CLASSIFICATION TASK-DRIVEN EFFICIENT FEATURE EXTRACTION FROM TENSOR DATA

by

HANIN ALAHMADI

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
November 2018

Abstract

Automatic classification of complex data is an area of great interest as it allows to make efficient use of the increasingly data intensive environment that characterizes our modern world. This thesis presents to contributions to this research area.

The first contribution relates to the problem of discriminative feature extraction for data organized in multidimensional arrays (that is, in tensors). In machine learning, Linear Discriminant Analysis (LDA) is a popular discriminative feature extraction method based on optimizing a Fisher type criterion to find the most discriminative data projection. In the past decade, various extension of LDA to high-order tensor data have been developed. The method proposed in this thesis is called the Efficient Greedy Feature Extraction method. It has two advantages. First of all, it avoids solving optimization problems of very high dimension. Secondly, the algorithm can be stopped when the extracted features are deemed to be sufficient for a proper discrimination of the classes. As other greedy methods, the proposed method extracts the features sequentially, one feature at each step. However, in contrast to the previously known greedy tensor LDA methods, we find a way to condition each step on all previous steps without enforcing orthogonality between the successive projection vectors. This makes our method more efficient than the others. The method is implemented using two slightly different objectives, namely the multiplicative and additive form of the Fisher criterion. The thesis presents the formulas used for the numerical solutions of the optimization problem in both cases. The method is tested both on synthetic data and on real data (fMRI data).

The second contribution of the thesis is an application of the above discriminative feature extraction methods to early detection of dementia disease. For this classification task, the classifier used are the "Learning with Privileged Information" (LUPI) extension of Generalized Matrix Learning Vector Quantization (GMLVQ) classifiers, and also Support Vector Machine (SVM+) that integrates privileged information via LUPI. In contrast to

the original data, Priviledged Information (PI) is the data that is used in the training stage but not in the testing one. It has been reported in the literature that the use of PI can significantly improve the test classification performance. For the early detection task, four cognitive scores are used as the original data while we employ our greedy feature extraction method to derive discriminative PI feature from fMRI data. This approach is of practical significance because fMRI data is quite costly to obtain in practice. The results from the experiments presented in this thesis demonstrate the advantage of using privileged information for the early detection task.

Acknowledgments

First of all, all thanks and praise to god. I would like to express my deepest thanks to my supervisor, Professor Peter Tino and my co-supervisor, Dr. Yuan Shen who supported me throughout my PhD. I appreciated Prof. Tino's and Dr. Shen's continuous encouragement, careful supervision and constructive guidance and also how they assisted me with improvements to my PhD research.

I would like to thank my thesis group members, Dr. John Bullinaria (RSMG rep) and Dr. Iain Styles, who have given their time, comments and interesting research during the development of this thesis. I am grateful to Dr. Shereen Fouad for our research collaboration and for providing me with the necessary GMLVQ code.

My special thanks go to the Taibah University who granted me generous financial and moral support throughout my PhD scholarship, and also to the Saudi Arabian Cultural Bureau, my royal embassy of Saudi Arabia, who assisted me during my past four years of study in the UK.

It would have been impossible to complete this thesis without the support and encouragement from my Mom and Dad; I am really grateful to them. Also, I would like to include a special note of thanks to my husband Yasser; I will remain indebted to you forever.

To my beautiful children Saud and Sufana, thank you for encouraging me with your smiles and for being in my life. To my dear brothers, I owe my deepest gratitude to you. I also extend my heartfelt thanks to my friends for their help, motivation and support.

This thesis was partly copy-edited for the conventions of language, spelling and grammar by Janets Proofreading Service.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EGFE	Efficient Greedy Feature Extraction
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
GSPCA	Greedy Sparse PCA
ORO	Ortho-Rank-One
LDOROTP	Local Discriminative Orthogonal Rank-One Tensor Projection
TVP	Tensor-to-Vector Projection
HODA	Higher Order Discriminant Analysis
SVD	Singular Value Decomposition
ICA	Independent Component Analysis
1D-LDA	1 Dimensional Linear Discriminant Analysis
2D-LDA	2 Dimensional Linear Discriminant Analysis
M-D LDA	Multidimensional Linear Discriminant Analysis
TR1A	Tensor Rank 1 Analysis
TR1DA	Tensor Rank 1 Discriminant Analysis
UMLDA	Uncorrelated Multilinear Discriminant Analysis
GTDA	General Tensor Discriminant Analysis
DATER	Discriminant Analysis with Tensor Representation
CMDA	Constrained Multilinear Discriminant Analysis

fMRI	Functional Magnetic Resonance Imaging
SSLDA	Spatially-Smooth Sparse LDA
MMAE	Macroaveraged Mean Absolute Error
GMLVQ	Generalised Matrix Learning Vector Quantization
SVM	Support Vector Machine
MCI	Mild Cognitive Impairment
AD	Alzheimer's Disease
LPI	Learning with Privileged Information
ROI's	Regions-of-Interests
ANOVA	Analysis of Variance
PSC	Percent Signal Change
SGF	Spatial Grouping Feature
FGF	Functional Grouping Feature
LVQ	Learning Vector Quantization
ITML	Information Theoretic Metric Learning
CD	Cognitive Data
TP	True Positive
TN	True Negatives
FN	False Negatives
FP	False Positives

FPR	False Positive Rate
TNR	True Negative Rate
FPR	False Negative Rate
FNR	True Negative Rate
n_{dots}	working memory
t_{delay}	cognitive inhibition
t^{d}_{disp}	divided attention
t^{s}_{disp}	selective attention

LIST OF NOTATION

m_i	mean of normal distributions
σ_i	variance of normal distributions
$\mathbf{x}$	a vector in R^N
x_o	a classification procedure is defined by a threshold
C_1	an object will be classified to belong to *Class 1*
C_2	an object will be classified to belong to *Class 2*
N_1	number of objects in class 1
N_2	number of objects in class 2
$\mathbf{w}$	a projection vector
$\mathbf{a}, \mathbf{b}$	two (left and right) projection vectors of size $d \times 1$ projecting the matrices into real numbers
$v_{\mathbf{w}}$	the reduced feature set as a map from R^N to R
L	the order of the tensor data
ℓ	mode index
n_ℓ	dimension of mode ℓ, $\ell = 1, \ldots, L$
K	class index, $k \in \{1, 2\}$
N^k	number of elements in each class
$M^{k,i}$	order L Tensor (represented as element of $R^{n_1 \times n_2 \times \ldots \times n_L}$),$k \in \{1, 2\}$,$i \in \{1, \ldots, N^k\}$
D	number of learned features

d feature index $d = 1, \ldots, D$

$\mathbf{a}_{\ell,d} \in R^{n_\ell}$ generating vector for mode $\ell$and feature d

$\mathbf{a}_\ell$=$\mathbf{a}_{\ell,1} \in R^{n_\ell}$ generating vector for mode $\ell$and first feature

$a_{\ell,d,j}$ the component j of vector $\mathbf{a}_{\ell,d}$, $j \in \{1, \ldots, n_\ell\}$

$a_{\ell,j} = a_{\ell,1,j}$ the component j of vector $\mathbf{a}_\ell$, $j \in \{1, \ldots, n_\ell\}$

$A_d = \mathbf{a}_{1,d} \circ \mathbf{a}_{2,d} \circ \ldots \circ \mathbf{a}_{L,d}$

$A_{-\ell,d} = \mathbf{a}_{1,d} \circ \ldots \circ \mathbf{a}_{\ell-1,d} \circ \mathbf{a}_{\ell+1,d} \circ \ldots \circ \mathbf{a}_{L,d} \in R^{n_1 \times n_2 \times \ldots \times n_{\ell-1} \times n_{\ell+1} \times \ldots \times n_L}$

$A_{-\ell} = A_{-l,1} = \mathbf{a}_1 \circ \ldots \circ \mathbf{a}_{\ell-1} \circ \mathbf{a}_{\ell+1} \circ \ldots \circ \mathbf{a}_L \in R^{n_1 \times n_2 \times \ldots \times n_{\ell-1} \times n_{\ell+1} \times \ldots \times n_L}$

$v_d^{k,i} \in R$ feature d representing the tensor $M^{k,i}$, $k \in \{1, 2\}$,$i \in \{1, \ldots, N^k\}$

$v^{k,i} \in R^D$ the vector of features representing the tensor $M^{k,i}$, $k \in \{1, 2\}$,$i \in \{1, \ldots, N^k\}$

$v^{k,i} = f_1^{k,i} \in R$ first feature representing the tensor $M^{k,i}$, $k \in \{1, 2\}$,$i \in \{1, \ldots, N^k\}$

m_d^k the averages of feature d in class k

$m^k = m_1^k$ the averages of the first feature in class k

S_d^k the total square variations of feature d in class k

$S^k = S_1^k$ the total square variations of the first feature in class k

F_m cost function of the Fisher multiplicative criterion for a single feature generating vector set

F_{mD} cost function of the Fisher multiplicative criterion for multiple feature generating vector sets

F_a cost function of the Fisher additive criterion for a single feature generating vector set

F_{aD} cost function of the Fisher additive criterion for multiple feature generating vector sets

$M^{k,i}_{q_\ell=j}$ for the tensor of order $L-1$ with components defined in (3.10)

Δ a tensor of order L defined as the difference of the averages of the tensor data in the two classes

$\Delta_{q_\ell=j}$ a tensor of order $L-1$ defined by (3.16)

$\Omega^k_{q_\ell=j}$ a tensor of order $L-1$defined by (3.18)

$\tilde{\Omega}^k_{q_\ell=j}$ a tensor of order $L-1$ defined by (3.21)

$\mathcal{N}_D$ constant defined by (3.23)

$\mathcal{D}_D$ constant defined by (3.24)

$\Omega^k_{q_\ell=j,D+1}$ a tensor of order $L-1$defined by (3.33)

$\tilde{\Omega}^k_{q_\ell=j,D+1}$ a tensor of order $L-1$defined by (3.36)

$\Omega^{*k}_{q_\ell=j}$ a tensor of order $L-1$ defined by (3.21)

$\Omega^{*k}_{q_\ell=j,D+1}$ a tensor of order $L-1$ defined by(3.46)

α_ℓ Lagrangian multipliers

n_r the number of volumes scanned during the trials with random sequence

n_s the number of volumes scanned during the trials with structured sequence

S the number of voxels

$I_s = \{i_1, ..., i_{n_s}\}$ the collection of "structured" volumes

$I_r = \{j_1, ..., j_{n_s}\}$ the collection of "random" volumes

G the graph structure of a single ROI is represented by so-called graph matrix

n fMRI time series of length

$\mathbf{y}_i = (y_{i1}, ..., y_{in})^\intercal$ linear cross-correlation of fMRI time series i

$\mathbf{y}_j = (y_{j1}, ..., y_{jn})^\intercal$ linear cross-correlation of fMRI time series j

μ the mean of individual fMRI time series

σ the standard deviation of individual fMRI time series

$\{\mathbf{x}_n : n = 1, ..., N\}$ N d-dimensional feature vectors for training

$\mathbf{m}_1 = \frac{1}{N_1} \sum_{\mathbf{x}_n \in \mathscr{C}_1} \mathbf{x}_n$ the mean vectors of *Class 1*

$\mathbf{m}_2 = \frac{1}{N_2} \sum_{\mathbf{x}_n \in \mathscr{C}_2} \mathbf{x}_n$ the mean vectors of *Class 2*

$\mathbf{S}_B$ the between-class covariance matrix

$\mathbf{S}_W$ the total within-class covariance matrix

$\mathbf{D}_B$ the between-class distance

$\mathbf{D}_W$ the total within-class distance

$\mathbf{w}_{\mathsf{opt}}$ the optimized $\mathbf{w}$

$\{v_n = \mathbf{w}_{\mathsf{opt}}^\intercal \mathbf{x}_n : n = 1, ..., N\}$ the extracted features

$\{\mathbf{X}_n : n = 1, ..., N\}$ N graph matrices of size $d \times d$ for training

$\mathbf{M}_1 = \frac{1}{N_1} \sum_{\mathbf{X}_n \in \mathscr{C}_1} \mathbf{X}_n$ mean matrices of *Class 1*

$\mathbf{M}_2 = \frac{1}{N_2} \sum_{\mathbf{X}_n \in \mathscr{C}_2} \mathbf{X}_n$ mean matrices of *Class 2*

$\mathbf{a}_{\mathsf{opt}}, \mathbf{b}_{\mathsf{opt}}$ the optimized $\mathbf{a}$ and $\mathbf{b}$

$\mathbf{X}$ an ROI as a cross-correlation graph represented by an $V \times V$ symmetric matrix

N number of subjects

N_p number of patients

N_c number of healthy controls

$\mathscr{C}_p$ the graph matrices of patients is collected in matrix sets

$\mathscr{C}_c$ the graph matrices of controls is collected in matrix sets

$v = \mathbf{a}^\intercal \mathbf{X} \mathbf{b}$ extract the discriminating feature v through a quadratic form

a_i each element a_i of $\mathbf{a}$ corresponds to a particular voxel i

$\mathbf{r}_i$ spatial position of voxel i

$\mathcal{N}(\mathbf{r}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ a set of K spatially smoothing Gaussian kernels, $k = 1, 2, ..., K$, in the voxel space

$\boldsymbol{\mu}_k$ position of spatially smoothing Gaussian kernel

$\boldsymbol{\Sigma}_k$ shape of spatially smoothing Gaussian kernel determined by the covariance matrix

$\tilde{\mathbf{a}}, \tilde{\mathbf{b}}$ the feature vectors $\mathbf{a}$ and $\mathbf{b}$ reduced

$\tilde{\mathbf{X}}$ the $S \times S$ graph matrix $\mathbf{X}$ is thus reduced to the $K \times K$ matrix

$\mathbf{Y}$ fMRI data of size $\in \mathbb{R}^{S \times T}$

T the number of volumes

$(\mathbf{x}_i, y_i)$ training data $\in \mathbb{R}^m \times \{1, \cdots, K\}, i = 1, 2, \cdots, n, y_i$ is class label

m the dimensionality of data

$\mathbf{w}_q$ $\in \mathbb{R}^m, q = 1, 2, 3, ..., L,$

$c(\mathbf{w}_q)$ $\mathbf{w}_q$ is characterized according to their location available in the input space and their class $\in \{1, ..., K\}$

$d(\mathbf{x}, \mathbf{w})$ the (squared) Euclidean distance between the input vectors and prototypes

Λ	$\in \mathbb{R}^{m \times m}$ is a positive definite matrix, $\Lambda \succ 0$
Ω	$\in \mathbb{R}^{m \times m}$ is a full-rank matrix
ϕ	a monotonic function
$\mathbf{w}^+$	closest prototype with correct label
$\mathbf{w}^-$	closest prototype with wrong label
$\mathscr{X}$	a training dataset that in the original training data live $\mathbf{\Lambda}$
$\mathscr{X}^*$	a training dataset that in another space where the privileged training data live $\mathbf{\Lambda}^*$
$\mathbf{\Lambda}_{\mathsf{new}}$	a new metric is learnt in the original space $\mathbf{\Lambda}$
S_+	a set of similar pairs
S_-	a set of dissimilar pairs
l^*	the upper bound for the distances of similar pairs in the privileged space
u^*	the lower bound for the distances of dissimilar pairs in the privileged space
a^*	lower percentile parameter in the privileged space
b^*	upper percentile parameter in the privileged space
a	lower percentile parameter in the original space
b	upper percentile parameter in the original space
N_d	collection of classifiers trained on different versions of downsampled majority class
Λ_1, Λ_2	two metric tensors, their eigenvalues are normalized to sum to 1
λ_j^i	eigenvalues where $i = 1, 2$ and $j = 1, 2, ..., d$
$\hat{\lambda}_j^i$	the normalized eigenvalues

$\hat{y}$	measures the per-class accuracy of class predictions $\hat{y}$ with respect to true class y on a test set
T_n	the number of test points whose true class is n
d_{ij}	Mahalanobis distance between the i-th and j-th feature vectors (denoted by d_{ij}^M)
I	two feature-generating vectors $\mathbf{a}$ and $\mathbf{b}$ from which we can derive a task-dependent importance matrix denoted by I

CHAPTER 1

INTRODUCTION

1.1 Introduction/Motivation

With the latest advances in sensor, storage, and networking technologies, ever larger data are being generated daily in a wide range of applications and the need to make good use of this data is increasing as well. Entire fields, some of them relatively recently developed, such as computer vision, audio processing, neuroscience, remote sensing, and data mining are important sources of big data and are benefiting from the emerging technologies to make efficient use of it. A reference such as [42], for example, speaks of a "data avalanche" as it refers solely to the field of genetic biology. As a result of this development, the interest in automated processing of big data has grown tremendously. An essential ingredient in the efficient processing of large amounts of data is the capability of data reduction, that is the ability to extract from the raw data those features that are relevant for further processing. Data reduction is an operation useful in itself as it allows for more efficient storage and transmission of data. It can also be used in certain situations as a manner of eliminating noise, or measurement errors from the data. This is a typical use of data reduction algorithm in image processing applications. However, for data reduction to be really useful, this operation has to be performed in such a manner that allows effective and efficient use of the data for the specific purpose for which it was collected in the first place. Classification is one such operation and a very important one. Being able to automate the

sorting of data into classes of interest is a very useful feature in many application areas. For example, medical applications require often to distinguish between healthy specimen and pathological one. This can be the case for results of laboratory analyses, for EKG or EEG plots, or for more complex form of information as psychological tests and fMRI images that are the main concern of the present work.

Most data encountered in the application fields mentioned before is naturally represented as multidimensional arrays. Mathematically, multidimensional arrays are referred as tensors [51]. The number of dimensions (ways) defines the order of a tensor. If N denotes the number of the tensor dimensions, then the elements (entries) of the tensor are addressed by L indices. Each index defines one mode. The tensor concept is a generalization of scalars, vectors and matrices. Indeed, scalars can be regarded as zero-order tensors, vectors can be considered first-order tensors and matrices are second-order tensors. Tensors of order three of higher, i.e. $L > 2$ are referred as higher-order tensors [22, 51]. One typical example of second order tensor data encountered in practice is a gray-level image in computer vision applications, in which case the spatial dimensions of the image represent the two modes. Another example comes from multi channel electroencephalography (EEG) signals in neuroscience, where the two modes consist of channel and time. Also an audio spectrogram can be processed as a two dimensional tensor with frequency and time as the two modes. A typical example of third order tensor is a three-dimensional (3D) model of an object in computer vision or computer graphics [82] in which case the three modes of the three spatial dimensions width, height and depth. Remote sensing using hyperspectral digital imagery collection [81] offers another example of natural third order tensor representation of data with two modes representing the spatial coordinates of the image and a third mode representing the spectral wavelength. Video image sequences are yet another example of data that can be naturally organized as third order tensors with two modes representing the spatial image coordinates and the third mode representing time. This point of view was successfully used for activity and gesture recognition in computer vision and in human-computer interaction applications [[15], [40]]. Social media

and network analysis [56] is a great source of big data nowadays and the higher order tensor point of view has been successfully employed in this area as well. For example [92] organized information from a database on scientific literature a three order tensor with the conference, author, and keyword fields as the three modes. Another similar example comes from web graph data mining and environmental sensor monitoring data [30]. In the case of web graph data the three modes were chosen to be source, destination, and text, whereas in the case of environmental sensor monitoring, three modes were type, location, and time. With the increased interest in cloud computing [3], there are interesting developments that use higher tensor techniques in this area. Such developments have been reported in works such as [23] that presents the MapReduce environment and [48] that presents the GigaTensor approach.

Returning to the topic of data reduction, it is clear that data reduction in case of data represented as higher order tensors presents peculiar challenges. In general, data reduction is the operation of transforming a high-dimensional dataset into a low-dimensional representation while retaining most of the information regarding the underlying structure or the actual physical phenomenon [53]. One important approach to data reduction consists in the supervised learning of a mapping from the higher dimensional input space to the lower dimensional output space. As the lower dimensional space can be regarded as a subspace of the input space, this approach is commonly referred as subspace learning. Traditional subspace learning algorithms are operating on vectors, that is, first-order tensors, such as Principal Component Analysis (PCA) [46], Independent Component Analysis (ICA) [44], Linear Discriminant Analysis (LDA) [27]. For an informative tutorial of these methods, we refer to [12].

All the previously mentioned references on data dimension reduction refer to data organized as vectors, or first-order tensors. Extending these methods to data organized as multidimensional arrays, or higher order tensors, presents serious difficulties. Even the extension of the PCA method to higher order tensor is nontrivial since the Singular Value Decomposition (SVD), that is the underlying numerical tool for the PCA method

cannot be extended easily to higher order tensors. An approach to generalizing the SVD to higher order tensor originates from a relatively old publication[98] and is called the Tucker decomposition. It has been extensively studied ever since and there are many particular cases such as PARAFAC (Parallel Factor Analysis) and NTD (Non-negative Tucker Decomposition) that have been proposed in the literature.

One possible solution was offered in [64] as the Multilinear PCA algorithm. An alternative discrete spectral framework called Greedy Sparse PCA (GSPCA), based on variation bounds on the covariance sub-spectrum derived by eigenvalue inclusion principle, was proposed in [71] . The technique is based on finding solution with sparse linear projections i.e. subject to a constraint on the number of non-zero entries that gives minimum reconstruction error for PCA. The sparse projection can be found either through (i) a mixed integer program that finds the optimal solution and (ii) a heuristic approach through combination of greedy forward search and greedy backward elimination. Utilizing greedy search and branch-and-bound methods to deal with small samples, the complexity of each step of greedy algorithm is $O(n^3)$, that will lead to $O(n^4)$ in total complexity of a full set of solutions [28].

The situation is even more complicated in case of data dimension reduction for classification purposes. Recently, in [103], elimination iterative algorithm for sparse principal component analysis was proposed. Two criteria imposed, the approximated minimal variance loss criterion and the minimal absolute value criterion, to select the eliminated variables in each iteration.

Of course, higher order order tensor data can always be folded into first order tensor (vector) data by folding it into a single array. In this way, conventional data reduction methods can be directly applied. However, this has the disadvantage of eliminating the intrinsic structure of the data and will lead to suboptimal results. Another, more subtle reason why this approach is not advised in the case of LDA is related to the Small Size Sample issue. Indeed, it is known that when there is few data available, the optimization problem that has to be solved during training tends to be badly conditioned numerically.

Multilinear Discriminant Analysis can effectively eliminate this problem [63],[101], [115].

In the past few years, research on a number of multilinear discriminant algorithms based on greedy techniques have shown that dimensionality reduction algorithms with image data encoded as 2D matrices or higher order tensors outperform the algorithms that represent the image data as vectors, particularly for the cases when the training samples are small.

One such dimensionality reduction algorithm performed on face images encoded as matrices or higher order tensors is the discriminant analysis with tensor representation (DATER) [109]. This technique takes the form of vector based LDA tensorisation where the 2D grayscale image is represented directly as matrices. In this method, the dimensionality reduction of the higher order tensors is achieved by iteratively learning the multiple interrelated discriminative sub-spaces through k-mode cluster based discriminant analysis. Eigenvalue decomposition method can be applied to solve the k-mode clustering. It is to note that both DATER and 2D-LDA are the direct extensions of LDA for handling tensor data and 2D data respectively. DATER can handle general high-order input whereas 2D-LDA handles only 2D matrix representations. Both these methods has better learnability than conventional LDA as their projection matrices are constrained to be a Kronecker product of smaller sizes matrices [107] resulting in small dimension of parameters to be estimated. However, DATER is shown to have much high sensitivity to the parameter settings [108]. Exhaustive method for determination of these parameters are not feasible as the subspace dimensions for tensor objects are usually very high e.g. the gait recognition problem is estimated to have 225,280 subspace dimensions. Consequently, reduction in the subspace dimensionality for such cases through DATER becomes very ineffective.

An alternate approach to extract the image feature is called ortho-rank-one (ORO) [43] based on tensor-to-vector projections. ORO adds orthogonal constraints on projection vectors by adopting GLOCAL [17] tensor representation prior to learning. ORO is a greedy tensor LDA algorithm coupling the successive projection vectors via orthogonality

constraints on them. The orthogonality constraint ensures that the estimated projection directions are mutually orthogonal. This iterative method solves for orthogonality constrained eigenvalue problem on one dimension and unconstrained eigenvalue problem on the other dimension. The ORO technique achieves much better results as compared to the original GLOCAL algorithm and other well-known tensor-based methods. However, this technique has limitation as it uses the gray image as features and is therefore unable to exploit the statistical and texture related information of the original image [93].
A more recent technique, inspired by ORO and based on supervised tensor-t for image feature extraction is Local Discriminative Orthogonal Rank-One Tensor Projection (LDOROTP) [106]. This technique has a weighting function which can encode the local discriminant information. The LDOROTP criteria are derived from the slight differences between matrices instead of the trace ratio which then causes a difficulty with a singular matrix. A data pre-processing, GLOCAL, ensures an effective and unchanging iterative scheme of solution. This technique guarantees a stable solution in solving the problem due to fixed orthogonal constraints, as compared to the random assignment scheme of ORO which may give sub-optimal solutions. Furthermore, imposing orthogonality constraints in the reduced data set captures maximum information about the input image by avoiding redundancy, as proposed in the feature extraction and classification method proposed in [79]. The improvement is achieved through orthogonal or non-negative tensor (multi-array) decompositions and higher order discriminant analysis (HODA), in which the input data is treated as tensors rather than the usual matrix representations.

It is also possible to view the low-dimensional representation as latent variables that have to be estimated, or as features that have to be learned. In the latter case, the data reduction operation is regarded as a feature extraction process. Notwithstanding the particular point of view, the extracted features are to be used to perform various tasks, for example, they can be fed into a classifier to identify class labels for the original input data. Dimension reduction of data for classification purposes presents particular challenges as it is important to keep information that is relevant in order to make better distinction

between classes. This requires extracting those features that vary little within the same class, but vary as much as possible from one class to another. We will formalize this idea later, but this is in essence the idea of Linear Discriminant Analysis, that we mentioned before.

The main subject of this thesis is the development of a supervised tensor-based data reduction method for automated classification and its application to the problem of early detection of dementia disease based on cognitive data supplemented with fMRI information. The method of data reduction proposed in this thesis is a contribution to the growing body of Multilinear Subspace Learning (MSL) methods, that are extension of subspace learning methods for data structured as higher order tensors. We refer to [66] for a survey of MSL methods and numerical algorithms to implement them. The method that we propose in this thesis that we call EGFE method differs from the other methods in the fact that it extracts the features one-by-one rather than all at the same time (e.g. 2D-LDA), in this case it selects non-redundant set of basis elements. As other greedy methods, the proposed method extracts the features sequentially, one feature at each step. However, in contrast to the previously known greedy tensor LDA methods, we find a way to condition each step on all previous steps without enforcing orthogonality between the successive projection vectors.

This method is applied to the practical problem of detecting Alzheimer's disease (AD) in the early stage in which it only manifests itself as a mild cognitive impairment (MCI) and we show that combining cognitive test data with fMRI information can be used effectively to select the best features that can be used for separating patients from healthy subjects. The selected features are used for classification using a Generalized Matrix Learning Vector Quantization (GMLVQ) classifier [38]. GMLVQ is a classification technique based on prototypes and it is part of the more general class of Generalized Vector Quantization techniques. The specific of GMLVQ is that the distance function used to determine the closest prototype and thus the class of the object is adapted during learning, which gives an additional level of flexibility to the classification method. Additionally, Support Vector

Machine (SVM) [19], in which the performed classification depends on black box behaviour, in prototype-based techniques, a decision boundary of classification is implemented with maximum margin.

The main contributions of this thesis are detailed in the following section.

1.2 Contributions

The two main contributions in the field of advanced machine learning methodologies for feature extraction and classification are the following:

- An efficient greedy feature extraction method for classification of data represented as higher order tensors;
- An approach to the early detection of dementia disease using a combination of data from cognitive tests and fMRI data as privileged information.

These contributions are described further in detail.

1.2.1 Greedy Methodology for Feature Extraction from Higher Order Tensor Data in Classification Tasks

The proposed Efficient Greedy Feature Extraction (EGFE) method methodology that is proposed in this thesis is based on the Fisher discrimination analysis theory that requires that the features are to be extracted in such way as to:

1. Minimize the within-class distances between the feature values.
2. Maximize the inter-class distances between the means of the feature values.

There are two ways to combine these objectives in a single optimization criterion: the multiplicative approach, in which case the ratio of the two objectives is chosen as the optimization criterion and the additive approach, in which case the (weighted) difference of the two objectives is chosen as the optimization criterion.

The proposed EGFE method works in both cases and it is based on the idea of extracting the discriminating features sequentially one by one. There are two advantages that this method has with respect to other conventional algorithms reported in the literature that attempt to extract directly an apriori specified number of features. The first advantage is that the optimization problem solved at each step is of smaller dimension than it would be required for extracting several features at the same time. The second advantage is that the sequential extraction of the features can be stopped when it is considered that the discrimination power of the already extracted features is sufficiently high (non-redundant set of basis elements). Rather than extracting a number of features that may be unnecessarily large, or a number of features that may be insufficient, our method can be tailored to extract the exact number of features that is necessary to discriminate between the classes. The proposed method extracts the features sequentially, one feature at each step.we use a way to condition each step on all previous steps without enforcing orthogonality between the successive projection vectors like existing greedy methods are doing.

1.2.2 Application in Early Detection of Dementia Disease

Application in biomedical domain in conjunction with privileged information in order to extract useful feature from brain imaging data. A methodology is proposed that combines data from cognitive tests, commonly used for detecting Mild Cognitive Impairment, which is an early stage of dementia disease, with fMRI data collected from the same subjects to train a classifier that uses for classification solely cognitive test data. This is interesting because fMRI data is not commonly available, and is much more expensive to collect, however it is better suited for discrimination than cognitive test data. Therefore, the fMRI data is used as privileged information, that is, only to help the training of the classifier.

The classifiers type that are used in this work are GMLVQ and SVM+ classifiers. The privileged fMRI data is used to modify the tensor metric used by the classifier during the training phase. However, the classification itself is solely based on cognitive test data.

The fMRI data in the form of interactivity graph matrices between the voxels within a region of interest is fed to the GMLVQ classifier only after it is reduced using our greedy feature extraction method to a small number of features based either on spatial grouping, or on functional grouping. In the case of SVM+, the privileged information is utilised to evaluate a slack variable model. It turns out that the use of privileged information can indeed improve the classification performance in comparison to the case where only cognitive test data is used for training the classifier. The thesis presents the formulas used by the optimization algorithms and tests the proposed algorithms on both synthetic data, as on measurement data, e.g. fMRI data.

1.3 Thesis Outline

The rest of the thesis is structured as follows.

- Chapter 2 addresses the basic information and previous research relevant to the rest of this work. After a general review of the literature on feature selection, the discussion is focused on the contribution to conventional LDA algorithms. The discussion is further concentrated on the development of 2D and multilinear extensions, with a brief of nonlinear extensions of LDA as well. The theoretical contributions are complemented with the development of numerical algorithms that address the various challenges that LDA algorithms present in practice. The chapter continues with a short overview of imaging and non-imaging applications of multilinear discriminant analysis. Finally, a list of key research questions is listed along with their concise answer.

- Chapter 3 introduces the Efficient Greedy Feature Extraction method for higher tensor data. Two versions of the method are developed corresponding to the optimization of a multiplicative form and, respectively, of an additive form of the Fisher optimization criterion. The two versions are compared with each other and some numerical issues are discussed.

- Chapter 4 presents our method for detecting mild cognitive impairment using machine learning on cognitive test results supplemented with privileged information based on fMRI data. First, it presents a method to to incorporate the fMRI data as privileged information with cognitive data during training. This includes an explanation of the collection process of cognitive test and fMRI data as well as the method of generating the graph matrix that describes the network activity as explained before. Subsequently, we use our EGFE method for feature extraction from the data thus obtained and we use the GMVLQ and SVM+ classifiers to process the reduced data set. This chapter also includes the experiment design for the experiments that are reported here. Finally, the results of a few numerical experiments are reported which allows to quantify the value of including privileged information with the cognitive test results in order to improve the classification performance. It discusses the extraction of first and second features from second order tensor data obtained from fMRI data using our EGFE methods and compares the performance of the classifier using these features over the case that the features are extracted using the 2D-LDA method.

- In Chapter 5, demonstrates generating synthetic data of higher order tensor data is considered, specifically third order tensor data, and the performance of the Efficient Greedy Feature Extraction method is demonstrating by extracting three discriminating features, Additionally, competitive experiments with ortho-rank-one method are examined.

- Finally, Chapter 6 gives a summary of the presented work and a list of possible research subjects that are suggested by the current work and could be considered in the future.

1.4 Publications From the Thesis

Hanin H Alahmadi, Yuan Shen, Shereen Fouad, Caroline Di B Luft, Peter Bentham, Zoe Kourtzi, and Peter Tino. Classifying cognitive profiles using machine learning with privileged information in mild cognitive impairment. Frontiers in computational neuroscience, 10, 2016.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Modern digital technology has given us access to an incredible amount of data and it is doing so at an increasing rate. However, making proper use of the data in order to make suitable decision has not registered the same rate of progress. Perhaps in no other field is this as obvious as in the case of image and video information. The proliferation of digital camera's in the modern society has generated a tremendous quantity of imaging data of all kind. However, most tasks that make use of such information have to be performed "manually", in the sense that human eyes are directly involved. This is a real limitation in many situations in which quick and reliable processing of visual information is required. On the other hand, the potential applications are very important and range from generic computer vision tasks, human face recognition, biomedical applications such as EEG signal processing.

This situation has stimulated a growing interest in the development of methods and techniques for automatic data processing in order to replace, at least partially, human decision making. These methods and techniques originate in probability and statistics, optimization, artificial neural networks, digital signal processing, artificial intelligence and other disciplines, and were organized in a new research field called Machine Learning. Although the exact boundary of this field are not exactly defined, one of the outstanding

problems that are central to this field is classification.

This operation, also known as pattern recognition, consists in assigning the *objects* in a set to a number *classes*, or *categories* based on a number of measurement data about the objects that are called *features*. Practically, classification amounts to establish a mapping between the set of all possible features and the set of classes.

There are two classes of classification methods: supervised and unsupervised. In the case of supervised classification, a set of apriori classified objects is available and the purpose of the automatic classification methods is to find a general mapping (or a *discriminant*) between the set of all possible features and the set of classes that can be used for an arbitrary object. This is called *training* and the set of apriori classified objects is called the *training set*. Typically, the mapping is chosen from a particular class of functions, for example, linear functions, which is the main interest of this work. In the case of unsupervised classification, there is no apriori class set and the task is to find possible partitions of the given objects into different classes. Although in practice there a number of apriori hypotheses that are used in the process of unsupervised classification, this is a more difficult problem that supervised classification. This work deals exclusively with supervised classification, and therefore the further discussion is limited to this topic.

The main difficulty of establishing the classification map is that the values of the features are "noisy" as they originate in measurements and therefore there is always some degree of uncertainty in doing the classification. For this reason, designing a classification process and especially evaluating a classification process is based on probabilistic and statistical techniques. Excellent overviews of these techniques as well as detailed presentations of some of them can be found in textbooks such as [96, 11, 73]. Since this work is a contribution towards efficient implementation of tensor discriminant classification, that is a subset of linear discriminant analysis classification, we will discuss this class of methods in more details. However, let us first mention that the most general classification techniques are based on Bayesian decision theory. Indeed, this theory allows to design classification methods that control directly the probability of classification error (the risk), or the

probability of correct classification (the performance). An important class of Bayesian classifiers are actually based on evaluating the probability that a given object belongs to a given class (conditioned by the features) and selecting the class with the highest probability. This is the essence of the *maximum likelihood* approach. Of course, these probabilistic methods are based on various hypotheses on the distribution of features, which may in some situations be difficult to check, or may not be satisfied. However, it is shown [96, Chapter 2, Section 2.4] that if classes are normally and identically distributed, optimal Bayesian classifiers are linear discriminant analysis LDA functions This mathematical result justifies and encourages the attempts to design linear classifiers irrespective of the actual distribution of classes and features. However, to cite again [96, pag. 33] a major problem associated with LDA is the large number of the unknown parameters that have to be estimated in the case of high-dimensional spaces". Indeed, in the case of imaging applications, if the LDA would be applied to the entire set of measurements (all the pixel values), the number of parameters of an LDA classifier would be in the order of millions even for a modestly accurate image type. The challenge thus is to select (a small number of) features that retain the essential characteristics of the data and allow for efficient classification.

Posed in this way, the problem of feature selection is a dimensionality reduction problem: a large number of features is reduced to a smaller number. However, caution is warranted since not every dimensionality reduction method is appropriate. For example, the most popular dimensionality reduction method, PCA (Principal Component Analysis, see e.g. [89]) may eliminate exactly those dimensions that are essential for classification. This problem is known for a long time as an early reference such as [35] reveals this problem. Notice that in this cited referece, PCA is called the Karhunen-Loeve transform, just as in other references, PCA is confused with the closely related technique of Singular Value Decomposition (SVD). For the similarities and differences between PCA and SVD, as well as for the more general method called *Independent Component Analysis* (ICA), we refer to [89]. The contrast between PCA and Fisher's LDA in the context of image classification is

also revealed in the more recent publication [9]. Efforts to combine the advantages of PCA in the case of dimensionality reduction with the advantages of LDA in class separation and thus classification have continued within the field of image recongnition as shown in publications such as [45, 113, 91]. A different direction of adapting PCA for handling classification problem was proposed in [6], called supervised PCA.

It is interesting to mention that, although modern needs have stimulated tremendously this kind of research, the problem and the main approach to its solution have a long history as they appeared in very old research on the classification of animal species in the works of R.A. Fisher [33, 34].

The structure of this chapter is as follows. In Section 2.2, we discuss classical LDA from the point of view of dimensionality reduction. The discussion is extended to 2D and multilinear discriminant analysis in Section 2.3. A discussion of nonlinear extensions of LDA is given in Section 2.4. Numerical algorithm issues for linear and multilinear discriminant analysis are presented in Section 2.5. Greedy algorithms for data reduction are discussed in 2.6. A short overview of imaging applications of multilinear discriminant analysis is given in Section 2.7.

2.2 Linear Discriminant Analysis Classification as a Dimensionality Reduction Procedure

The presentation below of LDA classification is partially suggested by [11, Subsection 4.1.4], but the particular details are original. The presentation is limited to the case of two classes, but extensions to multiple classes is possible. The first step in this presentation is to consider the simplest case in classification theory, that is uni-dimensional classification.

2.2.1 Univariate Classification

In this case, the objects are characterized by a single feature and we assume that the two classes are characterized by normal distributions of mean m_i and variance σ_i, $i = 1, 2$. We

assume that $m_1 < m_2$. In this case, a classification procedure is defined by a threshold x_o and an object will be classified to belong to class C_1 if $x < x_o$ and to belong to class C_2 if $x > x_o$. It can be proven (see e.g. [11, Subsection 1.5.1] that in order to minimize the probability of misclassification the threshold x_o has to be chosen in such a way that the two probabilities $p(x_o; C_1)$ and $p(x_o; C_2)$ are equal. Based on our assumption on the distributions of the two classes

$$\frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{(x_o-m_1)^2}{\sigma_1^2}} = \frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{(x_o-m_2)^2}{\sigma_2^2}}$$

that means (if $\sigma_1 \approx \sigma_2$)

$$\frac{(x_o - m_1)^2}{\sigma_1^2} \approx \frac{(x_o - m_2)^2}{\sigma_2^2} \tag{2.1}$$

which has as the only solution between m_1 and m_2

$$x_o = \frac{\frac{m_1}{\sigma_1} + \frac{m_2}{\sigma_2}}{\frac{1}{\sigma_1} + \frac{1}{\sigma_2}}. \tag{2.2}$$

The common value of the fractions in (2.1) can be readily computed and is equal to

$$J = \frac{(m_2 - m_1)^2}{(\sigma_1 + \sigma_2)^2}. \tag{2.3}$$

It is easy to see geometrically in figure 2.1, but it can also be justified analytically, that the probability of error decreases as a function of this ratio, that is, if J is larger, then the probability of error is smaller. Indeed, as the difference of the two means is greater relative to the two variances, the probability of error represented as the area of the intersection of the two areas in Figure 2.1.

The training process for the case of a single feature is simply based on estimating m_i and σ_i from the available training set. In case there are N_1 objects in class C_1 and N_2

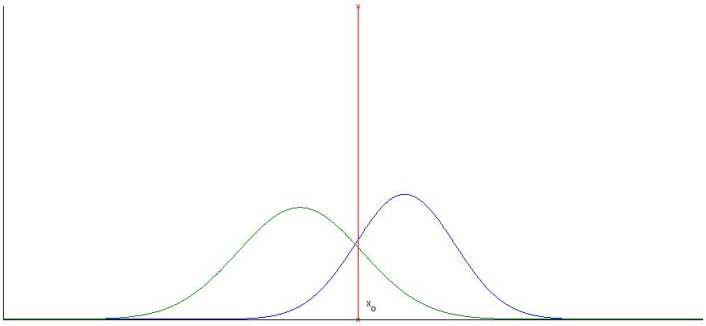

Figure 2.1: The solution x_o of the equation (2.1) is the approximate value of the optimal threshold for the univariate classification of two normally distributed classes. The probability of error is smaller as the area under the intersection of the two curves is smaller.

features in class C_2, then the estimates of the mean and variance are

$$\hat{m}_i = \frac{1}{N_i}\sum_{k=1}^{N_i} x_k^i,\ \hat{\sigma}_i^2 = \frac{1}{N_i}\sum_{k=1}^{N_i}(x_k^i - \hat{m}_i)^2,\ i = 1, 2.$$

2.2.2 Optimal Single Feature Extraction

Let us assume that each object is described by a vector of dimension N, and we want to classify each object into two classes C_1 and C_2. A linear discriminant for this problem is a function of the form

$$v(\mathbf{x}) = \mathbf{w}^T\mathbf{x} - x_o$$

defined by a vector $\mathbf{x}$ in R^N and a number x_o such that an object is classified in C_1 if $v(\mathbf{x}) < 0$ and it is classified in C_2 if $v(\mathbf{x}) > 0$.

As proven in [96, Section 2.4.] if the features in both classes are normally distributed with the *same* covariance matrix, the optimal solution of the classification problem is provided by a linear discriminant (whereas if the two classes have different covariance matrixes, a quadratic discrimininant is optimal). Designing the discriminant based on a training set can be regarded as a dimension reduction problem if the procedure of finding

$\mathbf{w}$ is separated from the procedure of finding x_o. Indeed, let us consider the linear mapping

$$v_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

as a map from R^N to R. This transforms the N dimensional feature vector into a single number, effectively reducing the multivariate classification problem to a univariate classification problem. For each choice of $\mathbf{w}$, the choice of x_o is performed using the procedure presented in the previous subsection. Thus $\mathbf{w}$ should be determined in such a way that (2.3) is maximal. In Figure 2.2, the case $N = 2$ is represented graphically.

Denoting by $\mathbf{x}_k^i$, $i = 1, 2$ the feature values of the objects in the training set, the reduced feature set are the images of these vectors through the map $v_{\mathbf{w}}$. If the feature values are normally distributed with means $\mathbf{m}_i$, $i = 1, 2$ and the same variance Σ, then the images through the map $v_{\mathbf{w}}$ are normally distributed with means $m_i = \mathbf{w}^T\mathbf{m}_i$, $i = 1, 2$ and variance $\sigma_1^2 = \sigma_2^2 = \mathbf{w}^T\Sigma\mathbf{w}$. The criterion (2.3) becomes

$$J = \frac{\mathbf{w}^T(\mathbf{m}_1^T - \mathbf{m}_2^T)(\mathbf{m}_1 - \mathbf{m}_2)\mathbf{w}}{4\mathbf{w}^T\Sigma\mathbf{w}}, \tag{2.4}$$

and $\mathbf{w}$ has to be chosen to maximize this criterion, as this will lead to the best classification performance on the reduced feature set.

Using the values of the feature vectors in the training set, the parameters of the criterion (2.4) can be estimated using the formula's

$$\hat{\mathbf{m}}_i = \frac{1}{N_i}\sum_{k=1}^{N_i}\mathbf{x}_k^i,\ i = 1, 2$$
$$\hat{\Sigma} = \frac{1}{(N_1 + N_2)}\sum_{i=1}^{2}\sum_{k=1}^{N_i}(\mathbf{x}_k^i - \hat{\mathbf{m}}_i)(\mathbf{x}_k^i - \hat{\mathbf{m}}_i)^T.$$

With these formula's, the training process should find the vector $\mathbf{w}$ such that the following criterion is maximized

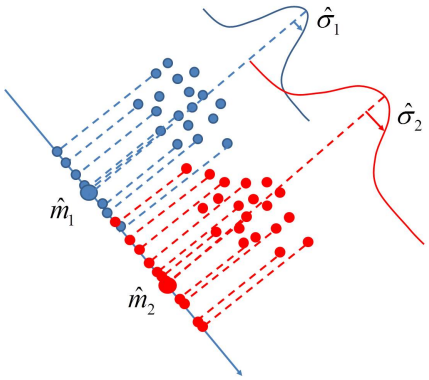

Figure 2.2: Mapping 2D data to 1D data through projection allows for univariate classification, but only if the projection direction is chosen appropriately. The two dimensional data corresponding to the two classes are represented in blue and red. Their image through the map v_w are represented as the projections on the line $w^T x = ct$. Clearly, the direction of the projection line determines the separation between the points that correspond to the two classes. By choosing w appropriately, the separation between these points can be improved.

$$J(\mathbf{w}) = \frac{\mathbf{w}^T(\hat{\mathbf{m}}_1^T - \hat{\mathbf{m}}_2^T)(\hat{\mathbf{m}}_1 - \hat{\mathbf{m}}_2)\mathbf{w}}{\mathbf{w}^T \left[\sum_{i=1}^{2}\sum_{k=1}^{N_i}(\mathbf{x}_k^i - \hat{\mathbf{m}}_i)(\mathbf{x}_k^i - \hat{\mathbf{m}}_i)^T\right]\mathbf{w}}. \tag{2.5}$$

This is the general form of the Fisher criterion for Linear Discriminant Analysis that we deduced here from the requirement of obtaining the best classification performance on the univariate feature set that results from reducing the original set through a linear mapping. The numerator of the fraction in (2.5) can be interpreted as a "between classes" variance and the denominator can be interpreted as a sum of "within classes" variances of the data.

We have deduced the expression of this criterion under the assumption that the variance of the data in the two classes is identical, which is a reasonable assumption as the data is gathered using the same sensors. In practice, this assumption may not always be satisfied since the variance of the measurement data may also be determined by the nature of the objects themselves, so may be determined by the class those objects belong to. In this case, a linear discriminant is not optimal. As it is shown in [96, Subsection 2.4.2], the optimal discriminant in this case is quadratic. However, if a linear discriminant is to be used, finding this discriminant by maximizing the criterion (2.5) is a reasonable choice, although suboptimal. Also, notice that according to [75], the linear discriminant is quite

robust when the data departs from the assumption of equal covariance for the two classes.

Notice that the Fisher criterion is not used to perform the classification itself. Rather it is used to find the vector $\mathbf{w}$ that is subsequenly used to extract a single feature $v_{\mathbf{w}}(\mathbf{x})$ from the vector of features to use for the classification itself. Clearly, the criterion (2.5) does not change if $\mathbf{w}$ is multiplied by a scalar, so it is reasonable to impose the additional constrained that its norm should be 1. The training process can be restated as solving the optimization problem

$$\max_{||\mathbf{w}||=1} J(\mathbf{w}).$$

The immediate extension of this idea is to extract multiple features from the same vector in order to decrease the complexity of the classification process. This idea is pursued in the following section.

2.2.3 Optimal Multiple Features Extraction

Let us assume as in the previous subsection that each object is described by a vector of dimension N, and we want to classify each object into two classes C_1 and C_2. There are simple situations that can be imagined for which the classification problem cannot be reduced to a univariate classification problem, as we have done in the previous subsection. In this case, a possible outcome is to try to reduce the number of features used for classification to a smaller number $\ell < N$, though larger than 1. A simple way to achieve this is by using a linear transformation.

For simplicity, take $\ell = 2$. In this case of two dimensional classification, we are looking for a linear map from R^N to R^2. Such a map is defined by two vectors $\mathbf{w}^1$ and $\mathbf{w}^2$ in R^N and has the form

$$v_{\mathbf{w}^1,\mathbf{w}^2}(\mathbf{x}) = \begin{bmatrix} (\mathbf{w}^1)^T\mathbf{x} \\ (\mathbf{w}^2)^T\mathbf{x} \end{bmatrix}.$$

In this case, we need to find the two vectors in such a way that the transformed data can be efficiently classified as a two dimensional data. One way to achieve this is to

maximize the following Fisher criterion that can be seen as an extension of the expression (2.4) to the case of the extraction of two features:

$$J(\mathbf{w}^1, \mathbf{w}^2) = \frac{(m_1 - m_2)^T (m_1 - m_2)}{\sigma_1^2 + \sigma_2^2}, \tag{2.6}$$

where

$$m_1 = \begin{bmatrix} \frac{1}{n_1} \sum_{i=1}^{n_1} (\mathbf{w}^1)^T \mathbf{x}_i^1 \\ \frac{1}{n_1} \sum_{i=1}^{n_1} (\mathbf{w}^2)^T \mathbf{x}_i^1 \end{bmatrix}, \; m_2 = \begin{bmatrix} \frac{1}{n_2} \sum_{j=1}^{n_2} (\mathbf{w}^1)^T \mathbf{x}_j^2 \\ \frac{1}{n_2} \sum_{j=1}^{n_2} (\mathbf{w}^2)^T \mathbf{x}_j^2 \end{bmatrix},$$

and

$$\sigma_1^2 = \sum_{i=1}^{n_1} (v_{\mathbf{w}^1, \mathbf{w}^2}(\mathbf{x}_i^1) - m_1)^T (v_{\mathbf{w}^1, \mathbf{w}^2}(\mathbf{x}_i^1) - m_1),$$
$$\sigma_2^2 = \sum_{j=1}^{n_2} (v_{\mathbf{w}^1, \mathbf{w}^2}(\mathbf{x}_j^2) - m_2)^T (v_{\mathbf{w}^1, \mathbf{w}^2}(\mathbf{x}_j^2) - m_2).$$

It is possible to rewrite (2.6) as an expression similar to (2.5) by introducing the trace operator

$$J(\mathbf{w}^1, \mathbf{w}^2) = \frac{\text{trace} \begin{bmatrix} (\mathbf{w}^1)^T \\ (\mathbf{w}^2)^T \end{bmatrix} S_m \begin{bmatrix} \mathbf{w}^1 & \mathbf{w}^2 \end{bmatrix}}{\text{trace} \begin{bmatrix} (\mathbf{w}^1)^T \\ (\mathbf{w}^2)^T \end{bmatrix} S_s \begin{bmatrix} \mathbf{w}^1 & \mathbf{w}^2 \end{bmatrix}}, \tag{2.7}$$

where

$$S_m = \left(\bar{\mathbf{x}}^1 - \bar{\mathbf{x}}^2\right) \left(\bar{\mathbf{x}}^1 - \bar{\mathbf{x}}^2\right)^T$$

and

$$S_s = \sum_{i=1}^{n_1} (\mathbf{x}_i^1 - \bar{\mathbf{x}}^1)(\mathbf{x}_i^1 - \bar{\mathbf{x}}^1)^T + \sum_{i=1}^{n_2} (\mathbf{x}_i^2 - \bar{\mathbf{x}}^2)(\mathbf{x}_i^2 - \bar{\mathbf{x}}^2)^T$$

with

$$\bar{\mathbf{x}}^1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{x}_i^1, \;\; \bar{\mathbf{x}}^2 = \frac{1}{n_2} \sum_{i=1}^{n_2} \mathbf{x}_i^2.$$

The training process amounts to chosing the vectors $\mathbf{w}^1$ and $\mathbf{w}^2$ that maximize the criterion (2.7).

In the same way, if $\ell > 1$ features are required, a linear map from R^N to R^ℓ is defined by a matrix $W \in R^{N\times\ell}$ as

$$v_W(\mathbf{x}) = W^T\mathbf{x}.$$

The best separation between the projected classes in R^ℓ can be obtained by maximizing the criterion

$$J(W) = \frac{\text{trace}W^T S_m W}{\text{trace}W^T S_s W}. \tag{2.8}$$

It is easy to show that multiplying the columns of W by arbitrary scalars will not change the value of J so it is reasonable to ask that the columns of W are constrained to have unit norm.

The approach to feature extraction for classification presented in this section is commonly referred in the literature as 1D-LDA (see e.g. [114]). As it can be seen, even in the case treated in previous section, it involves a nonlinear optimization problem with N decision variables. In many applications such as image classification, this can be a very large number, which makes this a very challenging problem from the numerical point of view. An even more serious problem occurs when the number of training samples is smaller than N. Indeed, from the expression of the matrix S_s, it is clear that it is a matrix of rank at most $n_1 + n_2$. If $n_1 + n_2 < N$, which is practically always the case for imaging applications, the matrix is singular and therefore the denominator of the criterion can be nulled, which means that the maximum is infinity, and it is actually attained for very many choices of the matrix W. This problem is known under the name *small sample size problem* and has often be tackled in the literature (see e.g. [52]). An overview of methods that modify the Fisher's discriminant problem to this case is given in [104]. The same reference proposes itself such a method called Penalized Linean Discriminant analysis. Another simple method was presented in [18] that essentially proposes to modify (2.8) to become

$$J(W) = \frac{\text{trace}W^T S_m W}{\text{trace}W^T (S_s + S_m) W}, \tag{2.9}$$

as a way of avoiding the singularity of the denominator that is characteristic for the small

sample size problem.

Again a different, more generic method that addresses this problem, and also the problem of reducing the dimensionality of the optimisation problem, is discussed in the next section.

For the multiclass extension of the concepts presented here, see [62].

2.3 Multilinear Discriminant Analysis

This is actually a particular case of linear classification, and not, as perhaps the name may suggest, a more general kind of classification. The basic idea is to use the existing structure in the feature vector that exists naturally in many of these applications. For example, in the case of imaging applications, the sensor measurements are naturally arranged in a matrix. It is the relation between the values of the neighboring pixels that determines the nature of the image. It is only natural to try to use the structure of the data while doing dimensionality reduction. This means that the optimization of the Fisher criterion could be done over a subclass of linear maps instead of the entire class of linear maps as in the case of (2.8). This was done first for the 2D (matrix) case, and then for more general tensor structures of the data. We review here successively both cases.

2.3.1 2D Linear Discriminant Analysis

Indeed, assume that the objects that are to be classified are represented as matrices of dimensions $d \times d$, that is the vector x in R^{d^2} is organized as a matrix $X \in R^{d \times d}$. A particular kind of linear map from R^N (in this case $N = d^2$) to R can be defined by using two vectors $\mathbf{a}$ and $\mathbf{b}$ in R^d and defining

$$v_{\mathbf{a},\mathbf{b}}(X) = \mathbf{a}^T X \mathbf{b}. \tag{2.10}$$

Now, instead of having to find a vector $\mathbf{w}$ in $R^N = R^{d^2}$ that is d^2 components, we have to find two vectors $\mathbf{a}$ and $\mathbf{b}$ in R^d, that is $2d$ components such that classification can be realized as explained before, for example, by maximizing the Fisher criterion. As d is typically a large number, this is a great simplification because we have to solve an optimization problem with $2d$ variables instead of d^2 variables, but the problem is the same: Find $\mathbf{a}, \mathbf{b}$ in R^d that maximize

$$J(\mathbf{a}, \mathbf{b}) = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2}, \tag{2.11}$$

where

$$m_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{a}^T X_i^1 \mathbf{b},\ m_2 = \frac{1}{n_2} \sum_{j=1}^{n_2} \mathbf{a}^T X_j^2 \mathbf{b},$$

and

$$\sigma_1^2 = \sum_{i=1}^{n_1} (\mathbf{a}^T X_i^1 \mathbf{b} - m_1)^2,\ \sigma_2^2 = \sum_{j=1}^{n_2} (\mathbf{a}^T X_j^2 \mathbf{b} - m_2)^2.$$

Introducing these expressions in (2.11), after some simple manipulation, the expression of the Fisher criterion can be written as

$$J(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T (\bar{X}_1 - \bar{X}_2) \mathbf{b} \mathbf{b}^T (\bar{X}_1 - \bar{X}_2)^T \mathbf{a}}{\sum_{j=1}^{2} \sum_{i=1}^{n_j} \mathbf{a}^T (X_i^j - \bar{X}_j) \mathbf{b} \mathbf{b}^T (X_i^j - \bar{X}_j)^T \mathbf{a}}, \tag{2.12}$$

with

$$\bar{X}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} X_i^j, j = 1, 2$$

Just as before, scaling $\mathbf{a}$ and $\mathbf{b}$ has no effect on the value of J so we add the constraint $\mathbf{a}^T \mathbf{a} = 1$ and $\mathbf{b}^T \mathbf{b} = 1$.

This is termed in [58, 114] the bilateral 2D-LDA to distinguish it from a further simplification of the method, the unilateral 2D-LDA, that uses $\mathbf{a} = \mathbf{b}$, further reducing the dimensionality of the optimization problem. However, notice that it is not necessary that the two vectors have the same dimensions. Indeed, the data itself may not consists of

square matrices. If the data consists of data in $R^{a\times b}$ for some numbers a, b not necessarily equal, then the map (2.10) is defined for arbitrary vectors $\mathbf{a} \in R^a$ and $\mathbf{b} \in R^b$.

Not only does the 2D-LDA provide a serious dimensional reduction for the optimization problem that has to be solved, but it also solves the small sample size problem that affects the 1D-LDA method in case of imaging applications. Since, the optimization problem associated with the 2D-LDA problem is a constrained version of the optimization problem associated with the 1D-LDA problem, it would appear that the performance provided by the classifier obtained from the 2D-LDA method has to be less than the performance of the classifier obtained through the 1D-LDA method. However, reference [114] gives a quite generic conditions for the case of imaging applications under which the 2D-LDA classifier is Bayes optimal and it also demonstrates its efficiency with some experimental results.

Just as in the case of 1D-LDA, the 2D-LDA can be extended to reduce the dimensionality of the data to a dimension larger than one. In this case, the problem to be solved is to find two matrices with orthonormal columns $A \in R^{a\times n_a}$ and $B \in R^{b\times n_b}$that maximize the criterion

$$J(A,B) = \frac{\text{trace}\left(A^T(\bar{X}_1 - \bar{X}_2)BB^T(\bar{X}_1 - \bar{X}_2)^T A\right)}{\sum_{j=1}^{2}\sum_{i=1}^{n_j}\text{trace}\left(A^T(X_i^j - \bar{X}_j)BB^T(X_i^j - \bar{X}_j)^T A\right)}. \tag{2.13}$$

For an extensive treatment of this problem, we refer to [112], where 2DLDA is considered separately and in combination with 1D-LDA. This latter combination entails a two phase data reduction procedure with a first 2DLDA phase that reduces the data to vector form, and a second 1D-LDA phase for further reducing the vector data.

2.3.2 M-D Linear Discriminant Analysis

In many applications, the data is naturally structured in multidimensional array. In this case, it is appealing to look at the data as representing multimode tensors and define the feature extraction procedure in terms of tensor algebra operations. A survey of feature extraction methods based on multidimensional array data and tensor algebra is presented in [66]. It includes all possible situations of reducing tensor data to lower dimensions: tensor

to scalar reduction, that is called Elementary Multilinear Projection, tensor-to-vector and tensor-to-tensor reductions. The last two differ only in the representation of the reduced data: the former presents the reduced data as vectors, whereas the latter presents the reduced data as tensors. The same reference gives also a detailed overview of the different learning algorithms that are based on the different dimension reduction techniques.

Assume the data can be naturally represented as an M-dimensional array, with the dimension of mode K denoted by m_k for $k = 1, \ldots, M$, then the training data consists of arrays $X_i^j \in R^{m_1 \times m_2 \times \ldots \times m_M}$ for $j = 1, 2$ and $i = 1, \ldots, n_j$ where n_j are the number of elements in class j. A map from the data space $R^{m_1 \times m_2 \times \ldots \times m_M}$ to R is defined by a set of vectors $\mathbf{a}^k \in R^{m_k}$and is defined as

$$v_{\{\mathbf{a}^k\}_k}(X) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \ldots \sum_{i_M=1}^{m_M} X_{i_1,i_2,\ldots,i_M} a_{i_1}^1 a_{i_2}^2 \ldots a_{i_M}^M \tag{2.14}$$

Using this map, it is possible to define the Fisher criterion as above in the 1D and 2D case, and subsequently, the M-D LDA problem is to find the vectors $a^k \in R^{m_k}$of unit norm that maximize the Fisher criterion

$$J(\{\mathbf{a}^k\}_k) = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2}, \tag{2.15}$$

where

$$m_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} v_{\{\mathbf{a}^k\}_k}(X_i^1),\ m_2 = \frac{1}{n_2} \sum_{j=1}^{n_2} v_{\{\mathbf{a}^k\}_k}(X_i^2),$$

and

$$\sigma_1^2 = \sum_{i=1}^{n_1} (v_{\{\mathbf{a}^k\}_k}(X_i^1) - m_1)^2,\ \sigma_2^2 = \sum_{j=1}^{n_2} (v_{\{\mathbf{a}^k\}_k}(X_i^2) - m_2)^2.$$

The general formulation of the optimization problem in the case that the Fisher criterion is used to reduce the dimension of the data from $R^{m_1 \times m_2 \times \ldots \times m_M}$ to $R^{m_1' \times m_2' \times \ldots \times m_M'}$ where some of the m_i' may be 1 can be found, for example, in [59]. An extensive exposition of the general case of tensor subspace discriminant analysis, including theoretical, numerical aspects and applications can be found in the monograph [63].

2.4 Nonlinear Extensions

It is known that linear classifiers may fail if the data is distributed in such a way that it can only be separated after a nonlinear transformation. This problem is typically approached by "kernelization" and was used in various classification methods. Of course, the essential step of such a method is to properly choose the kernel, or the nonlinear transformation that allows for data separation, or data reduction. A fairly new survey of such methods, with application to face recognition, is presented in [57]. An interesting relatively early development in this direction is presented in [90] which uses polynomial kernels in conjunction with a genetic algorithm optimization algorithm to find the optimal kernel for the given data.

Nonlinear extension of LDA could be implemented as follows: (1) employ a nonlinear transformation mapping data items into a feature space in which the transformed data items could be separated linearly; (2) formulate LDA in this feature space. Such non-linearization strategy is usually implemented by employing a kernel function to define the above non-linear mapping, say $k : R^N \times R^N \longrightarrow R, (x, y) \mapsto k(x, y)$, N is features number. The resulting feature space is a so-called Reproducing Kernel Hilbert Space [70]. Given M data points $\{x_1, ...x_M\} \in R^N$, we define the nonlinear map $\Phi : R^N \longrightarrow R^M (M \geq N)$: $\Phi(x) = \{k(x, x_1), k(x, x_2), ., k(x, x_M)\}(M \geq N)$. We then define the Firsher criterion as follows:

$$J_\Phi(W) = \frac{\text{trace} W^T S_m^\Phi W}{\text{trace} W^T S_s^\Phi W}. \tag{2.16}$$

where the class means, $\bar{\mathbf{x}}_\Phi^j = \frac{1}{n_j} \sum_{i=1}^{n_j} \Phi(\mathbf{x}_i^j), j = 1, 2,$

$$S_m^\Phi = \left(\bar{\mathbf{x}}_\Phi^1 - \bar{\mathbf{x}}_\Phi^2\right) \left(\bar{\mathbf{x}}_\Phi^1 - \bar{\mathbf{x}}_\Phi^2\right)^T$$

and

$$S_s^\Phi = \sum_{i=1}^{n_1} (\Phi(\mathbf{x}_i^1) - \bar{\mathbf{x}}_\Phi^1)(\Phi(\mathbf{x}_i^1) - \bar{\mathbf{x}}_\Phi^1)^T + \sum_{i=1}^{n_2} (\Phi(\mathbf{x}_i^2) - \bar{\mathbf{x}}_\Phi^2)(\Phi(\mathbf{x}_i^2) - \bar{\mathbf{x}}_\Phi^2)^T.$$

Different kernel functions are used for various applications and many applications can be

found in papers such as [24, 57, 67] and many others. Sometimes, the optimization of the Fisher criterion is accompanied by the adaptation of the kernel function to improve the overall performance such as in [24]. The same idea was applied in [57] to propose the Adaptive Quasiconformal Kernel Discriminant Analysis method specifically for face recognition applications.

Note that in the above, LDA operates on N-dimensional vectors and kernel LDA M-dimensional vectors. For tensor LDA, however, it operates on higher-order tensors. Therefore, the kernel LDA should also operate on higher-order tensors which requires a kernel defined on a product space of higher-order tensor. This is beyond the scope of this PhD. But our approach to tensor LDA consists of (1) project higher-order tensors onto low-dimensional feature vectors through a sequence of rank-1 projections; (2) define Fisher criterion on those feature vectors. Therefore, kernalization of our tensor LDA can adopt the same strategy as discussed in the above.

2.5 Algorithms for data reduction and LDA Classifiers

Given the motivation for using LDA classifiers for high dimensional data, as explained before, there has been a lot of interest in developing performant numerical algorithms to solve the optimization problem associated with the training procedure of LDA classifiers. The optimization of the Fisher criterion is a nonlinear optimization problem, so solving it efficiently, in particular, of data of high dimensions, is a challenging problem. This is in contrast to the dimensionality reduction problem using PCA. As mentioned before, the PCA dimensional reduction is essentially achieved using Singular Values Decomposition, for which a very efficient numerical algorithms have been developed. By selecting the first k largest singular values, PCA is able to offer dimensionality reduction to any desired lower dimension, provided that there is a size gap between the first k singular values and

the rest. Achieving a similar capability for data classification purposes is a much harder problem that has preoccupied many researchers in the field. An early solution to this problem was offered already in [35] in the form of optimal discriminant vectors. This solution, referred in the literature as the Foley-Sammon discriminant transformation was modified and improved in many ways over the years. Such developments can be found in [26] and [54] to name only a few of the papers that pursued this line of research.

Whereas the approach started by Foley and Sammon is applicable to the 1D-LDA and 2D-LDA cases, for higher order tensors a different approach is necessary, especially since tensor classifiers are typically used in applications where the number of available features is enormous. Essentially, the numerical methods that were proposed to solve this problem can be classified into methods that approach directly the problem of maximizing the Fisher criterion as expressed in (2.15) and that are called Scatter Ratio Maximization methods, and methods that aim to maximize the difference

$$J_d(\{\mathbf{a}_k\}_k) = (m_1 - m_2)^2 - \lambda(\sigma_1^2 + \sigma_2^2), \tag{2.17}$$

where λ is a scalar parameter that is usually found by cross-validation. The methods based on the maximization of the criterion (2.17) are called Scatter Difference Maximization methods.

One of the early approaches to this problem was proposed in [88] specifically for image processing makes use of a higher order tensor extension of the SVD, called the tensor rank problem, and consequently this approach was called the Tensor Rank 1 Analysis (TR1A). This approach was extended and improved in [94] and called Tensor Rank 1 Discriminant Analysis (TR1DA). The TR1A and TR1DA methods are both Scatter Difference Maximization methods. A very similar approach was presented in [7]. By contrast the method proposed in [65] is a Scatter Ratio Maximization method called Uncorrelated Multilinear Discriminant Analysis (UMLDA) as it aims at obtaining data reduction to a set of uncorrelated features. All these methods are performing tensor-to-

vector reduction in the sense that the reduced data set is in vector form. An overview of these methods, together with some tensor-to-tensor reduction methods is presented in [66]. However, the development of tensor-to-tensor reduction methods has continued and [59] proposes a few new ones, while comparing their performance with some of the previously existing methods. Thus, the Scatter Difference Maximization method for tensor-to-tensor reduction proposed in [95] and called General Tensor Discriminant Analysis (GTDA) is improved to eliminate an iteration step and the new method is called Direct General Tensor Discriminant Analysis. Also, the Scatter Ratio Maximization method for tensor-to-tensor reduction proposed in [109] and called Discriminant Analysis with Tensor Representation (DATER) is similarly improved and the new method is called Constrained Multilinear Discriminant Analysis (CMDA). The paper [59] proves that the proposed methods DGTDA and CMDA have some useful convergence properties.

A practical approach to data reduction for classification purposes is to perform in a first phase a general data reduction step using a variant of PCA, or ICA, followed by the classification specific data reduction algorithm such as LDA. This approach has the advantage that the more complex LDA algorithm will be applied to a lower dimensional data obtained from the first step. A higher order version of this two step approach is presented in [79]. The first phase of dimensionality reduction is performed using the Tucker decomposition, or its particular cases, such as the PARAFAC and NTD that we mentioned before in section 1.1. The second phase of dimensionality reduction is performed using a higher order discriminant analysis algorithm and the paper presents different ways of combining the two steps in the data reduction workflow. However, the algorithms proposed in the cited reference are not sequential so are requiring very serious computational efforts in the case of large data sets.

2.6 Greedy algorithms for data reduction

An important trend in the development of algorithms for data reduction has been the development of stepwise or greedy methods. As one of the contributions of this thesis has been specifically in this area, here is a short account of the prior developments in this area. The earliest references that we are aware in this direction is [109] that proposes a Tensor-to-Tensor projection approach.

A Tensor-to-Vector projection approach is taken in [43] and applied to face recognition. The optimization criterion used for the learning phase in this reference is not the same as the Fisher criterion, but quite similar. For a single feature extraction, the criterion is

$$F_{\{\mathbf{a}^k\}_k} = \frac{\sum_{i=1}^{N^1}\sum_{j=1}^{N^2}(v^{1,i}-v^{2,j})^2}{\sum_{i=1}^{N^1}\sum_{j=1}^{N^1}(v^{1,i}-v^{1,j})^2+\sum_{i=1}^{N^2}\sum_{j=1}^{N^2}(v^{2,i}-v^{2,j})^2}, \tag{2.18}$$

where, using the notation (2.14) introduced above

$$v^{j,i} = v_{\{\mathbf{a}^k\}_k}(X_i^j).$$

In addition, the generating vectors for the different features are required to satisfy an orthogonality constraint that is motivating by the aim of extracting features that are independent of each other. Due to this additional constraint, this approach was termed Orthogonal-Rank-One (ORO) data reduction.

A similar algorithm was proposed in [106] for an additive variant of the cost criterion used in [43] and was called Local Discriminative Orthogonal Rank-One Tensor Projection (LDOROTP) as it also included weighting functions in the criterion in order to capture local discriminant information. Abstract versions of the greedy feature extraction algorithm were presented in [49, 50].

The numerical algorithms proposed in [43] and [106] are based on reducing the problem to (generalized) eigenvalue problems of special matrices (or matrix pencils). In contrast to contrast to these references, the EGFE method developed in this thesis does use gradient

search methods for the optimization which has the advantage of being more generally applicable. For large scale problems, as those envisaged here, computational performance may actually be better than using eigenvalue computations. Although orthogonality constraints were not included in this work since there is no convincing example that they improve effectively the quality of the extracted features. However, it is not difficult to extend our method to include these constraints as well as additional constraints to the optimization problem.

2.7 Application Specific Developments

As explained before imaging applications are most interesting for the development of LDA classifiers. They involve objects described by a very large amount of data and the potential for practical applications is constantly increasing.

Arguably, one of the areas that attracted most of the attention as a great field for demonstrating the power of automatic classification method is face recognition. The problem has received attention for a very long time. The survey article [83] lists 47 references, some of them as old as the seventies. Apparently, the first paper that uses Linear Discriminant Analysis for face recognition is [9] and proposes a method that they call Fisherfaces. They show that it performs much better than a previously developed method based on PCA, and that was called Eigenfaces. This method was further improved over the years. For example, [60] proposes two improved version of Fisherfaces. Another improvement on the Fisherfaces method is presented in [45] based on an improvement of the Foley-Sammon discriminant transformation. All these papers combine in different ways dimensionality reduction through PCA at an early stage, and LDA dimensionality reduction for classification at a later stage. However, they all use the original image representation. By contrast, [61] uses a Gabor wavelet representation of the original image instead of the raw pixel information. Also based on the Gabor wavelet representation, but applying higher order tensor classification methods is the aim of [95], where it is shown

that these methods can improve significantly the classification performance with respect to 1D-LDA and 2D-LDA. In [67], a nonlinear enhancement of LDA in the form of an implicit kernel is used in order to improve the robustness of the method. A different improvement of the LDA approach for face recognition is applied in [110], based on tensor subspace classification and using a method that they call k-mode optimization for iterative learning. In [97], color information is used to improve face recognition, effectively working with a 3D tensor. However, the paper unfolds the data in 2D and uses 2D-LDA to perform the classification. An improvement of the 2D-LDA method for face recognition is proposed in [91] that attempts not only to maximize the Fisher criterion, but also to minimize the cross-correlation between the features in the reduced set. The same goal is pursued in the more recent publication [105]. A tensor discriminant approach to face recognition in color images is presented in [102]. Many recent contributions such as [8, 76] apply local versions of multilinear discriminant analysis to the face recognition problem.

Besides face recognition, there are many other fields of application for automatic classification and for which LDA is highly relevant. For example, an application of these methods for chemical spectroscopic analysis is reported in [52]. An application to the classification of tissues based on gene expression data, a step that is essential in medical diagnosis of various diseases is presented in [113].

The analysis of EEG data using multilinear LDA is explored in [41]. The paper compares an approach based on PCA reduction followed by a vector-based LDA algorithm with an algorithm using multilinear LDA with subspace constraints and shows that the multilinear approach provides superior performance. A recent contribution to the same area can be found in [80].

Functional MRI (fMRI) has also been a very interesting field of application for machine learning in general and tensor LDA in particular. In the case of fMRI, the raw features are voxels, which are naturally structured as a three dimensional array. The scarcity of human expertise in the field makes fMRI a natural application for machine learning, and there have been a lot of interest in the area in the past few years. One such contribution was reported

in [31]. The survey article [77], that discusses several machine learning classification techniques for fMRI applications, including Fisher's LDA, does not discuss the possibility of using tensor techniques in this field. However, it presents many interesting comparisons between different classifiers such as Logistic Regression, Gaussian Naive Bayes, SVM, and LDA and conclude that LDA can be a good choice for a classifier although it is seriously affected by the "small sample size" problem mentioned above. The recommended solution for this issue is a dimensionality reduction of the data before applying the LDA classifier. An earlier survey article [36] on the application of machine learning in fMRI research does not mention LDA at all, but it is a useful reference for the problem formulation and the type of techniques used in dimensionality reduction and classification of fMRI data. A special LDA training method that is capable of using the spatial structure of the features in the case of fMRI was proposed in [74] under the name Spatially-smooth Sparse LDA (SSLDA) without explicitly using tensor techniques. A more recent survey article comparing machine learning classification algorithms for the more general field of brain imaging that contains fMRI is [55]. Even as the cited reference discusses LDA and several versions like regularized LDA, it does not mention any application of tensor-based LDA. The first reference that uses tensor LDA to the analysis of fMRI data appears to be [5]. This particular area of research has received more interest later as reflected in contributions such as [20] and [47].

2.8 Research Questions

The key research questions addressed in this thesis are listed below together with a brief presentation of their answer.

- **How can the Linear Discriminant Analysis methodology to feature extraction be formulated so that it allows the efficient greedy extraction, i.e. in a stepwise way? Can this be formulated for data organized in tensor of arbitrary order?**

The primary goal of this work is on development of an efficient algorithm to extract a small number of features from higher-order tensor data while ensuring that the extracted features are adequate for the targeted classification task. A typical challenge in machine learning applications to biomedical problems is so-called small-sample size (SSS) problem. That is, the number of training samples could be much smaller than the dimension of such samples. This would cause over-fitting. Such problems are likely to occur for high-dimensional vector data and could become more severe for higher-order tensor data. The primary solution is dimension reduction via tensor decomposition. For discriminative dimensional reduction, however, the problem remains because it is still driven by classification tasks. One possible solution to this problem is to combine greedy methods with supervised dimension reduction. That is, one instead tackles a number of smaller dimension reduction problems at single greedy steps. However, inclusion of greedy strategy could significantly increase the algorithm complexity. To avoid this in a greedy approach to tensor LDA, the research question is to find an efficient way to condition each greedy step on its previous steps. The Efficient Greedy Feature Extraction method, that we proposed here and is presented in Chapter 3.

- **How can privileged information in the form of fMRI data be used in the training of classifiers to detect Mild Cognitive Impairment patients from healthy individuals based solely on the result of cognitive test data?**
  This question is motivated by the fact that fMRI data are much more costly to collect than cognitive test data. For some patients, the collection of fMRI data may not even be possible due to the presence of internal metal devices. Therefore, although they are much better in diagnosing, it is important to avoid if possible the use of fMRI data in practice. By using this data only in the learning phase, a methodology known as Learning with Privileged Information (LPI, see e.g. [100]), it should be possible in principle to improve the performance of classifiers based only on cognitive test data. In Chapter 4, we show that fMRI data can indeed be

used effectively as privileged information for training a Generalized Matrix Learning Vector Quantization (GMLVQ) classifier. The use of GMLVQ classifier was suggested by the previous application of the LPI approach for training such a classifier in [38]. Also, Support Vector Machine (SVM) approaches are used to compare the performance with GMLVQ. Initially, the fMRI data is organized as 2D matrix. Our greedy feature extraction algorithm is used to extract efficiently/successively the discriminating feature, one at each step from the fMRI data. The performance of the classifier using privileged information is compared with the baseline classifier using reliable statistical tests and the improvement is clearly established.

2.9 Chapter Summary

This chapter presents the background for the research presented in the rest of the paper. It starts with a presentation of linear discriminant analysis theory, both for classification and for feature extraction. The exposition continues with higher dimensional extensions of linear discriminant analysis, i.e. multilinear discriminant analysis and some nonlinear extensions.

The theoretical developments are complemented by the advances in numerical algorithms. Further, the numerous areas of applications are briefly reviewed. Without trying to be exhaustive, the presentation concentrates on applications to imaging data, in particular for medical applications. Applications to fMRI data processing are paid special attention as they are closely related to the work reported in this thesis.

CHAPTER 3

A NOVEL FRAMEWORK FOR EXTRACTING MULTIPLE FEATURES FROM TENSOR DATA IN A GREEDY WAY

3.1 Introduction

The theoretical basis of the method proposed to reduce the dimension of tensor data in order to simplify and improve classification. The principle of the method is schematically represented in Figure 3.1 and is based on the idea of successively determining the elements of the reduced vector data by applying at each step a tensor-to-scalar data reduction algorithm. If L denotes the order of the tensor data, each of the tensor-to-scalar data reduction step consists of determining a set of L vectors of unit norm $\{\mathbf{a}_\ell\}_{\ell=1,L}$ that are used to reduce the tensor data M to a scalar feature v through the inner-outer product operation

$$v = M \cdot (\mathbf{a}_1 \circ \mathbf{a}_2 \circ \ldots \circ \mathbf{a}_L)$$

in such a way that the separability between the classes represented by the tensor data is maximized. We call this set of a vectors $\mathbf{a}_\ell$, a feature generating vector set. Each feature correspond to a different feature generating vector set

The data that is used for the training consists of a set of tensors separated in two classes: $\{M^{1,i}, i = 1, \ldots, N^1\}$, and $\{M^{2,i}, i = 1, \ldots, N^2\}$. The outcome of the proposed

method is a number of feature generating vector sets $\{\mathbf{a}_{\ell,d}\}_{\substack{\ell=1\cdots L,\\ d=1\cdots D}}$, where D is the number of features, that realize the dimension reduction of the original tensor data to vector data in R^D. To simplify notation, we will omit the index d in the case that $d = 1$.

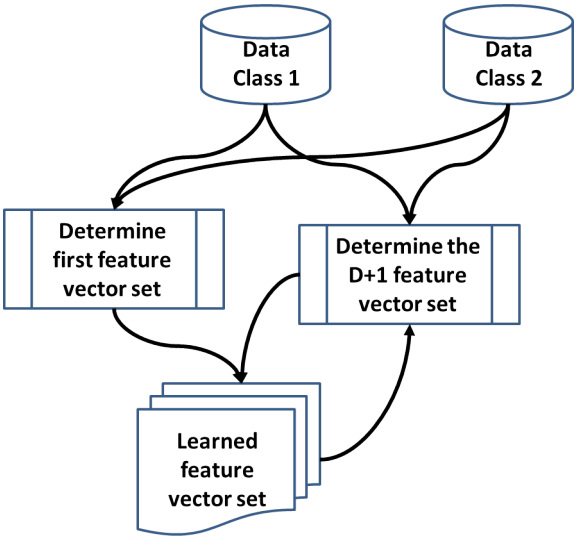

Figure 3.1: Schematic data flow for the proposed tensor-to-vector data reduction EGFE method, Data Class 1 and Data class 2 represent the tensor data used in the training process.

We distinguish in the scheme of Figure 3.1 between the process of finding the first set of feature generating vectors, that is a pure tensor-to-scalar reduction scheme, and the process of finding the subsequent sets of feature generating vectors. The latter has to keep track of the previously determined sets and maximize the separability of the two classes in a multidimensional space. This is the basic idea of our Efficient Greedy Feature Extraction method (EGFE).

The resulting feature generating vector sets are used for data reduction in the manner illustrated in Figure 3.2, with each set used to generate one element of the reduced vector data.

Following the Fisher methodology, the separability of the classes is maximized by maximizing the distance between the means of the data within the classes, while minimizing the variability of the data within the classes. As usual the variability of the data within

method is a number of feature generating vector sets $\{\mathbf{a}_{\ell,d}\}_{\substack{\ell=1\cdots L,\\ d=1\cdots D}}$, where D is the number of features, that realize the dimension reduction of the original tensor data to vector data in R^D. To simplify notation, we will omit the index d in the case that $d = 1$.

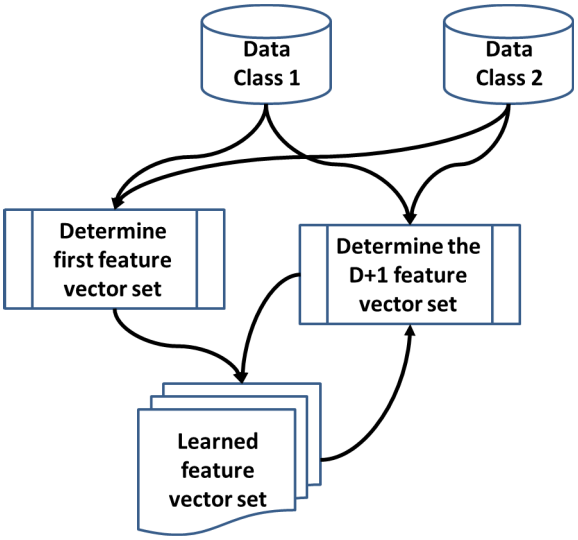

Figure 3.1: Schematic data flow for the proposed tensor-to-vector data reduction EGFE method, Data Class 1 and Data class 2 represent the tensor data used in the training process.

We distinguish in the scheme of Figure 3.1 between the process of finding the first set of feature generating vectors, that is a pure tensor-to-scalar reduction scheme, and the process of finding the subsequent sets of feature generating vectors. The latter has to keep track of the previously determined sets and maximize the separability of the two classes in a multidimensional space. This is the basic idea of our Efficient Greedy Feature Extraction method (EGFE).

The resulting feature generating vector sets are used for data reduction in the manner illustrated in Figure 3.2, with each set used to generate one element of the reduced vector data.

Following the Fisher methodology, the separability of the classes is maximized by maximizing the distance between the means of the data within the classes, while minimizing the variability of the data within the classes. As usual the variability of the data within

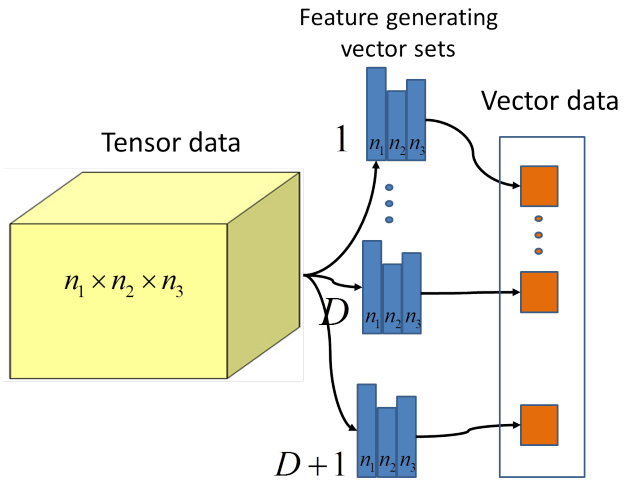

Figure 3.2: Reduction of tensor data to vector data using the feature generating vector sets that were determined by the proposed supervised learning process.

the classes is measured by the total squares variation. In order to reduce this simultaneous maximize-minimize problem to a single optimization problem, we use two conventional approaches . The first is the multiplicative approach that attempts at maximizing the ratio of the two quantities. The second is the additive approach that attempts to maximizing a weighted difference between the two quantities. In each case, maximization will tend to maximize the numerator, respectively the positive term, and minimize the denominator, respectively, the negative term. We will choose the one that works better.

In both cases, the data reduction algorithm reduces to a succession of optimization problems that can be solved in principle by any numerical algorithm. In this chapter, we clarify the implementation of these algorithms using gradient ascent, and therefore it is essential to derive the expression of the gradients of the optimization criterion with respect to the elements of the feature generating vectors. More details about the implementation of the optimization algorithms are presented in Section 3.5.

3.2 Multiplicative Criterion Case

As explained before, the training process has two parts. The first part consists of determining the first set of feature generating vectors and solves the following optimization

problem.

Problem 1. (Single feature extraction) Find the feature generating vectors $\mathbf{a}_1, \ldots, \mathbf{a}_L$, of unit norm, that maximize the multiplicative Fisher criterion

$$F_m = \frac{(m^1 - m^2)^2}{S^1 + S^2}, \tag{3.1}$$

where m^1, m^2 are defined as:

$$m^k = \frac{1}{N^k} \sum_{i=1}^{N^k} v^{k,i},\ k = 1, 2. \tag{3.2}$$

S^1 and S^2 are defined as:

$$S^k = \sum_{i=1}^{N^k} (v^{k,i} - m^k)^2,\ k = 1, 2. \tag{3.3}$$

The second part consists of determining the subsequent sets of feature generating vectors, assuming that a number of such sets $\{\mathbf{a}_{\ell,d}\}_{\substack{\ell = 1 \cdots L, \\ d = 1 \cdots D}}$ are already available and solves the following optimization problem.

Problem 2. (Subsequent features extraction) Given the first $D \geq 1$ sets of feature generating vectors, find the $D + 1$ set of feature generating vectors $\mathbf{a}_{1,D+1}, \ldots, \mathbf{a}_{L,D+1}$, of unit norm, that maximize the total multiplicative Fisher criterion

$$F_{mD} = \frac{\sum_{d=1}^{D+1} (m_d^1 - m_d^2)^2}{\sum_{d=1}^{D+1} (S_d^1 + S_d^2)} = \frac{\sum_{d=1}^{D} (m_d^1 - m_d^2)^2 + (m_{D+1}^1 - m_{D+1}^2)^2}{\sum_{d=1}^{D} (S_d^1 + S_d^2) + S_{D+1}^1 + S_{D+1}^2}, \tag{3.4}$$

where m_d^1, m_d^2 are defined as:

$$m_d^k = \frac{1}{N^k} \sum_{i=1}^{N^k} v_d^{k,i},\ k = 1, 2 \tag{3.5}$$

S_d^1 and S_d^2 are defined as:

$$S_d^k = \sum_{i=1}^{N^k} (v_d^{k,i} - m_d^k)^2,\ k = 1, 2. \tag{3.6}$$

These optimization problems can be solved in principle with any numerical algorithm for solving constrained optimization problem. We will go later in some detail about this, but no matter which numerical optimization method is chosen, it is required to be able to derive the gradient of the cost functions with respect to the decision variables, that are the elements of the feature generating vectors. Therefore, in the next two sections, we will derive expressions for the gradients of the cost functions (3.1) and (3.4).

3.2.1 Gradient Expression for Determining the First set of Feature Generating Vectors

Let ℓ be a fixed mode. The gradient of F_m with respect to the vector $\mathbf{a}_\ell$,

$$\nabla_{\mathbf{a}_\ell} F_m = \frac{2(m^1 - m^2)(S^1 + S^2)\nabla_{\mathbf{a}_\ell}(m^1 - m^2) - (m^1 - m^2)^2 \nabla_{\mathbf{a}_\ell}(S^1 + S^2)}{(S^1 + S^2)^2} \tag{3.7}$$

The gradients of m^k and S^k, $k = \{1, 2\}$ are given by the formulas

$$\nabla_{\mathbf{a}_\ell} m^k = \frac{1}{N^k} \sum_{i=1}^{N^k} \nabla_{\mathbf{a}_\ell} v^{k,i}, \tag{3.8}$$

$$\nabla_{\mathbf{a}_\ell} S^k = \sum_{i=1}^{N^k} 2(v^{k,i} - m^k)(\nabla_{\mathbf{a}_\ell} v^{k,i} - \nabla_{\mathbf{a}_\ell} m^k). \tag{3.9}$$

These formulas can be readily computed if the gradient of $v^{k,i} = \nabla_{\mathbf{a}_\ell} M^{k,i} \cdot (\mathbf{a}_1 \circ \mathbf{a}_2 \circ \ldots \circ \mathbf{a}_L)$ are known. The partial derivative of $v^{k,i}$ with respect to the jth coordinate of $\mathbf{a}_\ell$ is

$$\begin{aligned}\frac{\partial}{\partial a_{\ell,j}} f^{k,i} &= \frac{\partial}{\partial a_{\ell,j}} M^{k,i} \cdot (\mathbf{a}_1 \circ \mathbf{a}_2 \circ \ldots \circ \mathbf{a}_L) \\ &= \frac{\partial}{\partial a_{\ell,j}} \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} \ldots \sum_{q_L=1}^{n_L} M^{k,i}_{q_1,q_2,\ldots,q_L} a_{1,q_1} a_{2,q_2} \ldots a_{L,q_L} \\ &= \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} \ldots \sum_{q_{\ell-1}=1}^{n_{\ell-1}} \sum_{q_{\ell+1}=1}^{n_{\ell+1}} \ldots \sum_{q_L=1}^{n_L} M^{k,i}_{q_1,q_2,\ldots q_{\ell-1},j,q_{\ell+1},\ldots,q_L} \prod_{\substack{p=1 \\ p \neq \ell}}^{L} a_{p,q_p}\end{aligned}$$

To write this formula in a compact manner, let us denote by $M^{k,i}_{q_\ell=j}$, the tensor of order $L-1$ obtained from $M^{k,i}$ by fixing the index of the ℓ mode to j, i.e.

$$\left(M^{k,i}_{q_\ell=j}\right)_{q_1,q_2,\ldots q_{\ell-1},q_{\ell+1},\ldots,q_L} = M^{k,i}_{q_1,q_2,\ldots q_{\ell-1},j,q_{\ell+1},\ldots,q_L} \tag{3.10}$$

for $q_i = 1,\ldots,n_i, i = 1,\ldots,L,\ i \neq j$. Notice that $M^{k,i}_{q_\ell=j}$ is an element of $R^{n_1 \times n_2 \ldots n_{\ell-1} \times n_{\ell+1} \ldots \times n_L}$. Also introduce the outer product

$$A_{-\ell} = \mathbf{a}_1 \circ \ldots \circ \mathbf{a}_{\ell-1} \circ \mathbf{a}_{\ell+1} \circ \ldots \circ \mathbf{a}_L \tag{3.11}$$

that is also an element of $R^{n_1 \times n_2 \ldots n_{\ell-1} \times n_{\ell+1} \ldots \times n_L}$. With these notations, the partial derivative of $v^{k,i}$ can be written compactly as

$$\frac{\partial}{\partial a_{\ell,j}} v^{k,i} = M^{k,i}_{q_\ell=j} \cdot A_{-\ell}. \tag{3.12}$$

In a similar manner, we proceed to write compact formula's for each element of formula (3.7). First of all, the difference of the two means can be written as

$$
\begin{aligned}
m^1 - m^2 &= \frac{1}{N^1}\sum_{i=1}^{N^1} v^{1,i} - \frac{1}{N^2}\sum_{i=1}^{N^2} v^{2,i} \\
&= \left(\ \tfrac{1}{N^1}\sum_{i=1}^{N^1} M^{1,i} - \tfrac{1}{N^2}\sum_{i=1}^{N^2} M^{2,i} \ \right) \cdot A
\end{aligned}
$$

and introducing the L order tensor

$$
\Delta = \frac{1}{N^1}\sum_{i=1}^{N^1} M^{1,i} - \frac{1}{N^2}\sum_{i=1}^{N^2} M^{2,i} \tag{3.13}
$$

this becomes

$$
m^1 - m^2 = \Delta \cdot A. \tag{3.14}
$$

Applying formula (3.12), the $j$the component of the gradient with respect to the vector $\mathbf{a}_\ell$ is

$$
\begin{aligned}
\left[\nabla_{\mathbf{a}_\ell}(m^1 - m^2)\right]_j &= \frac{\partial}{\partial a_{\ell,j}} \left(\ \tfrac{1}{N^1}\sum_{i=1}^{N^1} v^{1,i} - \tfrac{1}{N^2}\sum_{i=1}^{N^2} v^{2,i} \ \right) \\
&= \frac{1}{N^1}\sum_{i=1}^{N^1} \frac{\partial}{\partial a_{\ell,j}} v^{1,i} - \frac{1}{N^2}\sum_{i=1}^{N^2} \frac{\partial}{\partial a_{\ell,j}} v^{2,i} \\
&= \ \tfrac{1}{N^1}\sum_{i=1}^{N^1} M^{1,i}_{q_\ell=j} \cdot A_{-\ell} - \tfrac{1}{N^2}\sum_{i=1}^{N^2} M^{2,i}_{q_\ell=j} \cdot A_{-\ell}.
\end{aligned} \tag{3.15}
$$

Introducing the $L-1$ order tensor

$$
\Delta_{q_\ell=j} = \frac{1}{N^1}\sum_{i=1}^{N^1} M^{1,i}_{q_\ell=j} - \frac{1}{N^2}\sum_{i=1}^{N^2} M^{2,i}_{q_\ell=j} \tag{3.16}
$$

this can be written in a compact form

$$
\left[\nabla_{\mathbf{a}_\ell}(m^1 - m^2)\right]_j = \Delta_{q_\ell=j} \cdot A_{-\ell}. \tag{3.17}
$$

Turning now to the total square variations, using (3.12) and (3.17) in (3.9), the $j$the

component of the gradient of S^k with respect to the vector $\mathbf{a}_\ell$is

$$\begin{aligned}\left[\nabla_{\mathbf{a}_\ell} S^k\right]_j &= \sum_{i=1}^{N^k} 2(v^{k,i} - m^k)(\frac{\partial}{\partial a_{\ell,j}} v^{k,i} - \left[\nabla_{\mathbf{a}_\ell} m^k\right]_j) \\ &= \sum_{i=1}^{N^k} 2(v^{k,i} - m^k)(M^{k,i}_{q_\ell=j} \cdot A_{-\ell} - \frac{1}{N^k} \sum_{i=1}^{N^k} M^{k,i}_{q_\ell=j} \cdot A_{-\ell})\end{aligned}$$

Taking the sum for both classes, and factoring out $A_{-\ell}$

$$\sum_{k=1}^{2} \left[\nabla_{\mathbf{a}_\ell} S^k\right]_j = \sum_{k=1}^{2} \sum_{i=1}^{N^k} 2(v^{k,i} - m^k)(M^{k,i}_{q_\ell=j} - \frac{1}{N^k} \sum_{i=1}^{N^k} M^{k,i}_{q_\ell=j}) \cdot A_{-\ell}$$

By introducing the $L-1$ order tensor

$$\Omega^k_{q_\ell=j} = \sum_{k=1}^{2} \sum_{i=1}^{N^k} 2(v^{k,i} - m^k)(M^{k,i}_{q_\ell=j} - \frac{1}{N^k} \sum_{i=1}^{N^k} M^{k,i}_{q_\ell=j}) \tag{3.18}$$

the last expression can be written in the compact form

$$\sum_{k=1}^{2} \left[\nabla_{\mathbf{a}_\ell} S^k\right]_j = \Omega^k_{q_\ell=j} \cdot A_{-\ell} \tag{3.19}$$

With these preparations, returning to formula (3.7),

$$\begin{aligned}[\nabla_{\mathbf{a}_\ell} F_m]_j &= \frac{2(m^1 - m^2)(S^1 + S^2)\left[\nabla_{\mathbf{a}_\ell}(m^1 - m^2)\right]_j}{(S^1 + S^2)^2} \\ &- \frac{(m^1 - m^2)^2\left[\nabla_{\mathbf{a}_\ell}(S^1 + S^2)\right]_j}{(S^1 + S^2)^2} \\ &= \frac{2(\Delta \cdot A)(S^1 + S^2)\Delta_{q_\ell=j} \cdot A_{-\ell} - (m^1 - m^2)^2 \Omega^k_{q_\ell=j} \cdot A_{-\ell}}{(S^1 + S^2)^2} \\ &= \frac{2(\Delta \cdot A)(S^1 + S^2)\Delta_{q_\ell=j} - (m^1 - m^2)^2 \Omega^k_{q_\ell=j}}{(S^1 + S^2)^2} \cdot A_{-\ell}\end{aligned} \tag{3.20}$$

and introducing the $L-1$ order tensor

$$\begin{aligned}\tilde{\Omega}^k_{q_\ell=j} &= \frac{2(\Delta\cdot A)(S^1+S^2)\Delta_{q_\ell=j} - (m^1-m^2)^2\Omega^k_{q_\ell=j}}{(S^1+S^2)^2} \\ &= \frac{1}{S^1+S^2}\left[2(\Delta\cdot A)\Delta_{q_\ell=j} - F_m\ \Omega^k_{q_\ell=j}\right]\end{aligned} \tag{3.21}$$

the j component of the gradient of the Fisher criterion is

$$\left[\nabla_{\mathbf{a}_\ell} F_m\right]_j = \tilde{\Omega}^k_{q_\ell=j}\cdot A_{-\ell} \tag{3.22}$$

3.2.2 Gradient Expressions for Determining Further Feature Generating Vector Sets

Introducing the notations

$$\mathcal{N}_D = \sum_{d=1}^{D}(m^1_d - m^2_d)^2 \tag{3.23}$$

and

$$\mathcal{D}_D = \sum_{d=1}^{D}(S^1_d + S^2_d) \tag{3.24}$$

the expression of the total Fisher criterion (3.4) becomes

$$F_{mD} = \frac{\mathcal{N}_D + (m^1_{D+1} - m^2_{D+1})^2}{\mathcal{D}_D + S^1_{D+1} + S^2_{D+1}}$$

Let ℓ be a fixed mode. Since $\mathcal{N}_D$ and $\mathcal{D}_D$ do not depend on the new feature set, the gradient of F with respect to the vector $\mathbf{a}_{\ell,D+1}$ is

$$\begin{aligned}\nabla_{\mathbf{a}_{\ell,D+1}} F_{mD} = &\frac{2(m^1_{D+1}-m^2_{D+1})\nabla_{\mathbf{a}_{\ell,D+1}}(m^1_{D+1}-m^2_{D+1})}{\mathcal{D}_D + S^1_{D+1} + S^2_{D+1}} \\ &- \frac{(\mathcal{N}_D + (m^1_{D+1}-m^2_{D+1})^2)\nabla_{\mathbf{a}_{\ell,D+1}}(S^1_{D+1}+S^2_{D+1})}{(\mathcal{D}_D + S^1_{D+1} + S^2_{D+1})^2}\end{aligned} \tag{3.25}$$

The gradients of m^k_{D+1} and S^k_{D+1} $k = \{1, 2\}$ are

$$\nabla_{\mathbf{a}_{\ell,D+1}} m^k_{D+1} = \frac{1}{N^k} \sum_{i=1}^{N^k} \nabla_{\mathbf{a}_{\ell,D+1}} v^{k,i}_{D+1}, \tag{3.26}$$

$$\nabla_{\mathbf{a}_{\ell,D+1}} S^k_{D+1} = \sum_{i=1}^{N^k} 2(v^{k,i}_{D+1} - m^k_{D+1})(\nabla_{\mathbf{a}_{\ell,D+1}} f^{k,i}_{D+1} - \nabla_{\mathbf{a}_{\ell,D+1}} m^k_{D+1}). \tag{3.27}$$

Now the computation of the gradient has reduced to the computation of the gradient of the features that is entirely similar to the gradient of the features for a single feature set that was explained in Section 3.2.1

$$\nabla_{\mathbf{a}_{\ell,D+1}} v^{k,i}_{D+1} = \nabla_{\mathbf{a}_{\ell,D+1}} M^{k,i} \cdot (\mathbf{a}_{1,D+1} \circ \mathbf{a}_{2,D+1} \circ \ldots \circ \mathbf{a}_{L,D+1})$$

Introducing the tensor of order $L - 1$

$$A_{-\ell,D+1} = \mathbf{a}_{1,D+1} \circ \ldots \circ \mathbf{a}_{\ell-1,D+1} \circ \mathbf{a}_{\ell+1,D+1} \circ \ldots \circ \mathbf{a}_{L,D+1} \tag{3.28}$$

the coordinates of the feature gradient are given by

$$\frac{\partial}{\partial a_{\ell,j,D+1}} v^{k,i}_{D+1} = M^{k,i}_{q_\ell = j} \cdot A_{-\ell,D+1}. \tag{3.29}$$

In a similar manner, we proceed to write compact formula's for each element of formula (3.25):

$$\begin{aligned} m^1_{D+1} - m^2_{D+1} &= \frac{1}{N^1} \sum_{i=1}^{N^1} v^{1,i}_{D+1} - \frac{1}{N^2} \sum_{i=1}^{N^2} v^{2,i}_{D+1} \\ &= \left(\frac{1}{N^1} \sum_{i=1}^{N^1} M^{1,i} - \frac{1}{N^2} \sum_{i=1}^{N^2} M^{2,i} \right) \cdot A_{D+1} \end{aligned}$$

and using (3.13), this can be written as

$$m^1_{D+1} - m^2_{D+1} = \Delta \cdot A_{D+1}. \tag{3.30}$$

Also, applying formula (3.29),

$$\begin{aligned}
\left[\nabla_{\mathbf{a}_{\ell,D+1}}(m^1_{D+1} - m^2_{D+1})\right]_j &= \frac{\partial}{\partial a_{\ell,j,D+1}} \left(\tfrac{1}{N^1} \sum_{i=1}^{N^1} v^{1,i}_{D+1} - \tfrac{1}{N^2} \sum_{i=1}^{N^2} v^{2,i}_{D+1} \right) \\
&= \frac{1}{N^1} \sum_{i=1}^{N^1} \frac{\partial}{\partial a_{\ell,j,D+1}} v^{1,i}_{D+1} - \frac{1}{N^2} \sum_{i=1}^{N^2} \frac{\partial}{\partial a_{\ell,j,D+1}} v^{2,i}_{D+1} \\
&= \frac{1}{N^1} \sum_{i=1}^{N^1} M^{1,i}_{q_\ell = j} \cdot A_{-\ell,D+1} \\
&\quad - \frac{1}{N^2} \sum_{i=1}^{N^2} M^{2,i}_{q_\ell = j} \cdot A_{-\ell,D+1}
\end{aligned} \tag{3.31}$$

and with the notation (3.16), this can be written as

$$\left[\nabla_{\mathbf{a}_{\ell,D+1}}(m^1_{D+1} - m^2_{D+1})\right]_j = \Delta_{q_\ell = j} \cdot A_{-\ell,D+1}. \tag{3.32}$$

Using (3.29) and (3.32) in (3.27),

$$\begin{aligned}
\left[\nabla_{\mathbf{a}_{\ell,D+1}} S^k_{D+1}\right]_j &= \sum_{i=1}^{N^k} 2(v^{k,i}_{D+1} - m^k_{D+1})(\frac{\partial}{\partial a_{\ell,j,D+1}} v^{k,i}_{D+1} - \left[\nabla_{\mathbf{a}_{\ell,D+1}} m^k_{D+1}\right]_j) \\
&= \sum_{i=1}^{N^k} 2(v^{k,i}_{D+1} - m^k_{D+1})(M^{k,i}_{q_\ell = j} \cdot A_{-\ell,D+1} \\
&\quad - \frac{1}{N^k} \sum_{i=1}^{N^k} M^{k,i}_{q_\ell = j} \cdot A_{-\ell,D+1})
\end{aligned}$$

Taking the sum for both classes, and factoring out $A_{-\ell,D+1}$

$$\sum_{k=1}^{2}\left[\nabla_{\mathbf{a}_{\ell,D+1}}S_{D+1}^{k}\right]_{j}=\sum_{k=1}^{2}\sum_{i=1}^{N^{k}}2(f_{D+1}^{k,i}-m_{D+1}^{k})\\ \times\left(M_{q_{\ell}=j}^{k,i}-\frac{1}{N^{k}}\sum_{i=1}^{N^{k}}M_{q_{\ell}=j}^{k,i}\right)\cdot A_{-\ell,D+1}$$

By introducing the $L-1$ order tensor

$$\Omega_{q_{\ell}=j,D+1}^{k}=\sum_{k=1}^{2}\sum_{i=1}^{N^{k}}2(v_{D+1}^{k,i}-m_{D+1}^{k})\left(M_{q_{\ell}=j}^{k,i}-\frac{1}{N^{k}}\sum_{i=1}^{N^{k}}M_{q_{\ell}=j}^{k,i}\right)\tag{3.33}$$

the last expression can be written in the compact form

$$\sum_{k=1}^{2}\left[\nabla_{\mathbf{a}_{\ell,D+1}}S_{D+1}^{k}\right]_{j}=\Omega_{q_{\ell}=j,D+1}^{k}\cdot A_{-\ell,D+1}\tag{3.34}$$

With these preparations, returning to formula (3.25),

$$\begin{aligned}\left[\nabla_{\mathbf{a}_{\ell,D+1}}F_{mD}\right]_{j}=&\frac{2(m_{D+1}^{1}-m_{D+1}^{2})\left[\nabla_{\mathbf{a}_{\ell,D+1}}(m_{D+1}^{1}-m_{D+1}^{2})\right]_{j}}{\mathcal{D}_{D}+S_{D+1}^{1^{2}}+S_{D+1}^{2^{2}}}\\ &-\frac{(\mathcal{N}_{D}+(m_{D+1}^{1}-m_{D+1}^{2})^{2})\left[\nabla_{\mathbf{a}_{\ell,D+1}}(S_{D+1}^{1^{2}}+S_{D+1}^{2^{2}})\right]_{j}}{(\mathcal{D}_{D}+S_{D+1}^{1^{2}}+S_{D+1}^{2^{2}})^{2}}\\ =&\frac{2(m_{D+1}^{1}-m_{D+1}^{2})\Delta_{q_{\ell}=j,D+1}\cdot A_{-\ell,D+1}}{\mathcal{D}_{D}+S_{D+1}^{1^{2}}+S_{D+1}^{2^{2}}}\\ &-\frac{(\mathcal{N}_{D}+(m_{D+1}^{1}-m_{D+1}^{2})^{2})\Omega_{q_{\ell}=j,D+1}^{k}\cdot A_{-\ell,D+1}}{(\mathcal{D}_{D}+S_{D+1}^{1^{2}}+S_{D+1}^{2^{2}})^{2}}\\ =&\left(\frac{2(m_{D+1}^{1}-m_{D+1}^{2})\Delta_{q_{\ell}=j,D+1}}{\mathcal{D}_{D}+S_{D+1}^{1}+S_{D+1}^{2}}\right.\\ &\left.-\frac{(\mathcal{N}_{D}+(m_{D+1}^{1}-m_{D+1}^{2})^{2})\Omega_{q_{\ell}=j,D+1}^{k}}{(\mathcal{D}_{D}+S_{D+1}^{1}+S_{D+1}^{2})^{2}}\right)\cdot A_{-\ell,D+1}\end{aligned}\tag{3.35}$$

and introducing the $L-1$ order tensor

$$\tilde{\Omega}^k_{q_\ell=j,D+1} = \frac{1}{\mathcal{D}_D + S^1_{D+1} + S^2_{D+1}} \big(2(m^1_{D+1} - m^2_{,D+1})\Delta_{q_\ell=j,D+1} \\ -F_m \Omega^k_{q_\ell=j,D+1},\big) \tag{3.36}$$

the j component of the gradient of the Fisher criterion is

$$\left[\nabla_{\mathbf{a}_{\ell,D+1}} F_{mD}\right]_j = \tilde{\Omega}^k_{q_\ell=j,D+1} \cdot A_{-\ell,D+1} \tag{3.37}$$

3.3 Additive Criterion Case

As in the case of the multiplicative criterion, the training process has two parts. The first part consists of determining the first set of feature generating vectors and solves the following optimization problem.

Problem 3. (Single feature extraction) Find the feature generating vectors $\mathbf{a}_1, \ldots, \mathbf{a}_L$, of l_2 norm, that maximize the additive Fisher criterion

$$F_a = (m^1 - m^2)^2 - \lambda(S^1 + S^2), \tag{3.38}$$

where m^1, m^2 are defined in (3.2) and S^1 and S^2 are defined in (3.3).

The second part consists of determining the subsequent sets of feature generating vectors, assuming that a number of such sets $\{\mathbf{a}_{\ell,d}\}_{\substack{\ell=1\cdots L,\\ d=1\cdots D}}$ are already available and solves the following optimization problem.

Problem 4. (Subsequent features extraction) Find the $D+1$ set of feature generating vectors $\mathbf{a}_{1,D+1}, \ldots, \mathbf{a}_{L,D+1}$, of a vector norm, that maximize the total additive Fisher criterion

$$F_{aD} = \sum_{d=1}^{D+1} (m^1_d - m^2_d)^2 - \lambda \sum_{d=1}^{D+1} (S^1_d + S^2_d), \tag{3.39}$$

where m_d^1, m_d^2 are defined in (3.5) and S_d^1 and S_d^2 are defined in (3.6).

Just as in the case of the multiplicative criterion, solving these optimization problems is dependent on the compuation of the gradients of the cost functions with respect to the decision variables. Therefore, in the next two sections, we will derive expressions for the gradients of the cost functions (3.38) and (3.39).

3.3.1 Gradient Expression for Determining the First Set of Feature Generating Vectors

Let ℓ be a fixed mode. The gradient of F_a with respect to the vector $\mathbf{a}_\ell$,

$$\nabla_{\mathbf{a}_\ell} F_a = 2(m^1 - m^2)\nabla_{\mathbf{a}_\ell}(m^1 - m^2) - \lambda \nabla_{\mathbf{a}_\ell}(S^1 + S^2) \tag{3.40}$$

The terms in the right hand side of relation (3.40) are readily written using the expressions(3.17) and (3.19) and the coordinates of the gradient of F_a are

$$\begin{aligned}
[\nabla_{\mathbf{a}_\ell} F_a]_j &= 2(m^1 - m^2)[\nabla_{\mathbf{a}_\ell}(m^1 - m^2)]_j - \lambda[\nabla_{\mathbf{a}_\ell}(S^1 + S^2)]_j \\
&= 2(m^1 - m^2)\Delta_{q_\ell = j} \cdot A_{-\ell} - \lambda \ \Omega^k_{q_\ell = j} \cdot A_{-\ell} \\
&= 2\Delta \cdot A(\Delta_{q_\ell = j} \cdot A_{-\ell}) - \lambda \ \Omega^k_{q_\ell = j} \cdot A_{-\ell}
\end{aligned} \tag{3.41}$$

and introducing the $L-1$ order tensor

$$\Omega^{*k}_{q_\ell = j} = 2\Delta \cdot A(\Delta_{q_\ell = j} \cdot A_{-\ell}) - \lambda \ \Omega^k_{q_\ell = j} \tag{3.42}$$

the j component of the gradient of the Fisher criterion is

$$[\nabla_{\mathbf{a}_\ell} F_a]_j = \Omega^{*k}_{q_\ell = j} \cdot A_{-\ell} \tag{3.43}$$

3.3.2 Gradient Expressions for Determining Further Feature Generating Vector Sets

With the notations introduced in Subsection 3.2.2, the expression of the total Fisher criterion becomes

$$F_a = \mathcal{N}_D + (m^1_{D+1} - m^2_{D+1})^2 - \lambda \cdot (\mathcal{D}_D + (S^1_{D+1} + S^2_{D+1}))$$

Let ℓ be a fixed mode. Since $\mathcal{N}_D$ and $\mathcal{D}_D$ do not depend on the new feature set, the gradient of F_a with respect to the vector $\mathbf{a}_{\ell,D+1}$ is

$$\begin{aligned} \nabla_{\mathbf{a}_{\ell,D+1}} F_a = \mathcal{N}_D &+ 2(m^1_{d+1} - m^2_{d+1}) \nabla_{\mathbf{a}_{\ell,D+1}} (m^1_{d+1} - m^2_{d+1}) \\ &- \lambda \cdot (\mathcal{D}_D + \nabla_{\mathbf{a}_{\ell,D+1}} (S^1_{d+1} + S^2_{d+1})) \end{aligned} \tag{3.44}$$

Substituting in this expression, the formulas (3.32) and (3.34), the coordinates of this gradient can be written as

$$\begin{aligned} \left[\nabla_{\mathbf{a}_{\ell,D+1}} F_a\right]_j &= \mathcal{N}_D + 2(m^1_{D+1} - m^2_{D+1}) \nabla_{\mathbf{a}_{\ell,D+1}} [(m^1_{D+1} - m^2_{D+1})]_j \\ &\quad - \lambda \cdot (\mathcal{D}_D + \nabla_{\mathbf{a}_{\ell,D+1}} [(S^1_{D+1} + S^2_{D+1})]_j) \\ &= \mathcal{N}_D + 2(m^1_{D+1} - m^2_{D+1}) \Delta_{q_\ell = j, D+1} \cdot A_{-\ell, D+1} \\ &\quad - \lambda \cdot (\mathcal{D}_D + \Omega^k_{q_\ell = j, D+1} \cdot A_{-\ell, D+1}) \\ &= \mathcal{N}_D + 2(\Delta \cdot A_{D+1} \cdot \Delta_{q_\ell = j, D+1} \cdot A_{-\ell, D+1}) \\ &\quad - \lambda \cdot (\mathcal{D}_D + \Omega^k_{q_\ell = j, D+1} \cdot A_{-\ell, D+1}) \\ &= (\mathcal{N}_D + 2(\Delta \cdot A_{D+1} \cdot \Delta_{q_\ell = j, D+1}) \\ &\quad - \lambda \cdot (\mathcal{D}_D + \Omega^k_{q_\ell = j, D+1})) \cdot A_{-\ell, D+1} \end{aligned} \tag{3.45}$$

and introducing the $L-1$ order tensor

$$\Omega^{*k}_{q_\ell=j,D+1} = \mathcal{N}_D + 2(\Delta \cdot A_{D+1} \cdot \Delta_{q_\ell=j,D+1}) - \lambda \cdot (\mathcal{D}_D + \Omega^{k}_{q_\ell=j,D+1}) \tag{3.46}$$

the j component of the gradient of the Fisher criterion is

$$\left[\nabla_{\mathbf{a}_{\ell,D+1}} F_a\right]_j = \Omega^{*k}_{q_\ell=j,D+1} \cdot A_{-\ell,D+1} \tag{3.47}$$

3.4 Algorithm Analysis and Discussions

In [59], a general framework for multi-linear discriminant analysis (MDA) of higher order tensor data has been formulated. This framework can be used to extract multiple discriminative features for classification tasks in non-greedy ways. The authors discussed the convergence issues of their algorithms in great details.

By the taxonomy presented in [59], our EGFE methods can be considered as the greedy version of so-called "Discriminant Analysis with Tensor Representation" (DATER), a subclass of MDA algorithms. To generate n features, DATER would perform a rank-n projection for each mode. For EGFE, we instead perform only rank-1 projections but need to repeat this step n times in a sequence to obtain n features. Note that both the rank-n projection matrix in DATER and the projection vector in EGFE should be jointly optimised with such matrices or vectors from the other modes. This numerical solution to this optimization problem is so-called alternating descent [10]. It is also phrased in [59] as "Block coordinate descent". For individual iterations of this optimization algorithm, one just needs to optimize the projection matrix or vector for single modes. This results in a convex optimization problem. But for all projection matrices/vectors together, the optimization problem is not convex. Indeed, one needs to answer the question whether or not this optimization procedure would converge.

It is reported that DATER does not converge over iterations (see Section 1 in [59]). To deal with this no convergence, the authors introduced a constraint for all projection matrices in DATER, that is, for each of these matrices, its column vectors needs be orthonormal with each other. Under this condition, the sequence of Fisher criterion values generated by the "block coordinate descent" algorithm is an asymptotically bounded sequence (See Theorem 4.2 in [59]). The revised DATER algorithm is referred to as "Constrained MDA" (CMDA). CMDA is the first "scatter ratio maximization"-based (that is, Fisher criterion maximization) MDA method that exhibits convergence. For our EGFE, this implies that all projection vectors must be normed. We have implemented this normalisation constraint in EGFE and thus it does converge (to a local extreme) over

the "block coordinate descent" iterations.

As we can see from the above, the orthogonality constraint actually is not necessary for the greedy approach. But it could be employed in the greedy method. We know that each greedy step must be coupled with its previous steps. Therefore, the two successive greedy steps can be coupled by enforcing the orthogonality of their corresponding projection vectors for each mode. This idea has been implemented in [93]. Recall that in EGFE, the coupling of two greedy steps is done via formulating a Fisher criterion conditioned on the features extracted from all previous steps. We provided a comparison (in chapter 5) between EGFE and orthogonal rank-one tensor projections (ORO) [93] and show that EGFE is comparable with ORO, although it is conceptually and practically simpler.

3.5 Implementation of the Optimization Algorithms

The pseudocode for the EGFE method is in Algorithm 1. In our experiments, the feature generating vector sets were initialised randomly, these were sampled uniformly from [0,1].

Recall that each of the Problems 1-4 is a maximization problem. They are further constrained (1) for the uniqueness of solution of these problems and (2) for the convergence of the optimization algorithms. The constraints enforce the norm of these feature generating vectors to be one. Let's denote by F the cost function, the Lagrangian function of that for the constrained optimization problems is given by

$$F - \frac{1}{2}\sum_{\ell=1}^{L} \alpha_\ell (||\mathbf{a}_\ell||^2 - 1),$$

where α_ℓ are the Lagrangian multipliers. The conditions for these constrained optimization

problems are

$$\nabla_{\mathbf{a}_\ell} F - \alpha_\ell \mathbf{a}_\ell = 0,$$
$$\mathbf{a}_\ell^T \mathbf{a}_\ell = 1.$$

To compute the optimal $\mathbf{a}_\ell$

$$\mathbf{a}_\ell = -\frac{1}{\alpha_\ell} \nabla_{\mathbf{a}_\ell} F \tag{3.48}$$

Combining these two relations, we obtain

$$\alpha_\ell = \mathbf{a}_\ell^T \nabla_{\mathbf{a}_\ell} F.$$

This relation can be used to obtain the optimal value of Lagrangian multiplier λ, provided in case that an estimate of the optimal $\mathbf{a}_\ell$ is known.

Algorithm 1 Pseudo-code for the EGFE method

INPUT: $S = \{(M_i, c_i) : i = 1, ..., N\}$ training data set where i denotes sample index, M_i the i-th order-L tensor, and $c_i \in \{0, 1\}$ the label of M_i;
D: the number of features to generate from and order-L tensor;
β: learning rate;
ΔF^{thres}: threshold value for stopping the alternating optimization loop.
OUTPUT:
$A = \{\mathbf{a}_{\ell,d} : \ell = 1, ..., L; d = 1, ..., D\}$: feature-generating vector sets (ℓ: mode index; d feature index.
$V = \{V_d : d = 1, ..., D\} = \{v_{i,d} : i = 1, ..., N; d = 1, ..., D\}$ feature set.
Algorithm:

For $d = 1, 2, ..., D$ (greedy feature extraction loop)
  Set counter $iter$ to 1;
  Initialize $A = \{\mathbf{a}_{\ell,d} : \ell = 1, ..., L; d = 1, ..., D\}$
  Compute the objective function F_d^{iter} using Eq.(3.1) for $d = 1$ and Eq.(3.4) for $d > 1$ (which requires the values of $\{V_1, ...V_{d-1}\}$)
  A loop for implementing Alternating Optimization (AO) loop
    For $\ell = 1, 2, ..., L$ (A loop scanning L tensor modes)
      Use Eq.(3.48) to compute the optimal $a_{\ell,d}(a_{\ell,d}^{opt})$ that maximizes F_d^{iter}
      Update $\mathbf{a}_{\ell,d}$ by $\mathbf{a}_{\ell,d}^{iter+1} = (1 - \beta)a_{\ell,d}^{iter} + \beta a_{\ell,d}^{opt}$
        $= \mathbf{a}_{\ell,d}^{iter} + \beta(\mathbf{a}_{\ell,d}^{opt} - \mathbf{a}_{\ell,d}^{iter})$
    End of the scan loop
    Increase $iter$ by 1
    Compute F_d^{iter} using Eq.(3.1) for $d = 1$ and Eq.(3.4) for $d > 1$ (which requires the values of $\{V_1, ...V_{d-1}\}$)
    Compute $\Delta F = F_d^{iter} - F_d^{iter-1}$
    if $\Delta F < \Delta F^{thres}$
      Set $A_d = \{\mathbf{a}_{\ell,d} = \mathbf{a}_{\ell,d}^{iter} : \ell = 1, .., L\}$
      Break the AO loop
    End of if
  End of the AO loop
  Use A_d to get generate $V_d = \{v_{i,d} : i = 1, ..., N\}$ from $S = \{(M_i, c_i) : i = 1, ..., N\}$
End of greedy feature extraction loop

3.6 Conclusion

The Efficient Greedy Feature Extraction (EGFE) method for higher order tensor data was presented in this chapter in detail. As other methods that extend Linear Discriminant Analysis to the case of higher order tensor data, this method is based on the optimization of a Fisher-like criterion. The difference is that the features are extracted from the data sequentially, one by one, which means that the optimization problem to be solved is, in general, of smaller dimensions than required by other methods.

Two forms of the Fisher-like criterion are considered: the multiplicative form and the additive form. The optimization of either of the two criteria realizes essentially the same task. In both cases, we are maximizing the difference between the averages of the two classes, while minimizing the scattering within each class. The additive criterion is dependent on an additional penalty parameter λ that has to be tuned, which adds some complexity to the numerical procedures.

One advantage enjoyed by the multiplicative criterion in the single feature generating set case is that it is invariant to the scaling of the vectors in the set. Indeed, the expression (3.1) does not change if each of the vectors is multiplied by some scalar. Indeed, if the feature generating vectors $\mathbf{a}_\ell$ are changed to $\mathbf{a}'_\ell = c_\ell \mathbf{a}_\ell$ for $\ell = 1, \ldots, L$ where c_ℓ are arbitrary non-zero factors, then the new features become $f^{k,i'} = (\prod_{\ell=1}^{L} c_\ell) f^{k,i}$, and therefore

$$m^{k'} = (\prod_{\ell=1}^{L} c_\ell) m^k, \; S^{k'} = (\prod_{\ell=1}^{L} c_\ell)^2 S^k,$$

but the value of F_m in (3.1) remains unchanged.

This can be used to simplify the optimization algorithm, by using an unconstrained steepest ascent step followed by a scaling to norm one of the new vectors that, as shown before does not change the value of the criterion. Explicitly, the iteration becomes

$$\mathbf{a}_\ell^{(k+1)} = \frac{\mathbf{a}_\ell^{(k)} + p_k \nabla_{\mathbf{a}_\ell} F_m(\mathbf{a}_\ell^{(k)})}{||\mathbf{a}_\ell^{(k)} + p_k \nabla_{\mathbf{a}_\ell} F_m(\mathbf{a}_\ell^{(k)})||}.$$

Of course, the step p_k needs to be chosen such that the norm in the denominator is not zero, but otherwise, after the scaling, the criterion will keep the larger value that was obtained after the unconstrained steepest ascent step. Notice that this simplification cannot be applied to the multiple feature case, and to the additive criterion optimization.

3.7 Chapter Summary

This chapter presents the theoretical basis and the implementation formulas for the EGFE method. This method is based on extracting features from higher-order tensor data in a sequential manner. Formulas for the computation of the gradient of the optimization criterion are derived for extracting the first feature, and for extracting one feature, after a number of features have already been extracted. These formulas are derived for two forms of the optimization criterion: the multiplicative form and the additive form. They can be used in the implementation of a gradient search method in order to solve the optimization problems that need to be solved in order to extract the most relevant features for classification. Some implementation issues, as well as some differences between the two optimization criteria are further discussed in this chapter.

CHAPTER 4

APPLICATION FOR EARLY DEMENTIA DETECTION

4.1 Introduction

Alzheimer's Disease (AD) is the most common neurodegenerative disease in ageing. It is characterised by the progressive impairment of neurons and their connections. Mild Cognitive Impairment (MCI) is the prodromal stage of AD. Thus, accurate diagnosis of MCI (i.e. the early stage of AD) is very important for timely treatment and delay of disease progression. As MCI results in detectable loss of cognitive function, cognitive test scores have been used diagnostically [1]. Further, MCI is known to cause changes in brain activation patterns as well as in brain connectivity. Therefore, fMRI has been increasingly used as a diagnostic tool of MCI patients [14, 16]. In machine learning terms, diagnosis of MCI patients can be formulated as a classification task to discriminate MCI patients from healthy controls. In the last decade , fMRI data has been used for studying brain connectivity. In particular, various statistical connectivity models have been developed to infer complex structure in fMRI brain connectivity. In cases where fMRI data are collected from patients with Mild Cognitive Impairment, such connectivity structure could be utilized to recognize typical behavior of different MCI types (when compared to healthy controls) [32]. For example, the functional brain connectivity of MCI patients is compared to that of Alzheimer's patients or that of adults with no cognitive deterioration. For a

comprehensive review, we refer to [32]. Among 79 studies included in this review article, clinical implications of brain connectivity estimation was evident in most cases . As an example, it is reported that increased activation in the hippocampus to solve memory tasks seems to predict early detection of Alzheimer's Disease (AD) [25, 72]. In this chapter, we present a novel classifier using cognitive test scores as inputs to the classifier and using fMRI data as privileged information.

In the recent literature on the classification tasks related to AD, we observe a clear trend: state-of-the-art machine learning techniques have been increasingly employed to take on new tasks. For example, a classification task should also provide insights into the relevance of the input features used for the task. In [14], Gaussian process classifiers have been employed for the discrimination between healthy controls and MCI patients as well as the the discrimination between MCI and AD patients. More importantly, Gaussian process classifiers have been used to automatically determine the relevant input features when training the classifier. In [16], a challenging classification task was tested, that is, discrimination of two subgroups of MCI patients. Patients in one subgroup will likely progress to AD but those in another group will not convert to AD. In the literature, this classification task is referred to as MCI-AD conversion prediction. This work incorporates data from both healthy subjects and AD patients for classification of MCI patients using the transfer learning framework. Transfer learning is a (relatively) new development in machine learning that aims to boost the performance of a classifier operating in one domain (e.g. MCI patients) by incorporating data from other domains (e.g. healthy subjects and AD patients).

Here we ask whether MCI patients differ in their cognitive skills from controls. Our task is to classify cognitive profiles in patients vs. controls based on cognitive scores and fMRI data. Our EGFE method (multiplicative) is utilised to extract first feature. Furthermore, we address the case when fMRI data are not available for classifying a new subject. To utilise the fMRI data for the task, we train our classifier on participants for whom both cognitive and fMRI data are available. After that, the trained classifier will

classify a new subject solely based on his/er cognitive test scores. This case is of relevance in practice because (1) When compared to cognitive data, the collection of neuroimaging data is much more time-consuming and expensive; (2) Many older individuals (e.g. those with a cardiac pacemaker) may not be safe for imaging such as fMRI scanning. On the other hand, neuroimaging data have more diagnostic power than cognitive data and thus should be used when available. In our work, the classifier is trained by adopting a "metric learning" based approach to *Learning with Privileged Information* (LPI) [37]. As transfer learning, LPI is also a new development in machine learning. In our context, cognitive data are the inputs to the classifier. In contrast, fMRI data act as privileged information that is used only for training the classifier (along with the cognitive data). As most classifiers operate based on a distance/similarity measure between pairs of input vectors, the metric tensor used to compute such distance is therefore crucial for the classification task. In the model of [37], the privileged information (in our case fMRI data) is used to modify the metric tensor (and hence the metric) in the original space (in our case cognitive test scores) to improve the classification accuracy in the original space. Intuitively, if cognitive test scores of two participants appear "similar", but their fMRI data shows different characteristics, the distance between the two cognitive test score vectors should be increased (and vice-verse). As the scale parameter in [14], the diagonal elements of the discriminative metric tensor can be used to automatically determine the relevant cognitive features. Furthermore, all the experiments that show good results will be compared with SVM and SVM+ classifiers to show the benefit of PI.

Additionally, the value of the additional feature is examined; we compared our EGFE methods (multiplicative and additive) with the 2D-LDA method. Our motivation is to extend the first extracted feature to extract multiple features (specifically second features in our case here), based on a given first feature, by using a greedy approach because of the small dataset. Our optimal solution of the objective function would be for both features (first and second features), although the first one is given. The way of extracting the features aims to squeeze data points not in one dimension but in every dimension

within the classes (patients and non-patients); this is the reason for assuming spherical Gaussian distribution. Our results showed that the multiplicative approach outperforms both the additive and 2D-LDA methods and has fewer miss-classification errors than the first feature; this proves that the second features are needed.

4.2 Materials

The cognitive and fMRI data used in this study were collected in the context of two behavioral & fMRI studies [4, 69, 68] in which the participants were asked to predict the orientation of a test stimulus following exposure to structured sequence of leftwards and rightwards oriented gratings, and no feedback were given. Both studies aimed to (1) test whether training on structured temporal sequences improves the ability to predict upcoming sensory events and (2) identify brain regions that support the ability of using implicit knowledge about the past for predicting future. In particular,[4] and [69] investigated how MCI patients differ from healthy controls in terms of (1) their ability to learn predictive structures as well as (2) their learning-dependent brain activation patterns. The diagnosis of MCI patients was made by an experienced consultant psychiatrist (PB) using the National Institute of Ageing and Alzheimer's association working group criteria [1].

In both studies, participants took part in two fMRI scans before and after behavioural training (i.e. pre- and post-training session) during which they completed 5–8 independent runs of the prediction task in each scanning session. Each run comprised 5 blocks of structured and 5 blocks of random sequences (3 trials per block) presented in a random counterbalanced order. In each trial, the participant was presented with a sequence of eight left and rightward oriented gratings (in rapid succession, 250ms + fixation 200ms) followed by a repeat of the same sequence. The participant was instructed to pay attention to the sequence and respond whether the test grating (randomly chosen grating during the second repeat) was correct or incorrect given that presented sequence. Even though the participants could not tell what exactly was the sequence structure, they learn how

to correctly predict whether the grating has the correct orientation given the presented sequence. In random sequence trials, the grating's orientations were randomly generated so the participant could not correctly predict them.

The fMRI data used in this study were acquired in a 3T Achieva Philips scanner at the Birmingham University Imaging Centre using a thirty two-channel head coil. Anatomical images were obtained using a sagittal three dimensional T1-weighted sequence with 175 slices (voxel size = $1 \times 1 \times 1$ mm^3) for localisation and visualisation of functional data. Functional data were acquired using a T2-weighted EPI sequence with 32 slices (whole-brain coverage; TR = 2 s; TE = 35 ms; flip angle = 73; voxel size = $2.5 \times 2.5 \times 4$ mm^3). All the data collection is from the same project, the same software is used for all subjects by applying same measurements devices under same conditions.

In [68], regions-of-interest (ROI) were identified by applying whole-brain general linear model analysis with a voxel-wise mixed-design three-way (ANOVA), that is,

session (pre- vs. post-training)×sequence (structured vs. random)×group (MCI vs. controls).

Statistical maps were cluster threshold corrected ($p < 0.05$). Table 1 in [68] listed all brain regions showing significant interaction between session, sequence, and group. For the study presented in this chapter, we combined two ROIs in the frontal region (Superior Frontal Gyrus, SFG, on the right hemisphere and Medial Frontal Gyrus, MFG, on the left hemisphere) and two ROIs in the cerebellar region (Cerebellar Lingual and Cullmen ROIs in both hemispheres). This resulted in a frontal ROI of size 126 and a cerebellar ROI of size 82. Also, a subcortical ROI (that is, the parahippocampal gyrus ROI of size 32) was selected for the study.

All 60 participants involved in this study had undergone cognitive skill tests (including working memory, cognitive inhibition and attentional skills). These tests provide four quantitative measures of different cognitive skills for each participant:

1. In the working memory task, a number of coloured dots are on display for half second.

Then, they disappear for 1 second and reappear with some dots having changed their colour. A participant is asked to judge whether a given dot has changed its colour or not. The participant's working memory skill can be measured by the maximal number of coloured dots on display for achieving a 70.7% test performance (denoted by n_{dots});

2. To quantify a participant's attention skill, the following cognitive task was performed: two objects are on display, one located at the display centre, another located on the periphery of the display. The peripheral object can only take one of eight equally distributed radial directions (with respect to the display center). The central object could be either car or truck silhouette, whereas the peripheral object must always be the truck silhouette. The participant was asked to identify the type of the central object (car vs. truck) and the location of the peripheral stimulus before the display was masked by white visual noise. This skill is measured by the minimal display time required for the participant to achieve 70% task performance. Depending on whether or not there are distractors on the display, the skill of divided or selective attention is measured (denoted by t^d_{disp} and t^s_{disp}, respectively);

3. The skill of inhibition is measured in a stop-signal test. A participant is first cued to perform a motor task. This is followed by a tone with some time delay, which signals task abortion. The quantity measuring the inhibition skill, t_{delay}, is given by the minimum delay time for achieving a 70.7% test performance.

Sixty participants are involved in this study. Thirty-four of them have both cognitive and fMRI data. Among these participants, nine MCI patients and nine healthy controls come from the cohort reported in [69]. The remaining sixteen healthy controls come from the cohort reported in [68]. The size of that cohort is twenty. Four of them are not included in this study because their cognitive data were missing. Note that for these thirty-four subjects having both cognitive and neuroimaging data for training of classifiers, MCI patients and healthy controls were age matched: mean age of MCI patients was 68.9 , and

mean age of controls was 68.3. The remaining twenty-six participants have cognitive data only. Among them, four MCI patients and five healthy controls come from [4] and [69]. The remaining seventeen participants are from unpublished studies but they participated exactly the same experiments as other participants. Note that all neuroimaging data used in this study are reported either in [69] or in [68].

4.3 Methods

4.3.1 Generation of fMRI Features

fMRI Signal Features

For each ROI and each (pre- and post training) session, we calculated percent signal change (PSC) by subtracting fMRI responses to random sequences from fMRI response to structured sequences and dividing by averaged fMRI response to both stimulus sequences. Let n_r and n_s denote the number of volumes scanned during the trials with random and structured sequences, respectively. For a ROI of size S, its PSC value is computed as follows:

$$PSC = \frac{1}{S}\sum_{s=1}^{S}\frac{\frac{1}{n_s}\sum_{i\in I_s} y_{si} - \frac{1}{n_r}\sum_{j\in I_r} y_{sj}}{\frac{1}{n_s}\sum_{i\in I_s} y_{si} + \frac{1}{n_r}\sum_{j\in I_r} y_{sj}} \tag{4.1}$$

where i and j denote volume index, s voxel index, $I_s = \{i_1, ..., i_{n_s}\}$ the collection of "structured" volumes and $I_r = \{j_1, ..., j_{n_s}\}$ the collection of "random" volumes. The above definition implies that PSC measures scaled fMRI-response to temporally structured stimuli and it is an overall measure averaged over both volumes and voxels.

fMRI Graph Features

Graph matrix Graph structure characterises the connectivity between nodes of a graph. In this study, the graph structure of a single ROI is represented by so-called

graph matrix G of size $S \times S$ where S denotes the ROI size. The value of G_{ij} measures the functional connectivity between voxel i and voxel j, and is computed as (linear) cross-correlation between two fMRI time series of length n on the voxel pair (denoted by $\mathbf{y}_i = (y_{i1}, ..., y_{in})^\intercal$ and $\mathbf{y}_j = (y_{j1}, ..., y_{jn})^\intercal$, respectively), that is,

$$G_{ij} = \frac{1}{n} \cdot \frac{\sum_{k=1}^{n} (y_{ik} - \mu_i) \cdot (y_{jk} - \mu_j)}{\sigma_i \cdot \sigma_j} \tag{4.2}$$

where μ and σ stand for the mean and standard deviation of individual fMRI time series. In the case of $i = j$, we obtain $G_{ij} = 1$. Note that G_{ij} is a connectivity measure independent of the activation intensity on each of two voxels.

Discriminative feature extraction Often, a classifier's inputs are not those raw data to be classified but the features extracted from the raw data. This can significantly reduce the input dimension, which tackles both "curse of dimensionality" and the small sample-size problem. Therefore, a good choice of feature vector plays an important role in classification. This is the motivation for extraction of discriminative features. The discriminative features are suitable because they are extracted in a task-driven & supervised manner. Linear Discriminant Analysis (LDA) is a machine learning technique for discriminative feature extraction. The assumption of LDA is that the feature vectors of each class are Gaussian-distributed. In LDA, high-dimensional feature vectors are projected into a lower-dimensional space and the projection matrix is optimized so that the classes are maximally separated in the projection space. To this end, the empirical covariance matrices need to be estimated using the feature vectors from individual classes. If the number of feature vectors is small and their dimension is high, the empirical estimates of covariance matrices are not accurate. Thus, LDA suffers from the same problem as classifiers do. So-called 2D-LDA has been proposed by [84] for the cases where data items are matrices (e.g. graph matrices in this study) and a direct application of standard LDA with vectorised matrices could fail due to the above-mentioned problem. In the following,

we summarise both standard LDA and 2D-LDA with the dimension of the projection space fixed to one.

For standard LDA, assume that we have N d-dimensional feature vectors, $\{\mathbf{x}_n : n = 1, ..., N\}$, for training in which N_1 feature vectors are from *Class 1* and $N_2 = N - N_1$ from *Class 2*. Denote these two subsets by $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively. The mean vectors of *Class 1* and *Class 2* are given by $\mathbf{m}_1 = \frac{1}{N_1} \sum_{\mathbf{x}_n \in \mathscr{C}_1} \mathbf{x}_n$ and $\mathbf{m}_2 = \frac{1}{N_2} \sum_{\mathbf{x}_n \in \mathscr{C}_2} \mathbf{x}_n$, respectively. Define the between-class covariance matrix $\mathbf{S}_B$ and the total within-class covariance matrix $\mathbf{S}_W$ as

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\intercal \tag{4.3}$$

and

$$\mathbf{S}_W = \sum_{\mathbf{x}_n \in \mathscr{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^\intercal + \sum_{\mathbf{x}_n \in \mathscr{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^\intercal. \tag{4.4}$$

The projection matrix $\mathbf{w}$ of size $d \times 1$ is optimized by maximizing the Fisher criterion defined by

$$J(\mathbf{w}) = \frac{\mathbf{w}^\intercal \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\intercal \mathbf{S}_W \mathbf{w}} = \frac{\mathbf{D}_B}{\mathbf{D}_W}. \tag{4.5}$$

$\mathbf{D}_B$ and $\mathbf{D}_W$ are referred to as the between-class distance and the total within-class distance. Denote the optimized $\mathbf{w}$ by $\mathbf{w}_{\text{opt}}$ and the extracted features are given as $\{v_n = \mathbf{w}_{\text{opt}}^\intercal \mathbf{x}_n : n = 1, ..., N\}$.

For 2D-LDA, assume that we have N graph matrices of size $d \times d$, $\{\mathbf{X}_n : n = 1, ..., N\}$, for training in which N_1 feature vectors are from *Class 1* and $N_2 = N - N_1$ from *Class 2*. Denote these two subsets by $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively. For *Class 1* and *Class 2*, their mean matrices are given by $\mathbf{M}_1 = \frac{1}{N_1} \sum_{\mathbf{X}_n \in \mathscr{C}_1} \mathbf{X}_n$ and $\mathbf{M}_2 = \frac{1}{N_2} \sum_{\mathbf{X}_n \in \mathscr{C}_2} \mathbf{X}_n$. In contrast to standard LDA, we need two (left and right) projection matrices (or vectors), denoted by $\mathbf{a}$ and $\mathbf{b}$ of size $d \times 1$ projecting the matrices into real numbers. Similarly, the between-class distance and the total within-class distance are defined as

$$\mathbf{D}_B = \mathbf{a}^\intercal (\mathbf{M}_2 - \mathbf{M}_1) \mathbf{b} \mathbf{b}^\intercal (\mathbf{M}_2 - \mathbf{M}_1) \mathbf{a} \tag{4.6}$$

$$= \mathbf{b}^\intercal (\mathbf{M}_2 - \mathbf{M}_1) \mathbf{a} \mathbf{a}^\intercal (\mathbf{M}_2 - \mathbf{M}_1) \mathbf{b} \tag{4.7}$$

and

$$\mathbf{D}_W = \sum_{\mathbf{X}_n \in \mathscr{C}_1} \mathbf{a}^\intercal(\mathbf{X}_n - \mathbf{M}_1)\mathbf{b}\mathbf{b}^\intercal(\mathbf{X}_n - \mathbf{M}_1)\mathbf{a} + \sum_{\mathbf{X}_n \in \mathscr{C}_2} \mathbf{a}^\intercal(\mathbf{X}_n - \mathbf{M}_2)\mathbf{b}\mathbf{b}^\intercal(\mathbf{X}_n - \mathbf{M}_2)\mathbf{a} \quad (4.8)$$

$$= \sum_{\mathbf{X}_n \in \mathscr{C}_1} \mathbf{b}^\intercal(\mathbf{X}_n - \mathbf{M}_1)\mathbf{a}\mathbf{a}^\intercal(\mathbf{X}_n - \mathbf{M}_1)\mathbf{b} + \sum_{\mathbf{X}_n \in \mathscr{C}_2} \mathbf{b}(\mathbf{X}_n - \mathbf{M}_2)\mathbf{a}\mathbf{a}^\intercal(\mathbf{X}_n - \mathbf{M}_2)\mathbf{b} \quad (4.9)$$

Note that $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{X}_n$, $n = 1, 2, ..., N$, are all symmetric matrix. The projection vectors $\mathbf{a}$ and $\mathbf{b}$ are optimized by maximizing $J(\mathbf{a}, \mathbf{b}) = \mathbf{D}_B/\mathbf{D}_W$ iteratively. At each iteration, we optimize $\mathbf{a}$ or $\mathbf{b}$ while keeping $\mathbf{b}$ or $\mathbf{a}$ fixed. This procedure is repeated until J has converged. Denote the optimized $\mathbf{a}$ and $\mathbf{b}$ by $\mathbf{a}_{\text{opt}}$ and $\mathbf{b}_{\text{opt}}$. The extracted features are given as $\{v_n = \mathbf{a}_{\text{opt}}^\intercal \mathbf{X}_n \mathbf{b}_{\text{opt}} : n = 1, ..., N\}$.

Note that the number of free parameters to be optimised is d^2 for standard LDA operating on vectorised graph matrices and $2d$ for 2D-LDA operating on graph matrices directly.

Small sample-size problem The main idea of this study is using costly but informative fMRI measurements as valuable privileged information in a classification task operating on cognitive features only. To do so the complex spatial-temporal structure in fMRI signals will need to be transformed into a set of indexes (scalars) that best discriminate between the classes.

In our approach we first capture the spatial-temporal structure of fMRI signals within an ROI as a cross-correlation graph. An ROI of S voxels will be represented as a full undirected graph with n nodes (one for each voxel) and the edge between nodes i and j is weighted by the value of the correlation coefficient between fMRI signals in the two voxels. Each such graph will in turn be represented by an $S \times S$ symmetric matrix $\mathbf{X}$ collecting the edge weights.

In this study we have two classes of N subjects - N_p patients and N_c healthy controls (that is $N = N_p + N_c$). The graph matrices of patients and controls are collected in matrix sets $\mathscr{C}_p$ and $\mathscr{C}_c$. Given the two sets of matrices, we propose to extract the discriminating

feature v through a quadratic form applied to graph matrix $\mathbf{X}$: $v = \mathbf{a}^\intercal \mathbf{X}\mathbf{b}$. Both $\mathbf{a}$ and $\mathbf{b}$ are a V-dimensional vectors determined via an optimization problem expressing the need to maximally separate the two classes, while keeping the within-class variability minimal. To find the projection vectors $\mathbf{a}$ and $\mathbf{b}$ we used our EGFE method (multiplicative) to extract first feature, which is working in a similar way as 2D-LDA [111].

For an ROI with S voxels, the discriminative features $\mathbf{a}$ and $\mathbf{b}$ are S-dimensional vectors, meaning that when determining $\mathbf{a}$ and $\mathbf{b}$ we have $2S$ free parameters. As the number of subjects N is smaller than $2S$, in order to avoid overfitting, the size of the graph representing spatial-temporal structure of cortical activations in that ROI needs to be reduced. Note that in our original formulation, each element a_i of $\mathbf{a}$ corresponds to a particular voxel i whose spatial position is $\mathbf{r}_i$. It is natural to expect that spatially close voxels will have similar activation patterns. We therefore introduce a set of K spatially smoothing Gaussian kernels $\mathcal{N}(\mathbf{r}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $k = 1, 2, ..., K$, in the voxel space, positioned at $\boldsymbol{\mu}_k$, shape determined by the covariance matrix $\boldsymbol{\Sigma}_k$. This leads to a decomposition:

$$a_i = \sum_{k=1}^{K} \tilde{a}_k \mathcal{N}(\mathbf{r}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.10}$$

The values of the smoothing kernels k at each voxel i can be collected in the smoothing matrix.

$$\mathbf{P}_{i,k} = \mathcal{N}(\mathbf{r}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.11}$$

The feature vectors $\mathbf{a}$ and $\mathbf{b}$ can then be written as $\mathbf{a} = \mathbf{P}\tilde{\mathbf{a}}$ and $\mathbf{b} = \mathbf{P}\tilde{\mathbf{b}}$, respectively. We have:

$$v = \mathbf{a}^\intercal \mathbf{X}\mathbf{b} = \tilde{\mathbf{a}}^\intercal \mathbf{P}^\intercal \mathbf{X}\mathbf{P}\tilde{\mathbf{b}} \tag{4.12}$$

The $S \times S$ graph matrix $\mathbf{X}$ is thus reduced to the $K \times K$ matrix

$$\tilde{\mathbf{X}} = \mathbf{P}\mathbf{X}\mathbf{P}^\intercal \tag{4.13}$$

and

$$v = \tilde{\mathbf{a}}^{\intercal}\tilde{\mathbf{X}}\tilde{\mathbf{b}} \tag{4.14}$$

For a given number K of Gaussian kernels, their position is determined by k-means clustering in the voxel space and the covariance matrices of each cluster were estimated from the voxel positions within the corresponding clusters.

The number of smoothing kernels K in the three ROIs with 32, 82 and 126 voxels was set to 3, 4 and 8, respectively. The largest ROI is contained in both hemispheres. Hence, the sub-ROIs within each hemisphere were clustered independently into 4 clusters. Spatial smoothing with Gaussian kernels described above expresses the assumption that nearby voxels should have similar functionality. We refer to this approach as Spatial Grouping (SG) and to the resulting feature as (SGF). An alternative approach would be to identify groups of voxels that are not only spatially close but also exhibit similarity in the activation time series (as quantified through cross-correlation) [13]. We thus obtain N functional clusterings of the voxel space, one for each subject. These groupings at the subject level are then merged into a single population based functional clustering of voxels through Consensus Clustering [13]. Given the resulting K voxel clusters, we calculated their means $\boldsymbol{\mu}_k$ and covariance matrices $\boldsymbol{\Sigma}_k$, thus obtaining a set of K "functionally informed" smoothing Gaussian kernels $\mathcal{N}(\mathbf{r}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. The reduced graph matrix $\tilde{\mathbf{X}}$ is then calculated as in eqs: (4.11) and (4.13). We refer to such functional voxel clustering as Functional grouping (FG)and to the resulting feature as (FGF).

Feature Generation Pipeline

Figure 4.1 illustrates the flow of fMRI feature generation. We obtain three fMRI features (PSC, FGF, SGF) independently from fMRI data $\mathbf{Y} \in \mathbb{R}^{S \times T}$. Recall that S is number of voxels and T is the number of volumes. Feature PSC is computed directly from $\mathbf{Y}$. To compute other two features, we first transform $\mathbf{Y}$ to a graph matrix $\mathbf{X}$ of size $S \times S$ and reduce $\mathbf{X}$ to $\tilde{\mathbf{X}}$ of size $K \times K$ with $(K < S)$ either through spatial projection or through

functional clustering. Finally, we extract SGF from $\tilde{\mathbf{X}}$ obtained by spatial projection and FGP from $\tilde{\mathbf{X}}$ obtained by functional clustering. Our EGFE method (multiplicative) is used, 2D-DLDA method has a similar way in extracting the first feature, as they are both optimising Fisher criteria [111].

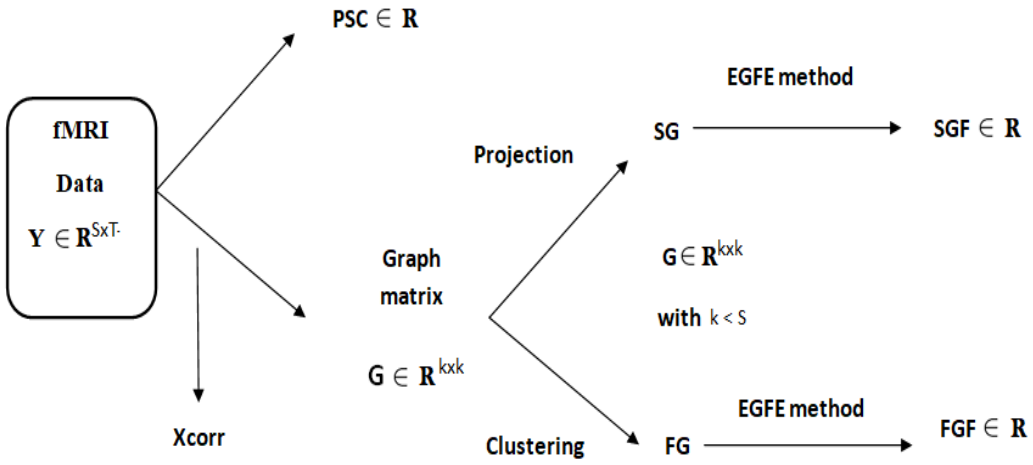

Figure 4.1: Illustration of fMRI feature generation pipeline: from BOLD signal data $\mathbf{Y}$ to three fMRI features (PSC, FGF, and SGF). $F\mathbf{G}$ and $S\mathbf{G}$ are the reduced version of graph matrix $\mathbf{G}$ via functional grouping and spatial grouping (respectively). Note that FGF and SGF are both discriminative features extracted from $F\mathbf{G}$ and $S\mathbf{G}$ in a supervised manner using our EGFE method (multiplicative).

4.3.2 Classification Tools

Generalized Matrix Learning Vector Quantization (GMLVQ)

The classification algorithms of Learning Vector Quantization (LVQ) [2] are supervised learning paradigms which work iteratively to modify the quantization prototypes to find the boundaries of the class. LVQ classifiers are represented by a set of vectors, so-called prototypes, embodying classes in the input space, and a distance metric on the input data. During training, prototypes are adapted in an iterative manner to define class borders. For each training point, the algorithm determines two closest prototypes, one with the

same class as the training point, and another with a different class. The position of the two closet prototypes are then updated, where same class prototype is moved closer to the data point, while different class prototype is pushed away from the data point. During testing, an unknown point is assigned to the class represented by the closest prototype with respect to the given distance.

The LVQ scheme, which is originally introduced by [87], applies Hebbian online learning in order to adapt prototype with training data. Subsequent, researchers proposed a number of modifications to the basic learning scheme. Such variations utilize an explicit cost functionality, whereas others allow for incorporating adaptive distance measures [85, 86].

Given training data $(\mathbf{x}_i, y_i) \in \mathbb{R}^m \times \{1, \cdots, K\}, i = 1, 2, \cdots, n$, where m denotes the dimensionality of data and K signifies the number of different classes. Typically, a LVQ network will include L prototypes $\mathbf{w}_q \in \mathbb{R}^m, q = 1, 2, 3, ..., L$, which is characterized according to their location available in the input space and their class $c(\mathbf{w}_q) \in \{1, ..., K\}$. At least one prototype in each class needs to be present. The overall number of prototypes is a model hyper-parameter that is to be optimized. The (squared) Euclidean distance $d(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^\intercal(\mathbf{x} - \mathbf{w})$ within $\mathbb{R}^m$ quantifies the distance between the input vectors and prototypes. The classification performed using the winner-takes all scheme: the data point $\mathbf{x}_i \in \mathbb{R}^m$ belongs to the label $c(\mathbf{w}_j)$ of the prototype $\mathbf{w}_j$ if and only if with $d(\mathbf{x}, \mathbf{w}_j) < d(\mathbf{x}, \mathbf{w}_q), \forall j \neq q$. For every prototype $\mathbf{w}_j$ with class $c(\mathbf{w}_j)$ a receptive field is defined within the input space. According to the LVQ model, points located in the respective field [1] will be assigned to the class $c(\mathbf{w}_j)$.

The aim of learning is to adapt prototypes automatically in such a way that the gap between data points of class $c \in \{1, ..., K\}$ and the corresponding prototypes with label c (the one that the data are belonging to) will be reduced to a minimum distance. During the stage of training for each data point $\mathbf{x}_i$ with class label $c(\mathbf{x}_i)$, the most proximal prototype with the same label is rewarded by pushing closer towards the training input; the most closest prototype with a different label will be disallowed by moving pattern $\mathbf{x}_i$

[1] The set of points in the input space is defined by the receptive field of prototype $\mathbf{w}$, where this prototype is picked as their winner.

away.

The Generalized Matrix LVQ (GMLVQ) is a recent extension of the LVQ that employs a full matrix tensor for a better measure of distance between two feature vectors. The new distance measure not only is capable of scaling individual features but also accounts for pairwise correlations between the features. Assuming $\Lambda \in \mathbb{R}^{m \times m}$ is a positive definite matrix, $\Lambda \succ 0$, the generalized form of the squared Euclidean distance is defined as

$$d_\Lambda(\mathbf{x}_i, w) = (\mathbf{x}_i - \mathbf{w})^\intercal \Lambda (\mathbf{x}_i - \mathbf{w}) \tag{4.15}$$

The positive definiteness of Λ is guaranteed by imposing $\Lambda = \Omega^\intercal \Omega$, where $\Omega \in \mathbb{R}^{m \times m}$ is a full-rank matrix. Furthermore, to prevent the degeneration of the algorithm, Λ is trace normalized after each learning step (i.e. $\sum_i \Lambda_{ii} = 1$) so that the summation of eigenvalues is kept fixed in the learning process. The model is trained in an online-learning fashion and the steepest descent method is employed to minimize the cost function given as:

$$f_{GMLVQ} = \sum_{i=1}^{n} \phi(\mu_\Lambda(\mathbf{x}_i)) \tag{4.16}$$

with

$$\mu_\Lambda(\mathbf{x}_i) = \frac{d_\Lambda(\mathbf{x}_i, \mathbf{w}^+) - d_\Lambda(\mathbf{x}_i, \mathbf{w}^-)}{d_\Lambda(\mathbf{x}_i, \mathbf{w}^+) + d_\Lambda(\mathbf{x}_i, \mathbf{w}^-)}, \tag{4.17}$$

where ϕ is a monotonic function (the identity function $\phi(l) = l$ is a common choice). The main advantage of the GMLVQ framework is that (unlike LVQ [85, 86]), it allows us to naturally incorporate privileged information through metric learning.

Privileged information (PI) guided GMLVQ

This chapter employs the Information Theoretic Metric Learning (ITML) approach [21] in order to incorporate privileged information into the learning phase of the GMLVQ.

Given a training dataset, we have one space where the original training data live and another space where the privileged training data live. They are denoted by $\mathscr{X}$ and $\mathscr{X}^*$,

respectively, and their corresponding global metric tensors are denoted by $\mathbf{\Lambda}$ and $\mathbf{\Lambda}^*$. The distances between the privileged training points in $\mathscr{X}^*$ are first computed using $\mathbf{\Lambda}^*$ and then are sorted in ascending order. Based on the closeness information in $\mathscr{X}^*$, the original training points are tagged in a categorical manner (similar and dis-similar). After that, the ITML approach is adopted to impose similarity constraints in the original space. The main goal is to learn a new metric in the original space (denoted by $\mathbf{\Lambda}_{\text{new}}$) so that under the new metric, the distance between two original training points is small if their counterparts in the privileged space are similar (close), and vice versa. Implementation of the above concept is described in the following.

The training dataset is given as $\{(\mathbf{x}_i, \mathbf{x}_i^*, y_i) : \mathbf{x}_i \in \mathscr{X}, \mathbf{x}_i^* \in \mathscr{X}^*, i = 1, 2, ..., N\}$. Recall that y represents class label. For each pair of two training examples, $1 \leq i < j \leq N$, we compute three different squared Mahalanobis distances as follows

$$d_{\mathbf{\Lambda}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\intercal \mathbf{\Lambda}(\mathbf{x}_i - \mathbf{x}_j), \mathbf{x}_i, \mathbf{x}_j \in \mathscr{X} \tag{4.18}$$

$$d_{\mathbf{\Lambda}^*}(\mathbf{x}_i^*, \mathbf{x}_j^*) = (\mathbf{x}_i^* - \mathbf{x}_j^*)^\intercal \mathbf{\Lambda}^*(\mathbf{x}_i^* - \mathbf{x}_j^*), \mathbf{x}_i^*, \mathbf{x}_j^* \in \mathscr{X}^* \tag{4.19}$$

$$d_{\mathbf{\Lambda}_{\text{new}}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\intercal \mathbf{\Lambda}_{\text{new}}(\mathbf{x}_i - \mathbf{x}_j), \mathbf{x}_i, \mathbf{x}_j \in \mathscr{X} \tag{4.20}$$

Note that $\mathbf{\Lambda}$ and $\mathbf{\Lambda}^*$ are both given whereas $\mathbf{\Lambda}_{\text{new}}$ needs to be learned. The metric tensor $\mathbf{\Lambda}_{\text{new}}$ should be optimized in a supervised manner so that $d_{\mathbf{\Lambda}_{\text{new}}}(\mathbf{x}_i, \mathbf{x}_j)$ will be shrunk if $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ are similar. Otherwise, $d_{\mathbf{\Lambda}_{\text{new}}}(\mathbf{x}_i, \mathbf{x}_j)$ will be enlarged. To this end, we form two sets of pairs of the training data points in the original space $\mathscr{X}$: S_+ is a set of similar pairs and S_- a set of dissimilar pairs. These two sets are formed using the proximity information in the privileged space $\mathscr{X}^*$ as follows:

1. If $d_{\mathbf{\Lambda}^*}(\mathbf{x}_i^*, \mathbf{x}_j^*) \leq l^*$and $y_i = y_j$(same class label), then $(\mathbf{x}_i, \mathbf{x}_j) \in S_+$;

2. If $d_{\mathbf{\Lambda}^*}(\mathbf{x}_i^*, \mathbf{x}_j^*) \geq u^*$and $y_i \neq y_j$(different class label), then $(\mathbf{x}_i, \mathbf{x}_j) \in S_-$.

Here, l^* and u^* represent the upper and lower bound for the distances of similar and dissimilar pairs, respectively, in the privileged space. The value of l^* is chosen as the upper

bound for the $< a^*$ percentile of all $d_{\mathbf{\Lambda}^*}(\mathbf{x}_i^*, \mathbf{x}_j^*)$ values, $1 \leq i < j \leq N$. Similarly, the value of u^* is chosen as the lower bound for the $> 1 - b^*$ percentile of all $d_{\mathbf{\Lambda}^*}(\mathbf{x}_i^*, \mathbf{x}_j^*)$ values, $1 \leq i < j \leq N$. At the same time, the choice of l^* and u^* is subject to the constraint $u^* > l^*$. Also, a^* and b^* are pre-determined with $0 < a^* < b^* < 1$.

In the GMLVQ framework, the privileged information is incorporated by fusing the metric $\mathbf{\Lambda}^*$ in the privileged space $\mathscr{X}^*$ with the metric $\mathbf{\Lambda}$ in the original space $\mathscr{X}$ (for more details, see [38]).

Support Vector Machine

The idea of kernel mapping is combined with statistical learning and optimisation techniques in supervised learning algorithms called SVMs. The simplest version of an SVM [19] is learning a separating hyper-plane (decision surface) between two classes labelled as $Y = \{1, -1\}$ and maximising the margin. Each class contains the closest training data points to it that solve an optimisation problem. If the principles of statistical properties of the maximal margin solution are followed, it can be a good valid generalisation. What is important is that working with higher dimensional spaces is possible because performance is not affected with the change in dimensionality.

The SVMs, when used for classification, are performed in a feature space of high dimensions, and a maximal margin separation is found using a linear classifier that defines a separating hyperplane. When the kernels are defined in the feature space, this hyperplane can be related to a non-linear decision boundary.

In case that the training set is not linearly separable, the standard SVM model allows the decision margin to make a few "mistakes" which are represented by slack variables (ξ_i).

Considering S is our training set, which contains labelled input vectors $(x_i, y_i), i = 1....m$, where $x_i \in R^n$and $y_i \in Y = \{\pm 1\}$. A linear classification rule f is a function

defined on R^n with values in Y specified via a pair (w, b), where $w \in R^n$ and $b \in R$, as

$$f(z_i) = \langle w, z_i \rangle + b \tag{4.21}$$

Here $\langle ., . \rangle$ represents the dot product and w, b are obtained as solutions of the optimization problem:

$$\min_{w,b,\xi_i} \frac{1}{2}||w||_2^2 + B\sum_{i=1}^{n} \xi_i \tag{4.22}$$

under the constraints,

$$y_i(\langle w, z_i \rangle + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ 1 \leq i \leq n$$

where $B \geq 0$ is a hyper-parameter that balances the goal between classification accuracy (i.e. keeping the slack variables as small as possible) and the smoothness of the decision boundary in the original space. The parameter B is obtained via tuning.

Privileged information (PI) guided SVM (SVM+)

Learning using privileged information with the SVM methodology was proposed [100] and is known as SVM+ .

Privileged information is additional information $x_i^* \in X^*$ available about a training example $x_i \in X$. This means it is only available during the training phase, but not in the testing phase. In the SVM+ model, a set of training triplets is given, $(x_1, x_1^*, y_n), \cdots, (x_n, x_n^*, y_n)$, $x_i \in X$ and $x_i^* \in X^*, y_i \in \{-1, 1\}, i = 1, \cdots, n$, generated by a fixed (unknown) probability measure $P(x, x^*, y)$. The classification rule is the same as in the previous case, i.e. it is defined by the function (4.21). However, in this case, next to the parameters w and b that define the classification rule, two additional parameters w^* and $b^*$are determined by solving the optimization problem

$$\min_{w,b,w^*,b^*} \frac{1}{2}||w||_2^2 + \frac{p}{2}||w^*||_2^2 + B\sum_{i=1}^{n}(\langle w^*, z_i^* \rangle + b^*) \tag{4.23}$$

under the constraints

$$y_i(\langle w, z_i\rangle + b) \geq 1 - (\langle w^*, z_i^*\rangle + b^*), \tag{4.24}$$

$$\langle w^*, z_i^*\rangle + b^* \geq 0 \tag{4.25}$$

A possible interpretation of this problem is that the function $\langle w^*, z_i^*\rangle + b^*$ is an estimator for the slack variables ξ_i in the previous case. There are two hyper-parameters in the objective function of the SVM+ model, $B, p > 0$ that have to be determined by tuning. The parameter p is a non-negative parameter that control the smoothness of the classification function. Again, the model can be non-linearized using the kernel trick: the triplets of training data $(x_1, x_1^*, y_1), \cdots, (x_n, x_n^*, y_n)$ are changed into $(z_1, z_1^*, y_1), \cdots, (z_n, z_n^*, y_n)$ with the help of mapping of vectors $x \in X$ into $z \in Z$ and $x^* \in X^*$ into $z^* \in Z^*$, where Z and Z^* represent the feature spaces related to the inner products $\langle z_i, z_j\rangle = k(x_i, x_j), \langle z_i^*, z_j^*\rangle = k^*(x_i^*, x_j^*)$ defined by kernels k and k^*.

Imbalanced class problem

Class imbalance occurs when there is a mismatch between sample sizes representing different classes. Class imbalance is one of the most common issues in classification. Unless explicitly treated, the classifier can be biased towards the majority class. In general, model fitting algorithms of various forms of classifiers assume balanced class distribution. A variety of methods have been proposed to tackle the class imbalance problem [e.g. [39]]. For example, the imbalance problem can be addressed by either upsampling the minority class(es) [78], or downsampling the majority class(es) [29], so that the training set becomes balanced.

Since the data sets available for our study are relatively small, instead of upsampling small minority class, we decided to downsample the majority class, and repeat the downsampling $N_d = 100$ times. Training portion of the minority class remains fixed and

each time the majority class is downsampled we construct a classifier based on balanced classes. We thus obtain a collection of N_d classifiers trained on different versions of downsampled majority class. These classifiers are then combined in an ensemble to form a single classifier using majority voting over the ensemble members.

Employing Different Types of PI

We have two different kinds of features extracted from fMRI signals and used as privileged information, namely percent change (PSC) in overall ROI activation and graph based features described above.

The PSC feature quantifies the relative activation difference in the whole ROI when subjects were shown structured vs. random stimuli. This is calculated both from both pre- and post-training fMRI data. We consider 3 ROIs, hence there are 6 PSC privileged information features. Analogously, for the graph-based spatial-temporal features, there is a single feature for each ROI, measured both pre- and post-training, yielding a totality of 6 graph-based privileged information features.

An obvious combination of PSC and graph-based features would be to concatenate them into 12-dimensional vector. However, given the small sample size of participants, such an approach might lead to overfitting. Therefore we constructed an alternative way of combining privileged information features, as outlined below.

We independently construct two classifiers operating in the original space, but trained with the two different kinds of privileged information. Given a test input, if both classifiers predict the same class label, that label is used as the model output. If, on the other hand, they disagree, we output the class label that is predicted with "more confidence" - i.e. smaller distance between the test input and the closest class prototype.

However, note that for the classification purposes, the metric tensor in a single classifier can be arbitrarily scaled, since only the relative relations between distances of test point to the class prototypes are relevant. Hence, in order to compare distances of the test point to the closest prototype in the two classifiers, we need to normalize the learnt metrics. We

do this by eigen-decomposing the two metric tensors Λ_1 and Λ_2 and normalizing their eigenvalues to sum to 1. In particular, the eigen-decomposition of Λ_i, $i = 1, 2$, reads $\Lambda_i = \mathbf{U}_i \, \mathbf{diag}(\lambda_1^i, \lambda_2^i, ...\lambda_d^i)\mathbf{U}_i^\intercal$. The normalized metric tensor is obtained as

$$\hat{\Lambda}_i = \mathbf{U}_i \, \mathbf{diag}(\hat{\lambda}_1^i, \hat{\lambda}_2^i, ..., \hat{\lambda}_d^i)\mathbf{U}_i^\intercal, \tag{4.26}$$

where the normalized eigenvalues are

$$\hat{\lambda}_j^i = \frac{\lambda_j^i}{\sum_{k=1}^d \lambda_k^i}. \tag{4.27}$$

Given a test input, when combining two ensemble classifiers C_1 and C_2, if they agree on the predicted label, we output that label as the overall label estimate. If, however, C_1 and C_2 disagree on the label, we prefer the label produced with "more certainty" - in our context - small average distance to the closest prototype. In particular, if C_1 is claiming class +1, we calculate the mean distance of the test input to the closest prototype of class +1 across those ensemble members that output class +1 (e.g. their closest prototype to the test input has label +1). Analogously, for C_2 claiming class -1, we record the mean distance of the test input to the closest prototype of class -1 across ensemble members outputting class -1. The overall class label of the combined classifier for the test input is the label with the minimal average distance to the closest prototype.

4.3.3 Experimental Design

The value of using brain imaging data as privileged information in our setting can be evaluated through two extreme cases:

- No privileged information is available - the models (classifiers) are constructed purely based on the cognitive data. We will refer to this case as M-CD;
- Privileged brain imaging data is always available and is used directly as input data in the classifier construction and testing, without the need to resort to learning with

privileged information. We will refer to this case as M-PD. The classifiers obtained in this regime with the PSC, FGF and SGF representations of brain imaging data are referred to as M-PSC, M-FGF and M-SGF, respectively.

When the classifiers are constructed in the framework of learning with privileged information, with cognitive data serving as classifier inputs and brain imaging data used as privileged information, depending on what representation of brain imaging data is used, we denote the resulting classifiers by M^+-CD-PSC, M^+-CD-FGF and M^+-CD-SGF.

As explained above, PSC representation of spatial-temporal structure of cortical activations within an ROI is the simplest one, integrating out both the spatial and temporal structures. In contrast, a more subtle representation is obtained in the graph based features FGF and SGF, integrating over time, but preserving aspects spatial structure. The PSC and graph based features may contain complementary information for the classification task and hence we further combine the classifiers obtained using brain imaging data into composite ones, in particular M^+-CD-PSC and M^+-CD-FGF are combined into a single classifier M^+-CD-PSC+FGF and the combination of M^+-CD-PSC and M^+-CD-SGF is referred to as M^+-CD-PSC+SGF. Analogously, M-PD-PSC and M-FGF are combined to form M-PSC+FGF and combination of M-PSC with M-SGF results in M-PSC+SGF. The overall model structure setup is illustrated in Figure 4.2.

4.4 Baseline Experiments

This section assesses the classification performance of the proposed methodology that incorporates fMRI as privileged information (PD) in the training phase, against baseline algorithms trained without PD, or trained solely with PD. Since we expect that the brain imaging fMRI data carry lot of information regarding possible MCI, the classifier trained directly on fMRI (M-PD) will provide a lower bound on the classification error that a classifier trained solely on cognitive data (M-CD) (carrying less information on possible MCI) cannot achieve. We expect that the power of learning with privileged information

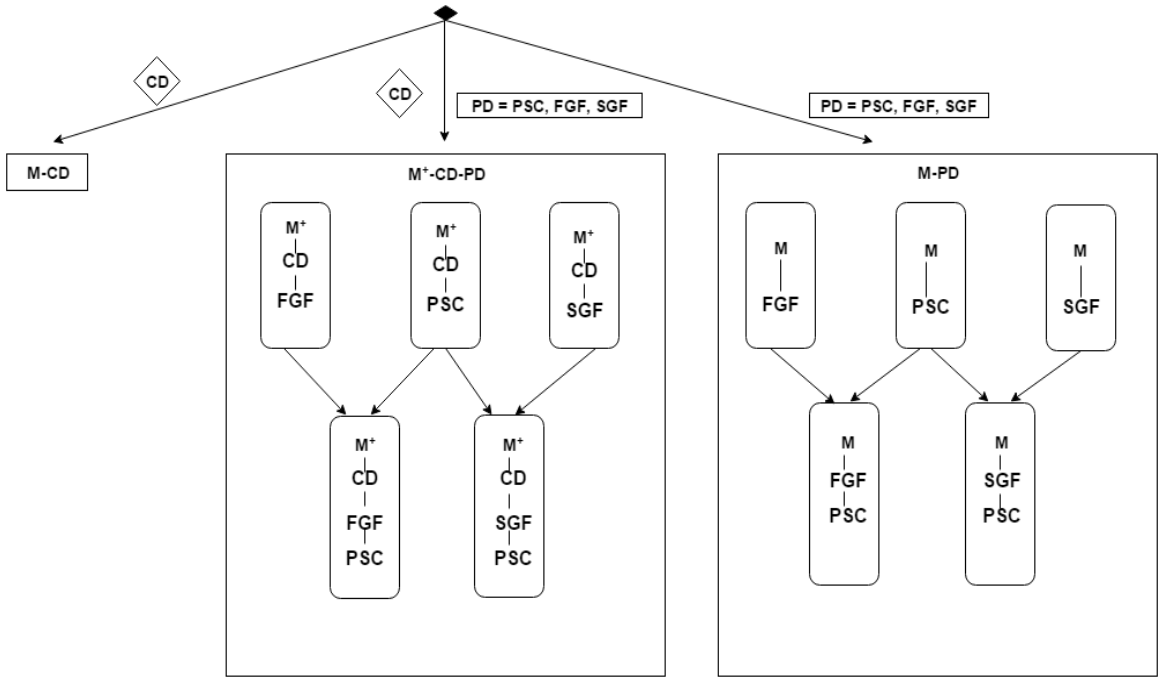

Figure 4.2: Schematic illustration of the experimental design described in Section4.3.3. The items in diamond shape denote data: (CD) for cognitive data, PD for privileged information data, PSC for Percent Signal Change, FGF for functionally grouped graph feature, and SGF for spatially grouped graph feature. M-XXX denotes a GMLVQ classifier that does not use privileged information while XXX denotes the inputs to this classifier. For example, M-PSC means a GMLVQ classifier with PSC features as its inputs. M^+-XXX-YYY denotes a GMLVQ classifier using feature XXX as its inputs and feature YYY as privileged information. For example, M^+-CD-PSC means a GMLVQ classifier using cognitive features as its inputs and PSC features as privileged information. M^+-XXX-YYY-ZZZ denotes a hybrid classifier that combines the classification output of classifier M^+-XXX-YYY and classifier M^+-XXX-ZZZ using a certain rule (e.g. majority voting rule).

will boost the classification performance, so that the classifier trained with CD as inputs, but able to incorporate fMRI indirectly in the training process (M^+-CD-PD), will have classification performance between the two extremes M-PD and M-CD, even though in the test phase, both M-CD and M^+-CD-PD classify solely based on CD. The methodology is formulated in the framework of prototype-based classification (GMLVQ) with metric learning [38, 85, 86]. In this experiment, the original and privileged features correspond to cognitive profiles and brain imaging data, respectively. The overall experimental design is explained in section 4.3.3.

4.4.1 Experimental Setup

In the M-PD case, we have in total a set of 34 subjects having both cognitive and brain imaging data, consisting of 9 patients and 25 controls. We create 50 training-test set splits by randomly sampling 6 and 17 patients and controls, respectively, to form the training set (the rest is in the test set). In the M-CD case we have 60 subjects having cognitive data, consisting 13 patients and 47 controls. Again, we created 50 training-test set splits by randomly sampling 9 and 33 patients and controls, respectively, to form the training set. We made sure that in each resampled training and test set there is an equal balance between subjects with and without PD.

As explained in section Section 4.3.2, to deal with class imbalance in the M-PD case, we construct ensemble classifiers by using the same set of 6 patients and repeatedly sampled 6 controls from the 17 training ones. Analogous setting was used in the M-CD case, this time with 9 patients and 33 controls.

In all experiments, the (hyper-)parameters of the ensemble classifiers were tuned via cross-validation on the training set of the first sub-split only. The found values were then fixed across the remaining 99 classifiers. In the GMLVQ classifier, data classes are represented by one prototype per class. The class prototypes are initialized as means of random subsets of training samples selected from the corresponding class. In the IT metric learning settings given in [38], lower (a, a^*) and upper (b, b^*) percentile bounds for the privileged and original spaces were tuned over the values of $5, 10, 15$ and of $85, 90, 95$, respectively.

Throughout the experiments we had one data set in the original space of CD. However, experiments were repeated for three different fMRI PD: PSC, SGF and FGF. PD of each subject is represented by 6 features, 3 pre-training and 3 post-training, corresponding to 3 ROIs. Due to the imbalanced nature of our classes we utilized the following below evaluation measures:

Confusion Matrix: it is a popular performance indicator for machine learning algorithms. It is organized along the the actual classes (rows) and the predicted ones columns)

[29]. In this study positive and negative examples represent patients and controls, respectively. In the confusion matrix, True Positive (TP) denotes the number of positive examples correctly classified, True Negatives (TN) is the number of negative examples correctly classified , False Positives (FP) is the number of negative examples incorrectly classified, False Negatives (FN) is the number of positive examples incorrectly classified as negative. The true positive rate ($TPR = \frac{TP}{TP+FN}$) measures the percentage of patients who are correctly classified, whereas the true negative rate ($TNR = \frac{TN}{TN+FP}$) measures the proportion of the correctly identified controls. False positive rate ($FPR = \frac{FP}{FP+TN}$), refers to the probability of falsely classifying the patients, whereas the false negative rate ($FNR = \frac{FN}{FN+TP}$) refers to the probability of falsely classifying the controls. Macroaveraged Mean Absolute Error is used for the classification errors across classes, $MMAE = \frac{1}{2}\left(\frac{\sum_{y_i=1}|y_i-\hat{y}_i|}{N_1} + \frac{\sum_{y_i=2}|y_i-\hat{y}_i|}{N_2}\right)$.

4.4.2 Classification Results

Statistical Test: All the experiment results are evaluated through a paired Wilcoxon signed-rank test. It is a non-parametric test that has no assumption about the distribution. The test is utilized with paired groups to measure the statistical significance of the difference between two classifiers' performances. The test is done for the case of the privileged brain imaging data, when CD is operating in the original data space and trained with PI (e.g M^+-CD-PSC). The null hypothesis states that the group means for the classifiers trained with and without privileged information are two samples from the same population.

We are primarily interested in classification performance of M^+-CD-PD classifiers, that is, classifiers using cognitive data as their inputs and incorporating brain imaging data as privileged information. this classification performance will be put in the context of performances when no brain imaging information is available (M-CD) and when the full brain imaging is available as input (M-PD). This will allow us to quantitatively investigate how much performance improvement over M-CD could be obtained by incorporating privileged information through metric learning. Following our experimental setup, we

obtained 50 MMAE estimates for each classifier summarised by the mean, standard deviation, median and the (25%, 75%) percentiles. The results are summarised in Tables 4.1 and 4.2.

Table 4.1 shows that for all five types of PD, M-PD outperforms M-CD. Recall that we have extracted three different features from the brain imaging data, namely PSC, SGF, and FGF, and all of them can be used as PD. For PSC, which is related to brain activation level, the corresponding median MMAE is reduced by relatively 41% when compared to that of M-CD. The other two types of PD, SGF and FGF, are related to brain connectivity pattern. When compared to the baseline classifier, the relative reduction of their median MMAE is about 26% and 41%, respectively. The above results indicate that PSC is useful as the graph feature (FGF), or even more useful (SGF). In principle, the activation level and connectivity pattern are two independent fMRI features. Therefore, PSC could be used as PD along with SGF or FGF. Row 6–7 in Table 4.1 show that the resulting classifier can either attain the classification performance of M-PSC in the case of SGF, or improve on it in the case of FGF. In summary, brain imaging data can contain more information that are relevant to the task than cognitive data.

Models	Mean	Std-Dev	Median	(25%, 75%) Percentile
M-CD	0.39	0.09	0.39	(0.31, 0.44)
M-PSC	0.23	0.16	0.23	(0.14, 0.33)
M-SGF	0.27	0.08	0.29	(0.21, 0.32)
M-FGF	0.25	0.11	0.23	(0.21, 0.30)
M-PSC+SGF	0.25	0.11	0.25	(0.21, 0.33)
M-PSC+FGF	0.24	0.12	0.23	(0.16, 0.30)

Table 4.1: Classification performance measured by Macroaveraged Mean Absolute Error (MMAE) for the baseline classifier, M-CD, and five different M-PD classifiers (see Column 1). For each classifier, we report both mean MMAE, its standard deviation, median MMAE and its (25%, 75%) percentile in Column 2 – 5, respectively. They were computed using the MMAE estimates obtained from 50 randomly created training-test splits.

Table 4.2 shows that for all five types of PD, M^+-CD-PD outperforms M-CD. In particular, PSC and SGF are the best two among the five PD types that are used as the privileged information along with CD as GMLVQ's inputs. Compared to M-CD, both

M^+-CD-PSC + M^+-CD-SGF show a reduction of their median MMAE by relatively 21%. This relative improvement is shrunk to 15%, 13%, and 8% for M^+-PSC+FGF, for M^+-PSC+SGF, and for M^+-FGF (respectively). We can note that the good choice of PI can help to improve over M-CD. In order to check whether there is a statistically significant improvement when integrating PI along CD in the training stage, we used a one-sided sign-rank test. It looks like there are some improvements in the case of M^+-CD-PSC and M^+-CD-SGF; while the less promising combination is in the case of M^+-CD-FGF. If there was a larger sample size, perhaps we could expect larger improvements.

Models	Mean	Std-Dev	Median	(25%, 75%) Percentile
M^+-CD-PSC	0.34	0.09	0.31	(0.27, 0.40)
M^+-CD-SGF	0.33	0.08	0.31	(0.26, 0.40)
M^+-CD-FGF	0.37	0.11	0.36	(0.30, 0.40)
M^+-CD-PSC+SGF	0.33	0.09	0.34	(0.24, 0.40)
M^+-CD-PSC+FGF	0.35	0.12	0.33	(0.24, 0.42)

Table 4.2: The same as in Table 4.1 but for evaluation of the classification performance of five different M^+-CD-PD classifiers, that is, the classifiers using CD as their inputs and PD as privileged information.

Table 4.3 presents the results of the average TPR and TNR of the models. The best two TPR results (0.69 and 0.72) were achieved by M^+-CD-PSC and M-FGF respectively; whereas the best two TNR results (0.88 and 0.89) were attained by M-PSC and M-PSC+FGF respectively.

4.4.3 Further Analysis

GMLVQ is a fully adaptive algorithm to learn global metric tensor which accounts for different importance weighting of individual features and pairwise interplay between the features, with respect to the given classification task. Hence, it allows us to study the task-dependent relevance of the input features by using the diagonal elements of the GMLVQ metric tensor matrix. Moreover, the global metric can be further optimized adaptively by incorporating privileged information into the GMLVQ model via the distance

Model	TPR	TNR
M-CD	0.60	0.60
M-PSC	0.64	0.88
M^+-CD-PSC	0.69	0.63
M-SGF	0.66	0.71
M^+-CD-SGF	0.72	0.60
M-PSC+SGF	0.64	0.87
M^+-CD-PSC+SGF	0.68	0.67
M-FGF	0.74	0.60
M^+-CD-FGF	0.56	0.69
M-PSC+FGF	0.38	0.89
M^+-CD-PSC+FGF	0.58	0.70

Table 4.3: Overall true positive rates (TPR) and true negative rates (TNR) on hold-out sets

relations revealed in the privileged space [37]. In the following we analyse the learned classification models in terms of the learned metric tensor and discuss possible implications regarding the cognitive and brain imaging fMRI features used in this study.

Cognitive features only

We first present a procedure to study the relevance of four cognitive features (working memory, cognitive inhibition, divided attention, and selective attention) using the GMLVQ metric (tensor) matrices obtained from the experiments whose classification results are discussed in Section 4.4.2. Each of these experiments resulted in 50 $\times$ 100 GMLVQ classifiers with the associated metric (tensor) matrices Λ obtained by training GMLVQ classifiers on 50 $\times$ 100 (small) data sets independently. Recall that these data sets were generated by first randomly splitting the whole training set into 50 smaller sets of equal size and then randomly downsampling the majority class to the size of the minority class in each split 100 times. In this way, we ensure that the training and test subsets have the same distribution, and we tried to do many resampling (100 times over 50 experiments of majority class) in training set without replacements of test set. However, many of the 50 $\times$ 100 classifiers performed poorly and they should not be included in the analysis of the relevant cognitive features. We therefore discard the data split producing the

ensemble classfier whose N_b-th best ensemble member (classifier) produced error larger than a threshold value denoted by E_{max}, and pool all ensemble members from each of the remaining splits for further analysis. This procedure is applied to three experiments as follows: M-CD, M^+-CD-PSC and M^+-CD-FGF. We found out that $N_b = 15$ and $E_{max} = 25\%$ worked universally across these data sets.

Each of the four cognitive features is associated with one of the four diagonal element in the metric (tensor) matrix. For each cognitive feature, its importance is measured by the frequency of its associated diagonal elements in $> 90\%$ percentile of the set of all diagonal elements from the metric (tensor) matrices selected by the above procedure. The left panel in Figure 4.3 shows that the divided attention (i.e. t^d_{disp}) is the most discriminative feature for the classification task (MCI patients vs. healthy controls).

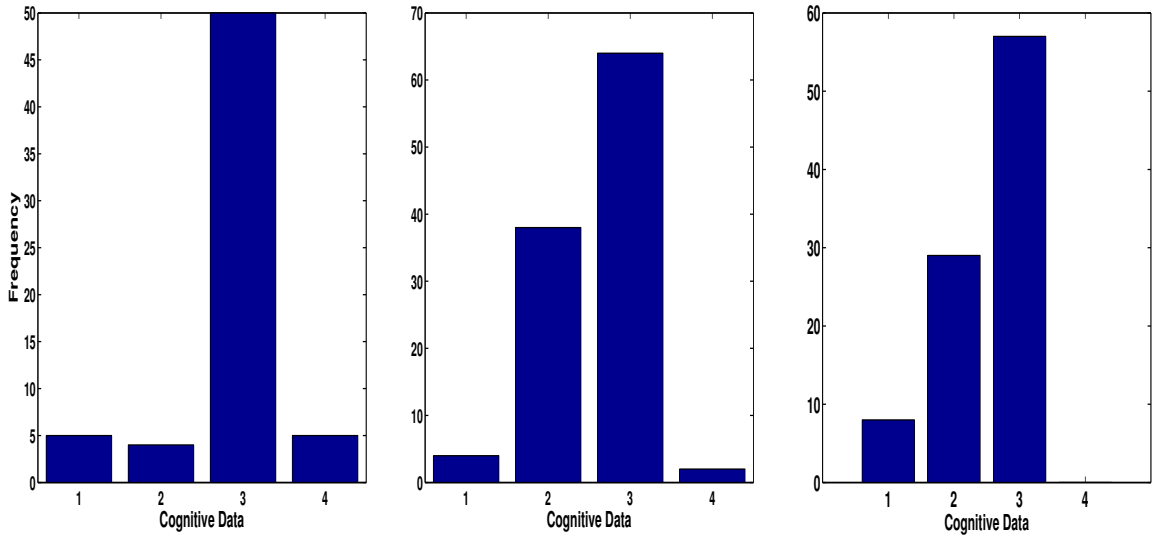

Figure 4.3: The importance histogram of the four cognitive features as follows: working memory (n_{dots}), cognitive inhibition (t_{delay}), divided attention (t^d_{disp}), and selective attention (t^s_{disp}) (numbered as 1, 2, 3, and 4 in the order). These features are used as the input to the following GMLVQ classifiers: M-CD, M^+-CD-PSC, and M^+-CD-FGF (from left to right). Note that each cognitive feature is associated with a diagonal element of the GMLVQ metric tensor matrix Λ and the importance histogram counts the number of each diagonal element in the $>90\%$ percentile of all diagonal elements from an ensemble of Λs.

Next, we studied the off-diagonal elements of those metric (tensor) matrices. Each off-diagonal element controls the interplay between two associated cognitive features.

To illustrate how this interplay works, we provide a toy example as follows: Denote a two-dimensional feature vector by $(x,\ y)$ and a 2×2 metric tensor by $\begin{pmatrix} \alpha & \gamma \\ \gamma & \beta \end{pmatrix}$. The distance between two feature vectors indexed by i and j is given by

$$d_{ij} = \underbrace{\alpha^2 \cdot (x_i - x_j)^2 + \beta^2 \cdot (y_i - y_j)^2}_{d_{ij}^M} + \underbrace{2\gamma \cdot (x_i - x_j)(y_i - y_j)}_{d_{ij}^2}. \tag{4.28}$$

The first two terms of d_{ij} is actually so-called Mahalanobis distance between the i-th and j-th feature vectors (denoted by d_{ij}^M). In the case of $\gamma = 0$, the diagonal term α and β are optimized by maximizing between-class Mahalanobis distances while minimizing within-class ones. When the metric matrix has non-zero off-diagonal elements, the distance measure has additional contribution d_{ij}^2 which can either enhance or collapse the total distance measure depending on *(i)* the sign of γ and *(ii)* the sign of *between-class* correlation (i.e. correlation between class-conditional means of x and y). For example, in the case of negative *between-class* correlation, negative γ can further enhance the class separation and vice versa.

To test whether the interplay between two cognitive features, indexed by i and j, is positive or negative, we performed two one-sided sign-rank tests for the hypotheses $\Lambda_{ij} > 0$ and $\Lambda_{ij} < 0$ (respectively) using the corresponding off-diagonal element from the selected GMLVQ metric (tensor) matrices. The upper-left panel of Figure 4.4 shows that there exists statistically significant, negative interplay between divided attention and two following cognitive features: (1) working memory (n_{dots}) and (2) cognitive inhibition (t_{delay}). From the lower-left panel, we found statistically significant, positive interplay between three cognitive features as follows: (1) working memory, (2) cognitive inhibition, and (3) selective attention (t^s_{disp}). Finally, note that there is no significant interplay between divided attention and selective attention.

To examine the relation between the interplay and *between-class* correlation revealed by Eq. 4.28, we need to determine whether or not there exists statistically significant *between-class* correlation between two of the four cognitive features. To this end, we first

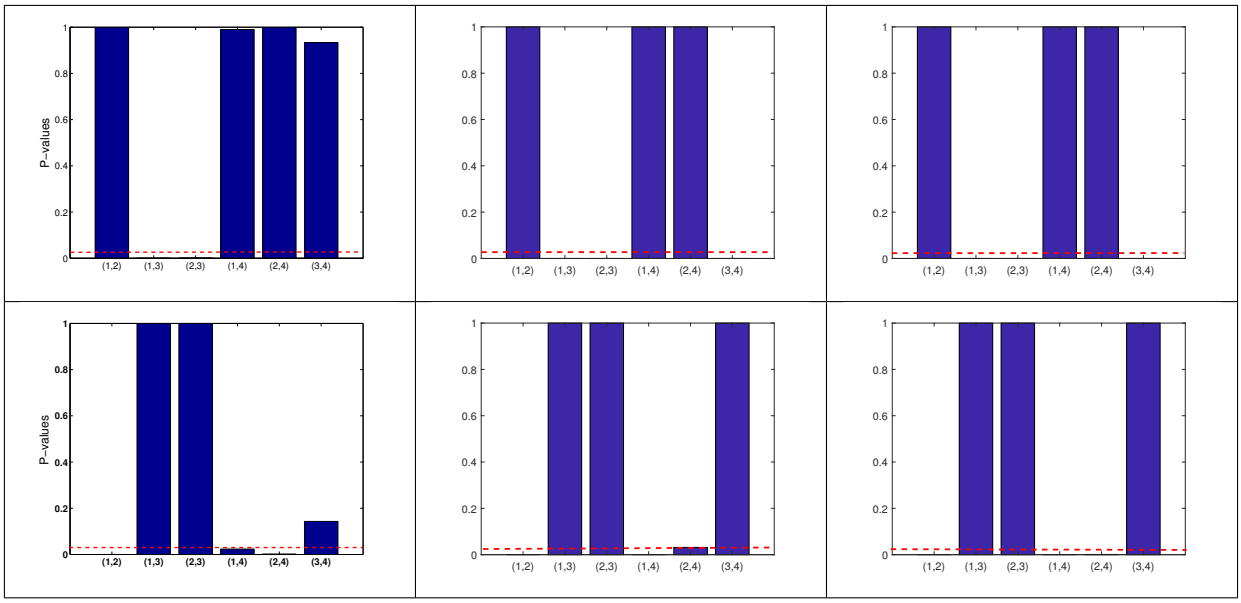

Figure 4.4: The p values of the one-sided sign-rank tests for studying the interplay between two of the following cognitive features: working memory (n_{dots}), cognitive inhibition (t_{delay}), divided attention (t^d_{disp}), and selective attention (t^s_{disp}) (numbered as 1, 2, 3, and 4 in the order). From each panel in the upper and lower row, one can read that if the p value is smaller than the threshold $p = 0.05$ (indicated by red dashed line), the interplay of two corresponding cognitive features is statistically significant and it takes a negative and positive value (respectively); These features are the inputs to three GMLVQ classifiers as follows: M-CD, M^+-CD-PSC, and M^+-CD-FGF (from left to right). Note that the tests used the off-diagonal elements of the GMLVQ metric tensor matrices.

used one-sided sign-rank test to determine, for each of the four features, whether its values for MCI patients are significantly larger or significantly smaller than those for healthy controls. For each pair of the cognitive features, if the outcomes of their tests are both statistically significant and are consistent with (or in opposite to) each other, then their *between-class correlation* is considered as positive (or negative). Otherwise, the *between-class* correlation is insignificant. From this analysis we observe (1) the class-conditional mean of working memory is positively correlated with that of cognitive inhibition; and (2) the class-conditional mean of divided attention is negatively correlated with that of working memory as well as that of cognitive inhibition. These observations agree with the observation of the interplay between the corresponding cognitive features, which can enhance the class separation. For the remaining pairs of the cognitive features, their *between-class correlation* is not significant. In Figure 4.5, we graphically illustrate the

presence or absence of these correlations.

In summary, though the divided attention seems to be the most relevant feature among the four cognitive features, all four features are indispensable for maximising the classification performance. This is because these exists *between-class correlation* between the features.

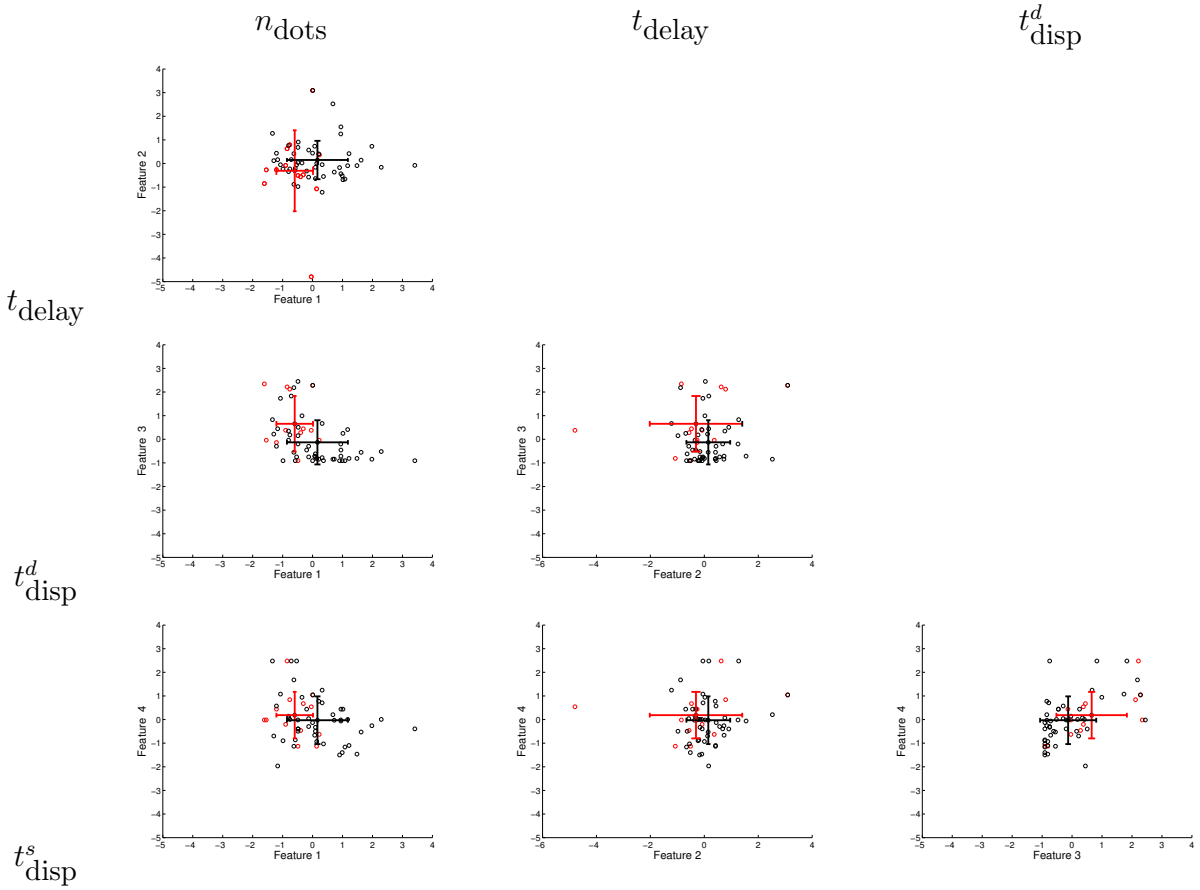

Figure 4.5: Scatter plot for six possible feature pairs from the four cognitive features as follows: Working memory (n_{dots}), Stop signal (t_{delay}), Divided attention (t^d_{disp}), and Selected attention (t^s_{disp}). For individual MCI patients and control subjects, their feature pairs (i.e. Feature 1 vs Feature 2) are displayed as red and blue dots (respectively). The corresponding class-conditional means and standard deviations are also displayed by coloured error bars. For each panel, the corresponding Feature 1 and Feature 2 are indicated at the top of each column and on the utmost left of each row (respectively).

fMRI features

We carried out the same relevance analysis for M-PSC, M-SGF, and M-FGF as for M-CD in Section 4.4.3. Recall that in these three experiments, the inputs to GMLVQ classifiers are comprised of six fMRI features as follows: **(i)** PSC-Cerebellar-Pre, PSC-Cerebellar-Post, PSC-Frontal-Pre, PSC-Frontal-Post, PSC-Subcortical-Pre, PSC-Subcortical-Post; **(ii)** SGF-Cerebellar-Pre, SGF-Cerebellar-Post, SGF-Frontal-Pre, SGF-Frontal-Post, SGF-Subcortical-Pre, SGF-Subcortical-Post; and **(iii)** FGF-Cerebellar-Pre, FGF-Cerebellar-Post, FGF-Frontal-Pre, FGF-Frontal-Post, FGF-Subcortical-Pre, FGF-Subcortical-Post (respectively). The fMRI feature "PSC-Cerebellar-Pre" denotes PSC feature that is derived from fMRI data measured in the cerebellar ROI and during the pre-training session. and the remaining fMRI features are abbreviated in the same way. Recall that PSC is referred to as Percent Signal Change, SGF as Spatially grouped Graph Feature and FGF as Functionally grouped Graph Feature.

Figure 4.6 shows that PSC-Frontal-Post and FGF-Frontal-Pre are the most discriminative fMRI feature in Experiment M-PSC and M-FGF (respectively). We first note that the most relevant feature in both cases is derived from the frontal ROI (that is, the largest ROI among the three ROIs used in this study). It is more interesting to address two following questions: (1) why is the post-training session is more relevant than the pre-training one, when PSC is used for the task; and (2) why is the opposite true when the graph feature is used for the task.

The left panel in Figure 4.7 shows that before training, the PSC level for MCI patients and healthy controls are on average comparable. However, training caused a remarkable increase of the PSC level for MCI patients but not for healthy controls. As a result, these two participant groups differ in their PSC level after the training. This is why PSC-Frontal-Post is identified as the most relevant feature for Experiment M-PSC. The right panel in Figure 4.7 shows that the graph feature FGF differs between MCI patients and healthy controls before training. This could be related to the suggestions that MCI may have caused changes in brain connectivity. We further observe that for both participant

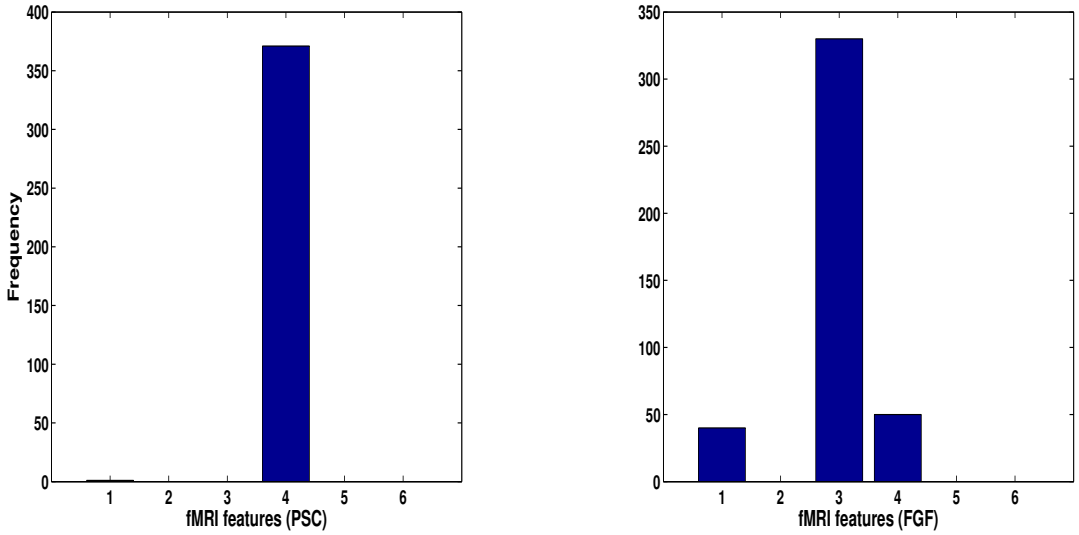

Figure 4.6: Left panel: The importance histogram of the six fMRI features as follows: PSC-Cerebellar-Pre, PSC-Cerebellar-Post, PSC-Frontal-Pre, PSC-Frontal-Post, PSC-Subcortical-Pre, and PSC-Subcortical-Post. (numbered as 1, ..., and 6 in the order). PSC is referred to as Percent Signal Change, Pre as Pre-training session, Post as Post-training session, Cerebellar (Frontal and Subcortical) as the cerebellar(frontal and subcortical, respectively) ROI. For example, PSC-Cerebellar-Pre means that the fMRI data were acquired before training and PSC feature was extracted from the cerebellar ROI). Right panel: The same as in the left panel but for the following fMRI features: FGF-Cerebellar-Pre, FGF-Cerebellar-Post, FGF-Frontal-Pre, FGF-Frontal-Post, FGF-Subcortical-Pre, and FGF-Subcortical-Post.

groups, training increased their FGF values but to different extents. After training, the difference between MCI patients and healthy controls became much less significant. This is why FGF-Frontal-Pre is identified as the most relevant feature for Experiment M-FGF. This observation allows us to speculate that training could "mitigate" the changes in brain connectivity caused by MCI.

The above analysis suggests that brain connectivity may have changed after training and this is significant particularly for MCI patients. In the following, we address the question whether a sub-network rather than the entire (local) network within the frontal ROI has changed. Recall that all 128 voxels in the frontal ROI are grouped into 7 spatially contiguous clusters. This results in a local brain network consisting of 7 nodes and 21 edges. Each off-diagonal element of the graph matrix G quantifies the connectivity between two nodes and measures the strength of the corresponding edge. Recall that the graph features

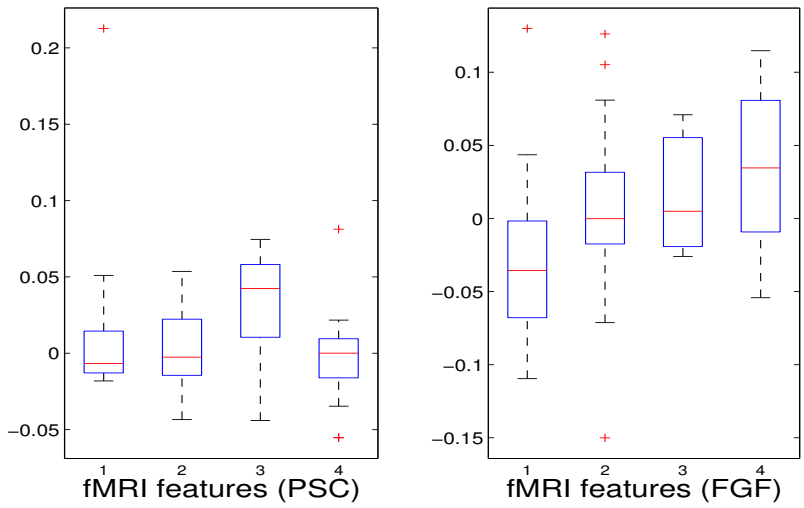

Figure 4.7: Left: Boxplot of the following fMRI features: FGF-Frontal-Pre for MCI patients, FGF-Frontal-Pre for healthy controls, FGF-Frontal-Post for MCI patients, and FGF-Frontal-Post for healthy controls (numbered as 1, 2, 3 and 4 in the order). Note that the y-axis represents the values of the corresponding fMRI features; Right: Boxplot of the following fMRI features: PSC-Frontal-Pre for MCI patients, PSC-Frontal-Pre for healthy controls, PSC-Frontal-Post for MCI patients, and PSC-Frontal-Post for healthy controls (numbered as 1, 2, 3 and 4 in the order).

FGF were extracted by applying multiplicative method. To this end, multiplicative method provides two feature-generating vectors $\mathbf{a}$ and $\mathbf{b}$ from which we can derive a task-dependent importance matrix denoted by I as follows:

$$I = \frac{1}{2}(\mathbf{a}\mathbf{b}^\intercal + \mathbf{b}\mathbf{a}^\intercal). \tag{4.29}$$

Each off-diagonal element of I measures the importance of the corresponding edge in terms of discriminating MCI patients from healthy controls. To identify possible sub-networks that have significantly changed after training, we are first to identify the edges whose importance measure has significantly changed after training. To this end, we generated an ensemble of the selected importance matrices using the procedure that was used to generate an ensemble of the selected GMLVQ metric (tensor) matrices for the relevance feature analysis. Subsequently, we conducted two one-sided sign rank tests for each of the 21 edges

to find those edges whose importance values have significantly increased or reduced after training. Denote the edge connecting node i and j by E_{ij}. This analysis revealed that the importance measure of three following edges has significantly increased: E_{17}, E_{16} and E_{64}. A significant reduction of its importance measure was observed for E_{65}. These four edges are displayed in Figure 4.8. Figure 4.9 highlighted a subtle difference between the sub network (i.e. E_{17}, E_{16} and E_{64}) and the single edge E_{65}. For the three-node sub-network, the connectivity strength is highest for MCI patients before training. For the single edge E_{65}, the connectivity strength is lowest for healthy controls before training. This suggests that FGF-Frontal-Pre, the most relevant feature in M-FGF, could be related to these three-node and single-node sub-networks.

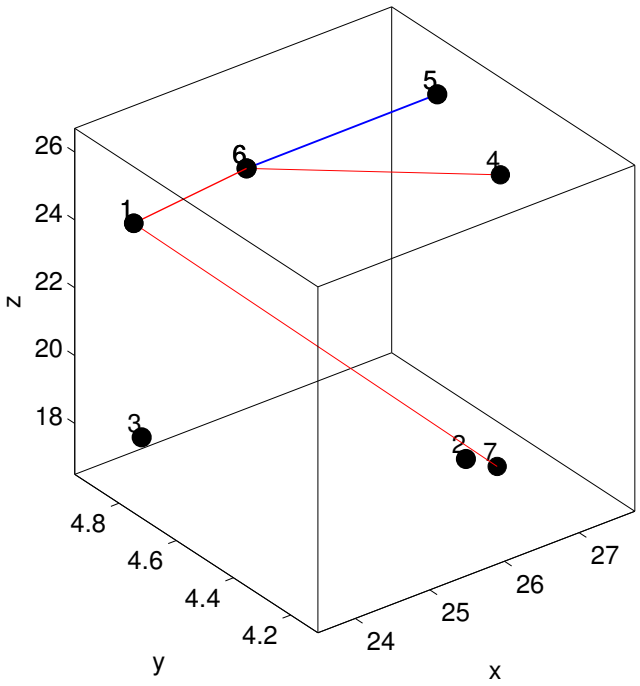

Figure 4.8: The node configuration for the frontal ROI which includes Superior Frontal Gyrus on the right hemisphere and Medial Frontal Gyrus on the left hemisphere. The straight lines indicate the edges whose importance for discriminating MCI patients from healthy controls has significantly changed. For the three-node subnetwork (indicated by red lines), its importance has increased after training. In contrast, the single-node subnetwork (indicated by blue line), training has reduced its importance.

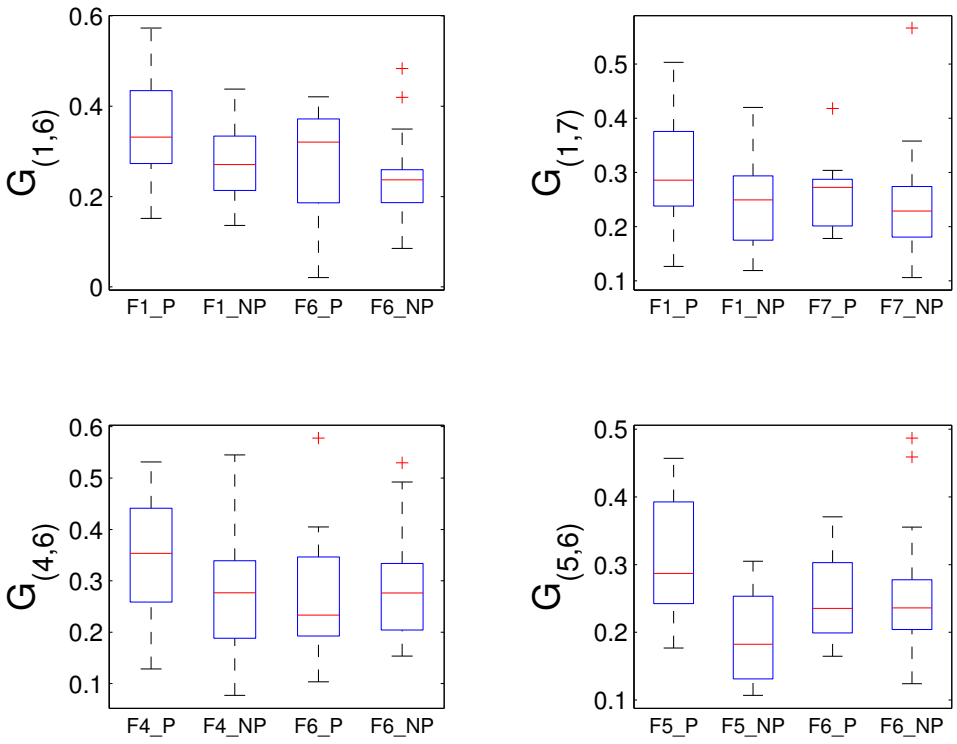

Figure 4.9: For the graph matrices generated in this study, we display four of their matrix elements which are associated with the four edges highlighted in Figure 4.8. $G_{1,6}$ in the upper-left panel, $G_{1,7}$ in the upper-right panel, and $G_{4,5}$ in the lower-left panel measure the connectivity of edge $E_{1.6}$, $E_{1,7}$ and $E_{4,5}$ (respectively) that form the three-node sub-network. Recall that the task-related importance of this sub-network has significantly increased after training. In contrast, $G_{5,6}$ in the lower-right panel measures the connectivity of edge $E_{5.6}$ and its task-related importance has significantly reduced after training. The four boxplots in each panel are associated with pre-training session & patient group, pre-training session & control group, post-training session & patient group, and and post-training session & control group (from left to right, numbered as 1, 2, 3, and 4 in the order).

Privileged information

In addition to M-CD, M-PSC and M-FGF, M^+-CD-PSC and M^+-CD-FGF were conducted to investigate GMLVQ classification of MCI patients and controls when fMRI features were incorporated as privileged information. The relevance of the four cognitive features in M^+-CD-PSC and M^+-CD-FGF was estimated from the diagonal elements of the metric tensors and displayed in the middle and right panel of Figure 4.3 (respectively).

Though PSC and FGF are two different kinds of fMRI features, we still consistently observed that cognitive inhibition and divided attention are the two most relevant cognitive features. Moreover, the relevance of divided attention is more profound than that of cognitive inhibition. When compared to M-CD, cognitive inhibition did emerge as a relevant feature only when the privileged information was incorporated. Also, Figure 4.4 shows that when compared to M-CD, the interplay between divided attention and selective attention became significantly positive in M^+-CD-PSC and M^+-CD-FGF, that is, the experiments in which the privileged information was incorporated.

4.4.4 Comparison of GMLVQ with SVM and SVM+ classifiers

The GMLVQ algorithm was compared against the SVM and SVM+ based models (explained in section 4.3.2). To run SVM and SVM+, it is required to cross validate tuning parameters (hyper-parameters), these are the kernel widths of the decision and slack model (correcting) functions for both cases without and with PI, and the regularization parameters. From Tables 4.4 and 4.5 it can be clarified that GMLVQ achieves a relatively better performance over the SVM and SVM+. The results are comparable to GMLVQ (Tables 4.1 and 4.2), especially in the cases where PI is incorporated in the training stage, than the results are good over CD. Similar to GMLVQ there is a statistically significant for cases M+-CD-PSC and M+-CD-SGF, while the less promising combination is in the case of M+-CD-FGF.

Models	Mean	Std-Dev	Median	(25%, 75%) Percentile
M-CD	0.41	0.08	0.40	(0.34, 0.47)
M-PSC	0.25	0.16	0.23	(0.16, 0.35)
M-SGF	0.30	0.08	0.30	(0.23, 0.38)
M-FGF	0.26	0.10	0.25	(0.23, 0.33)
M-PSC+SGF	0.27	0.10	0.28	(0.21, 0.33)
M-PSC+FGF	0.25	0.10	0.24	(0.20, 0.33)

Table 4.4: Classification performance measured by MMAE for the baseline classifier, M-CD, and five different M-PD classifiers (see Column 1). For each classifier, we report both mean MMAE, its standard deviation, median MMAE and its (25%, 75%) percentile in Column 2 – 5, respectively. They were computed using the MMAE estimates obtained from 50 randomly created training-test splits, the results of SVM.

Models	Mean	Std-Dev	Median	(25%, 75%) Percentile
M^+-CD-PSC	0.36	0.08	0.35	(0.29, 0.44)
M^+-CD-SGF	0.32	0.10	0.32	(0.23, 0.38)
M^+-CD-FGF	0.37	0.12	0.37	(0.24, 0.40)
M^+-CD-PSC+SGF	0.35	0.08	0.33	(0.29, 0.40)
M^+-CD-PSC+FGF	0.34	0.11	0.34	(0.27, 0.38)

Table 4.5: The same as in Table 4.4 but for evaluation of the classification performance of five different M^+-CD-PD classifiers, that is, the classifiers using CD as their inputs and PD as privileged information using SVM+.

4.5 The Value of Additional Features

4.5.1 Extracting fMRI Features within ROIs

We focused on examining networks across ROIs rather than studying networks within ROIs, but we did not have any improvement with second features. For this reason, we proposed a method that based on the assumption that a Region of Interest (ROI) is not functionally homogeneous. Therefore, each ROI is represented by more than one functionally homogeneous cluster, and the aim is to achieve a result in which each node is functionally homogeneous. The proposed method entails constructing 8×8 graph matrix by clustering a cerebellar ROI with 2 centroids,a frontal ROI with 4 centroids, and a subcortical ROI with 2 centroids. These different unique configurations of each ROI are used to compute one soft kernel of order8×8 with the distribution based on the number of voxels in each region. We applied this approach both for the case of voxels clustering based on their common function, that is FGF, and for the case of voxels clustering based on their proximity, that is SGF.

4.6 Experiments of Mix ROIs together for both First and Second features

4.6.1 Experimental Design and Setup

Both of these parts are the same as Section 4.3.3 and Section 4.4.1. In the experimental design, the only difference here is that we have experiments of feature 1 $v1$ and also experiments of feature 2 $v2$, data classification proceeds first based on a single feature $v1$ and a two dimensional feature vector $V2 = (v1, v2)$.

4.6.2 Classification Results

Using the Multiplicative Criterion

In comparison with when the experiment is conducted on only M-CD where MMAE result is 0.39 and its standard deviation is ± (0.09). Table 4.6 illustrates that using the second feature can provide a better percentage of improvements. For example, in the case of M^+-CD-FGF_{v1} there is percentage of improvement only 0%; whereas using a second feature, as in M^+-CD-FGF_{V2} it is 8%. This demonstrates that the second feature is needed as the number of miss-classification errors has reduced. Additionally, in the case of SGF the percentage of improvement is 8% with M^+-CD-SGF_{V2}, compared to the case of M^+-CD-SGF_{v1} it is only -3%.

Using the Additive Criterion

In the case of using the additive criterion, as shown in table 4.7, the miss-classification errors increased more than with the multiplicative method because of λ, which is regularisation parameter. It is determined by cross validation and it was chosen between $\{1 - 9\}$.The minimum number of miss-classification errors are given when $\lambda = 7$ with the learning rate=.5; for that reason, it is fixed for all the experiments. For example, the percentage of improvement is -21% with M-SGF_{v1}, while it is -3% with M-SGF_{V2}, however, the

Models	Mean	Std-Dev	Median	(25%, 75%) percentile
M-SGF_{v1}	0.40	± 0.14	0.39	(0.30, 0.54)
M-FGF_{v1}	0.23	± 0.13	0.23	(0.21, 0.33)
M-SGF_{V2}	0.35	± 0.12	0.38	(0.30, 0.40)
M-FGF_{V2}	0.17	± 0.11	0.16	(0.07,0.30)
M^+-CD-SGF_{v1}	0.36	± 0.10	0.36	(0.27, 0.44)
M^+-CD-FGF_{v1}	0.38	± 0.12	0.39	(0.31, 0.49)
M^+-CD-SGF_{V2}	0.33	± 0.10	0.33	(0.24, 0.39)
M^+-CD-FGF_{V2}	0.33	± 0.14	0.36	(0.26,0.49)

Table 4.6: MMAE results of extracting fMRI, $v1$ and $V2 = (v1, v2)$ by using Multiplicative criterion using GMLVQ classifier.

second features are decreased the the miss-classification errors. So far, our results show us that the multiplicative approach is much better for this purpose; the reason may be because there is no parameter for cross validation (tuning parameters).

Models	Mean	Std-Dev	Median	(25%, 75%) percentile
M-SGF_{v1}	0.40	± 0.15	0.47	(0.27, 0.50)
M-FGF_{v1}	0.37	± 0.09	0.38	(0.31, 0.40)
M-SGF_{V2}	0.38	± 0.12	0.38	(0.28,0.47)
M-FGF_{V2}	0.32	± 0.16	0.23	(0.20, 0.35)
M^+-CD-SGF_{v1}	0.47	± 0.07	0.40	(0.36, 0.52)
M^+-CD-FGF_{v1}	0.42	± 0.05	0.44	(0.36,0.55)
M^+-CD-SGF_{V2}	0.43	± 0.12	0.52	(0.31, 0.48)
M^+-CD-FGF_{V2}	0.47	± 0.09	0.50	(0.44, 0.47)

Table 4.7: MMAE results of extracting fMRI, $v1$ and $V2 = (v1, v2)$ by using additive criterion using GMLVQ classifier.

Comparing our Methods with 2D-LDA

By comparing tables 4.6, 4.7 with4.8, for multiplicative, additive and 2D-LDA approaches respectively; they clarify that the multiplicative and 2D-LDA methods are better than the additive method. For example, M-FGF_{v1} and M-FGF_{V2} in the case of the multiplicative

approach, the percentage of improvements is 41% and 59% receptively, and for the 2D-LDA case it is 51% and 59%. However, the additive method shows 3% and 41% for M-FGF_{v1} and M-FGF_{V2} respectively.

Additionally, M^+-CD-SGF_{v1} and M^+-CD-SGF_{V2} for the case of the multiplicative approach, the percentage of improvements is -3% and 8% respectively, and for the 2D-LDA case is 8% and 15%. Nonetheless, the additive method gives -3% and -52% for M^+-CD-SGF_{v1} and M^+-CD-SGF_{V2} respectively. This proves that the additive method is not as good as the other two methods. For this reason, the next compromise and analysis will be between the multiplicative and 2D-LDA approaches.

Models	Mean	Std-Dev	Median	(25%, 75%) percentile
M-SGF_{v1}	0.39	$\pm$ 0.12	0.38	(0.30, 0.5)
M-FGF_{v1}	0.24	$\pm$ 0.16	0.19	(0.14, 0.40)
M-SGF_{V2}	0.40	$\pm$ 0.12	0.38	(0.28,0.50)
M-FGF_{V2}	0.21	$\pm$ 0.12	0.16	(0.14, 0.30)
M^+-CD-SGF_{v1}	0.40	$\pm$ 0.13	0.40	(0.31, 0.48)
M^+-CD-FGF_{v1}	0.38	$\pm$ 0.12	0.38	(0.27,0.48)
M^+-CD-SGF_{V2}	0.34	$\pm$ 0.11	0.36	(0.24, 0.44)
M^+-CD-FGF_{V2}	0.39	$\pm$ 0.10	0.42	(0.31, 0.47)

Table 4.8: MMAE results of extracting fMRI, $v1$ and $V2 = (v1, v2)$ by using 2D-LDA using GMLVQ classifier.

4.6.3 Comparing two Approaches in case of Mix ROIs

In the case of one prototype for both the multiplicative criterion and the 2D-LDA method, we used the left-side sign rank test for both with each MMAE corresponding to the MMAE in the second method. This was carried out in order to test that the MMAE of the multiplicative is smaller than the MMAE of the 2D-LDA method. The results are as follows in table 4.9. It can be seen that the multiplicative criterion is better than 2D-LDA in extracting the second features for the case of integrating PI along CD in the training stage. There are some improvements in the case of M^+-CD-FGF_{v1} and M^+-CD-FGF_{V2}

for the multiplicative approach; while there is a less promising combination for the same cases CD-FGF_{v1} and M^{+}-CD-FGF_{V2} for 2D-LDA.

Models	p-value
M-SGF_{v1}	0.62
M-FGF_{v1}	0.37
M-SGF_{V2}	0.07
M-FGF_{V2}	0.05
M^{+}-CD-SGF_{v1}	0.89
M^{+}-CD-FGF_{v1}	0.55
M^{+}-CD-SGF_{V2}	0.93
M^{+}-CD-FGF_{V2}	0.00

Table 4.9: Left side sign rank test for both multiplicative and 2D-LDA methods

4.6.4 SVM and SVM+ for the multiplicative approach

The same experiments from Table 4.6 were repeated by SVM and SVM+ in Table 4.10. The misclassifications rates of GMLVQ were compared with the SVM and SVM+ approaches. In general, the obtained results agree with the previous findings that the classification performance of GMLVQ/SVM+ is improved by incorporating fMRI (as privileged information).

4.7 Conclusion

In this study, we employed GMLVQ classifiers to discriminate cognitive skills in MCI patients vs. healthy controls using cognitive and/or fMRI data. Specially, we have adopted a "Learning with privileged information (PI)" approach to combine cognitive and fMRI data. In this setting, fMRI data as an addition to cognitive data are only used to train GMLVQ classifier and classification of a new participant is solely based on cognitive data. As the inputs to GMLVQ classifier, the cognitive features include working memory,

Models	Mean	Std-Dev	Median	(25%, 75%) percentile
M-SGF_{v1}	0.42	± 0.12	0.42	(0.35, 0.49)
M-FGF_{v1}	0.24	± 0.11	0.23	(0.21, 0.30)
M-SGF_{V2}	0.38	± 0.11	0.40	(0.30, 0.43)
M-FGF_{V2}	0.20	± 0.11	0.21	(0.14,0.30)
M^+-CD-SGF_{v1}	0.37	± 0.10	0.40	(0.30, 0.44)
M^+-CD-FGF_{v1}	0.39	± 0.11	0.39	(0.31, 0.44)
M^+-CD-SGF_{V2}	0.34	± 0.11	0.36	(0.24, 0.44)
M^+-CD-FGF_{V2}	0.37	± 0.10	0.37	(0.29,0.46)

Table 4.10: MMAE results of extracting fMRI, $v1$ and $V2 = (v1, v2)$ by using SVM and SVM+ classifiers.

cognitive inhibition, divided attention and selective attention scores. Also, we extracted three different types of fMRI features from fMRI data as follows: PSC (percent signal change), and SGF (spatially grouped graph feature) and (functionally grouped graph feature).

We are well aware that our data is small and the reported results are indicative of improvement of integrating PI over CD in the training stage. Of course, it would be better if we had a larger data set (many more subjects). Our main question was whether fMRI as PI can help CD. Indeed, the p-values showed that there is a statistically significant improvement for the performances when PI is used in the training phase. We first tested our baseline GMLVQ classifier with four cognitive features as inputs. Its classification performance is measured by (25%, 75%) percentile of Macro-averaged Mean Absolute Error (MMAE), that is, (0.32, 0.44). The best of the five fMRI GMLVQ classifiers (i.e. the ones using the fMRI features as their inputs) yields a lower bound of classification error, which is (0.16, 0.30). Interestingly, the best of the PI-guided GMLVQ classifiers (i.e. the ones using the four cognitive features as their inputs and using the fMRI features as privileged information) have achieved (0.26. 0.40). This seems to show that incorporating fMRI features as privileged information may can significantly improve the classification performance of a baseline GMLVQ classifier for classification of cognitive skills in MCI

patients vs. controls.

Crucially, we have also performed "relevant feature analysis" for all three GMLVQ classifiers: the baseline GMLVQ classifier, the best fMRI-guided GMLVQ classifier, and the fMRI GMLVQ classifier. For the baseline classifier, "divided attention" is the only relevant cognitive feature for the classification task. When the privileged information is incorporated, divided attention remains the most relevant feature while cognitive inhibition becomes also relevant. The above results may suggest that attention-rather than only memory-plays an important role for the classification task. More interestingly, this analysis for the fMRI GMLVQ classifier suggests that (1) among three ROIs used, the frontal ROI seems to be the most relevant for the classification task; (2) when the PSC feature as an overall measure of fMRI response to structured stimuli is used as the inputs to the classifier, the post-training session seems to be the most relevant; and (3) when the graph feature reflecting underlying spatiotemporal fMRI pattern is used, the pre-training session seems to be the most relevant. Further analysis has indicated that training may cause an overall increase of the brain activity only for MCI patients while it may have "mitigated" the difference in brain connectivity pattern between MCI patients and healthy controls. Moreover, these training-dependent changes seem to be the most significant for a three-node sub-network in the frontal ROI. Taken together these results suggest that brain connectivity before training and overall fMRI signal after training are both diagnostic of cognitive skills in MCI.

The GMLVQ classifier was compared against SVM and SVM+, and the results were sometimes better and sometimes worse; it seems that they are comparable to show that fMRI as PI can help to learn the classifier over CD. Results were evaluated by utilizing a paired Wilcoxon signed-rank test, and in both classifiers GMLVQ and SVM+, there are statistically significant improvements in the cases of M^+-CD-PI.

Our study employs machine learning algorithms to investigate the neurocognitive factors and their interactions that mediate learning ability in Mild Cognitive Impairment. Our work is not limited to developing and validating machine learning approaches; in

contrast it advances our understanding of the neurocognitive mechanisms that mediate learning in health and disease. For example, the role of cognitive inhibition in cognitive profile classification seems to be significantly enhanced when brain imaging information (obtained in a sequence learning prediction task) is provided as privileged information. This opens questions about the possible interplay between circuits involved in cognitive inhibition and those involved in learning sequence prediction tasks. We also observed significant positive interplay between divided and selective attention when brain imaging data is used as privileged information. No such interplay was detected without the privileged information. Again, this raises interesting questions regarding circuitry involved in sequence prediction and the two attention types.

This chapter also compared our methods (multiplicative and additive) with 2D-LDA in extracting the second features and compared its miss-classification errors with the extracted first feature. The reported results illustrate that extracting the second features reduced the miss-classification errors.

4.8 Chapter Summary

In this chapter, we employed Generalised Matrix Learning Vector Quantization (GMLVQ) classifiers to discriminate patients with Mild Cognitive Impairment (MCI) from healthy controls based on their cognitive skills. Further, we adopted a "Learning with privileged information" approach to combine cognitive and fMRI data for the classification task. The resulting classifier operates solely on the cognitive data while it incorporates the fMRI data as privileged information (PI) during training. This novel classifier is of practical use as the collection of brain imaging data is not always possible with patients and older participants.

MCI patients and healthy age-matched controls were trained to extract structure from temporal sequences. We ask whether machine learning classifiers can be used to discriminate patients from controls based on the learning performance and whether

differences between these groups relate to individual cognitive profiles. To this end, we tested participants in four cognitive tasks: working memory, cognitive inhibition, divided attention, and selective attention. We also collected fMRI data before and after training on the learning task and extracted fMRI responses and connectivity as features for machine learning classifiers.

Our results show that the PI guided GMLVQ classifiers outperform the baseline classifier that only used the cognitive data. In addition, we found that for the baseline classifier, "divided attention" is the only relevant cognitive feature. When PI was incorporated, divided attention remained the most relevant feature while cognitive inhibition became also relevant for the task. Interestingly, this analysis for the fMRI GMLVQ classifier suggests that (1) when overall fMRI signal for structured stimuli is used as inputs to the classifier, the post-training session seems to be the most relevant; and (2) when the graph feature reflecting underlying spatiotemporal fMRI pattern is used, the pre-training session seems to be the most relevant. Further analysis reveals that for MCI patients, training may alter brain activation level as well as local brain connectivity pattern. Taken together these results may suggest that brain connectivity before training and overall fMRI signal after training are both diagnostic of cognitive skills in MCI. Moreover, we compared our methods (multiplicative and additive) with 2D-LDA to examine whether extracting the second features can decrease miss-classification rates compared to extracting the first features.

CHAPTER 5

EXPERIMENTS ON SYNTHETIC DATA

5.1 Introduction

The use of synthetic data sets for validation purpose, rather than real data, is commonly practiced in many research areas. Particularly, under circumstances where it is impossible to acquire actual data, due to time, budget or privacy concerns, artificial data can be used as a practical replacement. Synthetic data can be a good surrogate for real data, especially since it offers a controlled testing environment that meets specific, well defined conditions. This feature is very useful for proof of concept, purposes of verification or simulation. The synthetic data is needed in this thesis because in our case the real data is not readily available and the development of our algorithms was established before the real data becomes available.

Recall that the greedy tensor LDA algorithm developed in this PhD work generates discriminative features sequentially. When the newly generated feature cannot help further improve the task performance, the feature-generating process can be terminated. If this happened and the achievable performance remains low, we ask whether it is due to lack of information in the data or it is because the greedy algorithm fails to extract the remaining information hidden in the data. To answer this research question, it is absolutely necessary to use synthetic data which is generated in a controlled manner.

In this chapter, we show how the synthetic second and third order tensor data is

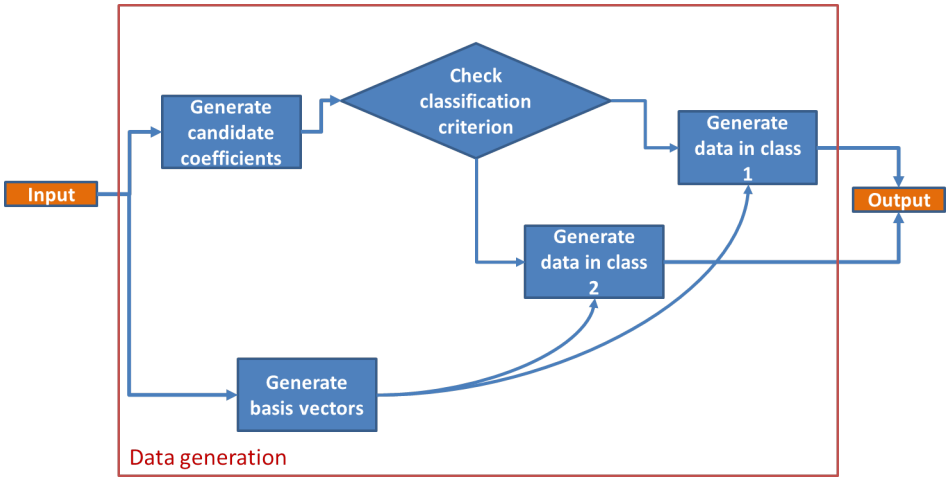

Figure 5.1: Pipeline scheme for the creation of the synthetic data.

constructed and we evaluate the performance of our methods using this data set. Additionally, we compare our method with the ORO method [43] that is also greedy tensor LDA algorithm like the EGFE method that we proposed by applying both methods on the same third order tensor data.

5.2 Synthetic Data Construction

Figure 5.1 depicts the pipeline of the data creation process. It illustrates in details how we synthetically generate 2nd and 3rd order tensors datasets. The input required for this process consists of the order of tensors, the dimensions of individual modes, and the number of data samples.

Essentially, the data construction proceeds by selecting at random a set of rank-1 tensors, parameterized by a number of randomly generated parameter values, and using some criterion to separate the generated tensors into two classes. The criterion is a nonlinear condition on the parameter values. For all the data that was used in our experiments, we set the dimension of all modes to $d = 6$. Notice that although the data has very high dimension ($d^2 = 36$ for order two tensors or $d^3 = 216$ for order three tensors), it is parameterized by a small number of parameters. The values of these parameters are used to divide the data into classes using some nonlinear criterion. Therefore, these

parameters should not be confused with the features extracted from the generated data. Those features are used for classification using a linear classifier, whereas the true class of the data depends in a nonlinear manner on the parameters that were used for data generation.

The different data sets are named according the following convention: the set D*xy[C][F]* stands for a set of tensor data of order x and using y parameters. In our case $x = 2, 3$ and $y = 2, 3$. An optional additional character C is used to indicate some additional properties of the data, as necessary to distinguish between different sets with the same x and y values. An optional additional character is used to indicate a data set that includes failures: $F = O$ indicates data with overlapping and $F = R$ indicates data with outliers.

5.2.1 Tensor data of order 2

Two parameters: the D22 data set

In this case, we generated a set of matrices $G \in R^{d \times d}$ by first choosing at random four vectors $X_1, X_2, X_3, X_4 \in R^d$ and then orthonormalising them with Gram-Schmidt algorithm. Subsequently, the set of data is generated using the relation

$$G = aX_1X_2^T + bX_3X_4^T, \tag{5.1}$$

where the parameters a and b are randomly generated using a Gaussian random number generator with zero mean and unit variance. The boundary line between two classes in (a, b)-plane is chosen to be of parabola shape. In case of $b > a^2$, the corresponding matrix G was labeled as *Class 1*, whereas those with $b < a^2$ the matrix is labeled as *Class 2*.

Three parameters: the D23 data set

In this case, six random and orthonormal vectors $X_1, X_2, X_3, X_4, X_5, X_6 \in R^d$ were chosen and the data was generated using the relation

$$G = aX_1X_2^T + bX_3X_4^T + cX_5X_6^T, \tag{5.2}$$

where the three parameters a, b and c are randomly generated using a Gaussian random number generator. The classification criteria was if $a^2 - b^2 > c$, the matrix G was included in *Class 1*, and otherwise in *Class 2*. Since the classification boundary in the a, b plane is a hyperbola, we call the data generated in this way, the D23 set.

A data set with overlapping classes D23O was also generated by using the criterion above only in case $|a^2 - b^2 - c| > \delta_{ov}$. Otherwise, if $|a^2 - b^2 - c| \leq \delta_{ov}$, then the data is included in *Class 1* with probability p and in *Class 2* with probability $1 - p$.

A data set with outlier classes D23R was also generated by adding a small proportion of outliers to the original data set. An outlier is generated in the same way as described above except the random numbers a, b and c are scaled in such a way that $|c - a^2 - b^2| > M_{outlier}$. Subsequently, the outlier data is labeled as *Class 1* if $a^2 - b^2 < c$ and as *Class 2* if $a^2 - b^2 > c$, which is the opposite criterion as the regular data. The number of outliers is a small fraction $p_{outlier} \lll 1$ from the total number of data.

Two parameters with parameterized classification criterion: the D22C data set

In the following, we describe a procedure that generates a population of rank-3 matrices with two free parameters accounting for individual variability. For this purpose, six orthonormal vectors $X_1, X_2, X_3, X_4, X_5, X_6 \in R^d$were randomly generated. They were subsequently used to construct three rank-1 matrices as follows: $B_1 = X_1X_2^T, B_2 = X_3X_4^T$ and, $B_3 = X_5X_6^T$. Following this, two rank-2 matrices were constructed by $BA = w_1B_1 + w_2B_2$ and $BB = w_2B_2 + w_3B_3$ where w_1, w_2, and w_3 are randomly chosen real numbers but kept

fixed for the experiment. Finally, the data, that is, rank-3 matrices, were generated as follows: $G = aBA + bBB$ where a and b are the two free parameter. Next, we describe how two classes of rank-3 matrices are defined. It is done via definition of two classes of points (a, b) in a two-dimensional plane. First, we generate a number of (a, b) pairs by sampling from a two-dimensional isotropic Gaussian distribution with some variance parameter σ^2. The points are to be divided into two groups through a smooth curve on the plane, that is, $a^2 - b^2 = c$ where c denotes some curve parameter. Concretely speaking, the rank3-matrices are labelled as *Class 1* when their corresponding a and b is subject to$a^2 - b^2 > c$,and vice versa. This data set is referred to as D22C because (1) two free parameters were used to specify individual rank-3 matrices; and (2) they were partitioned into two classes.

5.2.2 Tensor Data of order 3

To illustrate the capacity of the proposed method for higher order tensor data, we generate a few data sets with third order tensor data. This case was sufficient to demonstrate the main salient points of our method so data of order higher than three was not considered.

Three parameters: the D33 data set

In this case, six random and orthonormal vectors $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9 \in R^d$ were chosen and the data was generated using the relation

$$G = aX_1 \circ X_2 \circ X_3 + bX_4 \circ X_5 \circ X_6 + cX_7 \circ X_8 \circ X_9, \tag{5.3}$$

where the three coefficients a, b and c are randomly generated using a Gaussian random number generator. The boundary surface between the two classes in Three space (a, b, c) is chosen to be of hyperbolic paraboloid. Accordingly, we label (a, b, c) with $a^2 - b^2 > c$ as *Class 1* and those with $a^2 - b^2 < c$ as *Class 2*.

A data set with overlapping classes D33O was also generated by using the criterion

above only in case $|c - a^2 - b^2| > \delta_{ov}$. Otherwise, if $|c - a^2 - b^2| \leq \delta_{ov}$, then the data is included in *Class 1* with probability p and in *Class 2* with probability $1 - p$.

A data set with outlier classes D33R was also generated by adding a small proportion of outliers to the original data set. An outlier is generated in the same way as described above except the random numbers a, b and c are scaled in such a way that $|c - a^2 - b^2| > M_{outlier}$. Subsequently, the outlier data is labeled as Class 1 if $a^2 - b^2 < c$ and as *Class 2* if $a^2 - b^2 > c$, which is the opposite criterion as the regular data. The number of outliers is a small fraction $p_{outlier} \lll 1$ from the total number of data.

5.3 Graph Models

To further validate our greedy feature extraction algorithm, we also generated a "random graph" dataset with two explicitly defined classes of random graph.

The graph considered here (say Gr) consists of 16 nodes. Each of these 16 nodes is represented by a point in a two-dimensional plane. Let's denote this point pattern by $\mathbf{G} = \{G_i : i = 1, ..., 16\}$. Moreover, G is arranged as a 4×4 lattice grid within the unit square $[01] \times [01]$. That is,

$$G_i^1 = ... \quad \text{and} \quad G_i^2 = ...$$

Mathematically, this graph is described by a weight matrix (say W) of size 16×16 where W_{ij} represents the connection strength between node i and node j.

To define a graph structure on G, we randomly generate a (irregular) point pattern of size N over the unit square (say point pattern $X = \{X_1, ..., X_N\}$). These points could be generated uniformly over the unit square or otherwise. To define a graph structure, however, we impose the assumption that any point in X could be generated by a two-dimensional Gaussian distribution with mean vector μ and some covariance matrix $\mathbf{\Sigma}$ where μ must be one of 16 points in G. Accordingly, we compute the posterior probability

of X_i being sampled from $\mathcal{N}(G_k, \mathbf{\Sigma})$ as follows:

$$p(G_k|X_i) = \frac{\mathcal{N}(X_i; G_k, \mathbf{\Sigma})}{\sum_{k=1}^{k=16} \mathcal{N}(X_i; G_k, \mathbf{\Sigma})}$$

This results in a N-dimensional probability vector for G_k, that is

$$\mathbf{p}_{G_k} = [p(G_k|X_1), ..., p(G_k|X_N)]^\intercal$$

Finally, we define the connectivity strength between node i and j by

$$W_{ij} = \mathbf{p}_{G_i}^\intercal \mathbf{p}_{G_j}$$

To define two distinct classes of Ss and thus those of Ws, we introduce two distinct procedure to generate random pattern X. Instead of generating X uniformly over the unit square, we generate it uniformly over the upper-left corner of the unit square for class 1 and over the lower-right corner for class 2. Alternatively, we can generate X for class 1 by sampling N points from a two-dimensional Gaussian distribution with its mean vector located in the upper-left corner and that for class 2 with the mean vector located in the lower-right corner.

5.4 Numerical Results

For each test performed and reported in the sequel, we extracted sequentially three features based on the training data. Subsequently, the extracted data is used with a Generalized Matrix Learning Vector Quantization (GMLVQ) classifier is used as classification tool (explained in details in previous Chapter 4.3.2) to examine classification performance. The three extracted features are denoted by $v1$, $v2$, and $v3$ in the order how they were generated by the greedy procedure. With the extracted features, data classification proceeds first based on a single feature $v1$, on a two dimensional feature vector $V2 = (v1, v2)$ or based

on a three dimensional vector $V3 = (v1, v2, v3)$.

In this work, classification performance is measured by Macroaveraged Mean Absolute Error, which a macroaveraged version of Mean Absolute Error and it is a weighted sum of the classification errors across classes [37]. It measures the per-class accuracy of class predictions $\hat{y}$ with respect to true class y on a test set. In the case of two classes as we considered in this work $MMAE = \frac{1}{2}\left(\frac{\sum_{y_i=1}|y_i-\hat{y}_i|}{N_1} + \frac{\sum_{y_i=2}|y_i-\hat{y}_i|}{N_2}\right)$, where N_1 and N_2 is the number of test points in class 1 and 2, respectively.

5.4.1 Performance of the EGFE method on regular data

Figures 5.2 and 5.3 display those three that were features extracted from the order-2 and order-3 tensor data described above (respectively). The three features are denoted by $v1$, $v2$, and $v3$ in the order how they were generated in a greedy procedure. In each of these two figures, we display clouds of feature vectors $V2 = (v1, v2)$ for *Class 1* (in Blue) and *Class 2* (in Red) on the upper-left panel and those of $V3 = (v1, v2, v3)$ from three different view angles on the remaining panels. These figures visualise how the two classes are separated in the place of $(v1, v2)$ or $(v1, v2, v3)$. For the case of using single feature $v1$, separation of Class 1 and 2 can be visualised by projecting the data points in the upper-left panels onto the x-axis. For both order-2 and order-3 tensor data, it seems that class separation improves does improve by inclusion of additional discriminative features. This motivates us to check whether we can obtain the counterpart of this observation in classification performance, Generalized Matrix Learning Vector Quantization (GMLVQ) classifier is used as classification tool (explained in details in previous Chapter Section 4.3.2).

Table 5.1 summarizes the classification results obtained from our numerical experiment. Columns 2 and 3 in Table 5.1 display the mean miss-classification error and its standard deviation that were obtained from 50 independently generated tensor data sets. For both order-2 and order-3 tensor data, it is observed that the mean errors decrease with increasing number of the discriminative features. Next, we check whether this trend is

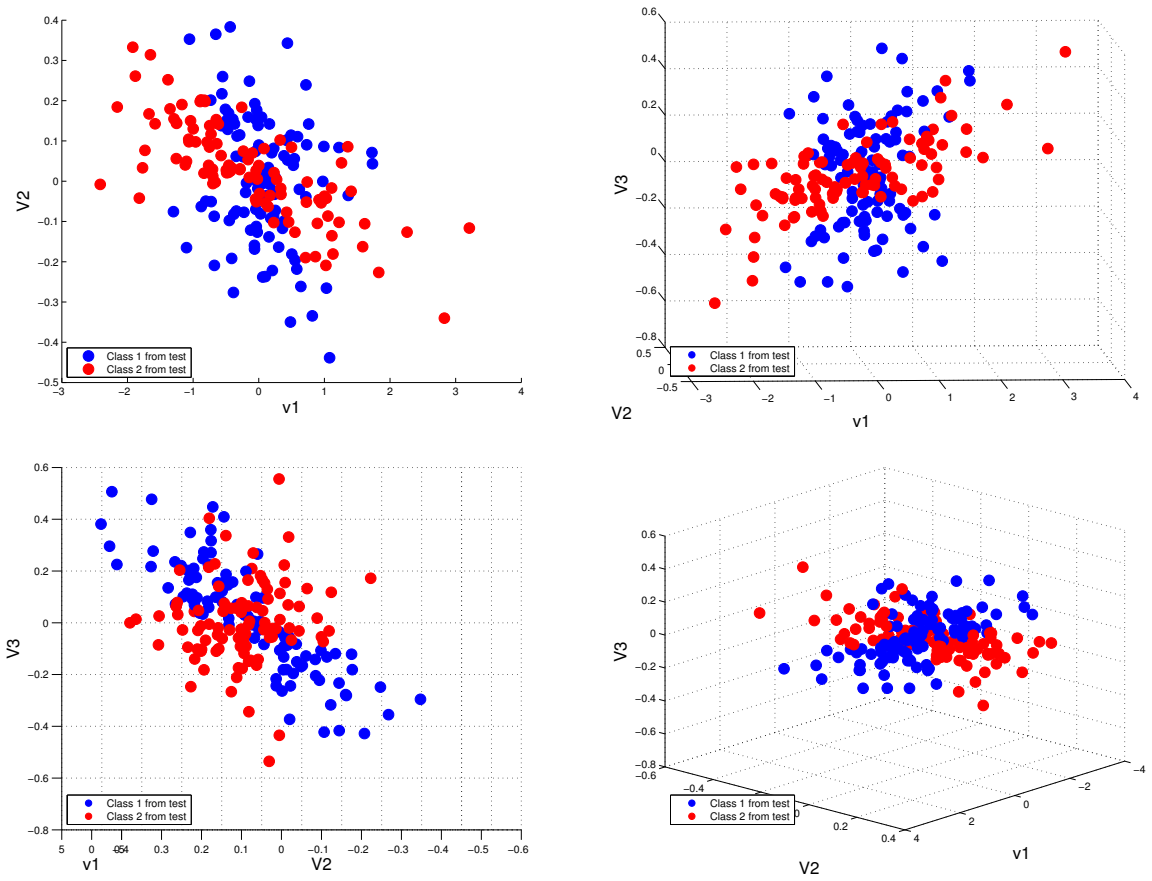

Figure 5.2: Two and three features extracted from order-2 tensors in data set D23, the upper-left panel is when we have only $v1$ and $V2$ of order-2 tensors. The upper-right panel is $v1$ axis of order-2 tensors, the lower-left panel is $V2$ axis of order-2 tensors and the lower-right panel is $V3$ axis of order-2 tensors.

statistically significant by testing two following hypotheses: (1) MMAE obtained from v_1 is greater that those from $V2$; (2) MMAE obtained from $V2$ is greater that those from $V3$. For this purpose, one-sided rank test is employed. Table 5.2 and 5.3 show that for both order-2 and order-3 data, the p-values are close to the commonly used threshold (that is, $p = 0.01$) and they decrease with inclusion of additional discriminative features.

In this numerical experiment, we generated synthetic tensors which are uniquely identified by two or three features (that is, the coefficients used for generating tensors by linear combination of fixed orthogonal rank-1 tensors). Based on these features, we further define two classes of those tensors in such way that the number of features can not be reduced by LDA without compromising classification performance. Our numerical

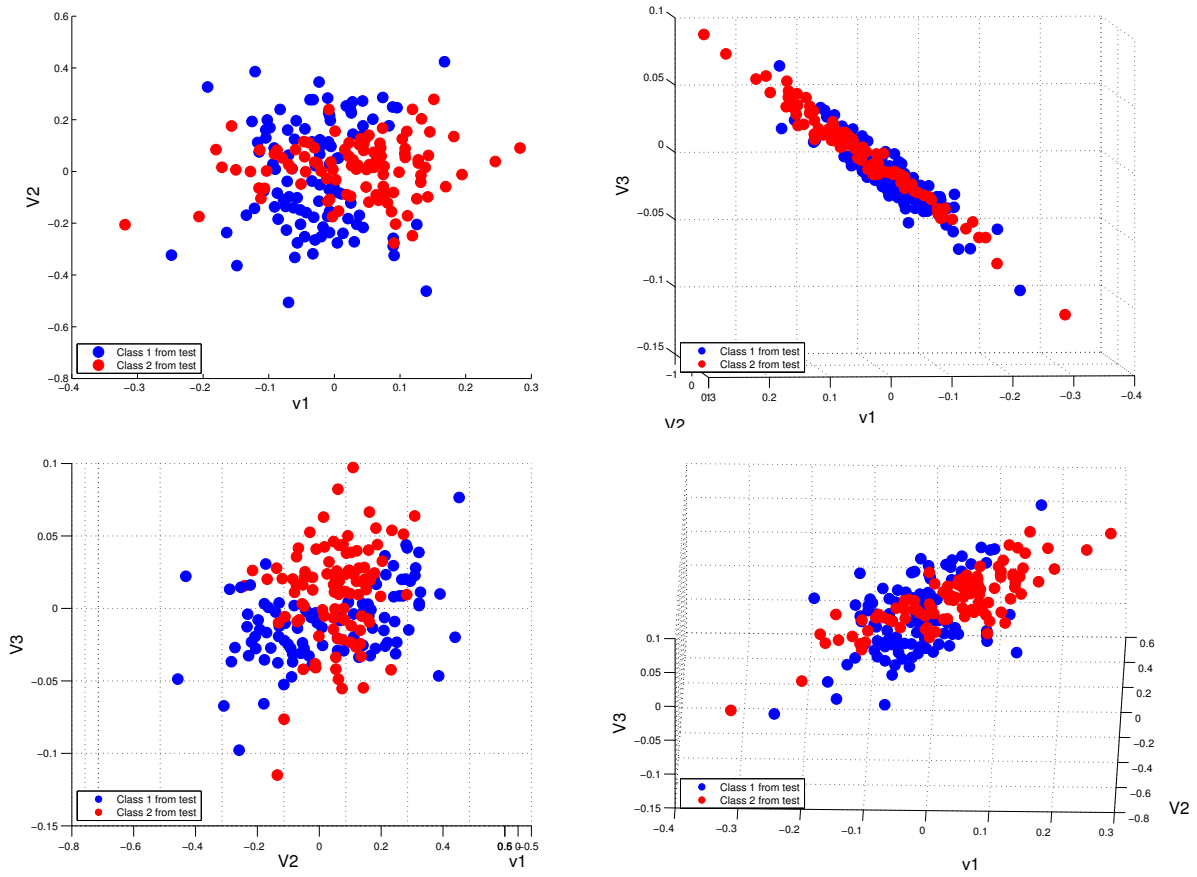

Figure 5.3: Two and three features extracted from order-3 tensors in data set D33, the upper-left panel is when we have only $v1$ and $V2$ of order-3 tensors. The upper-right panel is $v1$ axis of order-3 tensors, the lower-left panel is $V2$ axis of order-3 tensors and the lower-right panel is $V3$ axis of order-3 tensors.

experiments show that to achieve the best possible classification performance, we need (at least) three discriminatively extracted features. This is consistent with the setup of our experiment. Therefore, greedy feature extraction algorithm works in the way as we designed.

5.4.2 Performance of the EGFE method on data with overlapping

We examined how the performance of the EGFE method is degraded when the level of data overlapping increases, as specified by the parameter δ_{ov}. In all cases, the probability of the data in the overlapping region of being in each of the classes is equal i.e. $p = 0.5$.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D23	SOT_{v1}	0.23	$\pm$ 0.04	0.23	(0.21 , 0.27)
D23	SOT_{V2}	0.22	$\pm$ 0.03	0.22	(0.21 , 0.25)
D23	SOT_{V3}	0.21	$\pm$ 0.03	0.2	(0.19 , 0.22)
D33	TOT_{v1}	0.22	$\pm$ 0.02	0.21	(0.20 , 0.24)
D33	TOT_{V2}	0.20	$\pm$ 0.03	0.2	(0.20 , 0.23)
D33	TOT_{V3}	0.18	$\pm$ 0.02	0.18	(0.17 , 0.20)

Table 5.1: Macroaveraged Mean Absolute Error (MMAE) performance for extracting one, two or three data features from data sets D23 and D33 using the EGFE method based on the multiplicative cost criterion.

Models	p-value
$SOT_{v1} > SOT_{V2}$	0.14
$SOT_{v1} > SOT_{V3}$	0.06

Table 5.2: One side sign rank test of order-2 tensors multiplicative approach for the case of data set D23.

Models	p-value
$TOT_{v1} > TOT_{V2}$	0.08
$TOT_{v1} > TOT_{V3}$	0.06

Table 5.3: One side sign rank test of order-3 tensors of multiplicative approach for the case of data set D33.

The results are shown in Table 5.4 for $\delta_{ov} = 0.5$, in Table 5.5 for $\delta_{ov} = 0.6$ and in Table 5.6 for $\delta_{ov} = 0.8$. Comparing the data in these tables, it is obvious that the performance is becoming worse as the degree of overlapping increases, as it was expected. However for $\delta_{ov} = 0.5$ and 0.6, the performance increases as more features are extracted. In contrast, for $\delta_{ov} = 0.8$, the performance does not improve as more features are extracted, but actually degrades slightly. This means that for this degree of overlapping, the greedy feature extraction method stops working.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D23O	SOT_{v1}	0.26	± 0.03	0.27	(0.23 , 0.29)
D23O	SOT_{V2}	0.25	± 0.03	0.24	(0.22 , 0.30)
D23O	SOT_{V3}	0.23	± 0.03	0.23	(0.22 , 0.29)
D33O	TOT_{v1}	0.25	± 0.02	0.25	(0.22 , 0.29)
D33O	TOT_{V2}	0.23	± 0.02	0.22	(0.20 , 0.28)
D33O	TOT_{V3}	0.21	± 0.02	0.21	(0.20, 0.24)

Table 5.4: Macroaveraged Mean Absolute Error (MMAE) performance for extracting one, two or three data features from data sets with overlapping: $\delta_{ov} = 0.5$, $p = 0.5$.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D23O	SOT_{v1}	0.30	± 0.02	0.30	(0.28 , 0.33)
D23O	SOT_{V2}	0.28	± 0.02	0.29	(0.25 , 0.31)
D23O	SOT_{V3}	0.25	± 0.03	0.26	(0.22 , 0.30)
D33O	TOT_{v1}	0.30	± 0.02	0.29	(0.30 , 0.36)
D33O	TOT_{V2}	0.25	± 0.02	0.25	(0.24 , 0.29)
D33O	TOT_{V3}	0.23	± 0.03	0.24	(0.23, 0.27)

Table 5.5: MMAE performance for extracting one, two or three data features from data sets with overlapping: $\delta_{ov} = 0.6$, $p = 0.5$.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D23O	SOT_{v1}	0.41	± 0.02	0.42	(0.41 , 0.45)
D23O	SOT_{V2}	0.42	± 0.02	0.42	(0.42 , 0.45)
D23O	SOT_{V3}	0.42	± 0.04	0.43	(0.41 , 0.47)
D33O	TOT_{v1}	0.42	± 0.02	0.41	(0.40 , 0.43)
D33O	TOT_{V2}	0.39	± 0.03	0.39	(0.36 , 0.41)
D33O	TOT_{V3}	0.40	± 0.02	0.41	(0.36 , 0.42)

Table 5.6: MMAE performance for extracting one, two or three data features from data sets with overlapping: $\delta_{ov} = 0.8$, $p = 0.5$.

5.4.3 Performance of the EGFE method on data with outliers

Some peformance results for data with outliers are reported in Tables 5.7 and 5.8. It can be seen that the results are worse than those in Table 5.1 which means that the presence of outliers does effect the performance. However, increasing the number of features improves the classification performance.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D23R	SOT_{v1}	0.28	± 0.03	0.28	(0.24 , 0.32)
D23R	SOT_{V2}	0.27	± 0.02	0.26	(0.24 , 0.30)
D23R	SOT_{V3}	0.24	± 0.02	0.24	(0.20 , 0.26)
D33R	TOT_{v1}	0.26	± 0.03	0.25	(0.25, 0.34)
D33R	TOT_{V2}	0.24	± 0.03	0.24	(0.20, 0.25)
D33R	TOT_{V3}	0.23	± 0.02	0.22	(0.20, 0.26)

Table 5.7: MMAE performance for extracting one, two or three data features from data sets with outliers with $M_{outier} = 0.8$, $p_{outlier} = 0.04$.

Models	Mean	Std-Dev	Median	(25%, 75% percentile)	
D23R	SOT_{v1}	0.31	± 0.02	0.30	(0.30 , 0.35)
D23R	SOT_{V2}	0.27	± 0.02	0.27	(0.25 , 0.32)
D23R	SOT_{V3}	0.25	± 0.03	0.25	(0.23 , 0.31)
D33R	TOT_{v1}	0.30	± 0.02	0.30	(0.29, 0.35)
D33R	TOT_{V2}	0.26	± 0.02	0.26	(0.23 , 0.30)
D33R	TOT_{V3}	0.25	± 0.02	0.25	(0.20, 0.26)

Table 5.8: MMAE performance for extracting one, two or three data features from data sets with outliers with $M_{outier} = 0.8$, $p_{outlier} = 0.2$.

5.4.4 Performance of the EGFE method on data parameterized classification criterion

The test results on data set D22C is reported in Table 5.9. The corresponding results for the data set with overlapping D22CO are reported in Table 5.10. In the former case, extracting more than one features does improve performance. By contrast, in the latter case, the second and third features hardly improve performance even as the overlapping degree was smaller that that considered before.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D22C	SOT_{v1}	0.39	± 0.02	0.40	(0.39 , 0.41)
D22C	SOT_{V2}	0.36	± 0.02	0.36	(0.36 , 0.39)
D22C	SOT_{V3}	0.35	± 0.01	0.35	(0. 34, 0.37)

Table 5.9: MMAE performance for extracting one, two or three data features from data set D22C.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D22CO	SOT_{v1}	0.39	± 0.04	0.41	(0.37 , 0.43)
D22CO	SOT_{V2}	0.38	± 0.02	0.37	(0.36 , 0.41)
D22CO	SOT_{V3}	0.38	± 0.02	0.38	(0.35 , 0.4)

Table 5.10: MMAE performance for extracting one, two or three data features from data set D22CO, with $\delta_{ov} = 0.5$ $p = 0.5$.

5.4.5 Performance of the EGFE method on graph data model

We generated point pattern X for class 1 by sampling N points from a two-dimensional Gaussian distribution with its mean vector located in the upper-left corner and that for class 2 with the mean vector located in the lower-right corner.

The width of Gaussian distribution for class 1 was chosen to be 0.4, while it is 0.6 for class 2, to have some overlapping between the two classes, and the dimension of random graph set to $d = 16$. The test results on data graph data set Gr is reported in Table 5.11. By comparing $V2$ and $V3$ with $v1$, we can see that $V2$ does not improve the results, while $V3$ roughly improves it. The classification performance is measured by (25%, 75%) percentile of Macro-averaged Mean Absolute Error (MMAE), that is, (0.01, 0.03) for $v1$, but it yields a lower bound of classification error, which is (0.00, 0.01) for $V3$.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
Gr	SOT_{v1}	0.01	± 0.01	0.02	(0.01 , 0.03)
Gr	SOT_{V2}	0.03	± 0.03	0.01	(0.01 , 0.03)
Gr	SOT_{V3}	0.00	± 0.00	0.00	(0.00 , 0.01)

Table 5.11: MMAE performance for extracting one, two or three data features from data set Gr.

5.4.6 Comparison of the EGFE method with the ORO method

Also, we compare the classification performance between ORO [43] (explained in Chapter 2 Section 2.6) and our method in the case of the order-3 tensor data. Table 5.12 and 5.13 show that both methods achieve comparable results. For technical details of ORO method,

we refer to [43].

Recall that our method adopted a simple approach to conditioning successive greedy steps on the proceeding steps whereas this is achieved in ORO by constraining the orthogonality between the successive projecting vectors. The above experiments demonstrate that our simple method can achieve results which are comparable with those from ORO although it is conceptually and practically simpler.

Data set	Models	Mean	Std-Dev	Median	(25%, 75% percentile)
D33	$TOT_{v1} - ORO$	0.22	$\pm$ 0.02	0.22	(0.20 , 0.25)
D33	$TOT_{V2} - ORO$	0.20	$\pm$ 0.02	0.20	(0.19 , 0.23)
D33	$TOT_{V3} - ORO$	0.19	$\pm$ 0.02	0.19	(0.18 , 0.20)

Table 5.12: MMAE results of extracting synthetic data features, TOT for features ($v1$, $V2$ and $V3$) of ORO method in the case of data set D33.

Models	p-value
$TOT_{v1} - ORO > TOT_{v1}$	0.25
$TOT_{v1} - ORO < TOT_{v1}$	0.91
$TOT_{V2} - ORO > TOT_{V2}$	0.74
$TOT_{V2} - ORO < TOT_{V2}$	0.50
$TOT_{V3} - ORO > TOT_{V3}$	0.14
$TOT_{V3} - ORO < TOT_{V3}$	0.96

Table 5.13: One side sign rank test of order-3 tensors of classification errors between ORO approach and our multiplicative approach for the case of data set D33.

5.5 Conclusion

As real data was not available in time for testing our feature extraction method, we needed to construct synthetic data that can be conveniently used to test and validate the method and its software implementation. Simple algebraic criteria were used to split the data between the two classes and we have shown how separation between classes is improved by the extraction of more features. Results of tensors of order four and more were not included. Testing of the EGFE method on such data is easily done, but has not delivered qualitatively different results. After feature extraction, the reduced data was classified

using a GMVLQ classifier. The statistical performance tests have confirmed the intuitive picture provided by the graphical representation of the data. Further, the EGFE method was tested on synthetic data displaying realistic features such as overlapping and outliers classes. As the performance degraded as expected, the performance loss was moderate and proportional to the "failures" that were introduced in the data. Moreover, more realistic data were generated by random graph dataset, the results show that extracting the third features can improve the classification. Also, the performance of the EGFE method was compared with the performance of the ORO method, as an alternative greedy feature extraction method that was proposed in the literature. The results show very comparable performance, although our method is less complex since it does not require orthogonality constraints for the feature generating vector sets.

5.6 Chapter Summary

In order to test our method for feature extraction, we use primarily synthetic data. This chapter presents the process of creating this data for the case of second and third order tensors. A few numerical examples are worked out through the data extraction phase and the subsequent classification phase using a GMLVQ classifier. The performance of the EGFE method was compared with the performance of the ORO method, as an alternative greedy feature extraction method that was proposed in the literature. The classification performance is evaluated using statistical tests.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

Two contributions to advance machine learning were presented in this thesis. The first contribution is development of an efficient method for the greedy approach to tensor LDA. This is of particular relevance for classification of higher-order tensor data. The second contribution is the development of a diagnostic tool for early detection of dementia. This tool is of practical relevance as it can potentially boost the predictive performance through using costly privileged data in the training phase.

The method for feature extraction proposed in this thesis is referred as Efficient Greedy Feature Extraction (EGFE) and is a new development in Multilinear Discriminant Analysis as well as in discriminative feature extraction. All LDA methods are based on the idea of maximizing a Fisher-type criterion to obtain a reduced set of features. In contrast to other methods in the literature (see Chapter 2 for a detailed survey of such methods), our method extracts features sequentially and without unnecessary constraints. In a non-greedy LDA, all columns of the projection matrix need to be optimized jointly. In a greedy approach, however, this optimization task is reduced to a series of smaller ones that just optimizes the corresponding column of that projection matrix alone. Also, the greedy approach allows us to generate an additional feature only when it is needed. The Fisher optimization criterion can be either of multiplicative type, in which case, it is the

ratio of the interclass variance and the sum of intraclass variances, or it can be of additive type, in which case, it is the weighted difference between the two. The EGFE method can be applied in both cases and formulas were derived for the iterations of the optimization algorithm in each case.

The EGFE method is validated by numerical experiments using real data. For the early diagnosis of dementia disease, the techniques proposed in this thesis is a classifier equipped with "Learning with privileged information" component. The inputs to the base classifiers (that are, GLMVQ and SVM+ classifiers) is cost-effective cognitive scores while the discriminative features derived from expensive fMRI data were used as the privileged information. Note that the privileged information, which is fMRI feature in this work, is used only during the training of the classifier. The testing of the classifier is based on cognitive scores, as it is supposed to perform in practice. The fMRI features used include PSC (percent signal change), and SGF (spatially grouped graph feature) and FGF (functionally grouped graph feature). The input of the GMLVQ classifier consists are the cognitive scores is comprised of working memory, cognitive inhibition, divided attention and selective attention scores. The working of the algorithm is as follows: fMRI data is used to adapt the metric for the input data. Intuitively, if two cognitive test scores, which are the input of the classifier, appear to be "similar", but the corresponding fMRI data is different, the metric used to compare the input data is adapted such that the distance between the two test scores is increased. Alternatively, if two cognitive scores appear to be different, but the corresponding fMRI data is close, the input data metric is adapted such that the distance between the two test scores is decreased. In this way, the learning phase constructs an input metric tensor that effectively determines the most relevant cognitive test features. The input of the SVM+ classifier is the same as the GMLVQ classifier. The using of fMRI data is to estimate a slack variable model for the SVM+ classifier.

It is well understood that the data set that we had at our disposal is relatively small and therefore the reported results are only indicative of the potential of using PI in improving CD at the training stage. Of course, a larger data set would offer better validation of this

hypothesis. However, the reported results using p-values indicate statistically significant performance improvement in the case that PI was used in the training phase with respect to the baseline.

The results of the numerical experiments that we report in the thesis show that the use of fMRI feature data as privileged information can significantly improve the performance of the GMLVQ/SVM+ classifiers over the baseline classifiers that do not use privileged information for training.The numerical experiments show that GMLVQ has a slightly smaller misclassification error compared to SVM+, thus, we did the analysis for GMLVQ classifiers. We have conducted a "relevant feature analysis" for three different GMLVQ classifiers: the baseline GMLVQ classifier, the best fMRI-guided GMLVQ classifier, and the fMRI GMLVQ classifier. For the baseline classifier, "divided attention" is the only relevant cognitive feature for the classification task. When privileged information is incorporated, divided attention remains the most relevant feature while cognitive inhibition becomes also relevant. The above results suggest that attention, rather than only memory, plays an important role for the diagnosis task. More interestingly, the analysis of the fMRI GMLVQ classifier suggests three conclusion. First, among three ROIs used, the frontal ROI seems to be the most relevant for the classification task. Secondly, the PSC feature as an overall measure of fMRI response to structured stimuli is used as the inputs to the classifier, the post-training session seems to be the most relevant. Finally, when the graph feature reflecting underlying spatiotemporal fMRI pattern is used, the pre-training session seems to be the most relevant. Further analysis has indicated that training may cause an overall increase of the brain activity only for MCI patients while it may have "mitigated" the difference in brain connectivity pattern between MCI patients and healthy controls. Moreover, these training-dependent changes seem to be the most significant for a three-node sub-network in the frontal ROI. Taken together these results suggest that brain connectivity before training and overall fMRI signal after training are both relevant for the diagnostic of cognitive skills in MCI.

The EGFE method is also validated by numerical experiments using the synthetic

second- or third-order tensor data. For both higher-order cases, the synthetic data were generated by linear combination of three orthogonal rank-1 tensors. It is expected that for these synthetic data, the classification performance increases with the increasing number of the extracted features, indeed, our experiments have verified this conjunction for the third-order case. Furthermore, we compared our greedy method with the ORO method, a greedy tensor LDA method from the literature. The both methods differ in the way how they condition each iteration step on all steps proceeding it. Compared to ORO, they are comparable and our method did yield lower classification error with statistical significance. Furthermore, the method applied to more realistic higher-order tensor synthetic data (e.g. overlapping and failure mode cases), and the classification performance increases with the increasing the number of the extracted features. However, the missclassifications error is higher than the cases when pure synthetic data were used, that shows overlapping, outliers and failure modes were degraded as expected, comparing to the performance of the pure synthetic data.

The work reported in this part of the thesis uses machine learning algorithms to investigate the neurocognitive factors and their interactions that mediate learning ability in Mild Cognitive Impairment. However, it is is not limited to developing and validating machine learning approaches, but it also advances our understanding of the neurocognitive mechanisms that mediate learning in health and disease. For example, the role of cognitive inhibition in cognitive profile classification seems to be significantly enhanced when brain imaging information (obtained in a sequence learning prediction task) is provided as privileged information.

6.2 Future Work

One of the immediate directions for pursuing the work presented in this theses is to extend the EGFE method to the case of multiple classes. We have treated here only the case of binary classification. However, the Fisher type criteria for multiple classes, both the

multiplicative as the additive form, are well known in literature, and the EGFE method can be relatively easily extended to deal with those criteria in order to extract discriminating features for a multiple class classification task.

The most important and challenging directions for future investigation are in the area of applications, especially interdisciplinary applications. The development of the numerical tools for dealing with large and complex data has to be guided by the requirements from practical applications. Therefore, future work will address the application of the methods developed in this thesis to further complex data sets such as fourth-order tensor data, for example, depicting fMRI scan sequences in brain mapping research [99]. This is a 4D object with four modes: three spatial modes (column, row, and depth) and one temporal mode.

An interesting area of investigation both from theoretical and practical aspects is the extension of the data reduction EGFE method proposed in this thesis to nonlinear data analysis. As explained in Section 2.4, the kernel technique has been successfully used in nonlinear discriminant analysis before. However, this was only done for vector data and never for data organized as higher order tensors. Finding efficient ways to determine a nonlinear map in order to improve performance for the case of higher order tensor data remains a challenge for the future.

BIBLIOGRAPHY

[1] S. Albert, S. Dekosky, D. Dickson, B. Dubois, H. Feldman, N. Fox, and C. Phelps. The diagnosis of mild cognitive impairment due to Alzheimer's disease. *Alzheimers Dement*, 7(3):270–279, 2010.

[2] A. Arbib. *The Handbook of Brain Theory and Neural Networks*. MIT Press, 2nd edition, 2003.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[4] R. Baker, P. Bentham, and Z. Kourtzi. Learning to predict is spared in mild cognitive impairment due to alzheimer's disease. *Exp Brain Res*, 233(10):2859–2867, 2015.

[5] M. Barnathan, V. Megalooikonomou, C. Faloutsos, F. Mohamed, and S. Faro. TWave: High-order analysis of spatiotemporal data. *Advances in Knowledge Discovery and Data Mining*, pages 246–253, 2010.

[6] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

[7] C. Bauckhage, T. Käster, and J. K. Tsotsos. Higher order separable LDA using decomposed tensor classifiers.

[8] M. Belahcene, M. Laid, A. Chouchane, A. Ouamane, and S. Bourennane. Local descriptors and tensor local preserving projection in face recognition. In *Visual Information Processing (EUVIP), 2016 6th European Workshop on*, pages 1–6. IEEE, 2016.

[9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[10] J. C. Bezdek and R. J. Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pages 288–300. Springer, 2002.

[11] C. M. Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[12] C. J. Burges et al. Dimension reduction: A guided tour. *Foundations and Trends® in Machine Learning*, 2(4):275–365, 2010.

[13] C. Carpineto and G. Romano. Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval. *IEEE Trans Pattern Anal*, 34(12):15–26, 2012.

[14] E. Challis, P. Hurley, L. Serra, M. Bozzali, S. Oliver, and M. Cercignani. Gaussian process classification of Alzheimer's disease and mild cognitive impairment from resting-state fMRI. *NeuroImage*, 112:232–243, 2015.

[15] R. Chellappa, A. K. Roy-Chowdhury, and S. K. Zhou. Recognition of humans and their activities using video. *Synthesis Lectures on Image, Video & Multimedia Processing*, 1(1):1–173, 2005.

[16] B. Chen, M. Liu, D. Zhang, and D. Shen. Domain transfer learning for mci conversion prediction. *IEEE Transaction on Biomedical Engineering*, 62:232–243, 2015.

[17] H.-T. Chen, T.-L. Liu, and C.-S. Fuh. Learning effective image metrics from few pairwise examples. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1371–1378. IEEE, 2005.

[18] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

[19] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[20] I. Davidson, S. Gilpin, O. Carmichael, and P. Walker. Network discovery via constrained tensor analysis of fMRI data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–202. ACM, 2013.

[21] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. *in Proceedings of the 24th International Conference on Machine Learning, ser. ICML 07. New York, NY, USA: ACM*, pages 209–216, 2007.

[22] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[23] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[24] A. Diaf, B. Boufama, and R. Benlamri. Non-parametric fishers discriminant analysis with kernels for data classification. *Pattern recognition letters*, 34(5):552–558, 2013.

[25] B. Dickerson, D. Salat, D. Greve, E. Chua, E. Rand-Giovannetti, D. Rentz, L. Bertram, K. Mullin, R. Tanzi, D. Blacker, et al. Increased hippocampal activation in mild cognitive impairment compared to normal aging and ad. *Neurology*, 65(3):404–411, 2005.

[26] J. Duchene and S. Leclercq. An optimal transformation for discriminant and principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):978–983, 1988.

[27] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. 2nd. *Edition. New York*, page 55, 2001.

[28] A. dAspremont, F. Bach, and L. E. Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294, 2008.

[29] S. A. Elrahman and A. Abraham. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1(ISSN 2160-2174):332–340, 2013.

[30] C. Faloutsos, T. G. Kolda, and J. Sun. Mining large time-evolving data using matrix and tensor tools. In *ICDM Conference*, volume 565, 2007.

[31] Y. Fan, D. Shen, and C. Davatzikos. Detecting cognitive states from fMRI images by machine learning and multivariate classification. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 89–89. IEEE, 2006.

[32] L. Farràs-Permanyer, J. Guàrdia-Olmos, and M. Peró-Cebollero. Mild cognitive impairment and fmri studies of brain functional connectivity: the state of the art. *Frontiers in psychology*, 6:1095, 2015.

[33] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[34] R. A. Fisher. The precision of discriminant functions. *Annals of Human Genetics*, 10(1):422–429, 1940.

[35] D. H. Foley and J. W. Sammon. An optimal set of discriminant vectors. *IEEE Transactions on Computers*, 100(3):281–289, 1975.

[36] E. Formisano, F. De Martino, and G. Valente. Multivariate analysis of fMRI time series: classification and regression of brain responses using machine learning. *Magnetic resonance imaging*, 26(7):921–934, 2008.

[37] S. Fouad. *Metric Learning for Incorporating Privileged Information in Prototype-based Models*. Thesis of the degree of doctor of philosophy, School of Computer Science, 2013.

[38] S. Fouad, P. Tino, S. Raychaudhury, and P. Schneider. Incorporating privileged information through metric learning. *IEEE Transactions on Neural Networks and Learning System*, 24(7):1086–1098, 2013.

[39] V. Garcia, J. Sanchez, R. Mollineda, R. Alejo, and J. Sotoca. The class imbalance problem in patteren classification and learning. 2007.

[40] R. D. Green and L. Guan. Quantifying and recognizing human movement patterns from monocular video images-part ii: applications to biometrics. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(2):191–198, 2004.

[41] H. Higashi, T. M. Rutkowski, T. Tanaka, and Y. Tanaka. Subspace-constrained multilinear discriminant analysis for ERP-based brain computer interface classification. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*, pages 934–940. IEEE, 2015.

[42] D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre, et al. Big data: The future of biocuration. *Nature*, 455(7209):47–50, 2008.

[43] G. Hua, P. A. Viola, and S. M. Drucker. Face recognition using discriminatively trained orthogonal rank one tensor projections. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[44] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[45] Z. Jin, J.-Y. Yang, Z.-S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern recognition*, 34(7):1405–1416, 2001.

[46] I. T. Jolliffe. Springer series in statistics. *Principal component analysis*, 29, 2002.

[47] V. G. Kanas, E. I. Zacharaki, E. Pippa, V. Tsirka, M. Koutroumanidis, and V. Megalooikonomou. Classification of epileptic and non-epileptic events using tensor decomposition. In *Bioinformatics and Bioengineering (BIBE), 2015 IEEE 15th International Conference on*, pages 1–5. IEEE, 2015.

[48] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2012.

[49] R. Khanna, E. Elenberg, A. G. Dimakis, and S. Negahban. On approximation guarantees for greedy low rank optimization. *arXiv preprint arXiv:1703.02721*, 2017.

[50] R. Khanna, E. Elenberg, A. G. Dimakis, S. Negahban, and J. Ghosh. Scalable greedy feature selection via weak submodularity. *arXiv preprint arXiv:1703.02723*, 2017.

[51] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[52] W. Krzanowski, P. Jonathan, W. McCarthy, and M. Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Applied statistics*, pages 101–115, 1995.

[53] M. H. Law and A. K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE transactions on pattern analysis and machine intelligence*, 28(3):377–391, 2006.

[54] C. Lee and D. A. Landgrebe. Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):388–400, 1993.

[55] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller. Introduction to machine learning for brain imaging. *Neuroimage*, 56(2):387–399, 2011.

[56] A. Lenhart, K. Purcell, A. Smith, and K. Zickuhr. Social media & mobile internet use among teens and young adults. millennials. *Pew internet & American life project*, 2010.

[57] J.-B. Li, S.-C. Chu, and J.-S. Pan. Kernel discriminant analysis based face recognition. In *Kernel Learning Algorithms for Face Recognition*, pages 101–133. Springer, 2014.

[58] M. Li and B. Yuan. 2D-LDA: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[59] Q. Li and D. Schonfeld. Multilinear discriminant analysis for higher-order tensor data classification. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2524–2537, 2014.

[60] C. Liu and H. Wechsler. Enhanced Fisher linear discriminant models for face recognition. In *Proceedings. Fourteenth International Conference on Pattern Recognition.*, volume 2, pages 1368–1372. IEEE, 1998.

[61] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[62] M. Loog, R. P. W. Duin, and R. Haeb-Umbach. Multiclass linear dimension reduction by weighted pairwise Fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[63] H. Lu, K. N. Plataniotis, and A. Venetsanopoulos. *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC press, 2013.

[64] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19(1):18–39, 2008.

[65] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition. *IEEE Transactions on Neural Networks*, 20(1):103–123, 2009.

[66] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[67] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, 14(1):117–126, 2003.

[68] C. Luft, R. Baker, P. Bentham, and Z. Kourtzi. Learning temporal statistics for sensory predictions in mild cognitive impairment. *Neuropsychologia*, 75:368–380, 2016.

[69] C. Luft, R. Baker, A. Goldstone, Y. Zhang, and Z. Kourtzi. Learning temporal statistics for sensory predictions in aging. *Journal of Cognitive Neuroscience*, 28(3):1–15, 2015.

[70] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.*, pages 41–48. Ieee, 1999.

[71] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse pca: Exact and greedy algorithms. In *Advances in neural information processing systems*, pages 915–922, 2006.

[72] S. Mueller, D. Keeser, M. Reiser, S. Teipel, and T. Meindl. Functional and structural mr imaging in neuropsychiatric disorders, part 1: imaging techniques and their application in mild cognitive impairment and alzheimer disease. *American Journal of Neuroradiology*, 33(10):1845–1850, 2012.

[73] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[74] B. Ng, A. Vahdat, G. Hamarneh, and R. Abugharbieh. Generalized sparse classifiers for decoding cognitive states in fMRI. In *International Workshop on Machine Learning in Medical Imaging*, pages 108–115. Springer, 2010.

[75] T. J. O'Neill. Error rates of non-Bayes classification rules and the robustness of Fisher's linear discriminant function. *Biometrika*, 79(1):177–184, 1992.

[76] Y. Peng, P. Zhou, H. Zheng, B. Zhang, and W. Yang. Multilinear local Fisher discriminant analysis for face recognition. In *Chinese Conference on Biometric Recognition*, pages 130–138. Springer, 2016.

[77] F. Pereira, T. Mitchell, and M. Botvinick. Machine learning classifiers and fMRI: a tutorial overview. *Neuroimage*, 45(1):S199–S209, 2009.

[78] M. Perez-Ortiz, P. Gutierrez, P. Tino, and C. Hervas-Martinez. Over-sampling the minority class in the feature space. *IEEE Transaction on Neural Networksand Learning System*, 2015.

[79] A. H. Phan and A. Cichocki. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear theory and its applications, IEICE*, 1(1):37–68, 2010.

[80] E. Pippa, V. G. Kanas, E. I. Zacharaki, V. Tsirka, M. Koutroumanidis, and V. Megalooikonomou. EEG-based classification of epileptic and non-epileptic events using multi-array decomposition. *International Journal of Monitoring and Surveillance Technologies Research*, 2017.

[81] N. Renard and S. Bourennane. Dimensionality reduction based on tensor modeling for classification methods. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1123–1131, 2009.

[82] H. S. Sahambi and K. Khorasani. A neural-network appearance-based 3-D object recognition using independent component analysis. *IEEE transactions on neural networks*, 14(1):138–149, 2003.

[83] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, 25(1):65–77, 1992.

[84] J. Sato, C. Thomaz, E. Cardoso, A. Fujita, M. Martin, and E. Amaro. Hyperplane navigation: A method to set individual scores in fmri group datasets. *Neuroimage*, 42:1473–1480, 2008.

[85] P. Schneider. *Advanced methods for prototype-based classification*. PhD Dissertation, University of Groningen, 2010.

[86] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, Dec 2009.

[87] T. K. M. R. Schroeder and T. Huang. Self-organizing maps. 3rd edn. Springer-Verlag New York, Inc, 2001.

[88] A. Shashua and A. Levin. Linear image coding for regression and classification using the tensor-rank principle. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[89] J. Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[90] A. Sierra. High-order Fisher's discriminant analysis. *Pattern Recognition*, 35(6):1291–1302, 2002.

[91] F. Song and H. Li. Discriminant face images taking both the feature correlation and Fisher criterion into account. In *Sixth International Conference on Natural Computation (ICNC)*, volume 6, pages 3305–3308. IEEE, 2010.

[92] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.

[93] N. Tan, L. Huang, and C. Liu. Face recognition based on lbp and orthogonal rank-one tensor projections. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[94] D. Tao, X. Li, X. Wu, and S. Maybank. Tensor rank one discriminant analysis convergent method for discriminative multilinear subspace selection. *Neurocomputing*, 71(10):1866–1882, 2008.

[95] D. Tao, X. Li, X. Wu, and S. J. Maybank. General tensor discriminant analysis and Gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10), 2007.

[96] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 4th edition, 2009.

[97] M. Thomas, C. Kambhamettu, and S. Kumar. Face recognition using a color subspace LDA approach. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 1, pages 231–235. IEEE, 2008.

[98] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[99] V. G. van de Ven, E. Formisano, D. Prvulovic, C. H. Roeder, and D. E. Linden. Functional connectivity as revealed by spatial independent component analysis of fmri measurements during rest. *Human brain mapping*, 22(3):165–178, 2004.

[100] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

[101] L. Wang, X. Wang, and J. Feng. On image matrix based feature extraction algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1):194–197, 2006.

[102] S.-J. Wang, J. Yang, N. Zhang, and C.-G. Zhou. Tensor discriminant color space for face recognition. *IEEE Transactions on Image Processing*, 20(9):2490–2501, 2011.

[103] Y. Wang and Q. Wu. Sparse pca by iterative elimination algorithm. *Advances in computational mathematics*, 36(1):137–151, 2012.

[104] D. M. Witten and R. Tibshirani. Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011.

[105] F. Wu, X.-Y. Jing, X. Dong, Q. Ge, S. Wu, Q. Liu, D. Yue, and J.-Y. Yang. Uncorrelated multi-set feature learning for color face recognition. *Pattern Recognition*, 60:630–646, 2016.

[106] S. Wu, W. Li, Z. Wei, and J. Yang. Local discriminative orthogonal rank-one tensor projection for image feature extraction. In *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pages 367–371. IEEE, 2011.

[107] D. Xu, S. Lin, S. Yan, and X. Tang. Rank-one projections with adaptive margins for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1226–1236, 2007.

[108] D. Xu, S. Yan, D. Tao, L. Zhang, X. Li, and H.-J. Zhang. Human gait recognition with matrix representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7):896–903, 2006.

[109] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang. Discriminant analysis with tensor representation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 526–532. IEEE, 2005.

[110] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*, 16(1):212–220, 2007.

[111] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. *Advances in Neural Information Processing Systems*, 17:1569–1576, 2004.

[112] J. Ye, R. Janardan, Q. Li, et al. Two-dimensional linear discriminant analysis. In *NIPS*, volume 4, page 4, 2004.

[113] J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(4):181–190, 2004.

[114] W.-S. Zheng, J.-H. Lai, and S. Z. Li. 1D-LDA vs. 2D-LDA: When is vector-based linear discriminant analysis better than matrix-based? *Pattern Recognition*, 41(7):2156–2172, 2008.

[115] Y. Zhou, L. Bao, and Y. Lin. Fast second-order orthogonal tensor subspace analysis for face recognition. *Journal of Applied Mathematics*, 2014, 2014.

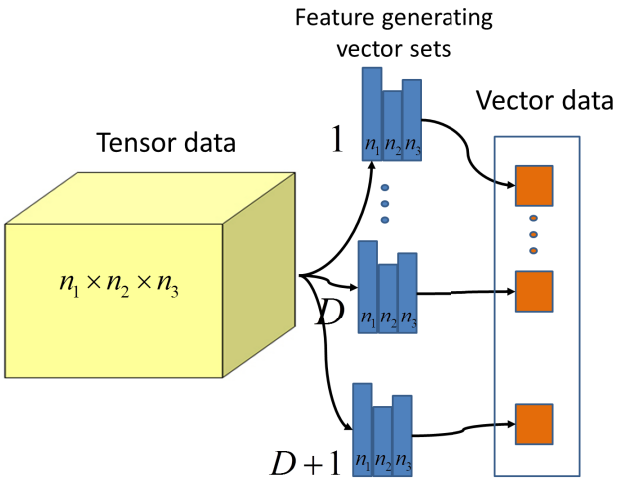

Figure 3.2: Reduction of tensor data to vector data using the feature generating vector sets that were determined by the proposed supervised learning process.

the classes is measured by the total squares variation. In order to reduce this simultaneous maximize-minimize problem to a single optimization problem, we use two conventional approaches . The first is the multiplicative approach that attempts at maximizing the ratio of the two quantities. The second is the additive approach that attempts to maximizing a weighted difference between the two quantities. In each case, maximization will tend to maximize the numerator, respectively the positive term, and minimize the denominator, respectively, the negative term. We will choose the one that works better.

In both cases, the data reduction algorithm reduces to a succession of optimization problems that can be solved in principle by any numerical algorithm. In this chapter, we clarify the implementation of these algorithms using gradient ascent, and therefore it is essential to derive the expression of the gradients of the optimization criterion with respect to the elements of the feature generating vectors. More details about the implementation of the optimization algorithms are presented in Section 3.5.

3.2 Multiplicative Criterion Case

As explained before, the training process has two parts. The first part consists of determining the first set of feature generating vectors and solves the following optimization

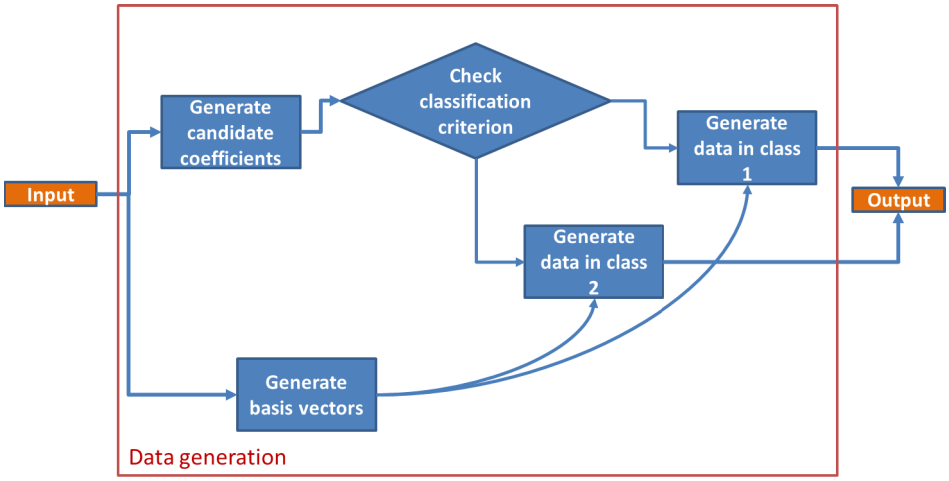

Figure 5.1: Pipeline scheme for the creation of the synthetic data.

constructed and we evaluate the performance of our methods using this data set. Additionally, we compare our method with the ORO method [43] that is also greedy tensor LDA algorithm like the EGFE method that we proposed by applying both methods on the same third order tensor data.

5.2 Synthetic Data Construction

Figure 5.1 depicts the pipeline of the data creation process. It illustrates in details how we synthetically generate 2nd and 3rd order tensors datasets. The input required for this process consists of the order of tensors, the dimensions of individual modes, and the number of data samples.

Essentially, the data construction proceeds by selecting at random a set of rank-1 tensors, parameterized by a number of randomly generated parameter values, and using some criterion to separate the generated tensors into two classes. The criterion is a nonlinear condition on the parameter values. For all the data that was used in our experiments, we set the dimension of all modes to $d = 6$. Notice that although the data has very high dimension ($d^2 = 36$ for order two tensors or $d^3 = 216$ for order three tensors), it is parameterized by a small number of parameters. The values of these parameters are used to divide the data into classes using some nonlinear criterion. Therefore, these