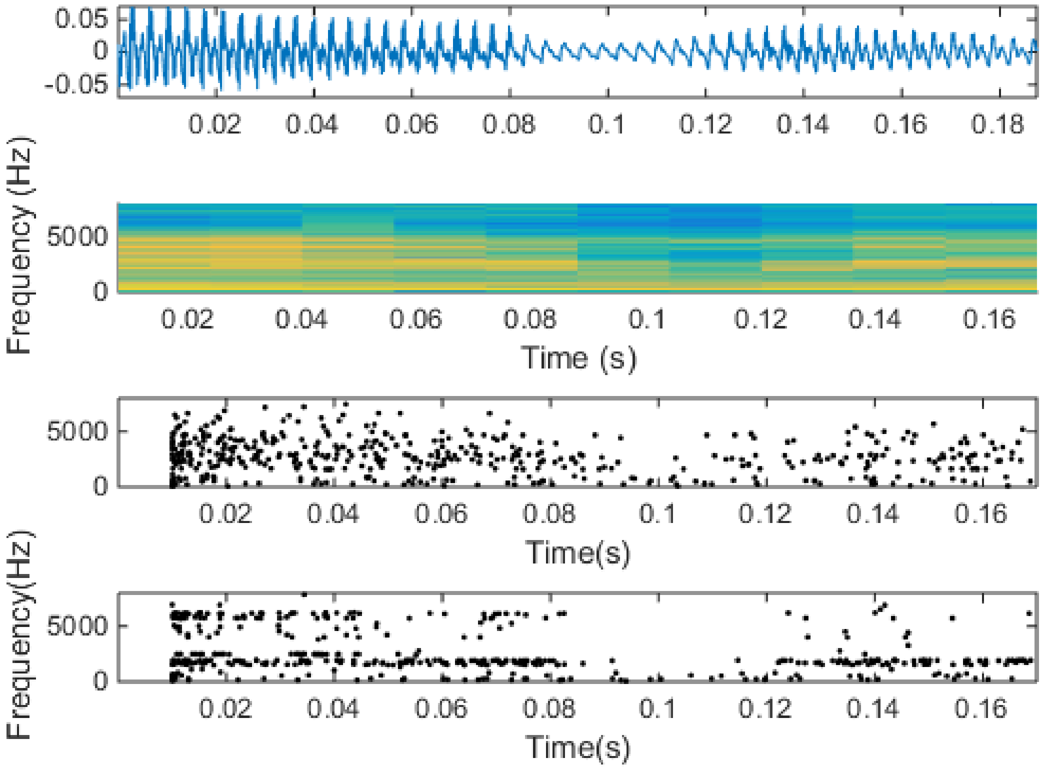

Fig. 4.14: Spikegrams for 200 ms of speech signal

To test ability of response to different frequencies in all subjects, the kernels were used for spike coding of puretones in the range of human hearing. The histogram of response to frequencies along with audiogram of a normal hearing subject, a subject with moderate hearing loss in some areas and a profound hearing loss subject as well as distribution of responses are shown in Figure 4.14. Since the puretones were uniformly distributed, the ideal distribution of response to frequencies should be a diagonal line but since only 32 features are learned, the plot for normal hearing subject is also an approximation of the line. The results are shown in 4.15 Furthermore, the curve in the audiogram is not the only factor which reflects frequency selectivity of a person. Information of other features which affect distribution of response are shown in Table 4.13. PTA is not shown in the table because it is directly calculated from audiogram.

An audiogram is a graphical representation of an individual's hearing sensitivity that plots the softest sound an individual can hear (threshold of audibility) as a function of frequency. Listeners are presented with 365 puretone stimuli at octave frequencies from 250 to 8000 Hz and are instructed to indicate the softest sound they can hear. This threshold level is then plotted for each ear separately on the audiogram with O's representing the right ear and X's representing the left ear. The hearing aid device was removed during this test but it was used during *AzBio* test. As it is shown in the literature, hearing impaired subjects have issues in high frequency regions, however, as the level of hearing loss decreases, there is an improvement in distribution of response. To have a more comprehensive comparison, the subjects were divided to three groups based on their audiograms: Normal hearing, Moderate hearing loss and Severe hearing loss. Average frequency selectivity of subjects was calculated for the three groups.

Figure 4.16 shows the quantile-quantile (q-q) plot for the three groups. A q-q plot is a graphical tool to determine if two groups of data come from the same distribution. The average frequency selectivity for the normal hearing group is as our expectation, even though only three normal hearing subjects are available. Having an approximation of the area with less frequency selectivity is a very helpful tool for audiologists to tune the cochlear implant.

Table 4.13: Information of subjects whose audiograms are shown.

$LevelofHearingLoss$	$Moderate$	$Severe$
AzBio	72.62	0
Age of onset	17	At birth

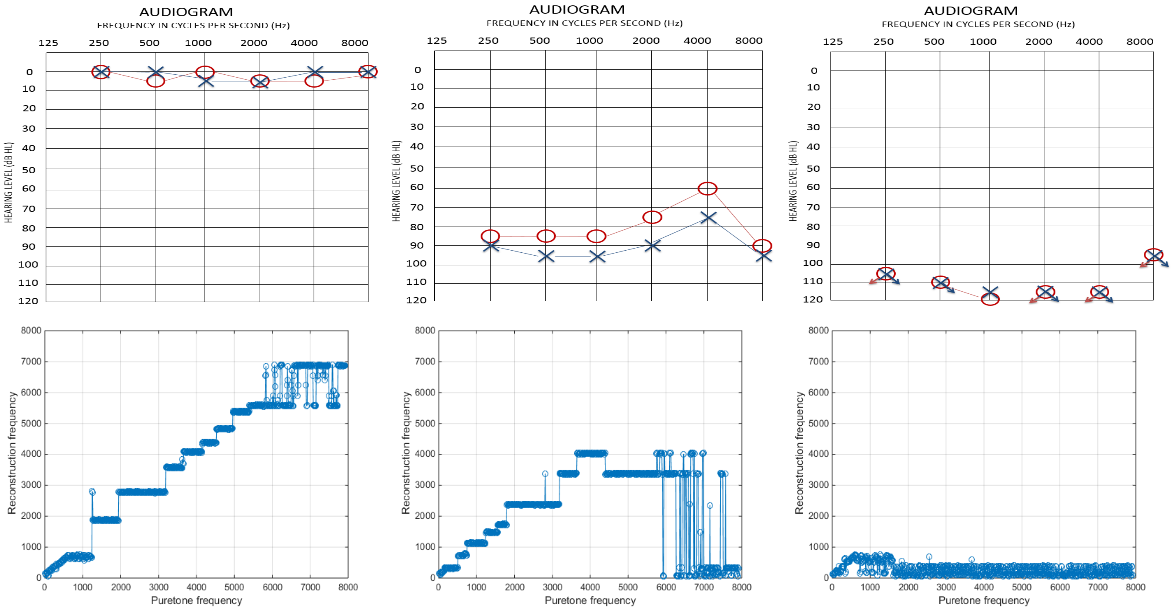

Fig. 4.15: Audiogram and frequency selectivity distribution for a normal hearing subject, a subject with moderate hearing loss and a subject with severe hearing loss.

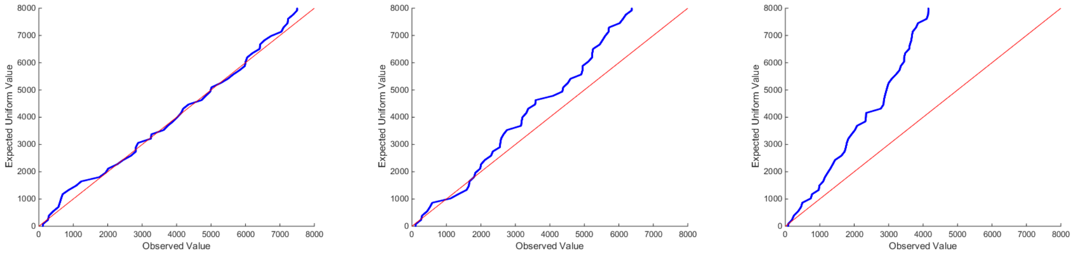

Fig. 4.16: Q-Q plot of three groups of subjects. Group1 indicates normal hearing subjects, Group2 and Group3 are subjects with moderate hearing loss and severe hearing loss, respectively.

5. Conclusion

In this thesis a detailed study of clustering of subsequences of time series is presented and shift invariant spherical clustering is introduced as a systematic approach of learning meaningful features from time series. Then, the features learned using shift invariant clustering is compared with other widely used unsupervised feature learning methods, shift invariant sparse coding and shift invariant non-negative matrix factorization, in five tasks: reconstruction, noise separation, prediction, classification, and simulating auditory filters from acoustic signals.

The results showed while clustering is very efficient and highly scalable, it can produce results which are quite close to other two feature learning methods and in some cases it it even more successful. In the task of prediction, clustering acquired more accuracy in the three different datasets. In classification, clusteing and sparse coding performed quite close in the 22 benchmark datasets. Results of classification using features were also compared with classification on raw data and in most of the datasets the important information of data were not lost and in many datasets feature learning improved the classification accuracy. In the task of noise separation, sparse coding generated the best results because it not only was able to separate white noise from speech but also more complicated noises and noises similar to speech. In the task of reconstruction, sparse coding was the best in speech reconstruction but clustering won the competition in reconstructing stock price signals. In order to simulate auditory filters from speech, both clustering and sparse coding were successful but sparse coding generated more efficient features.

The results were also analyzed with respect to the factors that may affect performance of algorithms. We showed that if a dataset contains time series with high degree of trend and serial correlation, the clustering algorithm is the best feature learning approach, however, if the average bandwidth of dataset is high, it is better if the features learn by sparse coding. Higher degree of frequency kurtosis and skewness increase performance of all three algorithms. Furthermore, number of features that should be learned is a function of size of dataset.

Since, in speech datasets, sparse coding generated features with higher quality, shift invariant sparse coding were applied on data of hearing impaired subjects and was able to successfully infer nature of hearing loss from their speech.

Publications:

Related publications:

- Bonny Banerjee, Masoumeh Heidari Kapourchali, Shamima Najnin, Lisa Lucks Mendel, Sungmin Lee, Chhayakanta Patro and Monique Pousson, ”Inferring hearing loss from learned speech kernels”, IEEE International Conference on Machine Learning and Applications, Anaheim, CA., December 18-20, 2016.[Acceptance Rate (Regular Papers): 24.69%]
- Masoumeh Heidari Kapourchali and Bonny Banerjee, ”Analysis of clustering and sparse coding for feature learning from time-series”, 15th Neural Computation and Psychology Workshop, Philadelphia, PA. , August 8-9, 2016.

Other publications:

- Shamima Najnin, Bonny Banerjee, Lisa Lucks Mendel, Masoumeh Heidari Kapourchali, Jayanta Kumar Dutta, Sungmin Lee, Chhayakanta Patro and Monique Pousson, ”Identifying hearing loss from learned speech kernels”, INTERSPEECH, September 8-12, San Francisco, CA., July 17-21, 2016.
- Lisa L. Mendel, Bonny Banerjee, Chhayakant Patro, Sungmin Lee, Monique Pous-

son, Shamima Najnin, Jayanta K. Dutta, Masoumeh H. Kapourchali, ”Tuning cochlear implants using patients’ speech production errors”, XXXIII World Congress of Audiology, Vancouver, Canada , September 18-21, 2016.

Bibliography

[AEB06] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 4, 7

[AL01] David Attwell and Simon B Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145, 2001. 5

[Alt92] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. 9

[Beh03] Sven Behnke. Discovering hierarchical speech features using convolutional non-negative matrix factorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2758–2763. IEEE, 2003. 5, 7

[BL14] Hilton Bristow and Simon Lucey. Optimization methods for convolutional sparse coding. *arXiv preprint arXiv:1406.2407*, 2014. 5

[CCMT90] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990. 12

[Che05] Jason R Chen. Making subsequence time series clustering meaningful. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005. 4

[Che07] Jason R Chen. Useful clustering outcomes from meaningful time series clustering. In *Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70*, pages 101–109. Australian Computer Society, Inc., 2007. 4

[CHKB13] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo EAPA Batista. Dtw-d: time series semi-supervised learning from a single example. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–391. ACM, 2013. 4

[CKH$^+$15] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/. 15, 20, 27

[CLN10] Adam Coates, Honglak Lee, and Andrew Y Ng. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001(48109):2, 2010. 1, 4

[CN11] Adam Coates and Andrew Y Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 921–928, 2011. 1

[Coa12] Adam Coates. *Demystifying unsupervised feature learning*. PhD thesis, Stanford University, 2012. 9

[CPR13] Rakesh Chalasani, Jose C Principe, and Naveen Ramakrishnan. A fast proximal method for convolutional sparse coding. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–5. IEEE, 2013. 5

[DBD09] Anne M Denton, Christopher A Besemann, and Dietmar H Dorr. Pattern-based time-series subsequence clustering using radial distribution functions. *Knowledge and Information Systems*, 18(1):1–27, 2009. 4

[DLM$^+$98] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *KDD*, volume 98, pages 16–22, 1998. 4

[FCNL01] Tak-chung Fu, Fu-lai Chung, Vincent Ng, and Robert Luk. Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, pages 26–29. Citeseer, 2001. 4

[FRG14] Pedro A Forero, Ketan Rajawat, and Georgios B Giannakis. Prediction of partially observed dynamical processes over networks via dictionary learning. *IEEE Transactions on Signal Processing*, 62(13):3305–3320, 2014. 5, 9

[HCSH15] Yongjun He, Deyun Chen, Guanglu Sun, and Jiqing Han. Dictionary evaluation and optimization for sparse coding based speech processing. *Information Sciences*, 310:77–96, 2015. 2, 5, 8, 10

[HD11] Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011. 5

[HDT02] Sherri K Harms, Jitender Deogun, and Tsegaye Tadesse. Discovering sequential association rules with constraints and time lags in multiple sequences. In *International Symposium on Methodologies for Intelligent Systems*, pages 432–441. Springer, 2002. 4

[HK06] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006. 9

[Hoy02] Patrik O Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565. IEEE, 2002. 5

[JJO11] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011. viii, 18, 19

[JLS02] Xiaoming Jin, Yuchang Lu, and Chunyi Shi. Distribution discovery: Local analysis of temporal rules. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 469–480. Springer, 2002. 4

[KL05] Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2):154–177, 2005. viii, 1, 4, 16

[LPLN09] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009. 5

[MBPS09] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009. 4

[MKBS09] Abdullah Mueen, Eamonn Keogh, and Nima Bigdely-Shamlo. Finding time series motifs in disk-resident data. In *2009 Ninth IEEE International Conference on Data Mining*, pages 367–376. IEEE, 2009. 4

[MLG$^+$08] Boris Mailhé, Sylvain Lesage, Rémi Gribonval, Frédéric Bimbot, and Pierre Vandergheynst. Shift-invariant dictionary learning for sparse representations: extending k-svd. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE, 2008. 5, 8

[MSRR13] Navin Madicar, Haemwaan Sivaraks, Sura Rodpongpun, and Chotirat Ann Ratanamahatana. Parameter-free subsequences time series clustering with various-width clusters. In *Knowledge and Smart Technology (KST), 2013*

5th International Conference on, pages 150–155. IEEE, 2013. 4

[MU01] Takaki Mori and Kuniaki Uehara. Extraction of primitive motion and discovery of association rules from motion data. In *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 200–206. IEEE, 2001. 4

[MZ93] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993. 7

[PPC08] Vamsi K Potluru, Sergey M Plis, and Vince D Calhoun. Sparse shift-invariant nmf. In *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*, pages 69–72. IEEE, 2008. 5, 7

[PT94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. 5

[RKLE12] Thanawin Rakthanmanon, Eamonn J Keogh, Stefano Lonardi, and Scott Evans. Mdl-based time series clustering. *Knowledge and information systems*, 33(2):371–399, 2012. 4

[RNR12] Sura Rodpongpun, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. Selective subsequence time series clustering. *Knowledge-Based Systems*, 35:361–368, 2012. 4

[SDL$^+$12] Anthony J Spahr, Michael F Dorman, Leonid M Litvak, Susan Van Wie, Rene H Gifford, Philipos C Loizou, Louise M Loiselle, Tyler Oakes, and Sarah Cook. Development and validation of the azbio sentence lists. *Ear and hearing*, 33(1):112, 2012. 12

[SGM00] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelli-*

gence for Web Search (AAAI 2000), pages 58–64, 2000. 6

[SL06] Evan C Smith and Michael S Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006. 5, 8, 10

[SYCC$^+$15] Mohammad Shokoohi-Yekta, Yanping Chen, Bilson Campana, Bing Hu, Jesin Zakaria, and Eamonn Keogh. Discovery of meaningful rules in time series. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1085–1094. ACM, 2015. 4

[TN12] Leo Taslaman and Björn Nilsson. A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS one*, 7(11):e46331, 2012. 5

[TSD00] Peter Tino, Christian Schittenkopf, and Georg Dorffner. Temporal pattern recognition in noisy non-stationary time series based on quantization into symbolic streams. lessons learned from financial volatility trading. 2000. 4

[Woh14] Brendt Wohlberg. Efficient convolutional sparse coding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177. IEEE, 2014. 5

[Woh16] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, 2016. 5

[WSH06] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006. 12, 13

[ZAT14] Seyedjamal Zolhavarieh, Saeed Aghabozorgi, and Ying Wah Teh. A review of subsequence time series clustering. *The Scientific World Journal*, 2014, 2014. 4

[ZSG90] Victor Zue, Stephanie Seneff, and James Glass. Speech database development at mit: Timit and beyond. *Speech Communication*, 9(4):351–356, 1990. 20