Towards On-line Domain-Independent Big Data Learning: Novel Theories and Applications

Thesis submitted in accordance with the requirements of
the University of Stirling for the degree of Doctor of Philosophy

by

Zeeshan Khawar Malik

Division of Computing Science and Mathematics
School of Natural Sciences
University of Stirling
Scotland, UK

November 2015

ABSTRACT

Feature extraction is an extremely important pre-processing step to pattern recognition, and machine learning problems. This thesis highlights how one can best extract features from the data in an exhaustively online and purely adaptive manner. The solution to this problem is given for both labeled and unlabeled datasets, by presenting a number of novel on-line learning approaches.

Specifically, the differential equation method for solving the generalized eigenvalue problem is used to derive a number of novel machine learning and feature extraction algorithms. The incremental eigen-solution method is used to derive a novel incremental extension of linear discriminant analysis (LDA). Further the proposed incremental version is combined with extreme learning machine (ELM) in which the ELM is used as a preprocessor before learning.

In this first key contribution, the dynamic random expansion characteristic of ELM is combined with the proposed incremental LDA technique, and shown to offer a significant improvement in maximizing the discrimination between points in two different classes, while minimizing the distance within each class, in comparison with other standard state-of-the-art incremental and batch techniques.

In the second contribution, the differential equation method for solving the generalized eigenvalue problem is used to derive a novel state-of-the-art purely incremental version of slow feature analysis (SLA) algorithm, termed the generalized eigenvalue based slow feature analysis (GENEIGSFA) technique. Further the time series expansion of echo state network (ESN) and radial basis functions (EBF) are used as a pre-processor before learning. In addition, the higher order derivatives are used as a smoothing constraint in the output signal. Finally, an online extension of the generalized eigenvalue problem, derived from James Stones criterion, is tested, evaluated and compared with the standard batch version of the slow feature analysis technique, to demonstrate its comparative effectiveness.

In the third contribution, light-weight extensions of the statistical technique known as canonical correlation analysis (CCA) for both twinned and multiple data streams, are derived by using the same existing method of solving the generalized eigenvalue problem. Further the proposed method is enhanced by maximizing the covariance between data streams while simultaneously maximizing the rate of change of variances within each data stream. A recurrent set of connections used by ESN are used as a pre-processor between the inputs and the canonical projections in order to capture shared temporal information in two or more data streams. A solution to the problem of identifying a low dimensional manifold on a high dimensional dataspace is then presented in an incremental and adaptive manner.

Finally, an online locally optimized extension of Laplacian Eigenmaps is derived termed the generalized incremental laplacian eigenmaps technique (GE-

NILE). Apart from exploiting the benefit of the incremental nature of the proposed manifold based dimensionality reduction technique, most of the time the projections produced by this method are shown to produce a better classification accuracy in comparison with standard batch versions of these techniques - on both artificial and real datasets.

Contents

List of Figures

List of Tables

DECLARATION

I understand the nature of plagiarism, and I am aware of the University's policy on this. I certify that this dissertation reports original work by me during my University project. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

15-10-2015

Signature Date

ACKNOWLEDGEMENTS

This thesis would not have been possible without the kind help and support of a large number of individuals. First and foremost, with lots of love and care I would like to thank my family members, especially my beloved parents, my father-in-law, my lovely wife and son, Hashim. Without their continued help, support and guidance, this would never have been possible. Thank you very much.

My whole hearted thanks to my most humble and friendly principal supervisor, Professor Amir Hussain, for his very generous and timely offer of the PhD position in his prestigious COSIPRA Lab. Thank you for your most inspirational and radical ideas in our countless conversations (over umpteen sleepless nights!), and your unparalleled guidance and support for writing and publishing our high-quality research contributions in leading high-impact multi-disciplinary journals - this PhD would definitely not have been possible without you!

I would also like to thank Prof. Amir Hussain for his most kind recommendation to Prof. Q.M. Jonathan Wu, Research Chair and Head of CVSS (Computer Vision and Sensing System) Lab at the University of Windsor, who offered me the position to work as a full time Research Associate in his lab during my Doctoral Studies; and offered me a stipend that covered living expenses of my family

members during my entire PhD. Thank you Prof. Q.M. Jonathan Wu and again to Prof. Amir Hussain for your kind help and support!

I am also thankful to Dr. Glen Campbell, from the University of California for proof-reading my whole thesis and for his kind and useful suggestions that played a key role in improving the quality of my thesis.

I would also like to thank my University (University of The Punjab), my department Director, Prof. Syed Mansoor Sarwar, and Prof. Mujahid Kamran, Vice Chancellor (University of the Punjab) who motivated me and offered financial assistance for my doctoral studies.

Lastly, I would like to thank my colleagues and friends from COSIPRA Lab at the University of Stirling and CVSS Lab in the University of Windsor whom I had the opportunity to discuss areas of mutual interests: Dr. Yimin Yang, Dr. Than Minh Nguyen and Kamran. I also would like to thank Grace, Lynn, Linda, Gemma, Linsey and Louise Bleakley for providing me generous kind support throughout my PhD.

GLOSSARY OF ABBREVIATIONS

LDA	Linear Discriminant Analysis
SFA	Slow Feature Analysis
CCA	Canonical Correlation Analysis
LE	Laplacian Eigenmaps
ELM	Extreme Learning Machine
ESN	Echo State Network
RBF	Radial Basis Function
GEV	Generalized Eigenvalue Problem
ILDA	Incremental Linear Discriminant Analysis
GENEIGSFA	Generalized Eigenvalue Based Slow Feature Analysis
MNIST	Mixed National Institute of Standards and Technology Database
GENILE	Generalized Incremental Laplacian Eigenmaps
ETL	Extract, Transform and Loading
GSVD-ILDA	Generalized Singular Value Decomposition - Incremental Linear Discriminant Analysis
PCA	Principal Component Analysis
CLDA	Complete Linear Discriminant Analysis
CCIPCA	Candidate Covariance-Free Incremental Principal Component Analysis
CCIMCA	Candidate Covariance-Free Incremental Minor Component Analysis
GTM	Generative Topographic Mapping
LLE	Local Linear Embedding
RVFL	Random Vector Functional Link
RNN	Recurrent Neural Network
MSE	Mean Square Error
TP	True Positive
TN	True Negative
FP	False Positive

FN	False Negative
PMF	Probability Mass Function
LASSO	Least Absolute Shrinkage and Selection Operator
GENILDA-ELM	Generalized Eigenvalue Based Linear Discriminant Analysis - Extreme Learning Machine
ESVM	Extreme Support Vector Machine
IPCA	Incremental Principal Component Analysis
HVCCA	High Variance Canonical Correlation Analysis

Chapter 1

Introduction

This chapter Firstly describes the motivation of this research. Secondly, this chapter presents the brief overview of the primary types of learning and processing of feature extraction techniques addressed in this thesis. Thirdly, this chapter describes the aim and objective of this research. Finally the organizational structure of this thesis followed by the list of publications is presented in the end.

1.1 Motivation

Extracting optimal feature sets from online data by transforming it to a reduced space for both clustering and classification is a fundamental and challenging problem in the area of machine learning and data mining[3, 4]. The main objective of this research is to suggest a novel light-weight online feature extraction mechanism which helps in extracting features from the data in an exhaustively incremental and adaptive manner.

Feature extraction is a very useful and popular technique due to the transformation of existing high dimensional features to a lower dimensional space. The word “selection” is used when selecting a subset of features from the existing features without transformation. Features extraction is very commonly used in audio, video and image processing [5], canonical correlation analysis [6], slow feature analysis [2], clustering [7] and classification [8] techniques. Feature extraction works for both supervised [9] and unsupervised methods [2].

In audio and video analysis, feature extraction acts as a foundation for audio and video key frame extraction [10] and classification [11]. It helps in extracting features most effective for classification. Key frame extraction is a very important research area in video information retrieval [12]. The key frames are generally extracted from each shot which leads to video summarization. A shot can be considered as a sequence of frames captured from a single camera operation. Detecting shot in a video sequence is a process of identifying visual discontinuities along the time domain. After segmenting the video into shots, the first and last frames of each shot are chosen as key frames. Further the key frames within each shot are extracted based on various models with a fixed or variable number of key frames [13]. Its online versions automatically segments and classifies audio-visual data from movies or television programs to ensure feasibility of real-time processing [14].

In real-time visual tracking scenarios, non-stationary image streams continuously change over time [15]. In these scenarios, the multiclass object detectors [16] and adaptive methods used for face recognition extract most effective features in a purely online and adaptive manner. This strategy leads to complexity reduction and enhancement of computational efficiency.

Using online feature extraction mechanisms masses of online discussion in the form of opinion and comments about consumer products are intelligently analyzed leading to effective decisions [17]. Similarly user-generated reviews of consumer products in plain text are adaptively mined to extract useful sentiments leading to analyses of the user's behaviour [18].

Further feature extraction has also played a very important role in the extraction of useful information from medical images for diagnostic analysis [19, 20]. Feature extraction helps doctors to diagnose clues and defects in their patients [21]. Medical imaging, especially X-ray based examination and ultrasonography, is critical at all levels of health care [22].

For example in public health and in preventive medicine, decisions are dependent on correct diagnosis. The use of diagnostic imaging service leads to the correct assessment of the underlying disease and suggests the response to the treatment.

Further feature extraction plays a vital role in speech recognition. It is used primarily in two different ways: 1) Specifically temporal analysis [23] and 2) spectral analysis [24]. The speech waveform is used in temporal analysis whereas spectral representation of speech signal is used in spectral analysis. Digital signal processing is the core of the speech recognition system. It helps in converting the speech waveform to some form of parametric information. Further it is used to find relevant information in the form of a small number of parameters or features.

Consequently similar segments can be grouped by comparing their features.

Additionally feature extraction is used for object detection in high dimensional images [25]. Feature extraction and object detection is one of the key research issues addressed by researchers and scientists in the fields of computer vision and digital photogrammetry [26][25]. Feature extraction helps in efficiently representing interesting portions of an image as a compact feature vector. Further it is often used to solve computer vision problems which include object detection [25] recognition [27], content-based image retrieval [28], face detection [29], face recognition [30] and texture classification [26].

1.2 Types of Processing

Feature extraction algorithms are fundamentally divided into two major categories: (1) Batch and (2) Incremental.

1.2.1 Batch

Batch version is used for one time learning of the whole data whereas incremental version is used for online adaptive learning. Batch version is preferred in scenarios when there is no change in the dataset expected. Incremental version is preferred in scenarios where the data is expected to be received in more than one chunk. In incremental learning, data is combined to the already learned information overtime facilitating online learning, since both types have there own advantages and disadvantages the choice of one or the other depends on the nature and type of problem.

Batch algorithms learn all the data at once but sometimes due to space availability are forced to delete learned data in order to make room for new data. Batch algorithms cannot learn new concepts in an adaptable manner. However incremental methods learn each chunk of training example adaptively at arrival without deleting the previously learned data.

1.2.2 Incremental

Incremental learning is applied in situations where input data is expected to arrive in a sequence in a temporal manner. Further it is capable of diagnosing the deficiency of its current theory and making the best choice in revising the theory. And finally it is conclusively proven [31] that incremental methods perform similarly to their equivalent batch-learning implementation while using fewer computational resources.

1.3 Learning

Learning is primarily divided into three broad categories: (1) supervised learning, (2) unsupervised learning and (3) re-reinforcement learning. In supervised learning also known as target-based learning, the target output is already known and the input data is trained on the basis of target output. Reinforcement learning or reward-based learning technique rewards every action which needs to be maximized to reach the actual destination/target. Unsupervised learning is a target-free learning. In unsupervised learning the model is trained without any target output. Supervised and unsupervised learning is the paradigm used throughout this thesis, and is discussed in detail in the subsections below.

1.3.1 Unsupervised Learning

Unsupervised learning techniques are self-organizing in nature and are designed to find optimal solution without any pre-defined target solution. In unsupervised learning, apart from Reinforcement learning, the goal is not to maximize the utility function. Clustering is one of the most popular examples of unsupervised learning [32]. The novel online learning approaches presented in this thesis are based on hebbian learning mechanism, a popular unsupervised feedforward learning technique [33]. Hebbian learning is described as an explanation for the adaptation of neurons in the brain during the learning process.

This biologically inspired paradigm of Hebbian learning is optimally defined as: "*If any two nodes in the network connected with each other fire simultaneously, the weights connecting them will be strengthened.* This concept is often summarized as "*Cells that fire together, wire together*" [34][35]. The implementation of this concept in artificial neural environment sometimes causes the weight grow out of bound which are then controlled by constraints to achieve stability. These constraints includes weight decay [36] and negative feedback techniques [37]. Unsupervised learning is similar to probability density estimation in statistics. The density estimates are used to construct probabilities of occurrence of certain events on the basis of pre-defined conditions. Unsupervised learning can also be defined as "*learning without a teacher*" [38]; is universally associated with the idea of using a collection of observation $(\mathbf{X}_1, ..., \mathbf{X}_n)$ sampled from a distribution $p(\mathbf{X})$ to describe properties of p$(\mathbf{X})$. It is in practice generically referred to as clustering [32], principal component analysis [39], association rule discovery

[40] and multidimensional scaling [41].

1.3.1.1 An Unsupervised Learning paradigm to Extract Invariant Features

One of the principle paradigm of unsupervised learning which is considered as baseline of contribution in this thesis is invariance/slowness. Invariance is one of the four principles of computation used as candidates to explain self-organization in the visual cortex on the one hand and to unsupervised analyze and represent high dimensional data sets in machine learning [42].

1. **Invariance/Slowness:** Learning invariance is one of the major problems in neural systems. Invariance means features related to the data of a particular object change very slowly and have no effect due to the change in shape, position, orientation, size and rotation of the object. The idea is also explained in terms of a signal which may change quickly due to changes in the sensing conditions, such as scale, location, and pose of the object. However there are certain features of the input signal which change very slowly or rarely such as the presence of a feature or an object. The objective of a neural system in learning invariance is therefore the extraction of slowly varying signals from the input.

   In computer vision, this concept is best utilized when trying to extract some meaningful representation of an object. Extracting meaningful representation from the input signal is currently an active area of research in computational neuroscience [2]. It is concerned with the way the brain learns to form a representation of these external causes from raw sensory input.

Another big challenge is to learn these invariances from the input signal in a completely unsupervised manner [43].

Learning invariance helps in identifying an object in the visual system even if the retinal image of an object is transformed considerably by commonly occurring changes in the environment. Further to use learning invariance one must learn the overall structure of an object to identify which object changes the least due to the change in sensing conditions.

1.3.2 Supervised Learning

Supervised learning invokes the idea of a "supervisor" that instructs the learning system on the labels to associate with training examples. In other words supervised learning works on labeled datasets. It entails learning a mapping between a set of input variables $\mathbf{X}$ and an output variable $\mathbf{Y}$ by applying this mapping to predict the outputs for unseen data. In supervised learning, two major tasks are achieved 1) classification [44] and 2) regression [45].

The classification task is to classify the observations in a set of finite labels. The accuracy of classifiers is most often based on the percentage of correct predictions divided by the total number of predictions.

There are at the minimum three popular techniques which are used to evaluate classification accuracy. Firstly to split the training set by using two-thirds for training and the other third for estimating performance. Secondly to divide the training set into mutually exclusive subsets, a form of cross validation technique;

and for each subset the classifier is trained on the union of all the other subsets. And thirdly to leave-one-out validation in which all the test subsets consist of a single instance. The choice among each of the evaluation methods is based on the amount of data available for training.

Secondly regression analysis is a widely used technique for prediction and forecasting, in the case where its use has substantial overlap with the field of machine learning. The focus of regression analysis is to analyze the relationship between a dependent variable and one or more independent variables. In other words, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed. The assessment of regression analysis is based on the statistical significance of the estimated relationship, between the closeness of actual relationship to the estimated relationship. The techniques most commonly used for regression analysis are linear regression [46], ordinary least squares [47] and logistic regression [48].

1.4 Aim and Contribution of Research

The aim of this research is to derive a novel state of the art online approach of existing supervised and unsupervised learning techniques. These techniques include linear discriminant analysis, slow feature analysis, canonical correlation analysis and laplacian eigenmaps.

The contribution to knowledge includes the following work

1. The online version of Linear Discriminant Analysis is linked with extreme

learning machine which is used as a pre-processor. This method is derived by considering the first two layers of ELM for random feature mapping followed by the incremental version. The novel incremental version with ELM is derived to maximize the discrimination among points belonging to two different classes and to minimize the distance between points within each class.

2. Two incremental versions are derived in the area of slow feature analysis. The first is based on the original slow feature analysis criteria, and the second is based on Stone's criterion. Additionally echo state network is used as a pre-processor with both versions. Radial basis functions are also tested as a pre-processor before both the proposed learning approaches.

3. In the area of canonical correlation analysis, the prime objective is to maximize the correlation between dual and multi-data streams. Echo state networks are used for generating time series information, and to derive correlation between temporal data.

4. Incremental laplacian eigenmaps, a locality preserving optimal manifold based learning technique, is one more contribution in the area of visualization in this thesis. The learning method is again derived by solving the generalized eigenvalue problem.

1.5 Structure of Thesis

Chapter 2, provides a review of the existing literature concerning all methods used as the baseline for contribution to knowledge in this thesis. Further it reviews the existing method of solving the generalized eigenproblem which is used as a bridge to derive novel incremental approaches in the area of slow feature analysis, linear discriminant analysis, canonical correlation analysis and laplacian eigenmaps.

Chapter 3, applies the incremental eigen solution method to derive novel state of the art linear discriminant analysis approach abbreviated as GENILDA. This method exploits the random feature expansion characteristic of Extreme Learning Machine (ELM) as a pre-processor between the input and the proposed algorithm. The rationale for using random features is to firstly remove the non-linearity in the data by mapping the input data from low to a high dimensional random feature space. This expansion before using GENILDA successfully resulted in further maximizing the discrimination among points between two different classes and minimizes the distance within each class. The proposed methods are illustrated on both artificial and real datasets [49].

Chapter 4, generalizes the differential equation to implement the biologically inspired slow feature analysis technique abbreviated as (GENEIGSFA) the generalized eigenvalue based slow feature analysis technique. The temporal activations of echo state networks are used as a pre-processor before learning. Radial basis functions are also used for expansion before learning. The effect of higher order derivatives are used as a smoothing constraint. By re-using James Stone Criterion an online generalized eigen value version of invariant feature extraction mechanism is also presented.

All the proposed methods are tested on artificially created datasets, the real MNIST digit dataset and the dataset of written character trajectories [43].

Chapter 5, implements the incremental eigen-solution method on twinned datasets to obtain extensions of the statistical technique known as canonical correlation analysis. A recurrent set of connections of echo state networks are used between the inputs and the canonical projections to capture shared temporal information between two datastreams. The proposed method is further exploited by considering maximizing the covariance between data streams while simultaneously maximizing the rate of change of variance within each data stream. Echo state network is again used as a pre-processor before learning.

Extracting the covariance information from more than two data streams simultaneously are also presented.

The comparative effectiveness of the proposed methods are illustrated on both artificial and real benchmark datasets [50].

Chapter 6, presents the locally optimum solution to the problem of identifying a low dimensional manifold on a high dimensional dataspace. This chapter presents a novel online version of Laplacian Eigenmaps, termed the Generalized Incremental Laplacian Eigenmaps (GENILE) algorithm. Its comparative performance is evaluated using both artificial and real dataset. Preliminary experimental results demonstrate consistent improvement in the classification accuracy of the proposed method compared to the other state-of-the-art techniques.[51]

1.6 Description of Datasets

This section provides the description of datasets used throughout the thesis:-

1.6.1 UCI Machine Learning Repositories [1]

1. **IRIS:** The dataset consist of 3 classes each of 50 instances. The total length of the dataset is 150. Each class refers to a type of iris plant. In all the three classes, one class is linearly separable from the other 2. The three classes are labeled as 1) Iris Setosa, 2) Iris Versicolour and 3) Iris Virginica.

2. **Liver-Disorder:** This dataset consist of 345 instances and 2 classes. Each class refers to 5 variables 1) mcv mean corpuscular volume, 2) alkphos alkaline aminotransferase, 3) sgpt alamine aminotransferase, 4) sgot aspartate aminotransferase, 5) gammagt gamma-glutamyl transpeptidase and 6) drinks number of half-pint equivalent of alcoholic beverages drunk per day.

3. **Vehicle:** This dataset consist of 946 instances having 18 input variables. The number of classes are 4 which includes 1) OPEL, 2) SAAB, 3) BUS and 4) VAN.

4. **Glass:** This dataset consist of 214 instances having 10 input variables. The number of classes are 7 which includes 1) building windows float processed, 2) building windows non float processed, 3) vehicle windows float processed, 4) vehicle windows non float processed, 5) containers, 6) tableware and 7) headlamps.

5. **Wine:** This dataset consist of 178 instances having 13 input variables. The

number of classes are 3 which describes the type of alcohol.

6. **Image Segmentation:** This dataset consist of 2310 instances drawn randomly from a database of 7 outdoor images. The input variables are 19 and total number of classes are 7.

7. **Vowel:** This dataset consist of 528 instances having 10 input variables. The total number of classes are 11 indexed by integers between 0-10.

8. **Sonar:** This dataset consist of 208 instances having 60 input variables. The total number of classes are 3 which includes 1) Sonar, 2) Mines and 3) Rocks.

9. **Banknote Authentication:** The banknote dataset is taken from genuine and forged banknote-like specimens. This dataset comprises five attributes: 1) variance of Wavelet transformed image, 2) skewness of Wavelet transformed image, 4) entropy of image, and 5) class information. The dataset is organized into two classes. The dataset has a total of 1372 instances.

1.6.2 Swiss Roll Dataset

The swiss roll dataset consist of 20,000 points. Each point in the data set is three dimensional. The three dimensional view of a swiss roll dataset is shown in Figure 1.1.

1.6.3 S-Curve Dataset

The s-curve dataset consist of 20,000 points. Each point in the data set is three dimensional. The three dimensional view of a s-curve dataset is shown in Figure

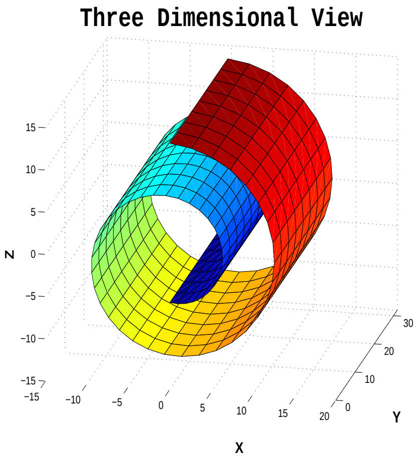


Figure 1.1: Swiss Roll

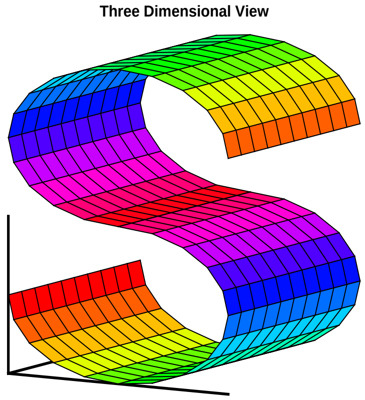


Figure 1.2: S-Curve

1.2.

1.6.4 Yale Dataset

The Yale database contains 165 grayscale images in GIF format of 15 individuals as shown in Figure 3.8. Each subject contains 11 images, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/o glasses, normal, right-light, sad, sleepy, surprised, and wink.

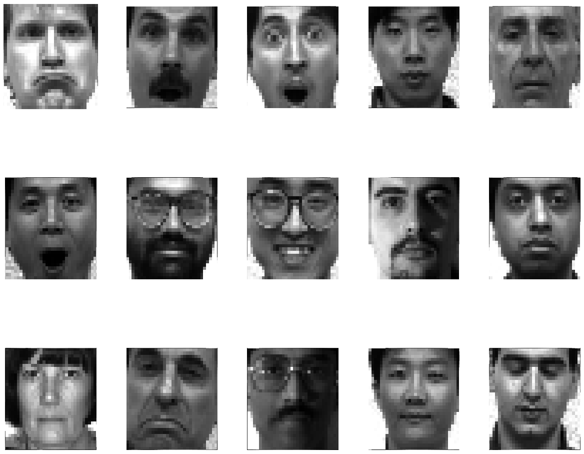


Figure 1.3: Yale Dataset

1.6.5 MNIST Digit Dataset

The MNIST handwritten digit dataset consist of a standardized and freely available set of 70,000 handwritten digits [52]. Each pattern consist of a zero to nine handwritten digit of size 28 x 28 pixels as shown in Figure 6.3. It has a training set of 60,000 examples, and a test set of 10,000 examples.

1.6.6 Character Trajectories

The character trajectories dataset [1] consist of 2858 character samples. The categories of characters are from the letters a to z. Each character can have a different number of pixels which is always between 174 x 205 in length with the standard three dimensional data giving x, y and z coordinates.

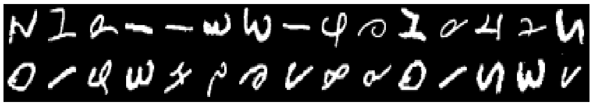


Figure 1.4: Digit Dataset

Figure 1.5: Character Trajectories Dataset

1.6.7 Cardiovascular Disease Dataset

This dataset consist of more than one different type of attributes of the patient labeled as cardiovascular and non-cardiovascular patients. The dataset consist of 558 records and each patient's record has 6 attributes.

1.7 Publications

The following journal papers have resulted from the research presented in this thesis.

1. Malik, Z.K., Hussain, A., and Wu, J., An Online Generalized Eigenvalue Version of Laplacian Eigenmaps for Visual Big Data, *Neurocomputing*, Elsevier (DOI: doi:10.1016/j.neucom.2014.12.119), 2015.

2. Malik, Z.K., Hussain, A., and Wu, J., Extracting online information from Dual and Multi Data Streams, *Neural Computing & Applications*, (In Press), 2015.

3. Malik, Z.K., Hussain, A., and Wu, J., Novel Biologically Inspired Approaches to Extracting Online Information From Temporal Data, *Cognitive Computation*, vol. 6, no. 3, pp. 595-607, 2014.

4. Malik, Z.K., Hussain, A., and Wu, J., A neural implementation of Linear Discriminant Analysis with Extreme Learning Machine, *Neural Network and Learning System, IEEE Transactions on*, (Accepted with Major Revision), 2015.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents the theoretical foundations underlying this thesis. Firstly, it defines some basic concepts used for feature extraction techniques. Secondly, it presents the differential equation used throughout this thesis. Thirdly, it provides a brief overview of the generalized basic framework used for intelligent analysis. The reason for presenting this overview is to highlight the key modules required to perform an intelligent task. Finally, the evaluation and validation techniques used in thesis are described.

2.2 Definition of Some Basic Concepts

2.2.1 Pattern

A *Pattern* is defined as a quantitative or structural description of an object or some other entity of interest [53]. It is usually arranged in the form of combination of feature vector as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ . \\ . \\ \mathbf{x}_n \end{bmatrix}$$

where $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ are the features.

2.2.2 Feature

Feature is defined as an individual characteristic of a phenomenon being observed [53]. Features are most of the time converted to one standard type such as numeric. Features in a pattern can be either discrete numbers or real continuous values, depending on the measurements of an object. The only requirement is that features should reflect the characteristic of desired objects and differ from the characteristics of other objects.

2.2.3 Class

A *Class* consists of a set of patterns that share some common properties. The collection of feature vectors of the same type of objects will naturally form one set. Due to variation among set of patterns in one class, the patterns extracted from the same type of class are seldom identical, however they are closely similar to one another within the same class and dissimilar to the patterns in another class.

The criteria of grouping same set patterns into one class is called cluster analysis [54]. Figure 2.1 shows in mathematical form an example of sample distribution of two classes in a single dimensional space with C classes:

$$\Omega_c \neq \phi, c = 1, ..., C; \Omega_c \bigcap \Omega_l = \phi, c \neq l \epsilon \{1, ..., C\} \tag{2.1}$$

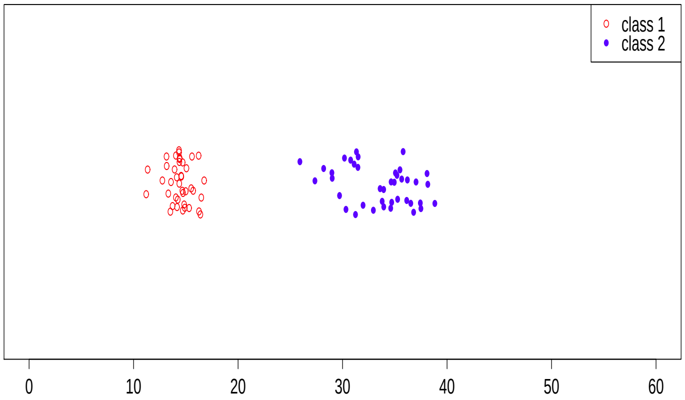


Figure 2.1: Sample demonstration of two classes

2.2.4 Classification Approaches

The most commonly used classification criteria are *distance*, *Bayes decision rule* also called *decision rule*, and *likelihood*:

1. Distance approach is the simplest and most commonly used criterion. In this approach the data is classified according to the class which is closest to it. Euclidean Distance and Mahalanobis distance are the two most commonly used forms.

   Suppose we have **C** classes. Let $(\mu_{\mathbf{j}}, \sum_j)$ be the known parameters of the set of class j, where $\mu_{\mathbf{j}}$ is the reference vector of class j, $\sum_j$ is the covariance. The square form of Euclidean distance of an observation vector $\mathbf{x}$ from class j is:

$$d_j(\mathbf{x}) = \|\mathbf{x} - \mu_{\mathbf{j}}\|^2 \tag{2.2}$$

The square form of Mahalanobis distance of x from class j is:

$$d_j(\mathbf{x}) = (\mathbf{x} - \mu_{\mathbf{j}})^T \textstyle\sum_j^{-1} (\mathbf{x} - \mu_{\mathbf{j}}) \tag{2.3}$$

In fact, Euclidean distance is a special case of Mahalanobis distance.

2. In Bayes decision rule an observation vector will be assigned to the class which has the largest *a posteriori probability* $p(\Omega_l|\mathbf{x})$. For example suppose we have C classes, $\Omega_1, \Omega_2, ..., \Omega_C$ and we know the *a priori* probability of each class $P(\Omega_i), i = 1, 2, ..., C$ and the conditional probability density $p(\mathbf{x}|\Omega_i)$, $i = 1, 2, ...C$, then *a posteriori* probability can be calculated by Bayes rule

$$p(\Omega_l|\mathbf{x}) = \frac{p(\mathbf{x}|\Omega_l)P(\Omega_l)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\Omega_l)P(\Omega_l)}{\sum_{i=1}^{J} p(\mathbf{x}|\Omega_i)P(\Omega_i)} = \frac{\text{likelihood x prior}}{\text{evidence}} \tag{2.4}$$

3. Likelihood approach is a special case of Bayes classification approach. In case the parameters of a class are known, likelihood will be the probability

density function (PDF). In this case, it is assumed that all of the *a priori* probabilities $P(\Omega_i)$ are equal and the distribution of classes are normal: $\mathbf{x} \sim N(\mu_i, \sum_i)$, i=1,2,...,K. Then we have

$$p(\Omega_j|\mathbf{x}) = p(\mathbf{x}|\Omega_i) \tag{2.5}$$

and

$$p(x|\Omega_i) = \frac{1}{|2\pi \sum_i |^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \sum_i^{-1}(\mathbf{x}-\mu_i)} \tag{2.6}$$

Logarithm of likelihood is usually further taken to make the calculation simpler which is given as follows:

$$P_i(\mathbf{x}) = -\frac{1}{2} ln|\sum_i| - \frac{n}{2} ln 2\pi - \frac{1}{2}(\mathbf{x}-\mu_i)^T \sum_i^{-1}(\mathbf{x}-\mu_i) \tag{2.7}$$

Based on the classification approach used in the input-output (Discriminant) function, the classifiers can be grouped into *Bayesian classifier*, *Likelihood classifier* and *distance classifier*. Figure 2.2 shows input-output functions (Discriminant Function) of these classifiers in a two-class problem.

2.2.5 Correlation

Correlation is a statistical concept of measuring the relationship between two random variables, or two sets of data. It shows how strongly a pair of variables are related.

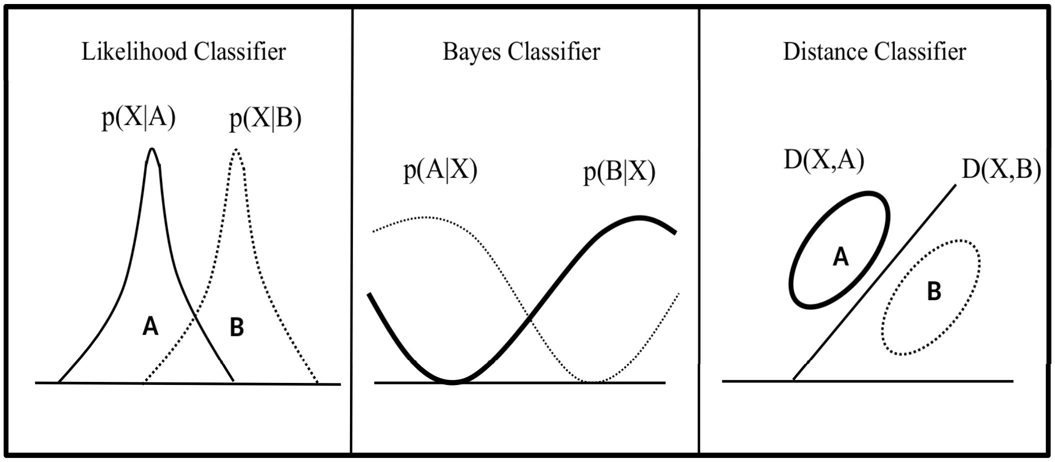


Figure 2.2: Discriminant function of Likelihood, Bayesian and Distance classifiers

A simple example population correlation coefficient P_{AB} between two variables A and B with expected values μ_A and μ_B and standard deviation γ_A and γ_B is defined as:

$$P_{AB} = \mathrm{corr}(A, B) - \frac{\mathrm{cov}(A, B)}{\gamma_A \gamma_B} = \frac{E[(A - \mu_A)(B - \mu_B)]}{\gamma_A \gamma_B} \qquad (2.8)$$

where E is the long-run average value (Expected value) operator, *cov* is a measure of how much two random variables change together and *corr* is the notation used for correlation coefficient.

2.2.6 Covariance

Covariance is another statistical measure which evaluates how much two random variables changes together. It provides a measure of the strength of the correlation between two or more sets of random variables. The covariance of two random variables A and B, each with sample size N is defined by the expectation value:

$$cov(A,B) = \sum_{i=1}^{N} \frac{(a_i - \bar{a})(b_i - \bar{b})}{N} \tag{2.9}$$

The covariance will be zero if the variables are not correlated with each other and nonzero if they are somewhat correlated with each other.

2.3 Differential Equation

In this thesis a differential equation is used for solving a generalized eigenvalue problem whose stable points are the eigenvectors corresponding to the maximum eigenvalue. Zhang and Leung [55] show that one method for finding the maximum eigenvalue of the generalized eigenproblem is:

$$\mathbf{A}\mathbf{w} = \lambda \mathbf{B}\mathbf{w}, \tag{2.10}$$

is to iteratively use

$$\begin{aligned} \Delta\mathbf{w} &= \mathbf{A}\mathbf{w} - f(\mathbf{w})\mathbf{B}\mathbf{w} \qquad (2.11) \\ \mathbf{w} &= \mathbf{w} + \eta\Delta\mathbf{w}, \end{aligned}$$

where η is a learning rate or step size. In (2.11) the first term on the right-hand side is considered as a standard Hebbian rule term [56], and the second term acts to bound the length of the vector $\mathbf{w}$. In (2.11) $f(\mathbf{w}) = \mathbf{w}^t\mathbf{w}$ becomes the continuous version of Oja's algorithm as mentioned by Zhang and Bao in [57].

The function $f(\mathbf{w}) : R^n - \{0\} \rightarrow R$ satisfies

1. $f(\mathbf{w})$ is locally Lipschitz continuous

2. $\exists M_1 > M_2 > 0 : f(\mathbf{w}) > \lambda_1, \forall \ \mathbf{w} : \parallel \mathbf{w} \parallel \geq M_1$ and $f(\mathbf{w}) < \lambda_n, \forall \ \mathbf{w} : 0 < \parallel \mathbf{w} \parallel \leq M_2$

3. $\forall \ \mathbf{w} \in R^n - \{0\}, \exists \ N_1 > N_2 > 0 : f(\theta\mathbf{w}) > \lambda_1, \forall \ \theta : \theta \geq N_1$ and $f(\theta\mathbf{w}) < \lambda_n, \forall \ \theta : 0 \leq \theta \leq N_2$ and $f(\theta\mathbf{w})$ is a strictly monotonically increasing function of θ in $[N_1, N_2]$.

where λ_1 is the greatest generalized eigenvalue and λ_n is the least eigenvalue.

These criteria imply that

1. The function is rather smooth.

2. It is always possible to find values of $\mathbf{w}_i, i = 1, 2$ large enough so that the functions of the weights exceed the greatest eigenvalue.

3. It is always possible to find values of $\mathbf{w}_i, i = 1, 2$ small enough so that the functions of the weights are smaller than the least eigenvalue.

4. For any particular value of $\mathbf{w}_i, i = 1, 2$, it is possible to multiply $\mathbf{w}_i, i = 1, 2$ by a scalar and apply the function to the result to get a value greater than the greatest eigenvalue.

5. Similarly, we can find another scalar so that, multiplying the $\mathbf{w}_i, i = 1, 2$, by this scalar and taking the function of the result gives us a value less than the smallest eigenvalue.

6. The function of this product is monotonically increasing between the scalars defined in 4 and 5.

Using the above criteria by solving the generalized eigenvalue problem has already been proved in [57] given in the appendix of this thesis which can easily be used for learning the highest principal component and for learning the least minor component of the input dataset.

2.4 Role of a Feature Extractor in the Generalized Machine Learning Model

In the following section the broader picture of a machine learning model is described. This picture clearly highlights the connections of other stages involved in completing an intelligent task and shows the prospective stage of feature extractor in the generalized machine learning model.

Figure 2.3 shows the generalized architecture of a machine learning model. In Figure 2.4 the same generalized architecture shown in Figure 2.3 is optimized by representing each layer with a single generalized name which clearly shows the main steps involved for intelligent data analysis. The following sections of this chapter presents a brief description of each phase of the generalized machine learning model including state of the art review of feature extraction techniques which are considered as a baseline for contribution in this thesis.

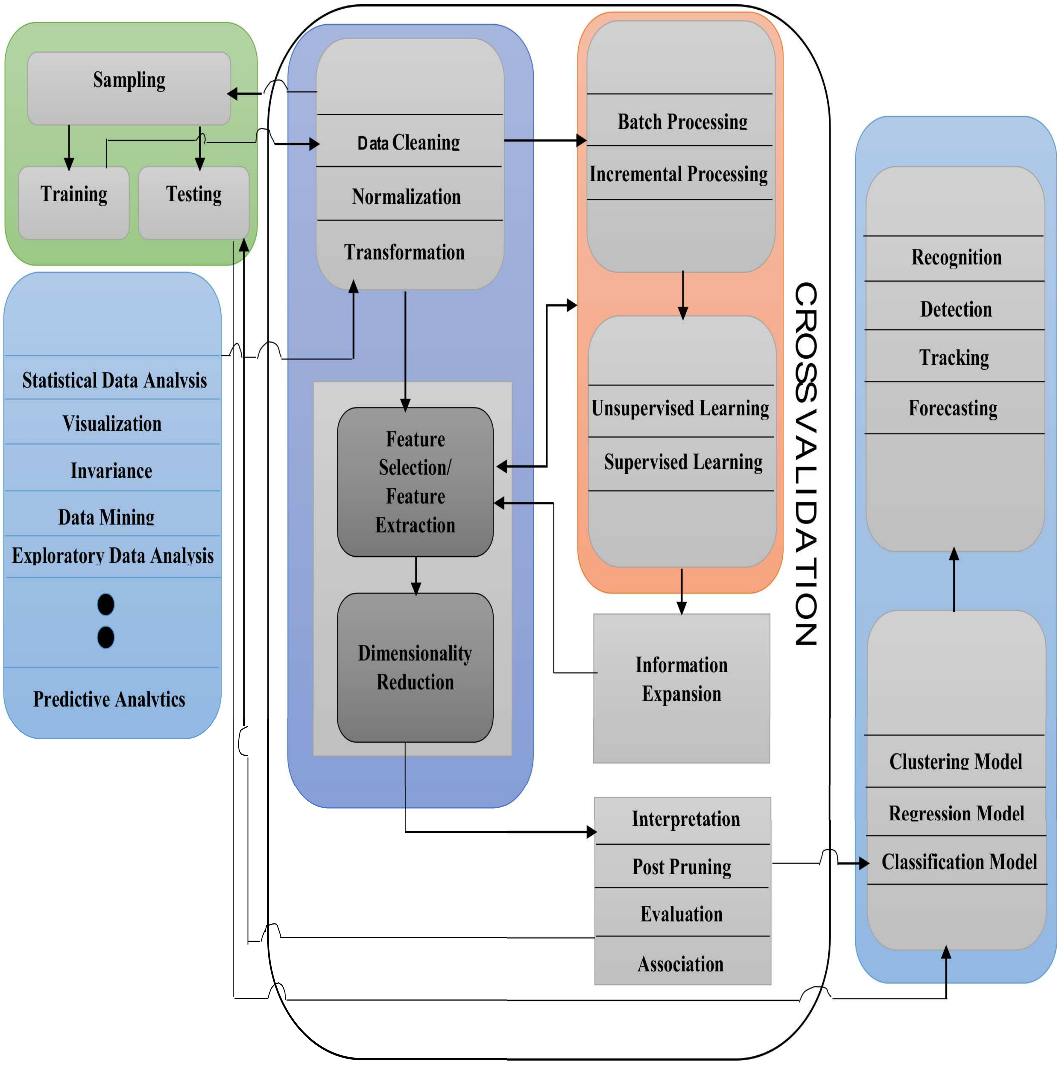


Figure 2.3: Generalized Flowchart of a Machine Learning Model

2.4.1 Data Science

Data science is a set of fundamental principles that support and guide the principled extraction of information and knowledge from data [58]. It is applied by employing techniques based on mathematics, statistics and information technology. This science is gradually becoming very popular in both industry and academia [58]. The main reason for its popularity is due to the rapid increase in the collection of data. Due to this increase, there is a need to mine useful

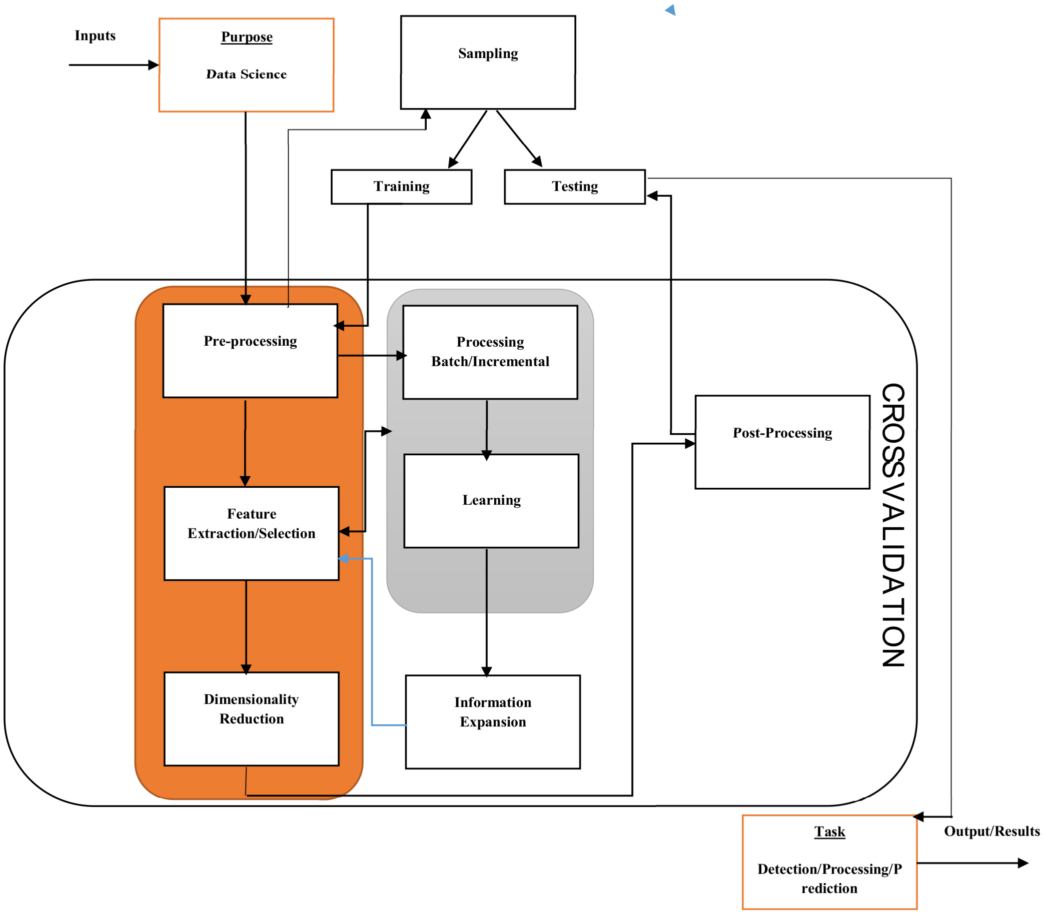


Figure 2.4: An optimized flowchart of a machine learning model

information from this data to assist industrial decision making.

2.4.1.1 Scalability of Massive Data

Due to the rapid growth of data science, the first and foremost issue is scalability or the capacity to scale expanding inputs rapidly. This capacity is referred as **Big Data** platform. This platform introduced a new trend of collecting and storing raw information including ETL (Extract, Transform and Loading) process [59], designing and developing easy, interpretable and adaptable analytics over **Big Data** repositories into order to derive intelligence and extract useful knowledge from them [60].

2.4.1.2 Filtering uncorrelated Data

With the enormous increase in size of **Big Data** repositories, there is a high possibility dealing with large amount of data that are uncorrelated to the kind of analytics to be designed. Hence filtering out uncorrelated data by extracting highly correlated data is another important challenge which significantly affects the quality of final analytics to be designed [60].

2.4.1.3 Sequentialization of Massive Data

Another aspect of **Big Data** is the sequentialization of massive data. The traditionally assumed way is the so-called batch, where all the data needed is available all at once. There are scenarios in the literature when the data is produced/delivered in a sequential/incremental manner. In case of processing massive datasets which follows the same pattern of coming in a sequence chunk by chunk; and therefore incrementalization of state of the art machine learning algorithm is useful for both streaming data and massive data that is too large to be loaded into memory all at once [61].

2.4.1.4 Dimensionality Reduction and Feature Extraction

Learning massive data with high dimensions increases the computational complexity of analytical algorithms. Dimensionality reduction plays a key role in solving the curse of dimensionality and providing meaningful summaries revealing the patterns underlying the data. It is useful to employ sequential learning algorithms for dimensionality reduction and feature extraction of massive dataset with such procedures one can achieve orthogonality in the input space, eliminate redundant and noise variables, perform learning in a lower dimensional and orthogonal input space and reduce variance in the estimator [61].

2.4.1.5 Strong Unstructured Nature of Data Sources

To design meaningful analytics, it is mandatory that input data be pre-processed to a suitable, structured format, and stored in the DFS (Distributed File System). Transformation from a unstructured to a structured format should be performed, according to a sort of goal-oriented methodology [61]. Prekopcsak et al [62] suggest that 80 % of the work consists of preprocessing and only 20 % of the modeling and evaluation.

2.4.2 Preprocessing Data

Pre-processing data is defined as one of the initial steps before using the data. In this phase the data is converted to a form qualitatively used for analysis. This phase includes data cleaning, normalization and transformation known as feature engineering.

1. *Data Cleaning:* This phase includes removing or fixing missing data, detecting outliers in the data and recording values such as mean, standard deviation and range. In order to extract useful features from the data, it is mandatory to clean the data by following the garbage-in-garbage-out principle [63]. For example some data do not address the problem or are incomplete.

2. *Normalization:* Scaling the features to a common range such as between 0 and 1 increases the level of standardization in the data. It is difficult

for a learning algorithm to learn data which is continuous in nature and in which each feature is measured in a different scale and has a different range of possible values [64]. The two most common methods for data normalization are

(a) min-max normalization:

$$\mathbf{w}^{'} = \frac{\mathbf{w} - \min_A}{\max_A - \min_A}(\text{newmax}_A - \text{newmin}_A) + \text{newmin}_A \qquad (2.12)$$

(b) z-score normalization:

$$\mathbf{w}^{'} = \frac{\mathbf{w} - \text{mean}}{\text{standdev}_A} \qquad (2.13)$$

where $\mathbf{w}$ is the old feature value and $\mathbf{w}^{'}$ is the new one.

3. *Transformation:* This final step is to transform the process data. It represents the decomposition and aggregation of features. Decomposition is important for features that represent a complex concept which may be more useful to a learning algorithm when split into its constituent parts. On the other hand, aggregating features into a single feature could lead to a more effective analysis. According to S.B. Kotsiantis [65] feature transformation provides a better discriminative ability than the best subset of given features.

2.4.3 Batch Learning

This procedure is also referred to as batch learning or non-sequential learning [66]. Suppose a learning system is specified by a parameter vector θ =

$(\theta^1, ..., \theta^m)^T \ \epsilon \ \mathbf{R}^m$. Let (x, y) be a discriminant pair, which the system learns, where $\mathbf{x} = (x_1, ..., x_r)^T \ \epsilon \ \mathbf{R}^r$ and $\mathbf{y} = (y_1, ..., y_s)^T \ \epsilon \ \mathbf{R}^s$. For each input-output (discriminant) pair, the loss function is defined as

$$d(\mathbf{x}, \mathbf{y}; \theta) \tag{2.14}$$

which evaluate the performance of learning system θ for given input $\mathbf{x}$ and desired output $\mathbf{y}$. A finite number of input-output examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}$ where $i = \{1, 2, ..., n\}$ are available and the goal is to obtain optimal learning θ_*.

2.4.4 Concept Drifts

Concept drifts represent scenarios in which the relation between the input data and the target variable changes over time [67].

Formally concept drift between time stamp t_1 and time stamp t_2 can be defined as

$$\exists \mathbf{X} : p_{t_1}(\mathbf{X}, \mathbf{y}) \neq p_{t_2}(\mathbf{X}, \mathbf{y}), \tag{2.15}$$

where p_{t_1} and p_{t_2} denotes the joint distribution at time t_1 and t_2 between the set of input variables $\mathbf{X}$ and the target variable $\mathbf{y}$. Concept drifts can be categorized into three types

1. **Real Concept Drift:** This drift refers to changes in the target variable without change in the input [68].

2. **Virtual Concept Drift:** This drift refers to incoming data change without

change in the target variable [69].

3. **Dual Concept Drift:** This drift refers to simultaneous change of incoming data and target variable [68].

These drifts can be learned primarily in one of two ways 1) Domain Independent Learning and 2) Domain Dependent Learning.

2.4.4.1 Domain Independent Learning

Domain independent learning algorithms learn every new chunk, or data instance, independently without depending on the previously processed information. Figure 2.5 shows a generalized architecture of domain independent learning. It takes less memory because no previous processed information is required for learning new adaptive inputs [67]. Further these algorithms calculate the predicted output from a new class or an existing class, similar to the incremental subspace versions of PCA and LDA methods [70][71].

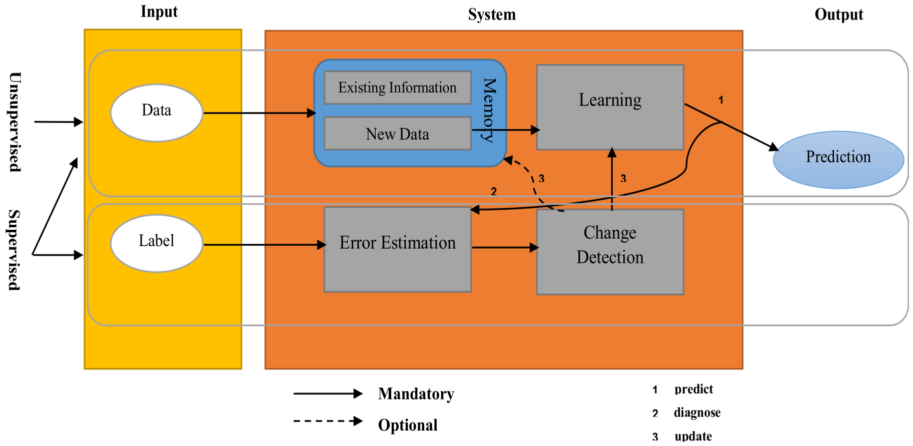


Figure 2.5: Generalized Architecture of Domain Independent Learning

2.4.4.2 Domain Dependent Learning

Domain dependent learning algorithms learn every new incoming chunk, or data instance, by using the previously processed information. Figure 2.6 shows a generalized architecture of domain dependent learning. These types of algorithms calculate the predicted output by using the existing adjacent information of the previous chunk, for example the most recently proposed incremental version of Laplacian Eigenmap [72] is very dependent on previously processed information.

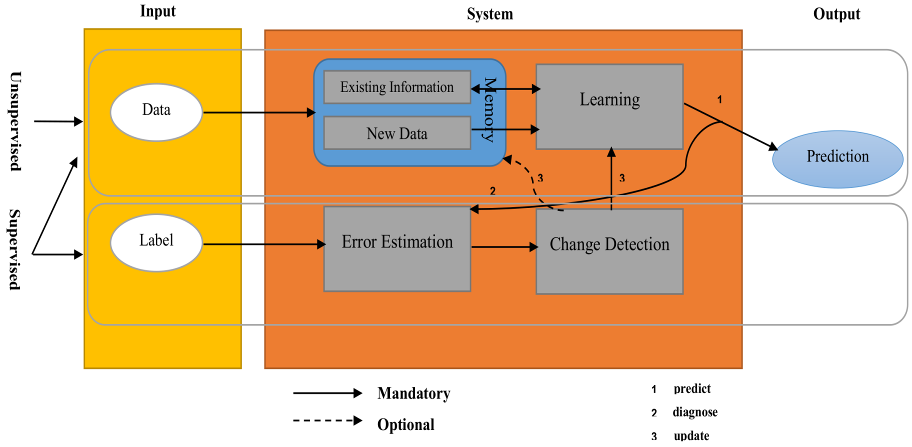


Figure 2.6: Generalized Architecture of Domain Dependent Learning

2.4.5 Types of Feature Extractors

The main aim of a feature extractor is to select or combine those features that preserve most of the information and to remove redundant components thereby improving the performance of subsequent classifiers. Feature reduction is primarily divided into two main categories: feature selection [73, 74] and feature extraction [75].

2.4.5.1 Feature Selection Techniques

Feature selection is defined as a process of detecting relevant features and discarding irrelevant ones [73]. It can also be defined as a process of selecting a subset of features from the original set of features [74]. A general definition of feature selection in [76] is:

Definition(Feature Selection): *Let $J(A^{'})$ be an evaluated measure to be optimized (say to maximize) defined as $J : A^{'} \subseteq A- > \Re$. The selection of a feature subset can be seen under three considerations:*

1. *Set $|A^{'}| = m < n$. Find $A^{'} \subset A$, such that $J(A^{'})$ is maximum*
2. *Set a value J_o this is, the minimum J that is going to be tolerated. Find the $A^{'} \subseteq A$ with smaller $|A^{'}|$, such that $J(A^{'}) \geq J_o$.*
3. *Find a compromise among minimizing $|A^{'}|$ and maximizing $J(A^{'})$(general case).*

From classification perspective feature selection techniques are primarily divided into three main categories [73].

1. Filter Feature Selection Techniques
2. Wrappers Feature Selection Techniques
3. Embedded Feature Selection Techniques

2.4.5.1.1 Filter Feature Selection Techniques Filter feature selection techniques calculate each feature's relevance separately and remove low scoring features. It can be considered as a feature scoring technique which ranks each feature in accordance with its relevance to the objective while removing low scoring ones. These methods are computationally very fast, simple and can easily be scaled to very high dimensional datasets [77]. The only drawback of these approaches is their independence with the classification algorithm. This means that these feature selection algorithm filters the subset of features without interacting with the classifiers leading to increased chances of raising the classification error compared to other feature selection approaches [78].

2.4.5.1.2 Wrapper Feature Selection Techniques Wrapper based feature selection techniques optimizes a predictor as a part of the selection process. Using the induction algorithm [79] the wrapper method uses various search techniques to select a subset of features. The search techniques popularly used in a wrapper based feature selection process is 1) Forward Selection and 2) Backward Selection.

1. **Forward Selection:** It starts with an empty set of features and greedily add features one at a time. The features at each step is added which produces the larger increase of the evaluation function with respect to the value of the current set [80].

2. **Backward Selection:** It starts with a set of features that contains all the features and discards features one at a time. The feature at each step is removed whose removal results in the larger increase in the evaluation function [80].

2.4.5.1.3 Embedded Feature Selection Techniques Embedded feature selection techniques perform feature selection in the process of training and are usually specific to given learning machines. These methods are similar to wrapper method. The only advantage of using embedded methods in comparison to wrapper methods is their interaction with the classification model during feature subset selection and thus are far computationally less expensive than wrapper methods [73]. In other words these methods like wrapper methods are defined as methods in which feature selection is performed automatically by the learning algorithms [80].

2.4.5.2 Feature Extraction

In machine learning and statistics, dimensionality reduction is a process of reducing the dimensions of data by mapping a set of high dimensional input points onto a low dimensional latent space in two different ways: (1) By only keeping the most relevant variables from the original dataset (this technique is called Feature Selection) or by exploiting the redundancy of the input data and (2) By finding a smaller set of new variables, each being a combination of the input variables, containing basically the same information as the input variables (this technique is called Feature Extraction). Feature extraction is exploratory and has applications in dimension reduction [81], automatic exploratory data analysis [82], data visualization [83] and knowledge discovery [84]. This section presents a complete theoretical foundation of feature extraction techniques which are considered as a baseline for contribution in this thesis.

2.4.5.2.1 Linear Discriminant Analysis Fisher's Linear Discriminant Analysis is one of the most popular supervised feature extraction technique used primarily for classification problems [85]. The prime objective of linear discriminant analysis is to find a single subspace for two or more sub classes of the data. The high dimensional data is projected on this subspace and the distance between data points within each class is reduced whereas the distance between data points in two or more different classes is maximized. In other words it is a method which finds a series of projections which maximizes the ratio of between class and within class variance. The projections of two outcomes and the decision region of Linear Discriminant Analysis (LDA) is shown in Figure 2.7.

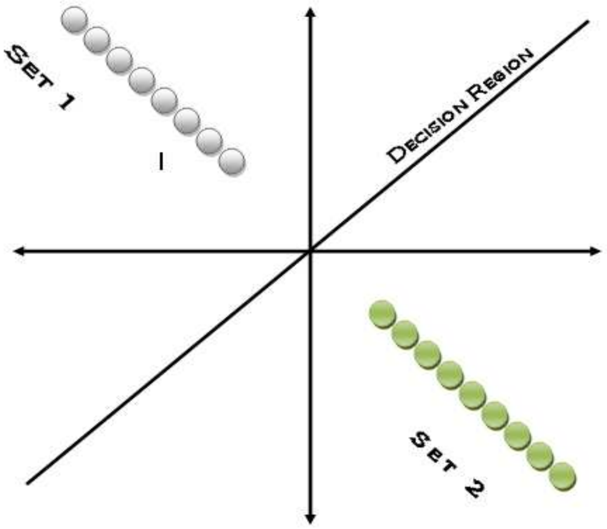


Figure 2.7: Visualization of two outcomes

Suppose $\mathbf{K}_1 = \{k_1^1, ..., k_{l_1}^1\}$ and $\mathbf{K}_2 = \{k_1^2, ..., k_{l_2}^2\}$ be the samples from two different classes belonging to the same data $\mathbf{K}$. Linear Discriminant Analysis is given by the vector $\mathbf{w}$ which maximizes

$$\mathbf{J}(\mathbf{w}) = \frac{\mathbf{w}^T S_{\mathbf{B}} \mathbf{w}}{\mathbf{w}^T S_{\mathbf{w}} \mathbf{w}} \tag{2.16}$$

where

$$S_{\mathbf{B}} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{2.17}$$

$$\text{and } S_{\mathbf{W}} = \sum_{i=1,2} \sum_{k \epsilon K_i} (\mathbf{k} - \mathbf{m}_i)(\mathbf{k} - \mathbf{m}_i)^T \tag{2.18}$$

are the between and within class scatter matrices respectively and $\mathbf{m}_i$ is defined as $\mathbf{m}_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \mathbf{k}_j^i$. The rationale behind maximizing $J(\mathbf{w})$ is to find a direction which maximizes the projected class means (the numerator) and simultaneously minimizes the classes variance in the same direction (the denominator).

Figure 2.8 shows the distribution of iris data by the first single dimension using standard Fisher's Linear Discriminant Analysis technique.

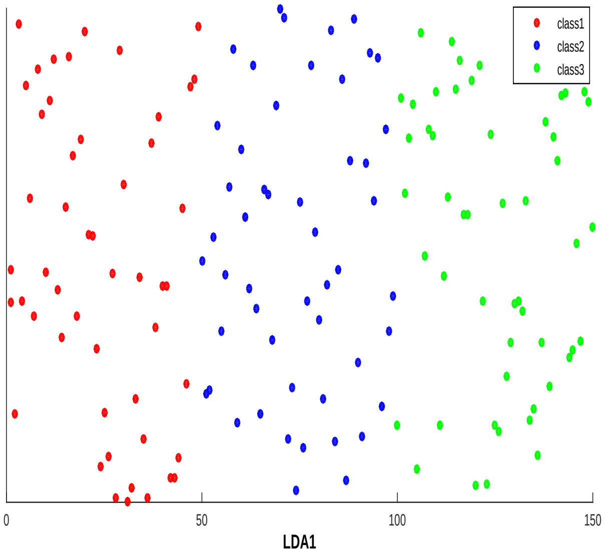


Figure 2.8: Single Dimensional Projection of IRIS Dataset using LDA

2.4.5.2.1.1 Related State of The Art In [86], the authors presented an incremental least square solution to linear discriminant analysis (LDA) by proposing its online incremental version. This approach dynamically updates the least square solution (minimizes the sum of squares of the errors best for data fitting) to LDA by calculating the pseudo-inverse of the centered data matrix, and the indicator matrix without eigen-analysis. This strategy makes the incremental updation mechanism simple. The only drawback of this method is its high computational complexity, since every new incoming instance requires updation of least square solution matrix and other intermediate matrices including centered matrix, mean matrix, indicator matrix and total scatter matrix. In [87] the authors have proposed a novel CCA-based incremental linear discriminant analysis method for action recognition. This procedure iteratively learns the multi-linear discriminant subspace using canonical correlation analysis. It performs incremental updation of the discriminant transformation matrix and maximizes the canonical correlations of the intra-class data samples while simultaneously minimizes the canonical correlations of the inter-class data samples. In [88] the authors used the concept of spanning set approximation for each new incoming data point to approximate all of the between-class, total and within class scatter matrices. The proposed method is computationally very expensive as it requires the updation of three matrices for each new point. Another incremental approach to linear discriminant analysis (LDA) [89] proposed incremental LDA deriving discriminant eigen-space in a streaming environment without updating the eigen-decomposition. By including a new data point, the means and the scatter matrices need to be recalculated. As a result, this method has also a computationally expensive criteria but the eigen decomposition has no update criteria. Infact, update is only required for mean and scatter matrices. The updation criteria for mean, within and between class matrices are presented for both sequential computation (one data point at a time), and for information coming in more than one chunks.

In [90] the authors have proposed an incremental supervised learning method called Generalized Singular Value Decomposition-Incremental Linear Discriminant Analysis (GSVD-ILDA) for adaptively learning face images. The proposed GSVD-LDA can incrementally learn an adaptive subspace instead of recomputing the LDA/GSVD again, efficiently reducing the computational cost.

The advantage of the proposed algorithm includes the processing of samples in chunks or in a sequence desired for large image datasets. Secondly by dynamically adding samples, the algorithm can lesser the computational cost. The only drawback of this method is that more than one updations are required including the updation of global mean, rank approximation of the left singular vectors, the corresponding singular values and projection matrix on each adaptive input. Similarly in [91] the authors resolved the scalability problem of complete linear discriminant analysis [92] technique. Which is a PCA plus LDA algorithm, by first presenting a new implementation of complete linear discriminant analysis (CLDA) in which two steps of QR decomposition, rather than singular value decomposition are used; to obtain the orthonormal bases of the range and null spaces of with-in scatter matrix followed by presenting its incremental version which efficiently perform QR decomposition adaptively on each new incoming chunk without recomputing the CLDA again. In [93] the authors proposed a fast incremental version of linear discriminant analysis including the computing and updating the QR factorization of the data matrix coming in both chunk by chunk and point by point manner. The only problem in this fast ILDA is the lack of incorporation of regularization approach to avoid over sampled problems.

Most of the incremental versions of linear discriminant analysis proposed in the past are domain dependent.

2.4.5.3 Invariance

How can we consistently recognize an object if it continuously changes in angle, eye position, distance, size and orientation? Learning invariance has always been a very important and interesting area of research. There are many contributions in this area of research and researchers in the past have derived novel generalized learning approaches for extracting invariant features from the data.

2.4.5.3.1 Slow Feature Analysis Slow feature analysis (SFA) is a technique which extracts slowly varying features by responding to input streams based on the assumptions that these features change slowly over time. In other words slow feature analysis is a technique which extracts slowly varying features which changes the least in the input dataset. For example if an object is placed in three different directions in a way that the first direction is quite different from the other two, it should still be recognized as the same object. Thus if images related to three different directions of the object are read by a slow feature technique, it will extract the least changeable features from all the three images. Slow features are not affected by the change in shape, size and position of the object. This means no matter how the object changes in size, shape and position, the slow feature analysis technique can still identify the object.

The slow feature analysis technique is proposed by L. Wiskott et al [94]. If we have a multidimensional input signal which are of non-linear nature $\mathbf{X}_t = [\mathbf{X}_1(t), ..., \mathbf{X}_N(t)]^T$ where t indicates time, we have to find a set of J real-valued instantaneous functions $g_j(X)$ such that the output signal $(\mathbf{Y}_j)_t := g_j(\mathbf{X}_t)$ minimize

$$\Delta(\mathbf{Y}_j) = \langle \dot{\mathbf{Y}}_t^2 \rangle, \tag{2.19}$$

under the constraint

$$\begin{aligned} \langle \mathbf{Y}_j \rangle &= 0 \quad \texttt{(zero mean)} & (2.20) \\ \langle \mathbf{Y}_j^2 \rangle_t &= 1 \quad \texttt{(unit variance)} & (2.21) \\ \forall i < j : \langle \mathbf{Y}_i \mathbf{Y}_j \rangle_t &= 0 \quad \texttt{(decorrelation and order)} & (2.22) \end{aligned}$$

where $\langle . \rangle_t$ and $\dot{\mathbf{Y}}$ indicates the temporal averaging and the derivative of $\mathbf{Y}$, respectively. These are not compulsory but if they are met, the resulting features would definitely be slowly varying. The diagrammatic view of decorrelated slow features extracted from non-linear input data is shown in Figure 2.9.

1. **Description of SFA** The extraction of slow features in batch mode is simplified by eigenvectors. If g_j are a linear combination of finite set of non-linear function F, then

$$\mathbf{y}_j(t) = g_j(\mathbf{x}(t)) = \mathbf{w}_j^T \mathbf{F}(\mathbf{x}(t)) = \mathbf{w}_j^T \mathbf{z}(t), \tag{2.23}$$

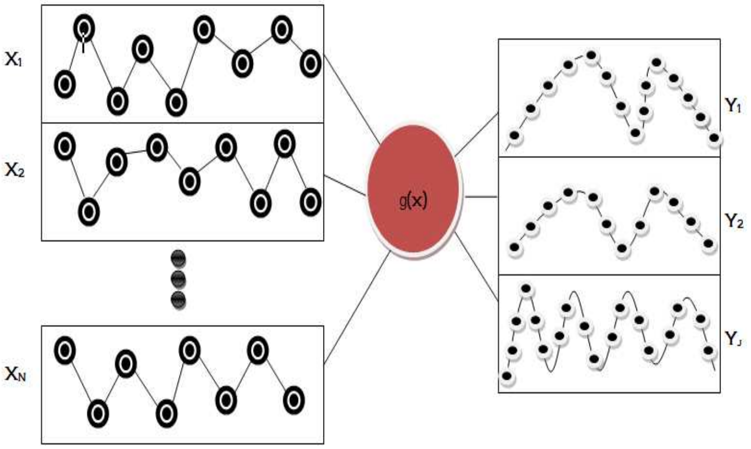


Figure 2.9: Slow Feature Analysis

so SFA will be to find the weight vector $\mathbf{w}$ by minimizing the rate of change of output variable.

$$\Delta(\mathbf{y}_j) = \langle \dot{\mathbf{y}}_j^2 \rangle = \mathbf{w}_j^T \langle \dot{\mathbf{z}} \dot{\mathbf{z}}^T \rangle \mathbf{w}_j \tag{2.24}$$

with the constraint as mentioned in equation (2.20), (2.21) and (2.22). The function F should be calculated in such a way that the covariance matrix should have zero mean and unit variance. The only condition which can be left is the orthonormality of the weight vector $\mathbf{w}$.

Wiskott and Sejnowski [94] developed a method of finding filters of the data which captures the most slowly changing features of the data set. They argued that these features can be captured by a filter which reduces the magnitude of the rate of change of data. They proposed the following: Let $\mathbf{x}$ be an image vector. The filter should try to reduce $E((\frac{d\mathbf{x}}{dt})^2)$ where

$E(.)$ denotes the temporal expectation and the derivatives are with respect to time.

However this alone is not enough: a filter which outputs a constant value would have $E(\frac{d\mathbf{x}}{dt}) = 0$ but be totally uninformative. Thus they argue what is needed is a filter , $\mathbf{w}$, which minimized this outcome but satisfied the conditions

$$\begin{aligned} E(\mathbf{w}^T\mathbf{x}) &= 0 \\ E(\mathbf{w}^T\mathbf{x}\mathbf{x}^T\mathbf{w}) &= 1 \end{aligned}$$

with subsequent filters, if any, being orthonormal.

This is shown to be equivalent to the minimum of the generalized eigenvalue-eigenvector pair

$$E\left((\frac{d\mathbf{x}}{dt})(\frac{d\mathbf{x}}{dt}^T)\right)\mathbf{w} = \lambda E(\mathbf{x}\mathbf{x}^T)\mathbf{w} \tag{2.25}$$

If we have zero mean data, we may consider the equation above as functions of the covariance matrices. The standard method [94] for solving (2.25) is to whiten the data so that $E(\mathbf{x}\mathbf{x}^T) = I$ and then to perform a principal component analysis on the derivatives which form the left hand side of (2.25).

Batch Principal Component Analysis [95] techniques are used for Batch Slow Feature Analysis (BSFA). As in (2.24) the function F is appropriately selected by a well-known process of whitening (or sphering) in which the data $\mathbf{x}$ is mapped to $\mathbf{z}$ with a zero mean and unit variance along each principal component (PC) direction and all the principal component (PC) are totally decorrelated from each other. The principal components with the smallest eigenvalues which minimize the rate of change of output are considered to be the slowest feature of the whitened covariance matrix.

Figure 2.10 shows the slowest varying feature extracted by Batch SFA [94] technique on 20 dimensional input vectors from 100 samples from an artificially created signal given as:

$$\mathbf{u}(t) = sin(\frac{t}{33}) + cos(\frac{t}{10} + \mu) \tag{2.26}$$

where t = 1,.....,2000 and $\mu = 3$ is an arbitrarily chosen phase term. The top sub figure in Figure 2.10 shows the original shape of the signal whereas the bottom sub figure shows the output of SFA.

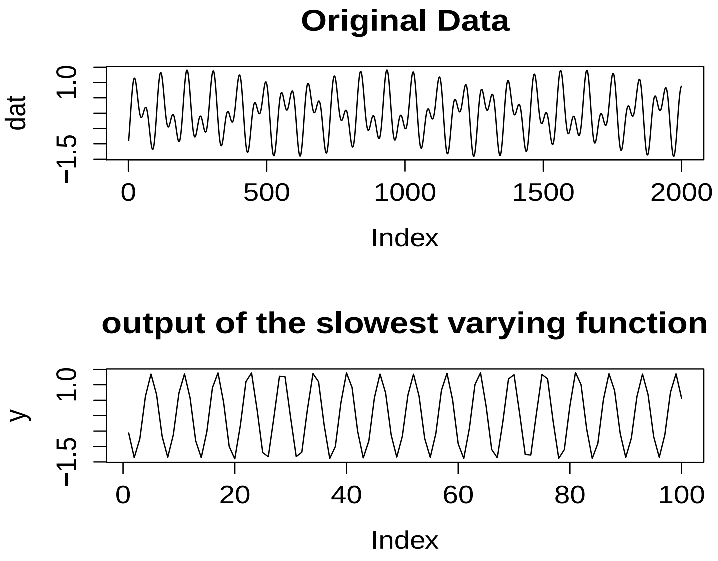


Figure 2.10: Slow Feature Analysis

2.4.5.3.2 James Stone's Criterion In 1996, Stone proposed [96] an unsupervised model for the extraction of salient visual parameters using spatial temporal smoothness constraints. The learning rule is based on the linear combination of hebbian and anti-hebbian weight update policy and the criteria is to learn salient visual parameters which change very slowly by maximizing the long-term variance of each unit's output and simultaneously minimizing the short term variance. The rationale behind considering long-term and short-term variance is explained by a real-time scenario which considers a sequence of images against an oriented, planar, textured surface moving relative to a fixed camera. The change in between frames of these two images is minimal. But simultaneously with this small change, a relative large change in the intensity of the individual pixel can occur. In other words there is a difference in the rate of change of the intensity of individual pixels and the corresponding rate of change of parameters associated with the imaged surface. Considering these Stone has proposed a model for extracting salient visual parameters. This model is based on a temporal smoothness constraint whose degree of smoothness can only be measured in terms of the short term variance associated with the sequence of output values. According to Stone a curve is considered smooth if the variance of the curve is minimal.

In order to implement this concept Stone proposed a multilayer neural model consisting of input, hidden and output layers labeled i, j and k. The state of an output unit $\mathbf{u}_k$ at each time t is $\boldsymbol{z}_{kt} = \sum_j \mathbf{w}_{jk} z_{jt}$ where $\mathbf{w}_{jk}$ is the value of weighted connection from hidden to output unit. The desired behavior in $\boldsymbol{z}_k$ is obtained by altering intra-unit weights by taking into account the long and short term variance of $\boldsymbol{z}_k$ shown in equation (2.27)

$$F = \mathtt{log}\frac{\mathbf{V}}{\mathbf{U}} = \mathtt{log}\frac{1/2\sum_{t=1}^{T}(\bar{\boldsymbol{z}}_t - \boldsymbol{z}_t)^2}{1/2\sum_{t=1}^{T}(\tilde{\boldsymbol{z}}_t - \boldsymbol{z}_t)^2}, \tag{2.27}$$

where the cumulative states $\bar{\boldsymbol{z}}$ and $\tilde{\boldsymbol{z}}$ are both exponentially weighted sums of states $\mathbf{z}$ given in equation (2.28) and (2.29).

$$\tilde{\boldsymbol{z}} = \lambda_S\tilde{\boldsymbol{z}}_t + (1-\lambda_S)\boldsymbol{z}_(t-1) : 0 \leq \lambda_S \leq 1, \tag{2.28}$$

$$\bar{\boldsymbol{z}} = \lambda_L\bar{\boldsymbol{z}}_t + (1-\lambda_L)\boldsymbol{z}_(t-1) : 0 \leq \lambda_L \leq 1, \tag{2.29}$$

where λ_L and λ_S are time decay constant and the half life of λ_L is much longer (typically 100 times longer) than the half life of λ_S.

2.4.5.3.3 Related State of The Art In [97] the authors developed an incremental method for slow feature analysis: This strategy is based on a combination of candid covariance-free incremental principal component analysis (CCIPCA) [98] and a covariance-free incremental minor component analysis (CCIMCA) [99]. This algorithm proceeds in two stages: the first whitens the data and removes any lower order principal components which are assumed to be noisy. The second stage performs the covariance-free minor component analysis. The first phase of the CCIPCA method is based on hebbian learning criteria. The second phase of CCIMCA is based on anti-hebbian learning criteria. The disadvantage of this technique is its two stages of computation on each new adaptive input which increases computational cost. Another drawback of this method is the repetitive use of CCIPCA in between whitening and derivation of adaptive data matrix. This problem is resolved by using CCIPCA only before whitening and deriving the matrix known as Fast Incremental Slow feature Analysis approach [100]. Similarly in [101] the authors proposed an online temporal video segmentation algorithm for incremental SFA. This method builds on a special kernelized version of IPCA [102], and produces a close approximation of SFA after each time stamp. The only limitation of this algorithm is its domain specificity to online temporal video segmentation.

The incremental versions of slow feature analysis proposed in the past are domain dependent, and requires previously processed information to learn a new adaptable chunk or data instance. The slow features can be extracted for the incoming adaptable data in a one pass incremental manner without using the existing adjacent information of the previously processed data [43].

2.4.5.4 Manifold Learning Approaches

Manifold based learning is an emerging and promising approach in non-linear dimensionality reduction techniques. Unlike other dimensionality reduction techniques such as principal component analysis [103] and multidimensional scaling [104], manifold-based learning technique find the most succinct low dimensional structure which is embedded in a higher dimensional space.

The concept of manifold based learning is defined as a topological space which is locally euclidean (Around every point, there is a neighborhood that is topologically the same as the open unit ball in R^n). Each point of an n-dimensional manifold is normally isomorphic to each other. The finest example of manifold is the shape of the earth where locally at each point on the surface of the earth is a 3D-coordinate system two for location and one for the altitude embedded in a 2D-sphere. In other words, it is a 2D manifold in a 3D space as shown in Figure 2.11.

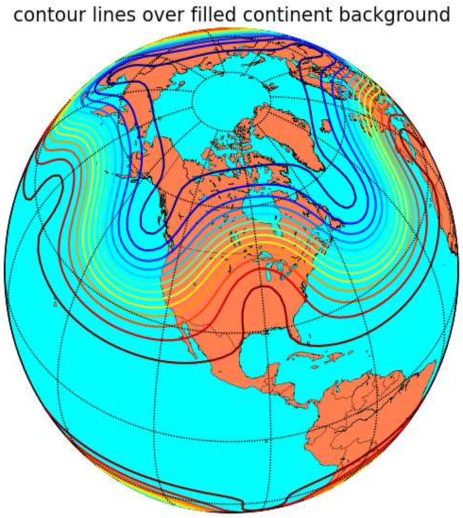

contour lines over filled continent background

Figure 2.11: An example of a manifold

As compared to other dimensionality reduction technique like PCA (Principal Component Analysis) where the actual goal is to find a set of mutually orthogonal basis functions which capture the directions of the maximum variance in the data so that the pairwise euclidean distances can be best preserved. In some datasets like face images, the data are sampled from a nonlinear low dimensional manifold embedded in a high dimensional space. This is the basis for using manifold based learning techniques. Manifold-based learning techniques have been used in many dimensionality reduction methods such as Generative Topographic Mapping (GTM) [105], Local Linear Embedding (LLE) [106], Laplacian Eigenmaps (LE) [107] and ISOMAP [108].

Local Linear Embedding and Laplacian Eigenmaps are based on preserving the local geometry of the data whereas ISOMAP is a global method normally attempts to preserve geometry at all scales.

The following section will focus on Laplacian Eigenmaps previously stated a manifold-based learning technique.

2.4.5.4.1 Laplacian Eigenmaps Laplacian Eigenmaps [107] is a locally optimized manifold-based dimensionality reduction technique which incorporates neighborhood information of the dataset into the graph. Secondly using the notion of the laplacian of the graph, a low dimensional representation of the dataset is computed which can optimally preserve local neighborhood information. This algorithm has a few local computations and one sparse eigenvalue problem.

Given l points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l$ in $\mathbf{R}^l$, we construct a weighted graph one for each point connected by the set of edges between neighboring points. The steps involved in the execution of a Laplacian Eigenmap are:

1. Step 1. [Construct an Adjacency Graph Matrix] Using the K-Nearest Neighbor algorithm on the complete dataset, create an edge between $\mathbf{x_i}$ and $\mathbf{x}_j$ if i is among the n nearest neighbor of j or j is among the n nearest neighbor of i.

2. Step 2. [Weighting the edges] There are two different variations for weighting the edges.

   (a) Heat Kernel. [$t \epsilon \Re$] if node i is connected with j put

$$\mathbf{W}_{ij} = e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}} \tag{2.30}$$

(b) Simple Approach. Set $\mathbf{W}_{ij} = 1$ if vertices i and j are connected by an edge and set $\mathbf{W}_{ij} = 0$ if vertices i and j are not connected by an edge.

3. Step 3. Construct the objective function. Consider the problem of mapping the weighted graph $\mathbf{G}$ to a lower dimensional space so that the connected points stay as close together as possible. Consider $\mathbf{y} = (y_1, y_2, ..., y_n)$ be such a map. A reasonable criteria of choosing an appropriate map is to minimize the following objective function:

$$\sum_{ij}(y_i - y_j)^2\mathbf{W}_{ij} \tag{2.31}$$

The minimization of the objective function is an attempt to avoid the heavy penalty which can occur if the neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Let $\mathbf{D}$ be a diagonal weight matrix, whose entries are (column or rows as $\mathbf{W}$ is a symmetric matrix) sums of $\mathbf{W}$. $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ji}$ and the laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{W}$. It turns out that for any $\mathbf{y}$, we can have

$$\frac{1}{2}\sum_{ij}(y_i - y_j)^2\mathbf{W}_{ij} = tr(\mathbf{y}^T\mathbf{L}\mathbf{y}) \tag{2.32}$$

The minimization problem can now be elaborated as arg min $tr(y^T\mathbf{L}y)$ such that

$$\mathbf{y}^T\mathbf{D}\mathbf{y} = 1 \tag{2.33}$$

and

$$\mathbf{y}^T\mathbf{D}1 = 0 \tag{2.34}$$

The bigger the $\mathbf{D}_{ii}$ is, the more significant $\mathbf{y}_i$ will be. There is a constraint as $\mathbf{y}^T\mathbf{D}\mathbf{y} = 1$ where constraint $\mathbf{y}^T\mathbf{D}1 = 0$ is to eliminate the trivial solution which collapses all vertices of $\mathbf{G}$ onto the real number 1.

4. Step 4. Compute the eigenvalues and eigenvectors by solving the generalized eigenvalue problem

$$\mathbf{L}\mathbf{f} = \lambda\mathbf{D}\mathbf{f} \tag{2.35}$$

where $\mathbf{D}$ is a diagonal weight matrix whose entries are the sum of each column of $\mathbf{W}$, i.e., $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is a laplacian matrix which is always symmetric and positive semi-definite.

Figure 2.12 shows the two dimensional projections of Laplacian Eigenmaps on swissroll dataset. Where $\mathbf{y}_1$ and $\mathbf{y}_1$ are eigenvectors corresponding to the second smallest and third smallest eigenvalues.

2.4.5.4.1.1 Related State of The Art In [72] the authors have proposed two incremental versions of Laplacian Eigenmaps. Firstly, an algorithm is presented which incrementally computes low dimensional representation of data set by optimally preserving local neighborhood information. Secondly, a submanifold analysis algorithm combined with an alternative formulation of linear incremental method is proposed to learn the new samples incrementally. The

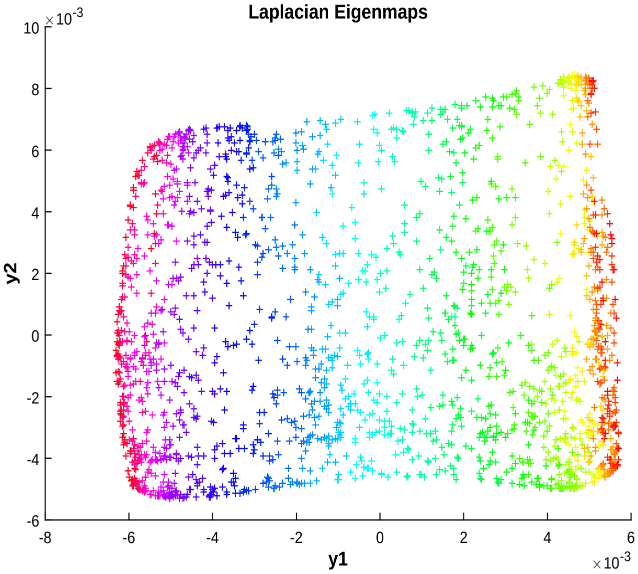


Figure 2.12: Two dimensional view of SwissRoll dataset using LE

only deficiency in these methods is the calculation of low dimensional embedding of the new adaptable information using an existing adjacent information of the previously processed data.

The low dimensional embedding of the incoming data can be calculated incrementally in one pass without using the existing adjacent information of the previously processed data as proposed in chapter 6 [109].

2.4.5.5 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis is a well-known technique since its first formulation ([110]). It is used primarily for finding filters between two streams of data that maximize the correlation between those two filters. This is now a standard data analysis technique.

In canonical correlation analysis between dual streams, the subspace vectors **a** and **b** from both the streams are extracted in such a way that the correlation will be maximized as shown diagrammatically in Figure (2.13) while projecting $\mathbf{X}$ onto $\mathbf{a}$ by $\mathbf{X}^* = \mathbf{a}^T\mathbf{X}$ and projecting $\mathbf{Y}$ onto $\mathbf{b}$ by $\mathbf{Y}^* = \mathbf{b}^T\mathbf{Y}$ between both the vectors.

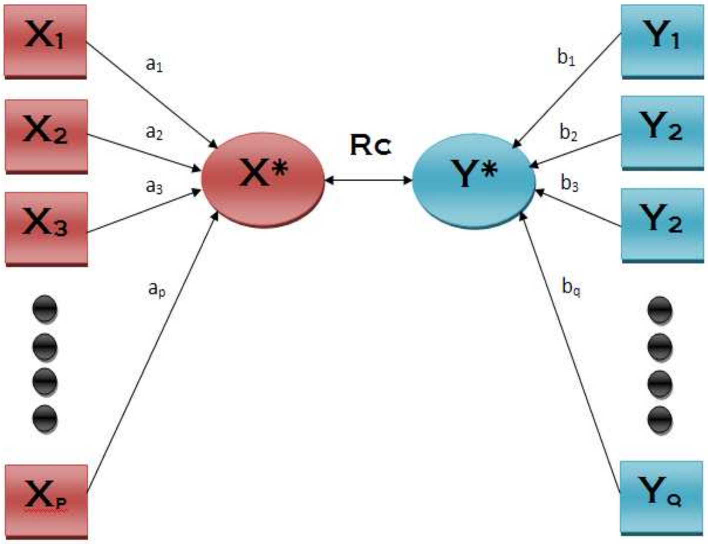


Figure 2.13: Canonical Correlation Analysis

Some online incremental versions of CCA in [111, 112, 113, 114] are often based on artificial neural networks. Some of these techniques involved minimizing the squared difference between the outputs of two twinned neural networks [115], each devoted to one of the data streams. The implementation of CCA in [115] is derived by phrasing the problem as of maximizing the objective function shown in equation 2.36.

$$J = E(\mathbf{y}_1\mathbf{y}_2) + \frac{1}{2}\lambda_1(1 - \mathbf{y}_1^2) + \frac{1}{2}\lambda_2(1 - \mathbf{y}_2^2), \tag{2.36}$$

where $\mathbf{y}_1$ and $\mathbf{y}_2$ are the outputs and λ_i is explicitly used as a Lagrange multiplier to put a constraint on the finite weight values. Similarly in [112] the authors have followed the concept of finding canonical correlations in [116] and derived an incremental version in which a method of finding canonical correlation is proposed by solving the generalized eigenvalue problem shown in (2.37).

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}, \tag{2.37}$$

where Σ_{ij} is the covariance matrix between $\mathbf{x}_i$ and $\mathbf{x}_j$. This concept is extended in [114] by applying gradient descent on the Bregman divergences between the two data streams; reservoirs used to reduce the non-linearity in the data.

Using this formulation they have shown [111] that the canonical correlation directions $\mathbf{w}_1$ and $\mathbf{w}_2$ may be found using

$$\begin{aligned} \frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 - f(\mathbf{w})\Sigma_{11}\mathbf{w}_1 \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 - f(\mathbf{w})\Sigma_{22}\mathbf{w}_2 \end{aligned}$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i\mathbf{x}_j^T), i, j = 1, 2$, where T denotes the transpose, we derive the instantaneous versions

$$\begin{aligned} \Delta\mathbf{w}_1 &= \eta(\mathbf{x}_1 y_2 - f(\mathbf{w})\mathbf{x}_1 y_1) \\ \Delta\mathbf{w}_2 &= \eta(\mathbf{x}_2 y_1 - f(\mathbf{w})\mathbf{x}_2 y_2) \end{aligned}$$

which was shown to provide a family of networks capable of performing CCA.

2.4.5.5.1 Multi-Set Canonical Correlation Analysis (MCCA) Multi-set canonical correlation analysis (MCCA) [117, 118] is a technique which can analyze linear relationship between 2 or more sets of variables. It can be considered as a generalized extension of CCA. The MCCA approach is proposed by Kettenring (1971) to find the co-variation across modalities and subjects. Thus we can summarize MCCA as a sum of squares cost shown diagrammatically in Figure 2.14.

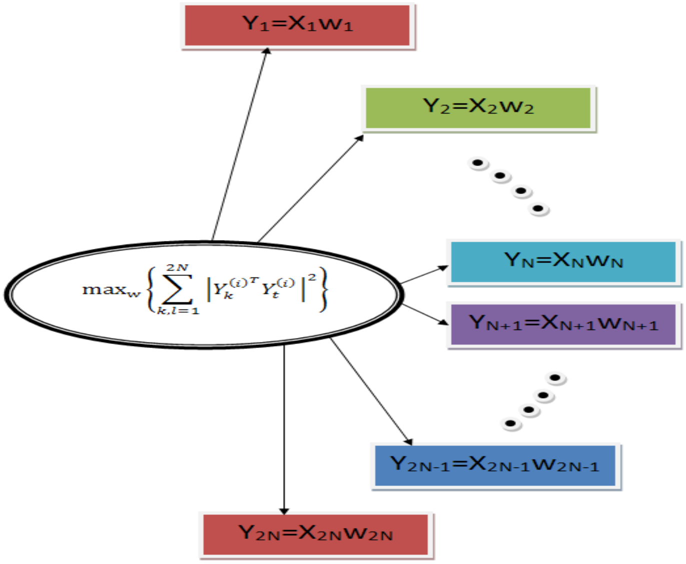

$\max_w \left\{ \sum_{k,l=1}^{2N} \left| Y_k^{(i)T} Y_t^{(i)} \right|^2 \right\}$
$Y_1=X_1w_1$
$Y_2=X_2w_2$
$Y_N=X_Nw_N$
$Y_{N+1}=X_{N+1}w_{N+1}$
$Y_{2N-1}=X_{2N-1}w_{2N-1}$
$Y_{2N}=X_{2N}w_{2N}$

Figure 2.14: Multi-set Canonical Correlation Analysis

2.4.5.5.2 Related State of The Art In [119] the authors proposed an adaptive formulation of the classical CCA algorithm based on matrix manifolds. They solved the optimization problem on matrix manifolds using classical gradient algorithm and designed the adaptive CCA algorithm based on rationale to make the algorithm capable of detecting the exact time stamp when change (designed for change detection in the correlation of two time series) occurred in the subspace. An incremental approach based on the recursive least square algorithm, has also been proposed in [120] for rank-one CCA problem. This procedure can cope with multiple orthogonal projections using a deflation scheme. The emphasis in this paper is the extension of CCA to cope with more than two data sets simultaneously. The authors performed a valuable comparative study of various algorithms. The strength of the proposed CCA method as a set of coupled least squares (LS) regression problems is its flexibility of execution, which leads to the development of both batch and adaptive learning algorithms for CCA. Another positive aspect is its faster convergence as it does not require pre-whitening (SVD) step. Further it is able to find the canonical vector and variates directly from the datasets. The weakness is that the proposed method can be considered applicable only for linear datasets compared to non-linear datasets which requires adoption of kernels for non-linear transformation.

Researchers have maximized the correlations between dual streams by introducing kernels [113, 121]. Recently a temporal kernel CCA approach has been proposed [122] in which a novel method based on kernels was introduced. This approach computes multivariate temporal filters between data sources with different dimensionality and temporal resolutions. The rationale for using kernels is to perform transformation, and to obtain a representation of the data in an implicit high-dimensional latent space. The word implicit is said because usually the actual representation in this space is not used since the kernel trick enables to manipulate algorithms by using only the dot product of the implicit representations. These methods are especially useful for a relatively small number of high dimensional samples. Further, they have been useful in solving a nonlinear problem by converting it into to a linear problem.

2.4.6 Information Expansion

This section primarily presents two techniques used as a pre-processor in the following chapters for linear transformation namely (1) Randomized Expansion [123, 124, 125, 126] and (2) Time series Expansion [127]. This section presents a brief overview of the networks that use randomized and time series expansion. Additionally this section also provides a brief overview of radial basis function a well known traditional information expansion technique.

2.4.6.1 Randomized Expansion

Wouter F. Schmidt et al [123] firstly proposed a feed forward neural network by randomly choosing the weights of the hidden layer, where the output layer is trained by a single layer learning rule or a pseudo-inverse strategy. The randomization of the hidden layer of the feedforward neural network calculates the weighted sum by using a squashing function $F(x)$ which maps the values of the random weights between 0 and 1. The squashing function is the sigmoid function given as:

$$F(x) = \frac{1}{1+e^{-x}} \tag{2.38}$$

To formulate this idea, suppose $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n, \mathbf{m}_{n+1}]^T$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n+1}]^T$ are the weight and input vectors respectively. The output at hidden layer is described as

$$O_{hidden} = F(\sum \mathbf{m}_i \mathbf{x}_i) = F(\mathbf{M}^T\mathbf{X}) \tag{2.39}$$

In mathematical formulation the output layer perform the following function:

$$O_{output} = \sum \mathbf{w}_i \mathbf{x}_i = \mathbf{W}^t \mathbf{X} \tag{2.40}$$

The total function computed by the network can be formulated as:

$$O_{net}(\mathbf{X}) = \sum_{1}^{hidden} \mathbf{w}_i F(\mathbf{M}_i^T \mathbf{X}) + \mathbf{w}_{hidden+1} \tag{2.41}$$

The weight vector $\mathbf{W}^*$ for which this equation is minimal is calculated as follows:

$$\mathbf{W}^* = \mathbf{R}^{-1} \mathbf{P} \tag{2.42}$$

where $\mathbf{R}$ and $\mathbf{P}$ are the input correlation matrix and input-target correlation vector respectively of the output unit.

RVFL

Random Vector Functional Link (RVFL) technique [124] also uses the randomized hidden nodes without training and training at the output. In this technique, the inputs are enhanced to $\mathbf{x}_n$ with elements $(\mathbf{x}_{n1}, \mathbf{x}_{n2}, ..., \mathbf{x}_{nj}, ..., \mathbf{x}_{nJ})$. The target outputs are $\mathbf{t}_{nk}$ and the weights are $\mathbf{w}_{kj}$. In functional-link networks, the outputs $\mathbf{t}_k$ can be treated independently of each other. The weights $\boldsymbol{\beta}_j$ is initially assigned with random values. For each input pattern the change in the weights is taken as:

$$\Delta \boldsymbol{\beta}_{nj} = \eta(\mathbf{t}_n - \mathbf{o}_n)\mathbf{x}_{nj} \tag{2.43}$$

The weights are updated once the changes are calculated for all the patterns in the training set as:

$$\boldsymbol{\beta}_j(k+1) = \boldsymbol{\beta}_j(k) + \sum_p \Delta\boldsymbol{\beta}_{nj} \tag{2.44}$$

The learning rate η may be increased as $(\mathbf{t}_p - \mathbf{o}_p)$ decreases.

The network only adjust the weights $\boldsymbol{\beta}_j$ to minimize the system error

$$E = \sum_j E_n = \sum_n (\boldsymbol{net}_n - \boldsymbol{penalty}_n)^2 \tag{2.45}$$

where symbol $\mathbf{penalty}_n$ is the target output and $\mathbf{net}_n$ is the actual output of the functional net.

Extreme Learning Machine

Recently a similar kind of new learning algorithm for single layer feedforward neural network[128] was proposed by Huang et al [125, 126]. The only difference between ELM and other algorithms is its high efficiency for auto encoder (Deep Belief Networks) [129]. Another important characteristic of ELM is its parameters of the hidden nodes. These parameters are randomly generated independently from the training samples and they are independent from each other in a wide type of neural networks and mathematical series/ expansion as well as in biological learning mechanism [130].

The reason for the evolution of extreme learning machine [131] is that it has the capacity of extremely fast learning. Since ELM is easy to implement, tends to achieve the smallest training error and good generalization performance. It is true that the learning speed of other feedforward neural networks is in general far slower than required and it has been a major bottleneck in their applications [132].

The model of ELM is similar to other standard multiple layer feed forward neural network models consisting mainly of three layers 1) input layer, 2) hidden layer and 3) output layer. The basic ELM neural architecture is shown in Figure 2.15.

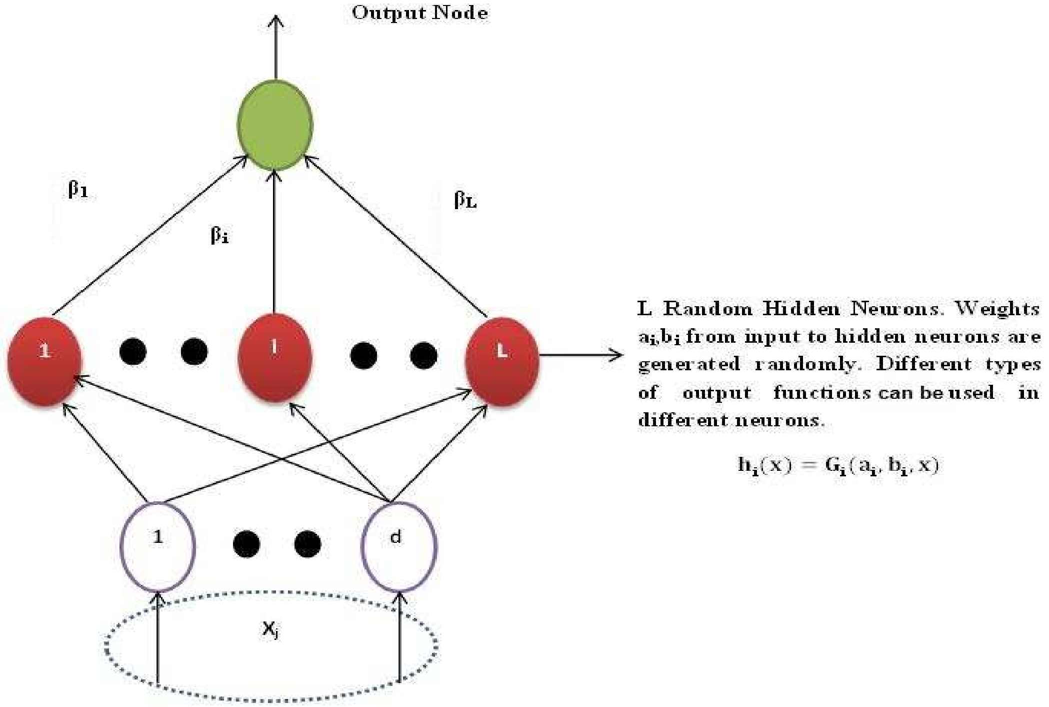

Output Node
β1
βi
βL
1
i
L
L Random Hidden Neurons. Weights ai,bi from input to hidden neurons are generated randomly. Different types of output functions can be used in different neurons.
hi(x) = Gi(ai, bi, x)
1
d
Xj

Figure 2.15: A Schematic Overview of an ELM

For **N** arbitrary distinct sample $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{in}]\ \epsilon\ \mathbf{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, ..., t_{im}]$, standard single layer feedforward network (SLFN) with $\tilde{\mathbf{N}}$ hidden nodes and activation function $g(\mathbf{x})$ are mathematically modeled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i.\mathbf{x}_j + b_i) = \mathbf{o}_j, j = 1, 2, ..., \mathbf{N}, \tag{2.46}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]$ is the weight vector connecting the ith hidden node and the input node, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{in}]^T$ is the weight vector connecting the ith hidden node and the output nodes, and b_i is the threshold of the ith hidden node $\mathbf{w}_i.\mathbf{x}_j$ denoting the inner product of $\mathbf{w}_i$ and $\mathbf{x}_j$.

The single-hidden layer feed forward neural networks (SLFNs) of extreme learning machine (ELM) is further enhanced to the generalized single hidden layer feedforward neural network [133].

The output function of ELM for generalized SLFNs is

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i \mathbf{h}_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\beta, \tag{2.47}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_L]^T$ is the vector of output weights from hidden to output nodes, and $\mathbf{h}(\mathbf{x}) = [h_1(x), h_2(x), ..., h_L(x)]$ is the output (row) vector of the hidden layer with respect to the input $\mathbf{x}$. The output function $\mathbf{h}_i(\mathbf{x})$ of hidden nodes may not be unique. In particular, in real applications, $\mathbf{h}_i(\mathbf{x})$ can be

$$h_i(\mathbf{x}) = G(\mathbf{a}_i, b_i, \mathbf{x}), \mathbf{a}_i \; \epsilon \; \mathbf{R}^d, b_i \; \epsilon \; \mathbf{R}, \tag{2.48}$$

where $G(\mathbf{a}_i, b_i, \mathbf{x})$ is a non-linear piecewise continuous function satisfying ELM universal approximation capability theorem [133, 134].

As the hidden neurons of ELM are totally independent from each other, different output activation functions can be used to compute output of each hidden neuron including sigmoid function, fourier function, hardlimit function, Gaussian function and multiquadrics function.

The equation (2.46) can be written compactly as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2.49}$$

where

$$H(\mathbf{w}_1, ..., \mathbf{w}_{\tilde{N}}, b_1, ..., b_{\tilde{N}}, \mathbf{x}_1, ..., \mathbf{x}_N)$$

$$= \begin{bmatrix} g(\mathbf{w}_1.\mathbf{x}_1 + b_1) & ... & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_1 + b_{\tilde{N}}) \\ & ... & \\ g(\mathbf{w}_1.\mathbf{x}_N + b_1) & ... & g(\mathbf{w}_{\tilde{N}}.\mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{Nx\tilde{N}} \tag{2.50}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ . \\ . \\ \beta_N^T \end{bmatrix} \text{ and } \boldsymbol{T} = \begin{bmatrix} t_1^T \\ . \\ . \\ t_N^T \end{bmatrix}_{Nxm} \tag{2.51}$$

In order to satisfy the learning criteria at the output stage, the standard ELM implementation uses the minimal norm least square method instead of the standard optimization method in the solution.

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{2.52}$$

where $\mathbf{H}^\dagger$ is the *Moore-Penrose generalized inverse* of matrix $\mathbf{H}$ [135].

Echo State Network

Echo State Network is another most popular type of Recurrent Neural Network (RNN) proposed by H. Jaeger [127] which is driven by a (single or multidimensional) time signal whose activations are generated using tanh (tan hyperbolic) function. These activations are used to perform linear classification/ regression.

Echo state networks consist of three layers of 'neurons': an input layer which is connected with random and fixed weights to the next layer which forms the reservoir. The neurons of the reservoir are connected to other neurons in the reservoir with a fixed, random, sparse matrix of weights. Typically only about 10 % of the weights in the reservoir are non-zero. The reservoir is connected to the output neurons using weights which are trained using error descent. In echo state networks only weights connected to the output neurons needs training making this network an easy and efficiently trained network. Echo State Networks are specialized in converting non-linear information into information of temporal nature.

The idea of reservoir can be stated as: $\mathbf{W}_{in}$ denotes the weights from the N_u, inputs $\mathbf{u}$, to the N_x reservoir units $\mathbf{x}$, $\mathbf{W}$ denotes the $N_x \times N_x$ reservoir weight matrix, and $\mathbf{W}_{out}$ denotes the $(N_x + 1) \times N_y$ weight matrix linking the reservoir units to the output units, denoted by $\mathbf{y}$ as shown diagrammatically in Figure 2.16.

The network dynamics are governed by

$$\mathbf{x}(t) = f(\mathbf{W}_{in}\mathbf{u}(t) + \mathbf{W}\mathbf{x}(t-1)) \tag{2.53}$$

where typically $f(.) = tanh(.)$ and t is the time index. The feed forward stage is given by

$$\mathbf{y} = \mathbf{W}_{out}\mathbf{x} \tag{2.54}$$

This is followed by a supervised learning of the output weights, $\mathbf{W}_{out}$. If we are using online learning, a simple least mean square rule gives

$$\mathbf{W}_{out} = \mathbf{W}_{out} + \eta(\mathbf{y}_{target} - \mathbf{y})\mathbf{x}^T \tag{2.55}$$

where η is a learning rate (step size) and $\mathbf{y}_{target}$ is the target output corresponding to the current input.

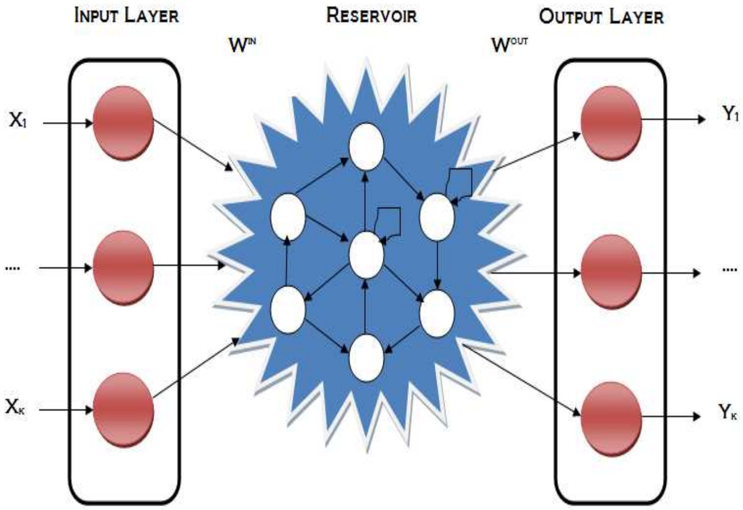


Figure 2.16: Topology of Echo State Network

2.4.6.2 Radial Basis Function

Radial Basis functions are simply a class of functions which can be employed in any sort of model (linear or nonlinear) and any sort of network (single-layer or multi-layer). According to [136] radial basis functions are traditionally associated with a single layer network shown in Figure 2.17.

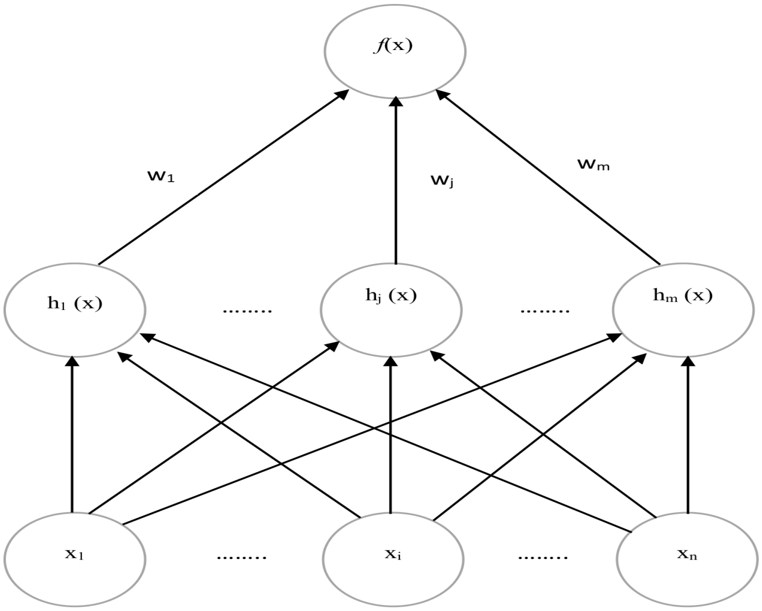


Figure 2.17: Radial Basis Function Network

The characteristic feature of radial basis function is that their response decreases (or increases) monotonically with distance from a central point. The parameters of radial basis function are (1) centre, (2) distance scale and (3) precise shape of the radial function.

A typical radial function is a Gaussian, in case of scalar input is:

$$h(x) = exp\left(-\frac{(x-c)^2}{r^2}\right) \tag{2.56}$$

where c is its centre and r radius be its parameters.

2.4.7 Post-Processing

Post-Processing techniques provide a symbolic filter for noisy and imprecise knowledge derived by machine learning algorithm [84]. This phase is used to simplify visualize or document the knowledge extracted from data. Post-processing is primarily categorized into following groups below:

2.4.7.1 Knowledge filtering, Rule truncation and postpruning

Post-pruning or Rule truncation is one primary task which is performed in the post-processing phase. This happens because the machine learning inductive algorithm split subsets of training objects to smaller subsets that would be genuinely consistent. To overcome this issue a tree or a decision set of rules are optimized by using postpruning (decision trees) or truncation (decision rules) techniques [137].

2.4.7.2 Interpretation

This phase includes the listing documentation of results, the transformation of knowledge to an understandable form and the visualization of the extracted knowledge [138]. Further this phase includes summary of the rules while combining them with a domain specific knowledge provided for the given task.

2.4.7.3 Evaluation

This phase includes the evaluation (or testing) of learned model on training data set. There are several widely used criteria for measuring the performance of learning including classification accuracy, comprehensibility, computational complexity and predictive accuracy [139].

2.4.7.4 Knowledge Integration

This phase is primarily used in those scenarios where the decision supporting systems needs to be combined from several models. Use of knowledge integration methods increases reliability and likelihood of success [84].

2.4.8 Model Evaluation Techniques

The section reviews some of the tools needed for assessing the quality of regression and classification models.

2.4.8.1 Evaluating Regression Quality

Following in [140]: Suppose $\hat{\theta}$ be the estimator of the unknown parameter θ from the random samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. The deviation from $\hat{\theta}$ to the true value θ which can be calculated by subtracting $\hat{\theta}$ from θ, $|\hat{\theta} - \theta|$ measures the regression quality.

Definition: The mean square error (MSE) [141] of an observer $\hat{\theta}$ of a parameter θ is the function of θ defined by $E(\hat{\theta} - \theta)^2$, and is denoted as $\mathbf{MSE}_{\hat{\theta}}$. Hence MSE is mathematically calculated as:

$$MSE_{\hat{\theta}} = E(\hat{\theta} - \theta)^2 \tag{2.57}$$

To calculate MSE, firstly calculate the squared difference between the predictions and true values. The sum of the squared errors (SSE) is calculated as

$$SSE = \sum_{\mathbf{x} \epsilon X} (f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2 \tag{2.58}$$

where $(\mathbf{x}, f(\mathbf{x}))$ can be generated by collecting data from experiments even though the true workings of $f(\mathbf{x})$ are unknown, $f(\hat{\mathbf{x}})$ is the function of estimation, m is the size of the data set used to build the model and $D = \{(X, y)\}$ is the set of training examples.

The mean squared error (MSE) divides the SSE by the count of data points used to produce a measure of error variance

$$MSE = \frac{SSE}{m} \tag{2.59}$$

and the root mean squared error (RMSE) calculates error in the same units as the output (i.e. a standard deviation)

$$RMSE = \sqrt{MSE} \tag{2.60}$$

An alternative method in [142] is to calculate Pearson's correlation coefficient (p) between the true and estimated output is:

$$p = \frac{\sum (\mathbf{X} - \bar{\mathbf{X}})(\mathbf{Y} - \bar{\mathbf{Y}})}{\sqrt{\sum (\mathbf{X} - \bar{\mathbf{X}})^2} \sqrt{(\mathbf{Y} - \bar{\mathbf{Y}})^2}} \tag{2.61}$$

where the value of p can vary between -1 and 1, but should never be negative where it is comparing predicted and actual values. Values closer to 1 indicates a better fit.

2.4.8.2 Measuring Accuracy, Precision-Recall

The binary classification models often aim to group each entry of a given dataset into two groups, positive and negative, according to the classification criteria. The performance of a classification model for each element of the dataset is based on four possible outcomes with regards to the classification:

1. true positive (TP): when an entry is positively classified
2. true negative (TN): when an entry is negatively classified
3. false positive (FP): when a negative entry is correctly classified as negative
4. false negative (FN): when a negative entry is wrongly classified as positive

Given P = TP + FN the number of real positives in the considered dataset, and N = TN + FP the number of real negatives, we can define [143]

1. sensitivity ratio = true positive rate (TPR) = recall = TP/P
2. specificity ratio= true negative rate (TNR) = TN/N
3. positive predictive value (PPV)= precision = TP/(TP+FP)
4. negative predictive value (NPV)= TN/(TN+FN)
5. accuracy = (TP+TN)/(P+N)
6. Error (Classification Error) = 1 - accuracy = 1 - (TP+TN)/P+N

A confusion matrix is shown in Table 2.1 [143] which shows the relationship between actual and predicted values.

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

Table 2.1: Confusion Matrix

2.4.8.3 Measuring PMF Model Quality

The output $\hat{P}(\mathbf{x})$ when modeling a probability mass function is an estimate of a probability $P(\mathbf{x})$ rather than a function output, so other methods of measuring model quality are used. One such measure is the likelihood of the data.

Given the model: $(\mathbf{x}|\theta)$ where θ is the set of parameters describing the model. The likelihood of a model is calculated as:

$$L(\theta|X) = \sum_{x \epsilon X} -ln(\hat{P}(\mathbf{x})) \tag{2.62}$$

where $\hat{P}(\mathbf{x})$ is the model's estimate of the probability of $\mathbf{x}$. The greater the probability, the better the fit of the model to the data [144]. The likelihood is maximized for a set of data and a model when the model's estimates of the probabilities of each pattern in the data match the distribution of the data.

2.4.8.4 Regularization

The criteria of controlling the complexity of regression and probability models as a way of managing how well they fit the data is known as regularization [145]. L_1 and L_2 norms [146] are the most common regularization approaches. A norm is the distance from the origin to the vector point. By considering the parameters of a model as a vector, θ, the norm is a reflection of both the number of parameters in a model and of their size. The norm $||\theta||$ measures the distance of a model from the simplest point, taking the simplest model possible to be at the origin (all parameters=0). The L_1 norm termed as Manhattan Distance is the sum of absolute values:

$$||\mathbf{x}||_1 = \sum_{x \epsilon X} |x| \tag{2.63}$$

and the L_2 norm also termed as Euclidean Distance is the sum of square of absolute values.

$$||\mathbf{x}||_2 = \sqrt{\sum_{x \epsilon X} x^2} \tag{2.64}$$

Regularization is used in the model fitting process by adding the norm to the cost function of the model so that it becomes $C_\theta(D) + \alpha||\theta||$. The parameter α controls the contribution of the regularization term. L_1 norm has the advantage of leading to simpler models whereas L_2 has the advantage of allowing a solution to be found by gradient descent. This type of regularization is used by regression models which includes ridge regression [147] and LASSO (least absolute shrinkage and selection operator) [148].

2.4.8.5 Entropy Based Methods

Entropy is defined in [149] as a measure of variations, or uncertainty of occurrence of an event. The larger the entropy is, the lesser the chances of occurrence of an event and vice versa. For example consider a pair of variables, (a, b) with marginal distributions of $p(a)$ and $p(b)$ and a joint distribution of $p(a, b)$.

The Shannon entropy of either variable (a for example, measured across samples in A) is measured as

$$H(A) = \sum_{a \epsilon A} -P(a) log P(a) \tag{2.65}$$

measures the variation in A. H(A) is maximized when A has a uniform distribution. The conditional entropy of b given a measures the uncertainty that remains in b when a is known and calculated as

$$H(B|A) = \sum_{a \epsilon A, b \epsilon B} log \frac{P(a)}{P(a,b)} \tag{2.66}$$

If knowing A leaves the entropy of B unchanged, then a and b are independent as $H(B) = H(B|A)$. Any reduction in $H(B)$ provided by knowing A is the information gain:

$$H(B) - H(B|A) \tag{2.67}$$

also known as the mutual information, and can be calculated as

$$I(A;B) = \sum_{a \epsilon A} \sum_{b \epsilon B} log \frac{P(a,b)}{P(a)P(b)} \tag{2.68}$$

This relative entropy is equivalent to the expectation of the Kullback-Leibler divergence [150] between the marginal distribution $p(a)$ and the joint $p(a,b)$.

2.4.9 Validation Techniques

Validation techniques for assessing how the results of a statistical analysis will generalize to an independent set or to an appropriate criteria is presented.

2.4.9.1 Holdout

This methods randomly divides the available dataset into two subsets, the training set and the testing set (or holdout set). The first subset is used to build the model, and the second subset is used to assess the predictability of a model. The holdout set generally underestimates the accuracy of the model, inducing a large bias, because only a portion of the data is available for the learning process. On the contrary, using a test subset with fewer data points will greatly increase the variance of the accuracy estimation [151].

2.4.9.2 Bootstrap

Given a dataset of size n, the bootstrap method creates a number of testing subsets by sampling n instances uniformly from data with replacements. For example some of the data instances will appear in the bootstrap sample multiple times while other data instances will not appear at all.

The probability p of any given data point being chosen after n samples, b the number of bootstrap samples, α_i the accuracy computed on the i-th sample, and a the accuracy computed on the whole training set, the accuracy estimate can be expressed as $\frac{1}{b}\sum_i p.\alpha_i + (1-p).a$.

The variance of accuracy is calculated as the variance of estimate for the samples. It has been proved that bootstrap has low variance, but, it may present an extremely large bias to some problems [151].

2.4.9.3 Cross Validation

K-Fold cross-validation technique is used to achieve an unbiased estimate of the model performance only when a limited amount of data is available [151]. The K-fold cross-validation technique divides the available dataset into k partitions of similar size. A validation model is built k times, using each time $k-1$ partitions as the training set and the remaining partition as the testing set. The estimation of accuracy of the validation model is the average of the accuracies computed for the developed k models. If the data distributions is composed of n data points, the n-fold cross-validation is also known as leave-one-out cross-validation [151].

2.5 Towards Domain independent learning

This chapter gave a thorough description and state of the art review of online feature extraction techniques. These procedures are considered as a baseline for contribution to this thesis. Additionally, to highlight the importance and use of feature extraction in machine learning problems, a broader picture of an intelligent system and description of its major phases was also presented.

This overview showed the potential of feature extraction techniques. The criticalities found in the published literature, highlighted the need for an online domain independent learning mechanism; which could effectively learn each chunk of adaptive input independently.

In the following chapters, online learning techniques will be proposed in the area of Linear Discriminant Analysis, Slow Feature Analysis, Canonical Correlation Analysis and Laplacian Eigenmaps. The main objective will be to minimize the dependency for learning new adaptable information from previously learned information.

The next chapter will describe an online version of Linear Discriminant Analysis linked with extreme learning machine as a pre-processor. As previously stated, the main objectives are Firstly to minimize the dependency on previously learned information. Secondly, to maximize the discrimination among points belonging to two different classes and to minimize the distance between points within each class.

Chapter 3

Incremental Linear Discriminant Analysis with Extreme Learning Machine

This chapter presents an incremental version of linear discriminant analysis, an algorithm focusing on one data point at a time in a completely adaptive manner. Further the proposed algorithm doesn't depend on previously learned information.

Additionally this algorithm is combined with a randomized expansion of extreme learning machine, before learning for linear transformation. This procedure maximizes the discrimination among points belonging to two different classes, and minimizes the distance among points within each category.

Finally the comparative effectiveness of the proposed algorithm is demonstrated using both artificial and real data sets.

3.1 The GENILDA (An Incremental Method For Generalized Eigenproblems) Method

The differential equation in [55] is applied to find the linear discriminant subspace by finding the maximum eigenvalue of

$$E((\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2))^T\mathbf{w}=\lambda' E((\mathbf{x}-\boldsymbol{\mu}_j)(\mathbf{x}-\boldsymbol{\mu}_j)^T)\mathbf{w}, \tag{3.1}$$

where $j=1,2$ and $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are within class means.

This gives us an update of

$$\Delta\mathbf{w}=\mathbf{S_B}\mathbf{w}-f(\mathbf{w})\mathbf{S_w}\mathbf{w}, \tag{3.2}$$

where $\mathbf{S_w}=E((\mathbf{x}-\boldsymbol{\mu}_j)(\mathbf{x}-\boldsymbol{\mu}_j)^T)$ is a within class scatter matrix, $j=1,2$ and $\mathbf{S_B}=E((\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T)$ is a between class scatter matrix. Where $\boldsymbol{\mu}_1=\frac{\sum_{i=1}^{n_1}\mathbf{x_i}}{n_1}$, n_1 ϵ data points in class 1 and $\boldsymbol{\mu}_2=\frac{\sum_{i=1}^{n_2}\mathbf{x_i}}{n_2}$, n_2 ϵ data points in class 2.

In fact to make the solution neural by updating the weights in an online mode is derived by replacing the between and within scatter matrices in equation (3.2) with the instantaneous values so that

$$\Delta\mathbf{w}=((\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T)\mathbf{w}-f(\mathbf{w})\sum_{j=1}^{2}(\mathbf{x}_i-\boldsymbol{\mu}_j)(\mathbf{x}_i-\boldsymbol{\mu}_j)^T\mathbf{w}, \tag{3.3}$$

where $i=1,2,3,...,N$ and $j=1,2$.

This makes the proposed algorithm incremental as all the data points are now individually learned point by point by updating the filter on each iteration. Learning point by point enables the proposed algorithm to calculate the subspace filter which can maximize the discrimination between data points belonging to different classes and simultaneously minimizes the distance between points within each class. Further the lipschitz continuous function $f(w)$ keeps the weight filter values away from reaching infinity (growing out of bounds) at each iteration and act more like a regularization function. Further, it helps the algorithm to converge quickly by reaching the stable equilibrium position in very few iterations.

In the above online algorithm for computing linear discriminant subspace affecting binomial classes, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ will be constant throughout the process, making the algorithm computationally less expensive.

Similarly, for multinomial classes the criteria for updating the weights in an online mode, by replacing the between and within scatter matrices in equation (3.2) will be

$$\Delta \mathbf{w} = \sum_{j=1}^{k} ((\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T)\mathbf{w} - f(\mathbf{w}) \sum_{j=1}^{k} ((\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^T)\mathbf{w}, \qquad (3.4)$$

where $i = 1, 2, 3, ..., N$, $k = 1, 2, 3, ..., C$ and $\mu = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}$ is the global mean of the whole data.

In equation (3.4), N represents the total number of data points and C represents the total number of classes. The overall mean of the whole data and the between class mean will be updated on the addition of new single input or new chunk as

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \frac{\mathbf{x}}{N+1} \tag{3.5}$$

and

$$\boldsymbol{\mu}_c' = \boldsymbol{\mu}_c + \frac{\mathbf{x}}{N+1}, \tag{3.6}$$

where c ϵ class label or

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \sum_{i=1}^{r} \frac{\mathbf{x}_i}{N+r} \tag{3.7}$$

and

$$\boldsymbol{\mu}_c' = \boldsymbol{\mu}_c + \sum_{i=1}^{r} \frac{\mathbf{x}_i}{N+r}, \tag{3.8}$$

where c ϵ class label and r ϵ chunk size.

Thus each new adaptable chunk, or data instance, is learned independently by using the previously learned filter, and only updating the within class mean, and global mean once on each new adaptable input. Hence this approach can be considered as a one-pass incremental learning algorithm which acquires knowledge with a single presentation of the training data.

In order to find the next filter based on the second highest eigenvalue, the matrices $\mathbf{S}_w$ and $\mathbf{S}_B$ are deflated and again used incrementally by solving the generalized eigenvalue problem.

Table 3.1: Algorithm 1 The GENILDA Algorithm

```
**Input:**
The labeled patterns, $(\mathbf{X}_N, \mathbf{Y}_N) = (\mathbf{x}_i, \mathbf{y}_i)$ where $i = 1, 2, 3, ..., N$
**Output:**
The mapping function of GENILDA: $f$:$\Re^{n_i} - > \Re^{n_o}$
**Step 1**: Calculate the (updated if any) or new $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ for binomial classes where
$\boldsymbol{\mu}_1 = \frac{\sum_{i=1}^{n_1} \mathbf{x_i}}{n_1}$, $n_1$ $\epsilon$ data points in class 1 and $\boldsymbol{\mu}_2 = \frac{\sum_{i=1}^{n_2} \mathbf{x_i}}{n_2}$, $n_2$ $\epsilon$ data points in class 2.
if multinomial calculate the updated (if any) or new $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, ..., \boldsymbol{\mu}_N$ and global mean $\boldsymbol{\mu}$ of the whole data.
**Step 2:**
For $i = 1$ to N iterations
**Step 2.1:** Calculate $\mathbf{S}_B$ where $\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$
for multinomial $\mathbf{S}_B = \sum_{j=1}^{k}((\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T)$
**Step 2.2:** Calculate $\mathbf{S}_w$ where $\mathbf{S}_w = \sum_{j=1}^{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^T$
For multinomial $\mathbf{S}_w = \sum_{j=1}^{k}((\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^T)$
**Step 2.3:** Calculate the updated (if any) or new eigenspace of $\mathbf{w}_1$ belonging to the highest eigenvalue
$\Delta\mathbf{w}_1 = \mathbf{S}_B\mathbf{w}_1 - f(\mathbf{w})\mathbf{S}_{\mathbf{w}}\mathbf{w}_1$
End
**Step 3:** Deflate $\mathbf{S}_B$ and $\mathbf{S}_w$ using
$\mathbf{S}_w^* = \mathbf{S}_w - \lambda\mathbf{w}\mathbf{S}_w\mathbf{w}^T$
$\mathbf{S}_B^* = \mathbf{S}_B - \lambda\mathbf{w}\mathbf{S}_B\mathbf{w}^T$
and then calculate the updated eigenspace for $\mathbf{w}_2$ belonging to second highest eigenvalue up to N eigenvectors repeating the same **Step 2**
```

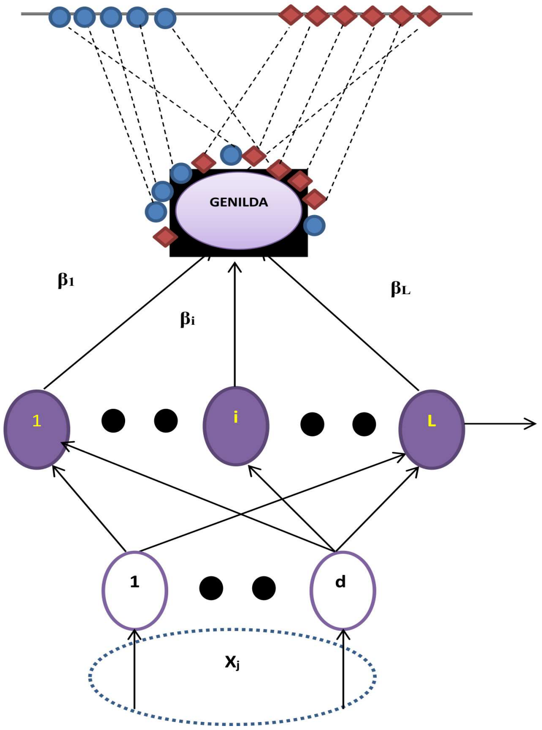
$$\mathbf{S}_w^* = \mathbf{S}_w - \lambda\mathbf{w}\mathbf{S}_w\mathbf{w}^T \tag{3.9}$$

$$\mathbf{S}_B^* = \mathbf{S}_B - \lambda\mathbf{w}\mathbf{S}_B\mathbf{w}^T \tag{3.10}$$

In this way N eigenvectors can be calculated corresponding to their eigenvalues while attaining maximum to minimum variance of data.

Based on the above discussion, the GENILDA algorithm is summarized as Algorithm 1 in Table 3.1.

The idea of combining ELM with the newly proposed GENILDA method is inspired from the contribution of ELM with previous machine learning supervised



L Random Hidden Neurons. Weights a_i, b_i from input to hidden neurons are generated randomly.

Figure 3.1: Architecture of GENILDA Algorithm

and unsupervised mechanisms [152, 153, 154, 155]. The generic algorithmic architecture of GENILDA-ELM is shown in Figure 3.1. The first two layers in Figure 3.1 belongs to ELM followed by GENILDA used instead of the output layer of ELM as a subspace calculator and incremental discriminator.

The purpose of combining the proposed techniques with ELM is to attain maximum computational advantage from the random feature mapping of ELM. The random expansion approach is much faster in comparison with the simple dot product, radial basis function (RBF) mapping or any other kernel-based feature mapping technique. In [152], an extreme support vector machine (ESVM) is proposed by combining ELM and proximal SVM. The ESVM algorithm has proven to be more accurate than the basic ELM model, and much more efficient because as there is no kernel matrix multiplication in ESVM. In [153], the traditional RBF kernel is replaced by the ELM kernel, leading to an efficient algorithm with matched accuracy of SVM. In [154] and [155], the manifold regularization framework is introduced into the ELM model to control both labeled and unlabeled data, extending ELM for semi-supervised learning. However the only constraint is that both methods are limited to binary classification problems, while not exploring the full power of ELM.

Previously researchers from various fields have contributed to ELM theory and applications. In [156] the authors have used ELM in attempting to solve complex multi-classification problems. The study of standard versions of SVM and ELM is conducted in [157] and focusing on their Vapnik-Chervonenkis (VC) dimension and performance using different training samples. The results show that ELM can generalize significantly in comparison with the standard SVM for large samples. ELM has been improved in various contexts to solve for ELM from online sequential data [158], noisy or missing data [159], imbalanced data [160] etc.

Finally, the GENILDA-ELM algorithm is summarized as Algorithm 2 in Table (3.2).

Table 3.2: Algorithm 2 The GENILDA-ELM Algorithm

```
Input:
The labeled patterns, $(\mathbf{X}_N, \mathbf{Y}_N) = (\mathbf{x}_i, \mathbf{y}_i)$ where $i = 1, 2, 3, ..., N$
Output:
The mapping function of GENILDA: $f$:$\Re^{n_i} - > \Re^{n_o}$
Step 1: Map the input data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$ to a random feature space
$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, ..., \mathbf{z}_N\}$ using
$\mathbf{Z} = \sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g(\mathbf{w}_i.\mathbf{x}_j + b_i)$
Step 2: Calculate the (updated if any) or new $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ for binomial classes where
$\boldsymbol{\mu}_1 = \frac{\sum_{i=1}^{n_1} \mathbf{x_i}}{n_1}$, $n_1$ $\epsilon$ data points in class 1 and $\boldsymbol{\mu}_2 = \frac{\sum_{i=1}^{n_2} \mathbf{x_i}}{n_2}$, $n_2$ $\epsilon$ data points in class 2.
if multinomial calculate the updated (if any) or new $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, ..., \boldsymbol{\mu}_N$ and global mean $\boldsymbol{\mu}$
of the whole data.
Step 3:
For $i = 1$ to N iterations
    Step 3.1: Calculate $\mathbf{S}_B$ where $\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$
    for multinomial $\mathbf{S}_B = \sum_{j=1}^{k}((\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T)$
    Step 3.2: Calculate $\mathbf{S}_w$ where $\mathbf{S}_w = \sum_{j=1}^{2}(\mathbf{z}_i - \boldsymbol{\mu}_j)(\mathbf{z}_i - \boldsymbol{\mu}_j)^T$
    For multinomial $\mathbf{S}_w = \sum_{j=1}^{k}((\mathbf{z}_i - \boldsymbol{\mu}_j)(\mathbf{z}_i - \boldsymbol{\mu}_j)^T)$
    Step 3.3: Calculate the updated (if any) or new eigenspace of $\mathbf{w}_1$ belonging to the highest
    eigenvalue
    $\Delta\mathbf{w}_1 = \mathbf{S}_B\mathbf{w}_1 - f(\mathbf{w})\mathbf{S}_\mathbf{w}\mathbf{w}_1$
End
Step 4: Deflate $\mathbf{S}_B$ and $\mathbf{S}_w$ using
$\mathbf{S}_w^* = \mathbf{S}_w - \lambda\mathbf{w}\mathbf{S}_w\mathbf{w}^T$
$\mathbf{S}_B^* = \mathbf{S}_B - \lambda\mathbf{w}\mathbf{S}_B\mathbf{w}^T$
and then calculate the updated eigenspace for $\mathbf{w}_2$ belonging to second highest eigenvalue
up to N eigenvectors repeating Step 3
```

3.2 Results and Discussion

In this section, the classification accuracy of the proposed methods has been compared with incremental PCA [98] and batch LDA [161] method. A detailed experimental study was conducted using datasets with many classes and small-dimensional features including datasets with many classes and large dimensional features. The main interest has been the evaluation of the discriminability of the proposed methods in comparison with standard techniques. In every experiment, the learning rate α is empirically set between 0.0001 and 0.00001, the total iterations for learning are 10,000 and the best selection of hidden neurons is always considered for the GENILDA and GENILDA-ELM methods.

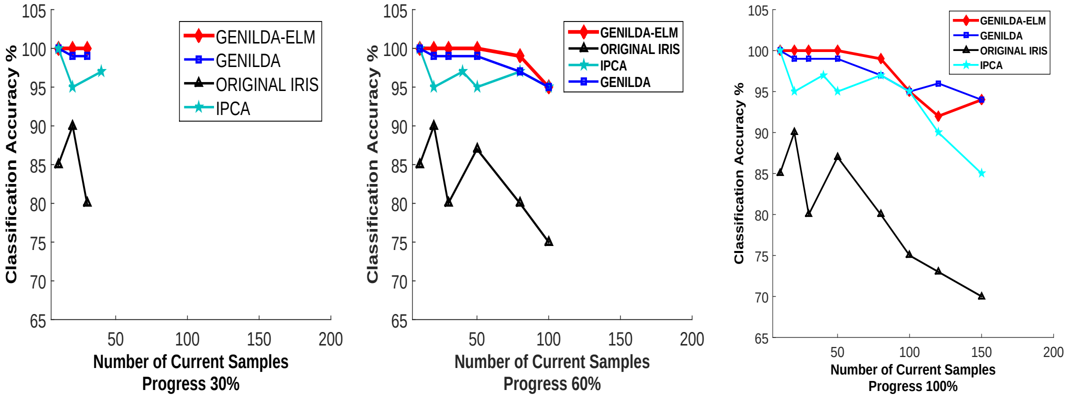


Figure 3.2: The variation of classification accuracy

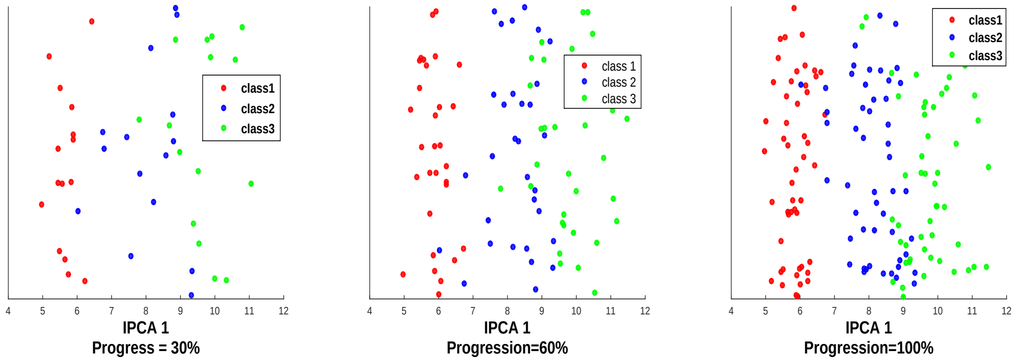


Figure 3.3: IPCA projection of Iris data onto the highest eigenvector

3.2.1 Experimental Setup

In all experiments, an initial feature space (eigenspace) is constructed using 15% of the samples, in order to ensure that at least two data classes are included according to the definition of equation (3.1).

The remaining training data are enumerated point by point or in the form of chunks without any consideration about the chunk size and the number of classes in each chunk. The remaining data is adaptively learned by drawing samples from the data one by one, or in chunk form by using GENILDA computation, as

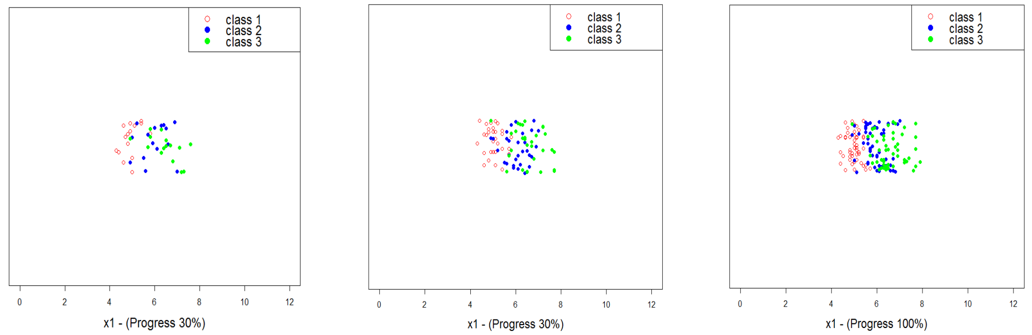


Figure 3.4: the original distribution of Iris data by the first dimension

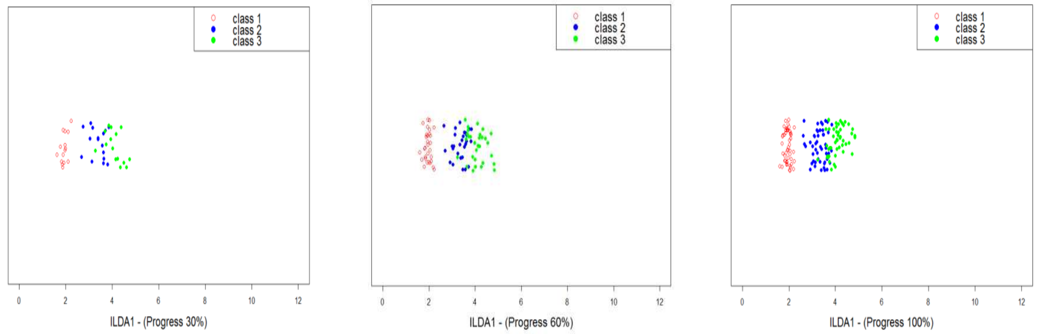


Figure 3.5: GENILDA projection of the Iris data onto the highest eigenvector

in Section 4.1, and the Incremental PCA computation described in [98].

Finally to test the efficiency of GENILDA and GENILDA-ELM for classification accuracy in comparison with the IPCA method, the features were encoded by projecting data previously presented to the updated discriminant eigenspace and then classifying the feature data using KNN classifiers (K=1). The classification accuracy is measured under a leave-one-out cross-validation policy. For each dataset the eigenvectors are ranked by their energy and by selecting a set of top energy eigenvectors.

Since the data in the experiment shown in Figures 3.2, 3.3, 3.5 and 3.6 is

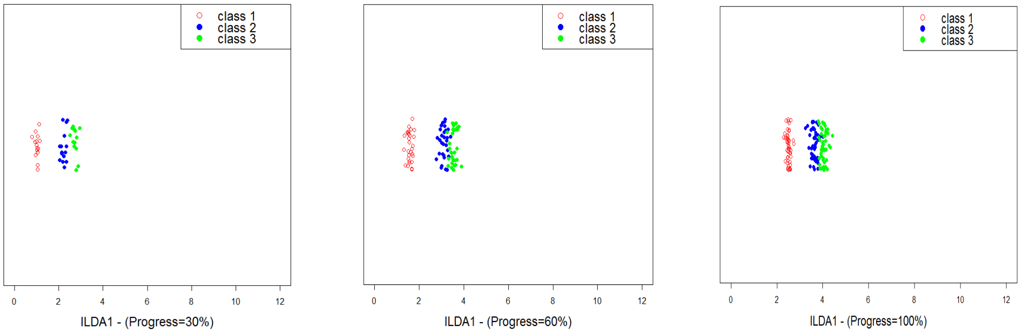


Figure 3.6: GENILDA-ELM projection of the Iris data on the highest eigenvector.

being input in the form of incremental learning, which does not occur in regular time intervals, the term learning stage is used instead of the usual time scale to define the percentage of samples presented so far at the current stage to measure the progress of incremental learning.

3.2.2 UCI Datasets

The class discriminability of the proposed methods is presented by conducting experiments on the database. This procedure consists of eight standard datasets selected from the UCI machine learning repository [1], where each dataset has 100% of continuous/integer values and no missing value. Table 3.3 shows that every dataset has no more than 12 classes and 60 dimension features.

The incremental learning of the proposed methods has been described in Section 4.1 with the existing standard incremental and batch versions proposed specifically for discrimination and best representation of classes on a low dimensional space.

To illustrate this procedure, Figures 3.2, 3.3, 3.4, 3.5 and 3.6 show a time course of the incremental learning over the Iris dataset. Developed by dividing the overall learning of the whole dataset into three learning stages. Whose progress of incremental learning are at 30 %, 60 % and 100 %. In Figure 3.2, shows the variation of classification accuracy and Figure 3.3, 3.5 and 3.6 are IPCA, GENILDA and GENILDA-ELM projections whereas Figure 3.4 is the distribution of iris data by the first single dimension.

As shown in Figure 3.6 the best representation of classes on a low dimensional space is produced by the proposed GENILDA-ELM method. All the sub figures in Figure 3.6 from left to right clearly shows the discrimination between points belonging to red class with other classes, whereas there is a very close discrimination noticed between blue and green class. This closeness between data points belonging to the blue and green class is similar to projections produced by other algorithms shown in Figure 3.2, 3.3 and 3.5. The only reason is the highest non-separability between data points belonging to both green and blue class in the high dimensional space. Still the GENILDA-ELM method best separate the data points belonging to both green and blue classes on a low dimensional space in comparison to the other state of the art approach shown in Figure 3.2, 3.3 , and 3.5.

GENILDA and GENILDA-ELM are compared with IPCA and feature distribution using the original data as a reference. Results clearly shows that the GENILDA-ELM outperformed both GENILDA and IPCA. The superiority of classification can also be deduced from the discrimination difference between corresponding GENILDA-ELM, GENILDA and IPCA projections in Figure 3.3, 3.5 and 3.6.

Table 3.3: Evaluated UCI Datasets

Name	**Input dim.**	**Class**	**Train data**	**Test data**
Iris	4	3	150	-
Liver-disorder	6	2	345	-
Vehicle	18	4	846	-
Glass	10	7	214	-
Wine	13	3	178	-
Segmentation	19	7	210	21000
Vowel	10	11	528	462
Sonar	60	2	208	-

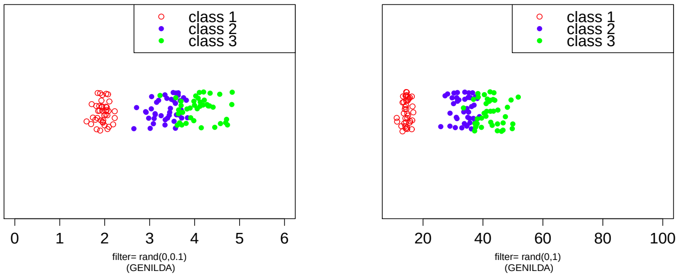


Figure 3.7: Projections by two different initialization of subspace filter (GENILDA)

Further Figure 3.2 shows the maximum reduction of within-class distance, as can be seen clearly with GENILDA-ELM projections in Figure 3.6.

Note that the change in the initial value of the subspace vector changes the magnitude of the resulting projections but makes no change in the discrimination of the data instances as shown in Figure 3.7. The magnitude of the left and right subfigure in Figure 3.6 are different due to the difference of random initialization of the subspace vector. This shows the stability of the discriminant model beginning from any random position and ending at a stable equilibrium position.

Table 3.4 presents the comparative results of GENILDA-ELM, ILDA and IPCA on the classification at the final incremental learning stage for eight UCL datasets, where the number of eigenvectors (denoted as No. of Eig.) specifies the

Table 3.4: Comparative Results of GENILDA, GENILDA-ELM, Batch LDA and IPCA on the classification at the final incremental learning stage for 8 UCL datasets

Name	**No. of Eig.**	**GENILDA**	**GENILDA-ELM**	**IPCA**	**Batch LDA**
Iris	2	98.0	**99.2**	93.3	98.0
Liver-disorder	3	**63.1**	**66.4**	58.8	62.6
Vehicle	9	**73.3**	**75.3**	57.8	75.4
Glass	6	**99.0**	**98.1**	87.6	96.6
Wine	7	**94.3**	**96.7**	87.6	96.6
Segmentation	6	**81.7**	**92.3**	81.4	83.9
Vowel	10	**59.7**	**61.3**	57.9	59.8
Sonar	6	**88.41**	**91.2**	73.5	81.2

dimension of LDA and IPCA eigenfeatures used in classification.

Table 3.4 above reveals that the proposed GENILDA method produced very similar results to the existing batch version of LDA. The combination of the proposed method with ELM outperforms the batch LDA and IPCA methods in almost every experiment on the selected 8 UCL datasets. The same number of eigenvectors were used for each approach to ensure stability. There is a very slight difference of accuracy noticed in vehicle data set in Table 3.4 between GENILDA-ELM and Batch LDA algorithm. This time Batch LDA came out slightly better than the proposed approaches. The only reason noted in this case is the unique characteristics of this dataset and its overall orientation in the high dimensional space which does not work best for the proposed algorithms compared to the standard state of the art Batch LDA algorithm. The classification accuracy of IPCA is lower than that of GENILDA and GENILDA-ELM. This suggests that the discriminant ability of GENILDA is almost equivalent to that of LDA and better than IPCA.

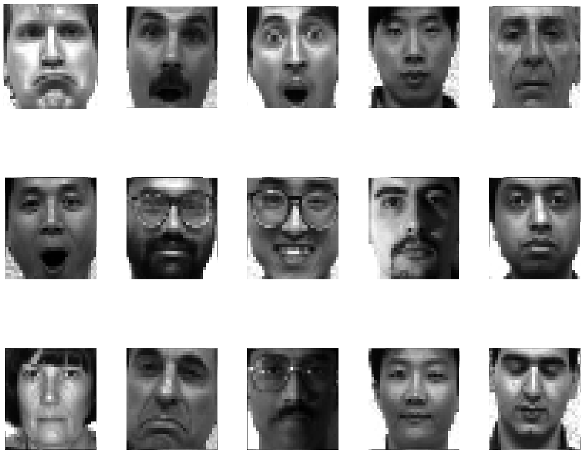


Figure 3.8: Yale Dataset

3.2.3 Face Dataset

The performance of the proposed GENILDA and GENILDA-ELM methods are compared with the incremental PCA method. A benchmark Yale database [162] is used consisting of high-dimensional features. The database consists of 165 grayscale each 1024 (32×32) dimensional images in GIF format of 15 individuals. Firstly the proposed algorithms compared with the standard techniques are tested on 15 different face images by considering a single pair of each person. The total length of the data are $30(15 \times 2)$ and the dimensions are 1024 each.

The projections produced by GENILDA, GENILDA-ELM and IPCA are shown in Figure 3.9, 3.10 and 3.11.

The class discriminability on the face dataset is shown by demonstrating the two dimensional projections produced after training on the first 15 pairs of faces each 1024 dimensional by running GENILDA, GENILDA-ELM compared with

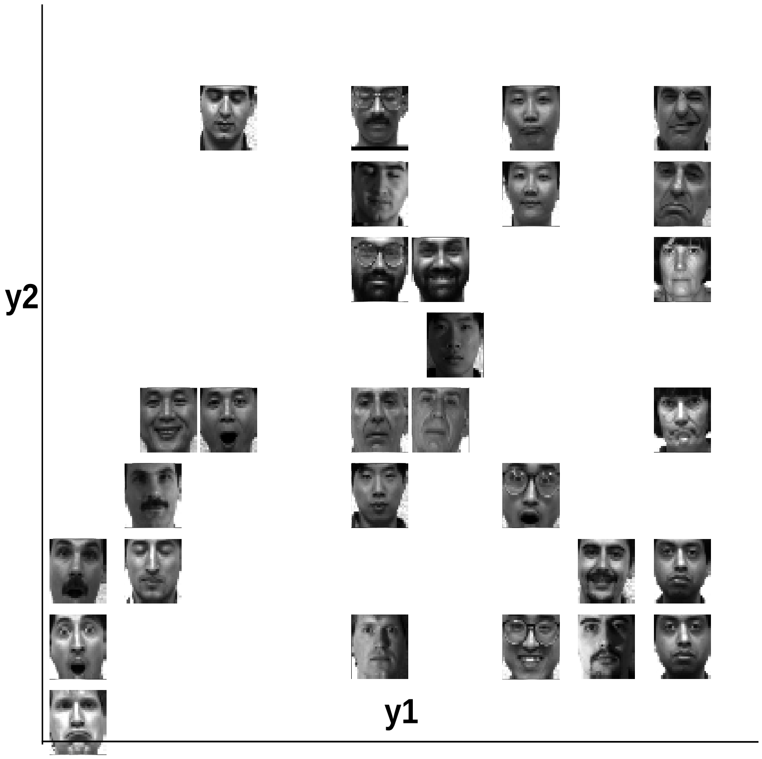


Figure 3.9: Two dimensional projections of GENILDA method

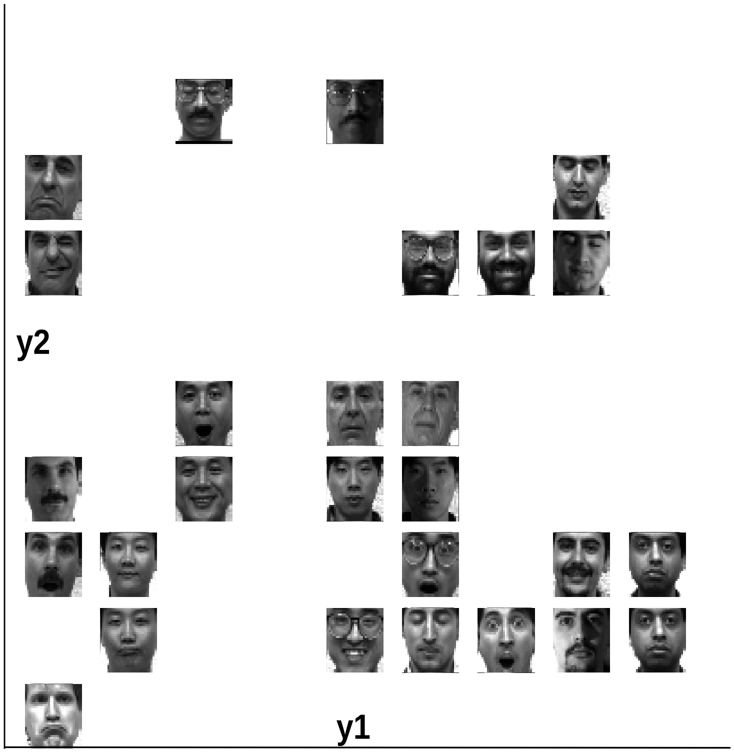


Figure 3.10: Two dimensional projections of GENILDA-ELM method

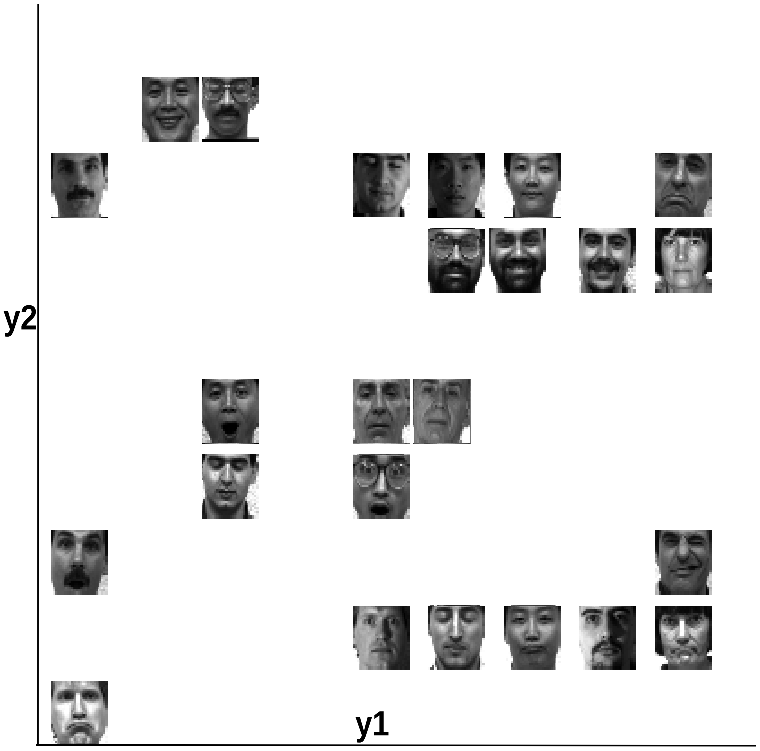


Figure 3.11: Two dimensional projections of IPCA method

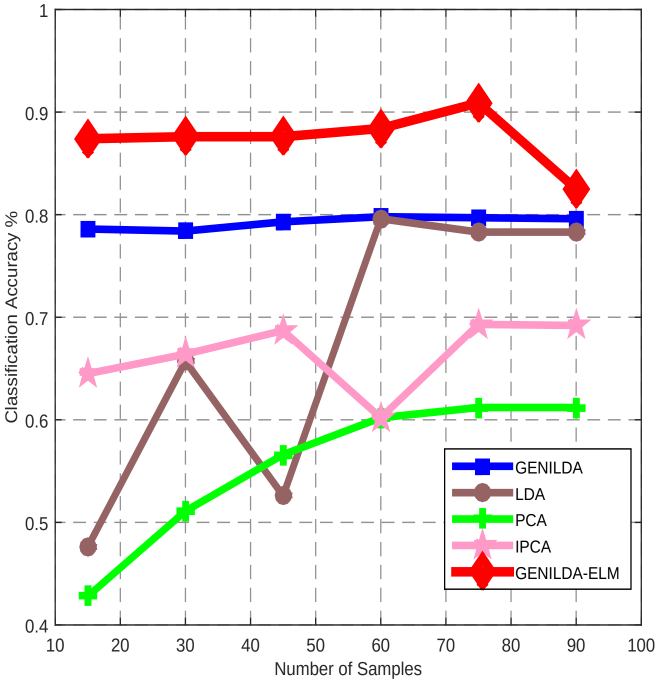


Figure 3.12: Variation of classification accuracy by GENILDA-ELM, GENILDA, LDA, PCA and IPCA with the increase of new data addition.

Table 3.5: Comparative Results of GENILDA, GENILDA-ELM, Batch LDA and IPCA on the classification at final incremental learning stage on Yale face dataset

		2-Train		3-Train		4-Train		5-Train	
Algo	Classifier	Acc.	Dim	Acc.	Dim	Acc.	Dim	Acc.	Dim
PCA	KNN	42.8	29	51.1	44	56.6	57	60.2	73
LDA	KNN	47.6	10	65.7	14	52.6	58	79.6	14
GENILDA	**KNN**	**75.6**	**10**	**76.4**	**14**	**76.9**	**58**	**78.8**	**14**
I-PCA	KNN	64.5	10	66.4	14	68.7	58	69.3	14
GENILDA-ELM	**KNN**	**87.6**	**10**	**90.9**	**14**	**89.4**	**58**	**82.5**	**14**

the existing IPCA method. Figures 3.9, 3.10 and 3.11 show that the projections produced by the GENILDA-ELM method have projected similar faces most of the time very close to each other in the reduced dimensions followed by the GENILDA method where the projections have some of the pairs projected far but most of time the pairs are projected very close. The most miserable performance came out to be of IPCA shown in Figure 3.11 where most of the time different faces are projected very close to each other and similar faces are projected very far.

Further, it can be noticed from the two dimensional projection produced by the proposed methods and standard state of the art approach on faces dataset in Figure 3.9, 3.10 and 3.11; that the projected $\mathbf{y}_1$ and $\mathbf{y}_2$ values of some of the faces came out exactly similar in all the methods which resulted in the projection of those faces on exactly similar position in a lower dimensional space. Therefore, those images cannot be seen properly in Figure 3.9, 3.10 and 3.11. Still the improvement in terms of difference in projections produced by the proposed method compared with the standard state of the art approaches is very clear and can easily be seen by a naked eye.

To show the improvement and precise calculation of the classification accuracy, the K-Nearest neighborhood algorithm is again ran by considering $k = 1$ on the discriminatory subspace produced by all the methods. Further the dimensions are increased from 2 by ranking the eigenvectors with their energies, and by selecting the set of top energy eigenvectors. Here too the classification accuracy is measured by the leave-one out cross validation policy. As it can seen in Table 3.5 the classification accuracy of the proposed methods is higher for GENILDA-ELM method compared with existing batch and incremental versions. After changing the number of eigenvectors, there is not a significant improvement of the classification accuracy. In Table 3.5 2-train refers to 1 pair totaling 30 images because the total length of unique images is 15. The number of training images is increased from 2 to 3 similar images as well increasing both the total from 30 to 45 as 3-train in Table 3.5 and the dimensions too but still the proposed techniques has produced maximum classification accuracy as shown Figure 3.12. Increasing to 4-train and 5-train improved results and outperformed the standard existing approaches, both incremental and batch.

Further the proposed method's classification accuracy is compared with IPCA by sequentially increasing the number of eigenvectors as shown in Figure 3.13.

It can be clearly seen from Figure 3.13 that GENILDA-ELM has a very slight variation in its classification accuracy by a gradual increase in the number of eigenvectors. When the count of eigenvectors reaches between 70 to 80 in Figure 3.13, the classification accuracy of GENILDA and IPCA almost becomes equal however the accuracy of GENILDA is consistent with slight changes due to the increase in the number of eigenvectors including a slight variation in the end.

The execution time of the proposed algorithms as shown in Figure 3.14 were

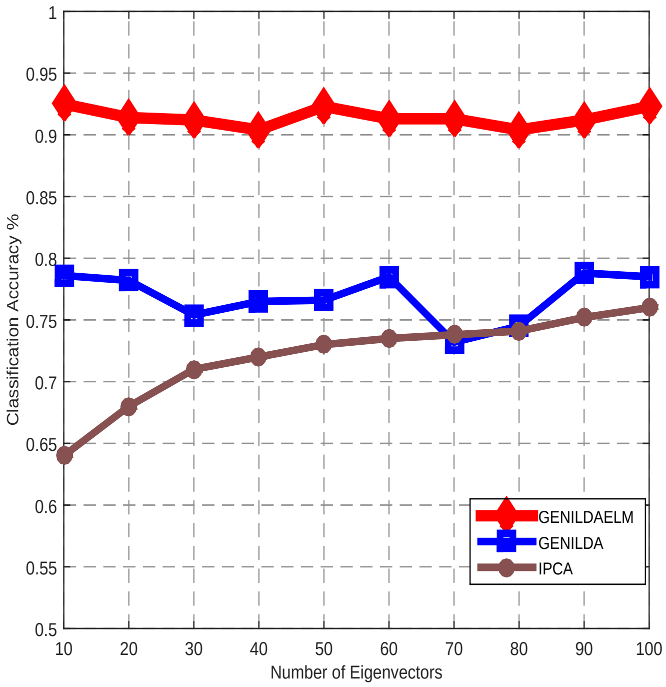


Figure 3.13: Variation of GENILDA, GENILDA-ELM versus IPCA on the final classification accuracy under different number of eigenfeature dimensions

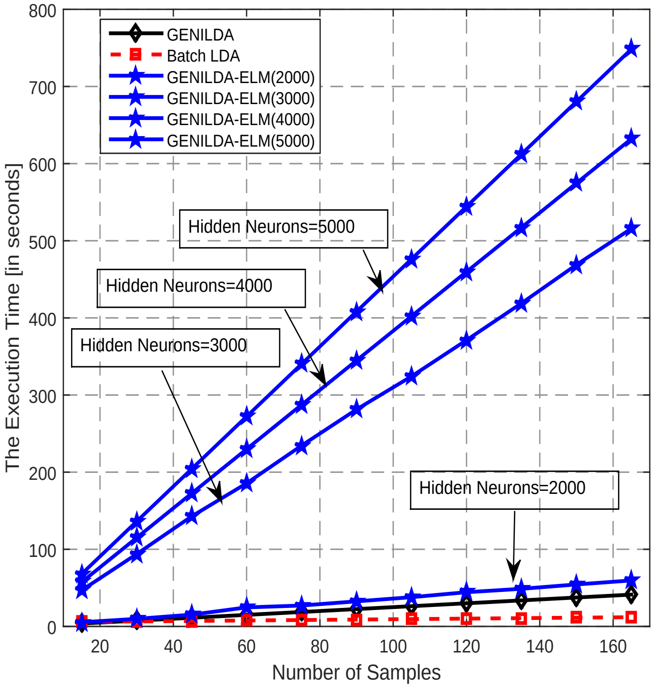


Figure 3.14: Variation of Time Taken By GENILDA and GENILDA-ELM to calculate discriminant eigenspace in batch mode

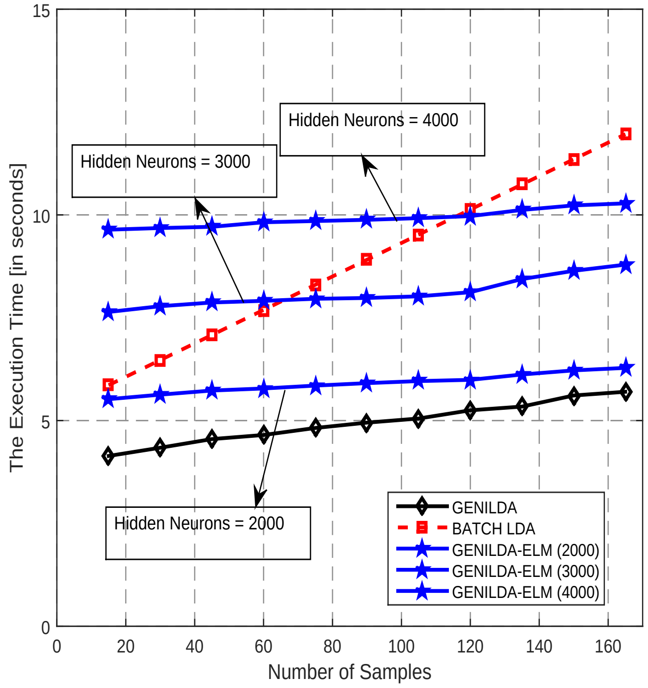


Figure 3.15: Variation of Time Taken By GENILDA and GENILDA-ELM to update the discriminant eigenspace with the increase of new data addition

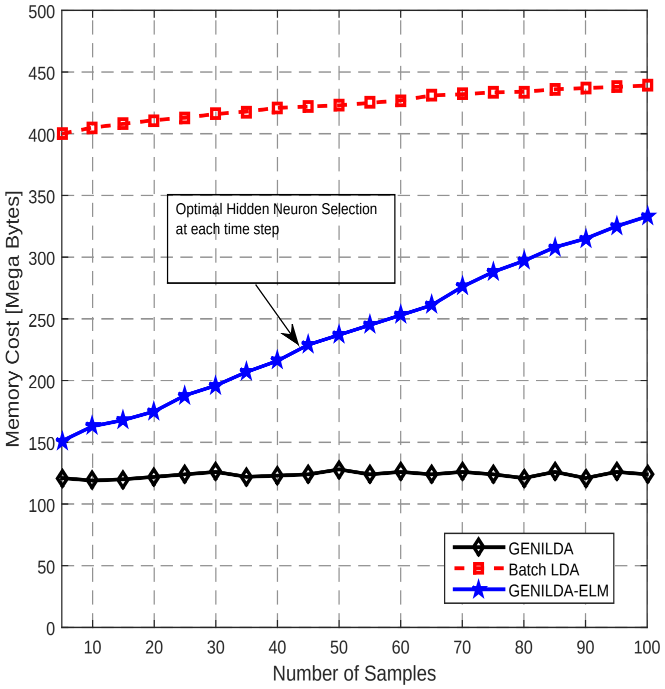


Figure 3.16: Variation of GENILDA and GENILDA-ELM on memory cost with the increase of data addition

higher each time the whole data samples were considered for learning. Batch LDA outperformed the proposed GENILDA and GENILDA-ELM algorithm when repetitively all the samples were learned. On the contrary, calculating the execution time in Figure 3.15 by breaking the samples into more than one random chunks and updating the discriminant eigenspace each time on the addition of each new chunk resulted the proposed algorithms significantly better compared to the batch LDA algorithm which is incapable of learning the data in more than one chunks.

Similarly, as shown in Figure 3.16 the memory cost of the proposed GENILDA algorithm was very low as compared to the batch LDA algorithm. This is due to the domain independent nature of the proposed algorithm. At each time stamp the adaptive input is learned independently without updating the scatter matrices and only updating the mean of each class. GENILDA-ELM memory consumption came out higher due to the random expansion of adaptive input for linear transformation at each time stamp. To reduce the memory cost the optimal number of hidden neurons should be considered at each time stamp.

3.3 Conclusions

This chapter presented an online generalized eigenvalue based LDA with ELM. This method has produced the most improved projections and classification results as compared to the standard batch and incremental versions of LDA and PCA. The conclusive properties of GENILDA and GENILDA-ELM are summarized as follows: 1) GENILDA has an equivalent power to batch LDA in terms of discriminability whereas GENILDA with ELM is significantly better than LDA;

2) GENILDA-ELM is remarkably effective for handling bursts of new classes coming in at different times; 3) As compared with IPCA, GENILDA-ELM is usually, but not guaranteed [163] to be superior to IPCA classification.

The only limitation of the proposed GENILDA-ELM methods is the high computational efficiency required for larger chunks due to their purely incremental point by point learning behaviour. Increasing the number of hidden neurons for randomized expansion using ELM algorithmic criteria also increases the computational complexity of the proposed algorithm. This problem can be resolved by dividing the larger chunks into smaller pieces and performing incremental learning using the proposed techniques. Nevertheless, GENILDA-ELM is the most useful method in situations where data is arriving point by point, or in the form of adaptive streams. This feature is highlighted especially in scenarios calling for classification in a fast and lightweight manner.

The next chapter describes the implementation of the biologically inspired online versions of slow feature analysis technique. The prime objective is to derive point by point learning mechanisms which can extract invariant features in an online manner.

Chapter 4

Novel Online Extensions of Invariant Feature Extraction

This chapter presents novel learning approaches for extracting invariant features from time series. Firstly incremental versions are presented which can extract invariant features by using a one-pass incremental learning criteria without leaving any dependency on the previously learned data. The incremental versions are derived in two ways 1) By using L. Wiskott criteria [2] and 2) By using James Stone's criteria [96]. Secondly echo state network is used as a preprocessor with the incremental learner. Thirdly higher-order derivatives are used as a smoothness constraint while extracting invariant features to see its effect on the overall output signal. Finally all the proposed methods are compared against standard state-of-the art, using datasets comprising artificial, MNIST digits and hand-written character trajectories.

4.1 The GENEIGSFA Using L. Wiskott Criteria (An Incremental Method For Generalized Eigenproblems)

The differential equation in [55] is applied to extract invariant feature by finding the maximum eigenvalue of:

$$E(\mathbf{x}\mathbf{x}^T)\mathbf{w} = \lambda' E\left((\frac{d\mathbf{x}}{dt})(\frac{d\mathbf{x}}{dt})^T \right) \mathbf{w}, \tag{4.1}$$

where λ' is the inverse of eigenvalue λ. This gives us an update of

$$\Delta\mathbf{w} = \Sigma_x \mathbf{w} - f(\mathbf{w})\Sigma_{\dot{x}}\mathbf{w}, \tag{4.2}$$

where $\Sigma_x = E(\mathbf{x}\mathbf{x}^T)$ and $\Sigma_{\dot{x}} = E\left((\frac{d\mathbf{x}}{dt})(\frac{d\mathbf{x}}{dt})^T \right)$.

In fact for a truly neural-based solution i.e. updating the weights in an online mode, the covariance matrices are replaced in (4.2) with instantaneous values so that

$$\Delta\mathbf{w} = \mathbf{x}_i\mathbf{x}_i^T\mathbf{w} - f(\mathbf{w})(\frac{d\mathbf{x}_i}{dt})(\frac{d\mathbf{x}_i}{dt})^T\mathbf{w}. \tag{4.3}$$

This method is termed as the Generalized Eigenvalue Slow Feature Analysis method (GENEIGSFA).

Echo state network is also used as preprocessor with this method by foregoing its output state. Reservoir activations are used as the data values before using the technique of slow feature analysis to update the reservoir to output weights so that the slow features of the time series can be identified.

4.2 The Incremental Invariant Feature Extraction Method Using J. Stone's Criterion (An Incremental Method For Generalized Eigenproblems)

An early attempt [96] to extract invariances from visual data uses a slightly different criterion: Stone argued that what is necessary is the extraction from visual signals of that part of a signal which changed least. Of course an easy transformation is to map all input signals to a constant value, such as, 0. However this procedure will present no information about its environment at all. To ensure that there is some variance in the output Stone suggested that the criterion which he wished to maximise was the ratio between the long term variance, V and the short term variance, U:

$$J = \log \frac{V}{U} = \log \frac{\sum_{t=1}^{T} (\tilde{z}_t - z_t)^2}{\sum_{t=1}^{T} (\bar{z}_t - z_t)^2}, \tag{4.4}$$

where $\tilde{z}$ is an estimate of the long term mean and $\bar{z}$ is an estimate of the local, short term mean. Both estimates are calculated by moving averages with appropriate smoothing parameters.

Intuitively, Stone suggested that the output signal should contain as much variance as possible overall, but as little variance as possible over the short term.

According to Stone [96] if the variance of the curve is minimal that curve can be considered to be maximally smooth. Another characteristic is the variability over time, which ensures learning of perceptually salient visual parameters using spatiotemporal smoothness constraint. The output can be made to reflect both smoothness and variability by forcing it to have both a small short-term variance, and a large long-term variance. This leads to the smaller variance of the output over small periods, relative to its variance over a longer period. To test this theory, Stone developed a learning method for updating a linear filtering parameter consisting of a mixture of Hebbian and anti-Hebbian learning.

Stone's criterion is used in here but with *exactly the same method as above* i.e. the generalised eigenproblem solver of Section 4.1. The criterion is posed as one of finding the generalised eigenvector of

$$\Sigma_L \mathbf{w} = \lambda \Sigma_S \mathbf{w}, \tag{4.5}$$

where Σ_L and Σ_S are the long-term and short-term estimates of the covariance matrices of the data respectively. In practice, $\Sigma_L = \Sigma$, the covariance matrix of the data set and estimate Σ_S using a standard update rule such as

$$\Sigma_S = (1 - \alpha)\Sigma_S + \alpha \mathbf{z}(t)\mathbf{z}^T(t), \tag{4.6}$$

where $0 < \alpha < 1$ is smoothing parameter.

4.3 Incorporating higher order changes

The higher order smoothness constraints is applied by considering the effect of minimising $E((\frac{d^n\mathbf{x}}{dt^n})^2)$, for $n = 2, 3, ...$ Then this changes (4.2) to

$$E(\mathbf{x}\mathbf{x}^T)\mathbf{w} = \lambda' E\left((\frac{d^2\mathbf{x}}{dt^2})(\frac{d^2\mathbf{x}}{dt^2})^T + (\frac{d\mathbf{x}}{dt})(\frac{d\mathbf{x}}{dt})^T\right)\mathbf{w}, \tag{4.7}$$

which gives us an update of

$$\Delta\mathbf{w} = \Sigma_x\mathbf{w} - f(\mathbf{w})[\Sigma_{\dot{x}} + \Sigma_{\ddot{x}}]\mathbf{w}, \tag{4.8}$$

where $\Sigma_x = E(\mathbf{x}\mathbf{x}^T)$, $\Sigma_{\dot{x}} = E\left((\frac{d\mathbf{x}}{dt})(\frac{d\mathbf{x}}{dt})^T\right)$ and $\Sigma_{\ddot{x}} = E\left((\frac{d^2\mathbf{x}}{dt^2})(\frac{d^2\mathbf{x}}{dt^2})^T\right)$.

As above, for a truly neural based solution i.e. updating the weights in online mode, the covariance matrices are again replaced in (4.8) with the instantaneous values so that

$$\Delta\mathbf{w} = \mathbf{x}_i\mathbf{x}_i^T\mathbf{w} - f(\mathbf{w})[(\frac{d\mathbf{x}_i}{dt})(\frac{d\mathbf{x}_i}{dt})^T + (\frac{d^2\mathbf{x}_i}{dt^2})(\frac{d^2\mathbf{x}_i}{dt^2})^T]\mathbf{w}. \tag{4.9}$$

This makes the proposed algorithm purely incremental in nature as all the data points are now individually learned point by point and perform learning by updating the filter on each iteration. Similarly as shown in (4.3) the GENEIGSFA method is replaced by instantaneous values of the covariance matrix. This approach can be learned independently point by point as shown in (4.8); while simultaneously minimizing its rate of change at each iteration.

Ofcourse, to put more or less emphasis on different terms to obtain

$$\Delta \mathbf{w} = \mathbf{x}_i \mathbf{x}_i^T \mathbf{w} - f(\mathbf{w})[\lambda(\frac{d\mathbf{x}_i}{dt})(\frac{d\mathbf{x}_i}{dt})^T + (1-\lambda)(\frac{d^2\mathbf{x}_i}{dt^2})(\frac{d^2\mathbf{x}_i}{dt^2})^T]\mathbf{w}, \tag{4.10}$$

for some $0 < \lambda < 1$.

In order to maintain biological plausibility, the higher order difference approximation is used to the covariance of the higher order derivatives. Thus

$$\frac{d^2\mathbf{x}_i}{dt^2} \approx (\mathbf{x}_{i+1} - \mathbf{x}_i) - (\mathbf{x}_i - \mathbf{x}_{i-1}). \tag{4.11}$$

4.4 Simulations of GENEIGSFA Using L. Wiskott Criteria

The first simulation uses

$$\mathbf{u}(t) = \sin(t/33) + \cos(t/10 + \mu), \tag{4.12}$$

where $t = 1, ..., 2000$ and $\mu = 3$ is an arbitrarily chosen phase term. μ is taken as nonzero since without this term, each of the two signals may reinforce the other. The dataset consists of 100 samples across 20 dimensions. To emulate the actual perception data and to investigate a method which is biologically feasible the sliding window is not used. (Although the subjective experience is one of continuity, with sacades and discrete spikes, the data is perceived in chunks). $(\frac{d\mathbf{x}_i}{dt})(\frac{d\mathbf{x}_i}{dt})^T$ is estimated as $(\mathbf{x}_i - \mathbf{x}_{i-1})(\mathbf{x}_i - \mathbf{x}_{i-1})^T$, a crude but an effective estimate.

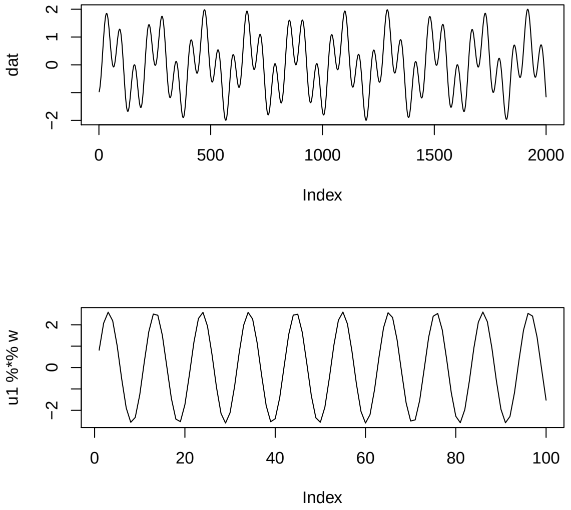


Figure 4.1: Top diagram: the data. Bottom: the filtered data using GENEIGSFA method.

Thus (4.3) becomes

$$\Delta \mathbf{w} = \mathbf{x}_i \mathbf{x}_i^T \mathbf{w} - f(\mathbf{w})(\mathbf{x}_i - \mathbf{x}_{i-1})(\mathbf{x}_i - \mathbf{x}_{i-1})^T \mathbf{w}. \tag{4.13}$$

In the top diagram of Figure 4.1, 2000 samples from (4.12) are shown and the bottom diagram displays the filtered data using the trained values of $\mathbf{w}$. The low frequency is clearly identified. A variety of functions were tested for $f(\mathbf{w})$ and not found to significantly alter the results.

The linear method can only be applied to linear data. Processing (4.12) is a linear combination of the two signals.

When non-linear data is considered drawn from 2000 samples,

$$\mathbf{u}(t) = \sin(t/10)\cos(t/33) + \cos(t/10)\cos(t/33 + \mu). \tag{4.14}$$

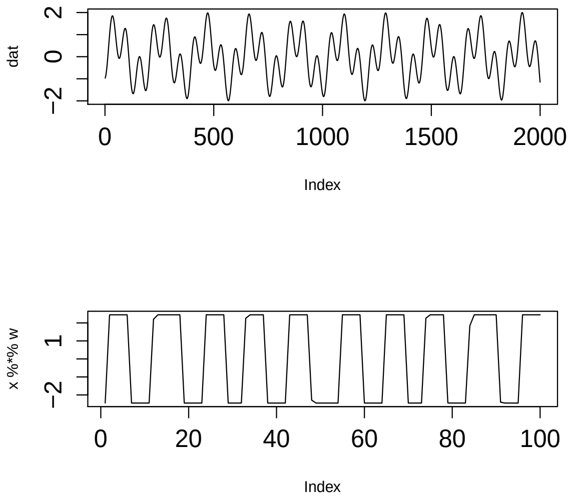


Figure 4.2: Top diagram: the data. Bottom: the filtered data with reservoir as input to the GENEIGSFA method.

where $t = 1, ..., 2000$ and $\mu = 3$ is an arbitrarily chosen phase term. The reservoir activations and their derivatives are used as the input to the SFA method. Results are shown in the bottom halves Figures 4.2, 4.3 and 4.4. It can be noted from the top diagram (the data) that there is some beating apparent in these figures (giving around 3 periods) and it is this that the filtered data identifies. These results are not always achieved: sometimes one or other of the faster signals is identified ($\cos(t/10)$ or $\cos(t/33)$). Although the reservoir was randomly created each time, the filter found changes with each simulation.

The results above were created with a non-sparse reservoir.

With a sparse reservoir a sensible filter was found but the very slowest signal is not identified as shown in Figure 4.5. The reason was due to the sparseness of the weight matrix of the reservoir thus leading to the production of temporally less sequential time series activations; which, when used for extracting invariant

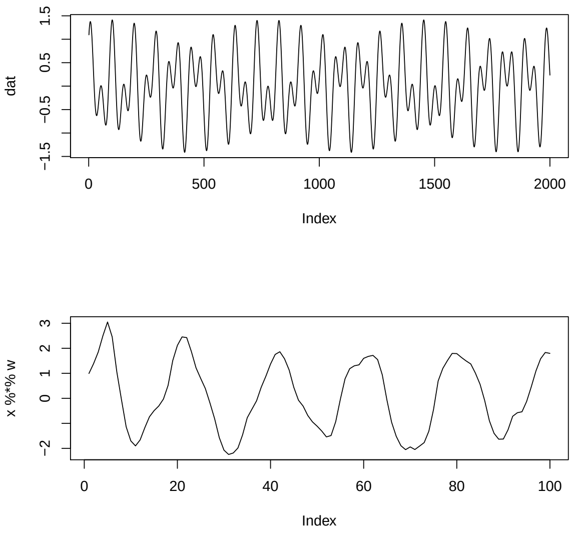


Figure 4.3: Top diagram: the data. Bottom: the filtered data with reservoir as input to the GENEIGSFA method.

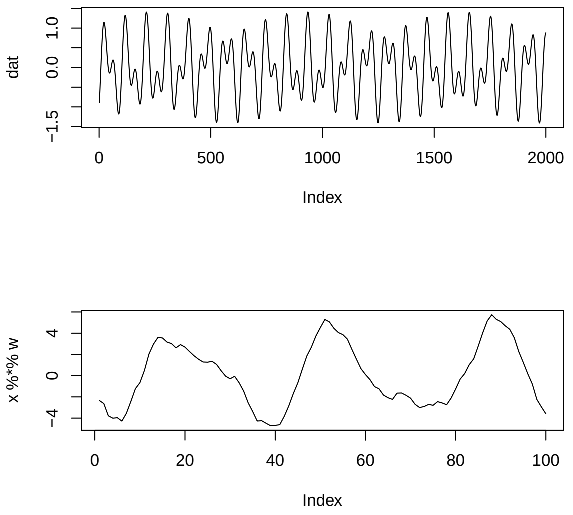


Figure 4.4: Top diagram: the data. Bottom: the filtered data with reservoir as input to the GENEIGSFA method.

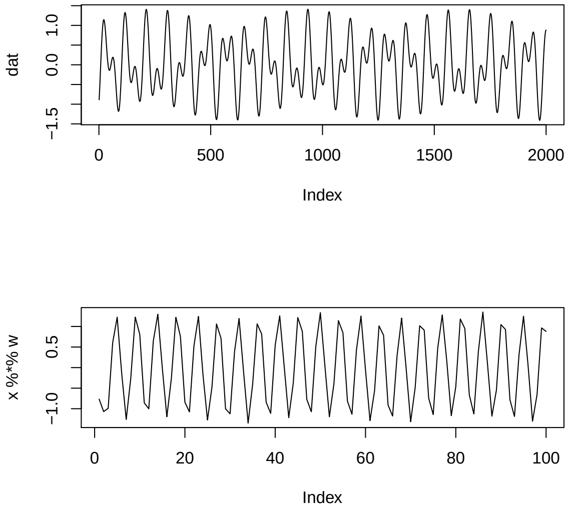


Figure 4.5: Top diagram: the data. Bottom: the filtered data with sparse reservoir as input to the GENEIGSFA method.

features by the proposed algorithm was able to identify the second or third slowest signal from the input dataset.

The experiment was also performed with moving windows so that

$$\begin{aligned}
\mathbf{u}_1 &= (u(1), u(2), \cdots, u(20))^T, && (4.15)\\
\mathbf{u}_2 &= (u(2), u(3,), \cdots, u(21))^T, && (4.16)\\
& \text{etc.}
\end{aligned}$$

but again a smooth filter is found similar to the filter from the sparse reservoir though sometimes a filter corresponding to a less slowly changing signal is found.

Observation reveals that not just any nonlinearity will do as preprocessing before the SFA stage: for example, using a radial basis function shown in Figure

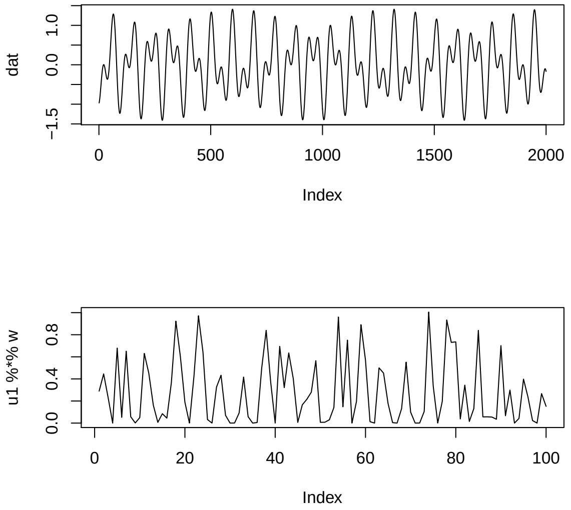


Figure 4.6: Top diagram: the data. Bottom: the filtered data with RBF as input to the GENEIGSFA method.

4.6, failed to identify the slowest structure, found little structured output at all. The reason was due to the non-sequential activations produced by the RBF kernels which, when used for extracting invariant features by the proposed algorithm was unable to identify the slowest signal from the input dataset at all.

4.5 Simulations of Incremental Invariant Feature Extraction Method Using J. Stone's Criteria

The experiment again begins with artificial dataset having linear combination,

$$\mathbf{u}(t) = \sin(t/33) + \cos(t/10 + \mu), \tag{4.17}$$

where $t = 1, ..., 1000$ and $\mu = 3$ is an arbitrarily chosen phase term. We use the sliding windows method so that

$$\begin{aligned}\mathbf{u}_1 &= (u(1), u(2), \cdots, u(100))^T, & (4.18)\\ \mathbf{u}_2 &= (u(2), u(3,), \cdots, u(101))^T. & (4.19)\\ &\text{etc.}\end{aligned}$$

Typical results are shown in Figure 4.7 for which reservoir is used of size $N_x = 120$ and 100000 iterations. The long-term covariance matrix is calculated just once, but the short term covariance matrix is updated using

$$\Sigma_S = \alpha\Sigma_s + (1-\alpha)\mathbf{x}(t)(\mathbf{x}(t))^T, \tag{4.20}$$

with α set to values between 0.7 and 0.95. The results as shown in Figure 4.7 were achieved with $\alpha = 0.85$ that a slowly moving signal has been found but very much more crudely than with the SFA criterion. In general Stone's criterion was found to produce less close approximations to the sinusoid signals than the SFA criterion.

Best results were achieved with a reservoir size approximately equal to the length of each data sample.

However the Stone-reservoir method failed to find the slowest filter with the nonlinear combination data:

$$\mathbf{u}(t) = \sin(t/10)\cos(t/33) + \cos(t/10)\cos(t/33 + \mu), \tag{4.21}$$

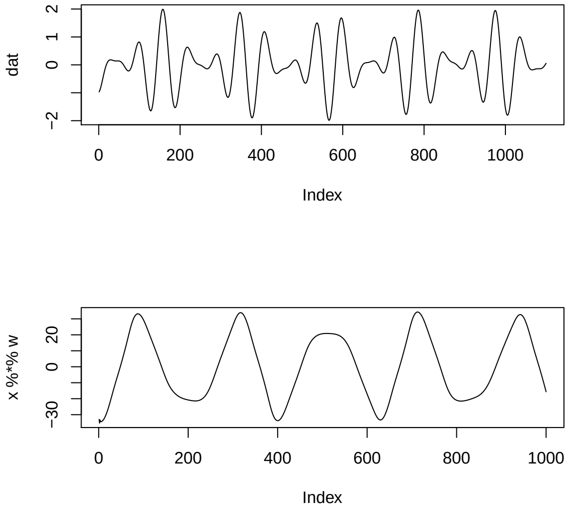


Figure 4.7: Results from a reservoir with Stone's Method in which $N_x = 120$. Best results occurred when the size of the reservoir approximated the length of each data sample.

but it did find the second slowest, Figure 4.8.

In the previous sections, artificially generated data sets are used to illustrate the various methods. In the next sections two real data sets are used 1) MNIST Digit [52] and 2) Character Trajectories Dataset [1].

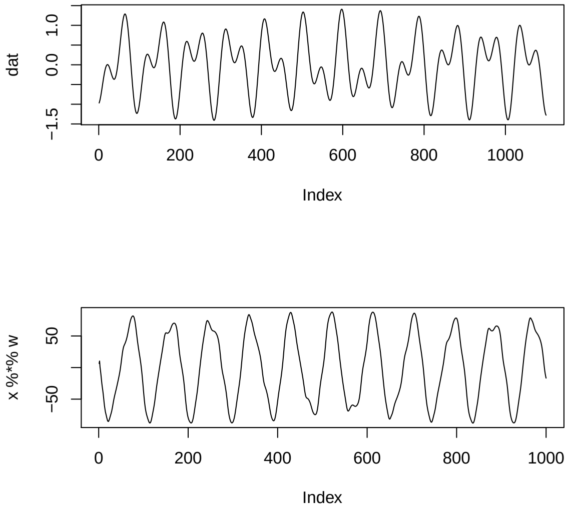


Figure 4.8: Results from a reservoir with Stone's Method in which $N_x = 120$. Best results were when the size of the reservoir approximated the length of each data sample.

4.6 Real Data

4.6.1 MNIST Dataset

The group of methods proposed in this chapter filter invariant features from input data, and are illustrated on MNIST handwritten digit dataset. The MNIST handwritten dataset consists of a standardized and freely available set of 70,000 handwritten digits. Each pattern consists of a handwritten digit of size 28 x 28 pixels. In the literature of SFA (Slow Feature Analysis) [164, 165] this handwritten digit dataset is mostly used for pattern recognition but here this dataset is used to show the effectiveness of the proposed algorithms in identifying change. The digits are read, sorted from 0-9 into ascending order and considered 1000 each. The reason for considering 1000 digits of the same type from each class is to form a sequential time series comprising similar patterns belonging to the same class. First 0's are incrementally learned in the time series, with an equal time interval assigned to the learning of each digit. The data of all 0's are then projected on to the filter obtained from learning 0's, to find the output. To identify the change the output of all 1's are also projected on to the same filter and the differences in the respective outputs are observed. To ensure that the slow features of each digit is learned, 1000 samples of each digit is presented 10 times to the network. This is done for all the digits by first calculating output using their own filter and then using the same filter to calculate the output of next digit. The learning rate is empirically taken as 0.00001 and the iteration for learning is 10 epochs of 1000 samples.

In Figures (4.9, 4.10, 4.11, 4.12, 4.13 and 4.14) show the results of digit 1 and 2. The x-axis in the Figures represent the sample number, and the Y-axis represents the actual output y.

Firstly, Figure 4.9 shows the results by using the standard slow feature analysis algorithm [2]. The change is identified by showing a difference in amplitude of the slow features coming from 2's according to filter of 1's followed by its own filter.

Secondly, Figure 4.10 shows the results of digit 1 and 2 using the GENEIGSFA method. The slow features coming from 2's according to filter of 1's is different in amplitude from the slow features of 1's on 1's filter. This change in amplitude shows the change in digit by the proposed incremental algorithm which is identifying while learning in sequence from 0 to 9.

Thirdly, the proposed incremental algorithm (GENEIGSFA) is tested on the activations produced by echo state network. The size of the reservoir is 100 and the other parameters are the same as above. The input weights and weights of the reservoir are initialized randomly between 0 to 1. Figure 4.11 shows that by using reservoir's activations the difference has become more pronounced, and the projections of digits on their own filter is quite different from the projections of digits on another digit's filter. Clearly, the combination of reservoir and incremental slow feature analysis (GENEIGSFA) is very powerful.

Fourthly, the higher-order derivatives have also been tested with the proposed incremental algorithm (GENEIGSFA) and the results are shown in figure 4.12. The value of λ is taken as 0.5. The increase or decrease in the value of lambda has minimal effect on the amplitude of the output signal. The role of this smoothness constraint is adding smoothness in the output feature is inclusive however the change in moving from one digit to the other is very prominent.

Finally the alternative incremental method was tested based on Stone's approach. In Stone's method the value of α is 0.5. The learning rate and the number of iterations are unchanged. Results are shown in figure 4.13 and 4.14. The prior figure shows the results without using reservoir whereas the latter with reservoir. The change is not really prominent with reservoir in figure 4.14 as compared to without using reservoir in figure 4.13 where the change in amplitude is clearer. The reason is due to the sequentially expanded time series activation of reservoir which did not proved as useful while extracting invariant features.

The minimum and maximum value of the output for digit '1' on its own filter

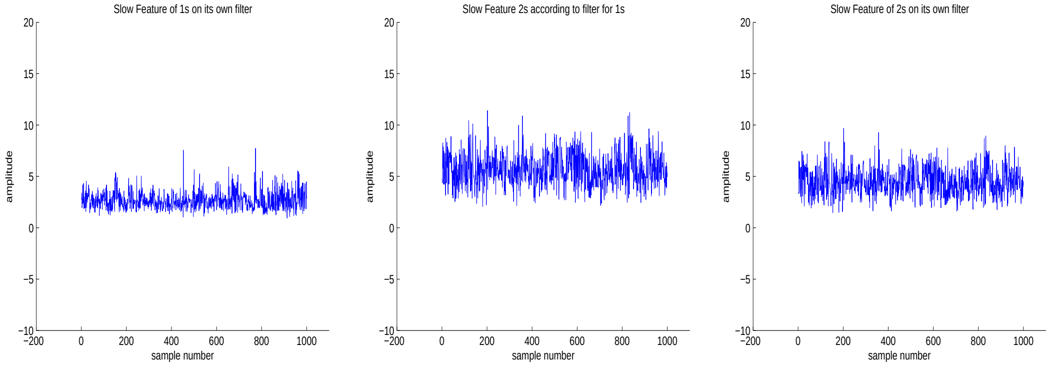


Figure 4.9: Standard SFA [2] Left: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

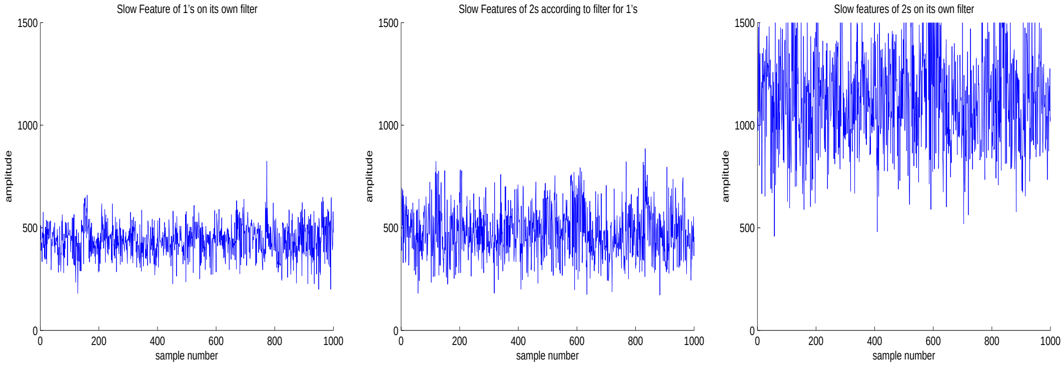


Figure 4.10: Incremental SFA Left: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

and digit '2' on 1's filter for all the techniques are shown in Table **??**. Table **??**. The value shows the dissimilarity between the output of 1 projected on 1's filter and the output for 2 projected on 1's filter by showing their range of output and calculating the correlation coefficient between the output of both signals. Table **??** shows that the correlation between the output of both the signals using a GenEigSfa method with Reservoir is less when compared to other methods. This shows that GenEigSfa method with Reservoir identifies the change more clearly than other methods.

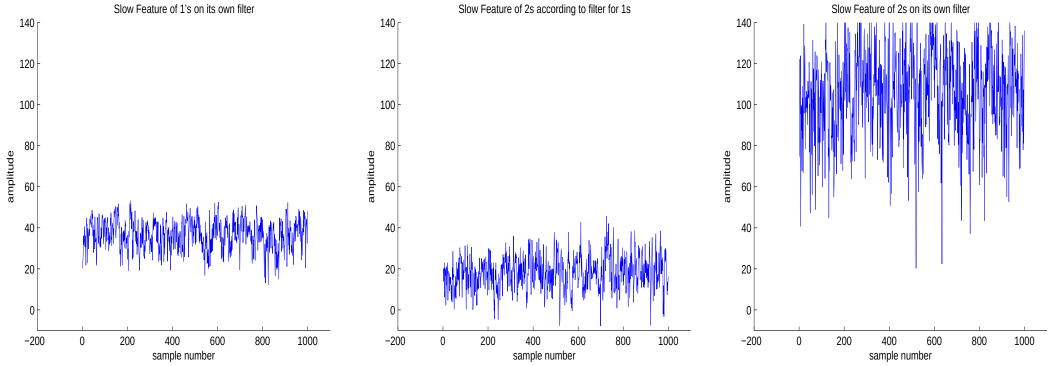


Figure 4.11: Incremental SFA with Reservoir Left: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

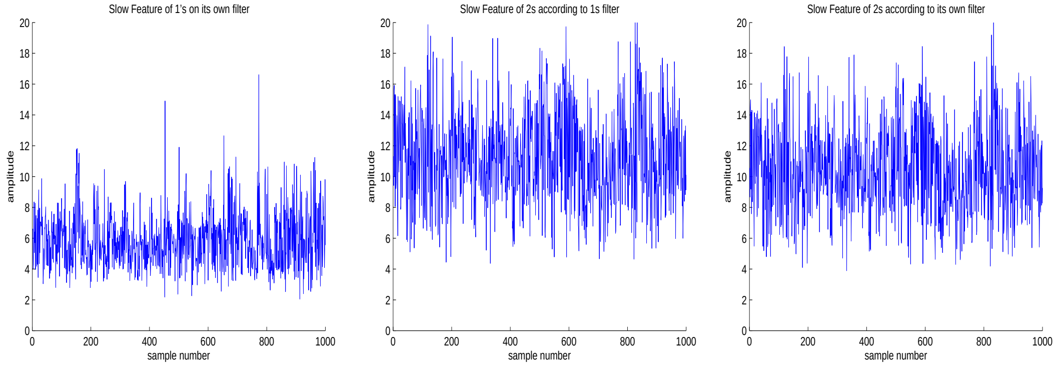


Figure 4.12: Incremental SFA with Smoothness Constraint: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

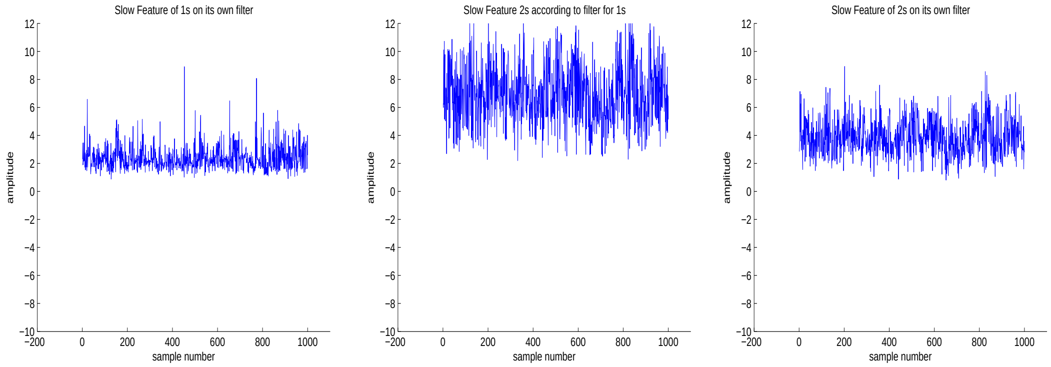


Figure 4.13: Stone's Method: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

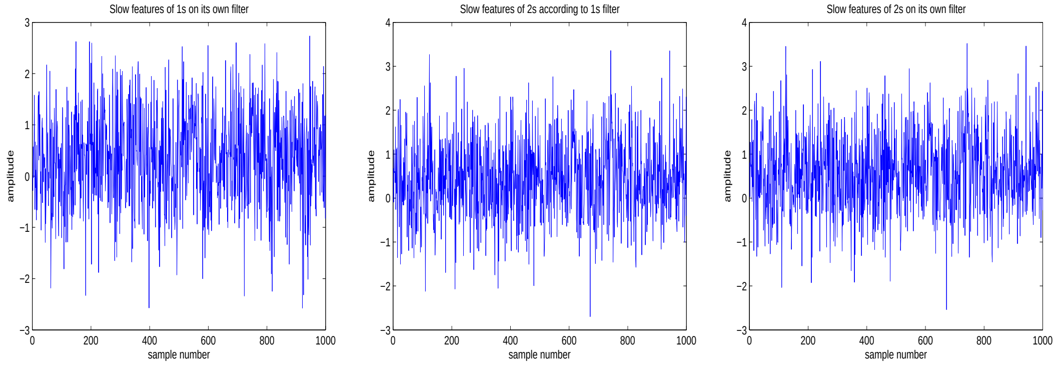


Figure 4.14: Stone's Method with Reservoir: Slow features of 1's on its own filter. Middle: Slow features of 2's according to 1's filter. Slow features of 2's on its own filter.

4.6.2 Character Trajectories

Next the character trajectories dataset [1] are used which consists of 2858 character samples. The categories of characters range from a to z. Each character can have a different number of pixels between 174 to 205 in length with the standard 3 dimensional data giving x, y, and z coordinates. 1000 characters are considered for training and the remaining characters are reserved for testing. Firstly 20 characters are filtered from each category out of 1000 characters. The rationale for extracting 20 characters of the same type is to create a small sequential time series consisting of patterns belonging to same class which can be learned together in a sequential manner. Dedicating equal time interval to each character for learning and extracting a cumulative average filter. Out of each group of 20 characters, the first group of a's are selected and the average filter for all the a's are extracted. The same procedure was done for all remaining groups. After extracting the average filter from each group the accuracy of the proposed methods are tested. A random 'a', 'b', 'c','d' and 'e' character were chosen from

Standard Slow Feature Analysis Algorithm			
	Min	**Max**	**CorCoef(y1y2)**
Range of output of 1s projected on 1s filter (y1)	0.9411	7.7330	**0.9807**
Range of output of 2s projected on 1s filter (y2)	2.0555	11.4144	
Incremental Learning (GENEIGSFA)			
Range of output of 1s projected on 1s filter (y1)	180.1291	820.7507	**0.9890**
Range of output of 2s projected on 1s filter (y2)	172.282	831.8435	
Incremental Learning (GENEIGSFA) with Reservoir			
Range of output of 1s projected on 1s filter (y1)	12.3146	53.1756	**0.9479**
Range of output of 2s projected on 1s filter (y2)	-7.8218	45.5810	
Incremental Learning (GENEIGSFA) with Higher Order Derivatives			
Range of output of 1s projected on 1s filter (y1)	2.0508	16.6154	**0.9824**
Range of output of 2s projected on 1s filter (y2)	4.3643	21.9050	
Incremental Learning (GENEIGSFA) using Stone's Criterion			
Range of output of 1s projected on 1s filter (y1)	0.8779	8.9089	**0.9773**
Range of output of 2s projected on 1s filter (y2)	2.1950	13.8449	

Table 4.1: Comparison of Magnitude of output signals

the testing set and these characters were projected onto the average filter of a's only. Meanwhile a single 'a' is selected randomly from the test dataset, and the slow features of this character is also extracted. The euclidean distance is calculated between the single character output and the projected a,b,c,d and e on the average filter of 20 a's and the result is found in Table 4.2.

4.6.3 Discussion

As it can be seen from Table 4.2, the proposed GENEIGSFA method has outperformed to the standard SFA method. Firstly, the euclidean distance between the output of 20 a's and the single 'a' by the GENEIGSFA method is less than that of the standard SFA method. Secondly, the distance is also shown graphically in an exponential way in Figure 4.15 where x-axis represents the exponent of euclidean distance Y and Y-axis represents the actual euclidean distance Y. The blue line shows the distance of the GENEIGSFA method which is less in

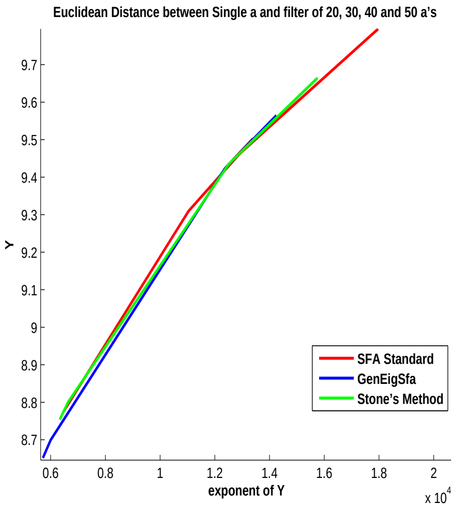


Figure 4.15: Comparative Euclidean Distance Graph

comparison to other methods. Hence shows the less distance between the output produced by the filter of 20 a's and the output of single 'a' projected on the filter of 20 a's. GENEIGSFA strength of recognizing character 'a' is much higher compared to the other proposed method and standard state of the art slow feature analysis criteria. All the techniques were found to be successful in recognizing the 'a' character because the distance of output of character 'a' is less than other characters.

Increasing the number of characters 'a' from 10 to 20, 30, 40 and 50 further increases the performance of GENEIGSFA approach; compared to the other proposed method based on stone's criteria and standard state of the art slow feature analysis technique. The difference gradually became lesser with the increase in the count of characters 'a'. GENEIGSFA always produces minimal distance with the increase in the count of character 'a' compared to other methods where difference is noted low but not as low as is produced by the proposed GENEIGSFA approach.

While comparing the euclidean distance between the single character ‘b’, ‘c’, ‘d’ and ‘e’ with the average filter of 10, 20, 30, 40 and 50 characters of ‘a’. The euclidean distance is observed very high. The reason is due to the difference in shape, size and orientation of other characters which is very different from the single character ‘a’. Therefore, all the technique’s output filter is quite different from the output filter for a single character ‘a’. This shows that all the techniques correctly identified the other characters not similar to single character ‘a’. Some characters like ‘b’ and ‘c’ which look more similar in shape, size and orientation produce lesser euclidean difference compared to the others whereas the difference is greater with the single character ‘a’.

Consequently the slow feature of ‘a’ is much closer to the average output of a’s in comparison to the projection of other characters on the average filter of 20 a’s.

Table 4.2: Euclidean Distance Matrix between a single ’a’ and the projected a,b,c,d and e on average filter for 20 a’s

No of a’s	Single Character	SFA Standard	GENEIGSFA	Stone’s Method
10	a	**9.8011**	**9.5646**	**9.6646**
10	b	12.7465	12.8941	12.5942
10	c	14.4252	14.5521	14.3502
10	d	27.2787	27.8709	27.0879
10	e	21.7497	21.6603	21.6602
20	a	**9.4469**	**9.4322**	**9.4422**
20	b	12.2026	12.4475	11.4916
20	c	14.0197	14.6512	13.6831
20	d	21.1748	26.1381	26.1505
20	e	21.8839	22.3803	21.3733
30	a	**9.5017**	**9.4221**	**9.4318**
30	b	11.4595	12.0080	10.0089
30	c	13.5355	12.7503	12.7511
30	d	26.7195	27.5374	24.5378
30	e	21.8156	22.7354	20.7354
40	a	**9.3095**	**8.6990**	**8.6998**
40	b	10.4456	11.6419	8.6429
40	c	13.0647	11.7851	11.7863
40	d	24.7328	24.3852	21.3862
40	e	20.7265	21.0699	19.0701
50	a	**8.7751**	**8.6513**	**8.6537**
50	b	9.6769	9.1272	8.1271
50	c	12.6343	12.0943	11.0930
50	d	23.2484	28.6332	18.6352
50	e	19.9421	20.7678	17.6574

4.7 Conclusion

This chapter presented two novel incremental techniques for extraction of information from temporal data. Firstly this chapter describe a purely incremental version of slow feature analysis (GENEIGSFA). Secondly it introduced an incremental version based on Stone’s criterion. Both methods were proposed for the purpose of incrementally extracting invariant features from the data. Further the approaches were tested on artificial and real datasets. A new smoothing criterion using higher-order derivatives was also proposed and tested. The result showed that it is often better to pre-process the inputs to the two information extraction techniques by using the outputs of an echo state network.

Finally this chapter concludes that the SFA criterion is more powerful than Stone's criterion. The results of MNIST digit dataset show that the combination of reservoir and SFA is very effective in significantly identifying classes of digits in image data.

The next chapter describes the solution to a problem of finding shared information in multiple data streams simultaneously. The prime objective is to incrementally extract shared temporal information from dual and multiple data streams.

Chapter 5

Extracting online information from Dual and Multiple Data Streams

This chapter presents a challenging problem of finding shared information in multiple data streams simultaneously. Firstly, an existing online version of canonical correlation analysis (CCA) is presented. Secondly, the incremental version of CCA is combined with reservoir of an echo state network (ESN) in order to capture shared temporal information in two data streams. Thirdly, another incremental version of CCA is presented by forcing it to ignore shared information that is created from static values using derivative information. Finally, a novel multi-set CCA method is presented which can identify shared information in more than two data streams simultaneously.

The comparative effectiveness of the proposed methods is demonstrated using artificial and real benchmark data sets.

5.1 Dual Data Streams

This chapter considers the problem of extracting common information from two or more data streams simultaneously. The standard statistical technique for identifying common structure in two data streams is known as Canonical Correlation Analysis (CCA). The purpose of this chapter is to derive a novel approach that can extract canonical information from two streams of time series.

$$\begin{gathered} abbabc \ast\ast\ast cdcdcda \ast cacdcd \\ \text{Direction of time} \longrightarrow \\ abb \ast\ast abccd \ast\ast cdcdaca \ast\ast\ast\ast\ast cdcd \end{gathered} \tag{5.1}$$

Series (5.1) shows a particular temporal pattern (*abbabccdcdcdacacdcd*) from an alphabet of 4 symbols which is to be found in two distinct time series. However, both series contain stray values which are not part of the pattern (shown by '*'s in the figure to indicate these are sections whose value we don't know, or care about). There is a clear and direct relationship between the two time series, but to identify the relevant pattern, we would typically require a technique such as dynamic time warping [166].

The main interest of the proposed work is the generalization of the problem in which the relationship between elements of the time series is not as direct as the above, but can be characterized by finding the canonical variates of the two time series. However, as can be seen from the above, a direct method will fail since the corresponding pairs do not necessarily appear at the same time instant. Therefore, this chapter uses the technique of reservoir computing to get a representation of the time series which contains information about the history of the time series. For example, at position 11 in the time series (5.1), the partial pattern, *abbabccd*, will exist in the reservoir's representation, albeit mixed with a representation of the don't care values, '*'.

Consider the problem of extracting information from two data streams simultaneously when these data streams contain information about each other which may be used to assist with on-going information gathering. These methods may be useful in a number of cases: for example the same underlying signal can be seen through different sensors which will often happen with the various scans of the human brain and heart. This chapter examines the correlation between different signals when there is an underlying hidden reason for the different signals.

5.1.1 Online Temporal CCA

However, canonical correlation analysis is a linear method. It is more interesting to consider methods that can find temporal relationships between pairs of data sets. This chapter uses the above method, but uses the reservoir activations for a pair of related time series and W_{out}^1 and W_{out}^2 in place of $\mathbf{w}_1$ and $\mathbf{w}_2$.

Reservoirs are used to extract information from two data streams simultaneously: Two reservoirs are used with fixed weights between inputs and reservoirs and fixed weights internal to the reservoirs but have two sets of trainable weights which are simultaneously adjusted so that they learn to predict each other's output as shown in figure 5.1.

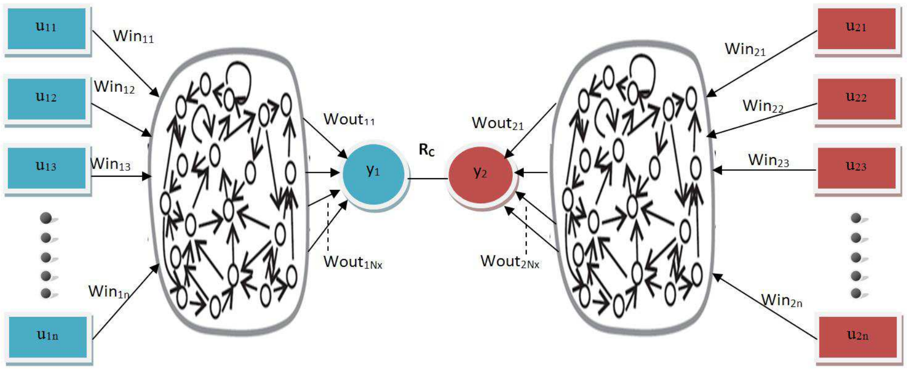


Figure 5.1: Dual Reservoir Streams

Thus simultaneously for paired inputs $\mathbf{u}_1$ and $\mathbf{u}_2$ can be given as:

$$\begin{aligned}
\mathbf{x}_1(t) &= f(\mathbf{W}_{in}^1\mathbf{u}_1(t) + \mathbf{W}^1\mathbf{x}_1(t-1)), \\
\mathbf{x}_2(t) &= f(\mathbf{W}_{in}^2\mathbf{u}_2(t) + \mathbf{W}^2\mathbf{x}_2(t-1)), \\
\mathbf{y}_1 &= \mathbf{W}_{out}^1\mathbf{x}_1, \\
\mathbf{y}_2 &= \mathbf{W}_{out}^2\mathbf{x}_2, \\
\Delta\mathbf{W}_{out}^1 &= \eta(\mathbf{x}_1\mathbf{y}_2 - f(\mathbf{W}_{out}^1)\mathbf{x}_1\mathbf{y}_1), \\
\Delta\mathbf{W}_{out}^2 &= \eta(\mathbf{x}_2\mathbf{y}_1 - f(\mathbf{W}_{out}^2)\mathbf{x}_2\mathbf{y}_2).
\end{aligned} \tag{5.2}$$

The resulting technique is termed as online Temporal CCA method, though clearly it can be used with image data where the relationship between subsequent pixels, or lines is spatial rather than temporal.

5.1.1.1 Artificial Data

Firstly, the proposed method is illustrated on an artificial dataset which has two related sources but the relation is maximized by discovering a temporal mapping. Let $\mathbf{u}_1 = \{\mathbf{u}_1(1), \mathbf{u}_1(2)\}$ and $\mathbf{u}_2 = \{\mathbf{u}_2(1), \mathbf{u}_2(2)\}$. Then the artificial data set has

$$\begin{aligned} \mathbf{u}_1(1) &= \sin(t), \\ \mathbf{u}_1(2) &= \cos(t), \\ \mathbf{u}_2(1) &= t, \\ \mathbf{u}_2(2) &= \tanh(t), \end{aligned} \tag{5.3}$$

where t increases from $-\pi$ to π in steps of 0.01. A 2-dimensional input vector is created having 1000 samples. In the experiment the learning rate was empirically set at 0.0001 and the number of iterations was set at 10000. A temporal correlation of 0.85 was produced, whereas the standard linear non-temporal value was 0.623 [167].

5.1.1.2 Real Data

In order to compare the proposed method with those reported earlier, a real dataset is taken from [168]. This dataset consist of 88 students who sat 5 exams, 2 of which were closed book exams while the other 3 were open book exams. Each student comprises a single exam over the two datasets, resulting two dimensional $\mathbf{u_1}$ (the closed book marks) and a three dimensional $\mathbf{u_2}$ (the open book marks). Since the main interest is to investigate a temporal technique, so it is ensured that the students were presented in a specific order: The average marks of the students over the 5 examinations are used by sampling them in descending order from highest overall marks to lowest. In the experiment the learning rate was empirically set at 0.0001, the size of reservoir was 50, and the number of iterations were set at 50000. The temporal correlation on student's data is 0.7687925 whereas the linear correlation using the standard statistical technique was 0.6630. Therefore the proposed temporal CCA algorithm produces a higher correlation. Validating the actual objective of the CCA approach to extract those features in an unsupervised manner from dual streams; that can maximize the correlation between them compared to the standard state of the art canonical correlation techniques which has produced smaller correlation.

5.1.2 Extracting High Variance Features

In chapter 4, an incremental solution of the generalized eigenproblem was proposed on the slow feature analysis criterion. This approach tries to identify invariances in a data set, and is based on finding the minimal eigenvalue of the covariance of a (single stream) data set while maximizing the eigenvalue of the covariance of the derivatives. This suggests a twist to standard CCA in which the main interest is to maximize the covariance while keeping constant the variance within each data stream while simultaneously maximizing the rate of change of variance within each data set. Intuitively, this explains the motivation of this technique that shows less interest in correlation based on constant values than in correlations in which the rate of change is greater.

This approach is implemented as

$$\left(\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho\left(\begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} - \begin{bmatrix} \Sigma_{1\dot{1}} & 0 \\ 0 & \Sigma_{2\dot{2}} \end{bmatrix}\right)\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}\right.,$$

where Σ_{ij} is the covariance matrix and $\Sigma_{i\dot{j}}$ is the covariance of derivatives of the data with respect to time.

The above method can also be written as

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho\begin{bmatrix} \Sigma_{11} - \Sigma_{1\dot{1}} & 0 \\ 0 & \Sigma_{22} - \Sigma_{2\dot{2}} \end{bmatrix}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}. \tag{5.4}$$

Therefore, the method of finding canonical correlation directions $\mathbf{w}_1$ and $\mathbf{w}_2$ would be

$$\begin{aligned} \frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 - f(\mathbf{w})(\Sigma_{11} - \Sigma_{1\dot{1}})\mathbf{w}_1, \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 - f(\mathbf{w})(\Sigma_{22} - \Sigma_{2\dot{2}})\mathbf{w}_2. \end{aligned}$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i\mathbf{x}_j^T), i, j = 1, 2$ and that $y_1 = \mathbf{w}_1.\mathbf{x}_1$, the instantaneous rules can be proposed as:

$$\begin{aligned} \Delta\mathbf{w}_1 &= \eta(\mathbf{x}_1\mathbf{y}_2 - f(\mathbf{w})((\mathbf{x}_1\mathbf{y}_1) - \left((\frac{d\mathbf{x_1}}{dt})(\frac{d\mathbf{x_1}}{dt}^T)\right)\mathbf{w}_1)), \\ \Delta\mathbf{w}_2 &= \eta(\mathbf{x}_2\mathbf{y}_1 - f(\mathbf{w})((\mathbf{x}_2\mathbf{y}_2) - \left((\frac{d\mathbf{x_2}}{dt})(\frac{d\mathbf{x_2}}{dt}^T)\right)\mathbf{w}_2)). \end{aligned}$$

In practice, to estimate $\frac{d\mathbf{x}}{dt}|_\tau$, $\mathbf{x}(\tau+1)-\mathbf{x}(\tau)$ is used termed as the time derivative of the input dataset. It is calculated by taking the difference between $\mathbf{x}(\tau+1)$ and $\mathbf{x}(\tau)$ where $\mathbf{x}(\tau+1)$ is the value of $\mathbf{x}$ at time $(\tau+1)$ and $\mathbf{x}(\tau)$ is the value of $\mathbf{x}$ at time τ.

Another way of minimizing the rate of change from both the covariance and cross-covariance between datasets X and Y is made. However better results are achieved by maximizing the covariance from within the datasets.

This technique is useful to find CCA for moving objects inside two or more images by extracting only high variance features where the rate of change is maximum. This technique is termed High Variance CCA, HVCCA.

5.1.2.1 Real Data

In order to validate the proposed HVCCA method with those reported earlier, is also tested on student exam data [168]. The correlation vectors of the new and previous methods which includes that standard statistical method and the one reported in [111] are shown in Table 5.1. The learning rate was empirically set at 0.0001, and the number of iterations at 50000.

1	Standard Statistics Maximum Correlation $\mathbf{w}_1$(0.0260 0.0518) $\mathbf{w}_2$(0.0824 0.00081 0.0035)	0.6630
2	Existing Neural Network Maximum Correlation $\mathbf{w}_1$ (0.0270 0.0518) $\mathbf{w}_2$ (0.0810 0.0090 0.0040)	0.6790
3	New Neural Network Maximum Correlation $\mathbf{w}_1$ (0.026 0.0518) $\mathbf{w}_2$ (0.0609 0.0084 0.0042)	0.68125

Table 5.1: Correlations and Weights of Real Data Experiment

Note, reservoir is not used to pre-process the data at this stage. Further the resulting correlation is not higher as previously when reservoir was used: this is only High Variance CCA.

5.1.2.2 Real Images

In order to compare the proposed HVCCA method with those reported earlier, two similar images are used as input. The images are obtained by extracting, the first 150 pixels from both images. Both the datasets are of equal length consisting of 150 rows and 150 columns each. The learning rate was empirically set at 0.0001 and the number of iterations were set at 50000. The experiment was conducted using the previous method reported in [111].

The images are shown below. The first 150 × 150 chunk of pixel data is read from the images. The results are displayed in Table 5.2 which clearly shows the greater correlation produced by the proposed High Variance CCA method compared to the standard state of the art CCA approach. Therefore the proposed method successfully maximizes the correlation between two image streams in comparison with the other which produces a slightly lower correlation shown in Table 5.2.

Existing Neural Network Maximum Correlation	0.7833631
High Variance CCA	0.7935415

Table 5.2: Correlations of Real Image Data Experiment

Note that the reservoirs are not used in this section.

Figure 5.2: Two Real Images Used in the Experiment

5.1.2.3 Temporal High Variance CCA

The High Variance method can be used with reservoir activations for a pair of related times series and $\mathbf{W}^1_{out}$ and $\mathbf{W}^2_{out}$ in place of $\mathbf{w_1}$ and $\mathbf{w_2}$.

This method is illustrated on an artificial data set having two related sources, and is maximised by discovering a nonlinear mapping. Let $\mathbf{u}_1 = \{\mathbf{u}_1(1), \mathbf{u}_1(2)\}$ and $\mathbf{u}_2 = \{\mathbf{u}_2(1), \mathbf{u}_2(2)\}$. Then the artificial data set has

$$\begin{aligned} \mathbf{u}_1(1) &= \sin(t), \\ \mathbf{u}_1(2) &= \cos(t), \\ \mathbf{u}_2(1) &= t, \\ \mathbf{u}_2(2) &= \tanh(t), \end{aligned} \tag{5.5}$$

where t increases from $-\pi$ to π in steps of 0.01. The learning rate was empirically set to 0.0001 and the number of iterations was 10000. The size of the reservoir is equal to 50. The correlations of 0.87 are obtained which contrasts with a correlation of 0.85 determined with the online CCA method of section 5.1.

5.2 Multi-Set Canonical Correlation Analysis

Multi-set Canonical Correlation Analysis (MCCA)[117, 118] is a technique used to analyse a linear relationship between more (than 2) sets of variables. It is considered as a generalized extension of CCA in essence.

For example consider three variables $\mathbf{x}_1$,$\mathbf{x}_2$ and $\mathbf{x}_3$. The method for finding canonical correlations of these variables can easily be extended for n terms. These three variables are then passed through a set of weights, $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ to give outputs $\mathbf{y}_1 = \mathbf{w}_1^T\mathbf{x}_1$, $\mathbf{y}_2 = \mathbf{w}_2^T\mathbf{x}_2$ and $\mathbf{y}_3 = \mathbf{w}_3^T\mathbf{x}_3$.

The criteria for finding Multi-set Canonical Correlations of three variables will be to find the greatest eigenvalue of:

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & 0 & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix}, \qquad (5.6)$$

where Σ_{ij} is the covariance matrix between $\mathbf{x}_i$ and $\mathbf{x}_j$.

The canonical correlation directions $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ are found using

$$\begin{aligned} \frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 + \Sigma_{13}\mathbf{w}_3 - f(\mathbf{w})\Sigma_{11}\mathbf{w}_1, \\ \frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 + \Sigma_{23}\mathbf{w}_3 - f(\mathbf{w})\Sigma_{22}\mathbf{w}_2, \\ \frac{d\mathbf{w}_3}{dt} &= \Sigma_{31}\mathbf{w}_1 + \Sigma_{32}\mathbf{w}_2 - f(\mathbf{w})\Sigma_{33}\mathbf{w}_3. \end{aligned}$$

As before, the instantaneous versions can be derived as:

$$\begin{aligned}\Delta\mathbf{w}_1 &= \eta\mathbf{x}_1(\mathbf{y}_2+\mathbf{y}_3-f(\mathbf{w})\mathbf{y}_1),\\ \Delta\mathbf{w}_2 &= \eta\mathbf{x}_2(\mathbf{y}_1+\mathbf{y}_3-f(\mathbf{w})\mathbf{y}_2),\\ \Delta\mathbf{w}_3 &= \eta\mathbf{x}_3(\mathbf{y}_1+\mathbf{y}_2-f(\mathbf{w})\mathbf{y}_3).\end{aligned}$$

The generalized Multi-set CCA criteria for n terms is given as

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} & ... & \Sigma_{1n} \\ \Sigma_{21} & 0 & \Sigma_{23} & ... & \Sigma_{2n} \\ \Sigma_{31} & \Sigma_{32} & 0 & ... & \Sigma_{3n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \Sigma_{n1} & \Sigma_{n2} & \Sigma_{n3} & ... & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ . \\ . \\ . \\ \mathbf{w}_n \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 & 0 & ... & 0 \\ 0 & \Sigma_{22} & 0 & ... & 0 \\ 0 & 0 & \Sigma_{33} & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 0 & 0 & 0 & ... & \Sigma_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ . \\ . \\ . \\ \mathbf{w}_n \end{bmatrix}, \tag{5.7}$$

from which the obvious generalisation can be given as:

$$\Delta\mathbf{w}_i = \eta\mathbf{x}_i(\sum_{j\neq i} y_j - f(\mathbf{w})y_i). \tag{5.8}$$

5.2.1 Artificial Data

MCCA is illustrated using an artificial data set which has three related sources. Further the relation is maximised by discovering a linear relationship among the three datasets. Let $\mathbf{u}_1 = \{\mathbf{u}_1(1), \mathbf{u}_1(2)\}$, $\mathbf{u}_2 = \{\mathbf{u}_2(1), \mathbf{u}_2(2)\}$ and $\mathbf{u}_3 = \{\mathbf{u}_3(1), \mathbf{u}_3(2)\}$. The artificial dataset has

$$\begin{aligned}
\mathbf{u}_1(1) &= \text{Gaussian Noise (Mean} = 0\text{, standard deviation} = 0.1) \\
\mathbf{u}_1(2) &= \sin(t) + \text{Gaussian Noise (Mean} = 0\text{, standard deviation} = 0.1) \\
\mathbf{u}_2(1) &= 1 - (2.6 - t) * (2.6 - t) + \text{Gaussian Noise (Mean} = 0\text{, standard deviation} = 0.1) \\
\mathbf{u}_2(2) &= \text{Gaussian Noise (Mean} = 0\text{, standard deviation} = 0.1) \\
\mathbf{u}_3(1) &= -(t-3) * (t-2) + \text{Gaussian Noise (Mean} = 0 \text{ and standard deviation} = 0.1) \\
\mathbf{u}_3(2) &= \text{Gaussian Noise (Mean} = 0 \text{ and standard deviation} = 0.1) \qquad (5.9)
\end{aligned}$$

where t increases from 0 to 3.33 in steps of $\frac{1}{300}$ for example, consist of a 3

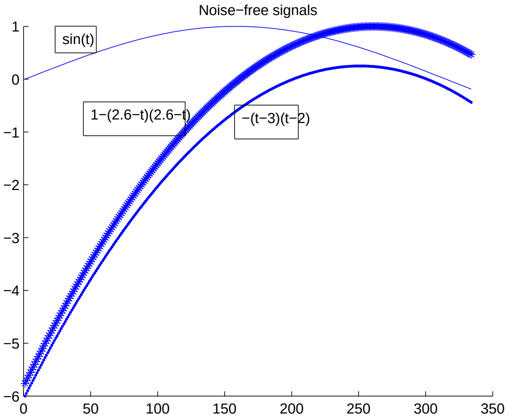


Figure 5.3: Artificial Noise free signals

stream data set of 1000 samples of two dimensional data. The learning rate was empirically set at 0.0001 and the number of iterations were set at 10000. Noise-free versions of the underlying signals of this dataset are shown in Figure 5.3. The multi-set correlation among the three variables is shown in Table 5.3.

	$\mathbf{u}_1$	$\mathbf{u_2}$	$\mathbf{u_3}$
$\mathbf{u_1}$	1.0000000	0.3351417	0.4862097
$\mathbf{u_2}$	0.3351417	1.0000000	0.8666971
$\mathbf{u_3}$	0.4862097	0.8666971	1.0000000

Table 5.3: Multi-Set Correlations Between $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

$\mathbf{w_1}$	0.01548205	**0.7630209**
$\mathbf{w_2}$	**1.05575**	0.0176861
$\mathbf{w_3}$	**0.7339293**	0.08696326

Table 5.4: Weights $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ of $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

It should be noted from Table 5.4 that the parts of each data stream contain true covariance information are those in which the weights are identifying and is highlighted in bold: the other dimensions contain only noise, with weight values two orders of magnitude less. The correlations in Table 5.3 illustrate the very strong correlation between the first elements in each of the second and third data streams. The second element of the first data stream contains a signal which is similar to these two correlated signals. However this signal isn't as close as they are to each other. Therefore the correlation found between signal 1 and the other 2 is somewhat less.

5.2.2 Temporal MCCA (Multiset Canonical Correlation Analysis)

This subsection presents a method which can be used to find non-linear relationships between pairs of data sets. The same MCCA method is used, but with reservoir activations for a pair of related time series and $\mathbf{W}^1_{out}$, $\mathbf{W}^2_{out}$ and $\mathbf{W}^3_{out}$ in place of $\mathbf{w_1}$, $\mathbf{w_2}$ and $\mathbf{w_3}$. Since there are three data streams, three separate reservoirs were used. Further three sets of output weights were updated. Results are shown in Table 5.5.

	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$
$\mathbf{u}_1$	1.0000000	0.3543267	0.5176265
$\mathbf{u}_2$	0.3543267	1.0000000	0.8899055
$\mathbf{u}_3$	0.5176265	0.8899055	1.0000000

Table 5.5: Multi-Set Non-Linear Correlations Between $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

It can be seen that the use of the reservoirs has produced larger values in the non-diagonal weights.

5.2.3 High Variance Multi-Set CCA

The idea remains the same as for multi streams of data but aiming to maximize the changes within each data stream separately.

The criteria for finding High Variance Multi-set Canonical Correlations of three variables will be given as:

$$\begin{bmatrix} 0 & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & 0 & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix} = \rho \left(\begin{bmatrix} \Sigma_{11} - \Sigma_{1\dot{1}} & 0 & 0 \\ 0 & \Sigma_{22} - \Sigma_{2\dot{2}} & 0 \\ 0 & 0 & \Sigma_{33} - \Sigma_{3\dot{3}} \end{bmatrix} \right) \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{bmatrix}. \tag{5.10}$$

The method of finding High Variance canonical correlation directions $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ is then

$$
\begin{aligned}
\frac{d\mathbf{w}_1}{dt} &= \Sigma_{12}\mathbf{w}_2 + \Sigma_{13}\mathbf{w}_3 - f(\mathbf{w})(\Sigma_{11} - \Sigma_{\dot{1}1})\mathbf{w}_1, \\
\frac{d\mathbf{w}_2}{dt} &= \Sigma_{21}\mathbf{w}_1 + \Sigma_{23}\mathbf{w}_3 - f(\mathbf{w})(\Sigma_{22} - \Sigma_{\dot{2}2})\mathbf{w}_2, \\
\frac{d\mathbf{w}_3}{dt} &= \Sigma_{31}\mathbf{w}_2 + \Sigma_{32}\mathbf{w}_2 - f(\mathbf{w})(\Sigma_{33} - \Sigma_{\dot{3}3})\mathbf{w}_3.
\end{aligned}
$$

Using the fact that $\Sigma_{ij} = E(\mathbf{x}_i\mathbf{x}_j^T), i, j = 1, 2$ and that $\mathbf{y}_1 = \mathbf{w}_1.\mathbf{x}_1$, the instantaneous rules can be given as:

$$
\begin{aligned}
\Delta\mathbf{w}_1 &= \eta(\mathbf{x}_1\mathbf{y}_2 + \mathbf{x}_1\mathbf{y}_3 - f(\mathbf{w}) \left(\mathbf{x}_1\mathbf{y}_1 - \left((\frac{d\mathbf{x_1}}{dt})(\frac{d\mathbf{x_1}}{dt}^T) \right)\right), \\
\Delta\mathbf{w}_2 &= \eta(\mathbf{x}_2\mathbf{y}_1 + \mathbf{x}_2\mathbf{y}_3 - f(\mathbf{w}) \left(\mathbf{x}_2\mathbf{y}_2 - \left((\frac{d\mathbf{x_2}}{dt})(\frac{d\mathbf{x_2}}{dt}^T) \right)\right), \\
\Delta\mathbf{w}_3 &= \eta(\mathbf{x}_3\mathbf{y}_1 + \mathbf{x}_3\mathbf{y}_2 - f(\mathbf{w}) \left(\mathbf{x}_3\mathbf{y}_3 - \left((\frac{d\mathbf{x_3}}{dt})(\frac{d\mathbf{x_3}}{dt}^T) \right)\right).
\end{aligned}
$$

The same artificial dataset used in section 5.2.1 is used for the High Variance approach. Table 5.6 shows that the High Variance method has produced slightly higher correlations as compared to the generalized approach. It can be seen more clearly from the values of the weight vectors shown in Table 5.7 that the method is ignoring the noise parts of each data stream and concentrating on the signal parts.

	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$
$\mathbf{u_1}$	1.0000000	0.3357379	0.5029659
$\mathbf{u_2}$	0.3357379	1.0000000	0.8704137
$\mathbf{u_3}$	0.5029659	0.8704137	1.0000000

Table 5.6: Multi-Set Correlations Between $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

Again emphasizing that these results are without the use of a reservoir.

$\mathbf{w_1}$	-0.04771852	**0.7397458**
$\mathbf{w_2}$	**1.073291**	0.006790616
$\mathbf{w_3}$	**0.7530229**	-0.08743803

Table 5.7: Weights $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ of $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

5.2.4 Temporal High Variance MCCA

In this subsection the same above criteria is followed by using reservoir activations to create a new method by which one can compute High Variance canonical correlations among multi-set data. It can be seen from Table 5.8 that the correlations are a bit higher with reservoirs as compared to the generalized technique, but not as high as with temporal CCA using the reservoirs.

	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{u_3}$
$\mathbf{u_1}$	1.0000000	0.3443723	0.5158995
$\mathbf{u_2}$	0.3443723	1.0000000	0.9183531
$\mathbf{u_3}$	0.5158995	0.9183531	1.0000000

Table 5.8: Multi-Set Non-Linear Correlations Between $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$

5.3 Comparative Analysis of MCCA approaches on Real Data

MNIST data set [52] is used as a real dataset consisting of 60000 training patterns containing 0-9 handwritten digits and 10000 test patterns of the same digits (0-9). Each digit consists of 784 pixels which are of 28×28 pixels enclosed in a bounding box. Every digit of the same type is slightly different from every other in terms of position, size and shape. In order to compute multi-set canonical correlation using the proposed methods one digit is randomly chosen from every class (0-9), and the generalized multi-set canonical correlations between digits belonging to different classes is calculated. The learning rate of the algorithms is empirically set and the total number of iterations for learning all set of digits are 100000. A combined comparative results of all the methods related to MCCA (Multi-set Canonical Correlation Analysis) proposed in this chapter are shown in Table 5.9

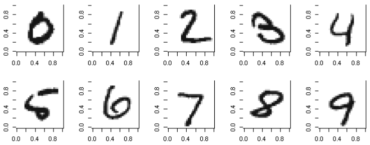


Figure 5.4: Ten Digit Used in the Experiment

5.3.0.1 Discussion

It can be seen in Table 5.9, that some figures have a high correlation with each other for e.g. 6 and 8 while others have a lower correlation for e.g. 3 and 1. The reason is the similarity in the shape of digits for example digit 6 looks very similar to the digit 8. On the other hand digit 1 generally looks very dissimilar to the digit 3. Another reason is that in the MNIST digit dataset each class of digit has more than one image of digits that are different in shape, size and orientation from each other. This difference sometimes makes them more similar to those digits with whom they generally dont look very similar. This is the reason why digits like 1, 4 do not come out as highly correlated as other digits. Therefore, this is the only reason for surprisingly lower correlations produced between some digits by most of the approaches.

Further, all the proposed MCCA approaches of this chapter are applied on this dataset. The proposed temporal linear approaches like GMCCA and HVMCCA overall produces smaller correlations in comparison with the other non-linear approaches like GMMCA(R) (with reservoir) and HVMCCA(R) (with reservoir). GMCCA(R) produces consistent results in comparison with HVMCCA(R). This is due to ignorance of constant information in HVMCCA(R) which do not prove really useful in some digits due to their overall orientation and therefore surprisingly produces a smaller correlation for some of the digits. The only reason why linear approaches like GMCCA and HVMCCA don't produce higher correlation is due to the very strong non-linear nature of the MNIST digit dataset. Still, the latter produces higher correlations compared to the other only due to its criteria of ignoring constant information.

In Table 5.10, the 2-tailed t-values is used to compare the performance of the various methods, comparing them in pairs: when written $a > b, 99\%$, which mean that the improvement in performance of **a** over **b** is significant at the 99% confidence interval; similarly $\mathbf{a} < \mathbf{b}, 99\%$ means that **b** improves **a** with **a** significance value greater than the 99% confidence interval.

According to the performance measurement chart shown in Table 5.10, it can be seen that the addition of reservoirs to the method always improves the canonical correlations with respect to the identification of individual figures. Note that HVMCCA is better than GMCCA (99% confidence interval) but the addition of reservoirs reverses this. The conjecture is that the reservoirs themselves are adding variance though this is a feature which requires further analysis. The rationale behind deriving all these new methods is to extract selected features from the data which can further maximize the correlation between two and more streams in comparison with the previously derived techniques in a completely unsupervised manner. All the techniques performed consistently well for different kinds of data. Temporal CCA is good on numeric time series data. Similarly High Variance CCA (HVCCA) works well on image data. Temporal High Variance CCA proves useful in extracting time series information from image data. Specifically, each technique is designed to work on a particular kind of data stream.

	Method	0	1	2	3	4	5	6	7	8	9
0	GMCCA	1.000	0.571	0.817	0.749	0.693	0.765	0.882	0.851	0.828	0.826
	GMCCA(R)	1.000	0.903	0.979	0.957	0.942	0.966	0.979	0.982	0.965	0.986
	HVMCCA	1.000	0.641	0.876	0.842	0.825	0.856	0.855	0.927	0.875	0.925
	HVMCCA(R)	1.000	0.759	0.934	0.955	0.974	0.973	0.954	0.961	0.961	0.989
1	GMCCA	0.571	1.000	0.705	0.566	0.135	0.735	0.431	0.792	0.714	0.599
	GMCCA(R)	0.903	1.000	0.866	0.816	0.822	0.841	0.896	0.910	0.854	0.906
	HVMCCA	0.641	1.000	0.721	0.719	0.292	0.795	0.518	0.743	0.795	0.564
	HVMCCA(R)	0.759	1.000	0.697	0.723	0.679	0.742	0.697	0.746	0.747	0.749
2	GMCCA	0.817	0.705	1.000	0.779	0.314	0.632	0.785	0.704	0.876	0.592
	GMCCA(R)	0.979	0.866	1.000	0.981	0.935	0.987	0.971	0.973	0.987	0.982
	HVMCCA	0.876	0.721	1.000	0.839	0.553	0.767	0.793	0.798	0.886	0.707
	HVMCCA(R)	0.934	0.697	1.000	0.903	0.928	0.888	0.917	0.867	0.926	0.939
3	GMCCA	0.749	0.566	0.779	1.000	0.387	0.747	0.736	0.724	0.934	0.601
	GMCCA(R)	0.957	0.816	0.981	1.000	0.919	0.978	0.957	0.956	0.989	0.953
	HVMCCA	0.955	0.723	0.903	1.000	0.944	0.987	0.935	0.961	0.981	0.954
	HVMCCA(R)	0.955	0.723	0.903	1.000	0.944	0.987	0.935	0.961	0.981	0.954
4	GMCCA	0.693	0.135	0.314	0.387	1.000	0.594	0.505	0.636	0.367	0.826
	GMCCA(R)	0.942	0.822	0.935	0.919	1.000	0.943	0.957	0.944	0.932	0.941
	HVMCCA	0.825	0.292	0.553	0.595	1.000	0.744	0.674	0.748	0.558	0.913
	HVMCCA(R)	0.974	0.679	0.928	0.944	1.000	0.958	0.975	0.937	0.947	0.972
5	GMCCA	0.765	0.735	0.632	0.747	0.594	1.000	0.631	0.844	0.733	0.878
	GMCCA(R)	0.967	0.841	0.987	0.978	0.943	1.000	0.962	0.963	0.989	0.967
	HVMCCA	0.855	0.795	0.767	0.872	0.745	1.000	0.714	0.903	0.846	0.901
	HVMCCA(R)	0.973	0.742	0.888	0.987	0.958	1.000	0.951	0.962	0.966	0.969
6	GMCCA	0.882	0.431	0.785	0.736	0.505	0.631	1.000	0.611	0.802	0.576
	GMCCA(R)	0.979	0.896	0.971	0.957	0.957	0.962	1.000	0.985	0.967	0.976
	HVMCCA	0.855	0.518	0.793	0.722	0.674	0.714	1.000	0.737	0.799	0.723
	HVMMCA(R)	0.975	0.697	0.917	0.935	0.975	0.951	1.000	0.951	0.958	0.959
7	GMCCA	0.851	0.792	0.704	0.724	0.636	0.844	0.611	1.000	0.815	0.874
	GMCCA(R)	0.982	0.910	0.973	0.956	0.944	0.963	0.985	1.000	0.973	0.987
	HVMCCA	0.927	0.743	0.798	0.914	0.748	0.903	0.737	1.000	0.914	0.917
	HVMCCA(R)	0.954	0.746	0.867	0.961	0.937	0.962	0.951	1.000	0.965	0.943
8	GMCCA	0.828	0.714	0.876	0.934	0.367	0.733	0.802	0.815	1.000	0.628
	GMCCA(R)	0.965	0.854	0.987	0.989	0.932	0.989	0.967	0.973	1.000	0.969
	HVMCCA	0.875	0.795	0.886	0.955	0.558	0.846	0.799	0.914	1.000	0.764
	HVMCCA(R)	0.961	0.747	0.926	0.981	0.948	0.966	0.958	0.965	1.000	0.945
9	GMCCA	0.826	0.599	0.592	0.601	0.826	0.878	0.576	0.874	0.628	1.000
	GMCCA(R)	0.986	0.906	0.983	0.953	0.941	0.967	0.976	0.987	0.969	1.000
	HVMCCA	0.925	0.564	0.707	0.787	0.913	0.901	0.723	0.917	0.764	1.000
	HVMCCA(R)	0.989	0.749	0.939	0.954	0.973	0.969	0.959	0.943	0.945	1.000

Table 5.9: Multi-Set Correlations Between digit 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. GMCCA(Generalized Multi-Set Canonical Correlation Analysis), GMCCA(R)(Generalized Multi-Set Canonical Correlation Analysis with Reservoir), HVMCCA(High Variance Multi-Set Canonical Correlation Analysis), HVMCCA(R)(High Variance Multi-Set Canonical Correlation Analysis with Reservoir).

GMCCA-GMCCA(R)	< 99 %
HVMCCA-GMCCA	> 99 %
HVMCCA(R)-GMCCA	> 99 %
GMCCA(R)-HVMCCA	> 99 %
GMCCA(R)-HVMCCA(R)	> 99 %
HVMCCA(R)-HVMCCA	> 99 %

Table 5.10: Performance Measurement. Paired-wise Comparison with a confidence interval of 99 %

5.4 Conclusion

This chapter presented extensions to find the canonical correlation analysis of a data set. In particular it includes:

1. Used reservoir activations to captured information on temporal, or image data. The online weight adaptation algorithm were used to create a novel method known as Temporal CCA.

2. Used a technique suggested by the Slow Feature Analysis method [2][43] to ensure that the correlations did not come from static signals.

3. Developed online Multi-set CCA methods which are incremental in nature.

4. Combined the above techniques on real and artificial data sets.

It can be concluded that the generalised online method for finding canonical correlations is more appropriate for numeric data (artificial data as well as student exam data). The temporal high variance method is more appropriate for image data sets (MNIST digit data) because in images most of the time the constant data needs to be ignored.

The next chapter will present a generalized incremental laplacian eigenmaps (GENILE), a novel online version of the Laplacian Eigenmaps, one of the most popular manifold-based dimensionality reduction techniques which solves the generalized eigenvalue problem.

Chapter 6

An Online Generalized Eigenvalue Version of Laplacian Eigenmaps

This chapter presents a novel online version of locally optimized Laplacian Eigenmaps (LE), using a manifold-based dimensionality reduction techniques, and solving the generalized eigenvalue problem. Firstly, the comparative performance of the proposed methods with the standard Laplacian Eigenmaps is evaluated on two popular artificial datasets, swissroll and s-curve datasets. Secondly, the proposed online methods are benchmarked against a number of standard batch-based and other manifold-based learning techniques. Finally they are evaluated on the real MNIST digit, bank-note, and heart disease datasets.

Preliminary experimental results demonstrate consistent improvements in the classification accuracy of the proposed method in comparison with standard batch-based manifold-learning techniques.

6.1 Introduction

Most of the traditional techniques used for feature extraction and dimensionality reduction come in both batch and incremental versions. Of all the dimensionality reduction methods proposed in the past, manifold-based learning techniques for feature extraction and dimensionality reduction have gained great popularity. Most of these techniques run in batch mode. Very few incremental approaches based on manifold learning have been proposed. The major difficulty arises in scenarios involving incoming data arriving in multiple chunks from time to time. Batch model algorithms repetitively recalculate the previous chunks at each new input, which becomes computationally very expensive and less efficient.

There are several scenarios for explaining the benefit of incremental learning, and how it overcomes the problem of the high computational and memory cost. Instead of considering a single new entry, the most common scenario should be considered, where the data is coming in more than one chunk and in a sequence. Then the problem of incremental learning involving incoming data in this way can be stated as follows:

Assume $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, \mathbf{x}_{t_1+2}, ..., \mathbf{x}_{t_1+t_2}]$ contains data of two chunks from the whole dataset. Suppose the low-dimensional coordinates $\mathbf{y}_i$ of $\mathbf{x}_i$ representing the first two chunks at time step t_1 and t_2 have already been produced. When the third chunk comes at time step t_3, incremental learning should independently figure out, how to project this chunk of information onto the low dimensional space.

In many scenarios, it is uncommon for all the data to be present before learning. For example social networking site data, online web transaction data, and data received through sensors could be missing. These kinds of data are mostly collected, and stored in raw data form in a distributed file structure storage environments such as Hadoop or Cassandra. Analytical programming environments such as java, matlab and revolution R extract data from these storage sites ranging from terabytes to petabytes and perform learning on these big datasets. The incremental learning technique is best suited to these scenarios because the huge amount of transactional data cannot be learned at once. Instead, the best choice is to learn the data in the form of chunks, or more appropriately, one data point at a time in a completely adaptive manner.

6.2 Manifold-Based Learning

Consider the problem of observing some images. There are factors to be considered like the view angle, rotation and the lighting angle of the pixel intensities. And the data in the high-dimensional space attains a complex non-linear structure. These changes don't occur abruptly, and the data can be reasonably assumed to lie approximately on a (Riemannian) manifold. This is one reason for manifold-based learning technique's gaining a lot of attention.

In this chapter, an online version of Laplacian Eigenmaps is proposed, which is a manifold-based learning technique performed by first building a graph by incorporating the neighborhood information of the dataset. Then, using the notion of the laplacian of the graph, computes a low dimensional representation of the data that optimally preserves local neighborhood information in a certain sense.

The core idea of Laplacian Eigenmaps [107] is very simple: To calculate locally one sparse eigenvalue problem with minimal computation. Since the first part has minimal computation, the idea for creating an incremental version is to solve the sparse eigenvalue problem incrementally therefore making its memory cost more efficient and applicable for deep analysis.

In the following section, the state-of-the-art incremental version of Laplacian Eigenmaps [107] is described.

6.3 Generalized Incremental Laplacian Eigenmaps

The rationale for deriving new online versions in the presence of an existing extension of Laplacian Eigenmaps already proposed in [72] is to highlight four key findings.

1. For Big Data computations, the two positive semi-definite matrices produced for learning will be quite big in size, and require a large amount of computations, which can only be solved by incrementally learning each vector point by point for both matrices.

2. If the data is online in nature, and a light-weight adaptable learning mechanism is required, the proposed online version will be a preferable choice

compared to the standard Laplacian Eigenmaps approach.

3. It is not always important as mentioned in [107, 72] to consider only minimum eigenvalues in producing low-dimensional projections for Laplacian Eigenmaps.

4. The low dimensional embedding can also be calculated incrementally, and independently in one pass without using the existing adjacent information of the previous chunk as in [72].

6.3.1 Procedure:

Let $\mathbf{L}$ be the laplacian matrix, and $\mathbf{D}$ be the diagonal matrix where each value of $\mathbf{D}$ is the sum of each column of $\mathbf{W}$ as explained in [107]. As shown in [55], the optimal weights for a linear projection can be found as the solution of the generalized eigenproblems

$$\mathbf{Lw} = \lambda \mathbf{Dw}. \tag{6.1}$$

Therefore the method can be used described in [55] to obtain

$$\begin{aligned} \Delta\mathbf{w} &= \mathbf{Lw} - f(\mathbf{w})\mathbf{Dw}, \\ \mathbf{w} &= \mathbf{w} + \eta\Delta\mathbf{w}, \end{aligned} \tag{6.2}$$

where $\mathbf{L}$ and $\mathbf{D}$ are both symmetric and semi-definite matrices. Both the matrices are calculated prior to the learning process. The matrices $\mathbf{L}$ and $\mathbf{D}$ are computed on the basis of construction of an adjacency graph matrix on the high dimensional data space followed by creating a weight matrix $\mathbf{W}$. The learning only requires updating of subspace filter $\mathbf{w}$ by using matrices $\mathbf{L}$ and $\mathbf{D}$. Then, by using the generalized eigenvector solution, the filter $\mathbf{w}$ in (6.2) finds the eigenvector corresponding to the maximum eigenvalue. The interesting thing to note here is the selection of eigenvector corresponding to the maximum eigenvalue instead of considering the minimum eigenvalue as in [107, 72], which leads to the loss of a lot of variance of the data and the actual overall orientation of data lying on the high dimensional space. Considering smaller variance, or smallest eigenvalue indicates data to be close to the mean but will make no improvement in projecting the neighboring points closer to one another. In other words, transforming the data to a different direction, and attaining the maximum variance of the data by processing each data point incrementally can produce better results as shown in this paper. Therefore, due to the incremental nature of the proposed algorithm which learns every chunk of data point by point, by projecting it on the eigenvector corresponding to the maximum eigenvalue produces better results, as compared to the standard version of Laplacian Eigenmaps.

It can be seen from (6.1) that the proposed algorithm uses a mixture of batch and online methods since the whole data is used at any one time and the weights are updated incrementally. For a truly neural solution, by updating the weights in an online mode, *and using only one sample at a time*, the matrix $\mathbf{L}$ and $\mathbf{D}$ can be replaced with the instantaneous values so that

$$\mathbf{L}_i\mathbf{w} = \mathbf{D}_i\mathbf{w}, \tag{6.3}$$

where $i = 1, 2, 3, ..., n$. Here $\mathbf{L}_i$ and $\mathbf{D}_i$ means the laplacian matrix $\mathbf{L}$ which is computed as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and the diagonal matrix $\mathbf{D}$ whose entries are sum of each column of $\mathbf{W}$, i.e., $D_{ii} = \sum_j W_{ij}$ will be learned point by point in a purely incremental manner. In order to find the next filter corresponding to the second largest eigenvalue, the matrices $\mathbf{L}$ and $\mathbf{D}$ can be deflated, and again incrementally solve the generalized eigenvector problem to find a second filter:

$$\begin{aligned} \mathbf{L}^* &= \mathbf{L} - \lambda\mathbf{w}\mathbf{L}\mathbf{w}^T, \quad (6.4) \\ \mathbf{D}^* &= \mathbf{D} - \lambda\mathbf{w}\mathbf{D}\mathbf{w}^T, \quad (6.5) \end{aligned}$$

with $\mathbf{L}$ and $\mathbf{D}$ as positive semi-definite matrices and $\mathbf{w}$ being the filter corresponding to the largest eigenvalue. In order to find the next filter corresponding to the second highest eigenvalue $\mathbf{L}$ and $\mathbf{D}$ matrices need to be deflated first by using (6.4) and (6.5).

According to (6.2), the same learning process is conducted using the deflated $\mathbf{L}$ and $\mathbf{D}$, and the next filter corresponding to the second largest eigenvalue is found. The major steps involved in the execution of the incremental algorithm for the next chunk at time step t_2 are shown in Table 6.1.

Table 6.1: The Computing Procedure of the generalized incremental Laplacian Eigenmaps (GENILE) Algorithm

Input: The input patterns $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_{t_2}]$ where $t_2 \ \epsilon$ Next Chunk at time step t_2

Output: The mapping function: $f : R_i^{\mathbf{t_2}} - > R_o^{\mathbf{t_1+t_2}}$

Step 1: [Construct an adjacency Graph Matrix of the new chunk at time step t_2]
Using the K-Nearest Neighbor Algorithm on the whole chunk at time step t_2 and create an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\mathbf{x}_i$ is among the K nearest neighbor of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the K nearest neighbor of $\mathbf{x}_i$ of the chunk at time step t_2.

Step 2:: [Weighting the edges independently of the chunk at time step t_2]

Heat Kernel. $[t \epsilon t_2]$ if node i is connected with j put
$W_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}$,

Simple Approach. Set $W_{ij} = 1$ if vertices i and j are connected
by an edge and set $W_{ij} = 0$ if vertices i and j are not connected by an edge.

Step 3: Construct an objective function of the new chunk independently. Consider $\mathbf{y} = [y_1, y_2, y_3, ..., y_{t_2}]$. The criteria to minimize would be similar to the existing method but this time the minimization will be performed independently for each chunk:

$\sum_{ij} (y_{t_{2_i}} - y_{t_{2_j}})^2 W_{t_{2_{ij}}}$,

and independently calculate $\mathbf{L}$ and $\mathbf{D}$ of the new chunk at the time step t_2
where $D_{t_{2_{ii}}} = \sum_j W_{t_{2_{ji}}}$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$

Step 4: Use the updated eigenspace from the previous chunk belonging to the highest eigenvalue by incrementally solving the GEV of the new chunk at time step t_2 to produce updated $\mathbf{w_1}^{\mathbf{new}}$ where $\mathbf{w_1}^{\mathbf{new}} = \mathbf{w_{t_1+t_2}}$ is the updated eigenspace of the first dimension $\mathbf{L}_i \mathbf{w_1}^{t_1+t_2} = \lambda \mathbf{D}_i \mathbf{w_1}^{t_1+t_2}$.

Step 5: Use the updated eigenspace from the previous chunk belonging to the second highest eigenvalue by deflating $\mathbf{L_{new}}$ and $\mathbf{D_{new}}$ and incrementally solving the GEV of the new chunk at time step t_2 to produce the updated $\mathbf{w_2}^{\mathbf{new}}$ where $\mathbf{w_2}^{\mathbf{new}} = \mathbf{w_2}^{t_1+t_2}$ is the updated eigenspace of the second dimension .

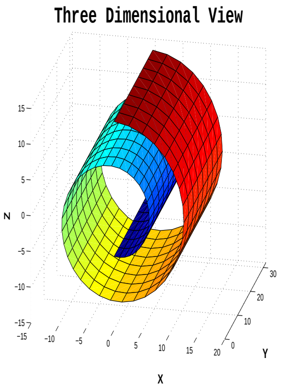


Fig. 6.1: Swiss Roll Dataset

6.3.2 Experiment on an Artificial Dataset

Swiss roll is used as an artificial dataset for the initial experiment. It consists of 20,000 datapoints, and each data point has three dimensions. Since the proposed method is incremental, the data is divided into four different chunks and perform dimensionality reduction on each chunk separately by using the same learned filters $\mathbf{w_1}$ and $\mathbf{w_2}$ of the previous chunk for the next chunk coming ahead. The learning rate was empirically set to 0.00001, and the number of iterations for learning each chunk was 10,000. The learning rate and the number of iterations for learning were initialized with the most appropriate values after checking their effect on the output. The results of the experiment conducted incrementally are shown in Figure 6.2. For the experiments on this artificial dataset the weight matrix $\mathbf{W}$ was defined by

$$W_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{t}}, \text{if vertices } i \text{ and } j \text{ are connected} \\ \\ 0, \text{otherwise} \end{cases}$$

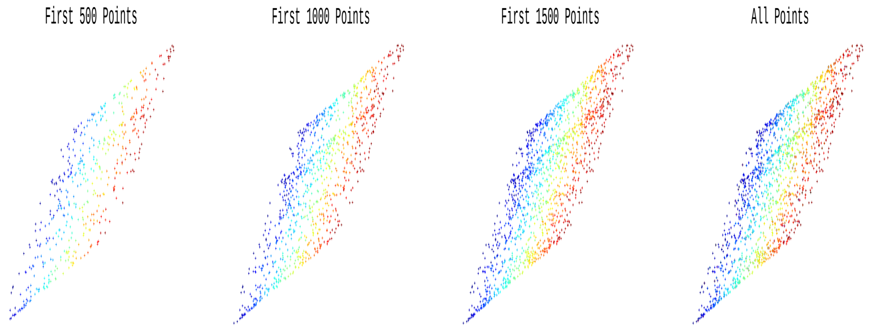


Fig. 6.2: Incremental Laplacian Eigenmaps First-left: Projections for first 500 datapoints. Second-left: Projections for first 1000 datapoints. Third-left: Projections for first 1500 datapoints. Fourth-left: Projections for the first 2000 datapoints

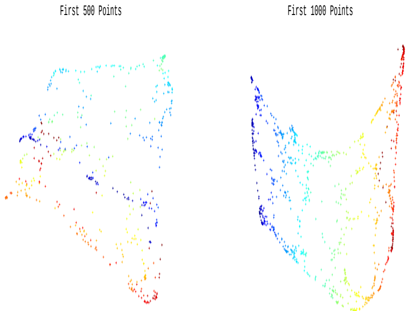


Fig. 6.3: Batch Laplacian Eigenmaps First-left: Projections for first 500 datapoints. Second-left: Projections for the first 1000 datapoints.

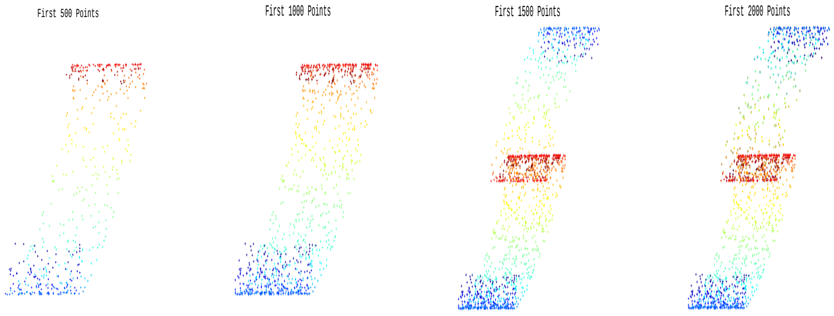


Fig. 6.4: Incremental Laplacian Eigenmaps First-left: Projections for first 500 datapoints. Second-left: Projections for first 1000 datapoints. Third-left: Projections for first 1500 datapoints. Fourth-left: Projections for the first 2000 datapoints

6.3.2.1 Discussion

It has been commonly remarked that the swiss roll dataset is used to evaluate algorithms for manifold-based learning techniques [169] most of the times. The projections computed by the proposed method and standard state of the art Laplacian Eigenmaps technique are coloured on the basis of the low dimensional embedding which can work as an indices for the computed low dimensional projections already contained in the swiss roll dataset. The low dimensional projections should be properly clustered where colour of each value of the low dimensional projection will be determined by its corresponding low dimensional embedding already contained in the dataset. The size of the low dimensional embedding should be equivalent to the size of the computed low dimensional projections. Each value in the vector of low dimensional embedding will be linearly mapped to the colour in the current colormap of the low dimensional plot. The different columns help in evaluating the performance of the dimensionality reduction technique by its computed low dimensional projections. In Figure 6.2, the projections calculated by using the novel incremental version of the Laplacian Eigenmaps are shown by splitting the first 2000 datapoints into four chunks assuming the leftmost to be the first chunk of 500 datapoints whose projections are first calculated and displayed. The next 500 datapoints together with the previous chunk are shown in the second sub-figure from the left, followed by the other two, including the previously processed datapoints, as shown in the third and fourth sub-figures from the left. It can be easily seen that all the datapoints of the swiss roll dataset are properly revealed in two dimensions with minimum collisions despite the fact that the shape is rather flat than rolled but still all the points are properly clustered in the reduced dimensions.

In Figure 6.3 the same swiss roll dataset is tested using batch Laplacian Eigenmaps and tried to learn the manifold of the high dimensional dataset in two separate chunks where the first chunk consists of the first 500 datapoints and the second chunk consists of the first 1000 datapoints. As one can see very easily, the projections produced in both cases are completely different in shape and show no continuity. Of course the factor of repetitive re-calculation is always fermenting the computational efficiency in terms of batch processing.

Further, by projecting the data on minimum eigenvectors in case of standard batch Laplacian Eigenmaps version produces a very sparse kind of shape of a swiss roll with gaps between projected data points in the form of holes. On the other hand, projecting the data in a purely adaptive manner on the eigenvector corresponding to the maximum eigenvalue produces very symmetrical results with no gap or sparseness between points in the reduced dimensions. This shows the inconsistency, computational complexity and non adaptable nature of the batch Laplacian Eigenmaps technique which every time re-calculates the low dimensional projection for each chunk independently, thus producing projections of different shape and size as shown clearly in Figure 6.3. Whereas we learn that with the proposed incremental learner, the low dimensional projections for each chunk are computed in a purely adaptable manner leading to embedding for each chunk which is consistent in shape and size as shown in Figure 6.2.

The simulation results of the proposed incremental algorithm are also demonstrated on the s-curve dataset in Figure 6.4. In the case of the s-curve dataset, the proposed approach is able to produce the results in two dimensions that almost represent the shape of a s-curve as compared to the results produced on the swiss roll dataset. According to the results of real dataset shown in the next section, the improvement in clustering, classification and its purely incremental nature as compared to the standard batch laplacian approach are the actual strengths of the incremental algorithm.

6.3.3 Experiment on MNIST Digit Dataset

In terms of real data, the performance evaluation of the novel purely incremental approach compared with its standard batch version is demonstrated on the MNIST digit dataset [52]. The MNIST digit dataset consist of 60,000 training patterns containing 0-9 handwritten digits and 10,000 test patterns. Each digit contains 784 pixels. The experiment is conducted by taking 500 datapoints beginning with the first four digits (0, 1, 2 and 3) and trying to learn the manifold of the high-dimensional data separately using both the standard and the incremental approach. The results are shown in Figure 6.5 in reduced dimensions.

Table 6.2: The confusion matrices. Left: Batch Laplacian Eigenmaps (LE). Right: GENILE.

	0	**1**	**2**	**3**
0	**480**	1	15	4
1	1	**482**	14	3
2	17	38	**345**	100
3	12	9	218	**216**

	0	**1**	**2**	**3**
0	**490**	0	7	3
1	0	**487**	9	4
2	15	27	**451**	7
3	6	9	28	**457**

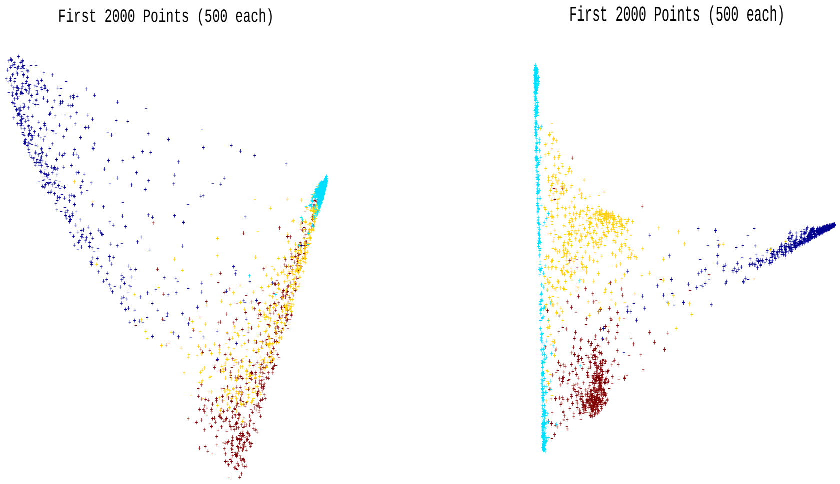


Fig. 6.5: Left: Unfolding first 2000 points (500 each of digits 0,1,2 and 3) using standard LE. Right: Projections of first 2000 points (500 each of digits 0, 1, 2 and 3) using GENILE.

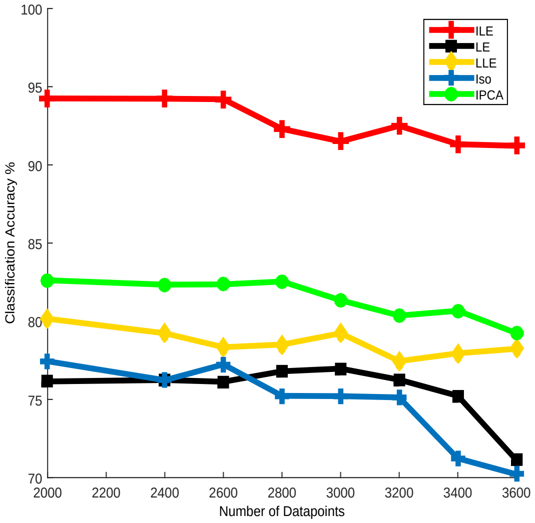


Fig. 6.6: Comparison Between GENILE and other Dimensionality Reduction Methods

Table 6.3: Classification Accuracy of MNIST Digit Dataset

Algorithn	Classifier	Data Length	Dimensions	Accuracy %
IPCA	KNN	2000	784 (28 x 28)	77.44
LLE	KNN	2000	784 (28 x 28)	80.16
Isometric Projection	KNN	2000	784 (28 x 28)	82.62
LE	KNN	2000	784 (28 x 28)	76.15
GENILE	**KNN**	**2000**	**784(28x28)**	**94.25**

6.3.3.1 Discussion

The projections visualized in Figure 6.5 show the same type of digits placed most of the times closer to each other using both the standard and incremental algorithms. In order to clarify the classifications and misclassifications of all the digits, the k-nearest neighbor (kNN) algorithm was executed on the projections of the data with $k = 5$. Assumed as the middle value after considering all the values of k from 1 to 10 in an odd manner. This procedure enables these neighbors to vote for the class of the particular datapoint: the most frequent digits among each digit's five neighbors will be considered as that particular digit's group. The results of the kNN algorithm for both the approaches are shown in the form of a confusion matrix in Table 6.2. According to the confusion matrix, the subtable on the left side shows the results of batch Laplacian Eigenmaps. The subtable on the right shows the results of the generalized incremental Laplacian Eigenmaps (GENILE). Results revealed that both the standard and incremental techniques that the latter outperformed the other by showing the correct classification 1885 times out of 2000. The standard approach shows the correct classification 1523 times out of 2000, which is much less than the other. For further clarification, the number of digits were increased in each class and tested its impact on the classification accuracy compared with other incremental and manifold-based learning algorithms. The comparative results are shown in Figure 6.6. There is a clear sign of improvement by the proposed method compared with the existing batch version and other dimensionality reduction mechanism. The classification accuracy produced by the proposed method is always above 90 % whereas with all the other method including incremental principal component analysis (IPCA) [170], local linear embedding (LLE) [106], Isometric projection [171] and Laplacian Eigenmaps [107] the classification accuracy is always below 90 % as shown in Table 6.3. The reason for the high classification accuracy produced by the proposed method is its point-by-point learning nature which actually made a very clear difference of improvement in the classification accuracy compared with the existing batch Laplacian Eigenmaps approach.

6.3.4 Experiment on the Banknote Authentication Data Set

Next the banknote dataset is taken from genuine and forged banknote-like specimens [167]. This dataset comprises five attributes: 1) variance of Wavelet transformed image, 2) skewness of Wavelet transformed image, 4) entropy of image, and 5) class information. The dataset is organized into two classes. We have tried to learn the manifold of this high-dimensional dataset and tried to reduce the dimensions to properly visualize the dataset in two dimensions. The dataset has a total of 1372 instances. The manifold of the whole dataset is learned by using both the standard and the novel incremental approach and the results in reduced dimensions are visualized in Figure 6.7.

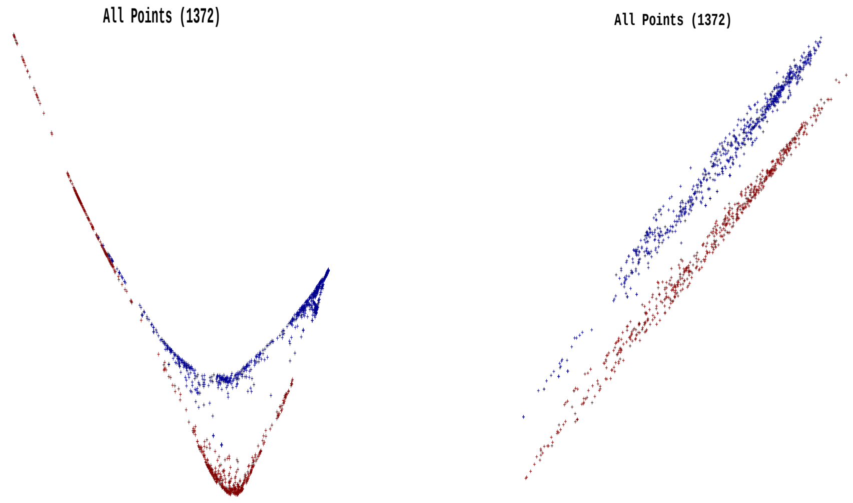


Fig. 6.7: Left: Unfolding all 1327 points using standard LE. Right: Projections of all 1327 points using GENILE.

Table 6.4: The confusion matrices. Left: Batch Laplacian Eigenmaps (LE). Right: GENILE.

	0	**1**
0	**757**	5
1	26	**539**

	0	**1**
0	**762**	0
1	11	**554**

6.3.4.1 Discussion

Here again the projections visualized in Figure 6.7 show the same category of notes placed closed to each other in the low dimensional latent space. The classification of both classes is clearly visible in both projections, however in order to find out which algorithm has produced a slightly greater degree of accuracy in terms of classifying the data, the k-nearest neighbor algorithm is again run on the reduced projections produced by both the algorithms. The value of k is again

Table 6.5: Classification Accuracy of Banknote Dataset

Algorithn	Classifier	Data Length	Dimensions	Accuracy %
IPCA	KNN	1372	5	77.23
Isometric Projection	KNN	1372	5	94.52
LLE	KNN	1372	5	95.21
LE	KNN	1372	5	94.46
GENILE	**KNN**	**1372**	**5**	**95.92**

taken as 5. Each datapoint's five nearest neighbors were checked. The datapoint was labeled based on the maximum number of datapoints from each class. The results of both experiments are shown in the form of confusion matrix in Table 6.6. This table clearly shows a slightly higher degree of classification produced by the novel incremental algorithm. This finding is very rarely found in any other incremental version of dimensionality reduction methods. Similarly, the comparison is performed of the classification accuracy of the proposed method with other existing approaches including IPCA [170], LLE [106] and Isometric projection [171] as shown in Table. 6.5. The classification accuracy of the proposed method is still higher compared with the other methods as shown clearly in Table 6.5.

6.3.5 Experiment on Cardiovascular Disease Dataset

This is a manually collected dataset constructed from gathering more than one type of patient attribute which helped in designing a framework for labeling the patients as cardiovascular or non-cardiovascular based on standard supervised classifiers. This real time dataset is used to check the performance of the novel incremental version of Laplacian Eigenmaps in comparison with the traditional approach. The features of the original dataset were reduced to six using a standard decision tree algorithm [172]. The reduced features were then clustered and classified by using the standard k-means algorithm so that patients of the same type come closer to each other in the higher dimensional space. The total number of patients is 558 and each patient has six attributes. The projections produced by both the methods are visualized in Figure 6.8.

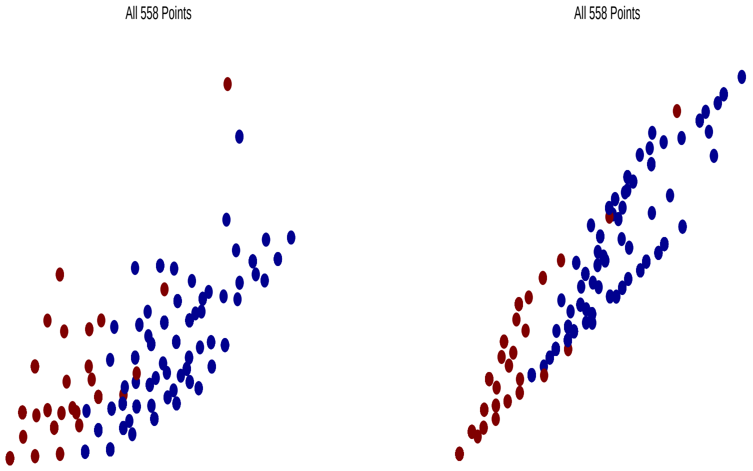


Fig. 6.8: Left: Unfolding all 558 points using standard LE. Right: Projections of all 558 points using GENILE.

Table 6.6: The confusion matrices. Left: Batch Laplacian Eigenmaps (LE). Right: GENILE.

	0	**1**
0	**391**	4
1	7	**156**

	0	**1**
0	**394**	1
1	5	**158**

6.3.5.1 Discussion

In Figure 6.8, the projections produced by both the methods are not clearly observable. A confusion matrix is created which will enable us to find the difference in the classification accuracy between the classes produced by the traditional

Table 6.7: Classification Accuracy of Banknote Dataset

Algorithn	Classifier	Data Length	Dimensions	Accuracy %
IPCA	KNN	558	6	86.43
LLE	KNN	558	6	84.23
Isometric Projection	KNN	558	6	85.32
LE	KNN	558	6	98.02
GENILE	**KNN**	**558**	**6**	**98.93**

batch and by the newly proposed incremental method. According to Table 6.6, the batch version has attained a classification accuracy of 98.02 percent, whereas the new algorithm attained a classification accuracy of 98.93 percent, which is slightly higher than its batch version. The rationale for using this manually collected dataset is to test the practical significance of the proposed method; further comparing it with the other methods including IPCA [170], LLE [106], Isometric projection [171] and LE [107] shown in Table 6.7. The classification accuracy produced by the proposed method is still higher with very close difference from standard LE, but very large difference from other manifold-based incremental learning and batch techniques.

6.4 Conclusion

This chapter has presented a novel online version of the Laplacian Eigenmaps, termed the Generalized Incremental Laplacian Eigenmaps (GENILE). Experimental results showed that the proposed technique can be viewed as a purely incremental technique. It is able to consider each datapoint separately while processing the whole dataset. Traditional incremental methods proposed in the literature don't work separately in each instance. Results have also demonstrated a consistently higher classification accuracy of the developed method and its strongly adaptable nature, compared to the standard laplacian technique, which is considered inapplicable to online data, especially in scenarios involving incoming data in multiple chunks. The tradeoff is the high computational efficiency required for larger chunks due to the purely incremental point by point nature of the proposed technique. This problem can be resolved by dividing the larger chunk into smaller pieces and then performing the incremental learning using the proposed technique.

Finally next chapter will present conclusion and future directions of this thesis.

Chapter 7

Conclusion and Future Work

This Thesis presented online novel learning approaches for Slow Feature Analysis, Canonical Correlation Analysis, Linear Discriminant Analysis and Laplacian Eigenmaps.

As demonstrated throughout this thesis, online domain independent learning has a huge potential for memory efficient solutions, specifically for Big datasets. Commercial Big Data distributions are capable of storing unstructured Big datasets which are then quantified and read using map reducing techniques. The quantified Big Data information read from Big Data clusters are then stored in non-relational distributed databases, such as the Apache HBASE non-relational model, and the Google Big Data model which runs on top of Big data environments like Hadoop and Cassandra. A number of mechanisms for reading and writing the data onto such non-relational models are being extensively used nowadays, such as Hiveql for sql experts, java native HBASE API for java experts, and thrift client HBASE library for python experts etc [168]. Learning normally follows this stage where data is already transformed into a pre-processed, quantified and filtered format. Further preprocessing can be performed before applying machine learning algorithms. At this stage, to further reduce the computational complexity, incremental and online learning techniques of the type proposed in this thesis, are currently considered the best choice to carry out chunk by chunk or point by point sequential and scalable data science [61]. Further, since these online approaches are highly effective at dealing with Big data incrementally, on a chunk by chunk or point by point basis, their performance can thus be better analyzed on small datasets as opposed to Big datasets. Further, as prevalent in real-world Big Data applications, all Big datasets are normally accessed as chunks, in order to best utilize the strength of online domain independent and domain dependent incremental techniques. In conclusion, we therefore hypothesise that the set of novel light-weight domain-independent online theories and algorithms developed in this thesis, can be effectively exploited to handle incremental learning, and associated sequential concept drifts, in both structured and unstructured Big datasets [173]

The First developed algorithm for Linear Discriminant Analysis was an online generalized eigenvalue based LDA. The proposed algorithm was further combined with ELM for linear transformation. Results showed that the proposed online algorithm termed GENILDA, has an equivalent power in terms of discriminability compared with batch LDA. Further GENILDA when combined with ELM did significantly better than LDA.

The proposed algorithm, combined with ELM, was computationally expensive due to the inclusion of a preprocessing phase for random feature mapping. However, this same combination resulted in significantly higher accuracy compared to standard batch LDA. The time complexity was shown to be totally dependent on the optimal selection of hidden neurons when random feature mapping was considered. On the contrary, the proposed online algorithm without ELM, apart from its incremental nature has experimentally shown similar time complexity compared to standard batch LDA. However, both proposed online algorithms proved to be significantly more memory efficient than standard batch LDA, and other state of the art incremental and batch techniques.

The second developed online algorithms were designed for extracting invariant features from temporal data. Primarily, two algorithms were proposed for extracting invariant features using i) L. Wiskott [2] criteria and using ii) James Stone's [96] criteria. Firstly it was concluded that the proposed online invariant feature extraction mechanism using L. Wiskott's criteria was significantly more powerful than Stone's criteria. Secondly the proposed online SFA technique using L. Wiskott's criteria combined with reservoir was significantly affective in identifying classes of digits in image data.

The third developed algorithms were derived to find shared information in multiple data streams simultaneously. An existing online method was combined with reservoir to capture shared temporal information in two data streams. Another incremental version was derived by forcing it to ignore shared information that was created from static values using derivative information. Additionally, the proposed solutions for capturing shared information were all given for multi-data streams.

Results showed that the online method combined with reservoir were significantly better on numeric data whereas temporal high variance method was more appropriate for image datasets.

The fourth built algorithm was a novel online extension of Laplacian Eigenmaps termed the generalized incremental Laplacian Eigenmaps. Results showed that the proposed technique could be viewed as an incremental technique able to consider each data point individually. Results demonstrated a consistently higher classification accuracy of the build method and a strong adaptable nature compared to the standard Laplacian Eigenmaps algorithm.

Further research is required in the following areas:-

Firstly, the proposed novel online feature extraction techniques need to be analyzed by incorporating other feature mapping techniques serving as a preprocessor for linear transformation. Specifically the newly derived multiple layered echo state network, currently being reported by the author in [174], will be used as a pre-processor to evaluate its effect on the output.

Secondly the randomization of the filter needs more research and new criteria could be derived to fine tune the filter during initialization for faster convergence. Techniques like normalization of the randomly initialized weight matrix, and presetting the boundaries of the weight matrix within the range of minimum and maximum eigenvalues of the symmetrical and positive definite matrices could most probably improve the speed of convergence of this algorithm.

Thirdly the derived novel incremental techniques could be applied for several

Big Data applications, particularly those related to 1) Detection, 2) Recognition, 3) Tracking and 4) Forecasting.

Fourthly, customization of the weight decay function, and negative feedback techniques could be employed to analyze their effect on the convergence of the proposed algorithms.

Fifthly, different structures of reservoirs and state-of-the-art extreme learning machines could be researched to extract different information from single or multiple data streams.

Further, standard deflation techniques [175] will be recursively used to deflate the maximum eigenvector to its minimum. This minimum eigenvector will then be used to calculate projections following the same criteria proposed for the novel incremental Laplacian eigenmaps algorithm.

Finally, whilst the preliminary evaluations and conclusions reported in this thesis should be treated with care, they do give interesting on the limits of such methods. What is still needed is further extensive evaluation using a range of real-world Big Data Sets, benchmarked against other state-of-the-art incremental, and batch-based learning approaches.

Bibliography

[1] [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html.

[2] L. Wiskott and T. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[3] Z. Hai, K. Chang, J. J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 623–634, 2014.

[4] N. A. Khadjeh, S. Aghabozorgi, W. T. Ying, and D. C. L. Ngo, "Text mining for market prediction: A systematic review," *Expert Systems with Application*, vol. 41, no. 16, pp. 7653–7670, 2014.

[5] E. Izquierdo-Verdiguier, L. Gomez-Chova, L. Bruzzone, and G. Camps-Valls, "Semisupervised kernel feature extraction for remote sensing image analysis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 9, pp. 5567–5578, 2014.

[6] H. Gatignon, "Canonical correlation analysis," *Statistical Analysis of Management Data, Springer US, DOI: http://dx.doi.org/10.1007/978-1-4614-8594-0-7*, pp. 217–230, 2014.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[8] H. Guo and S. B. Gelfand, "Classification trees with neural network feature extraction," *Neural Networks, IEEE Transactions on*, vol. 3, no. 6, pp. 923–933, 1992.

[9] B. Scholkopft and K. Mullert, "Fisher discriminant analysis with kernels," *Neural networks for signal processing IX*, vol. 1, no. 1, p. 1, 1999.

[10] J. Calic and E. Izuierdo, "Efficient key-frame extraction and video analysis," in *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pp. 28–33, IEEE, 2002.

[11] Y. Song, W. Wang, and F. Guo, "Feature extraction and classification for audio information in news video," in *Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009. International Conference on*, pp. 43–46, IEEE, 2009.

[12] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, pp. 866–870, IEEE, 1998.

[13] C. Sujatha and U. Mudenagudi, “A study on keyframe extraction methods for video summary,” in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, pp. 73–77, IEEE, 2011.

[14] J. Wu, X. Hua, H. Zhang, and B. Zhang, “An online-optimized incremental learning framework for video semantic classification,” in *Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 320–323, ACM, 2004.

[15] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[16] A. Opelt, A. Pinz, and A. Zisserman, “Incremental learning of object detectors using a visual shape alphabet,” in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 1, pp. 3–10, IEEE, 2006.

[17] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, “Deriving marketing intelligence from online discussion,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 419–428, ACM, 2005.

[18] Y. Jo and A. H. Oh, “Aspect and sentiment unification model for online review analysis,” in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 815–824, ACM, 2011.

[19] P. Lipson, A. L. Yuille, D. O'Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal, "Deformable templates for feature extraction from medical images," in *Computer VisionECCV 90*, pp. 413–417, Springer, 1990.

[20] K. Doi, "Current status and future potential of computer-aided diagnosis in medical imaging," *The British Journal of Radiology, http://dx.doi.org/10.1259/bjr/82933343*, vol. 78, no. 2005, 2014.

[21] J. Diz, G. Marreiros, and A. Freitas, "Using data mining techniques to support breast cancer diagnosis," in *New Contributions in Information Systems and Technologies*, pp. 689–700, Springer, 2015.

[22] D. J. Brenner and E. J. Hall, "Computed tomography-an increasing source of radiation exposure," *New England Journal of Medicine*, vol. 357, no. 22, pp. 2277–2284, 2007.

[23] S. Gordon-Salant and P. J. Fitzgibbons, "Temporal factors and speech recognition performance in young and elderly listeners," *Journal of Speech, Language, and Hearing Research*, vol. 36, no. 6, pp. 1276–1285, 1993.

[24] C. Wang and R. X. Gao, "Wavelet transform with spectral post-processing for enhanced feature extraction [machine," *Journal of Speech, Language, and Hearing Research*, vol. 36, no. 6, pp. 1276–1285, 1993.

[25] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Computer Vision–ECCV 2014*, pp. 345–360, Springer, 2014.

[26] R. Maani, S. Kalra, and Y. H. Yang, "Noise robust rotation invariant features for texture classification," *Pattern Recognition*, vol. 46, no. 8, pp. 2103–2116, 2013.

[27] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*, pp. 387–402, Springer, 2013.

[28] E. Rashedi, P. H. Nezamabadi, and S. Saryazdi, "A simultaneous feature adaptation and feature selection method for content-based image retrieval systems," *Knowledge-Based System, doi:10.1016/j.knosys.2012.10.011*, vol. 39, pp. 85–94, 2013.

[29] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic, "Local evidence aggregation for regression-based facial point detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 5, pp. 1149–1163, 2013.

[30] M. Yang, L. Zhang, S. K. Shiu, and D. Zhang, "Robust kernel representation with statistical local features for face recognition," *Neural Network and Learning Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 900–912, 2013.

[31] J. Read, A. Bifet, B. Pfahringer, and G. Holmes, "Batch-incremental versus instance-incremental learning in dynamic and evolving data," in *Advances in Intelligent Data Analysis XI*, pp. 313–323, Springer, 2012.

[32] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[33] D. O. Hebb, *The organisation of behaviour: a neuropsychological theory.* Wiley, 1952.

[34] N. Doidge, *The brain that changes itself: Stories of personal triumph from the frontiers of brain science.* Penguin, 2007.

[35] B. Widrow, Y. Kim, and D. Park, "The hebbian-lms learning algorithm," *Computational Intelligence Magazine, IEEE*, vol. 10, no. 4, pp. 37–53, 2015.

[36] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International conference on artificial intelligence and statistics*, pp. 215–223, 2011.

[37] C. Fyfe, *Hebbian learning and negative feedback networks.* Springer Science & Business Media, 2007.

[38] A. Agrawala, "Learning with a probabilistic teacher," *Information theory*, vol. 16, no. 4, pp. 373–379, 1970.

[39] I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2002.

[40] J. Alcala-Fdez, R. Alcala, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 5, pp. 857–872, 2011.

[41] J. Sun, M. Crowe, and C. Fyfe, "Extending metric multidimensional scaling with bregman divergences," *Pattern recognition*, vol. 44, no. 5, pp. 1137–1154, 2011.

[42] P. Berkes, *Temporal slowness as an unsupervised learning principle: self-organization of complex-cell receptive fields and application to pattern recognition.* Doctoral dissertation, Ph.D Thesis, Institute for Theoretical Biology, Humboldt University, Berlin, 2005.

[43] Z. K. Malik, A. Hussain, and Q. M. J. Wu, "Novel biologically inspired approaches to extracting online information from temporal data," *Cognitive Computation*, vol. 6, no. 3, pp. 595–607, 2014.

[44] B. Kotsiantis, S., I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[45] A. R. Brasier and H. Ju, "Analysis and predictive modeling of asthma phenotypes," in *Heterogeneity in Asthma*, pp. 273–288, Springer, 2014.

[46] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[47] P. S. Dhillon, D. P. Foster, S. M. Kakade, and L. H. Ungar, "A risk comparison of ordinary least squares vs ridge regression," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1505–1511, 2013.

[48] A. DeMaris and S. H. Selman, “Logistic regression,” in *Converting Data into Evidence, ISBN: 978-1-4614-7791-4*, pp. 115–136, Springer, 2013.

[49] Z. K. Malik, A. Hussain, and Q. M. J. Wu, “A neural implementation of linear discriminant analysis with extreme learning machine,” *Neural Network and Learning System, IEEE Transactions on*, Revised Re-Submitted, 2015.

[50] Z. K. Malik, A. Hussain, and Q. M. J. Wu, “Extracting online information from dual and multi data streams,” *Neural Computation & Applications*, In Press, 2015.

[51] Z. K. Malik, A. Hussain, and Q. M. J. Wu, “A novel generalized isometric projection approach for on-line learning,” *Knowledge and Data Engineering, IEEE Transactions on*, (submitted), 2015.

[52] Y. LeCun and C. Cortes, “The mnist database of handwritten digits,” *The Dataset is available at http://yann.lecun.com/exdb/mnist/.*

[53] R. C. Gonzalez and M. G. Thomason, *Syntactic Pattern Recognition, An Introduction.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1978.

[54] W. A. Barbakh, Y. Wu, and C. Fyfe, “Review of clustering algorithms,” *Non-Standard Parameter Adaption for Exploratory Data Analysis*, vol. 249, pp. 7–28, 2015.

[55] Q. Zhang and Y. Leung, "A class of learning algorithms for principal component analysis and minor component analysis," *Neural Networks, IEEE Transactions on*, vol. 11, no. 2, pp. 529–533, 2000.

[56] E. Kuriscak, P. Marsalek, J. Stroffek, and P. G. Toth, "Biological context of hebb learning in artificial neural network, a review," *Neurocomputing, doi:10.1016/j.neucom.2014.11.022*, vol. 152, pp. 27–35, 2015.

[57] B. Zhang and Z. Bao, "Dynamical system for computing the eigenvectors associated with the largest eigenvalue of a positive definite matrix," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 790–791, 1994.

[58] F. Provost and T. Fawcett, "Data science and its relationship to big data and data-driven decision making," *Big Data*, vol. 1, no. 1, pp. 51–59, 2013.

[59] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data impact," *MIS quaterely*, vol. 36, no. 4, pp. 1165–1188, 2012.

[60] A. Cuzzocrea, I. Y. Song, and K. C. Davis, "Analytics over large-scale multidimensional data: the big data revolution!," in *In Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*, pp. 101–104, ACM, 2011.

[61] E. Fokoue, "A taxonomy of big data for optimal predictive machine learning and data mining," *arXiv preprint arXiv:1501*, vol. 006, 2015.

[62] Z. Prekopcsak, G. Makrai, T. Henk, and C. Gaspar-Papanek, “Radoop: Analyzing big data with rapidminer and hadoop,” in *In Proceedings of the 2nd RapidMiner Community Meeting and Conference (RCOMM 2011)*, 2011.

[63] J. I. Maletic and A. Marcus, “Data cleansing: Beyond integrity analysis.,” in *IQ*, pp. 200–209, Citeseer, 2000.

[64] H. A. Ben and J. Weston, “A users guide to support vector machines,” in *Data mining techniques for the life sciences*, pp. 223–239, Springer, 2010.

[65] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised learning,” *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[66] N. Murata, “A statistical study of on-line learning,” *Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK*, pp. 63–92, 1998.

[67] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[68] J. Gao, W. Fan, J. Han, and S. Y. Philip, “A general framework for mining concept-drifting data streams with skewed distributions,” in *In SDM*, pp. 3–14, 2007.

[69] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, “A case-based technique for tracking concept drift in spam filtering,” *Knowledge-Based Systems*, vol. 18, no. 4, pp. 187–195, 20105.

[70] D. Skocaj and A. Leonardis, “Weighted incremental subspace learning,” in *The proceeding of European workshop on Cognitive Vision, Zürich, Switzerland*, pp. 19–20, Citeseer, 2002.

[71] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar, “ldr/qr: An incremental dimension reduction algorithm via qr decomposition,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 9, pp. 1208–1222, 2005.

[72] P. Jia, J. Yin, X. Huang, and D. Hu, “Incremental laplacian eigenmaps by preserving adjacent information between data points,” *Pattern Recognition Letters*, vol. 30, no. 16, pp. 1457–1463, 2009.

[73] Y. Saeys, I. Inza, and P. Larranaga, “A review of feature selection techniques in bioinformatics,” *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[74] V. Bolon-Canedo, N. Sanchez-Marono, and A. Alonso-Betanzos, “A review of feature extraction methods on synthetic data,” *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519, 2013.

[75] X. Wang, “Feature extraction and dimensionality reduction in pattern recognition and their application in speech recognition,” *Doctoral Dissertation, Griffth University*, 2002.

[76] L. C. Molina, L. Belanche, and A. Nebot, “Feature selection algorithm: A survey and experimental evaluation in data mining,” *ICDM 2003, Proceedings, 2002 IEEE International Conference on, IEEE*, 2002.

[77] Y. Saeys, I. Inza, and P. Larranaga, “A review of feature selection techniques in bioinformatics,” *bioinformatics*, vol. 23, no. 19, pp. 306–313, 2007.

[78] V. Bolon-Canado, N. Sanchez-Marono, and A. Alonso-Betanzos, “A review of feature selection methods on synthetic data,” *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519, 2013.

[79] D. T. Larose, “k-nearest neighbor algorithm,” *Discovering Knowledge in Data: An Introduction to Data Mining*, pp. 90–106, 2005.

[80] A. Navot, “On the role of feature selection in machine learning,” *Doctoral Dissertation Hebrew University*, 2006.

[81] L. J. van der Maaten, E. O. Postma, and H. J. van den Harik, “Dimensionality reduction: A comparitive review,” *Journal of Machine Learning Research*, vol. 10, no. 1-41, pp. 66–71, 2009.

[82] G. V. Kass, “An exploratory technique for investigating large quantities of categorical data,” *Applied Statistics*, vol. 29, no. 2, pp. 119–127, 1980.

[83] U. M. Fayyad, A. Wierse, and G. G. Grinstein, *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

[84] I. Bruha and A. Famili, "Postprocessing in machine learning and data mining," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 110–114, 2000.

[85] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Institute for Signal and Information Processing*, 1998.

[86] L. Liu, Y. Jiang, and Z. Zhou, "Least square incremental linear discriminant analysis," in *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pp. 298–306, IEEE, 2009.

[87] C. C. Jia, S. J. Wang, X. J. Peng, W. Pang, C. Y. Zhang, C. G. Zhou, and Z. Z. Yu, "Incremental multi-linear discriminant analysis using canonical correlations for action recognition," *Neurocomputing, doi:10.1016/j.neucom.2011.11.006*, vol. 83, 2012.

[88] T. Kim, S. Wong, B. Stenger, J. Kittler, and R. Cipolla, "Incremental linear discriminant analysis using sufficient spanning set approximations," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[89] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *Systems, Man, and Cybernetics,*

Part B: Cybernetics, IEEE Transactions on, vol. 35, no. 5, pp. 905–914, 2005.

[90] H. Zhao and P. C. Yuen, "Incremental linear discriminant analysis for face recognition," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 1, pp. 210–221, 2008.

[91] G. F. Lu, J. Zhou, and Y. Wang, "Incremental learning of complete linear discriminant analysis for face recognition," *Knowledge-Based Systems, DOI:doi:10.1016/j.knosys.2012.01.016*, vol. 31, pp. 19–27, 2012.

[92] J. Yang and J. Y. Yang, "Why can lda be performed in pca transformed space?," *Pattern recognition*, vol. 36, no. 2, pp. 563–566, 2003.

[93] D. Chu, L. Z. Liao, M. Ng, and X. Wang, "Incremental linear discriminant analysis: A fast algorithm and comparisons," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. PP, no. 99, p. 10.1109/TNNLS.2015.2391201, 2015.

[94] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation*, vol. 14, no. 2, pp. 715–770, 2002.

[95] I. T. Jolliffe, "Principal component analysis," *Springer-Verlag, Newyork*, 1986.

[96] J. V. Stone, "Learning perceptually salient visual parameters using spatiotemporal smoothness constraints," *Neural Computation*, vol. 8, no. 7, pp. 1463–1492, 1996.

[97] V. R. Kompella, M. Luciv, and J. Schmidhuber, "Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams," *Neural Computation*, vol. 24, no. 11, pp. 2994–3024, 2012.

[98] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 8, pp. 1034–1040, 2003.

[99] D. Peng, Z. Yi, and W. Luo, "Convergence analysis of a simple minor component analysis algorithm," *Neural Networks*, vol. 20, no. 7, pp. 842–850, 2007.

[100] B. Yousefi and C. K. Loo, "Development of fast incremental slow feature analysis (f-incsfa)," in *In Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–6, IEEE, 2012.

[101] S. Liwicki, S. Zafeiriou, and M. Pantic, "Incremental slow feature analysis with indefinite kernel for online temporal video segmentation," in *In Computer Vision-ACCV 2012*, vol. 7725, pp. 162–176, 2013.

[102] A. Levey and M. Lindenbaum, "Sequential karhunen-loeve basis extraction and its application to images," *Image Processing, IEEE Transactions on*, vol. 9, no. 8, pp. 1371–1374, 2000.

[103] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1, pp. 37–52, 1987.

[104] W. S. Torgerson, “Multidimensional scaling: I. theory and method,” *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[105] C. M. Bishop, M. Svensen, and C. K. Williams, “Gtm: The generative topographic mapping,” *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.

[106] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[107] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[108] M. Balasubramanian and E. L. Schwartz, “The isomap algorithm and topological stability,” *Science*, vol. 295, no. 5552, pp. 7–7, 2002.

[109] Z. K. Malik, A. Hussain, and Q. M. J. Wu, “An online generalized eigenvalue version of laplacian eigenmaps for visual big data,” *Neurocomputing, doi:10.1016/j.neucom.2014.12.119*, 2015.

[110] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[111] Z. Gou and C. Fyfe, “A family of networks which perform canonical correlation analysis,” *International Journal of Knowledge-based Intelligent Engineering Systems*, vol. 5, pp. 76–82, April 2001.

[112] Z. K. Gou and C. Fyfe, “A canonical correlation neural network for multicollinearity and functional data,” *Neural Networks*, vol. 17, no. 2, pp. 285–293, 2003.

[113] P. L. Lai and C. Fyfe, “Kernel and nonlinear canonical correlation analysis,” *International Journal of Neural Systems*, vol. 10, no. 5, pp. 365–377, 2001.

[114] X. Wang, M. Crowe, and C. Fyfe, “Dual stream data exploration,” *International Journal of Data Mining, Modelling and Management*, vol. 4, no. 2, pp. 188–202, 2012.

[115] P. L. Lai and C. Fyfe, “Kernel and nonlinear canonical correlation analysis,” *International Journal of Neural Systems*, vol. 10, no. 5, pp. 365–377, 2001.

[116] J. O. Ramsay, “Functional data analysis,” *John Wiley and Sons Inc*, 2006.

[117] J. R. Kettenring, “Canonical analysis of several sets of variables,” *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[118] A. A. Nielsen, “Multi-set canonical correlations analysis and multispectral, truely multitemporal remote sensing data,” *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 293–305, 2002.

[119] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy, “Adaptive canonical correlation analysis based on matrix manifolds,” *arXiv preprint arXiv:1206.6453*, 2012.

[120] J. Via, “A learning algorithm for adaptive canonical correlation analysis of several data sets,” *Neural Networks*, vol. 20, no. 1, pp. 139–152, 2007.

[121] S. Akaho, “A kernel method for canonical correlation analysis,” *arXiv preprint cs/0609071*, 2006.

[122] F. Biessmann, F. C. Meinecke, A. Gretton, A. Rauch, G. Rainer, N. K. Logothetis, and K. R. Mller, “Temporal kernal cca and its application in multimodal neuronal data analysis,” *Machine Learning*, vol. 79, no. 1-2, pp. 5–27, 2010.

[123] W. F. Schmidt, M. Kraaijveld, R. P. Duin, *et al.*, “Feedforward neural networks with random weights,” in *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pp. 1–4, IEEE, 1992.

[124] Y. H. Pao and Y. Takefji, “Functional-link net computing,” *IEEE Computer Journal*, vol. 25, no. 5, pp. 76–79, 1992.

[125] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, pp. 985–990, IEEE, 2004.

[126] D. Wang and G. B. Huang, "Protein sequence classification using extreme learning machine," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 3, pp. 1406–1411, IEEE, 2005.

[127] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Advances in neural information processing systems*, pp. 593–600, 2002.

[128] L. P. Wang and C. R. Wan, "Comments on "the extreme learning machine"," *Neural Networks, IEEE Transactions on*, vol. 19, no. 8, pp. 1494–1495, 2008.

[129] L. L. C. Kasun, H. Zhou, and G. B. Huang, "Representational learning with extreme learning machine for big data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31–34, 2013.

[130] G. B. Huang, "An insight into extreme learning machine: random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 1–15, 2014.

[131] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[132] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*, vol. 43, no. 1, pp. 3–31, 2000.

[133] G. B. Huang, L. Chen, and C. K. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[134] N. Y. Liang, G. B. Huang, P. Saratchandran, and N. Sundararajan, “A fast and accurate online sequential learning algorithm for feedforward networks,” *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[135] R. Penrose, “A generalized inverse for matrices,” *In Mathematical proceedings of the Cambridge philosophical society*, vol. 51, no. 3, pp. 406–413, 1995.

[136] D. S. Broomhead and D. Lowe, “Radial basis functions, multi-variable functional interpolation and adaptive networks,” tech. rep., DTIC Document, 1988.

[137] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila, “Pruning and grouping discovered association rules,” 1995.

[138] K. Cox, S. Eick, and G. Wills, “Visual data mining: recognizing telephone calling fraud,” *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 86–102, 1997.

[139] G. F. Cooper, C. F. Aliferis, R. Ambrosino, J. Aronis, B. G. Buchanan, R. Caruana, M. J. Fine, C. Glymour, G. Gordon, B. H. Hanusa, *et al.*, “An

evaluation of machine-learning methods for predicting pneumonia mortality," *Artificial intelligence in medicine*, vol. 9, no. 2, pp. 107–138, 1997.

[140] J. Devore, *Probability and Statistics for Engineering and the Sciences.* Cengage Learning, ISBN-13: 978-0-538-73352-6, 2015.

[141] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? a new look at signal fidelity measures," *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98–117, 2009.

[142] G. Hall, "Pearsons correlation coefficient," *In other words*, vol. 1, p. 9, 2015.

[143] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *In Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[144] B. M. Jedynak and S. Khudanpur, "Maximum likelihood set for estimating a probability mass function," *Neural computation*, vol. 17, no. 7, pp. 1508–1530, 2005.

[145] C. Elkan, "Predictive analytics and data mining," *Retrieved from cseweb.ucsd.edu*, 2013.

[146] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *In Proceedings of the twenty-first international conference on Machine Learning, ACM*, p. 78, 2004.

[147] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[148] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[149] N. R. Pal and S. K. Pal, "Entropy: a new definition and its applications," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 5, pp. 1260–1270, 1991.

[150] S. Kullback, *Information theory and statistics.* Courier Corporation, 1968.

[151] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *In International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1137–1145, ACM, 1995.

[152] Q. Liu, Q. He, and Z. Shi, "Extreme support vector machine classifier," in *Advances in Knowledge Discovery and Data Mining*, pp. 222–233, Springer, 2008.

[153] B. Frénay, M. Verleysen, *et al.*, "Using svms with randomised feature spaces: an extreme learning approach.," in *ESANN*, 2010.

[154] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "Selm: Semi-supervised elm with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, 2011.

[155] L. Lina, D. Liu, and J. Ouyang, "A new regularization classification method based on extreme learning machine in network data," *Journal of Information and Computational Science*, vol. 9, no. 12, pp. 3351–3363, 2012.

[156] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, 2012.

[157] X. Liu, C. Gao, and P. Li, “A comparative analysis of support vector machines and extreme learning machines,” *Neural Networks*, vol. 33, pp. 58–66, 2012.

[158] P. Horata, S. Chiewchanwattana, and K. Sunat, “Robust extreme learning machine,” *Neurocomputing*, vol. 102, pp. 31–44, 2013.

[159] W. Zong, G. B. Huang, and Y. Chen, “Weighted extreme learning machine for imbalance learning,” *Neurocomputing, DOI: 10.1016/j.neucom.2012.08.010*, vol. 101, pp. 229–242, 2013.

[160] P. Hall, D. Marshall, and R. Martin, “Merging and splitting eigenspace models,” *Pattern analysis and machine intelligence, IEEE transactions on*, vol. 22, no. 9, pp. 1042–1049, 2000.

[161] M. Welling, “Fisher linear discriminant analysis,” *Department of Computer Science, University of Toronto*, vol. 3, 2005.

[162] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, “Learning a spatially smooth subspace for face recognition,” in *In Computer Vision and Pattern Recognition, 2007*, pp. 1–7, CVPR’07. IEEE Conference on, 2007.

[163] A. M. Martinez and A. C. Kak, “Pca versus lda,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 228–233, 2001.

[164] P. Berkes, “Pattern recognition with slow feature analysis,” *Cognitive Science EPrint Archive*, 2005.

[165] H. Yaping, Z. Jiali, T. Mei, Z. Qi, and L. Siwei, “Slow feature discriminant analysis and its application on handwritten digit recognition,” in *Proceedings of International Joint Conferent on Neural Networks, Atlanta Georgia,USA*, pp. 1294–1297, 2009.

[166] D. Lemire, “Faster retrieval with a two-pass dynamic-time-warping lower bound,” *Pattern Recognition*, vol. 42, no. 9, pp. 2169–2180, 2009.

[167] P. L. Lai, *Neural Implementations of Canonical Correlation Analysis.* PhD thesis, University of Paisley, 2000.

[168] K. V. Mardia, J. T. Kent, and B. J. M., “Multivariate analysis,” in *Academic Press*, 1979.

[169] J. B. Tenenbaum, V. DeSilva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[170] P. M. Hall, A. D. Marshall, and R. R. Martin, “Incremental eigenanalysis for classification.,” in *BMVC*, vol. 98, pp. 286–295, 1998.

[171] D. Cai, X. He, and J. Han, "Isometric projection," *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 1, p. 528, 2007.

[172] O. Maimon and L. Rokach, "Data mining with decision trees: theory and applications," 2008.

[173] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *Computational Intelligence Magazine, IEEE*, vol. 10, no. 4, pp. 12–25, 2015.

[174] H. A. Malik, Z.K. and Q. Wu, "Multi-layered echo state machine: A novel architecture and algorithm," *Cybernetic, IEEE Transactions on*, 2015, (accepted).

[175] L. W. Mackey, "Deflation methods for sparse pca," in *Advances in neural information processing systems*, pp. 1017–1024, 2009.