Local Feature Selection for Data Classification

Narges Armanfard, James P. Reilly, Majid Komeili

Abstract—Typical feature selection methods choose an optimal global feature subset that is applied over all regions of the sample space. In contrast, in this paper we propose a novel *localized* feature selection (LFS) 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 method for measuring the similarities of a query datum to each of the respective classes is also proposed.
The proposed method makes no assumptions about the underlying structure of the samples; hence the method is insensitive to the distribution of the data over the sample space. The method is efficiently formulated as a linear programming optimization problem. Furthermore, we demonstrate the method is robust against the over-fitting problem. Experimental results on eleven synthetic and real-world data sets demonstrate the viability of the formulation and the effectiveness of the proposed algorithm. In addition we show several examples where localized feature selection produces better results than a global feature selection method.

Index Terms—Local Feature Selection, Classification, Linear Programming.

1 Introduction

In many applications nowadays, data sets are characterized by hundreds or even thousands of features. Typically, there is often an insufficient number of objects to adequate represent the distribution of these high-dimensional feature spaces. Hence, dimensionality reduction is an important issue in a wide range of scientific disciplines. Many approaches for dimensionality reduction have been proposed in the literature [1], [2], [3]. Dimensionality reduction methods can be roughly categorized into two groups: feature extraction (also known as subspace learning) [4], [5], [6] and feature selection [7], [8], [9], [10], [11].

Feature extraction methods, like principal component analysis [5], linear discriminant analysis [12] and independent component analysis [13] mix original features to produce a new set of features. Since such features are a combination of the original features, the physical interpretation in terms of the original features may be lost. In addition to linear methods, there are also some nonlinear feature extraction methods which assume that data of interest lie on an embedded nonlinear manifold [6], [14], [15], [16]. Manifold learning techniques often need a large amount of training data and dense sampling on a manifold. Such rich training data may not be available in some real-world applications [17]. On the other hand, in many applications it is desired to reduce not only the dimensionality, but also the number of features that are to be considered. Unlike feature extraction, feature selection returns a subset of the original features without any transformation.

In this study, the feature selection process is considered for data classification. Given a set of training samples and their associated classes, the feature selection problem involves finding a subset of features that leads to an "optimal" characterization of the different classes. Conventional feature selection algorithms select a single common feature set for characterizing all regions of the sample space. In fact, these methods assume that a single feature subset can optimally characterize sample space variations.

In this paper we offer an alternative to the conventional feature selection approaches by introducing the novel concept of *localized* feature selection, where the optimal feature subset varies over the sample space in a manner that optimally adapts to local variations. We propose that an enhanced mathematical description of the sample space could be obtained by allowing various groups of samples in different regions to be associated with their own distinct feature set, which is optimized for that specific region.

Embedding local information is not inherently a new idea. LLE [6], Isomap [14], NPE [18] and MFA [19] apply local information for feature extraction (not feature selection) in which the physical interpretation of the induced co–ordinate system is lost. Bi-clustering approaches [20], [21], [22] apply local information for clustering of samples and features simultaneously, but these are basically unsupervised learning algorithms. Some more related approaches such as Logo [23], Simba [24], Relief [25] and MetaDistance[26] consider local sample behavior for feature selection, but all these algorithms suffer from the requirement that the entire sample space be modeled by a single common feature set.

In this paper, the concept of localized feature selec-

- *N. Armanfard and J. P. Reilly are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada.*
  E-mail: {armanfn, reillyj}@mcmaster.ca
- *M. Komeili is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada.*
  E-mail: mkomeili@ece.utoronto.ca

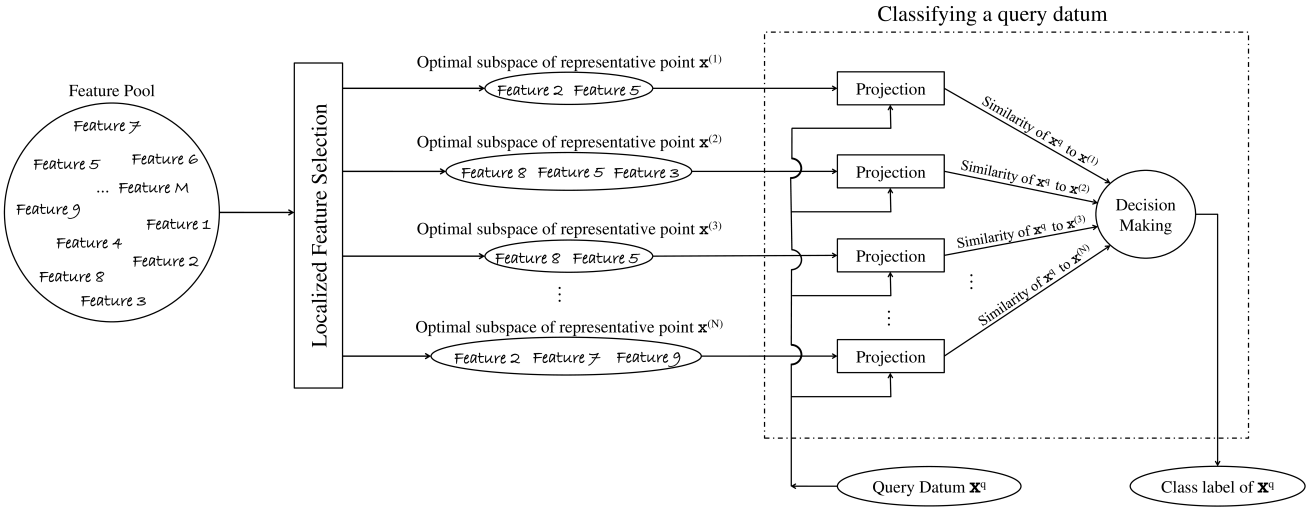

Fig. 1: Block diagram of the proposed method where each training sample $\mathbf{x}^{(i)}\, i = 1, \ldots, N$ is considered to be a representative point of its neighboring region and an optimal feature set (possibly different in size and membership) is selected for that region. Feature sets of all representative points are used for classification of a query datum $\mathbf{x}^q$. The detail of the feature selection and classification is presented in Sections 3.1 and 3.2, respectively.

tion is realized by considering each training sample as a representative point of its neighboring region and by selecting an optimal feature set for that region. The optimal feature set is such that, within its corresponding co–ordinate system, the within class distances and the between class distances are locally minimized and maximized, respectively. Since the optimal feature set is no longer constant over the sample space, ordinary classifiers are no longer appropriate for the proposed method. We therefore propose a localized classification procedure that has been adapted for our purposes. We refer to the proposed algorithm as the Localized Feature Selection (LFS) method.

The LFS method has several advantages. First, we make no assumptions regarding the distribution of the data over the sample space. The proposed approach therefore allows us to handle variations of the samples in the same class over the sample space, and to accommodate irregular or disjoint sample distributions. Moreover, we show later that the performance of the LFS method is robust against the overfitting problem. The proposed method also has the advantage that the underlying optimization problem is formulated as a linear programming optimization problem. Furthermore, feature selection process for different regions of 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.

An overview of the proposed method is shown in Fig. 1. An early version of this paper appeared in [27]. The remaining portion of this paper is organized as follows: Section 2 briefly reviews recent feature selection algorithms. Details of the proposed method for local feature selection and classification are presented in Section 3. In Section 4, experimental results, which demonstrate the performance of the proposed method over a range of synthetic and real-world data sets, are presented. Conclusions are drawn in Section 5.

2 Related Work

Feature selection has been an active research area in past decades. In this section we briefly review some of the main ideas of various feature selection approaches for data classification.

Some of the conventional feature selection approaches assign a common discriminative feature set to the whole sample space without considering the local behavior of data in different regions of the feature space [28], [7], [9], [29]. For example in [7] a common feature set is selected using a minimal redundancy maximal relevance criterion, which is based on mutual information. In [9] a common discriminative feature set is selected through maximizing a class separability criterion in a high-dimensional kernel space. In [29] a common feature set is computed using an evolutionary method, which is a combination of a differential evolution optimization method and a repair mechanism based on feature distribution measures. One conventional feature selection approach which seems to be close to the proposed LFS algorithm is feature selection using the Fisher criterion [12] (FDA) that computes a score for each feature based on maximizing between class distances and minimizing within class distances in the data space spanned by the corresponding feature. The main drawback of this algorithm, besides ignoring the local behavior of the samples, is that it considers features independently, leading to a sub-optimal subset of features.

On the other hand, several approaches exist that try to improve classification accuracy by local investigation of the feature space. One such approach are

the margin-based feature selection methods [25], [30], [24], [31], [32], [23], [33]. These methods are instance-based, where each feature is weighted to achieve maximal margin. The "margin" of a data point is defined as the distance between the nearest same-labeled data (near-hit) and the nearest differently labeled data (near-miss). RELIEF [25] detects those features which are statistically relevant to the target. One drawback of RELIEF is that the margins are not reevaluated during the learning process. Compared to RELIEF, the Simba algorithm [24] reevaluates margins based on the learned weight vector of features. However since its objective function is non-convex, it is characterized by many local minima. Recently, a local margin–based feature selection method is presented in [23], which uses local learning to decompose a complex nonlinear problem into a set of locally linear problems. In [26] local information is embedded in feature selection through combining instance-based and model-based learning methods. Although all these approaches use local information to determine an optimal feature set, the selected feature set is still forced to model the entire sample space.

3 Proposed 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. The overfitting issue with regard to the proposed algorithm is discussed in Section 3.3.

3.1 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 $\mathbf{x}^{(i)}$.

3.1.1 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:

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

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).

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \in \mathbf{y}^{(i)},\ j \neq i \\ &\max_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \notin \mathbf{y}^{(i)} \end{aligned} \tag{2}$$

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 in Section 3.1.4.

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:

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \in \mathbf{y}^{(i)},\ j \neq i \\ &\max_{\mathbf{f}^{(i)}} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2, j \notin \mathbf{y}^{(i)} \\ &\text{s.t.} \begin{cases} f_m^{(i)} \in \{0,1\}, \quad m = 1, \ldots, M \\ 1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} \leq \alpha \end{cases} \end{aligned} \tag{3}$$

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.

3.1.2 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 (3) is invariant to this replacement. Considering the sample projection definition in (1) and the fact that the problem variables $f_m(i), m = 1, \ldots, M$ are binary, each objective function of (3) can be simplified as follows:

$$\begin{aligned} w_j^{(i)} \left\| \mathbf{x}_p^{(i)} - \mathbf{x}_p^{(j)} \right\|_2^2 &= w_j^{(i)} \left\| \left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)} \right) \otimes \mathbf{f}^{(i)} \right\|_2^2 \\ &= w_j^{(i)} \sum_{m=1}^{M} \left(\delta_{j,m}^{(i)} f_m^{(i)} \right)^2 \\ &= w_j^{(i)} \sum_{m=1}^{M} f_m^{(i)} \delta_{j,m}^{(i)^2} \\ &= w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \end{aligned} \tag{4}$$

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 (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 (5). The problem is then reformulated by simultaneously minimizing the former and maximizing the later.

$$\begin{aligned} &total\ intra-class\ distance: \\ &\qquad \sum_{j \in \mathbf{y}^{(i)}} \left(w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \right) \triangleq \mathbf{a}^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \\ &total\ inter-class\ distance: \\ &\qquad \sum_{j \notin \mathbf{y}^{(i)}} \left(w_j^{(i)} \boldsymbol{\Delta}_j^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \right) \triangleq \mathbf{b}^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \end{aligned} \tag{5}$$

We see that (3) is in the form of an integer program, which is known to be computationally intractable [34]. However this issue is readily addressed through the use of a standard and widely–accepted approximation of an integer programming problem [35], [36], [34]. Here, we replace (relax) the binary constraint in (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 (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.

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} \mathbf{a}^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \\ &\max_{\mathbf{f}^{(i)}} \mathbf{b}^{(i)^{\mathsf{T}}} \mathbf{f}^{(i)} \\ &\text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} \leq \alpha \end{cases} \end{aligned} \tag{6}$$

There are several ways to re-configure a multi-objective problem into a standard form [34], [37], [38] 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 (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 [34]). In this paper, we use the ϵ-constraint method as described by (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

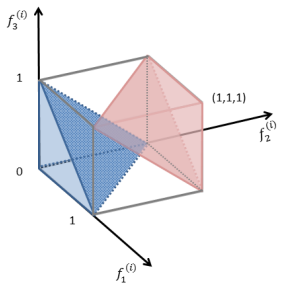

Fig. 2: The polyhedron $\mathcal{P}$ in the case of a 3-D original feature space, i.e. the data dimension M is 3, where α is set to 2. It is a unit cube (defined by $0 \leq f_m^{(i)} \leq 1,\ m = 1, \ldots, 3$) in which two regions, i.e. blue and red pyramids, are removed. The blue pyramid is the intersection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

guarantee the combined inter-class distances are in excess of the value of the parameter $\epsilon^{(i)}$.

$$\min_{\mathbf{f}^{(i)}} \mathbf{a}^{(i)^\mathsf{T}}\mathbf{f}^{(i)} \quad \text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha \\ \mathbf{b}^{(i)^\mathsf{T}}\mathbf{f}^{(i)} \geq \epsilon^{(i)} \end{cases} \tag{7}$$

The parameter $\epsilon^{(i)}$ must be determined such that the optimization problem defined in (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.

3.1.3 Problem feasibility

The optimization problem defined in (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 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). 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:

$$\max_{\mathbf{f}^{(i)}} \mathbf{b}^{(i)^\mathsf{T}}\mathbf{f}^{(i)} \quad \text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}^{(i)} \leq \alpha. \end{cases} \tag{8}$$

Effectively, (8) corresponds to an extreme Pareto point where the weighting given to the intra-class distance term (the first objective in (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:

$$\min_{\mathbf{f}_\beta^{(i)}} \mathbf{a}^{(i)^\mathsf{T}}\mathbf{f}_\beta^{(i)} \quad \text{s.t.} \begin{cases} f_{m,\beta}^{(i)} \in [0,1], m = 1, \ldots, M \\ 1 \leq \mathbf{1}^\mathsf{T}\mathbf{f}_\beta^{(i)} \leq \alpha \\ \mathbf{b}^{(i)^\mathsf{T}}\mathbf{f}_\beta^{(i)} \geq \beta\epsilon_{max}^{(i)}. \end{cases} \tag{9}$$

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 (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 [35], [36], [34] is applied to the optimal point of (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 (9) and provides the minimum value for the objective function of (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 3.2.

3.1.4 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 (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:

$$\begin{aligned} w_j^{(i)} &= \frac{1}{N}\left(\sum_{k=1}^{N} \exp\left(-\left(d_{ij|k} - d_{ij|k}^{min}\right)\right)\right) \\ d_{ij|k} &= \left\|\left(\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right) \otimes \mathbf{f}^{*^{(k)}}\right\|_2 \\ d_{ij|k}^{min} &= \begin{cases} \min\limits_{v \in \mathbf{y}^{(i)}} d_{iv|k} & \text{, if } y^{(j)} = y^{(i)} \\ \min\limits_{v \notin \mathbf{y}^{(i)}} d_{iv|k} & \text{, if } y^{(j)} \neq y^{(i)} \end{cases} \end{aligned} \tag{10}$$

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.

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 (10);
    Compute $\epsilon_{max}^{(i)}$ through solving (8);
    for $\beta \leftarrow 0$ **to** 1 **do**
      Compute $\mathbf{f}_{\beta}^{(i)}$ through solving (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 3.2;
  end
end

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

3.2 Class similarity measurement

A consequence of the localized feature selection approach is that, since there is no common set of features across the sample space, conventional classifiers are inappropriate. We now discuss how to build a classifier for the localized scenario. The proposed classifier structure is based on measuring the similarity of a query datum $\mathbf{x}^q$ to a specific class using the optimal feature sets specified by the $\left\{\mathbf{f}^{\star^{(i)}}\right\}_{i=1}^{N}$.

The proposed method assumes that the sample space consists of N (probably overlapping) regions, where each region is characterized by its representative point $\mathbf{x}^{(i)}$, its class label $y^{(i)}$ and its optimal feature set $\mathbf{f}^{\star^{(i)}}$. We define each region to be a hypersphere $\mathcal{Q}^{(i)}$ in the co–ordinate system defined by $\mathbf{f}^{\star^{(i)}}$, which is centered at $\mathbf{x}_p^{(i)}$. The radius of $\mathcal{Q}^{(i)}$ is determined such that the "impurity level" within $\mathcal{Q}^{(i)}$ is less than the parameter γ. The "impurity level" is the ratio of the normalized number of samples with differing class label to the normalized number of samples with the same class label. In all our experiments, γ is fixed at the value 0.2.

To assess the similarity $S_{Y_\ell}(\mathbf{x}^q)$ of a query datum $\mathbf{x}^q$ to class $Y_\ell \in \mathcal{Y}$, we measure the similarity of $\mathbf{x}^q$ to all regions whose class label is Y_ℓ. To this end we define a set of binary variables $s_i(\mathbf{x}^q), i = 1, \ldots, N$ such that $s_i(\mathbf{x}^q)$ is set to 1 if $\mathbf{x}^q \in \mathcal{Q}^{(i)}$ and the class label of the nearest neighbor of $\mathbf{x}^q$ is $y^{(i)}$; otherwise it is set to 0. The variable $s_i(\mathbf{x}^q)$ may be interpreted as a weak classifier which shows the similarity of $\mathbf{x}^q$ to the ith region. The similarity $S_{Y_\ell}(\mathbf{x}^q)$ is then obtained as follows:

$$S_{Y_\ell}(\mathbf{x}^q) = \frac{\sum_{i \in \mathbb{Y}_\ell} s_i(\mathbf{x}^q)}{\eta_\ell} \tag{11}$$

where $\mathbb{Y}_\ell$ indicates the set of all regions whose class labels are Y_ℓ. The cardinality of $\mathbb{Y}_\ell$ is η_ℓ. After computing the similarity of $\mathbf{x}^q$ to all classes, the class label of $\mathbf{x}^q$ is the one which provides the largest similarity.

If query sample $\mathbf{x}^q$ does not fall in any of the $\mathcal{Q}^{(i)}$s, our desire is to assign its class as the class label of the nearest sample to $\mathbf{x}^q$. The question is "what coordinate system should be used to determine the nearest neighboring sample". To address this matter, we use a majority voting procedure of the class labels within the set of all nearest neighboring samples. This nearest neighbor set consists of those samples which have the nearest distances to the query datum as measured over each of the N local co–ordinate systems. The number of votes for each class is normalized to the number of samples within that class. It is to be noted that on the basis of our experiments, the percentage of such a situation occurring is very rare – only 0.03%.

We now discuss a method for determining a suitable value for β (which corresponds to the selection

of a suitable point in the Pareto set). We examine different values of $\beta \in [0,1]$ in increments of 0.05. For each value, we solve (9) followed by the randomized rounding process. This determines the candidate local co–ordinate system for the respective value of β, i.e. $\mathbf{f}_{\beta}^{\star(i)}$, and therefore specifies the candidate $\mathcal{Q}^{(i)}$ and the weak classifier s_i. The corresponding local clustering performance may then be determined using a leave-one-out cross-validation procedure, using the respective weak classifier results over the training samples situated within the corresponding $\mathcal{Q}^{(i)}$ as a criterion of performance. The Pareto optimal point corresponding to the value of β which yields best local performance is then selected as the binary solution $\mathbf{f}^{\star(i)}$ at the current iteration (see line 11 of Algorithm 1).

3.3 Discussion about overfitting

In the following we discuss the overfitting issue with the proposed method. Let the available feature pool be denoted by the set $\mathcal{X}$. Let us consider the idealized scenario where for each localized region, we can partition $\mathcal{X}$ into the two disjoint sets $\mathcal{X}_R^{(i)}$ and $\mathcal{X}_I^{(i)}$ such that $\mathcal{X}_R^{(i)} \cup \mathcal{X}_I^{(i)} = \mathcal{X},\ i = 1,\ldots,N$. The sets $\mathcal{X}_R^{(i)}$ and $\mathcal{X}_I^{(i)}$ contain only the relevant and irrelevant features, respectively. Let $\eta_R^{(i)}$ denote the cardinality of $\mathcal{X}_R^{(i)}$.

For the time being, let us consider the hypothetical situation where $\alpha = \eta_R^{(i)}$. We note that "relevant" features are those which encourage local clustering behaviour, which is quantified by the optimization problem of (9). We therefore make the assumption that all features in $\mathcal{X}_R^{(i)}$ are sufficiently relevant to be selected as local features by the proposed procedure; i.e., with high probability, they are the solution to (9), followed by the randomized rounding process. If we now let α grow above the value $\eta_R^{(i)}$, features in $\mathcal{X}_I^{(i)}$ become candidates for selection. Because these features do not encourage clustering, then with high probability these features must be given a low f-value in order to satisfy the optimality of (9). Thus there is a low probability that any feature in $\mathcal{X}_I^{(i)}$ will be selected by the randomized rounding procedure. We recall that any solution selected by the randomized rounding procedure must also satisfy the constraints; therefore, such a solution remains feasible, due to the *inequality* constraint involving α in (9). Therefore in this idealized scenario, we see that as α grows, the number of selected features tends to saturate at the value $\eta_R^{(i)}$.

In the more practical scenario, the features may not separate so cleanly into the relevant and irrelevant groups as we have assumed, with the result that "partially relevant" features may continue to be selected as α grows. Therefore the risk of overfitting is not entirely eliminated for real data sets. Nevertheless, as we demonstrate in Section 4, a saturation effect of the number of selected features in real data scenarios is clearly evident.

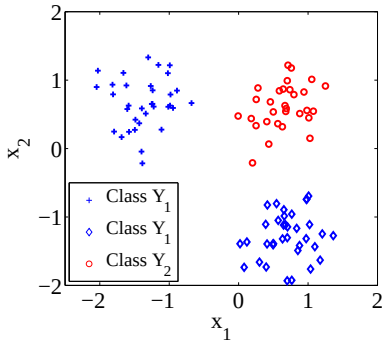

Fig. 3: Illustration of the synthetic data set in terms of its relevant features x_1 and x_2, after feature values are transformed into their z-scores.

In summary, the LFS algorithm inherently tends to select only relevant features and rejects irrelevant features. This imposes a limit on the number of selected features. Thus the LFS method tends to be immune to the overfitting problem. This behavior is in contrast to that of current feature selection methods which inherently do not penalize over-estimation of the number of selected features.

Further, the proposed algorithm deals with the effect of outlier training samples through the aggregation process of (11) where the final decision is based on the *average* of the "weak classifier" results $s_i(\mathbf{x}^q), i = 1,\ldots,N$. Since $s_i(\mathbf{x}^q)$ is either 0 or 1, if the number of outlier samples in each class is much smaller than the number of true samples within that class, as in well-behaved classification problems, then the effect of outlier samples in the final classification result is diminished.

4 Experimental Results

The performance of the proposed method is demonstrated by performing a large-scale experiment on one synthetic data set and ten real-world binary classification problems and is compared against six well-known and state-of-the-art feature selection algorithms including FDA [12], Simba [24], mRMR [7], KCSM [9], Logo [23] and DEFS [29].

As is shown in Fig. 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 [9], 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 [23]. The artificial irrelevant features are

TABLE 2: Characteristics of the real-world data sets used in the experiments.

Data set	# Train	# Test	# Features (M)
Sonar [28]	100	108	60(100)
DNA [9]	100	3086	180(100)
Breast [28]	100	469	30(100)
Adult [39]	100	1505	119(100)
ARR [40]	100	320	278(100)
ALLAML [41]	60	12	7129
Prostate [41]	90	12	5966
Duke-breast [42]	30	12	7129
Leukemia [28]	60	12	7070
Colon [40]	50	12	2000

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

independently sampled from a Gaussian distribution with zero-mean and unit-variance.

Details of real-world data sets are summarized in Table 2. The total number of available samples in each case is the sum of entries in columns 2 (# train) and 3 (# test). Following [23], the performance of the various feature selection algorithms on each data set is evaluated using a bootstrapping algorithm. To this end, each algorithm is run 10 times on each data set. For each run, the number of data points as shown in column 2 of Table 2 is randomly selected to be the training set, and the remaining samples (whose number is indicated in the third column of Table 2) are used as test samples for that run. The average performance over all 10 runs is recorded. For a fair comparison between feature selection algorithms, the training and test sets for each run is common for all algorithms.

To increase the challenge of the classification problem, following [23], the set of original features of the data sets "Sonar", "DNA", "Breast", "Adult" and "ARR" have been augmented by 100 irrelevant features, independently sampled from a zero-mean and unit-variance Gaussian distribution. Data sets "ALLAML", "Prostate", "Duke-breast", "Leukemia" and "Colon" 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 to their z-score values. These real data sets represent applications where expensive feature selection methods such as an exhaustive search cannot be used directly.

The code for our comparison feature selection methods are all available on the respective author's websites, with the exception of KCSM, which was obtained directly from the author. The default settings for each algorithm are used. In the case of the Simba algorithm, following [23], a nonlinear sigmoid activation function is used with sigmoid parameter set to 1.

Apart from the parameter α, which is analogous to the number of selected features in our comparison feature selection algorithms, the proposed method has two additional user–defined parameters: the number of iterations τ (see Section 3.1.4) and the level of impurity γ (see Section 3.2). Generally, these parameters can be estimated through cross validation and be tuned for each data set to provide the most accurate classification results. However, in this case, for a fair comparison, they are not tuned and set respectively to 2 and 0.2, i.e. default values. These values are fixed during all our experiments on all data sets.

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

4.1 Classification accuracy

Since the comparison feature selection algorithms do not inherently incorporate a classifier, an SVM classifier with an RBF kernel is used to estimate the classification accuracy corresponding to the features selected by our comparison feature selection algorithms on each data set. To this end, after performing feature selection on the training samples, an SVM classifier with the top-t selected features is trained with training data and tested on the test data. Default values for the SVM classifier are used for both the training and the test phase. In our experiments, t ranges from 1 to 30 since there is no performance improvement for larger values, with the exception of the data set "Adult".

TABLE 1: Minimum classification error (in percent) and standard deviation (in percent) of the different algorithms. Standard deviations are presented in parentheses.

Data set	LFS	FDA	Simba	mRMR	KCSM	Logo	DEFS	SVM (no feature selection)
Sonar	**22.87(3.92)**	26.11(4.29)	25.24(3.67)	28.70(2.61)	26.85(3.49)	26.75(3.44)	27.81(6.67)	49.90(4.81)
DNA	**13.41(1.88)**	13.94(2.71)	14.43(4.77)	13.75(2.96)	35.95(17.04)	15.35(5.73)	18.68(5.02)	49.70(2.04)
Breast	**6.37(1.33)**	7.71(1.92)	8.89(1.28)	8.29(2.19)	7.73(1.63)	8.25(1.36)	11.01(2.49)	37.61(0.60)
Adult	**22.27(1.46)**	24.65(0.33)	24.65(0.67)	24.75(7.58)	24.85(1.10)	24.53(7.85)	26.37(2.05)	24.65(0.33)
ARR	33.06(2.60)	46.68(9.95)	33.53(6.46)	31.59(3.25)	33.34(9.039)	33.93(5.32)	**31.42(4.73)**	43.68(1.19)
ALLAML	**1.66(3.51)**	5.00(4.30)	25.50(16.49)	7.50(8.28)	32.50(17.76)	7.50(7.29)	14.66(10.50)	38.33(15.81)
Prostate	**4.16(4.39)**	6.66(6.57)	12.66(8.79)	8.33(7.58)	33.33(16.66)	8.33(7.85)	13.66(9.56)	57.50(10.72)
Duke-breast	**10.83(7.90)**	17.50(8.28)	30.83(12.86)	21.66(5.82)	33.33(16.19)	21.66(11.91)	26.66(14.61)	63.33(10.54)
Leukemia	**3.33(4.30)**	5.00(5.82)	12.00(8.06)	5.00(5.82)	32.50(13.86)	6.66(5.27)	16.83(10.85)	35.83(14.19)
Colon	**9.16(0.08)**	11.66(8.95)	34.50(13.58)	19.16(5.62)	12.50(9.00)	20.83(10.57)	26.66(14.12)	36.66(17.21)
Average	**12.71**	16.49	22.22	16.87	27.29	17.38	21.38	43.72

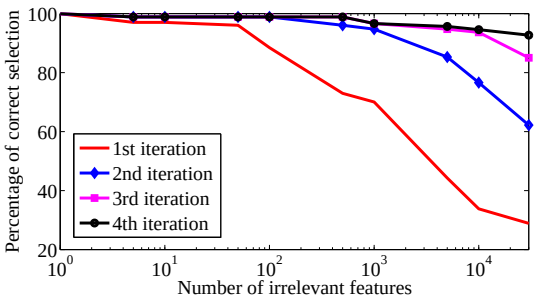

Fig. 4: Percentage of correct feature selection over four successive iterations of the proposed algorithm for the synthetic data set, where the samples are contaminated with a varying number of irrelevant features. The parameter α is set to 2.

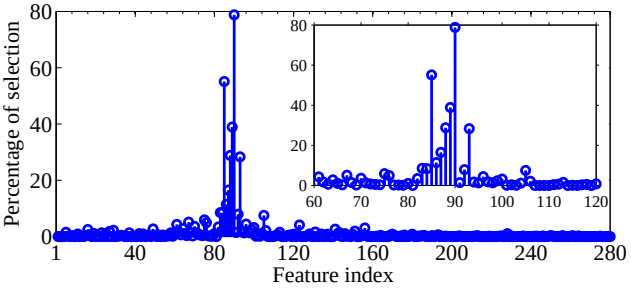

Fig. 5: Selected features for "DNA" data set. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 5.

Here the minimum error rate for all methods, except LFS and Logo, happens out of this range and is equal to the case where no feature selection is performed, i.e., when all candidate features are selected. The performance in this case is 24.65% (see the last column of Table 1 for the Adult data set). For a fair comparison, the parameter α (analogous to t) of the proposed LFS method also ranges from 1 to 30.

The minimum classification error and the corresponding standard deviation as determined by the bootstrapping procedure described earlier is presented in Table 1. For reference, the classification error rate of the SVM classifier performed on the data sets without any feature selection is also reported in the last column of Table 1. Since the performance in this case is generally very low, this result implies that, without feature selection, classification suffers from the presence of irrelevant features and the curse of dimensionality [43]. The best result for each data set is shown in bold. Among the seven algorithms, the proposed LFS algorithm yields the best results in nine out of the ten 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 seven algorithms.

4.2 Iterative weight definition and correct feature selection

As illustrated in Fig. 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. 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 interval between 61 to 120 [44], [9]. This observation provides a good means of evaluating LFS performance on a real world data set. Fig. 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.

4.3 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. 6 and Fig. 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 3.3.

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,

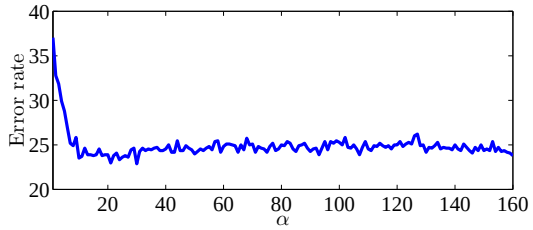

Fig. 6: Classification error rate of the proposed method for data set "Sonar" where the parameter α ranges from 1 to the maximum possible value of $M = 160$.

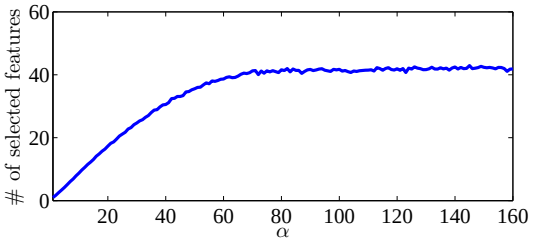

Fig. 7: Averaged cardinality of the optimal feature sets $\mathbf{f}^{*^{(i)}}$ $i = 1, \ldots, N$ versus the parameter α where α ranges from 1 to the maximum possible value of $M = 160$.

the proposed LFS algorithm is not too sensitive to this parameter. Moreover, as illustrated in Fig. 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. 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.

4.4 Overlapping Feature Sets?

The reader may be interested to know if there is any overlap between the optimal feature sets of the representative points. To answer this question, the normalized histogram over all feature sets for the

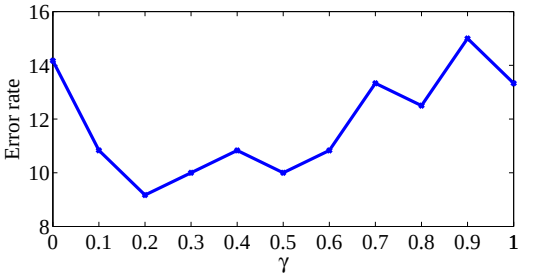

Fig. 8: Classification error rate of the proposed method for data set "Colon" where the parameter γ ranges from 0 to 1.

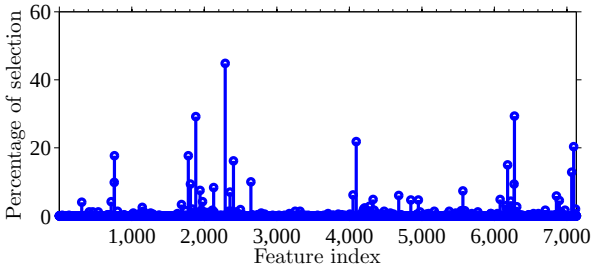

Fig. 9: Selected features for "ALLAML" data set. The Height of each feature index indicates what percentage of representative points select the respective feature as a discriminative feature where α is set to the typical value of 5.

data set "ALLAML" is shown in Fig. 9, where the parameter α is set to a typical value of 5. The height of each feature index indicates what percentage of representative points select the respective feature. As is expected, there are some overlap between region specific feature sets, but it is evident there does not appear to be one common feature set that works well over all regions of the sample space. Indeed, with the proposed method and these results, we assert the assumption of a common feature set over the entire sample space is not necessarily optimal in real world applications.

The most common features may be interpreted as the most informative features in terms of classification accuracy over the sample space. The less common features may be interpreted as being informative features, but only relevant for a small group of samples; e.g. in the context of biology/genetics applications, the less common features may be interpreted as being important in the discrimination of some small subpopulation of samples.

One may be interested to know the classification accuracy in the context of a global selection scheme; i.e., we select the top 5 dominant features from Fig. 9 as produced by the LFS method, and then feed them into an SVM classifier. Using such a feature set, the error rate is 6.66% which is in the range of that of the other methods, but nevertheless significantly greater than the error rate (1.66%) corresponding to the proposed LFS region-specific feature selection method (see Table 1). This example is a further demonstration of the effectiveness of modeling the feature space locally.

4.5 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 3.1, the normalized histogram over the ℓ_1–norm distances between the relaxed solutions and their corresponding binary solutions is shown in Fig. 10. The height of each bar indicates what fraction of the representative points have the corresponding

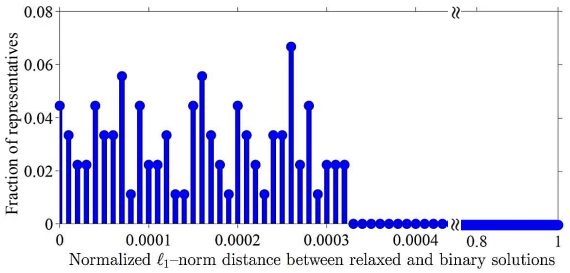

Fig. 10: Histogram of distances between relaxed solutions and their corresponding binary solutions for data set "Prostate" where α is set to the typical value of 5.

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.

4.6 CPU time:

The computational complexity for computing a feature set for each representative point depends mainly on the data dimension. Fig. 11 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 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 weather 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 it

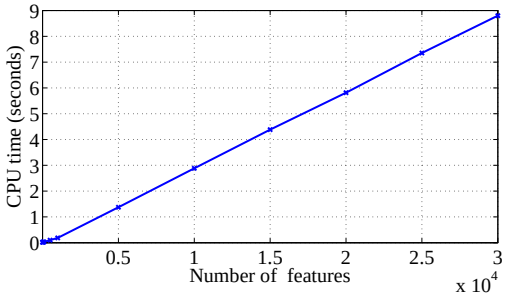

Fig. 11: The CPU time (seconds) taken by the proposed algorithm to perform feature selection for one representative point $\mathbf{x}^{(i)}$ with a given β on the synthetic data set where the parameter α is set to 2.

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

5 Conclusions

In this paper we present an effective and practical method for local feature selection for application to the data classification problem. Unlike most feature selection algorithms which pick a "global" subset of features which is most representative for the given data set, the proposed algorithm instead picks "local" subsets of features that are most informative for the small region around the data points. The cardinality and identity of the feature sets can vary from data point to data point. The process of computing a feature set for each region is independent of the others and can be performed in parallel.

The LFS procedure is formulated as a linear program, which has the advantage of convexity and efficient implementation. The proposed algorithm is shown to perform well in practice, compared to previous state-of-the-art feature selection algorithms. Performance of the proposed algorithm is insensitive to the underlying distribution of the data. Furthermore we have demonstrated that the method is relatively invariant to an upper bound on the number of selected features, and so is robust against the overfitting phenomenon.

Acknowledgement

The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and MITACS.

References

[1] I. K. Fodor, "A survey of dimension reduction techniques," 2002.
[2] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 4–37, 2000.
[3] P. Langley, *Selection of relevant features in machine learning*. Defense Technical Information Center, 1994.
[4] A. R. Webb, *Statistical pattern recognition*. Wiley, 2003.
[5] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.
[6] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[7] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.
[8] H.-L. Wei and S. A. Billings, "Feature subset selection and ranking for data dimensionality reduction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 162–166, 2007.
[9] L. Wang, "Feature selection with kernel class separability," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 9, pp. 1534–1546, 2008.
[10] H. Zeng and Y.-m. Cheung, "Feature selection and kernel learning for local learning-based clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1532–1547, 2011.

[11] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *Neural Networks, IEEE Transactions on*, vol. 13, no. 1, pp. 143–159, 2002.

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

[13] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4, pp. 411–430, 2000.

[14] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[15] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[16] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[17] Y. M. Lui and J. R. Beveridge, "Grassmann registration manifolds for face recognition," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 44–57.

[18] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1208–1213.

[19] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton, "Pattern classification using a mixture of factor analyzers," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*. IEEE, 1999, pp. 525–534.

[20] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 1, no. 1, pp. 24–45, 2004.

[21] Y. Cheng and G. M. Church, "Biclustering of expression data." in *Ismb*, vol. 8, 2000, pp. 93–103.

[22] I. S. Dhillon, "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*. ACM, 2001, pp. 269–274.

[23] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1610–1626, 2010.

[24] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 43.

[25] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*. Morgan Kaufmann Publishers Inc., 1992, pp. 249–256.

[26] Z. Liu, W. Hsiao, B. L. Cantarel, E. F. Drábek, and C. Fraser-Liggett, "Sparse distance-based learning for simultaneous multiclass classification and feature selection of metagenomic data," *Bioinformatics*, vol. 27, no. 23, pp. 3242–3249, 2011.

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

[28] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *The Journal of Machine Learning Research*, vol. 13, pp. 27–66, 2012.

[29] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Feature subset selection using differential evolution and a statistical repair mechanism," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11 515–11 526, 2011.

[30] I. Kononenko, "Estimating attributes: analysis and extensions of relief," in *Machine Learning: ECML-94*. Springer, 1994, pp. 171–182.

[31] Y. Sun, "Iterative relief for feature weighting: algorithms, theories, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1035–1051, 2007.

[32] B. Chen, H. Liu, J. Chai, and Z. Bao, "Large margin feature weighting method via linear programming," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 10, pp. 1475–1488, 2009.

[33] B. Liu, B. Fang, X. Liu, J. Chen, and Z. Huang, "Large margin subspace learning for feature selection," *Pattern Recognition*, 2013.

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[35] M. T. Thai, "Approximation algorithms: Lp relaxation, rounding, and randomized rounding techniques," Lecture Notes, University of Florida, 2013.

[36] A. Souza, "Randomized algorithm & probabilistic methods," Lecture Notes, Humboldt University of Berlin, 2001.

[37] C. L. Hwang, A. S. M. Masud *et al.*, *Multiple objective decision making-methods and applications*. Springer, 1979, vol. 164.

[38] G. Mavrotas, "Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems," *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009.

[39] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[40] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "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*, vol. 96, no. 12, pp. 6745–6750, 1999.

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

[42] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Marks, and J. R. Nevins, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11 462–11 467, 2001.

[43] R. E. Bellman and S. E. Dreyfus, "Applied dynamic programming," 1962.

[44] G. John. Dna dataset (statlog version) - primate splice-junction gene sequences (dna) with associated imperfect domain theory. [Online]. Available: https://www.sgi.com/tech/mlc/db/DNA.names