Localized Feature Selection for Classification

LOCALIZED FEATURE SELECTION FOR CLASSIFICATION

BY

NARGES ARMANFARD, M.Sc. (Electrical and Computer Engineering),

Tarbiat Modares University, Tehran, Iran

A THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2016) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Localized Feature Selection for Classification
AUTHOR:	Narges Armanfard M.Sc. (Electrical and Computer Engineering), Tarbiat Modares University, Tehran, Iran
SUPERVISOR:	Dr. James P. Reilly
NUMBER OF PAGES:	xvii, 140

Lay Abstract

This study proposes a novel form of pattern classification method, which is formulated in a way so that it is easily executable on a computer. Two different versions of the method are developed. These are the LFS (localized feature selection) and lLFS (logistic LFS) methods. Both versions are appropriate for analysis of data with complex distributions, such as datasets that occur in biological signal processing problems. We have shown that the performance of the proposed methods is significantly improved over that of previous methods, on the datasets that were considered in this thesis.

The proposed method is applied to the specific problem of determining the prognosis of a coma patient. The viability of the formulation and the effectiveness of the proposed algorithm are demonstrated on several synthetic and real world datasets, including comatose subjects.

Abstract

The main idea of this thesis is to present the novel concept of localized feature selection (LFS) for data classification and its application for coma outcome prediction.

Typical feature selection methods choose an optimal global feature subset that is applied over all regions of the sample space. In contrast, in this study we propose a novel localized feature selection approach whereby each region of the sample space is associated with its own distinct optimized feature set, which may vary both in membership and size across the sample space. This allows the feature set to optimally adapt to local variations in the sample space. An associated localized classification method is also proposed.

The proposed LFS method selects a feature subset such that, within a localized region, within-class and between-class distances are respectively minimized and maximized. We first determine the localized region using an iterative procedure based on the distances in the original feature space. This results in a linear programming optimization problem. Then, the second method is formulated as a non-linear joint convex/increasing quasi-convex optimization problem where a logistic function is applied to focus the optimization process on the localized region within the unknown co-ordinate system. This results in a more accurate classification performance at

the expense of some sacrifice in computational time. Experimental results on synthetic and real-world data sets demonstrate the effectiveness of the proposed localized approach.

Using the LFS idea, we propose a practical machine learning approach for automatic and continuous assessment of event related potentials for detecting the presence of the mismatch negativity component, whose existence has a high correlation with coma awakening. This process enables us to determine prognosis of a coma patient. Experimental results on normal and comatose subjects demonstrate the effectiveness of the proposed method.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Dr. James P. Reilly, for his continuous support of my PhD research. His patience, motivation, enthusiasm and knowledge have guided and nurtured me during my PhD. Thank you for being a supportive advisor and wonderful mentor, and for giving me the freedom and encouragement to pursue my ideas. I hope to emulate your passion for your work in whatever field I find myself in next on this journey.

Next, I would like to thank Dr. John Connolly, for much needed guidance in the field of coma outcome prediction.

Thank you to my committee, Dr. Tim Davidson, Dr. Shahram Shirani and Dr. Gary Hasey for your insights, critiques and encouragement. Our meetings have always been incredibly useful and helped me to focus and articulate my ideas and map out the direction in which I wanted to take my project. Thank you for your time.

I would like to thank the administrative team at the Department of Electrical and Computer Engineering, McMaster University. I wish to express my appreciation to Cheryl Gies for her constant support.

I also would like to express my gratitude to my friends and colleagues at McMaster University who helped me go through my PhD study.

To my parents, M. Ali and Razieh, thank you so much for your unending support

and encouragement through all the years, without which all my achievements have simply been impossible. Your constant support and encouragement have propelled me forward. You are wonderful and inspiring examples of how to balance career and family, achieving your professional goals while continuing to be truly superb parents. Thank you for always believing in me and pushing me to try harder. And here goes my thanks to my siblings who are all so much part of me and my memories of childhood. Their love and support still sustain me today.

Last but not least, I owe my deepest thanks to my loving husband, Majid, for his unconditional love, brilliant ideas, patience, sacrifices, and encouragements. I appreciate all his help and support through my research.

Notation and abbreviations

ML	Machine Learning
LFS	Local(ized) Feature Selection
lLFS	logistic Localized Feature Selection
VC	Vapnik Chervonenkis
SVM	Support Vector Machine
RBF	Radial Basis Function
RoL	Region of Locality
MMN	Mismatch Negativity
EEG	Electroencephalogram
GCS	Glasgow Coma Scale
ERP	Event Related Potential
std	Standard
dev	Deviant
TPR	True Positive Rate
TNR	True Negative Rate
LOO	Leave-One subject-Out
IEEE	Institute of Electrical and Electronics Engineers

Contents

List of Figures

Declaration of Academic Achievement

This research presents analytical and computational work carried out solely by Narges Armanfard, herein referred to as "the author," with advice and guidance provided by the academic supervisor Prof. James P. Reilly and with guidance and advice provided by Prof. John F. Connolly. Information that is presented from outside sources which has been used towards analysis or discussion has been cited when appropriate. All other materials are the sole work of the author.

Chapter 1

Introduction

Dimensionality reduction is a very important component in data classification applications. It is an antidote to what Bellman referred to as the "curse of dimensionality" (Bellman and Dreyfus, 1962). It is well known that the performance of typical classifiers notably drops when the number of available objects is not adequate in comparison to the number of candidate features (Weston *et al.*, 2000). A typical approach to addressing this problem is to apply some form of dimensionality reduction to the candidate feature set before the classification process. Dimensionality reduction plays an important role in big data problems, such as e.g., in the medical field, where oligonucleotide microarray data is used for identification of cancer-associated gene expression profiles of prognostic or diagnostic value (Van't Veer *et al.*, 2002; Wang *et al.*, 2005; Sun *et al.*, 2010). In this case the number of available samples is less than a hundred while the raw data are characterized by thousands of features. Among this large gene set, only a small subset of these features is relevant to the determination of cancerous tumor spread or/and growth. Thus some form of dimensionality reduction technique is required to identify this small subset of relevant features.

Dimensionality reduction approaches can be classified into two categories. The first is feature extraction (Webb, 2003; Jolliffe, 2005; Roweis and Saul, 2000; Oveisi *et al.*, 2012) which is also called subspace learning. The second category is feature selection (Peng *et al.*, 2005; Wei and Billings, 2007; Wang, 2008; Zeng and Cheung, 2011; Kwak and Choi, 2002; Chakraborty and Pal, 2015). Feature extraction approaches, like PCA (Jolliffe, 2005), LDA (Duda *et al.*, 2001) and ICA (Hyvärinen and Oja, 2000), perform dimensionality reduction through combining original features to find a new set of features. Typically, extracted features lose their physical interpretation in terms of the original features. Feature selection approaches perform dimensionality reduction, with no transformation, by selecting a subset of the original features. Hence, feature selection approaches retain the physical interpretability property in terms of the selected features. In this study we consider the feature selection aspect of the dimensionality reduction problem.

Traditionally, feature selection approaches are categorized into wrapper and filter approaches. Wrapper approaches evaluate a feature subset based on the accuracy of a specific classifier on a specific data set. Filter methods evaluate a feature subset based on its information content instead of optimizing the performance of any specific classifier. The interested reader may refer to (Gui *et al.*, 2016; Kohavi and John, 1997; Sánchez-Maroño *et al.*, 2007) for more details.

Feature selection algorithms can also be categorized into batch methods and online algorithms. In the former the feature selection task is conducted in an offline phase where all features of training instances are given while the online feature selection algorithms assume that the full feature space is unknown in advance. The online methods are appropriate for the applications where the training samples or features

arrive in a sequential manner (Wu *et al.*, 2013; Wang *et al.*, 2014; Yu *et al.*, 2014). In this study the batch algorithms are considered and discussed.

From another point of view, conventional feature selection algorithms assume that all regions of sample space can be optimally characterized by a common subset of features (Wang, 2008; Peng *et al.*, 2005; Guyon and Elisseeff, 2003; Liu and Motoda, 2007; Brown *et al.*, 2012; Khushaba *et al.*, 2011). These approaches can be roughly categorized into two major groups. The first group includes approaches that select a common feature subset with no consideration of the local behavior of the samples over the sample space. For example, in (Peng *et al.*, 2005), a common subset of features is selected using a mutual information based approach that utilizes a minimal-redundancy maximal-relevance criterion. In (Zhu *et al.*, 2007a), a common feature set is computed based on a genetic algorithm (GA) where the GA solutions are fine tuned based on a Markov blanket algorithm; the embedded Markov blanket-based memetic operators add or delete features from a GA solution. (Aliferis *et al.*, 2010) presents an algorithm to learn local causal structure around a target variable of interest by focusing on both identification of variables that are direct causes or direct effects of the target and discovery of Markov blankets. In (Wang, 2008), a common discriminative feature subset is obtained by maximizing a class separability criterion. In (Khushaba *et al.*, 2011), a differential-evolution based algorithm is used for computing a common feature set. The Fisher criterion is used in (Duda *et al.*, 2001) where each feature score is computed based on minimizing intra–class distances and maximizing inter–class distances. In (Cheng *et al.*, 2011), a common feature set is selected based in spirit on Fisher's discriminant analysis, where in defining the class separability, it incorporates the kernel trick to map each original input to a higher dimensional

kernel space. In (Tao *et al.*, 2015), a common set is computed through a combination of linear discriminant analysis and sparsity regularization. In (Ramona *et al.*, 2012) a feature subset is determined based on two criteria designed for the optimization of the SVM, including kernel target alignment and kernel class separability. In (Xiang *et al.*, 2012) a common feature subset is computed through expanding a nonconvex paradigm into a sparse group feature selection process. The selection algorithm *elastic net*, presented in (Zou and Hastie, 2005), combines the algorithmic ideas of Least Angle Regression (LARS) (Efron *et al.*, 2004), the computational benefits of ridge regression and the tendency towards sparse solutions of the LASSO. In (Sun *et al.*, 2014) a feature selection method, for microarray data classification, is presented that is based on partial least squares and theory of Reproducing Kernel Hilbert Space (Shawe-Taylor and Cristianini, 2004).

The second group applies local information of the sample space for computing an optimal feature subset (Kira and Rendell, 1992; Kononenko, 1994; Gilad-Bachrach *et al.*, 2004; Sun, 2007; Chen *et al.*, 2009; Sun *et al.*, 2010; Liu *et al.*, 2013). For example, Bi-Clustering approaches (Madeira and Oliveira, 2004; Cheng and Church, 2000; Dhillon, 2001) use local information for simultaneously clustering data and features. In (Li *et al.*, 2008), data clustering is realized through a greedy feature selection algorithm which can assign a specified feature set to each cluster. However, these algorithm s are unsupervised feature selection approaches. The approaches more relevant in the present case are "margin" based algorithms that are supervised and embed local information. These methods select features based on maximizing "margin", where "margin" of a sample is defined as the difference between the distance to the nearest differently labeled sample and the distance to the nearest same labeled

sample. For example, in the sample-based RELIEF algorithm (Kira and Rendell, 1992), feature weights are iteratively updated according to the margin of a randomly selected sample at the current iteration. The main drawback of RELIEF is that the neighboring samples are predefined in the original feature space, which yields degraded margin estimates in the presence of irrelevant features. The Simba algorithm (Gilad-Bachrach *et al.*, 2004) is an enhancement of the RELIEF algorithm in that during the learning process, margins are reevaluated based on the learned feature vector. The main drawback of Simba is that its objective function is non-convex and hence is characterized by the presence of local minima. In (Sun *et al.*, 2010), a local learning–based feature selection method is presented in which a complex non-linear problem is decomposed into a set of locally linear problems. In (Liu *et al.*, 2011) local information is embedded in feature selection through combining instance-based and model-based learning methods. However, the main disadvantage of this second group of algorithms is that they still generate a common feature set for the whole sample space.

Thus we see that current feature selection schemes impose a global set of features that are common across the entire sample space. Such schemes are inherently restricted in their ability to adapt to statistical variations (i.e., non–stationarities), across the sample space. These variations could be the result of a change in operating conditions of the underlying generative process. In this study, we introduce an alternative view to the traditional concept of a common feature set. We introduce what we believe is the novel concept of *localized* feature selection. The concept of localized feature selection is implemented by considering each sample of the training

set as a representative point for its neighboring region. A unique (and possibly distinct) feature subset is selected for every such region, based on an optimality criterion that encourages local clustering over that region. Because the selected feature subset varies over the sample space, conventional classifiers are no longer appropriate for the proposed algorithm. We therefore present a localized classification procedure that has been adapted to the proposed scenario.

The proposed approach has several advantages. First, it accommodates non–stationarities in the underlying data distribution, because no assumptions are made about the distribution of data over the sample space. Therefore, it allows irregular and/or disjoint distributions of samples. The proposed approach is also effective when the sample space lies on a non-linear manifold, since an optimal feature subset can be selected to fit the local behavior in each region of the manifold. Second, the proposed method may be less sensitive to overfitting relative to other methods. The overfitting phenomenon may be considered from two perspectives: feature selection and classification. With regard to feature selection, with alternative methods such as (Peng *et al.*, 2005; Khushaba *et al.*, 2011; Zhu *et al.*, 2007a), the number of selected features is determined in advance by a user–defined parameter. The value of this parameter is often difficult to determine and if this parameter is set too high, features may be selected whether they are relevant or not, a fact which introduces vulnerability to overfitting. In contrast, we show that the proposed algorithm limits the number of selected features only to those features which are most relevant, and so in this sense is less vulnerable than other methods to overfitting. Further, with regard to classification, we investigate the Vapnik Che1rvonenkis (VC) dimension for the proposed classifier structure. Under certain assumptions, we show that the value

of the VC dimension for the localized classifier is moderate. A modest value of VC dimension also implies reduced sensitivity to overfitting.

The rest of this thesis is organized as follows: Chapter 2 presents and demonstrates the localized feature selection and classification idea (referred as LFS method). The LFS method is published in IEEE Transactions on Pattern Analysis and Machine Intelligence. An improved version of the LFS feature selection method is presented in Chapter 3, referred as logistic Localized Feature Selection lLFS. The idea of the lLFS approach is submitted to IEEE Transactions on Neural Network and Learning Systems which is under second revision. An application of the proposed localized feature selection idea for automatic and continuous detection of Mismatch Negativity (MMN) is presented in Chapter 4. Conclusions and future works are presented in Chapter 5.

Chapter 2

Localized Feature Selection (LFS)

2.1 Abstract

The main idea of this thesis is to present the novel concept of localized feature selection (LFS) for data classification and its application for coma outcome prediction.

Typical feature selection methods choose an optimal global feature subset that is applied over all regions of the sample space. In contrast, in this study we propose a novel localized feature selection approach whereby each region of the sample space is associated with its own distinct optimized feature set, which may vary both in membership and size across the sample space. This allows the feature set to optimally adapt to local variations in the sample space. An associated localized classification method is also proposed.

The proposed LFS method selects a feature subset such that, within a localized region, within-class and between-class distances are respectively minimized and maximized. We first determine the localized region using an iterative procedure based on the distances in the original feature space. This results in a linear programming

optimization problem. Then, the second method is formulated as a non-linear joint convex/increasing quasi-convex optimization problem where a logistic function is applied to focus the optimization process on the localized region within the unknown co-ordinate system. This results in a more accurate classification performance at the expense of some sacrifice in computational time. Experimental results on synthetic and real-world data sets demonstrate the effectiveness of the proposed localized approach.

Using the LFS idea, we propose a practical machine learning approach for automatic and continuous assessment of event related potentials for detecting the presence of the mismatch negativity component, whose existence has a high correlation with coma awakening. This process enables us to determine prognosis of a coma patient. Experimental results on normal and comatose subjects demonstrate the effectiveness of the proposed method.

2.2 Proposed LFS Method

The proposed method is presented in two parts: feature selection and class similarity measurement. In the former, a discriminative subset of features is selected for each of the sample space regions. In the latter, a localized classifier structure for measuring the similarity of a query datum to a specific class is presented.

2.2.1 Feature selection

Assume that we encounter a classification problem with N training samples $\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N} \subset \mathbb{R}^{\mathrm{M}} \times \mathcal{Y}$ where $\mathcal{Y} = \{Y_1, \ldots, Y_c\}$ is the set of class labels, $\mathbf{x}^{(i)}$ is the i^{th} training sample

containing M features and $y^{(i)} \in \mathcal{Y}$ is its corresponding class label.

To implement the proposed localized feature selection scheme, we consider each training sample $\mathbf{x}^{(i)}$ to be a representative point for its neighboring region and assign an M-dimensional indicator vector $\mathbf{f}^{(i)} \in \{0,1\}^M$ to $\mathbf{x}^{(i)}$ that indicates which features are optimal for local separation of classes. If the element $f_m^{(i)} = 1$, then the mth feature is selected for the ith sample, otherwise it is not. The optimal indicator vector $\mathbf{f}^{(i)}$ is computed such that, in its respective subspace, the neighboring samples with class label similar to $y^{(i)}$ cluster as closely as possible around $\mathbf{x}^{(i)}$, whereas samples with differing class labels are as far away as possible. No assumptions are made that require the classes to be unimodal, nor on the probability distribution of the samples. In this work, Euclidean distance is used as the distance measure.

The following will present the process of calculating $\mathbf{f}^{(i)}$ corresponding to the representative point $\mathbf{x}^{(i)}$.

Initial formulation

Assume that $\mathbf{x}_p^{(k,i)}$ is the projection of an arbitrary training sample $\mathbf{x}^{(k)}$ into the subspace defined by $\mathbf{f}^{(i)}$ as follows:

$$\mathbf{x}_p^{(k,i)} = \mathbf{x}^{(k)} \otimes \mathbf{f}^{(i)}\,, k = 1, \ldots, N \tag{2.1}$$

where $\otimes$ is the element-wise product. In the sequel, projection into the space defined by $\mathbf{f}^{(i)}$ is implied, so dependence on i in $\mathbf{x}_p^{(k,i)}$ is suppressed.

We want to encourage clustering behaviour – i.e. in the neighborhood of $\mathbf{x}_p^{(i)}$, we want to find an optimal feature subset $\mathbf{f}^{(i)}$ so that, in the corresponding local co–ordinate system, we satisfy the following two goals:

1. neighboring samples of the same class are closely situated around $\mathbf{x}_p^{(i)}$, and simultaneously,

2. neighboring samples with different classes are further removed from $\mathbf{x}_p^{(i)}$.

To realize these goals, we define $N-1$ objective functions which are weighted distances of all within- and between-class samples to be respectively minimized and maximized as in (2.2).

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \in \mathbf{y}^{(i)},\ j \neq i \\ &\max_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \notin \mathbf{y}^{(i)} \end{aligned} \tag{2.2}$$

where $\mathbf{y}^{(i)}$ is the set of all training samples with class label similar to $y^{(i)}$. The quantity $w_j^{(i)}$ is the weight of the corresponding distance where, in order to concentrate on neighboring samples and reduce the effect of remote samples on the objective functions, higher weights are assigned to the closer samples of $\mathbf{x}_p^{(i)}$. Weights decrease exponentially with increasing distance from $\mathbf{x}_p^{(i)}$. However, measuring sample distances from $\mathbf{x}_p^{(i)}$ is a challenging issue since these distances should be measured in the local co–ordinate system defined by $\mathbf{f}^{(i)}$, which is unknown at the problem outset. To overcome this issue, we use an iterative approach for computing $\mathbf{f}^{(i)}$, where at each iteration weights are determined based on the distances in the co–ordinate system defined at the previous iteration. The following discussion assumes the weights have been determined in this manner. Further discussion on the computation of the weights is given later in Section Weight definition.

There are constraints that must be considered in our optimization formulations. Since we are looking for an indicator vector $\mathbf{f}^{(i)} = (f_1^{(i)}, f_2^{(i)}, \ldots, f_M^{(i)})^\mathsf{T}$ the problem

variables $f_m^{(i)}, m = 1, \ldots, M$ are restricted to 0 and 1, where $(.)^\mathsf{T}$ is transpose operator. Because there must be at least one active feature in $\mathbf{f}^{(i)}$, the null binary vector must be excluded, i.e. $1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)}$ where $\mathbf{1}$ is an M dimensional vector with all elements equal to 1. Furthermore, we would like to limit the maximum number of active features to a user–settable value α, i.e. $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha$, where α must be an integer number between 1 and M. Therefore, the feature selection problem for the neighboring region of $\mathbf{x}^{(i)}$ can be written as follows:

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \in \mathbf{y}^{(i)},\ j \neq i \\ &\max_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \notin \mathbf{y}^{(i)} \\ &\text{s.t.} \begin{cases} f_m^{(i)} \in \{0, 1\}, \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha \end{cases} \end{aligned} \tag{2.3}$$

where the notation $\{\cdot\}$ is used to indicate a discrete set, whereas the notation $[\cdot]$ is used later to indicate a continuous interval.

In the next section, the above optimization problem is reformulated into an efficient linear programming optimization problem.

Problem reformulation

To obtain a well-behaved optimization problem, in the following, we use the squared Euclidean distance instead of the Euclidean distance itself. It is apparent that the optimal solution of (2.3) is invariant to this replacement. Considering the sample projection definition in (2.1) and the fact that the problem variables $f_m(i)$, $m =$

$1, \ldots, M$ are binary, each objective function of (2.3) can be simplified as follows:

$$\begin{aligned} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2^2 &= w_j^{(i)} \left\| \left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)} \right) \otimes \mathbf{f}^{(i)} \right\|_2^2 \\ &= w_j^{(i)} \sum_{m=1}^{M} \left(\delta_{j,m}^{(i)} f_m^{(i)} \right)^2 \\ &= w_j^{(i)} \sum_{m=1}^{M} f_m^{(i)} \delta_{j,m}^{(i)^2} \\ &= w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \end{aligned} \tag{2.4}$$

where $\boldsymbol{\Delta}_j^{(i)} = \left(\delta_{j,1}^{(i)^2}, \delta_{j,2}^{(i)^2}, \ldots, \delta_{j,M}^{(i)^2} \right)^\mathsf{T} \triangleq \left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)} \right) \otimes \left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)} \right)$. $\left(f_m^{(i)} \right)^2$ in the second line is replaced with $f_m^{(i)}$ due to the first constraint in (2.3). The important conclusion drawn is that the objective functions are *linear* in terms of the problem variables.

Using the summation of all weighted within-class distances and all weighted between-class distances in the sub-feature space defined by $\mathbf{f}^{(i)}$, we define the *total intra-class distance* and the *total inter-class distance* as in (2.5). The problem is then reformulated by simultaneously minimizing the former and maximizing the later.

$$\begin{aligned} &total\, intra - class\, distance: \\ &\qquad \sum_{j \in \mathbf{y}^{(i)}} \left(w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \right) \triangleq \mathbf{a}^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \\ &total\, inter - class\, distance: \\ &\qquad \sum_{j \notin \mathbf{y}^{(i)}} \left(w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \right) \triangleq \mathbf{b}^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \end{aligned} \tag{2.5}$$

We see that (2.3) is in the form of an integer program, which is known to be

computationally intractable (Boyd and Vandenberghe, 2004). However this issue is readily addressed through the use of a standard and widely–accepted approximation of an integer programming problem (Thai, 2013; Souza, 2001; Boyd and Vandenberghe, 2004). Here, we replace (relax) the binary constraint in (2.3) with linear inequalities $0 \leq f_m^{(i)} \leq 1, m = 1, \ldots, M$. This procedure restores the computational efficiency of the program. A randomized rounding procedure (to be discussed further) that maps the linear solution back onto a suitable point on the binary grid, then follows.

These reformulations result in (2.6), which is a multi-objective optimization problem consisting of two linear objective functions that are to be simultaneously minimized and maximized, along with $2M + 2$ linear constraints.

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} \mathbf{a}^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \\ &\max_{\mathbf{f}^{(i)}} \mathbf{b}^{(i)^\mathsf{T}} \mathbf{f}^{(i)} \\ \text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^\mathsf{T} \mathbf{f}^{(i)} \leq \alpha \end{cases} \end{aligned} \tag{2.6}$$

There are several ways to re-configure a multi-objective problem into a standard form (Boyd and Vandenberghe, 2004; Hwang *et al.*, 1979; Mavrotas, 2009) with a single objective function; e.g. a linear combination of the objective functions. In the multi-objective case, the concept of optimality is replaced with *Pareto* optimality. A Pareto optimal solution is one in which an improvement in one objective requires a degradation of another. Since our multi-objective optimization problem is convex (because both objective functions and the constraints defined in (2.6) are convex) the set of achievable objectives Λ is also convex. The solution to a multi-objective

optimization problem is not unique and consists of the set of all Pareto optimal points that are on the boundary of the convex set Λ. Different points in the set correspond to different weightings between the two objective functions. The set of Pareto points is unique and independent of the methodology by which the two functions are weighted (for more detail about Pareto optimal approach see (Boyd and Vandenberghe, 2004)). In this study, we use the ϵ-constraint method as described by (2.7), such that instead of maximizing the total inter-class distance, we force it to be greater than some constant $\epsilon^{(i)}$. In this way we can map out the entire Pareto optimal set by varying a single parameter, $\epsilon^{(i)}$. One advantage of this approach is that we can guarantee the combined inter-class distances are in excess of the value of the parameter $\epsilon^{(i)}$.

$$\min_{\mathbf{f}^{(i)}} \mathbf{a}^{(i)\mathsf{T}}\mathbf{f}^{(i)}$$
$$\text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} \leq \alpha \\ \mathbf{b}^{(i)\mathsf{T}}\mathbf{f}^{(i)} \geq \epsilon^{(i)} \end{cases} \tag{2.7}$$

The parameter $\epsilon^{(i)}$ must be determined such that the optimization problem defined in (2.7) is feasible. In the next section we present an approach to automatically determine a value of the parameter $\epsilon^{(i)}$ which guarantees that the feasible set is not empty.

Problem feasibility

The optimization problem defined in (2.7) is feasible if there is at least one point that satisfies its constraints. The constraints $f_m^{(i)} \in [0,1], m = 1, \ldots, M$ indicate that the

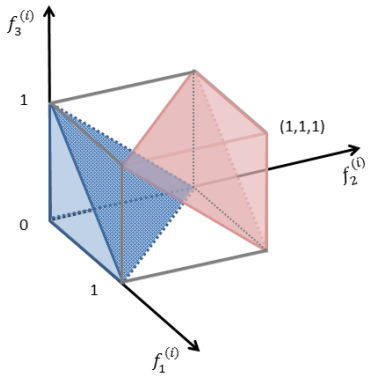


Figure 2.1: The polyhedron $\mathcal{P}$ in the case of a 3-D original feature space, i.e. the data dimension M is 3, where α is set to 2. It is a unit cube (defined by $0 \leq f_m^{(i)} \leq 1,\ m = 1, \ldots, 3$) in which two regions, i.e. blue and red pyramids, are removed. The blue pyramid is the intersection between unit cube and the half space $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} < 1$, and the red pyramid is the intersection between the half space $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} > \alpha$ and the unit cube.

optimum point must be inside a unit hyper-cube. The constraints $1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha$ indicate that the optimum point must be within the space between two parallel hyper-planes defined by $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} = 1$ and $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} = \alpha$. Since α is an integer number greater than or equal to 1, the space bounded by these two parallel hyper-planes is always non-empty and its intersection with the unit hyper-cube is also non-empty. In fact, the intersection of the spaces defined by $f_m^{(i)} \in [0, 1], m = 1, \ldots, M$ and $1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha$ is a polyhedron $\mathcal{P}$ that can be seen as a unit cube in which two parts are removed; the first part is the intersection between the half-space $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} < 1$ and the unit hyper-cube, and the second is the intersection between the half-space $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} > \alpha$ and the unit hyper-cube (see Fig. 2.1). If the intersection between the polyhedron $\mathcal{P}$ and the half-space defined by $\mathbf{b}^{(i)^\mathsf{T}}\mathbf{f}^{(i)} \geq \epsilon^{(i)}$, i.e. the last constraint, is non-empty then the optimization problem is feasible. The maximum value $\epsilon_{max}^{(i)}$ that $\epsilon^{(i)}$ can take such that the intersection remains non-empty is the solution to the following feasibility LP

problem:

$$\max_{\mathbf{f}^{(i)}} \mathbf{b}^{(i)\mathsf{T}}\mathbf{f}^{(i)}$$
$$\text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} \leq \alpha. \end{cases} \tag{2.8}$$

Effectively, (2.8) corresponds to an extreme Pareto point where the weighting given to the intra-class distance term (the first objective in (2.6)) is zero. Finally, we set $\epsilon^{(i)} = \beta\epsilon_{max}^{(i)}$ where β lies between zero and one. In this way, the optimization problem is always feasible and by changing β we can map out the entire Pareto optimal set corresponding to different relative weightings of intra- vs. inter-class distances. Here we define the Pareto optimal point corresponding to a specific value of β as $\mathbf{f}_\beta^{(i)}$; furthermore we define the set $\left\{\mathbf{f}_\beta^{(i)}\right\}_{\beta\in[0,1]}$ as the complete Pareto optimal set. The final reformulation of the problem may therefore be expressed as:

$$\min_{\mathbf{f}_\beta^{(i)}} \mathbf{a}^{(i)\mathsf{T}}\mathbf{f}_\beta^{(i)}$$
$$\text{s.t.} \begin{cases} f_{m,\beta}^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}_\beta^{(i)} \leq \alpha \\ \mathbf{b}^{(i)\mathsf{T}}\mathbf{f}_\beta^{(i)} \geq \beta\epsilon_{max}^{(i)}. \end{cases} \tag{2.9}$$

where $\mathbf{f}_\beta^{(i)} = (f_{1,\beta}^{(i)}, f_{2,\beta}^{(i)}, \ldots, f_{M,\beta}^{(i)})^{\mathsf{T}}$. This formulation has the desirable form of a *linear program* and hence is convex.

The solution to (2.9) provides a solution for each element of $\mathbf{f}_\beta^{(i)}$ over the continuous range $[0,1]$ that may be considered close to the corresponding *binary* Pareto optimal

solution $\mathbf{f}_{\beta}^{*^{(i)}}$. To obtain $\mathbf{f}_{\beta}^{*^{(i)}}$, a randomized rounding process (Thai, 2013; Souza, 2001; Boyd and Vandenberghe, 2004) is applied to the optimal point of (2.9), i.e. $\mathbf{f}_{\beta}^{(i)}$, where $f_{m,\beta}^{(i)}$ is set to one with probability $f_{m,\beta}^{(i)}$ and is set to zero with probability $(1 - f_{m,\beta}^{(i)})$ for $m = 1, \ldots, M$. To explore the entire region surrounding the Pareto optimal $\mathbf{f}_{\beta}^{(i)}$, the randomized rounding process is repeated 1000 times and the point that simultaneously satisfies constraints of (2.9) and provides the minimum value for the objective function of (2.9) is chosen as the binary Pareto point $\mathbf{f}_{\beta}^{*^{(i)}}$. Among the binary Pareto optimal points $\left\{\mathbf{f}_{\beta}^{*^{(i)}}\right\}_{\beta\in[0,1]}$ the one which yields the best local clustering of samples is chosen as the *binary* feature vector $\mathbf{f}^{*^{(i)}}$ corresponding to the representative point $\mathbf{x}^{(i)}$. This process is explained more in detail in Section 2.2.2.

Weight definition

In order to compute the sub-feature set $\mathbf{f}^{*^{(i)}}$ corresponding to the representative point $\mathbf{x}^{(i)}$, the proposed method focuses on the neighboring samples by assigning higher weights to them. However, the computation of the weights is dependent on the co–ordinate system, which is defined by $\mathbf{f}^{*^{(i)}}$, which is unknown at the problem outset. To overcome this problem, we use an iterative approach. At each iteration, weights $w_j^{(i)}, j = 1, \ldots, N,\ j \neq i$ (see (2.3)) are computed using the previous estimates of $\mathbf{f}^{*^{(i)}}, i = 1, \ldots, N$. Initially, the weights are all assigned uniform values. Empirically, if two samples are close to each other in one space, they are also close in most of the other sub-spaces. Therefore we define $w_j^{(i)}$, using the distance between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$

in all N subspaces obtained from the previous iteration, in the following manner:

$$\begin{aligned} w_j^{(i)} &= \frac{1}{N}\left(\sum_{k=1}^{N}\exp\left(-\left(d_{ij|k}-d_{ij|k}^{min}\right)\right)\right) \\ d_{ij|k} &= \left\|\left(\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\right)\otimes\mathbf{f}^{*^{(k)}}\right\|_2 \\ d_{ij|k}^{min} &= \begin{cases} \min\limits_{v\in\mathbf{y}^{(i)}} d_{iv|k} & \text{, if } y^{(j)}=y^{(i)} \\ \min\limits_{v\notin\mathbf{y}^{(i)}} d_{iv|k} & \text{, if } y^{(j)}\neq y^{(i)} \end{cases} \end{aligned} \tag{2.10}$$

where $\mathbf{f}^{*^{(k)}}$, $k=1,\ldots,N$ are known from the previous iteration. Such a definition implies all the $w_j^{(i)}$ are normalized over $[0,1]$.

The pseudo code of the proposed feature selection method is presented in Algorithm 1 where the parameter τ is the number of iterations and is set to its default value 2 in all our experiments.

2.2.2 Class similarity measurement

The localized feature selection approach results in optimal feature set variation over the sample space. Hence conventional classifiers are inappropriate. In this section we build a classifier which is appropriate for the localized scenario. The proposed localized classifier classifies a query datum $\mathbf{x}^q$ based on measuring distances in the induced feature spaces defined by the optimal feature sets $\mathbf{f}^{\star^{(i)}}, i=1,\ldots,N$.

The proposed localized feature selection algorithm assumes that the sample space is formed from N, probably overlapped, regions around representative points. Here, we define each region to be a hyper-sphere $\mathcal{Q}^{(i)}$ centered at $\mathbf{x}_p^{(i)}$ (i.e., the projection of $\mathbf{x}^{(i)}$ into the subspace defined by $\mathbf{f}^{(i)}$) with class label $y^{(i)}$. In this study, we determine

Input: $\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$, τ, α

Output: $\left\{\mathbf{f}^{\star(i)}\right\}_{i=1}^{N}$

Initialization: Set $\mathbf{f}^{\star(i)} = (0, \ldots, 0)^{\mathsf{T}}$, $i = 1, \ldots, N$;
for *iteration* $\leftarrow 1$ **to** τ **do**
  $\mathbf{f}_{\text{prev.}}^{\star(i)} = \mathbf{f}^{\star(i)}$, $i = 1, \ldots, N$;
  for $i \leftarrow 1$ **to** N **do**
    Compute $w_j^{(i)}$, $j = 1, \ldots, N-1$ using $\left\{\mathbf{f}_{prev.}^{\star(k)}\right\}_{k=1}^{N}$ as in (2.10);
    Compute $\epsilon_{max}^{(i)}$ through solving (2.8);
    for $\beta \leftarrow 0$ **to** 1 **do**
      Compute $\mathbf{f}_{\beta}^{(i)}$ through solving (2.9);
      Compute $\mathbf{f}_{\beta}^{\star(i)}$ through randomized rounding of $\mathbf{f}_{\beta}^{(i)}$;
    end
    Set $\mathbf{f}^{\star(i)}$ equal to the member of $\left\{\mathbf{f}_{\beta}^{\star(i)}\right\}_{\beta\in[0,1]}$ which yields the best local performance as explained in Section 2.2.2;
  end
end

Algorithm 1: pseudo code of the proposed feature selection algorithm.

the radius, i.e. $r^{(i)}(\gamma)$, of $\mathcal{Q}^{(i)}$ such that the "impurity level" within the hyper-sphere $\mathcal{Q}^{(i)}$ is not greater than the user-defined parameter γ. The "impurity" level is the ratio of the number of inter-class samples within $\mathcal{Q}^{(i)}$ to the number of intra-class samples within $\mathcal{Q}^{(i)}$. In all our experiments, γ is fixed at the value 0.2 (its default value).

The similarity $S_{Y_\ell}(\mathbf{x}^q; \gamma)$ of query datum $\mathbf{x}^q$ to class $Y_\ell \in \mathcal{Y}$ is measured based on how many hyper-spheres with class label Y_ℓ contain $\mathbf{x}^q$. To this end, we define a set of binary variables $s^{(i)}(\mathbf{x}^q; \gamma) : \mathbb{R}^M \rightarrow \{0, 1\}$, $i = 1, \ldots, N$, defined as follows:

$$s^{(i)}(\mathbf{x}^q; \gamma) = \text{step}[r^{(i)}(\gamma) - \left\|\mathbf{x}_p^{(i)} - \mathbf{x}_p^{(q)}\right\|_2] \tag{2.11}$$

where

$$\text{step}(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

The $s^{(i)}(\mathbf{x}^q;\gamma)$ may be interpreted as "weak" classifiers that indicate the similarity of $\mathbf{x}^q$ to the corresponding region[1]. The similarity of $\mathbf{x}^q$ to the class Y_ℓ, $S_{Y_\ell}(\mathbf{x}^q;\gamma)$, is computed through aggregation of the "weak" classifier results corresponding to the regions whose class labels are the same as Y_ℓ, as follows:

$$S_{Y_\ell}(\mathbf{x}^q;\gamma) = \frac{\sum_{i\in\mathbb{Y}_\ell} s_i(\mathbf{x}^q;\gamma)}{\eta_\ell} \tag{2.13}$$

where $\mathbb{Y}_\ell$ indicates the set of all regions whose class labels are Y_ℓ. η_ℓ is the cardinality of $\mathbb{Y}_\ell$. We compute the $S_{Y_\ell}(\mathbf{x}^q;\gamma), \ell = 1,\ldots,c$ and the class label of $\mathbf{x}^q$, i.e. y^q, is the one which has the largest similarity :

$$y^q = \underset{Y_l\in\mathcal{Y}}{\text{argmax}}\{S_{Y_1}, S_{Y_2}, \ldots, S_{Y_c}\}. \tag{2.14}$$

If $\mathbf{x}^q$ is not situated in any of the hyper-spheres $\mathcal{Q}^{(i)}$ $i = 1,\ldots,N$, then we would like its class label to be determined based on the class label of its nearest neighboring sample. However, since there are N local co–ordinate systems in which to measure distance, which one or ones are appropriate? To address this matter, we evaluate the set of distances of all N nearest neighbors as measured in each co–ordinate system. The class of $\mathbf{x}^q$ is then determined using a majority voting procedure over the corresponding classes in the set. The number of votes for each

[1]Heuristically, slightly better results may be obtained if the neighboring sample of $\mathbf{x}^q$ is also considered–i.e., $s^{(i)}(\mathbf{x}^q;\gamma)$ is set to 1 if the output of equation (2.11) is 1 and the class label of the nearest neighbor is $y^{(i)}$. However, since here $\gamma = 0.2$ the effect of the neighboring sample is small.

class is normalized to the total number of samples within that class. It is to be noted that such a situation is a rare occurrence in all our experiments – only 0.03%.

In the following we discuss an approach to determine an appropriate value for β which results in the selection of a suitable point in the Pareto set. We solve (2.9) for different values of β followed by the randomized rounding process to obtain $\mathbf{f}_{\beta}^{\star(i)}$, where β ranges from 0 to 1 with increments of 0.05. Each candidate binary vector $\mathbf{f}_{\beta}^{\star(i)}$ defines a local co-ordinate system and therefore specifies the respective hyper-sphere $\mathcal{Q}^{(i)}$ and the weak classifier $s^{(i)}$. The local clustering performance corresponding to $\mathbf{f}_{\beta}^{\star(i)}$ is then determined using a leave-one-out cross-validation procedure over the training samples situated within $\mathcal{Q}^{(i)}$. Performance is evaluated using decisions from the respective weak classifier $s^{(i)}$. Finally, among the candidate binary points $\left\{\mathbf{f}_{\beta}^{\star(i)}\right\}_{\beta\in[0,1]}$ the one with the best local clustering performance is chosen as the optimum binary feature set $\mathbf{f}^{\star(i)}$ corresponding to the representative point $\mathbf{x}^{(i)}$ (see line 11 of Algorithm 1).

Fig. 2.2 shows a block diagram of the proposed algorithm.

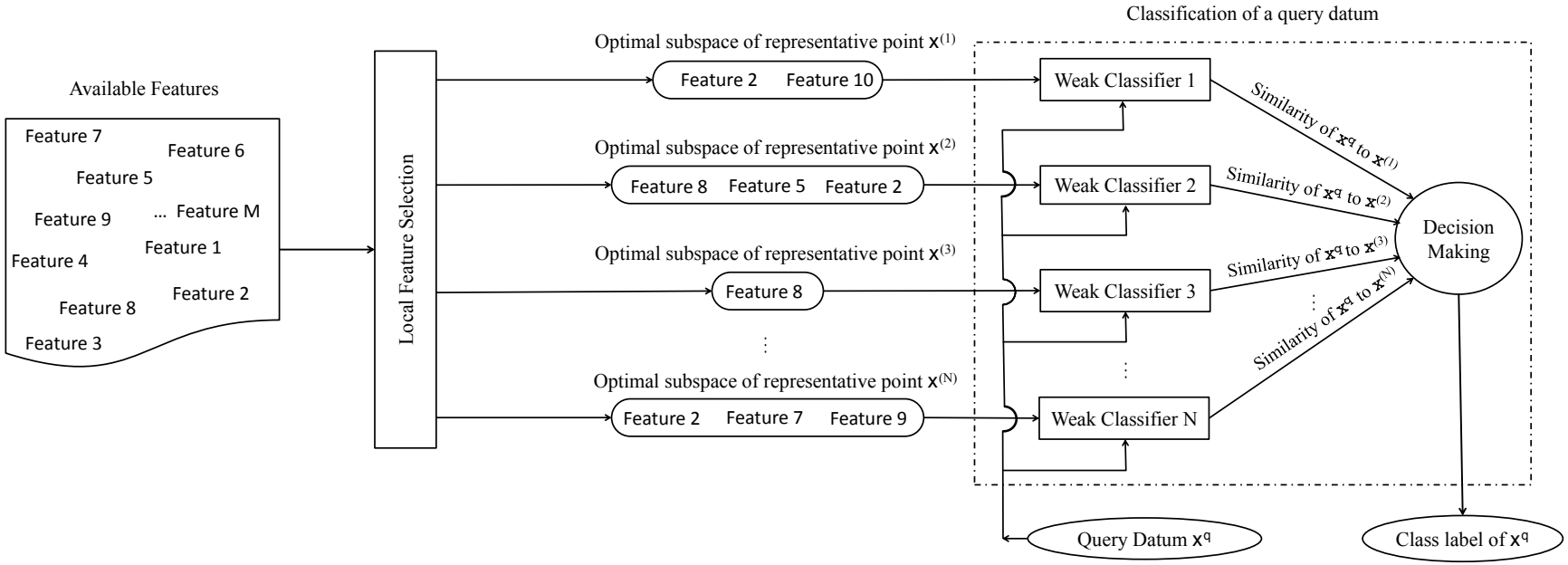


Figure 2.2: Block diagram of the proposed algorithm for data classification. The neighboring region of each representative point is modeled by an optimal feature subset selected from the available feature pool. Details of the local feature selection and classification procedures for a query datum $\mathbf{x}^q$ are presented in Sections 2.2.1 and 2.2.2, respectively.

2.3 Properties of the proposed algorithm

In this section we present three important properties of the proposed approach defined in Section 2.2. These properties are 1) that the proposed localized classifier defined in Section 2.2.2 has a modest Vapnik–Chervonenkis (VC) dimension, 2) that the proposed approach is insensitive to the overfitting problem, and 3) that the proposed feature selection method may be parallelized.

2.3.1 Vapnik Chervonenkis (VC) dimension

The Vapnik–Chervonenkis (VC) dimension (Vapnik, 1998) is used to quantify the "power" of a classifier to separate points in a feature space. A classifier with a larger VC value indicates higher classification power, yet may be prone to over–fitting, compared to one with a lower VC dimension.

A classifier structure may be represented by a family $\mathcal{F}$ of functions parameterized by a set θ, such that $\mathcal{F} = \{f(\mathbf{x};\theta) : \mathbb{R}^M \rightarrow \mathcal{Y}\}$ where $\mathbf{x}$ is a training sample. For example, in the case of the linear perceptron, $f = \text{sign}\{\theta_1^T\mathbf{x} - \theta_2\}$ where $\theta^T = \left[\theta_1^T, \theta_2\right]$. Consider a training set $\mathbf{X}_N = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^N \subset \mathbb{R}^{\mathrm{M}} \times \mathcal{Y}$. Then $\mathcal{F}$ "shatters" this set if there exist values of θ which can correctly classify the training samples corresponding to all possible c^N combinations of the respective y–values where c is cardinality of $\mathcal{Y}$. The VC dimension is the largest N which can be shattered. For example, in the case of a two class problem, the linear perceptron classifier has a VC dimension of $M + 1$ (Burges, 1998). The VC dimension plays an important role in establishing bounds on the performance of the classifier.

The VC dimension h for the LFS classifier is developed in the Appendix A, and under certain assumptions, is shown to be equal to the value $L(\lceil\frac{1}{\gamma}\rceil - 1)$, where L is

the number of clusters in the training set and $\lceil \cdot \rceil$ denotes the ceiling function.

The fact that the LFS classifier has a finite VC dimension means that a variety of learning theoretic performance bounds can be applied in this situation. One such bound relates to how well a learning algorithm trained on a finite training set will generalize to unseen data (Vapnik, 1998). In this respect and under the assumption that all training points are drawn *i.i.d* from some distribution $\mathcal{D}(\mathbf{x}, y)$, i.e. $\mathbf{X}_N \sim \mathcal{D}^N$, and under the assumption that future test points will drawn from the same distribution, we can define an *empirical* risk and an *expected* risk (Burges, 1998; Vapnik, 1998), respectively as in (2.15) and (2.16):

$$\mathcal{R}_N(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} |y^{(i)} - f(\mathbf{x}^{(i)}; \theta)|, \tag{2.15}$$

$$\mathcal{R}(\theta) = \int \frac{1}{2} |y - f(\mathbf{x}; \theta)| d\mathcal{D}(\mathbf{x}; y). \tag{2.16}$$

Assuming the empirical loss converges uniformly to the expected loss, then with probability $1 - \xi$, $\xi \in [0, 1]$, the following bound holds:

$$\mathcal{R}(\theta) \leq \mathcal{R}_N(\theta) + \sqrt{\frac{h(\log(\frac{2N}{h}) + 1) - \log(\frac{\xi}{4})}{N}}. \tag{2.17}$$

This bound indicates that, by minimizing $\mathcal{R}_N(\theta)$ over θ for a given training set, a minimum upper bound on expected performance over unseen samples is established if h is finite. See (Vapnik, 1998) for details.

Furthermore, a finite value of h permits us to make assertions regarding the *sample complexity* of the classifier. To this end we define the the optimal risk $\mathcal{R}^\star$ as follows:

$$\mathcal{R}^{\star}(\theta) = \inf_{\theta} \mathcal{R}(\theta) \tag{2.18}$$

Then a good training algorithm will generate an $\mathcal{R}_N(\theta)$ close to $\mathcal{R}^{\star}(\theta)$, or more precisely (Anthony and Bartlett, 1999), for a positive real number $\rho \in [0, 1]$, which is prescribed in advance, we have

$$\underset{\mathbf{x}_N \sim \mathcal{D}^N}{Pr}\{\mathcal{R}_N(\theta) < \mathcal{R}^{\star}(\theta) + \rho\} \geq 1 - \psi, \tag{2.19}$$

where $\psi \in [0, 1]$ tends to be a small value. N_o is the sample complexity. It indicates the number of training samples required for the error of the classifier to be well behaved. If a learning system has a finite VC dimension h, then the value of N_o can be bounded (Anthony and Bartlett, 1999) as follows:

$$N_o(\rho, \psi) \leq \frac{64}{\rho^2}\left(2h\log(\frac{12}{\rho}) + \log(\frac{4}{\psi})\right). \tag{2.20}$$

In many cases these bounds are of not much value in the practical setting, since they have been demonstrated to be very loose in some situations (Burges, 1998). However, these bounds do give us a sense that the empirical risk is not far from the expected risk for a reasonable value of N. Further, (2.20) suggests that the number of training samples required to guarantee a certain level of performance varies only logarithmically with the parameters ψ and ρ. Both these points suggest that with the LFS classifier, we can expect well behaved error performance, i.e., that the classifier will generalize well to new, unseen samples, under modest values of N.

2.3.2 LFS and the overfitting issue

Both the feature selection and classification processes contribute to the overfitting problem. As is discussed in Section 2.3.1 and Appendix A, the LFS classifier has a finite and moderate VC dimension value which is independent of the dimension of the feature space in which the classification is performed. Therefore it is less prone to the overfitting than a method with a high or infinite value of h (Burges, 1998).

We now discuss the LFS feature selection procedure with respect to overfitting. Assume that the set $\mathcal{X}$ denotes the set of all available features. Consider an ideal scenario in which, for each localized region, the set of available features $\mathcal{X}$ can be partitioned into two disjoint sets $\mathcal{X}_R^{(i)}$ and $\mathcal{X}_I^{(i)}$ so that $\mathcal{X}_R^{(i)} \cup \mathcal{X}_I^{(i)} = \mathcal{X},\ i = 1, \ldots, N$. $\mathcal{X}_R^{(i)}$ and $\mathcal{X}_I^{(i)}$ respectively denote the set of relevant and irrelevant features. Assume that the cardinality of $\mathcal{X}_R^{(i)}$ is $\zeta_R^{(i)}$.

Assume a hypothetical situation where the parameter α is set to $\zeta_R^{(i)}$. Note that "relevant" features are those that encourage local clustering behavior quantified by the optimization problem defined in (2.9). In this way, we assume that the features in $\mathcal{X}_R^{(i)}$ are sufficiently relevant to be selected by the proposed algorithm; i.e. the features in $\mathcal{X}_R^{(i)}$ with high probability are selected as the solution to (2.9) followed by the randomized rounding procedure. If α now grows above the value $\zeta_R^{(i)}$, the features in $\mathcal{X}_I^{(i)}$ become candidates to be selected. Since the features in $\mathcal{X}_I^{(i)}$ are "irrelevant" features, i.e. do not encourage local clustering behavior, their respective element in the optimal solution of (2.9) must be given a low value, i.e. a value close to zero in order to satisfy optimality. Hence, the features in $\mathcal{X}_I^{(i)}$, with high probability, are not selected after the randomized rounding process. Such a solution remains feasible because of the *inequality* constraint involving α in (2.9). Therefore, in this idealized

scenario, as α increases, the cardinality of the selected localized feature set tends to saturate at the level $\zeta_R^{(i)}$.

In the more practical scenario, the available feature set $\mathcal{X}$ may not be clearly partitioned into relevant and irrelevant features as we have assumed; hence, as α grows, "partially relevant" features may continue to be selected. Nonetheless, as is demonstrated in Section 2.4.5, the saturation behavior of the number of selected features is clearly evident in real-world scenarios.

In summary, the proposed LFS feature selection method chooses only relevant features. In this respect, it is less vulnerable to overfitting than methods which select a predetermined number of features. If this number is too high, then as indicated previously, these methods can select noisy features, making them prone to overfitting. Thus, both the LFS feature selection and classifier procedures are insensitive to the overfitting problem in the sense we have indicated.

2.3.3 LFS can be parallelized

The feature selection procedure for any representative point is independent of all other such points. This enables the localized feature selection process to be performed in parallel.

2.4 Experimental results

2.4.1 Experimental set-up

In this section we perform several experiments on one synthetic and ten binary real-world data sets to demonstrate the effectiveness of the proposed feature selection

algorithm.

In real world applications, obtaining labeled examples to be used as training samples is often very expensive and time consuming, as it requires the effort of human annotators, who must often be quite skilled. For instance, obtaining a single labeled example for protein shape classification, which is one of the grand challenges of biological and computational science, requires months of expensive analysis by expert crystallographers (Zhu *et al.*, 2003). Therefore one of the most relevant problems in the field of feature selection is where only a small number of training points is available for the training phase.

Small sample sizes, and their inherent risk of imprecision and overfitting, pose a great challenge for many modeling problems (Saeys *et al.*, 2007; Sima and Dougherty, 2006; Braga-Neto and Dougherty, 2004; Bolón-Canedo *et al.*, 2014). Hence the real world data sets used in our experiments have small number of training samples. Performance of the LFS method on data sets with relatively large number of training points is not the focus of this study.

The proposed algorithm is compared with eight state-of-the-art feature selection algorithms: Logo[2] (Sun *et al.*, 2010), FMS[3] (Cheng *et al.*, 2011), MBEGA[4] (Zhu *et al.*, 2007a), Elasticnet[5] (based on LARS-EN) (Zou and Hastie, 2005), kPLS[6] (Sun *et al.*, 2014), MetaDistance[7] (Liu *et al.*, 2011), DEFS[8] (Khushaba *et al.*, 2011) and mRMR[9] (Peng *et al.*, 2005) where the first 7 methods are specifically developed for

[2]http://plaza.ufl.edu/sunyijun/PAMI2.htm
[3]http://www2.cs.siu.edu/ ∼ qcheng/featureselection_pubfolder/index.html
[4]http://csse.szu.edu.cn/staff/zhuzx/MAFS.html
[5]http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3897
[6]https://github.com/sqsun/kernelPLS
[7]http://metadistance.igs.umaryland.edu/
[8]http://www.mathworks.com/matlabcentral/fileexchange/30877-differential-evolution-based-channel-and-feature-selection
[9]http://penglab.janelia.org/proj/mRMR/

the sparse data case where a small number of training samples are given. To have a fair comparison, parameters of all these feature selection algorithms as well as the proposed LFS algorithm were set to the default values suggested by the respective authors. In the case of the Elasticnet method, in the training phase, the regularization parameter δ that determines the weight of the l_2 penalty ranges from 10^{-3} to 10^3 (evenly spaced on the log-scale) where for each given δ the entire regularization path corresponded to the l_1 penalty is considered. Among the entire grid corresponding to these two regularization parameters, the node that provides the best fit on the training data (based on the Akaike's Information Criterion) is chosen as the regularization parameters corresponded to the l_2 and l_1 penalties for using in the test phase (see (Sjöstrand, 2005) for more details).

In order to evaluate classification accuracies corresponding to the features selected by our comparison algorithms, we use the SVM classifier with an RBF kernel. In each case, the top t features are selected by the respective algorithm, and then the SVM classifier is trained using the sampled training data in the induced feature subspace defined by these top-t features. Finally the sampled test data, in the respective induced subspace, are classified using the trained SVM, where following (Li *et al.*, 2015; Lovato *et al.*, 2016; Zhu *et al.*, 2007a; Peng *et al.*, 2005; Wang, 2008; Cheng *et al.*, 2011; Gilad-Bachrach *et al.*, 2004; Khushaba *et al.*, 2011; Zhu *et al.*, 2007b) the SVM classifier parameters are set to their default values (in MATLAB). To provide a fair comparison, parameter of the proposed localized classifier (i.e. γ) is also set to its default value 0.2 and is fixed during all experiments.

The proposed algorithm is implemented in MATLAB on a computer with an Intel(R) Core i7-2600 CPU @ 3.4 GHz and 16 GB RAM.

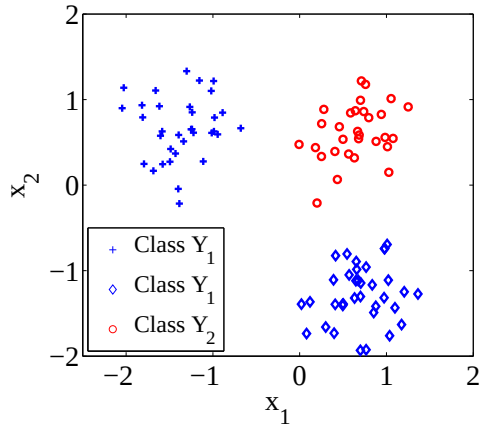


Figure 2.3: Illustration of the synthetic data set in terms of its relevant features x_1 and x_2, after feature values are transformed into their z-scores.

2.4.2 Data sets

We present our results using both synthetic and real-world data sets.

As is shown in Fig. 2.3, the synthetic data set is distributed in a two dimensional feature space where class Y_1 data is split into two discrete clusters. The features x_1 and x_2 of all subclasses '$\diamond$', '$+$' and '$\circ$' are drawn from Normal distributions with unit variances. Besides the two relevant features x_1 and x_2, following (Wang, 2008), each sample is artificially contaminated by adding a varying number of irrelevant features, ranging in number from 1 to $30,000$, as a means of testing the capability of the proposed method to detect only the most relevant features. The number $30,000$ is deemed to be a reasonable upper limit for most scientific applications (Sun *et al.*, 2010). The artificial irrelevant features are independently sampled from a Gaussian distribution with zero-mean and unit-variance.

Characteristics of the real-world data sets used for the experiments are summarized in Table 2.1. The total number of available labeled samples in each data set is given by the sum of the second and third columns.

Table 2.1: Characteristics of the real-world data sets used in the experiments.

Data set	**# Train**	**# Test**	**# Features (M)**
Sonar (Brown *et al.*, 2012)	100	108	60(100)
DNA (Wang, 2008)	100	3086	180(100)
Breast (Brown *et al.*, 2012)	100	469	30(100)
Adult (Bache and Lichman, 2013)	100	1505	119(100)
ARR (Alon *et al.*, 1999)	100	320	278(100)
Prostate (Nie *et al.*, 2010)	90	12	5966
Duke-breast (West *et al.*, 2001)	30	12	7129
Leukemia (Brown *et al.*, 2012)	60	12	7070
Colon (Alon *et al.*, 1999)	50	12	2000
Nervous system (Pomeroy *et al.*, 2002)	48	12	7129

The number of artificially added irrelevant features is indicated in parentheses.

To increase the challenge of the classification problems, following (Sun *et al.*, 2010), the original features of the data sets "Sonar", "DNA", "Breast", "Adult" and "ARR" are artificially augmented by 100 irrelevant features, independently sampled from a standard Normal distribution. Data sets "Prostate", "Duke-breast", "Leukemia", "Colon" and "Nervous system" are microarray data sets where in each case the number of features is significantly larger than the number of samples..

Each feature variable in the synthetic data set and the real-world data sets have been transformed beforehand to their z-score values.

2.4.3 Accuracy of classification

In this section, classification performance of the proposed LFS algorithm is compared with eight well-known feature selection algorithms indicated in Section 2.4.1.

In our experiments, the number of selected features t in our comparison feature selection algorithms and the parameter α of the LFS algorithm (which is analogous

to the parameter t) ranges from 1 to 30 for data sets "Sonar", "DNA", "Breast", "Prostate", "Duke-breast", "Leukemia" and "Colon", 1 to 60 for data set "Adult", 1 to 100 for data set ARR and 1 to 35 for data set "Nervous system", since there is no performance improvement for our comparison algorithms for larger values.

Following (Sun *et al.*, 2010), for each data set, a bootstrapping algorithm is used to evaluate the feature selection algorithms' performance. For this purpose, for a given t (α), each feature selection algorithm is run 10 times on each data set, where for each run the respective number of available data points, presented in the second column of Table 2.1, are randomly selected as training samples and the remaining data points, the number of which is indicated in the third column of Table 2.1, are used as test samples for that run. The average performance and the standard deviation over all 10 runs are recorded. For a fair comparison of different feature selection algorithms, the training and test sets for each run are common for all algorithms.

The minimum classification error rate, the corresponding standard deviation and the number of selected features t (α), for each algorithm on each data set, is reported in Table 2.2. In order to demonstrate the necessity for feature selection, we also report the classification error rate which results from applying the SVM classifier with an RBF kernel on each data set without prior feature selection. These results, shown in the last column of Table 2.2, are significantly degraded with respect to the case when feature selection is used, and thus demonstrate that the feature selection process is indeed an important component of the data classification process. The best result for each data set is shown in bold. Among the nine algorithms, the proposed LFS algorithm yields the best results in eight out of the ten data sets. (The improved version of the LFS method, i.e. the lLFS algorithm discussed in next

chapter, outerforms all methods on all data sets.) The last row shows the classification error rates averaged over all data sets. This row indicates that the proposed LFS method performs noticeably better on average than the other eight algorithms.

Table 2.2: Minimum classification error (in percent) of the different algorithms. The corresponding standard deviation (in percent) and t (α) are respectively reported in parenthesis. The last column corresponds to the classification results using SVM with no feature selection.

Data set	**LFS**	**Logo**	**FMS**	**MBEGA**	**Elasticnet**	**kPLS**	**MetaDist**	**DEFS**	**mRMR**	**SVM**
Sonar	**22.9**(3.9,30)	26.8(3.4,8)	28.8(2.6,14)	29.4(8.0,2)	27.7(4.2,5)	26.8(6.3,3)	28.9(9.9,12)	27.8(6.7,8)	28.7(2.6,1)	49.9(4.8)
DNA	**13.4**(1.9,15)	15.3(5.7,5)	15.3(1.8,6)	18.0(4.7,4)	16.1(4.7,3)	13.4(2.5,3)	27.0(10.7,6)	18.7(5.0,3)	13.8(3.0,3)	49.7(2.0)
Breast	**6.4**(1.3,11)	8.3(1.4,7)	7.7(1.4,9)	9.1(1.5,18)	8.8(1.5,3)	8.2(1.6,5)	12.9(6.0,9)	11.0(2.5,8)	8.3(2.2,4)	37.6(0.6)
Adult	**22.3**(1.5,30)	24.5(1.9,8)	24.7(0.3,46)	24.5(0.7,26)	24.6(0.5,19)	24.7(0.3,35)	24.3(1.0,9)	24.7(0.3,28)	24.8(0.3,30)	24.7(0.3)
ARR	33.1(2.6,29)	33.9(5.3,8)	32.2(2.9,34)	31.8(7.4,18)	38.7(3.6,9)	40.0(7.0,6)	40.7(5.7,80)	**31.4**(4.7,7)	31.6(3.3,10)	43.7(1.2)
Prostate	**4.2**(4.4,6)	8.3(7.9,3)	6.7(6.6,11)	7.5(8.3,18)	7.5(6.1,8)	6.7(8.6,2)	40.0(11.0,72)	13.7(9.6,4)	8.3(7.6,7)	57.5(10.7)
Duke-breast	**10.8**(7.9,3)	21.7(11.9,7)	24.2(13.3,4)	21.7(14.8,14)	32.5(14.4,11)	20.8(9.0,8)	38.3(19.7,10)	26.7(14.6,3)	21.7(5.8,5)	63.3(10.5)
Leukemia	3.3(4.3,30)	6.7(5.3,2)	**2.5**(4.0,2)	8.3(6.8,26)	6.7(5.3,3)	3.3(4.3,3)	26.7(8.6,18)	16.8(10.9,4)	5.0(5.8,8)	35.8(14.2)
Colon	**9.2**(0.1,21)	20.8(10.6,2)	13.3(9.0,6)	20.8(4.4,16)	15.0(11.0,3)	19.2(13.1,5)	25.0(6.8,3)	26.7(14.1,2)	19.2(5.6,4)	36.7(17.2)
Nervous sys.	**26.7**(9.5,4)	33.3(14.2,9)	35.0(20.3,20)	33.3(8.8,14)	35.0(14.0,12)	31.7(18.3,15)	30.0(9.0,7)	32.5(17.8,12)	32.5(16.4,2)	37.5(16.8)
Average	**15.2**	20.0	19.0	20.5	21.3	19.5	29.4	23.0	19.4	43.6

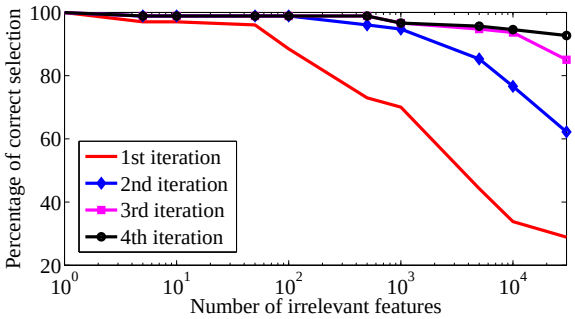


Figure 2.4: Percentage of correct feature selection over four successive iterations of the proposed algorithm for the synthetic data set, where the samples are contaminated with a varying number of irrelevant features. The parameter α is set to 2.

2.4.4 Iterative weight definition and correct feature selection

As illustrated in Fig. 2.3, the distribution of class Y_1 of the synthetic data set has two disjoint subclasses, whereas class Y_2 is a compact class with one mode. Samples of subclass '+' can be discriminated from samples of class Y_2 using only the relevant feature x_1. In a similar way, samples of subclass '$\diamond$' require only x_2, whereas samples of class '$\circ$' require both x_1 and x_2 . The results of applying the proposed method to the synthetic data set over four successive iterations is shown in Fig. 2.4, where samples have been contaminated with additional irrelevant features ranging in number from 1 to 30,000. Each point shows the percentage of samples for which the expected feature(s), (i.e. x_1 for samples within subclass '+', x_2 for samples within '$\diamond$' and $\{x_1, x_2\}$ for samples within '$\circ$'), are correctly identified. It can be seen that the performance is refined from one iteration to another, especially for a higher number of irrelevant features. The most significant improvement happens at the second iteration; hence, as mentioned previously, the default value of τ is set to 2.

The data set "DNA" has a "ground truth", in that much better performance has been previously reported if the selected features are those with indexes in the

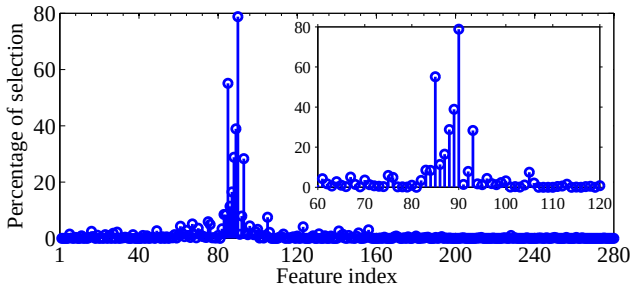


Figure 2.5: Selected features for "DNA" data set. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 5.

interval between 61 to 120 (John, 1994; Wang, 2008). This observation provides a good means of evaluating LFS performance on a real world data set. Fig. 2.5 shows the result of applying the proposed LFS method to the data set "DNA", where the height of each feature index indicates the percentage of representative points which select these ground-truthed features as a member of their optimal feature set. These results demonstrate that the proposed method mostly identifies features with indexes from 61 to 105. Thus they are well matched to the "ground truth". The proposed method also performs very well in discarding the artificially added irrelevant features, i.e. features with indexes from 181 to 280.

2.4.5 Sensitivity of the proposed method to α and γ

To show the sensitivity of the proposed method to the parameter α, the classification error rate and the cardinality of the optimal feature sets (averaged over all N sets) versus α, for data set "Sonar", are respectively shown in Fig. 2.6 and Fig. 2.7 where α ranges from 1 to the maximum possible value of $M = 160$. These results demonstrate the robustness of the proposed LFS algorithm against overfitting as discussed in Section 2.3.2.

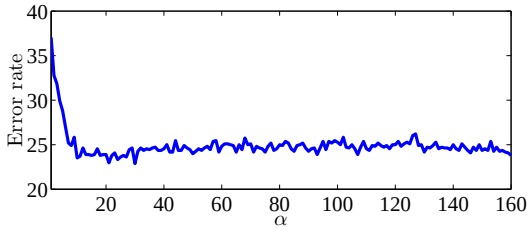


Figure 2.6: Classification error rate of the proposed method for data set "Sonar" where the parameter α ranges from 1 to the maximum possible value of $M = 160$.

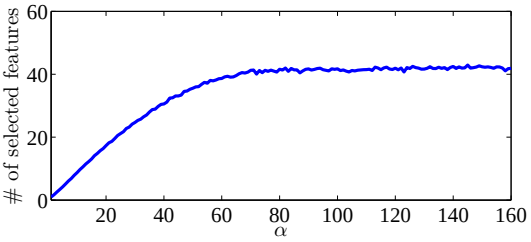


Figure 2.7: Averaged cardinality of the optimal feature sets $\mathbf{f}^{*(i)} i = 1, \ldots, N$ versus the parameter α where α ranges from 1 to the maximum possible value of $M = 160$.

Note that estimating an appropriate value for the number of selected features is generally a challenging issue. This is usually estimated using a validation set or based on prior knowledge, which may not be available in some applications. As can be seen, the proposed LFS algorithm is not too sensitive to this parameter. Moreover, as illustrated in Fig. 2.7, the cardinality of the optimal feature sets saturates for a sufficiently large value of α.

The error rate of the proposed method versus the impurity level parameter γ for data set "Colon" is shown in Fig. 2.8 where γ ranges from 0 to 1. Small (large) values of γ can be interpreted as a small (large) radius of the hyper-spheres. This demonstrates that the error rate is not too sensitive to a wide range of values of γ. As one may intuitively guess, we found that impurity levels in the range of 0.1 to 0.4 are appropriate. As mentioned previously, throughout all our experiments, γ is set

to 0.2 without tuning. This value is seen to work well over all data sets.

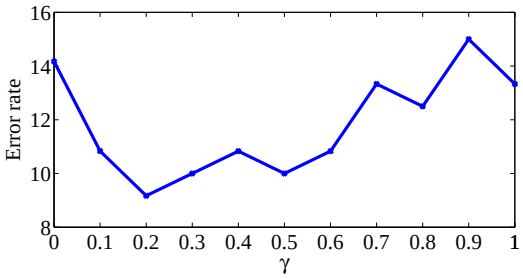


Figure 2.8: Classification error rate of the proposed method for data set "Colon" where the parameter γ ranges from 0 to 1.

2.4.6 How far is the binary solution from the relaxed one?

To demonstrate that the relaxed solutions are a proper approximation of the final binary solutions, obtained from the randomized rounding process explained in Section 2.2.1, the normalized histogram over the ℓ_1–norm distances between the relaxed solutions and their corresponding binary solutions is shown in Fig. 2.9. The height of each bar indicates what fraction of the representative points have the corresponding value as their ℓ_1–norm distance. The ℓ_1–norm distances are normalized relative to the data dimension M. As may be seen, the relaxed solutions are appropriate approximations of the binary solutions.

2.4.7 CPU time

The computational complexity for computing a feature set for each representative point depends mainly on the data dimension. Fig. 2.10 shows the CPU time taken by the proposed method (using MATLAB) to perform feature selection for one representative point on the synthetic data set, with the number of irrelevant features

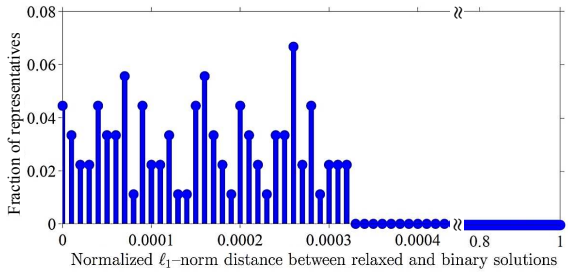


Figure 2.9: Histogram of distances between relaxed solutions and their corresponding binary solutions for data set "Prostate" where α is set to the typical value of 5.

ranging from 1 to 30000. As may be seen, the figure shows linear complexity of the LFS method with respect to feature dimensionality.

Note that the feature selection process for each representative point is independent of the others and can be performed in parallel. For instance, in the case of a data set with 100 training samples (i.e. $N = 100$) and 10,000 features (i.e. $M = 10,000$) on a typical desktop computer with 12 cores, the required processing time in the training phase is almost 25 seconds. Note again that this is the *training phase* time which is performed off-line. On the other hand, the *test phase* only involves testing whether the query datum contained within the specified hyper-spheres and determining the class label of its nearest neighbors. This is much faster than the training process, since

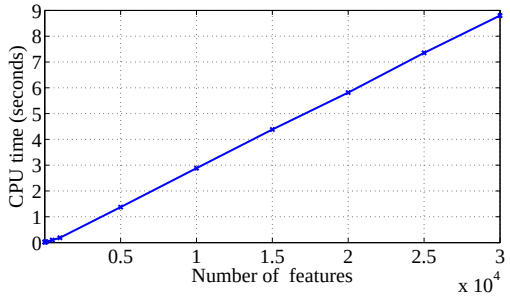


Figure 2.10: The CPU time (seconds) taken by the proposed algorithm to perform feature selection for one representative point $\mathbf{x}^{(i)}$ with a given β on the synthetic data set where the parameter α is set to 2.

it requires no optimization. In our experiments, the test phase is typically performed in a fraction of a second.

2.5 Conclusions

In this chapter we introduce the concept of localized feature selection. The proposed local feature selection algorithm adaptively assigns a specific optimal feature subset to each of the sample space regions, in contrast to traditional methods, which select a common feature set for the entire sample space. This allows the feature set to optimally adapt to local variations of the sample space.

The process of computing a specific feature subset for each region is independent of those of other regions and hence can be performed in parallel. Since the proposed algorithm makes no assumptions regarding the data distribution over the sample space, it is also an appropriate approach for the case where the data are distributed on a non-linear and/or a disjoint manifold. The LFS procedure is formulated as a linear program, which has the advantage of convexity and efficient implementation. A query datum is classified through aggregation of "weak" classifier results which are based on the selected region-specific feature subsets. The Vapnik–Chervonenkis (VC) dimension is determined and, under certain assumptions, is found to have a finite, moderate value. This, in combination with the fact that the method selects only relevant features, suggest the LFS method is not overly sensitive to the overfitting problem. Experimental results demonstrate the superior performance of the proposed algorithm on a large variety of data sets.

The following chapter is a reproduction of a submitted paper to IEEE Transactions on Neural Network and Learning Systems:

Narges Armanfard, James P. Reilly, "Logistic Localized Modeling of the Sample Space for Feature Selection and Classification", submitted to IEEE Transactions on Neural Network and Learning Systems, April 2016.

Chapter 3

Logistic Localized Feature Selection (lLFS)

3.1 Abstract

This chapter presents an improved version of the localized feature selection idea proposed in Section 2.2.1.

Let $\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N} \subset \mathbb{R}^{\mathrm{M}} \times \mathcal{Y}$ be the training data set of a c-class classification problem where N is the number of training samples, $\mathbf{x}^{(i)}$ is an M dimensional feature vector, $\mathcal{Y} = \{Y_1, \ldots, Y_c\}$ is the set of all class labels and $y^{(i)} \in \mathcal{Y}$ is the class label of the ith training sample $\mathbf{x}^{(i)}$.

As discussed in the previous chapter, our main idea for locally modeling the sample space is to assign a specific optimal feature subset to each of the sample space regions. To realize this goal, we assume that each sample $\mathbf{x}^{(i)}$ is a representative point for its neighboring region. For each representative point $\mathbf{x}^{(i)}$, we compute an M-dimensional indicator vector $\mathbf{f}^{(i)} \in \{0,1\}^{M}, i = 1, \ldots, N$, which indicates the relevant features

for the neighboring region of $\mathbf{x}^{(i)}$. We use the notation $\{\cdot\}$ to indicate a discrete set. For example, if the second and the fourth features are the relevant features for the neighboring region of $\mathbf{x}^{(i)}$, all elements of $\mathbf{f}^{(i)}$ are zero except the second and fourth ones. Thus $\mathbf{f}^{(i)}$ defines a local co–ordinate system, or *frame*. The vector $\mathbf{f}^{(i)}$ is computed such that, in the ith frame, neighboring samples of $\mathbf{x}^{(i)}$ whose class labels are similar to that of $\mathbf{x}^{(i)}$, i.e. $y^{(i)}$, cluster as closely as possible around $\mathbf{x}^{(i)}$, whereas samples with different class labels are as far removed as possible from $\mathbf{x}^{(i)}$.

Determining the neighboring samples is a challenging issue since these distance measures depend on the local co–ordinate system, which is determined by $\mathbf{f}^{(i)}$, which is unknown at the problem outset. In Section 2.2.1, the neighboring samples are mainly determined using an iterative approach initiated based on the distances in the original feature space. This is not a reliable procedure in the presence of a large number of irrelevant features, since distance measurements can vary strongly between the selected feature space and the original feature space. In this chapter, the distance measurement problem is alleviated, since the underlying optimization problem is formulated such that distances are a function of the unknown vector $\mathbf{f}^{(i)}$. Distances are measured using a logistic function metric within the corresponding co-ordinate system. This enables the optimization process to focus on a localized region within the sample space. We refer to the proposed algorithm as the logistic Localized Feature Selection (lLFS) method. lLFS is efficiently formulated as a joint convex/increasing quasi-convex optimization problem with a unique global optimum point. The local classification approach presented in Section 2.2.2 is utilized for measuring the similarity of a new input data point to each class. Using the lLFS method, similar to the LFS algorithm, feature selection processes for different regions

of the sample space are independent from each other and can therefore be performed in parallel. The computer implementation of the method can therefore be fast and efficient.

The proposed lLFS algorithm is presented in Section 3.2. Performance of the lLFS algorithm, on eleven synthetic and real-world data sets, is demonstrated in Section 3.3. Conclusions are drawn in Section 3.5.

3.2 Proposed lLFS method

This section is organized as follows. Section 3.2.1 presents the proposed formulation for the improved local feature selection lLFS. Accompanying optimization problem is treated in Section 3.2.2. Section 3.2.3 explains that the final formulation for the lLFS method is a joint convex/increasing quasi convex optimization problem with unique global optimum point. A procedure for determining the two required parameters of the proposed formulation is presented in Section 3.2.4.

3.2.1 Problem definition

Let $\mathcal{S}^{(i)}$ be the subspace of the original M–dimensional feature space whose axes correspond to the selected features. That is, an axis corresponding to a candidate feature is contained in $\mathcal{S}^{(i)}$ if the corresponding element of $\mathbf{f}^{(i)}$ is 1. Denote $\mathbf{x}_p^{(i)}$ as the projection of the ith training sample $\mathbf{x}^{(i)}$ into $\mathcal{S}^{(i)}$. In this study, the feature set $\mathbf{f}^{(i)} = (f_1^{(i)}, f_2^{(i)}, \ldots, f_M^{(i)})^\mathsf{T}$ is found such that the clustering behavior in the neighborhood of $\mathbf{x}_p^{(i)}$ is optimum with respect to the following two objectives:

- other samples of the same class cluster as closely as possible around $\mathbf{x}_p^{(i)}$, and

simultaneously,

- samples with different classes are separated as far as possible from $\mathbf{x}_p^{(i)}$, where distances in each case are measured within $\mathcal{S}^{(i)}$.

To quantify these goals, we consider the respective objective functions $\mathcal{U}_1$ and $\mathcal{U}_2$, defined by (3.1a) and (3.1b) as follows:

$$\mathcal{U}_1(\mathbf{f}^{(i)}) = \frac{1}{n-1} \sum_{j;y^{(j)}=y^{(i)},j\neq i} \mathcal{G}\left((\mathbf{a}_j^T\mathbf{f})^{(i)};\sigma^{(i)},\lambda\right) \tag{3.1a}$$

$$\mathcal{U}_2(\mathbf{f}^{(i)}) = \frac{1}{N-n} \sum_{j;y^{(j)}\neq y^{(i)}} \mathcal{G}\left((\mathbf{a}_j^T\mathbf{f})^{(i)};\sigma^{(i)},\lambda\right) \tag{3.1b}$$

The functions $\mathcal{U}_1$ and $\mathcal{U}_2$ may be regarded as local intra– and inter–class distance measures, respectively. The role of the function $\mathcal{G}(\cdot)$ is described later. The term $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ is the ℓ_1–norm of the distance vector between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ in $\mathcal{S}^{(i)}$. In fact, the simpler notation $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ replaces the more correct but awkward expression $\mathbf{a}_j^{(i)^\mathsf{T}}\mathbf{f}^{(i)}$; $\mathbf{a}_j^{(i)}$ is the ℓ_1 distance vector between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ in the original feature space, i.e. $\mathbf{a}_j^{(i)} = \left|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right|$ where $|\cdot|$ denotes the absolute value of the elements of the vector. The variables λ and $\sigma^{(i)}$ are parameters to be defined later in Sec. 3.2.4. The variable n is the number of samples whose class labels are $y^{(i)}$ and $(\cdot)^\mathsf{T}$ is transpose operator.

The local feature selection process may then be formulated in the context of the

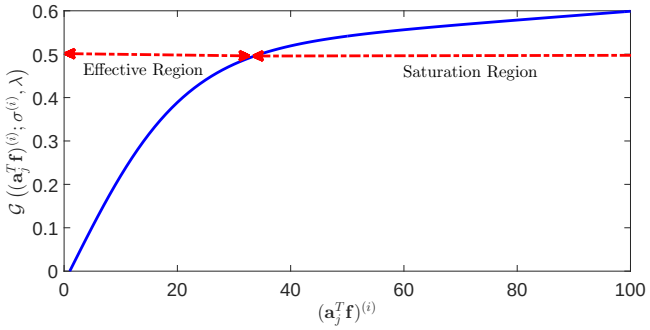


Figure 3.1: The function $\mathcal{G}(\cdot)$, which is a shifted logistic function with an additional linear term, where the parameters $\sigma^{(i)}$ and λ are set to the typical values 0.1 and 0.001, resp.

following optimization problem:

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} \mathcal{U}_1(\mathbf{f}^{(i)}) \\ &\max_{\mathbf{f}^{(i)}} \mathcal{U}_2(\mathbf{f}^{(i)}) \\ \text{s.t.} \begin{cases} f_m^{(i)} \in \{0,1\}, \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T \mathbf{f}^{(i)} \leq \alpha, \end{cases} \end{aligned} \tag{3.2}$$

Similar to the case of LFS, some constraints are considered in (3.2). Since $\mathbf{f}^{(i)}$ is an indicator vector, the problem variables are either 0 or 1. Since there must be at least one active feature, the null indicator vector is discarded, i.e. $1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)}$ where $\mathbf{1}$ is an M dimensional vector whose elements are all 1. Furthermore, we would like to set an upper bound on the number of selected features using a user-settable constant parameter α, hence the constraint $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha$ is also included.

We note that the distance measure $(\mathbf{a}_j^T \mathbf{f})^{(i)}$ is transformed by the modified logistic function $\mathcal{G}$ (see Fig. 3.1), which for the purposes of this study, is defined as

$$\mathcal{G}\left(z; \sigma, \lambda\right) \quad = \quad \frac{1}{1 + \exp\left(-\sigma z\right)} - 0.5 + \lambda z. \tag{3.3}$$

Since optimization algorithms in general are gradient-driven, changes in variables at the next iteration depend on the gradients at the current iteration. As explained later, λ is set to a small value, so the linear term in (3.3) may be neglected for the time being. In this case, the gradient of the logistic function for the large–distance samples in (3.1a) and (3.1b), (i.e., those in the saturation region shown in Fig. 3.1) have a small value and hence do not contribute significantly to changes in the $\mathcal{U}_1$ and $\mathcal{U}_2$ at the next iteration. On the other hand, terms for which the quantity $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ has a small–to–medium value (i.e., for a point in the effective region), we note that $\mathcal{G}$ in these cases is approximately linear. Since the large–distance terms can be neglected, the optimization problem of (3.2) thus becomes approximately equivalent to

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} \tfrac{1}{n-1} \sum_{j\in RoL,\, y^{(j)}=y^{(i)}} (\mathbf{a}_j^T\mathbf{f})^{(i)} \\ &\max_{\mathbf{f}^{(i)}} \tfrac{1}{N-n} \sum_{j\in RoL,\, y^{(j)}\neq y^{(i)}} (\mathbf{a}_j^T\mathbf{f})^{(i)} \\ &\text{s.t.} \begin{cases} f_m^{(i)} \in \{0,1\}, \quad m=1,\ldots,M \\ 1 \le \mathbf{1}^T\mathbf{f}^{(i)} \le \alpha, \end{cases} \end{aligned} \tag{3.4}$$

which corresponds directly to satisfying goals 1 and 2 as desired. The set of sample points for which $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ is in the effective region of $\mathcal{G}$ are considered as the *region of locality (RoL)* of the point $\mathbf{x}_p^{(i)}$.

Therefore, within $\mathcal{S}^{(i)}$, through the objective functions of (3.2), the large–distance samples have little effect on the selection of $\mathbf{f}^{(i)}$, whereas the small–distance samples have a stronger effect on the selection of $\mathbf{f}^{(i)}$. Therefore, the purpose of transforming the distance measure $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ by $\mathcal{G}(\cdot)$ is to influence the choice of $\mathbf{f}^{(i)}$ by "focusing" the objective functions on samples that are close to $\mathbf{x}_p^{(i)}$; i.e., to encourage localization

in the feature selection process.

The existence of the linear term in (3.3) introduces a (small) gradient in the objective functions with respect to $\mathbf{f}^{(i)}$. This is so that potentially relevant samples that are far from $\mathbf{x}_p^{(i)}$ at a current iteration of the optimization process have the potential to become close to $\mathbf{x}_p^{(i)}$ in an appropriate co-ordinate system in subsequent iterations.

Note that to measure the distance between two samples in the original space, other standard definitions (e.g. Euclidean distance) may also be used. However, for the purpose of this study, following (Sun *et al.*, 2010), we use the ℓ_1 distance because it provides a linear combination of the feature-wise distances (with no transformation) which preserves the logistic function behavior with respect to each elemental distance measure.

3.2.2 Optimization process

The optimization problem posed by (3.2) is a discrete binary program and hence is computationally intractable (Boyd and Vandenberghe, 2004). As is discussed in Section 2.2.1, a standard and widely–accepted way to alleviate this difficulty is relaxation of the binary variables, i.e. replacing $f_m^{(i)} \in \{0, 1\}$ with $f_m^{(i)} \in [0, 1]$ $m = 1, \ldots, M$, followed by a randomized rounding process (Thai, 2013; Souza, 2001; Boyd and Vandenberghe, 2004). Here, the notation $[\cdot]$ denotes a continuous interval, whereas $\{\cdot\}$ denotes a binary set, as before.

The optimization problem defined in (3.2) is a multi-objective optimization problem. In a similar manner to the LFS method discussed in Section 2.2.1, the individual objective functions are combined using the concept of the ϵ-constraint (Coleman

et al., 1999) as shown in (3.5):

$$\min_{\mathbf{f}^{(i)}} \mathcal{U}_1(\mathbf{f}^{(i)})$$
$$\text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T \mathbf{f}^{(i)} \leq \alpha \\ \mathcal{U}_2(\mathbf{f}^{(i)}) \geq \epsilon^{(i)} \end{cases} \tag{3.5}$$

Here, the inter–class distance measure (relating to $\mathcal{U}_2$ in (3.1b)) becomes a constraint, and is forced to be greater than a parameter $\epsilon^{(i)}$. In this way, we can map out the entire Pareto optimal set by varying this single parameter. This procedure guarantees that the transformed inter-class distances are in excess of the value of $\epsilon^{(i)}$.

We must determine the parameter $\epsilon^{(i)}$ such that the feature selection problem defined in (3.5) is feasible. (3.5) is feasible if its constraint set is non–empty. In the following we present an effective approach to specify a value for $\epsilon^{(i)}$ that guarantees feasibilty.

Similar to the LFS formulations discussed in Section 2.2.1, the optimum point must be inside the intersection of an M-dimensional unit hyper-cube defined by $f_m^{(i)} \in [0,1], m = 1, \ldots, M$ and the space bounded by the two parallel hyper-planes $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} = 1$ and $\mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} = \alpha$. This intersection defines a non-empty polyhedron $\mathcal{P}$. For an illustration of the geometry of $\mathcal{P}$, see Fig. 2.1.

The maximum feasible value $\epsilon_{max}^{(i)}$ of $\epsilon^{(i)}$ is determined by solving the maximum value of $\mathcal{U}_2$ over $\mathcal{P}$. This is equivalent to finding the extreme Pareto optimal point where the weighting assigned to the within-class distance term, i.e. $\mathcal{U}_1$, is zero. Hence,

$\epsilon_{max}^{(i)}$ is the solution to the feasibility problem defined in (3.6):

$$\epsilon_{max}^{(i)} = \max_{\mathbf{f}^{(i)}} \mathcal{U}_2(\mathbf{f}^{(i)})$$
$$\text{s.t.} \begin{cases} 0 \leq f_m^{(i)} \leq 1, \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T \mathbf{f}^{(i)} \leq \alpha \end{cases} \tag{3.6}$$

Finally, the parameter $\epsilon^{(i)}$ in (3.5) is replaced with the value $\beta\epsilon_{max}^{(i)}$, where $0 \leq \beta \leq 1$. In this way, the feature selection problem is always feasible and the entire Pareto optimal set corresponding to different relative weightings of the objective functions (3.1a) and (3.1b) can be mapped out through variation of β. In the following, the Pareto point corresponding to a specific value of β is defined as $\mathbf{f}_\beta^{(i)}$ where $\mathbf{f}_\beta^{(i)} = (f_{1,\beta}^{(i)}, f_{2,\beta}^{(i)}, \ldots, f_{M,\beta}^{(i)})^\mathsf{T}$; therefore, the complete Pareto optimal set is defined as $\left\{\mathbf{f}_\beta^{(i)}\right\}_{\beta\in[0,1]}$. The problem of interest now becomes:

$$\min_{\mathbf{f}_\beta^{(i)}} \mathcal{U}_1(\mathbf{f}_\beta^{(i)})$$
$$\text{s.t.} \begin{cases} f_{m,\beta}^{(i)} \in [0,1], \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T \mathbf{f}_\beta^{(i)} \leq \alpha \\ \mathcal{U}_2(\mathbf{f}_\beta^{(i)}) \geq \beta\epsilon_{max}^{(i)}. \end{cases} \tag{3.7}$$

The optimum point obtained from solving (3.7) defines the relaxed solution such that each element of $\mathbf{f}_\beta^{(i)}$ exists in the continuous range $[0, 1]$. However, the final (binary) solution $\mathbf{f}_\beta^{*(i)}$ must be over the discrete set $\{0, 1\}$ as in (3.2); i.e., the solution $\mathbf{f}_\beta^{(i)}$ to (3.7) must be snapped onto a binary grid. This procedure is performed by applying a randomized rounding process to $\mathbf{f}_\beta^{(i)}$, as discussed in Section 2.2.1, so that the mth

element is set to 1 (active) with probability $f_{m,\beta}^{(i)}$ and is set to zero (inactive) with probability $(1 - f_{m,\beta}^{(i)})$ where $m = 1, \ldots, M$. We repeat the randomized rounding process a thousand times. The choice for the binary optimum vector $\mathbf{f}_{\beta}^{*^{(i)}}$ is the one which provides the minimum value for the objective function of (3.7), as well as satisfying all constraints.

The final value $\mathbf{f}^{*^{(i)}}$, corresponding to the best value of β from the set $\left\{\mathbf{f}_{\beta}^{*^{(i)}}\right\}_{\beta\in[0,1]}$, is chosen as the one which provides the best local clustering performance of the training samples. The procedure for determining the best local clustering performance is similar to that of the LFS method discussed in Sect. 2.2.2.

Algorithm 2 presents the pseudo code of the proposed feature selection algorithm. The problem variables are initialized to uniform values that satisfy the constraint $\mathbf{1}^{\mathsf{T}}\mathbf{f}_{\beta}^{(i)} \leq \alpha$. Note that since the problem does not suffer from the presence of local minima (as discussed in Section 3.2.3), the initial point does not affect the solution, although it may affect the computational time.

3.2.3 Problem convexity

In this section we discuss the convexity property of the optimization problems defined in (3.6) and (3.7). By definition, $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ is always positive; hence the terms $\mathcal{G}\left((\mathbf{a}_j^T\mathbf{f})^{(i)}; \sigma^{(i)}, \lambda\right)$ in (3.1a) and (3.1b) are always positive. Thus the function $\mathcal{G}$ is both concave and increasing quasi-convex (see Fig. 3.1) (Boyd and Vandenberghe, 2004). Equation (3.6) defines an optimization problem whose objective function is concave, because it is the summation of $N - n$ concave functions (see (3.1b)). The constraint set is linear and hence defines a convex feasible set. Thus (3.6) is a convex problem.

Input: $\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$, α

Output: $\left\{\mathbf{f}^{\star(i)}\right\}_{i=1}^{N}$

Initialization: Set $\mathbf{f}_{\beta}^{(i)} = \frac{1}{\alpha}(1,\ldots,1)^{\mathsf{T}}$ $i = 1,\ldots,N$, $\beta \in [0,1]$; $\lambda = \frac{0.01}{\alpha}$;
for $i \leftarrow 1$ **to** N **do**
  Compute distance vectors $\mathbf{a}_j^{(i)} = \left|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right|$;
  Compute $\sigma^{(i)}$ through solving (3.8) using the initial values;
  Compute $\epsilon_{max}^{(i)}$ through solving (3.6);
  for $\beta \leftarrow 0$ **to** 1 **do**
    Compute $\mathbf{f}_{\beta}^{(i)}$ through solving (3.7);
    Randomized rounding process of $\mathbf{f}_{\beta}^{(i)}$ to obtain binary feature vector $\mathbf{f}_{\beta}^{\star(i)}$;
  end
  Set $\mathbf{f}^{\star(i)}$ equal to the member of $\left\{\mathbf{f}_{\beta}^{\star(i)}\right\}_{\beta\in[0,1]}$ which yields the best local clustering performance as explained in Section 3.2.5;
end

Algorithm 2: pseudo code of the proposed feature selection algorithm.

The objective function of (3.7) is a strictly increasing quasi-convex function since it is the summation of $n-1$ strictly increasing quasi-convex functions (see (3.1b)). The constraint set of (3.7) is convex and feasible. Therefore, (3.7) defines a quasi-convex problem with a unique global minimum. (Boyd and Vandenberghe, 2004). Since both problems have unique global optima, they have the computational advantage of not being trapped in local minima, with the solution being invariant to the initialization procedure.

3.2.4 Determination of the parameters of $\mathcal{G}(\cdot)$

We discuss a procedure for determining values of the parameters $\sigma^{(i)}$ and λ. This procedure depends on the feature values being normalized into their respective z–score values beforehand.

The value of the parameter $\sigma^{(i)}$ in $\mathcal{G}(\cdot)$ is defined such that, in the subspace defined by the initial value of $\mathbf{f}_\beta^{(i)}$ in the optimization procedure, the farthest sample from $\mathbf{x}^{(i)}$, denoted by $\varphi^{(i)}$, sits on the knee point of $\mathcal{G}(\cdot)$; hence $\sigma^{(i)}$ is the solution of (3.8):

$$\begin{aligned} &\frac{1}{1+\exp\left(-\sigma^{(i)}\varphi^{(i)}\right)} - 0.5 = 0.47, \\ &\text{where } \varphi^{(i)} = \max_{j=1:N, j\neq i} \left\{ (\mathbf{a}_j^T \mathbf{f}_\beta)^{(i)} \right\}. \end{aligned} \tag{3.8}$$

The number 0.47 above is chosen to be representative of the knee point of $\mathcal{G}(\cdot)$ (see Fig. 3.1). The intuition behind (3.8) is that no sample should fall within the saturation region during the first iteration of the optimization process, so that effectively all samples are considered by the objective function of (3.7).

The parameter λ controls the contribution of the samples that are in the saturation region (see Fig. 3.1). The addition of the linear term in (3.3) allows potentially close samples that are far form $\mathbf{x}_p^{(i)}$ in a current iteration, i.e. situated in the saturation region, to have the potential to migrate into the effective region of $\mathcal{G}(\cdot)$ in subsequent iterations. Thus we require a small gradient in the saturation region relative to the gradient in the effective region. As α grows, the slope of the effective region decreases, because elements in $\mathbf{f}_\beta^{(i)}$, and consequently $\varphi^{(i)}$, may increase; which results in a decrease of $\sigma^{(i)}$ in the solution to (3.8). Hence, as α grows, the slope of saturation region, i.e. λ, should decrease. Thus, in our experiments, the value of λ is set

heuristically according to the value $\frac{0.01}{\alpha}$. This form allows λ to vary inversely with α as required. The value 0.01 in the numerator allows the slope of the saturation region to be small enough compared with that of the effective region.

Note that the values for $\sigma^{(i)}$ and λ are set once during the initialization process of the algorithm according to the procedure just described. They are not varied further during execution. The parameter values used to produce the results shown in Sect. 3.3 were set according to this procedure and were not tuned to improve performance.

3.2.5 Class similarity measurement

Similar to the LFS method, the consequence of the logistic localized feature selection approach lLFS is that, since there is no common set of features across the sample space, conventional classifiers are inappropriate. Hence, the localized classifier proposed in Section 2.2.2 is used for the purpose of class similarity measurement. Furthermore, similar to the LFS method, performance of the lLFS algorithm is relatively invariant to an upper bound on the number of selected features (i.e. α). See Sections 2.3.1 and 2.3.2 for more details.

The process of determining an appropriate value for β, which results in the selection of a suitable point in the Pareto set, is also similar to that of the LFS method discussed at the last paragraph of Section 2.2.2.

3.3 Experimental results

3.3.1 Experimental set-up

In this section we perform several experiments on one synthetic and ten binary real-world data sets to demonstrate the effectiveness of the proposed feature selection algorithm. As is discussed in Section 2.4, the focus of this study is on the challenging problems where a small number of samples are available for training.

The proposed algorithm is compared with LFS and eight other state-of-the-art feature selection algorithms Logo, FMS, MBEGA, Elasticnet, kPLS, MetaDistance, DEFS and mRMR as are described in Section 2.4.1. The parameters of all algorithms are set to the default values. The SVM classifier with an RBF kernel, as is described in Section 2.4.1, is used as the required classifier for the global feature selection algorithms.

The proposed algorithm is implemented in MATLAB on a computer with an Intel(R) Core i7-2600 CPU @ 3.4 GHz and 16 GB RAM.

3.3.2 Data sets

The synthetic, or "toy" data set, as is shown in Fig. 3.2, is distributed in a two dimensional feature space defined by x_1 and x_2 in which class Y_1 has two disjoint sub-classes shown by ■ and ◄, whereas samples of class Y_2, shown by ♦, have a uni-modal distribution. Samples of each subclass are drawn from unit variance Normal distributions. In order to test the capability of the proposed lLFS method to identify only the relevant features x_1 and x_2, following (Wang, 2008), each sample is artificially contaminated by augmenting it with 100 *iid* irrelevant features drawn from a standard

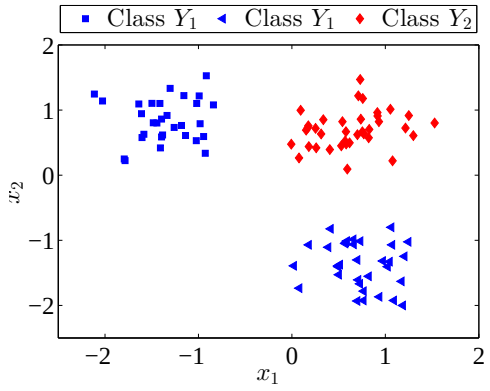


Figure 3.2: Illustration of the synthetic data set in terms of its relevant features x_1 and x_2.

Normal distribution.

The real world data sets used in our experiments are the same as what we used in Section 2.4.2 described in Table 2.1. In the case of microarray data sets "Prostate", "Duke-breast", "Leukemia", "Colon" and "Nervous system", to speed up the simulations, for the lLFS method only, we prune to 300 features beforehand. This will only have the effect of slightly degrading of performance of the proposed algorithm. In this study "Logo" (Sun *et al.*, 2010) is used for pruning, although other approaches may be used.

Each feature variable in the synthetic data set and the real-world data sets have been transformed beforehand to their z-score values.

3.3.3 Accuracy of classification

In this section, classification performance of the proposed lLFS algorithm is compared with LFS and the eight state-of-the-art feature selection algorithms indicated in Section 3.3.1.

In our experiments, similar to Section 2.4.3, the number of selected features t in

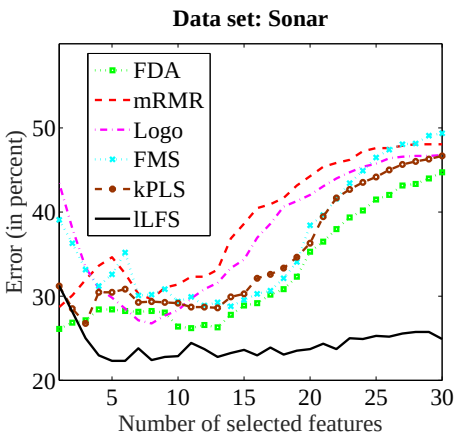
our comparison feature selection algorithms and the parameter α of the LFS and the lLFS algorithm (which is analogous to the parameter t) ranges from 1 to 30 for data sets "Sonar", "DNA", "Breast", "Prostate", "Duke-breast", "Leukemia" and "Colon", 1 to 60 for data set "Adult", 1 to 100 for data set ARR and 1 to 35 for data set "Nervous system", since there is no performance improvement for our comparison algorithms for larger values. For each data set, a bootstrapping algorithm is used to evaluate the feature selection algorithms' performance, as is described in Section 2.4.3.

The minimum classification error rate, the corresponding standard deviation and the number of selected features t (α), for all the eight global algorithms and the LFS method on each data set, is reported in Table 3.1 where, for each data set, the best result over all the ten algorithms is shown in bold. The average of the classification error rates over all the ten data sets is shown in the last row of the table. In order to demonstrate the necessity for feature selection, we also report the classification error rate which results from applying the SVM classifier with an RBF kernel on each data set without prior feature selection. These results, shown in the last column of Table 3.1, are significantly degraded with respect to the case when feature selection is used, and thus demonstrate that the feature selection process is indeed an important component of the data classification process.

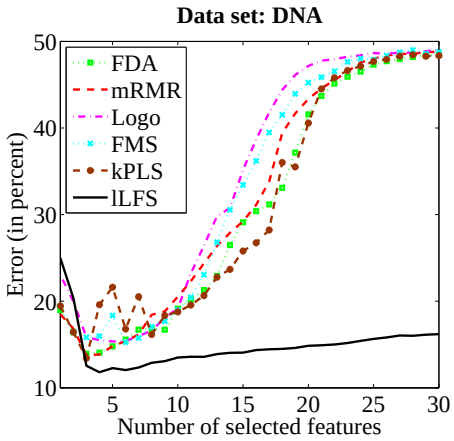
Table 3.1: Minimum classification error (in percent) of the different algorithms. The corresponding standard deviation (in percent) and t (α) are respectively reported in parenthesis. The last column corresponds to the classification results using SVM with no feature selection.

Data set	**lLFS**	**LFS**	**Logo**	**FMS**	**MBEGA**	**Elasticnet**	**kPLS**	**MetaDist**	**DEFS**	**mRMR**	**SVM**
Sonar	**22.3**(3.4,5)	22.9(3.9,30)	26.8(3.4,8)	28.8(2.6,14)	29.4(8.0,2)	27.7(4.2,5)	26.8(6.3,3)	28.9(9.9,12)	27.8(6.7,8)	28.7(2.6,1)	49.9(4.8)
DNA	**11.8**(1.8,4)	13.4(1.9,15)	15.3(5.7,5)	15.3(1.8,6)	18.0(4.7,4)	16.1(4.7,3)	13.4(2.5,3)	27.0(10.7,6)	18.7(5.0,3)	13.8(3.0,3)	49.7(2.0)
Breast	**6.2**(1.4,17)	6.4(1.3,11)	8.3(1.4,7)	7.7(1.4,9)	9.1(1.5,18)	8.8(1.5,3)	8.2(1.6,5)	12.9(6.0,9)	11.0(2.5,8)	8.3(2.2,4)	37.6(0.6)
Adult	**20.6**(1.6,19)	22.3(1.5,30)	24.5(1.9,8)	24.7(0.3,46)	24.5(0.7,26)	24.6(0.5,19)	24.7(0.3,35)	24.3(1.0,9)	24.7(0.3,28)	24.8(0.3,30)	24.7(0.3)
ARR	**27.6**(3.0,23)	33.1(2.6,29)	33.9(5.3,8)	32.2(2.9,34)	31.8(7.4,18)	38.7(3.6,9)	40.0(7.0,6)	40.7(5.7,80)	31.4(4.7,7)	31.6(3.3,10)	43.7(1.2)
Prostate	**4.2**(4.4,6)	**4.2**(4.4,6)	8.3(7.9,3)	6.7(6.6,11)	7.5(8.3,18)	7.5(6.1,8)	6.7(8.6,2)	40.0(11.0,72)	13.7(9.6,4)	8.3(7.6,7)	57.5(10.7)
Duke-breast	**7.5**(8.3,27)	10.8(7.9,3)	21.7(11.9,7)	24.2(13.3,4)	21.7(14.8,14)	32.5(14.4,11)	20.8(9.0,8)	38.3(19.7,10)	26.7(14.6,3)	21.7(5.8,5)	63.3(10.5)
Leukemia	**2.5**(4.0,16)	3.3(4.3,30)	6.7(5.3,2)	**2.5**(4.0,2)	8.3(6.8,26)	6.7(5.3,3)	3.3(4.3,3)	26.7(8.6,18)	16.8(10.9,4)	5.0(5.8,8)	35.8(14.2)
Colon	**9.2**(0.1,21)	**9.2**(0.1,21)	20.8(10.6,2)	13.3(9.0,6)	20.8(4.4,16)	15.0(11.0,3)	19.2(13.1,5)	25.0(6.8,3)	26.7(14.1,2)	19.2(5.6,4)	36.7(17.2)
Nervous sys.	**26.7**(9.5,4)	**26.7**(9.5,4)	33.3(14.2,9)	35.0(20.3,20)	33.3(8.8,14)	35.0(14.0,12)	31.7(18.3,15)	30.0(9.0,7)	32.5(17.8,12)	32.5(16.4,2)	37.5(16.8)
Average	**13.8**	15.2	20.0	19.0	20.5	21.3	19.5	29.4	23.0	19.4	43.6

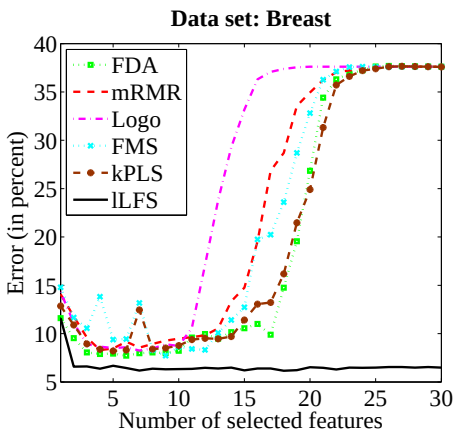
Furthermore, for each data set, the classification error rate versus the number of selected features (i.e. t (α)) for the lLFS method and the top half of our comparison global feature selection algorithms, that show the best performance on the basis of the last row of Table 3.1, are shown in Fig. 3.3. This figure besides the results reported in Table 3.1 show that the classification accuracy of the proposed lLFS algorithm is significantly improved relative to the other methods considered, and provides the lowest error rate over all data sets.


Data set: Sonar
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

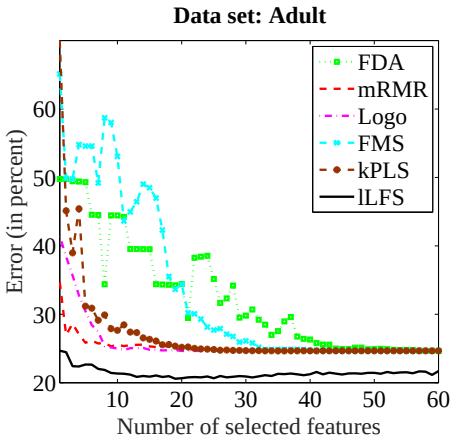
(a)


Data set: DNA
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

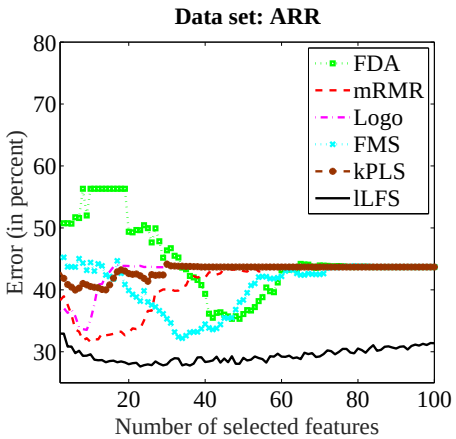
(b)


Data set: Breast
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

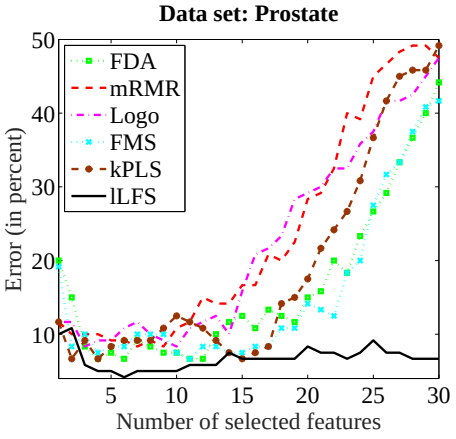
(c)


Data set: Adult
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

(d)


Data set: ARR
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

(e)


Data set: Prostate
FDA
mRMR
Logo
FMS
kPLS
lLFS
Error (in percent)
Number of selected features

(f)

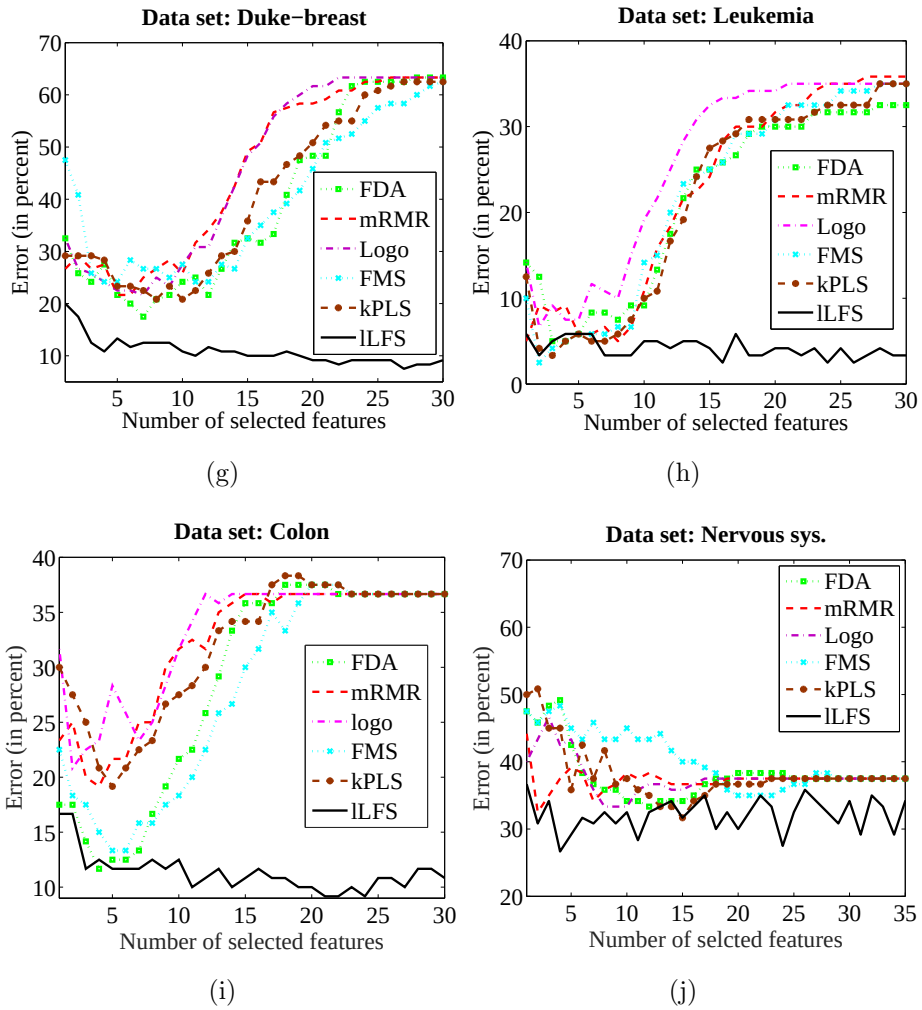


Figure 3.3: Classification error (in percent) versus number of selected features for the proposed lLFS method and the top 5 of our comparison feature selection algorithms over all 10 real world data sets.

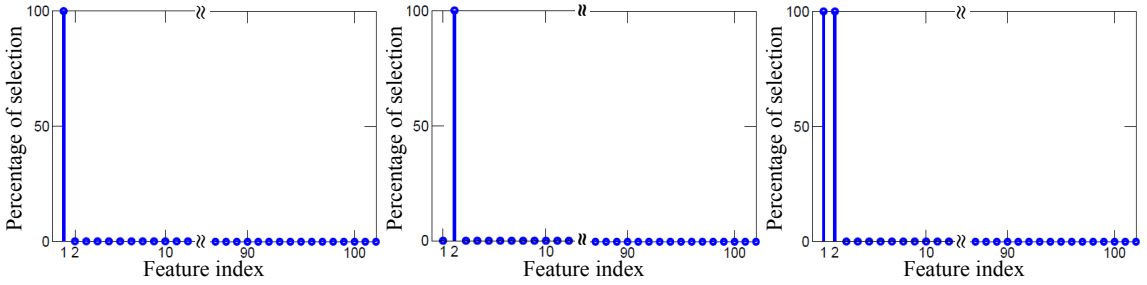
In addition, in order to demonstrate that the improved relative performance of the lLFS method is not just a reflection of the performance of the SVM classifier, we perform an additional set of classification experiments using two alternative classifiers: logistic regression and Adaboost (with a decision tree as a weak learner) (McCullagh and Nelder, 1989; Freund and Schapire, 1997). These classifiers are used in conjunction with the top half of our comparison feature selection algorithms that show the best performance on the basis of Table 3.1. The average of the minimum classification errors (using these classifiers) over all ten data sets are presented in Table 3.2. We see that the improved performance of the lLFS method persists in this case also.

Table 3.2: Minimum classificatio error (in percent) of the top half comparison feature selection algorithms using two alternative classifiers: Adaboost (first value) and Logistic regression (second value).

Data set	**Logo**	**FMS**	**kPLS**	**mRMR**
Sonar	30.8,30.2	26.7,32.2	27.4,29.6	28.7,31.8
DNA	15.3,14.2	15.8,14.5	13.6,13.4	14.4,13.9
Breast	9.0,7.3	7.7,7.4	7.3,7.0	7.7,7.2
Adult	24.0,23.6	24.0,23.5	24.7,24.7	22.9,22.3
ARR	36.7,35.4	35.9,35.6	37.8,36.8	34.4,31.8
Prostate	9.2,5.8	10.0,9.2	6.7,8.3	9.2,9.2
Duke-breast	20.8,15.8	20.0,25.8	20.0,22.5	13.3,24.2
Leukemia	4.2,5.0	2.5,4.2	3.3,4.2	2.5,3.3
Colon	15.8,17.5	17.5,17.5	15.8,15.8	16.7,14.2
Nervous sys.	36.7,35.8	33.3,40.0	32.5,38.3	38.3,34.2
Average	20.2,19.1	19.3,21.0	18.9,20.1	18.8,19.2

3.3.4 Relevant feature identification

In the following, we demonstrate the performance of the proposed method in identifying relevant features using the synthetic data set and the data set "DNA", for which there is a "ground truth".



(a) Class Y_1, subclass ■ (b) Class Y_1, subclass ◄ (c) Class Y_2 ♦

Figure 3.4: Selected features for the synthetic data set. The height of each feature index indicates what percentage of the representative points in (a) subclass ■ of class Y_1, (b) subclass ◄ of class Y_1 and (c) class Y_2 shown by ♦ select the respective feature as a member of their optimal feature subset, where α is set to 2.

The synthetic, or "toy" data set shown in Fig. 3.2 is included for the sole purpose of demonstrating that the proposed method is capable of identifying relevant and distinct feature sets in the presence of a large number of contaminating features, in a disjoint data space. We see that samples of class ♦ require both relevant features x_1 and x_2 to be discriminated from class Y_1, whereas samples of subclass ■ require only x_1 and samples of subclass ◄ require only x_2. Fig. 3.4 shows the performance of the proposed local feature selection algorithm on the synthetic data set. For each subclass, the height of each feature index indicates what percentage of the samples within that subclass selects the respective feature. As can be seen, the lLFS method has perfect performance in selecting feature x_1 for subclass ■, feature x_2 for subclass ◄ and features $\{x_1, x_2\}$ for class ♦, as well as perfectly discarding all irrelevant features indexed from 3 to 102. Note that the sample distribution is unknown at the problem outset, due to the contamination by the hundred irrelevant features. This "toy" example demonstrates the ability of the lLFS method to select a feature set that optimally adapts to local variations in the sample space.

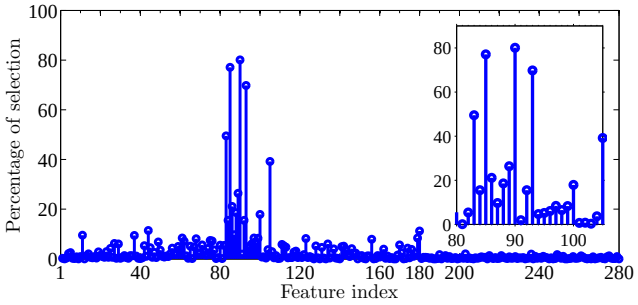
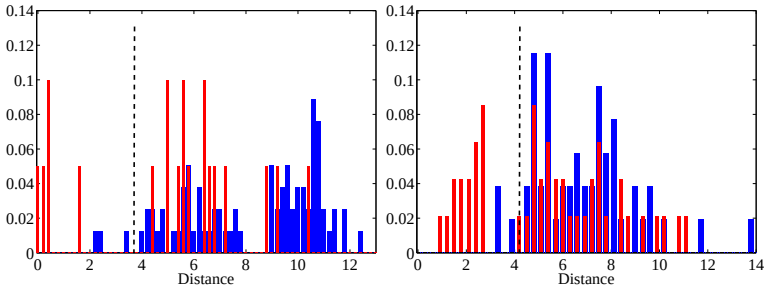


Figure 3.5: Selected features for "DNA" data set. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 10.

As mentioned earlier in section 2.4.4, the data set "DNA" is generally used for detecting the "presence" or "absence" of a splice junction in a given deoxyribonucleic acid (DNA) sequence (Wang, 2008). It has been previously shown that improved performance in most cases is observed if the attributes closest to the junctions are used (John, 1994; Wang, 2008). These attributes correspond to features indexed from 61 to 120. We therefore have a good idea beforehand what the good features are, and thus have an available "ground truth" for this example. The result of applying the proposed method on the data set "DNA" is shown in Fig. 3.5, where the height of each feature index indicates the percentage of representative points that select the respective feature as a member of their optimal discriminative sub-feature set. This figure demonstrates that the lLFS method mostly selects attributes indexed from 80 to 105, that are well matched to the "ground truth", as well as discarding the artificially added irrelevant features, which are indexed from 181 to 280.



(a) (b)

Figure 3.6: Distribution of samples around a typical representative point of (a) "Adult" data set and (b) "ARR" data set. In each case, the normalized histogram of within-class distances from the respective representative point is shown in red, and that for between-class distances is in blue. The dashed black line indicates the value of the radius of the respective $\mathcal{Q}^{(i)}$, for the specified level of impurity $\gamma = 0.2$.

3.3.5 Validation of the localized feature selection concept

In this section, we present two examples which demonstrate the efficacy of this concept. In the first example, we show that the distribution of samples around various representative points from typical real–world data sets is not uniform, suggesting that the underlying statistical behaviour varies from one region to the next. In the second example, we show that the optimal selected features vary considerably over the representative regions. These two examples validate the motivation for the localized approach, at least in these cases.

Clustering around representative points

To demonstrate the performance of the proposed algorithm in forming a within-class cluster around representative points, the distribution of sample distances from two typical representative points, selected respectively from the data sets "Adult" and "ARR", are shown in Fig. 3.6. Here the normalized histogram of within-class samples

is shown in red and between-class samples in blue. The height of each bar in the red (blue) histogram indicates what proportion of the within-class (between-class) samples corresponds to the respective distance from the representative point. All distances are computed in the respective induced feature subspace. As may be seen, there is a cluster of within–class samples, where the distances from the corresponding representative point are relatively small. This group forms the desired cluster. We note that the inter-class samples are distributed further from the representative point, as desired. Fig. 3.6 illustrates an important concept related to lLFS, in that only the *localized* clustering behavior is significant, and so not all within–class samples are required to lie close to the respective representative point. In this respect, it is interesting to note that in both cases in the figure, there is a second cluster of within–class samples (outside the $\mathcal{Q}^{(i)}$ radius). However, in this case, unlike that of the close–in cluster, we see these samples are heavily contaminated with between–class samples. So in this far–away region, the feature space corresponding to the representative point is not appropriate for separating the classes and that a different set of coordinates may be more effective in this case. Thus, we see this example provides an instance which shows how an adaptive feature selection scheme has potential for improved performance over one which uses a common set of features.

Overlap of the optimal feature subsets

To what extent do the selected features vary over the representative regions? To address this question, in Fig. 3.7 we show the normalized histogram of the selected features over all feature subsets for the data set "Duke-breast", where the parameter α is set to a typical value of 10. The height of each feature index indicates what

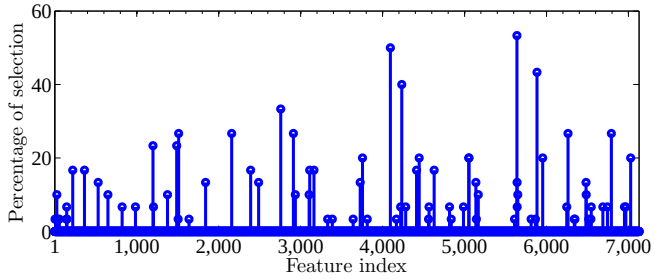


Figure 3.7: Histogram of selected features for "Duke-breast" data set. The height of each feature index indicates what percentage of representative points select the respective feature as a member of their optimal sub feature set. The parameter α is set to the typical value of 10.

percentage of the representative points select that respective feature as a member of their selected feature subset. We see the selected feature set indeed varies over the set of available training samples, as a consequence of the adaptability property of the lLFS method. As expected, the optimal feature subsets overlap to some extent, but it is also evident that there is no common feature subset that pervades over all regions. This experiment demonstrates that in typical problems there exist a large number of common features that are selected by a significant number of representative points, and a less common set of features that are informative, but only for some small sub–populations of the sample space. The most commonly selected features perform most of the discrimination task, and therefore provide a form of "interpretability" of the features. However, the less common features are still important, in that they can provide "specialized" information relevant to discrimination, but only over the small sub-populations. It is clear that the ability to offer this specialized information cannot be afforded with a method employing a global feature set.

The reader may also be interested to know what would be the classification accuracy if the top 10 dominant features, i.e. most informative features, are selected as global features and fed into the SVM classifier with an RBF kernel. The classification

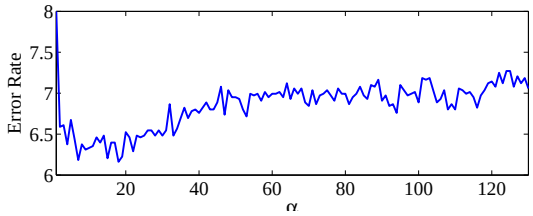


Figure 3.8: Classification error rate (in percent) of the proposed method for the data set "Breast" where the parameter α ranges from 1 to the maximum possible value of M, i.e. 130.

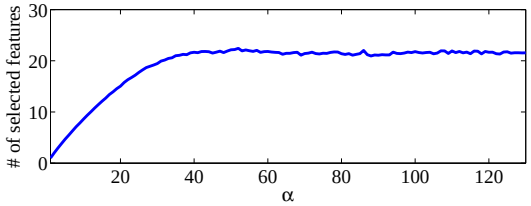


Figure 3.9: Averaged number of active features in the optimal feature sets $\mathbf{f}^{*^{(i)}} i = 1, \ldots, N$ versus the parameter α. α ranges from 1 to the maximum possible value of $M = 130$.

error rate using such a sub–feature set is 18.33%, which is in the range of the error rate of our comparison algorithms, but nevertheless is significantly greater than the 7.5% error rate corresponding to the proposed algorithm, as presented in Table 3.1 for the Duke–Breast data set. This result illustrates the effectiveness of including the less–common features for this case, and hence gives an example of the advantage of an adaptable feature selection approach.

3.3.6 Sensitivity to the parameter α

With this example, we provide a demonstration of the property of the proposed method where the selected number of features tends to saturate at a value corresponding to the number of relevant features for the respective region, as previously discussed in Sect. 2.3.2. To demonstrate this point, the classification error rate of the proposed method and the number of selected features (averaged over all N feature

sets) for the data set "Breast", for all possible values of α (i.e. $1 \leq \alpha \leq 130$) are shown in Fig. 3.8 and Fig. 3.9, respectively. The saturation effect is clearly evident from the figures. The saturation value can be obtained by examining the behavior for a sufficiently large value of α; for example, in the case of the data set "Breast", as can be seen in Fig. 3.9, the saturation value is 21. This value is the maximum number of features that each local region may require.

3.3.7 How far is the binary solution from the relaxed one?

To demonstrate that the relaxed solutions are proper approximations of the final binary solutions obtained from the randomized rounding process explained in Section 3.2.2, the normalized distribution of the distances of binary elements from the corresponding relaxed elements for data set "Duke-breast" is shown in Fig.3.10. The height of each bar indicates what percentage of elements have the corresponding value as the distance between their binary solution and the linear approximation. This result demonstrates that the relaxed solutions are appropriate approximations of the binary solutions.

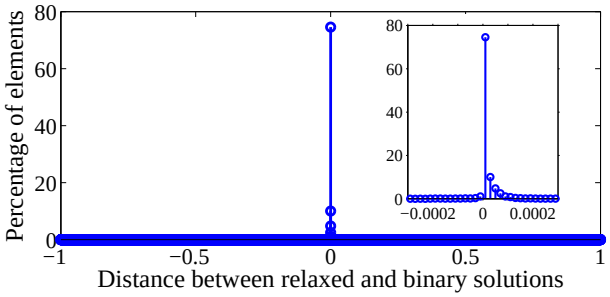


Figure 3.10: The normalized histogram of distances of binary elements from the corresponding relaxed elements for data set "Duke-breast". The parameter α is set to the typical value of 10.

3.3.8 lLFS with large number of irrelevant features

A reader may be interested to see performance of the proposed lLFS method in selecting relevant features in the presence of thousands of irrelevant features. To this end, performance of the lLFS method on the real world data set "DNA" (that its "ground truth" is defined in Section 3.3.4) is shown in Fig. 3.11 where samples of "DNA" are contaminated with 10^5 *iid* irrelevant features. As is shown, after feature selection, the lLFS algorithm correctly select attributes indexed from 80 to 105 that are well matched to the "ground truth", as well as discarding the artificially added irrelevant features indexed from 181 to 100180. This experiment besides the results reported in Table 3.1 confirms the performance of the proposed method for identification of the relevant features in the presence of thousands of irrelevant features.

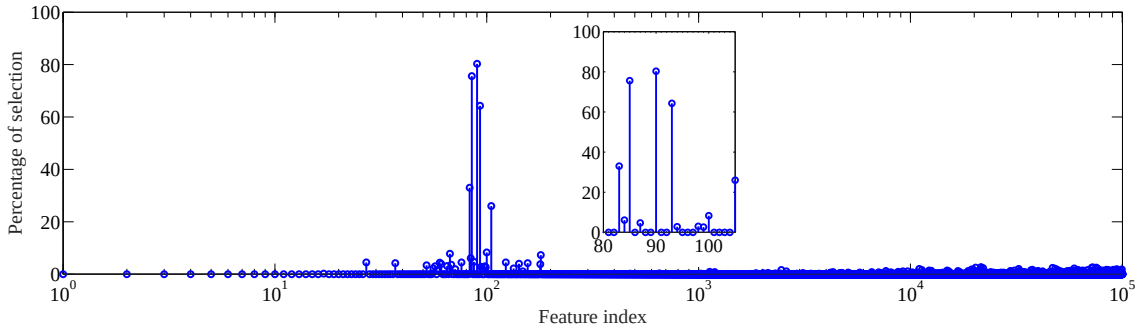


Figure 3.11: Selected features for "DNA" data set where each sample is augmented with 10^5 *iid* irrelevant features. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 10.

3.3.9 CPU time

The required CPU time for computing optimal feature subsets for lLFS and our comparison feature selection methods are presented in Table 3.3. Note that these

algorithms are implemented in different programming languages (i.e. c, Matlab and Java) therefore their time is not comparable.

lLFS method is implemented in MATLAB where we use the package "fmincon" for solving the convex and quasi-convex optimization problems defined in (3.6) and (3.7). Because the proposed lLFS method can be parallelized, depending on the available number of CPU cores, the required CPU time lies between the two extremes reported in the second column of Table 3.3. If a computer has K cores, the CPU time of the lLFS method will be 1/K of the upper extreme– therefore the lower extreme corresponds to the case where N cores are available (i.e. the required time for computing the optimal feature subset for a representative point) and the upper extreme corresponds to the case where there is no parallelization (i.e. N times the lower extreme). For example, since the personal computer used in this study has 8 cores, the required CPU time is 1/8 of the upper extreme values. Note that these computation times could be substantially further reduced by executing the algorithm in a faster language such as C. Note further that, the feature selection process is performed in the training phase, which is off-line and we are not making any claim that the lLFS competes with regard to speed in training. On the other hand, the more critical on-line test phase, i.e. classification of a query datum, is performed much more quickly, once training is complete – the average test phase time over the data sets employed in this study is 6 ms. This is because the classification process requires no optimization and only involves testing whether the query datum is contained within the specified hyper-spheres, and determining the class label of its nearest neighbors.

Table 3.3: CPU time (sec.) taken for feature selection by different algorithms.

Data set	**lLFS**	**Logo**	**FMS**	**MBEGA**	**Elasticnet**	**kPLS**	**MetaDist**	**DEFS**	**mRMR**
Sonar	[0.59,59.40]	0.28	0.05	159.35	0.05	0.06	0.08	10.01	0.10
DNA	[1.66,165.62]	0.39	0.06	116.99	0.07	0.06	0.12	10.50	0.17
Breast	[0.47,47.31]	0.24	0.05	83.83	0.04	0.06	0.12	10.08	0.08
Adult	[0.71,71.01]	0.45	0.05	286.59	0.06	0.06	1.51	10.55	0.16
ARR	[2.12,212.46]	0.35	0.06	103.40	0.07	0.06	0.21	10.11	0.23
Prostate	[1.92,172.42]	3.14	0.15	243.73	4.19	0.16	1.79	16.17	0.69
Duke-breast	[1.97,59.10]	0.45	0.08	229.96	2.68	0.16	0.43	21.45	0.82
Leukemia	[1.70,101.83]	0.94	0.15	330.77	2.02	0.14	0.74	15.82	0.90
Colon	[2.12,106.12]	0.64	0.08	44.36	0.87	0.16	0.30	25.43	0.84
Nervous sys.	[2.06,98.78]	0.76	0.09	260.87	2.15	0.14	0.60	15.90	1.05
Average	[1.53,109.41]	0.76	0.08	185.98	1.22	0.11	0.59	14.60	0.50

As is discussed in Section 2.4.1 data classification with a small number of training samples is one of the most challenging classification problems (Saeys *et al.*, 2007; Sima and Dougherty, 2006; Braga-Neto and Dougherty, 2004). As is demonstrated in Sections 3.3.3 to 3.3.8, the lLFS method is a great fit to such cases because it considers each training sample as a representative point for its neighboring region and compute an optimal feature subset for that region. However a reader may be interested to know about the required CPU time when there is a relatively large training set. To this end, the CPU time required for computing the optimal feature subset of a representative point versus the number of the training samples N is shown in Fig. 3.12 where N is increased up to 10^4. As may be seen, the figure shows linear complexity of the CPU time (for one representative point) with respect to the number of training points. Therefore, considering the fact that the lLFS method computes a feature subset for each training point, the complexity of the proposed lLFS algorithm with respect to the number of training points is $\frac{N^2}{K}$ where K is the number of available CPU cores (see Section 3.3.9).

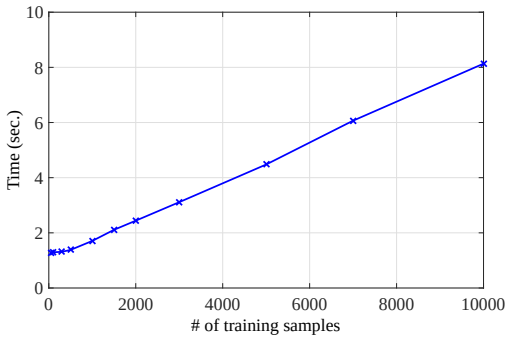


Figure 3.12: CPU time taken for computing the optimal feature subset of a representative point versus number of training samples N on a synthetic data set (with similar distribution as is illustrated in Fig. 3.2 where all three data clusters have the same number of sample points) where α is set to 2 and the data set is contaminated with 5000 irrelevant features.

3.4 lLFS vs. LFS

- As is shown in Table 3.1, lLFS has more accurate classification performance compared to LFS because lLFS defines the neighboring samples as a function of the optimal feature subset which uses a logistic function. On the other hand, the LFS algorithm defines the neighboring samples based on the distances in the original feature space. This may not be a reliable procedure in the presence of a large number of irrelevant features.

- lLFS is formulated as a non-linear joint convex/quasi convex optimization problem while LFS has been formulated as a linear programming problem, which can be solved faster. Hence, lLFS trades complexity for performance.

- Both LFS and lLFS use the localized classification approach presented in Section 2.2.2. Therefore, the finite moderate VC dimension of the localized classifier in combination with the fact that they select only relevant features, suggests both the lLFS and LFS methods are not overly sensitive to the overfitting problem.

- For both the lLFS and the LFS algorithms, the process of computing a feature subset for each representative point is independent of those of other representative points and hence can be performed in parallel.

3.5 Conclusions

In this chapter we improved the proposed localized feature selection (LFS) approach presented in Chapter 2 through alleviating the distance measurement problem related

to the neighboring samples determination. This improvement is realized by formulating the underlying optimization problem such that the distances are a function of the unknown optimal feature subset. Distances are measured using a logistic function metric within the corresponding co-ordinate system. This enables the optimization process to focus on a localized region within the sample space. The proposed logistic localized feature selection (lLFS) method is formulated as a joint convex/increasing quasi-convex optimization problem with no local minima. The localized classification approach presented in Section 2.2.2 is utilized for measuring the similarity of a new input data point to each class. The finite moderate VC dimension of the localized classifier in combination with the fact that the lLFS method selects only relevant features, suggest the lLFS method is not overly sensitive to the overfitting problem. lLFS can be performed in parallel. Experimental results demonstrate the superior performance of the lLFS over LFS and the previous state-of-the-art feature selection algorithms on a large variety of data sets.

The following chapter is a reproduction of a ready to submitted paper to IEEE Transactions on Biomedical Engineering:

Narges Armanfard, James P. Reilly, John F. Connolly "Automatic and Continous Detection of Mismatch Negativity: Application to Coma Outcome Prediction", To be submitted to IEEE Transactions on Biomedical Engineering, October 2016.

Chapter 4

Automatic and Continous Detection of Mismatch Negativity: Application to Coma Outcome Prediction

4.1 Abstract

Accurate and fast detection of event related potential (ERP) components is an unresolved issue in neuroscience and critical health care. Mismatch negativity (MMN) is a component of the ERP to a deviant stimulus in a sequence of identical stimuli that has good correlation with coma awakening. All of the previous studies for MMN detection are based on visual inspection of the averaged ERPs (over a long recording time) by a skilled clinician. However, in practical situations, such an expert may

not be available or familiar with all aspects of evoked potential methods. Further, we may miss important clinically essential events due to the implicit averaging process used to acquire the ERPs. In this chapter, using the LFS method proposed in Chapter 2, we propose a practical machine learning (ML) approach for automatic and continuous assessment of the ERPs for detecting the presence of the MMN component. The proposed method consists of two phases: learning and testing. The trained model obtained from the learning phase is used in the testing phase to assess the brain response of a test subject to a deviant stimulus to detect the presence of the MMN component. The method is capable of detection over intervals as short as two minutes. This finer time resolution enables identification of waxing and waning cycles in level of consciousness. We show evidence that suggests the existence of even short waxing periods is highly predictive of recovery. Experimental results on 25 normal and comatose subjects demonstrate the effectiveness of the proposed method for automatic and continuous assessment of ERPs for MMN detection.

4.2 Introduction

Coma is a state of prolonged unconsciousness that can be caused by a variety of problems, e.g. traumatic brain injury, stroke, brain tumor, drug or alcohol intoxication (Young *et al.*, 1998). Continuous assessment of level of consciousness as well as coma outcome prediction, acquired as reliably and as soon as possible, are important aspects of patient care. Outcome prediction is important for patients, their relatives and attendant medical staff because of the limited availability of intensive-care therapy, the demands of planning individual patient management, and the need for counseling relatives with realistic expectations. Online assessment of comatose patients is

very important because it provides us the capability to detect short increases in the level of consciousness, thus improving both outcome prediction and the rehabilitation process[1] (Kane *et al.*, 2000).

Traditional approaches for coma outcome prediction are mainly based on a set of clinical observations (e.g. asymmetry in pupillary responsiveness, dilatation and constriction, verbal and motor responses) (Marmarou *et al.*, 2007; Mushkudiani *et al.*, 2008; Lee *et al.*, 2010; Teasdale and Jennett, 1974) and electrophysiological techniques (based on the classical resting state electroencephalogram and on evoked responses to sensory stimulations) (Greenberg *et al.*, 1977; Rappaport *et al.*, 1977).

The Glasgow Coma Scale (GCS) is the most common clinical indicator that describes level of consciousness based on clinical assessment. The scale consists of 3 parts: assessment of eye opening, verbal response, and best motor response (Jones, 1979). The GCS ranges between 3 (deep unconsciousness) to 15 (best response) where the score 8 or less corresponds to the comatose state. However, treatment in the intensive-care unit, with intubation and sedation, often confounds clinical assessment such that prediction of outcome for an individual patient can be difficult (Kane *et al.*, 2000).

The role of neurophysiological methods has been reviewed in (Chiappa and Hill, 1998) where it is quite clear that electroencephalogram (EEG) and early evoked potentials allows an objective assessment and provides useful prognostic information in comatose patients. Early evoked potentials like primary somatosensory responses in the 30-ms range and brainstem auditory evoked potentials have been used for more than two decades due to their high predictive value. Brainstem auditory evoked

[1] A passive rehabilitation regimen could be feasible when suitable markers indicate higher levels of consciousness.

potentials test the function of the auditory nerve and auditory pathways in the brainstem. These are electrical responses of the auditory pathways that occur within 10 to 15 milliseconds of an appropriate acoustic stimulus (Petrova, 2009; Jewett and Williston, 1971). Somatosensory evoked potentials are generated by stimulation of afferent peripheral nerve fibers by either physiological (eg., muscle stretch) or electrical means (typically, a square wave of 0.2- to 2-millisecond duration is delivered to a peripheral nerve by electrodes) (Chawla *et al.*, 2016). However, early evoked potentials are a good predictor only for poor coma outcome (accuracy $> 98\%$ when there are no focal injuries) and their presence does not guarantee a good coma outcome (Madl *et al.*, 1996; Zandbergen *et al.*, 1998; Robinson *et al.*, 2003; Fischer *et al.*, 2006).

Recently, long latency event-related potentials (ERPs) have been introduced as useful predictors of good coma outcome (Lew *et al.*, 2006) (the potential application of long latency ERPs in clinical practice is reviewed in (Kane *et al.*, 2000)). Appropriate auditory paradigms elicit long latency ERPs even in the absence of the patient's attention, making them useful in the assessment of altered states of consciousness.

A passive oddball paradigm of demonstrated utility consists of two types of stimuli (Holeckova *et al.*, 2006): standard tones and deviant tones, where repetitive standard tones are interspersed with slightly deviant stimuli. This paradigm elicits two different long latency ERP components: N1 and mismatch negativity (MMN). The presence of N1 and MMN (elicited at respectively about 100 and 150 millisecond post-stimulus) provides evidence of basic brain function. The N1 is an obligatory sensory response evoked by each tone (i.e. both standard and deviant) and highlights the encoding of acoustic input in the auditory cortex. The MMN is an automatic response to deviants and highlights preserved automatic sensory memory processes. The presence of the

MMN demonstrates proper functioning of pre-attentive cognitive processes. These ERPs are elicited without requiring the subject's active involvement.

Clinical studies on coma patients demonstrate that the MMN has good correlation with coma awakening (Morlet and Fischer, 2014; Fischer *et al.*, 1999). The reported results show that more than 90% of patients who were considered as non-awake showed no MMN (i.e. a high specificity) and more than 90% of patients in whom MMN was detected returned to consciousness (i.e. a high positive predictive value). But only about 30% of patients who had regained consciousness showed MMN (i.e. a low sensitivity).

One of the important yet unresolved issues in the literature is accurate and fast detection of ERP components (e.g. MMN and N1). One of the major drawbacks of all previous studies is that they all require visual inspection by a skilled clinician (Morlet and Fischer, 2014) while, in practical situations, such an expert is not likely available. Another difficulty with current methods is that assessment must be performed based on the average of ERP signals over a long recording time (typicaly on the order of 30 min) (Duncan *et al.*, 2009; Morlet and Fischer, 2014), in order to reduce the effect of background EEG noise. However, in this study we show evidence that the level of consciousness of coma victims to "waxes and wanes" over durations much shorter than the interval used to average the signal. We also provide evidence that even short durations of increased level of consciousness are relevant to coma outcome prediction. Thus a significant disadvantage of using excessive averaging is that we may miss such important, clinically relevant events. The use of excessive averaging could be one of the reasons for the low sensitivity of the MMN reported in clinical studies. Therefore, automatic detection of ERP components over as short a time

frame as possible is necessary to provide the most salient clinical information on the current state and prognosis of the patient.

In this study we extend the pioneering work of (Morlet and Fischer, 2014; Fischer *et al.*, 1999) and alleviate the above difficulties associated with coma prognosis by proposing an advanced machine learning (ML) technique. This technique automatically and continuously detects, over a relatively short window of 2 minutes, whether the brain responses to deviant tones include the MMN component or only the obligatory N1 component. If the former, the patient is likely to emerge.

Machine learning (known also as data mining or pattern recognition) methods have been previously used in several EEG applications, including the analysis of EEG signals for epilepsy (Ghosh-Dastidar *et al.*, 2008; Guler and Ubeyli, 2007), in evaluating residual functional deficits following concussion (Cao *et al.*, 2008), to classify sleep stage in animals (Crisler *et al.*, 2008), for distinguishing age of infants (Ravan *et al.*, 2011), and to predict and investigate the response and effect of selective serotonin reuptake inhibitor (SSRI), and clozapine (CLZ) treatments for major depressive disorder and schizophrenia (Ravan *et al.*, 2015; Khodayari-Rostamabad *et al.*, 2013). The machine learning methodology proposed in this study employs mathematically-structured, optimization-based (i.e. based on the LFS methods) machine learning techniques.

The proposed methodology consists of two phases: learning and testing. In the former, through use of training subjects a model is trained, and in the testing phase the trained model is used to identify the presence of the MMN component in the brain response of a test subject to deviant tones. The learning phase is realized in a feature-selection/classification framework where labeled training ERP samples

are needed– labels indicate whether an ERP response has an MMN component or has only the obligatory component N1. Due to the low sensitivity of current MMN averaging methods for coma outcome prediction, determination of accurate labels for training purposes is not possible. An alternative approach to collecting labels is to have a skilled clinician visually inspect ERP responses averaged over short windows (on the order of 2 minutes). However, the cost of gathering sufficient data with such an approach is prohibitive. Therefore to deal with this problem, in this study we employ an indirect approach to training and testing where labeled training data is provided exclusively by healthy subjects. This approach is explained in more detail in Sect. 4.4.

The proposed method provides an objective facility that will significantly improve the efficacy of health care for coma victims, in that it lowers the demands for skilled personnel and thus can reduce cost. Furthermore, because the effective averaging window of the proposed method has been reduced to 2 minutes, the waxing and waning cycles of the patient can now be detected. Experimental results are presented that demonstrate this phenomenon. The accuracy and prognostic power of the MMN is now significantly improved, in that the detection of short duration waxing intervals is highly relevant to predicting outcome. Thus the proposed method can be an important aid when formulating decisions on whether to continue or terminate life support for comatose patients.

The remaining portion of this chapter is organized as follows. Section 4.3 describes the oddball paradigm used in this study and the EEG recording process. The proposed methodology is presented in Section 4.4. Experimental results and Conclusions are presented in Sections 4.5 and 4.6, respectively.

4.3 Passive oddball paradigm and EEG recording

In this study, the N1 and MMN components are elicited using a modification of a classic auditory oddball paradigm, as described in part in (Fischer *et al.*, 2008). Stimuli consist of standard tones (85%) and deviant tones (15%). These stimuli are randomly presented; however, each deviant is preceded by at least two standard tones. In this study we use a duration deviant that is one of the most robust types of "deviant" features, both for evoking the MMN but also for producing one of the most stable MMN waveforms over time (Escera *et al.*, 2000).

This passive oddball paradigm is applied to 22 healthy normal subjects and 3 comatose patients where standard and deviant stimuli are tones of 800 Hz lasting for 75 ms and 30 ms, respectively. The rise/fall is 5 ms. The stimulus onset asynchrony (SOA) of the standard and deviant tones is 610 ms; thus, any stimulus preceded by a standard or deviant tone begins 610 ms after the onset of that tone. The stimuli are presented in a session of 1880 items (with a total duration of approximately 30 minutes), of which 280 are deviants and the remaining are standards.

EEG signals of subjects under the passive oddball paradigm are recorded with a 32-channel BioSemi headcap[2] with a standard sampling rate of 512 Hz.

4.4 Proposed methodology

In this section we propose a machine learning–based algorithm for automatic and continuous assessment of a subject.

As indicated previously, it is necessary to use an indirect approach for collecting

[2]http://www.biosemi.com/headcap.htm

labels for the training data. In this study, training data is collected only from healthy subjects. Provision of labels is then implicit, since with high probability healthy subjects respond appropriately to both standard and deviant tones. The proposed machine learning algorithm is trained to discriminate between the states having only N1 vs. having N1 + MMN over the healthy training set (as seen in Sect. 4.5.1, this task is performed with 92.7% accuracy). We then use the proposed machine learning algorithm to determine the *similarity* between the standard and deviant responses of a (coma) test subject to those of the aggregate healthy subjects. If the similarities are high, then with high probability the MMN component (in addition to the N1 component) exists in the test subject. Since previous studies have verified that the presence of the MMN is highly indicative of recovery, the efficacy of the proposed indirect approach for determining prognosis is verified. A quantitative measure for determining similarity is described in the sequel.

Details of the learning and testing phases of the proposed method are respectively presented in Sections 4.4.1 and 4.4.2.

4.4.1 Learning phase

We define two classes of data where the first class Y_1 corresponds to the presence of N1 only and the second class Y_2 corresponds to the presence of the MMN component in addition to the N1. In this section, we train a feature subspace in which class Y_2 is discriminated from class Y_1 where the required training points for both classes Y_1 and Y_2 are provided from healthy brain responses to both standard tones (providing N1 alone) and deviant tones (providing N1 plus MMN) respectively.

The learning phase consists of three stages: 1) pre-processing 2) feature extraction

and 3) feature selection. In the pre-processing step, training points for both classes Y_1 and Y_2 are extracted from the healthy training subjects under the passive oddball paradigm explained in Section 4.3. An artifact removal algorithm is then applied. In the second step, each of the extracted training points is represented by a large number of candidate features. Then, in the last step, the candidate feature set is reduced to the most relevant features such that within the subspace defined by these relevant features, class Y_2 is optimally discriminated from class Y_1. The remaining portion of this section gives details of the learning phase stages.

Pre-processing

The relevant ERP components, corresponding to both standard and deviant stimuli, are contained within an interval 0 to 300 ms after the stimulus onset.

To better highlight N1 and MMN components in pathological recordings and to remove the effect of eye blink and muscle artifacts, the extracted epochs are filtered by a band-pass FIR filter from 2 Hz to 30 Hz with a filter order of 40 (Morlet and Fischer, 2014; Armanfard *et al.*, 2016c,a) . Then the epochs in which the variance of the Vertical-EOG channel exceeds 500 μv^2 or the signal peak to peak (on any electrode) exceeds 100 μv are excluded (Ravan *et al.*, 2015).

Finally, to generate reliable and stable training points, for each training subject, deartifacted epochs associated with standard and deviant tones are averaged. Consequently, each training subject provides two 32-channel training samples: 1) the averaged clean signals corresponding to standard tones (i.e. class Y_1) and 2) the averaged clean signals corresponding to deviant tones (i.e. class Y_2). These two averaged signals for a typical training subject (at channel Fz) are shown in Fig. 4.1 where,

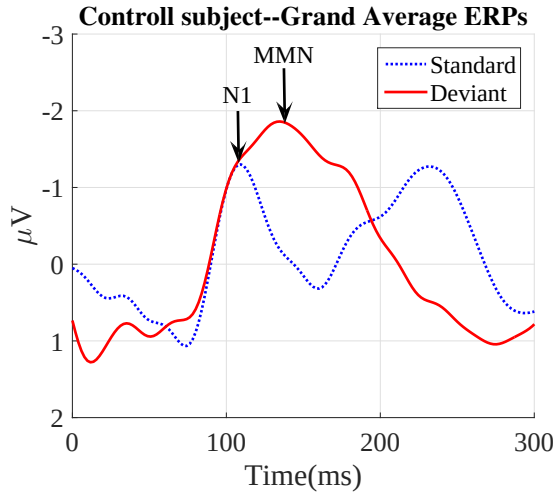


Figure 4.1: Averaged ERPs corresponding to standard (blue) and deviant (red) stimuli of a typical normal subject at channel Fz. The obligatory N1 component is elicited for both standard and deviant stimuli while the MMN occurs only for the deviant.

as expected, the standard signal has only the N1 component and the deviant signal has both the N1 and MMN components. Therefore, if there are N subjects used for training, after pre-processing, there are $2\times N$ 32-channel training samples: N training samples with class label Y_1 and N training samples with class label Y_2.

Feature extraction

In this step, each training point obtained from the previous part is represented with a large number M of candidate features. The candidate feature set consists of various statistical quantities at each channel. These quantities are kurtosis, skewness, variance, maximum, minimum and power in eight different frequency bands: Alpha-band (8Hz to 13Hz), Beta1-band (13Hz to 20 Hz), Delta-band (1Hz to 4Hz), Lower-band (1Hz to 8Hz), Total-band (1Hz to 30 Hz), Beta-band (13Hz to 30 Hz), Beta2-band

(20Hz to 30Hz) and Theta-band (4Hz to 8Hz). In addition, the wavelet decomposition vector with wavelet 'rbio6.8' [3] at level 3 is also considered (i.e. 62 features per channel). Consequently, all together, each channel is represented with 75 features.

Finally, we concatenate all channels' features and represent each training point i with an M-dimensional feature vector $\mathbf{x}^{(i)} \in \mathbb{R}^M, i = 1, \ldots, 2N$, ($N$ samples corresponding to standard responses, and N for deviant) with accompanying class label $y^{(i)} \in \{Y_1, Y_2\}$ where $M = 2400$ (i.e. 32×75). In this study, it is noted that the number of candidate features M far exceeds the number of training samples $2 \times N$.

Feature Selection

Apparently, not every extracted feature in the candidate feature set discriminates equally between the brain states (i.e. the states corresponding to classes Y_1 and Y_2). Irrelevant features may degrade the accuracy and efficiency of the similarity measurement in the test phase (Jain *et al.*, 2000). Therefore, the candidate feature set must be reduced to contain only the most relevant features. This task is performed using a feature selection process. Since all ERPs have the obligatory component N1, the relevant features are selected such that, within the corresponding induced subspace, training points in class Y_2 are optimally discriminated from those in class label Y_1.

Many feature selection approaches are presented in the literature (Peng *et al.*, 2005; Sun *et al.*, 2010; Cheng *et al.*, 2011; Duda *et al.*, 2001; Cheng *et al.*, 2011). Almost all of these methods select a *global* common feature subset for all regions of the

[3] available in MATLAB

sample space. These methods may not be appropriate for complex classification problems (such as classification of biological signals) with sparse/disjoint/irregular/non-stationary sample distributions – e.g. in cases where samples in the same class separate into multiple clusters that have different statistical characterizations that cannot be modeled by the same feature subset. The proposed localized feature selection (LFS) method presented in Chapter 2 (published in (Armanfard *et al.*, 2016b) and (Armanfard and Reilly, 2013)) addresses these concerns by allowing various groups of samples in different regions of the sample space to be associated with their own distinct optimal feature set, which may vary both in membership and size across the sample space. The LFS algorithm considers the feature vector of each training sample $\mathbf{x}^{(i)}$ as a representative point for its neighboring region, and selects an optimal distinct feature subset, indicated by the binary vector $\mathbf{f}^{(i)}$, for each of these regions. Each of these distinct feature subsets are selected such that, within the corresponding induced subspace, class Y_2 is locally discriminated from class Y_1 through a localized clustering procedure. That is, we minimize local intra–class distances and simultaneously maximize local inter–class distances. $\mathbf{f}^{(i)}$ is an M-dimensional indicator vector that indicates which candidate features are optimal for local separation of classes. If the m^{th} element of $\mathbf{f}^{(i)}$ is 1, then the m^{th} feature is selected for the ith region, otherwise it is not. With this method, no assumptions are made on the probability distribution of the samples. Therefore disjoint or multi–model distributions are accommodated. In this work, following (Armanfard *et al.*, 2016b) the Euclidean distance is used as the distance measure.

In this study, training points are extracted from different individuals; this imposes intra-class variations over the sample space due to nonstationarity. Furthermore, we

are dealing with a sparse problem where $N << M$. The LFS method has proven to be very successful for feature selection under such conditions. We therefore use this method for our experiments presented in the following Section 4.5.

4.4.2 Testing phase

In this section, the selected feature subsets, computed in the previous section, are utilized for automatic and continuous assessment of a test subject. To this end, the proposed procedure updates its detection results continuously according to what we refer to as *protocol P1*. That is, the results are updated at one–minute intervals, where each result uses ERP data extending two minutes into the past.

The testing phase consists of three stages: pre-processing, feature extraction and similarity measurement where each phase is explained in the following:

Pre-Processing

The test subject is assessed according to Protocol P1 described above. To this end, 300-ms epochs (0 to 300 ms after stimulus onset) corresponding to the standard and deviant tones are extracted. Then, as described in section 4.4.1, epochs are de-artifacted. The clean 32-channel epochs corresponding to standard and deviant tones are then averaged. Therefore, after performing the pre-processing steps, every one–minute intervals, there are two 32-channel signals, corresponding to standard and deviant tones, that are based on recorded data within the past 2 minute window.

Feature Extraction

Every one–minute interval, $M = 2400$ candidate features are extracted from each clean averaged signal (computed over the last 2-min window), as explained in Section 4.4.1. These features form two "query" M-dimensional vectors $\mathbf{x}^q_{\text{std}}(t), \mathbf{x}^q_{\text{dev}}(t) \in \mathbb{R}^M$ corresponding to standard and deviant stimuli respectively, where t denotes the interval index and the superscript q denotes "query". (In the sequel, we suppress the subscript notation "std" and "dev" and the dependence on time, for notational clarity.) In the next section, the proposed localized classifier presented in Section 2.2.2 (published in (Armanfard *et al.*, 2016b)) is employed to measure the similarity of the $\mathbf{x}^q$ to the training set.

Similarity measurement

In a similar manner described earlier in Section 2.2.2, we associate a hyper-sphere $\mathcal{Q}^{(i)}$ whose class label is $y^{(i)}$ with the ith, $i = 1, \ldots, 2N$ training point where the similarity $S_{Y_\ell}(\mathbf{x}^q; \gamma)$ of query datum $\mathbf{x}^q$ to class $Y_\ell \in \{Y_1, Y_2\}$ is measured based on how many hyper-spheres with class label Y_ℓ contain $\mathbf{x}^q$ (see (2.11)-(2.13)).

Therefore, if both the two query data over a particular window corresponding to the standard and deviant ERPs show high similarity to respectively class Y_1 and Y_2, then the brain function over that window is similar to the normal brain function (i.e. standard stimuli elicited N1 and the deviant stimuli elicited both N1 and MMN).

In this way, by using protocol P1, we can detect the presence of MMN components that are elicited over the relatively short window of 2 minutes, with one minute updates. Conventional methods require a significantly longer window (at least 30 minutes), i.e. at least 250 deviant responses to detect the presence of MMN, e.g.

(Morlet and Fischer, 2014; Fischer *et al.*, 1999).

4.5 Experiments

In Section 4.5.1, performance of the proposed methodology is demonstrated on the normal subjects. In Section 4.5.2 performance of the proposed method for automatic and continuous assessment of comatose patients is demonstrated.

4.5.1 Performance on normal subjects

The performance of the proposed methodology on normal subjects is demonstrated through a Leave-One subject-Out (LOO) cross validation strategy; one round of cross-validation involves partitioning the $N+1$ total available subjects into sets which include N training subjects and one test subject. The learning phase explained in section 4.4.1 is applied to the N training subjects and then the test phase explained in Section 4.4.2 is applied to the one remaining test subject to validate the performance of the trained model. Here, since ground truth of the test subject is known, validation is performed in a classification framework where class label y^q of a query datum $\mathbf{x}^q$ is defined as follows:

$$y^q = \underset{Y_l \in \mathcal{Y}}{\operatorname{argmax}} \{S_{Y_1}, S_{Y_2}\}. \tag{4.1}$$

where $S_{Y_\ell}, \ell = 1, 2$ is defined in Section 4.4.2.

In this study the total number of available normal subjects $N+1$ is 22; hence the number of available training points at each LOO round is 42 (i.e. $2 \times N$) where 21 points correspond to Class Y_1 and 21 points correspond to class Y_2. The LOO procedure is performed 22 times with different partitioning in each round.

Three criteria are used for performance evaluation: these are true positive rate (TPR), true negative rate (TNR) and accuracy, which are defined as follows:

$$TPR = \sum_{k=1}^{N+1} \frac{TP^{(k)}}{P^{(k)}},\ TNR = \sum_{k=1}^{N+1} \frac{TN^{(k)}}{N^{(k)}}, \tag{4.2}$$

$$Accuracy = \sum_{k=1}^{N+1} \frac{TP^{(k)} + TN^{(k)}}{P^{(k)} + N^{(k)}}, \tag{4.3}$$

where, at round k of the LOO process, $TP^{(k)}$ ($TN^{(k)}$) is the number of query data with class Y_2 (Y_1) that are correctly predicted as class Y_2 (Y_1), and $P^{(k)}$ ($N^{(k)}$) is the total number of query points with class label Y_2 (Y_1) where queries are extracted based on the P1 protocol. High accuracy with equally distributed TPR and TNR is desired.

In these experiments, the parameter α used by the LFS method, that sets an upper limit on the number of selected features, ranges from 1 to 10.

The maximum accuracy along with the corresponding TPR and TNR of the proposed methodology is presented in the second column of Table 4.1 where the default parameter values are used for the LFS method. The high prediction accuracy (92.7%) demonstrates the effectiveness of the proposed methodology for automatic and continuous assessment of a healthy subject to identify the presence of the MMN component.

Furthermore, to demonstrate better performance of the localized algorithm in comparison with the global feature selection methods, the classification performance of the proposed strategy using the top half of our comparison global feature selection algorithms is also recorded. In this case, since the localized classifier explained in section 4.4.2 is not appropriate for a global feature selection scheme, the SVM is used as a classifier with parameters set to their default values. Following our experiments

Table 4.1: Maximum Accuracy of the proposed method, along with the corresponding TPR and TNR (in percent), averaged over 22 runs, using both local and global feature selection algorithms. Standard deviations (in percent) are presented in parentheses.

	LFS	**Logo**	**FMS**	**kPLS**	**mRMR**
Accuracy	**92.7(8.1)**	87.1(8.9)	86.8(10.0)	84.9(11.4)	86.9(9.3)
TPR	92.9	80.4	83.6	78.4	79.4
TNR	92.4	93.9	90.0	91.5	94.4

in section 2.4, our comparison global feature selection algorithms are Logo (Sun *et al.*, 2010), MFA (Cheng *et al.*, 2011), kPLS (Sun *et al.*, 2014) and mRMR (Peng *et al.*, 2005) where the number of selected features also ranges from 1 to 10. The maximum accuracy along with the corresponding TPR and FNR (averaged over 22 runs) is reported in columns 3-6 of Table 4.1. The results demonstrate the effectiveness of the localized approach compared with global feature selection methods for the prediction problem defined in this study.

4.5.2 Performance on comotouse patients

In this section the proposed methodology presented and verified in Sections 4.4 and 4.5.1 is applied to our 3 comatose patients for which measurements are available. The patient's response to the auditory oddball paradigm is assessed according to P1, to establish the similarity of the patient's brain function to that of a normal brain.

Patient 1 is a 29-year old male who was involved in a motor vehicle collision. We began testing 27 days post-injury. He regained consciousness and was extubated 50 days after his admission. We recorded multiple sessions of the auditory oddball paradigm as explained in section 4.3 over a period of 48 hours to capture the modulation of the MMN as the patients level of consciousness changed. In total, 17 sessions

of data were acquired. Continuous EEG was recorded from 32 sites positioned according to the 10-20 standard. He presented with a right parietal subdural hematoma, acute traumatic subarachnoid hemorrhage and diffuse axonal injury, and scored 4 on the Glasgow Coma Scale at the initiation of data collection.

Patient 2 is a 21-year old male who was involved in a collision between an all-terrain vehicle and a tree. He was thrown from the vehicle and wasn't wearing a helmet. Recording began 13 days after his admission and concluded the next evening due to his transfer to a stepdown unit. He was GCS 7 at the time of recording. In total, 7 sessions of data were acquired. Continuous EEG was recorded from 32 sites positioned according to the 10-20 standard. He was transferred to stepdown 15 days post-injury and, after 2 days, he was GCS 10. He slowly recovered over the next couple days, showing more and more alertness, tracking, and command following. His improvement while in stepdown was slow. His discharge summary states: *His best GCS was 10/15. He remained unable to use the left side of his body, he cannot use his left leg. There was minimal movement of the right lower limb. With the right upper limb, he was able to do a thumbs up, snap his fingers and hold a ball.* He was then discharged to the rehab unit 2 months after his collision. 3 months later, he was discharged from rehab to his home. His discharge summary shows that he was no longer in a minimally conscious state, as he was walking without aids and was working on his grasp. He was reading a newspaper and answering questions about it.

Patient 3 was a 55-year old male. He was admitted after he was involved in a motor vehicle collision with multiple roll overs. He was brought in by air ambulance. His notes say he had a subarachnoid hemorrhage, and was GCS 3 at the scene. The patient was on life support for about 9 days when recording began. The family

decided to withdraw support the following day and he continued without breathing support for about five days until he died. Recordings stopped before extubation. He was GCS 4 to 5 during recording. In total, 9 sessions of data were acquired.

For all patients, recordings are conducted in sessions, where each session is about 30-min in duration, with an inter-session time of about 2 hours. In our experiments, each session consists of 17 2-min epochs, updated in 1-minute intervals, according to protocol P1. Each epoch has about 18 deviant trials and 125 standard trials.

Automatic and Online assessment

Traditional approaches visually inspect the grand average ERPs (over all the recording sessions) at a limited number of channels (typically channels Fz and Cz) to detect the presence of N1 and MMN component (Holeckova *et al.*, 2006; Morlet and Fischer, 2014). Such grand average ERPs for the case of our 3 patients at channels Fz and Cz are shown in Fig. 4.2.

It is seen from these figures that in all cases, the N1 and MMN components are not clearly discernible from either response, or if they are discernible, they are both very weak. Yet, in the case of patients 1 and 2, they recovered, suggesting these components may indeed be present in their ERP responses, although obscured. We believe this outcome is a manifestation of the poor sensitivity of the conventional MMN test to determine prognosis based on the grand average ERPs. In the case of patient 3, the N1 and MMN components are also not discernible in Fig. 4.2. However the recovery status of patient 3 is ambiguous in this case since he died a considerable time after life support was withdrawn (later we present evidence that if life support was not withdrawn, he may have recovered). Therefore we do not make claims one

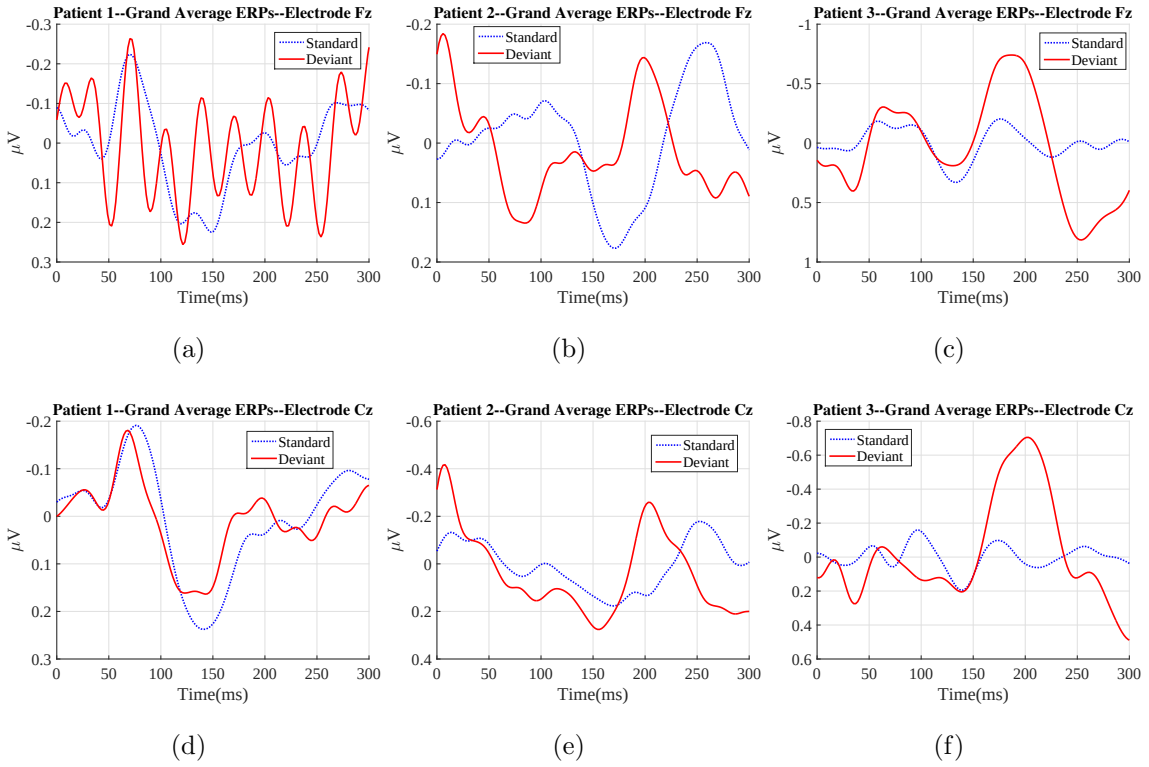


Figure 4.2: Grand average of the ERPs for patients 1-3 for the sites positioned at electrodes Fz and Cz.

way or the other as to whether the N1 and MMN components should exist in this case.

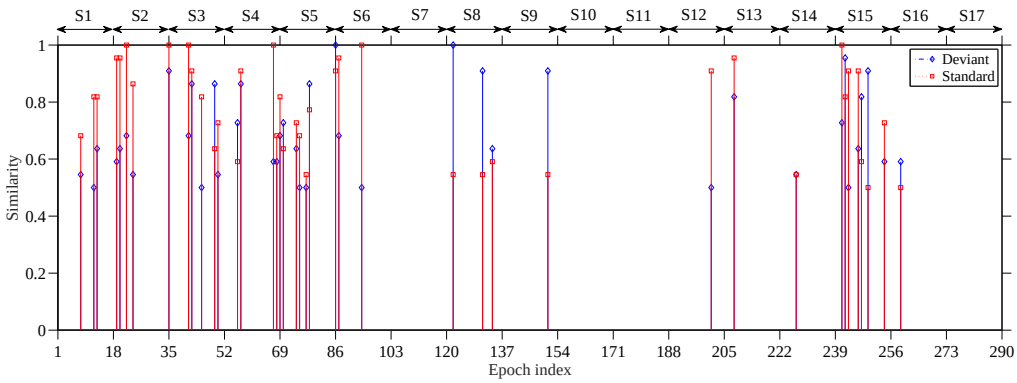
We hypothesize that the lack of discernibility of the N1 and MMN components in the grand averages in Fig. 4.2 is a result of the waxing and waning of the patients' level of consciousness throughout the recording interval; i.e., the presence of possible N1 and MMN components at the highest level of consciousness is obscured, or smeared out, by the averaging procedure. To verify this idea, we use patients 1-3 as test subjects and, as is discussed in Section 4.4.2, extract query data from the patients according to Protocol P1 where the required parameters are set to the same values as those used for the normal subjects.

Fig. 4.3 shows similarities, but only for the "active" 2-min epochs. Active epochs are those for which both standard and deviant similarities, as defined by (2.13), exceed a threshold Θ, which in our experiments is set to 0.5.

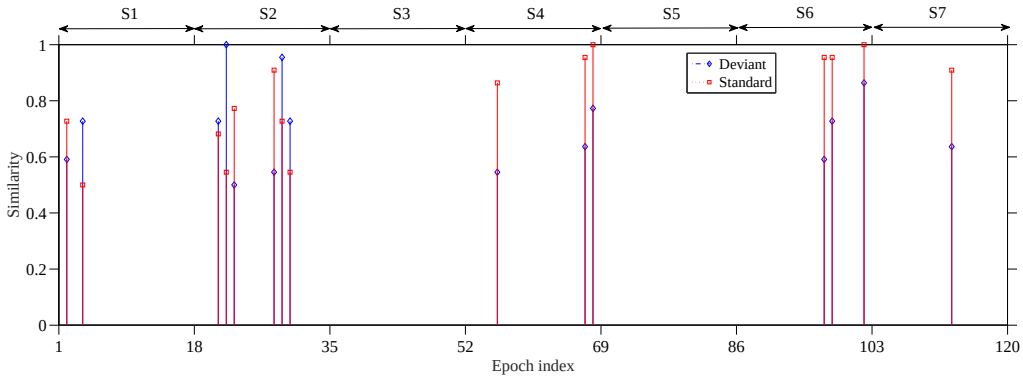
Fig. 4.3(a) shows similarity measures of the patients vs. the 2-minute epoch index. The first 17 ticks correspond to the 17 2-min epochs of session 1 (S1) and 18-34 correspond to the epochs of session 2, etc. Sessions are concatenated even though they are not contiguous in time. This figure demonstrates that patient 1 has many epochs in which the brain shows significant responses to both standard and deviant stimuli– i.e. the brain activity has high similarity to that of normal subjects. However, due to the significantly shortened averaging window, this figure also verifies the existence of the waxing and waning behavior of the patient's level of consciousness, where in some sessions there are a large number of epochs that show high similarity to normal brain function (waxing) while in other sessions there is a decreased number of epochs in which the patient's brain works similarly to normal brain (waning).

Fig. 4.3(b) demonstrates the waxing and waning behavior of the level of consciousness for patient 2. As expected, since this patient never fully recovered to a normal state and was in a low-mid level of consciousness for a longer time compared to patient 1, a lower similarity compared to the patient 1 is shown.

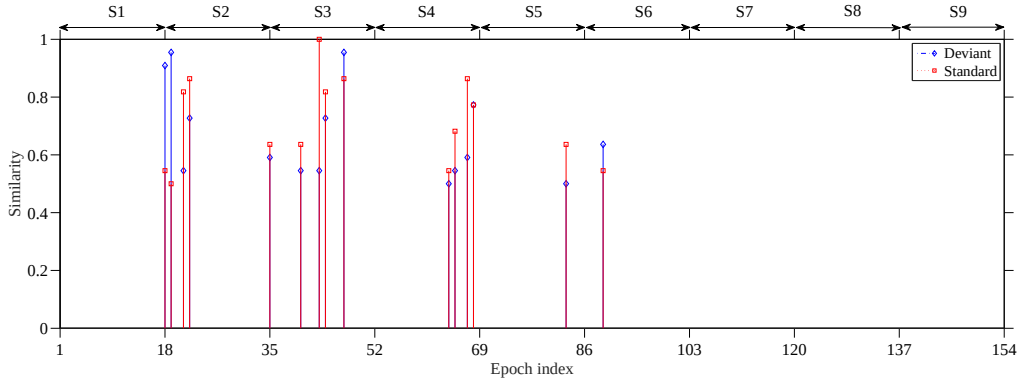
Fig. 4.3(c) demonstrates that, as we may have suspected, there are relatively few epochs in which the patient's brain response to stimuli is similar to that of normal brains. However, a significant level of similarity is indicated in the earlier sessions. At the time of extubation he could have been in a waning state as indicated by the later sessions, where there are no active epochs.



(a) Patient 1–regained consciousness



(b) Patient 2–slowly regained consciousness



(c) Patient 3–died

Figure 4.3: Similarity of the ERPs of patients 1-3 to the corresponding ERPs of normal subjects, vs. epoch index. Similarities are shown for the active epochs where Θ is set to 0.5. The red and blue graphs are respectively corresponded to $S_{Y_1}(x^q_{std};\gamma)$ and $S_{Y_2}(x^q_{std};\gamma)$.

Table 4.2: waxing criterion C_{wax} for the three comatose patients with different value of the parameter Θ.

	Patient 1	**Patient 2**	**Patient 3**
$\mathbf{\Theta = 0.4}$	20.76	16.81	13.73
$\mathbf{\Theta = 0.5}$	14.19	12.61	9.8
$\mathbf{\Theta = 0.6}$	6.23	5.88	2.61

We propose a measure indicating what percentage of the recording time each patient is in the waxing state, as follows:

$$C_{wax} = \frac{\#\,\text{active epochs}}{\#\,\text{of total epochs}} \times 100. \tag{4.4}$$

The waxing criteria C_{wax} for each patient using three different threshold values Θ are shown in Table 4.2. As is expected, for all Θ values, $C_{wax}(\text{patient 1}) \geq C_{wax}(\text{patient 2}) \geq C_{wax}(\text{patient 3})$ which indicates a higher chance of regaining consciousness for patients 1 and 2 compared with patient 3.

Verification of the results

In this section we provide extra verification of the proposed methodology for identification of the epochs that have MMN. To this end, we examine some typical active and non-active 2-minute epochs from different sessions of all patients to determine whether the decision output by the proposed method is in agreement with the conclusions drawn from visual inspection of those epochs.

The average standard and deviant ERPs over Sessions 4 and 5 of patient 1 are shown in Fig. 4.4. From the figure, which shows only electrodes Fz and Cz, an N1 component can be seen in the averaged signal of Session 4. However, visual inspection over all channels suggests there is no evidence of the MMN in any of the 32 channels

of Session 4. The averaged signals over Session 5 show a very weak negative wave ($< 0.25\mu V$) at about 100 ms but one cannot be sure whether this is indeed an N1 or noise. For similar reasons, one cannot be definite about whether the small inflection appearing after N1* at electrode Cz is an MMN component (we denote such tiny indefinite inflections by the superscript *). Therefore, since there is no definitive MMN component that can be detected by visual inspection, the sub-grand average signals indicate that there is no evidence of an appropriate brain response to the deviant stimuli; whereas the detection results of the proposed method shown in Fig. 4.3(a) indicate that there are many active epochs in Sessions 4 and 5 (i.e. S4 and S5) that have high similarity to the normal brain function.

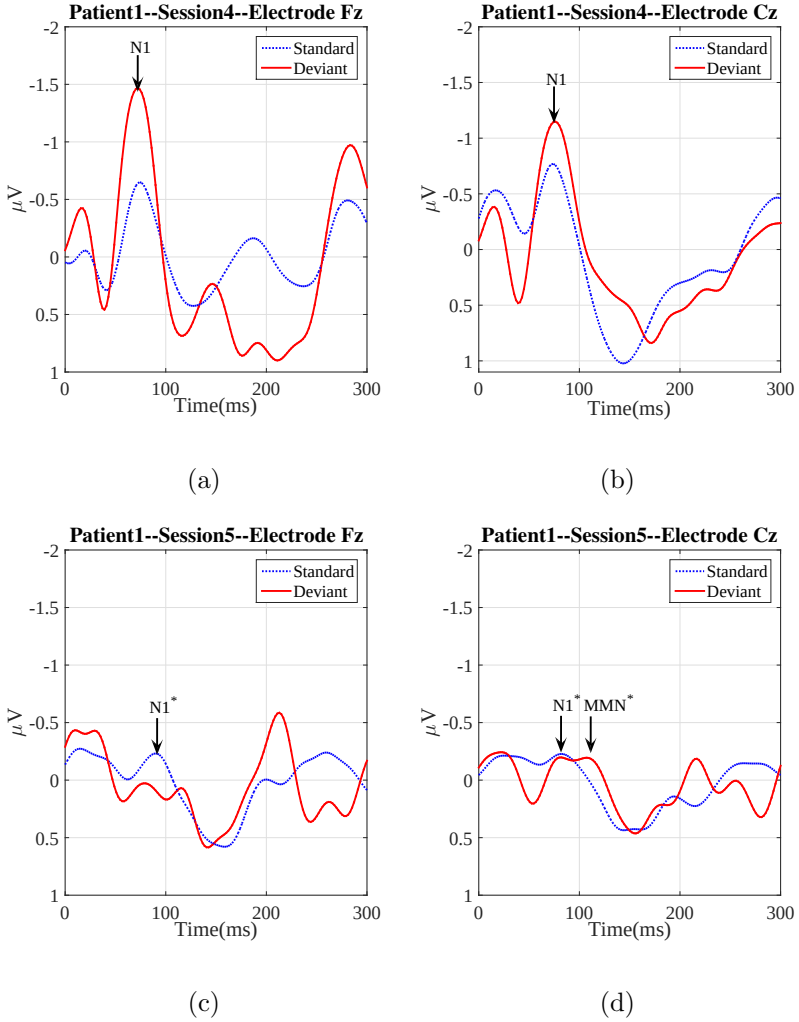


Figure 4.4: Sub-grand average of standard and deviant ERPs of patient 1 at sessions 4 and 5 at (a), (c) channel Fz and (b), (d) channel Cz.

Averaged signals corresponding to standard and deviant tones over four typical *single* 2-min epochs are shown in Fig. 4.5. Visual inspection of these epochs demonstrates that an MMN component exists at the 6th epoch of session 4 (i.e. S4), (Fig. 4.5(a)) and the 10th epoch of Session 5 (i.e. S5) (Fig. 4.5(d)). However, at the 13th epoch of S4 (Fig. 4.5(b)) and the 3rd epoch of S5 (Fig. 4.5(c)) there is no MMN neither in channel Fz nor in any other electrode location. As expected, an

N1 clearly appears in all of these epochs, since this is an obligatory response to each tone. Note that the components appearing in these figures cannot be detected from the grand average and the sub-grand average signals shown in Figs. 4.2 and 4.4. This can be considered a verification of our idea that waxing and waning of the level of consciousness exists, and that excessive averaging can obscure this behaviour.

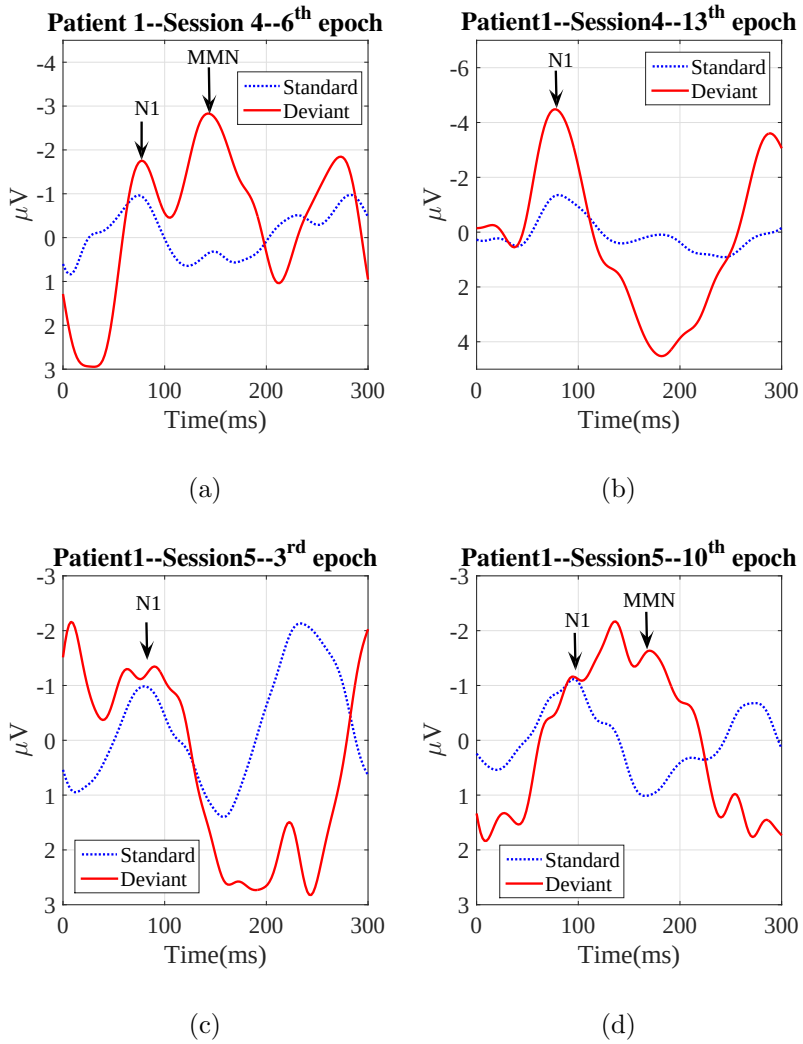
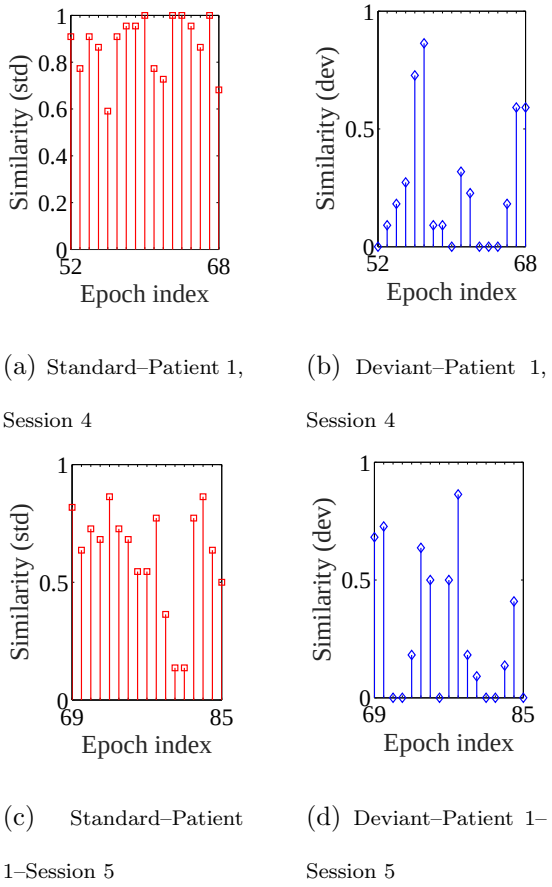


Figure 4.5: Average of standard and deviant ERPs, at channel Fz, of patient 1 at epochs a) 6th and b) 13th of session 4, and epochs c) 3rd and d) 10th of Session 5.

We use these visual observations as ground truth and then verify the performance of the proposed method based on the calculated similarities to normal subjects. To this end, the similarity of the patient 1 brain function to that of the normal brain over all epochs of Sessions 4 and 5 are shown in Fig. 4.6, which again shows that the patient's brain reaction to the deviant tones varies over time (waxing and waning). Fig. 4.6(a) and Fig. 4.6(c) show that, as expected, patient's brain response to the standard stimuli has a good similarity to those of the normal subjects– this verifies the existence of the obligatory response of the brain to standard tones. Figs. 4.6(b) and 4.6(d) demonstrate that epochs 6 and 10 respectively of Sessions 4 and 5 show a high similarity to the corresponding ERPs of normal subjects – i.e. the ERPs corresponding to the deviant stimuli within these epochs have an MMN component and the patient is likely to emerge from coma (which in fact he did). Furthermore, these figures demonstrate that, over epochs 13 and 3 of respectively Sessions 4 and 5, there is zero similarity between patients response to the deviant stimuli and those of the normal subjects – this indicates that the patient didn't have an MMN component within these epochs (waning). These results are consistent with the visual inspection results shown in Fig. 4.5. This verifies performance of the proposed method in detecting the presence of MMN over a short 2-min window.



(a) Standard–Patient 1, Session 4

(b) Deviant–Patient 1, Session 4

(c) Standard–Patient 1–Session 5

(d) Deviant–Patient 1–Session 5

Figure 4.6: Similarities of (a),(c) standard (b),(d) and deviant ERPs of patient 1 respectively at Sessions 4 and 5 to those of the normal training subjects. Each vertical bar is corresponds to a 2-min epoch.

In a manner similar to that of patient 1, in the following the performance of the proposed method is verified over two typical single 2-min epochs of patient 2 using visual inspection. To this end the sub-grand average of sessions 2 and 3 are shown in Fig.4.7 where there are some indefinite components shown by N1* and MMN*. They are indefinite because either the N1 does not appear in both standard and deviant responses (session 2), or they are too weak to be reliable (session 3).

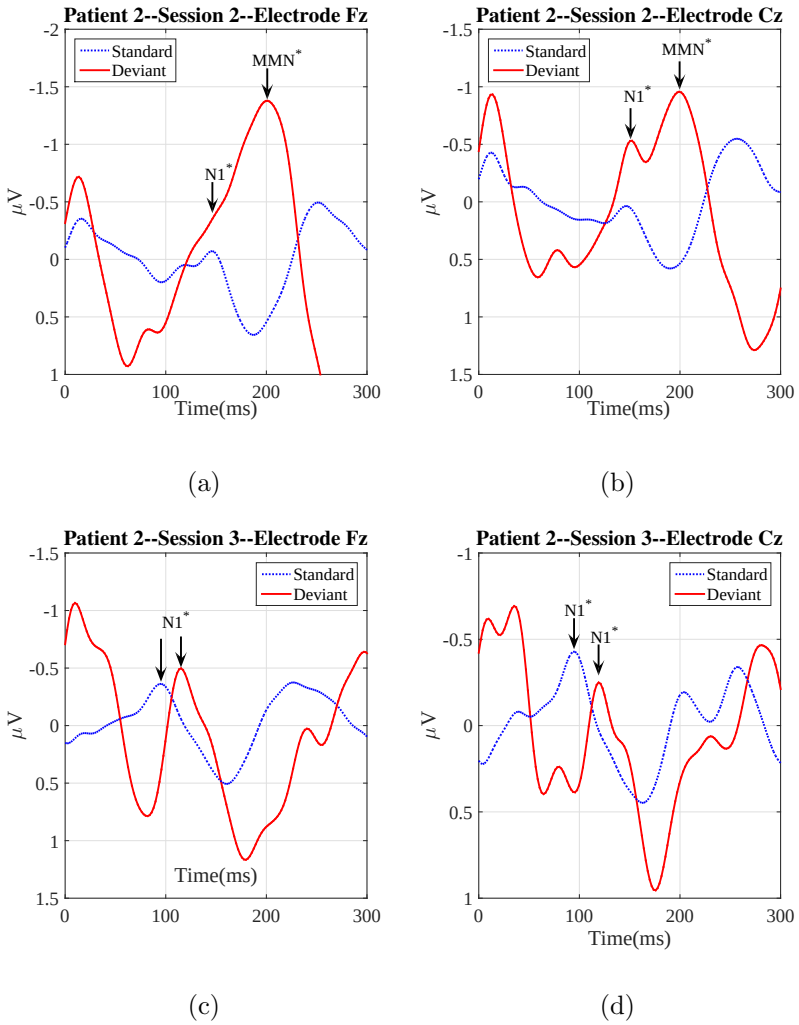
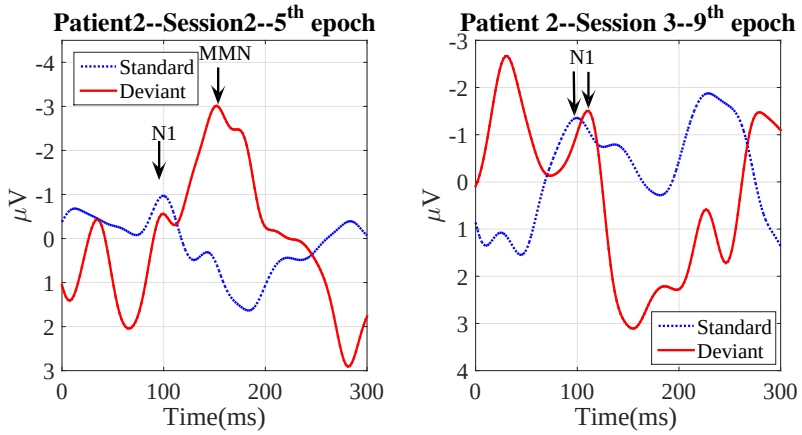


Figure 4.7: Sub-grand average of standard and deviant ERPs of patient 2 at sessions 2 and 3 at (a), (c) channel Fz and (b), (d) channel Cz.

The average of the ERPs corresponding to the standard and deviant stimuli over two typical epochs of sessions 2 and 3 are shown in Fig. 4.8, where visual inspection indicates that the N1 component appears in both these epochs, but the MMN is manifest only in the 5th epoch of session 2.

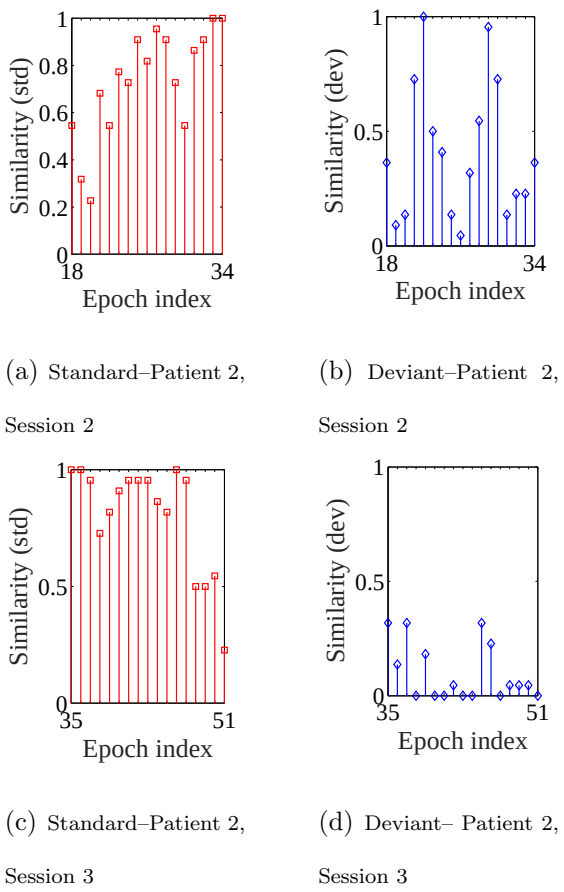
As in the case of patient 1, we again use these observations as ground truth. To this end, the similarity of the brain function of patient 2 over all epochs of sessions 2 and 3



(a) (b)

Figure 4.8: Average of standard and deviant ERPs, at channel Fz, of patient 2 at a) 5th epoch of session 4 and b) 9th epoch of Session 3.

to normal brain function is shown in Fig. 4.9. As expected, all 2-min epochs of sessions 2 and 3 have high similarity to the standard tones which indicates the presence of the obligatory component N1 over all epochs. The high and low similarities respectively shown in Figs. 4.9(b) and 4.9(d) demonstrate that the patient's consciousness was in his waxing phase in session 2 and in his waning phase in session 3.



(a) Standard–Patient 2, Session 2

(b) Deviant–Patient 2, Session 2

(c) Standard–Patient 2, Session 3

(d) Deviant– Patient 2, Session 3

Figure 4.9: Similarities of (a), (c) standard (b), (d) and deviant ERPs of patient 2 at Sessions 2 and 3 to those of the normal train subjects. Each bar is corresponded to a 2-min epoch.

In addition, the 5th epoch of session 2 has a very high similarity to both standard and deviant responses of the normal subjects which indicates the presence of MMN component at this epoch – this is consistent with the ground truth obtained by visual inspection of this epoch (see Fig. 4.8(a)). Furthermore, the 9th epoch of session 3 has a respectively high and zero similarity to the standard and deviant responses of the normal subjects, which indicates that the N1 component is elicited in this epoch while the MMN is not– this result is also consistent with the ground truth obtained

by visual inspection of this epoch (see Fig. 4.8(b)) which again demonstrates the performance of the proposed method for automatic and continuous assessment of ERPs for the purpose of coma outcome prediction.

Note that both patients 1 and 2 demonstrate intervals where an MMN is present, indicating a higher level of consciousness over these periods. Since these patients recovered, there is evidence to suggest that even short periods of attentiveness are predictive of a positive prognosis.

The same verification procedure is also applied again to patient 3. Recall that this patient died five days after extubation. The grand average signals shown in Fig. 4.2 give an indefinite indication that an MMN component might be elicited at about 200 ms. It is indefinite because the respective N1 component is too weak. To examine whether this patient indeed exhibits an MMN component, the sub-grand average signals over sessions 2 and 3 of patient 3 are shown in Fig. 4.10. This figure demonstrates that the MMN component may be elicited in session 3; however, we cannot verify the presence of the N1 component in the deviant response. Further, it demonstrates that some indefinite components appeared in the deviant response of Session 2, however these inflections are either too weak or the N1 component doesn't appear properly in the standard response.

To examine whether any MMN is elicited, we look at two typical epochs of these sessions. In this vein, the average of the ERPs corresponding to the standard and deviant stimuli over these two epochs are shown in Figs. 4.11 and 4.12. These figures clearly demonstrate the presence of N1 and MMN components in epochs 2 and 13 of respectively sessions 2 and 3. We use these visual inspections as ground truth for the proposed method and investigate the similarities corresponding to these two epochs.

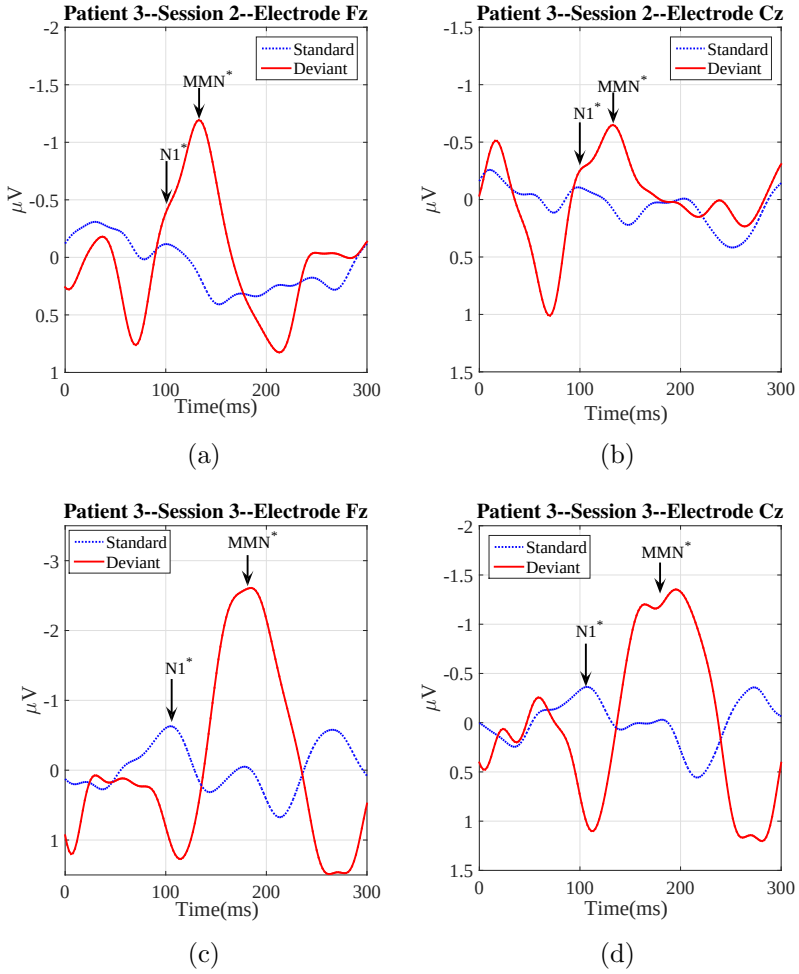


Figure 4.10: Sub-grand average of standard and deviant ERPs of patient 3 at sessions 2 and 3 at (a), (c) channel Fz and (b), (d) channel Cz.

To this end, the similarity of all epochs of sessions 2 and 3 to the corresponding ERPs of the normal subjects are shown in Fig. 4.13. These similarities indicate that there is a high similarity between the ERPs corresponding to the 2nd and 13th epochs of sessions 2 and 3 and those of the normal brain function. This high similarity correctly matches the ground truth.

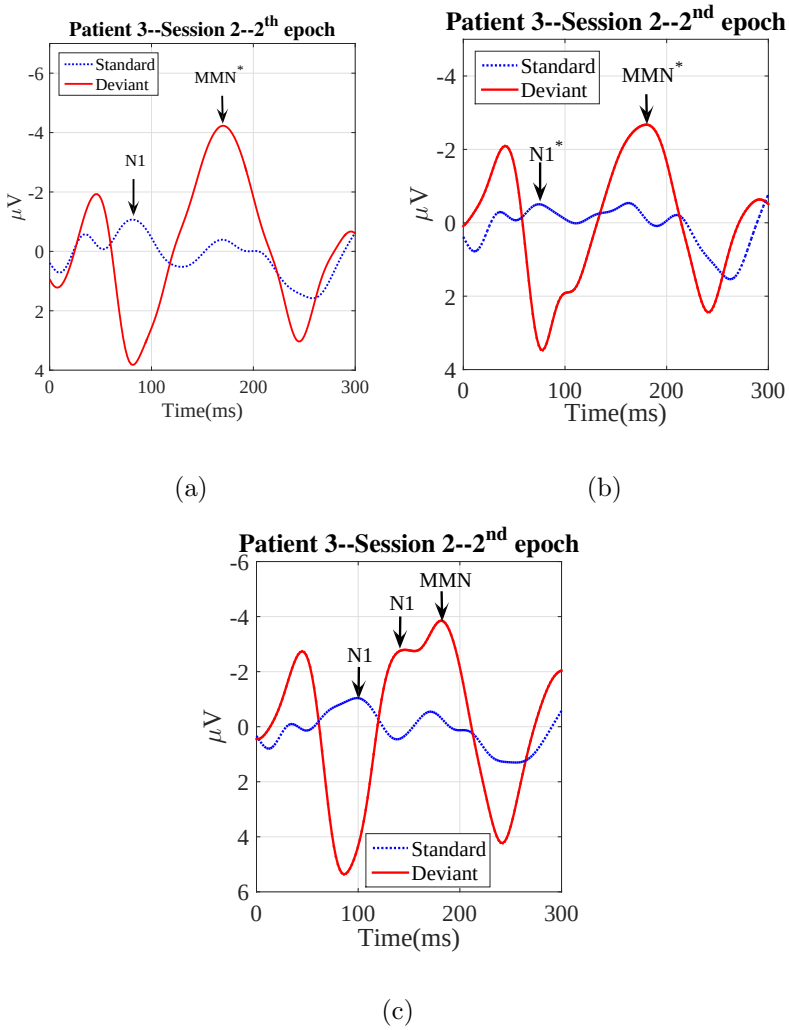
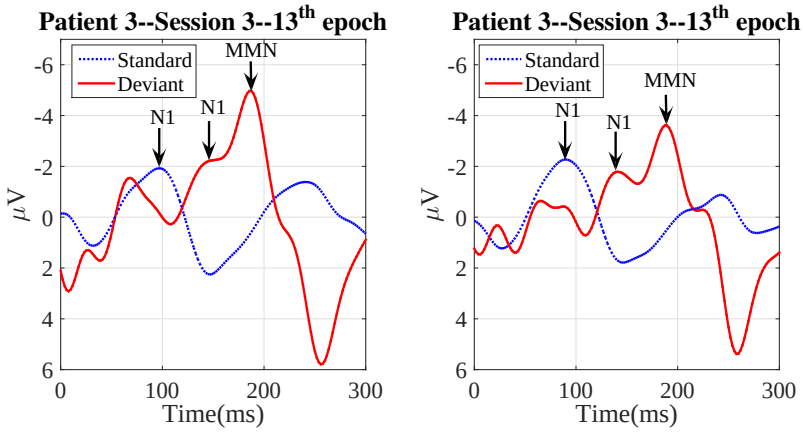
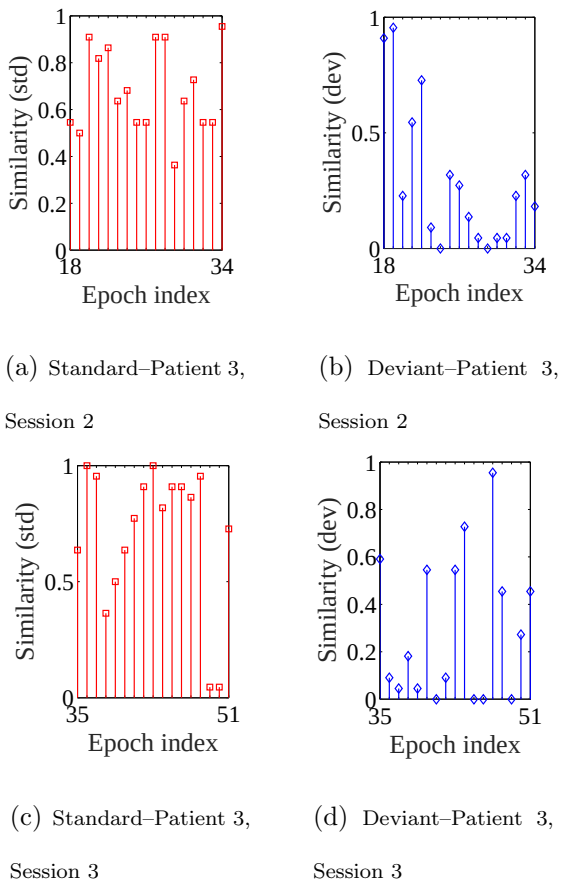


Figure 4.11: Average of standard and devinat ERPs at 2nd epoch of session 2 of patient 3, at channels a) Fz b) Cz c) C4.



(a) (b)

Figure 4.12: Average of standard and devinat ERPs of patient 3 occured at 13th epoch of session 3, at channels a) Fz b) FC1.



(a) Standard–Patient 3, Session 2

(b) Deviant–Patient 3, Session 2

(c) Standard–Patient 3, Session 3

(d) Deviant–Patient 3, Session 3

Figure 4.13: Similarities of (a), (c) standard and (b), (d) deviant ERPs of patient 2 at Sessions 2 and 3 to those of the normal train subjects. Each bar is corresponded to a 2-min epoch.

4.6 Discussion and Conclusions

In this study we proposed a machine learning based methodology for automatic and continuous assessment of ERPs for identifying the presence of the MMN component. The method consists of two phases: learning and testing. In the former discriminative sub-spaces are trained using the LFS method and available training points, and then the trained spaces are used in the test phase for continuous assessment of a test

subject.

Due to the impracticality of collecting sufficient data from coma victims alone for training purposes, an indirect approach to training and testing are employed in this study, whereby the training data is provided exclusively by healthy subjects, rather than coma patients. Experiments on data collected from healthy subjects show that the proposed machine learning method has high accuracy (92.7%) in discriminating between responses to standard and deviant tones.

The prognosis of a (coma) subject is determined by assessing the similarity of the subject's ERP responses to those of healthy subjects. A high similarity in the deviant responses indicates that the MMN exists with high probability. Since the presence of the MMN is highly correlated with recovery, a high similarity (as indicated by the proposed method) therefore suggests recovery. Thus, the proposed method gives a practical and accurate approach for determining coma prognosis. The effectiveness of the proposed method in assessing coma prognosis has been established by comparing similarity results to the ground truth obtained by visually examining ERP responses averaged over short windows. In addition, the proposed method reduces load, cost and impracticality of requiring frequent expert assessment during the course of a day.

We have seen that the presence of the MMN can be obscured by averaging over excessively long windows. The proposed machine learning approach is capable of detecting the presence of the MMN over short, 2–minute windows. This ability to detect over short windows verifies the existence of waxing and waning cycles of consciousness in a coma victim. Furthermore, our results suggest that the existence of even very short intervals of a high level of consciousness is strongly related to recovery. Since the proposed method is capable of detecting over such short intervals, the sensitivity

of the approach is therefore improved over that of previous methods.

The high sensitivity of the proposed method can be demonstrated by the fact that there is no clear evidence of an MMN component in any of the patients in Figure 4.2. Thus the prognosis of conventional methods for these victims would have been negative. In contrast, since intervals of high similarities to healthy responses were demonstrated in patients 1 and 2, the prognosis rendered by proposed method is positive. The fact that patients 1 and 2 did recover verifies the improved performance of the proposed method in these cases. Furthermore, a high specificity of more than 92% of the proposed methodology is demonstrated on normal subjects (see TNR in table 4.1).

An additional consideration that the proposed method reveals is with regard to patient 3. It is clear from Figure 4.3 that periods of high similarity did exist, indicating he may have recovered if life support had continued. Thus we see an instance where the decision to withdraw life support may have been premature.

We have proposed a machine learning method for coma prognosis that involves selection of features that are relevant to coma prognosis. Hence, the selected features could perhaps give us clues about the neurological function of recovery. In this study however we have made no effort to investigate this matter, although it would be an interesting avenue to pursue as future work.

Chapter 5

Conclusions

5.1 Research summary

In this study we introduce the concept of localized feature selection. The proposed local feature selection algorithm adaptively assigns a specific optimal feature subset to each of the sample space regions, in contrast to the traditional methods, which select a common feature set for the entire sample space. This allows the feature set to optimally adapt to local variations of the sample space.

The process of computing a specific feature subset for each region is independent of those of other regions and hence can be performed in parallel. Since the proposed algorithm makes no assumptions regarding the data distribution over the sample space, it is also an appropriate approach for the case where the data are distributed on a non-linear and/or a disjoint manifold.

The proposed localized feature selection idea is first realized through a linear programming optimization problem where the neighboring samples are determined using an iterative approach initiated based on the distances in the original feature

space (Chapter 2). Then, to have a more accurate determination of the neighboring samples, the localized feature section idea is formulated as a joint convex/increasing quasi-convex optimization problem with no local minima where a logistic function is applied to focus the optimization process on the localized region within the unknown co-ordinate system (Chapter 3).

A query datum is classified through aggregation of "weak" classifier results which are based on the selected region-specific feature subsets. The Vapnik–Chervonenkis (VC) dimension is determined and, under certain assumptions, is found to have a finite, moderate value. This, in combination with the fact that the proposed localized approach selects only relevant features, suggest both the LFS and the lLFS methods are not overly sensitive to the overfitting problem. In this study we specifically consider the more challenging problems where a small number observations are available for training. Experimental results demonstrate the superior performance of the proposed algorithm on a large variety of data sets.

In Chapter 4, using the proposed localized feature selection and classification idea, we propose a practical machine learning (ML) approach for automatic and continuous assessment of the ERPs for detecting the presence of the MMN component which has a good correlation with coma awakening. The method is capable of detection over windows as short as two minutes. This finer time resolution enables identification of waxing and waning cycles in level of consciousness. Experimental results on normal and comatose subjects demonstrate effectiveness of the proposed method for automatic and continuous assessment of ERPs for MMN detection.

5.2 Future research

In this PhD study, we investigate the localized feature selection idea for supervised data classification where the class label of the training data are given. As a future work the idea of local feature selection can be extended to the unsupervised data clustering where there is no label for the training data.

In addition the proposed localized idea can be extended to the semi-supervised case. One approach to semi-supervised learning e.g. (Gu *et al.*, 2013; Bennett *et al.*, 1999) is to use the few labelled samples to iteratively classify the unlabelled points, retraining the classifier in each iteration. The difficulty with such approaches is they are generally slow, due to the intensive retraining process. The proposed feature selection method offers a major advantage in this respect, since the incorporation of a new unlabelled data sample may be implemented with only a single optimization on the new data point without extensive retraining involving all available samples, thus rendering the proposed method approach much faster.

Section 2 shows linear complexity of the LFS method with respect to feature dimensionality – i.e. the required CPU time for computing a feature subset increases linearly with the number of available candidate features. This interesting linear behavior is because of the way that the constraint set is defined. As a future work it would be interesting to investigate properties of the constraint set to define theoretical complexity bounds and show why the results are linear?

In this work, the proposed formulations presented for LFS and lLFS algorithms are solved using *linprog* and *fmincon* functions available in MATLAB. As a future work, it would be interesting to investigate possibly better computational algorithms that are better suited to the specific objective function – e.g. , both maximization and

minimization objective functions used in the LFS and lLFS algorithms are positive, and this could make way for an improved solver.

Another way to extend and improve the proposed localized idea is to investigate the localized classification in a probabilistic sense, using e.g. kernel density functions instead of the hyperspheric classifier. More like a soft decision than a hard decision.

In this study, the proposed framework for MMN detection is based on the ERPs happened in 2-min intervals. As a future work one may try to reduce the interval duration to be shorter than 2 minutes and investigate what is the maximum lower limit and what price do we pay?

The proposed localized idea can model non-stationarity of the EEG signals. In this study the proposed idea is applied for coma outcome prediction. However, the proposed localized modeling has more potential applications including identifying effective medications for treatment of depression and schizophrenia, identifying traumatic brain injury from either resting state EEG or ERPs, applications to brain computer interface, etc.

Appendix A

Vapnik Chervonenkis dimension of the localized classifier

In this section the VC dimension of the proposed LFS classifier, defined in Section 2.2.2 is discussed. To this end, for simplicity, we only consider the case where the number of classes is two, i.e. $\mathcal{Y} = \{Y_1, Y_2\}$. Based on Sections 2.2.2 and 2.3.1, the family of functions $\mathcal{F} = \{f(\mathbf{x}^{(i)}; \gamma)\}$ for the LFS classifier is given such that the functions $f(\cdot;\cdot)$ are defined as

$$f(\mathbf{x}^{(i)}; \gamma) = \underset{Y_l \in \mathcal{Y}}{\operatorname{argmax}} \{S_{Y_1}(\mathbf{x}^{(i)}; \gamma), S_{Y_2}(\mathbf{x}^{(i)}; \gamma)\} \tag{A.5}$$

where $S_{Y_l}(\mathbf{x}^{(i)}; \gamma)$, $l = 1, 2$ is defined in (2.11) to (2.13). The only parameter which varies in f is the radius of the hyper-spheres, controlled through γ.

It is necessary to make assumptions in the derivation of the VC dimension for the LFS classifier. First, we assume that within the kth frame (i.e. the induced space corresponding to $\mathbf{x}^{(k)}$), some points of the same class as $\mathbf{x}^{(k)}$ form a cluster around

$\mathbf{x}_p^{(k)}$. We assume that samples within the same data cluster are close enough such that the localized cluster centers and the radii of the corresponding weak classifiers are similar enough so that the query datum falls within all hyper-spheres. This is not unreasonable since, in the proposed localized feature selection approach, the corresponding feature subset of the kth frame is selected to encourage clustering. Furthermore, we assume the underlying problem is well–behaved so that the number of clusters L does not go to infinity as $N \rightarrow \infty$, where N is the total number of training points.

Theorem: We are given the LFS class of functions $\mathcal{F}$ as described. Then, under the stated assumptions, the VC dimension is $L(\lceil \frac{1}{\gamma} \rceil - 1)$.

Proof:

Recall the radius of a weak classifier grows until the "impurity level" of the corresponding hype-sphere $\mathcal{Q}^{(k)}$ is not greater than the predefined parameter γ, where "impurity level" is the ratio of the number of samples with opposite class label to the number of samples having the same class label as $\mathbf{x}_p^{(k)}$. It follows therefore that, in the shattering process to define the VC dimension, each weak classifier corresponding to the cluster $\mathcal{L}_l$ misclassifies $\lfloor \gamma \, |\mathcal{L}_l| \rfloor$ samples where $\lfloor \cdot \rfloor$ denotes the floor function and $|\mathcal{L}_l|$ is the cardinality of the lth cluster where $l = 1, \ldots, L$. Therefore, in the shattering process, there is no mis-classification as long as $\lfloor \gamma |\mathcal{L}_l| \rfloor = 0$, i.e. the maximum cardinality of the lth cluster without any classification error is $\lceil \frac{1}{\gamma} \rceil - 1$ where $\lceil \cdot \rceil$ denotes the ceiling function. Hence, over L clusters, the LFS classifier can shatter *at least* $L(\lceil \frac{1}{\gamma} \rceil - 1)$ samples.

Now, assume the case where there is an extra training point added; i.e., there are altogether a total of $L(\lceil \frac{1}{\gamma} \rceil - 1) + 1$ training samples. This extra training point will be

situated in one of the existing clusters. Without loss of generality, in the shattering process, we assign label Y_1 to the samples of this cluster and label Y_2 to the samples of the other clusters. The number of samples with label Y_1 is η_1. The radii of all weak classifiers associated with the cluster Y_1 must now grow until one sample from class Y_2 is mis-classified, i.e. until the impurity level is not greater than γ. This sample will be mis-classified by all η_1 weak classifiers (see the assumptions). Therefore, for this sample, the first term of the argmax function in (A.5) is 1 while the second term is less than or equal to 1. Hence, for any value of L, the classifier output is Y_1 or 0 (i.e no decision), which is a wrong decision. Similarly, by increasing the number of training points in the shattering process, there will be a class label combination in which at least one point will be mis-classified.

Thus the number of points that can be shattered is *at most* $L(\lceil \frac{1}{\gamma} \rceil - 1)$, i.e. the VC dimension of the LFS classifier is $L(\lceil \frac{1}{\gamma} \rceil - 1)$.

Bibliography

Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S., and Koutsoukos, X. D. (2010). Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, **11**(Jan), 171–234.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**(12), 6745–6750.

Anthony, M. and Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.

Armanfard, N. and Reilly, J. P. (2013). Classification based on local feature selection via linear programming. In *Machine Learning for Signal Processing (MLSP), IEEE International Workshop on*, pages 1–6.

Armanfard, N., Komeili, M., Reilly, J. P., Mah, R., and Connolly, J. F. (2016a). Automatic and continuous assessment of ERPs for mismatch negativity detection.

In *2016 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.

Armanfard, N., Reilly, J. P., and Komeili, M. (2016b). Local feature selection for data classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(6), 1217–1227.

Armanfard, N., Komeili, M., Reilly, J. P., and Pino, L. (2016c). Vigilance lapse identification using sparse EEG electrode arrays. In *Electrical and Computer Engineering (CCECE), 2016 IEEE 29th Canadian Conference on*.

Bache, K. and Lichman, M. (2013). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, `http://archive.ics.uci.edu/ml`, accessed January 2014.

Bellman, R. E. and Dreyfus, S. E. (1962). Applied dynamic programming.

Bennett, K., Demiriz, A., *et al.* (1999). Semi-supervised support vector machines. *Advances in Neural Information processing systems*, pages 368–374.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., Benítez, J. M., and Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*, **282**, 111–135.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Braga-Neto, U. M. and Dougherty, E. R. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, **20**(3), 374–380.

Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, **13**(Jan), 27–66.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**(2), 121–167.

Cao, C., Tutwiler, R. L., and Slobounov, S. (2008). Automatic classification of athletes with residual functional deficits following concussion by means of EEG signal using support vector machine. *IEEE Transactions on Neural Systems and Rehabilitation engineering*, **16**(4), 327–335.

Chakraborty, R. and Pal, N. R. (2015). Feature selection using a neural framework with controlled redundancy. *Neural Networks and Learning Systems, IEEE Transactions on*, **26**(1), 35–50.

Chawla, J., Burneo, J. G., and Barkley, G. L. (2016). Clinical applications of somatosensory evoked potentials. *Medscape*.

Chen, B., Liu, H., Chai, J., and Bao, Z. (2009). Large margin feature weighting method via linear programming. *Knowledge and Data Engineering, IEEE Transactions on*, **21**(10), 1475–1488.

Cheng, Q., Zhou, H., and Cheng, J. (2011). The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**(6), 1217–1233.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Ismb*, volume 8, pages 93–103.

Chiappa, K. H. and Hill, R. A. (1998). Evaluation and prognostication in coma. *Electroencephalography and Clinical Neurophysiology*, **106**(2), 149–155.

Coleman, T., Branch, M. A., and Grace, A. (1999). *Optimization Toolbox for Use with MATLAB: User's Guide, Version 2*. Math Works, Incorporated.

Crisler, S., Morrissey, M. J., Anch, A. M., and Barnett, D. W. (2008). Sleep-stage scoring in the rat using a support vector machine. *Journal of Neuroscience Methods*, **168**(2), 524–534.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons.

Duncan, C. C., Barry, R. J., Connolly, J. F., Fischer, C., Michie, P. T., Näätänen, R., Polich, J., Reinvang, I., and Van Petten, C. (2009). Event-related potentials in clinical research: guidelines for eliciting, recording, and quantifying mismatch negativity, P300, and N400. *Clinical Neurophysiology*, **120**(11), 1883–1908.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., *et al.* (2004). Least angle regression. *The Annals of Statistics*, **32**(2), 407–499.

Escera, C., Yago, E., Polo, M. D., and Grau, C. (2000). The individual replicability of mismatch negativity at short and long inter-stimulus intervals. *Clinical Neurophysiology*, **111**(3), 546–551.

Fischer, C., Morlet, D., Bouchet, P., Luaute, J., Jourdan, C., and Salord, F. (1999). Mismatch negativity and late auditory evoked potentials in comatose patients. *Clinical Neurophysiology*, **110**(9), 1601–1610.

Fischer, C., Luauté, J., Némoz, C., Morlet, D., Kirkorian, G., and Mauguière, F. (2006). Improved prediction of awakening or nonawakening from severe anoxic coma using tree-based classification analysis. *Critical Care Medicine*, **34**(5), 1520–1524.

Fischer, C., Dailler, F., and Morlet, D. (2008). Novelty P3 elicited by the subjects own name in comatose patients. *Clinical Neurophysiology*, **119**(10), 2224–2230.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**(1), 119–139.

Ghosh-Dastidar, S., Adeli, H., and Dadmehr, N. (2008). Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. *IEEE Transactions on Biomedical Engineering*, **55**(2), 512–518.

Gilad-Bachrach, R., Navot, A., and Tishby, N. (2004). Margin based feature selection-theory and algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 43.

Greenberg, R. P., Becker, D. P., Miller, J. D., and Mayer, D. J. (1977). Evaluation of brain function in severe human head trauma with multimodality evoked potentials: Part 2: Localization of brain dysfunction and correlation with posttraumatic neurological conditions. *Journal of Neurosurgery*, **47**(2), 163–177.

Gu, Z., Yu, Z., Shen, Z., and Li, Y. (2013). An online semi-supervised brain–computer interface. *IEEE Transactions on Biomedical Engineering*, **60**(9), 2614–2623.

Gui, J., Sun, Z., Ji, S., Tao, D., and Tan, T. (2016). Feature selection based on structured sparsity: A comprehensive study. *IEEE Transactions on Neural Networks and Learning Systems*. DOI: 10.1109/TNNLS.2016.2551724.

Guler, I. and Ubeyli, E. D. (2007). Multiclass support vector machines for EEG-signals classification. *IEEE Transactions on Information Technology in Biomedicine*, **11**(2), 117–126.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**(Mar), 1157–1182.

Holeckova, I., Fischer, C., Giard, M.-H., Delpuech, C., and Morlet, D. (2006). Brain responses to a subject's own name uttered by a familiar voice. *Brain Research*, **1082**(1), 142–152.

Hwang, C. L., Masud, A. S. M., *et al.* (1979). *Multiple objective decision making-methods and applications*, volume 164. Springer.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, **13**(4), 411–430.

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(1), 4–37.

Jewett, D. L. and Williston, J. S. (1971). Auditory-evoked far fields averaged from the scalp of humans. *Brain*, **94**(4), 681–696.

John, G. (1994). DNA dataset (statlog version) - primate splice-junction gene sequences (DNA) with associated imperfect domain theory. `https://www.sgi.com/tech/mlc/db/DNA.names`, accessed January 2014.

Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.

Jones, C. (1979). Glasgow coma scale. *AJN The American Journal of Nursing*, **79**(9), 1551–1557.

Kane, N. M., Butler, S. R., and Simpson, T. (2000). Coma outcome prediction using event-related potentials: P3 and mismatch negativity. *Audiology and Neurotology*, **5**(3-4), 186–191.

Khodayari-Rostamabad, A., Reilly, J. P., Hasey, G. M., de Bruin, H., and MacCrimmon, D. J. (2013). A machine learning approach using EEG data to predict response to ssri treatment for major depressive disorder. *Clinical Neurophysiology*, **124**(10), 1975–1985.

Khushaba, R. N., Al-Ani, A., and Al-Jumaily, A. (2011). Feature subset selection using differential evolution and a statistical repair mechanism. *Expert Systems with Applications*, **38**(9), 11515–11526.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, **97**(1), 273–324.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. In *Machine Learning: ECML-94*, pages 171–182.

Kwak, N. and Choi, C.-H. (2002). Input feature selection for classification problems. *Neural Networks, IEEE Transactions on*, **13**(1), 143–159.

Lee, Y., Phan, T., Jolley, D., Castley, H., Ingram, D., and Reutens, D. (2010). Accuracy of clinical signs, SEP, and EEG in predicting outcome of hypoxic coma a meta-analysis. *Neurology*, **74**(7), 572–580.

Lew, H. L., Poole, J. H., Castaneda, A., Salerno, R. M., and Gray, M. (2006). Prognostic value of evoked and event-related potentials in moderate to severe brain injury. *The Journal of Head Trauma Rehabilitation*, **21**(4), 350–360.

Li, Y., Dong, M., and Hua, J. (2008). Localized feature selection for clustering. *Pattern Recognition Letters*, **29**(1), 10–18.

Li, Y., Si, J., Zhou, G., Huang, S., and Chen, S. (2015). Frel: a stable feature selection algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, **26**(7), 1388–1402.

Liu, B., Fang, B., Liu, X., Chen, J., Huang, Z., and He, X. (2013). Large margin subspace learning for feature selection. *Pattern Recognition*, **46**(10), 2798–2806.

Liu, H. and Motoda, H. (2007). *Computational methods of feature selection*. Chapman and Hall/CRC.

Liu, Z., Hsiao, W., Cantarel, B. L., Drábek, E. F., and Fraser-Liggett, C. (2011). Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data. *Bioinformatics*, **27**(23), 3242–3249.

Lovato, P., Bicego, M., Kesa, M., Jojic, N., Murino, V., and Perina, A. (2016). Traveling on discrete embeddings of gene expression. *Artificial Intelligence in Medicine*, **70**, 1–11.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **1**(1), 24–45.

Madl, C., Kramer, L., Yeganehfar, W., Eisenhuber, E., Kranz, A., Ratheiser, K., Zauner, C., Schneider, B., and Grimm, G. (1996). Detection of nontraumatic comatose patients with no benefit of intensive care treatment by recording of sensory evoked potentials. *Archives of Neurology*, **53**(6), 512–516.

Marmarou, A., Lu, J., Butcher, I., McHugh, G. S., Murray, G. D., Steyerberg, E. W., Mushkudiani, N. A., Choi, S., and Maas, A. I. (2007). Prognostic value of the Glasgow Coma Scale and pupil reactivity in traumatic brain injury assessed prehospital and on enrollment: an impact analysis. *Journal of Neurotrauma*, **24**(2), 270–280.

Mavrotas, G. (2009). Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation*, **213**(2), 455–465.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

Morlet, D. and Fischer, C. (2014). MMN and novelty P3 in coma and other altered states of consciousness: A review. *Brain Topography*, **27**(4), 467–479.

Mushkudiani, N. A., Hukkelhoven, C. W., Hernández, A. V., Murray, G. D., Choi, S. C., Maas, A. I., and Steyerberg, E. W. (2008). A systematic review finds methodological improvements necessary for prognostic models in determining traumatic brain injury outcomes. *Journal of Clinical Epidemiology*, **61**(4), 331–343.

Nie, F., Huang, H., Cai, X., and Ding, C. H. (2010). Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In *Advances in Neural Information Processing Systems*, pages 1813–1821.

Oveisi, F., Oveisi, S., Erfanian, A., and Patras, I. (2012). Tree-structured feature extraction using mutual information. *Neural Networks and Learning Systems, IEEE Transactions on*, **23**(1), 127–137.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**(8), 1226–1238.

Petrova, L. D. (2009). Brainstem auditory evoked potentials. *American Journal of Electroneurodiagnostic Technology*, **49**(4), 317–332.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y., Goumnerova, L. C., Black, P. M., Lau, C., *et al.* (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**(6870), 436–442.

Ramona, M., Richard, G., and David, B. (2012). Multiclass feature selection with kernel gram-matrix-based criteria. *Neural Networks and Learning Systems, IEEE Transactions on*, **23**(10), 1611–1623.

Rappaport, M., Hall, K., Hopkins, K., Belleza, T., Berrol, S., and Reynolds, G. (1977). Evoked brain potentials and disability in brain-damaged patients. *Archives of Physical Medicine and Rehabilitation*, **58**(8), 333–338.

Ravan, M., Reilly, J. P., Trainor, L. J., and Khodayari-Rostamabad, A. (2011). A machine learning approach for distinguishing age of infants using auditory evoked potentials. *Clinical Neurophysiology*, **122**(11), 2139–2150.

Ravan, M., Hasey, G., Reilly, J. P., MacCrimmon, D., and Khodayari-Rostamabad, A. (2015). A machine learning approach using auditory odd-ball responses to investigate the effect of clozapine therapy. *Clinical Neurophysiology*, **126**(4), 721–730.

Robinson, L. R., Micklesen, P. J., Tirschwell, D. L., and Lew, H. L. (2003). Predictive value of somatosensory evoked potentials for awakening from coma. *Critical Care Medicine*, **31**(3), 960–967.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**(5500), 2323–2326.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**(19), 2507–2517.

Sánchez-Maroño, N., Alonso-Betanzos, A., and Tombilla-Sanromán, M. (2007). Filter methods for feature selection–a comparative study. In *Intelligent Data Engineering and Automated Learning-IDEAL 2007*, pages 178–187. Springer.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Sima, C. and Dougherty, E. R. (2006). What should be expected from feature selection in small-sample settings. *Bioinformatics*, **22**(19), 2430–2436.

Sjöstrand, K. (2005). Matlab implementation of LASSO, LARS, the elastic net and SPCA. `http://www2.imm.dtu.dk/pubdb/p.php?3897`, Version 2.0, accessed September 2016.

Souza, A. (2001). Randomized algorithm & probabilistic methods. Lecture Notes, Humboldt University of Berlin.

Sun, S., Peng, Q., and Shakoor, A. (2014). A kernel-based multivariate feature selection method for microarray data classification. *PloS One*, **9**(7), 1–12.

Sun, Y. (2007). Iterative relief for feature weighting: algorithms, theories, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(6), 1035–1051.

Sun, Y., Todorovic, S., and Goodison, S. (2010). Local-learning-based feature selection for high-dimensional data analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **32**(9), 1610–1626.

Tao, H., Hou, C., Nie, F., Jiao, Y., and Yi, D. (2015). Effective discriminative feature selection with nontrivial solution.

Teasdale, G. and Jennett, B. (1974). Assessment of coma and impaired consciousness: a practical scale. *The Lancet*, **304**(7872), 81–84.

Thai, M. T. (2013). Approximation algorithms: LP relaxation, rounding, and randomized rounding techniques. Lecture Notes, University of Florida, `http://optnetsci.cise.ufl.edu/class/cot5442sp15/Notes/Rounding.pdf`.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., *et al.* (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(6871), 530–536.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley and Sons Inc.

Wang, J., Zhao, P., Hoi, S. C., and Jin, R. (2014). Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, **26**(3), 698–710.

Wang, L. (2008). Feature selection with kernel class separability. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**(9), 1534–1546.

Wang, Y., Klijn, J. G., Zhang, Y., Sieuwerts, A. M., Look, M. P., Yang, F., Talantov, D., Timmermans, M., Meijer-van Gelder, M. E., Yu, J., *et al.* (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, **365**(9460), 671–679.

Webb, A. R. (2003). *Statistical pattern recognition*. Wiley.

Wei, H.-L. and Billings, S. A. (2007). Feature subset selection and ranking for data dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **29**(1), 162–166.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R., and Nevins, J. R. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, **98**(20), 11462–11467.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for svms. In *NIPS*, volume 12, pages 668–674.

Wu, X., Yu, K., Ding, W., Wang, H., and Zhu, X. (2013). Online feature selection with streaming features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(5), 1178–1192.

Xiang, S., Shen, X., and Ye, J. (2012). Efficient sparse group feature selection via nonconvex optimization. *ArXiv preprint arXiv:1205.5075*.

Young, G. B., Ropper, A. H., and Bolton, C. F. (1998). Coma and impaired consciousness: a clinical perspective.

Yu, K., Wu, X., Ding, W., and Pei, J. (2014). Towards scalable and accurate online feature selection for big data. In *2014 IEEE International Conference on Data Mining*, pages 660–669.

Zandbergen, E. G., de Haan, R. J., Stoutenbeek, C. P., Koelman, J. H., and Hijdra, A. (1998). Systematic review of early prediction of poor outcome in anoxicischaemic coma. *The Lancet*, **352**(9143), 1808–1812.

Zeng, H. and Cheung, Y.-m. (2011). Feature selection and kernel learning for local learning-based clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**(8), 1532–1547.

Zhu, X., Ghahramani, Z., Lafferty, J., *et al.* (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.

Zhu, Z., Ong, Y.-S., and Dash, M. (2007a). Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, **40**(11), 3236–3248.

Zhu, Z., Ong, Y.-S., and Dash, M. (2007b). Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **37**(1), 70–76.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), 301–320.