Logistic Localized Modeling of the Sample Space for Feature Selection and Classification

Narges Armanfard, James P. Reilly, *Member, IEEE*, and Majid Komeili

***Abstract*— Conventional feature selection algorithms assign a single common feature set to all regions of the sample space. In contrast, this paper proposes a novel algorithm for *localized* feature selection for which each region of the sample space is characterized by its individual distinct feature subset that may vary in size and membership. This approach can therefore select an optimal feature subset that adapts to local variations of the sample space, and hence offer the potential for improved performance. Feature subsets are computed by choosing an optimal coordinate space so that, within a localized region, within-class distances and between-class distances are, respectively, minimized and maximized. Distances are measured using a logistic function metric within the corresponding region. This enables the optimization process to focus on a localized region within the sample space. A local classification approach is utilized for measuring the similarity of a new input data point to each class. The proposed logistic localized feature selection (lLFS) algorithm is invariant to the underlying probability distribution of the data; hence, it is appropriate when the data are distributed on a nonlinear or disjoint manifold. lLFS is efficiently formulated as a joint convex/increasing quasi-convex optimization problem with a unique global optimum point. The method is most applicable when the number of available training samples is small. The performance of the proposed localized method is successfully demonstrated on a large variety of data sets. We demonstrate that the number of features selected by the lLFS method saturates at the number of available discriminative features. In addition, we have shown that the Vapnik–Chervonenkis dimension of the localized classifier is finite. Both these factors suggest that the lLFS method is insensitive to the overfitting issue, relative to other methods.**

***Index Terms*— Convex optimization, data classification, feature selection, local feature selection, local sample space modeling, quasi-convex optimization.**

I. 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" [1]. It is well known that the performance of typical classifiers notably drops when the number of available objects is not adequate in comparison with the number of candidate features [2]. 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 are used for the identification of cancer-associated gene expression profiles of prognostic or diagnostic value [3]–[5]. 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 [6]–[9] which is also called subspace learning. The second category is feature selection [10]–[15]. Feature extraction approaches, such as principal component analysis (PCA) [7], linear discriminant analysis (LDA) [16], and independent component analysis (ICA) [17], 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 paper, 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 [18]–[20] 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 off-line 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 [21]–[23]. This paper considers batch algorithms.

From another point of view, conventional feature selection algorithms assume that all the regions of sample space can be optimally characterized by a common subset of features [10],

Manuscript received April 29, 2016; revised November 9, 2016 and February 16, 2017; accepted February 17, 2017.

N. Armanfard and J. P. Reilly are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4L8, Canada (e-mail: farmanfn@mcmaster.ca; reillyjg@mcmaster.ca).

M. Komeili is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 2E4, Canada (e-mail: mkomeili@ece.utoronto.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2676101

[12], [24]–[27]. 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 [10], a common subset of features is selected using a mutual information-based approach that utilizes a minimal-redundancy maximal-relevance criterion. In [28], redundancy among features is measured based on normalized mutual information where the authors claim that their method is a an enhancement over [10]. In [29], 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. Paper [30] presents an algorithm to learn local causal structures 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 [12], a common discriminative feature subset is obtained by maximizing a class separability criterion. In [27], a differential-evolution-based algorithm is used for computing a common feature set. The Fisher criterion is used in [16] where each feature score is computed based on minimizing intraclass distances and maximizing interclass distances. In [31], 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 [32], a common set is computed through a combination of linear discriminant analysis and sparsity regularization. In [33], a feature subset is determined based on two criteria designed for the optimization of the support vector machine (SVM), including kernel target alignment and kernel class separability. In [34], 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 [35], combines the algorithmic ideas of least angle regression (LARS) [36], the computational benefits of ridge regression, and the tendency toward sparse solutions of the LASSO. In [37], a feature selection method for microarray data classification is presented that is based on partial least squares and theory of reproducing kernel Hilbert space [38].

The second group applies local information of the sample space for computing an optimal feature subset [5], [39]–[44]. For example, biclustering approaches [45]–[47] use local information for simultaneously clustering data and features. In [48], data clustering is realized through a greedy feature selection algorithm, which can assign a specified feature set to each cluster. However, these algorithms 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 [39], 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 [41] 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 nonconvex and hence is characterized by the presence of local minima. In [5], a local learning-based feature selection method is presented in which a complex nonlinear problem is decomposed into a set of locally linear problems. In [49], 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., nonstationarities), across the sample space. These variations could be the result of a change in operating conditions of the underlying generative process. In this paper, 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 logistic localized feature selection (lLFS) algorithm. We therefore present a localized classification procedure that has been adapted to the proposed scenario. We refer to the proposed algorithm as the lLFS method.

The proposed lLFS approach has several advantages. First, it accommodates nonstationarities in the underlying data distribution, because no assumptions are made about the distribution of data over the sample space. Therefore, lLFS allows irregular and/or disjoint distributions of samples. The lLFS method is also effective when the sample space lies on a nonlinear manifold, since an optimal feature subset can be selected to fit the local behavior in each region of the manifold. Second, the lLFS 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 [10], [27], and [29], 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 lLFS algorithm limits the number of selected features only to those features which are most discriminative, and so in this sense is less vulnerable than other methods to overfitting. Furthermore, with regard to classification, we investigate the Vapnik–Chervonenkis (VC) dimension for the proposed

classifier structure. Under certain assumptions, we show that the value of the VC dimension for the lLFS classifier is moderate. A modest value of the VC dimension also implies reduced sensitivity to overfitting. In addition, the lLFS algorithm is formulated as a pair of optimization problems, one of which is convex and the other is an increasing quasi-convex problem where both have a unique global optimum point. Thus, the lLFS algorithm converges to the optimal solution regardless of the initialization point.

We do not promote the off-line training phase of the lLFS method as being computationally fast (although it is considerably faster in the test phase). Rather, its advantages are with respect to performance. The performance of the method is demonstrated with multiple data sets over a range of N (number of training samples) and M (number of candidate features).

II. Proposed Method

A. Overview

Let $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N} \subset \mathbb{R}^{\mathrm{M}} \times \mathcal{Y}$ 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 $\mathbf{x}^{(i)}$.

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 discriminative 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 discriminative features for the neighboring region of $\mathbf{x}^{(i)}$, all the elements of $\mathbf{f}^{(i)}$ are zero except the second and fourth ones. Thus, $\mathbf{f}^{(i)}$ defines a local coordinate 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 coordinate system, which is determined by $\mathbf{f}^{(i)}$, which is unknown at the problem outset. In the early version of this paper, presented in [50] and [51], the neighboring samples are mainly determined 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 paper, the distance measurement problem is alleviated, since the underlying optimization problem is formulated such that distances are a function of the unknown vector $\mathbf{f}^{(i)}$.

In Section II-B, we present the proposed algorithm for computing the binary discriminative feature subset corresponding to the representative point $\mathbf{x}^{(i)}$. In Section II-C, we present a simple yet effective classification process through the aggregation of multiple weak classifier results, which are based on the region-specific feature subsets. The properties of the proposed method are presented in Section III. Performance of the proposed algorithm, on eleven synthetic and real-world data sets, is demonstrated in Section IV. The conclusions are drawn in Section V.

B. Feature Selection

This section is organized as follows. Section II-B1 presents the proposed formulation for local feature selection. The accompanying optimization problem is treated in Section II-B2. A procedure for determining the two required parameters of the proposed formulation is presented in Section II-B3.

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 paper, 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.

1) Other samples of the same class cluster as closely as possible around $\mathbf{x}_p^{(i)}$, and simultaneously.
2) 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

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

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

The functions $\mathcal{U}_1$ and $\mathcal{U}_2$ may be regarded as local intraclass and interclass 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., $\mathbf{a}_j^{(i)} = |\mathbf{x}^{(i)} - \mathbf{x}^{(j)}|$, 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 Section II-B3. 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 following optimization problem:

$$\begin{aligned} &\min_{\mathbf{f}^{(i)}} \mathcal{U}_1(\mathbf{f}^{(i)}) \\ &\max_{\mathbf{f}^{(i)}} \mathcal{U}_2(\mathbf{f}^{(i)}) \\ &\text{s.t.} \begin{cases} f_m^{(i)} \in \{0, 1\}, & m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T\mathbf{f}^{(i)} \leq \alpha. \end{cases} \end{aligned} \quad (2)$$

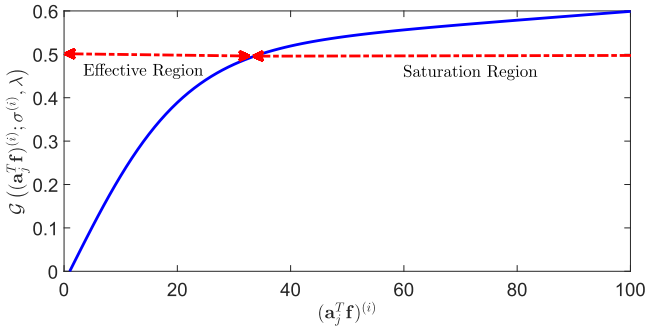

Fig. 1. Function $\mathcal{G}(\cdot)$, which is a shifted logistic function with an additional linear term, where the parameters $\sigma^{(i)}$ and λ are set to the typical values 0.1 and 0.001, respectively.

Some constraints are considered in (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, and 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. 1), which for the purposes of this paper, is defined as

$$\mathcal{G}(z;\sigma,\lambda) = \frac{1}{1+\exp(-\sigma z)} - 0.5 + \lambda z. \tag{3}$$

Since optimization algorithms in general are gradient driven, the 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) may be neglected for the time being. In this case, the gradient of the logistic function for the large-distance samples in (1a) and (1b) (i.e., those in the saturation region shown in Fig. 1) have a small value and hence does not contribute significantly to changes in $\mathcal{U}_1$ and $\mathcal{U}_2$ at the next iteration. On the other hand, terms for which the quantity $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ have 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 (2) thus becomes approximately equivalent to

$$\begin{aligned}
&\min_{\mathbf{f}^{(i)}} \frac{1}{n-1} \sum_{j\in \text{RoL},\, y^{(j)}=y^{(i)}} \left(\mathbf{a}_j^T\mathbf{f}\right)^{(i)} \\
&\max_{\mathbf{f}^{(i)}} \frac{1}{N-n} \sum_{j\in \text{RoL},\, y^{(j)}\neq y^{(i)}} \left(\mathbf{a}_j^T\mathbf{f}\right)^{(i)} \\
&\text{s.t.} \begin{cases} f_m^{(i)} \in \{0,1\}, & m = 1,\ldots,M \\ 1 \leq \mathbf{1}^T\mathbf{f}^{(i)} \leq \alpha \end{cases}
\end{aligned} \tag{4}$$

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 $\mathcal{G}$ is considered as the *region of locality* of the point $\mathbf{x}_p^{(i)}$.

Therefore, within $\mathcal{S}^{(i)}$, through the objective functions of (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) 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 coordinate system in the 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 paper, following [5], we use the ℓ_1 distance, because it provides a linear combination of the featurewise distances (with no transformation), which preserves the logistic function behavior with respect to each elemental distance measure.

2) Optimization Process: The optimization problem posed by (2) is a discrete binary program and hence is computationally intractable [52]. 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 [52]–[54]. Here, the notation $[\cdot]$ denotes a continuous interval, whereas $\{\cdot\}$ denotes a binary set, as before. The randomized rounding procedure (to be discussed further) maps the linear solution back onto a suitable point on the binary grid.

The optimization problem defined in (2) is a multiobjective optimization problem. In this case, the concept of optimality is replaced with Pareto optimality [52]. One approach to solving a multiobjective optimization problem is to linearly combine each of the objective functions into a single objective function. The solution of a multiobjective problem is therefore not unique and consists of the set of all Pareto optimal points, each of which may be obtained through different weightings of the individual objective functions. A Pareto optimal solution is one in which an improvement in one objective function results in the degradation of another. The *set* of Pareto optimal points is unique and independent of the methodology by which the two objective functions are treated [52].

In this paper, the individual objective functions are combined using the concept of the ϵ-constraint [55] as shown in

$$\begin{aligned}
&\min_{\mathbf{f}^{(i)}} \mathcal{U}_1(\mathbf{f}^{(i)}) \\
&\text{s.t.} \begin{cases} f_m^{(i)} \in [0,1], & m = 1,\ldots,M \\ 1 \leq \mathbf{1}^T\mathbf{f}^{(i)} \leq \alpha \\ \mathcal{U}_2(\mathbf{f}^{(i)}) \geq \epsilon^{(i)}. \end{cases}
\end{aligned} \tag{5}$$

Here, the interclass distance measure [relating to $\mathcal{U}_2$ in (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 interclass 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 (5) is feasible. Equation (5) is

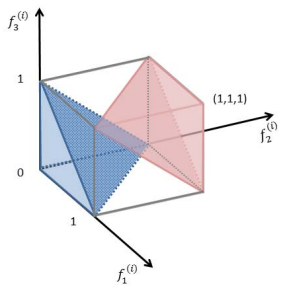

Fig. 2. Intersection of a unit cube and two half-spaces, in the case of a 3-D feature space, i.e., $M = 3$, where the parameter $\alpha = 2$. The unit cube is defined by $0 \leq f_m^{(i)} \leq 1$, $m = 1, \ldots, 3$. The blue and red pyramids are the intersections of the half-spaces, respectively defined by $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} < 1$ and $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} > \alpha$, and the unit cube. The polyhedron $\mathcal{P}$ (white region of cube) is a unit cube in which the red and the blue pyramids have been removed.

feasible if its constraint set is nonempty. In the following, we present an effective approach to specify a value for $\epsilon^{(i)}$ that guarantees feasibility.

The constraints $f_m^{(i)} \in [0, 1], m = 1, \ldots, M$ indicate that the optimum solution must be within an M-dimensional unit hypercube. The two constraints $1 \leq \mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)}$ and $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} \leq \alpha$ indicate that the optimum point must also be inside the space bounded by the two parallel hyperplanes $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} = 1$ and $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} = \alpha$ which is also nonempty, because by definition, the integer parameter α is greater than or equal to 1. Hence, the optimum point must be inside the intersection of the unit hypercube and the space between the parallel hyperplanes. This intersection defines a nonempty polyhedron $\mathcal{P}$. In fact, the polyhedron $\mathcal{P}$ is a unit cube in which two parts have been removed: the intersection between the unit cube and the half-space $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} > \alpha$ and the intersection between the unit cube and $\mathbf{1}^{\mathsf{T}}\mathbf{f}^{(i)} < 1$. Note that the maximum value that α can take is equal to the total number of available features, i.e., M. For an illustration of the geometry of $\mathcal{P}$, see Fig. 2.

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

$$\epsilon_{\max}^{(i)} = \max_{\mathbf{f}^{(i)}} \mathcal{U}_2(\mathbf{f}^{(i)}) \quad \text{s.t.} \begin{cases} 0 \leq f_m^{(i)} \leq 1, & m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T\mathbf{f}^{(i)} \leq \alpha. \end{cases} \tag{6}$$

Finally, the parameter $\epsilon^{(i)}$ in (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 (1a) and (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 $\{\mathbf{f}_\beta^{(i)}\}_{\beta\in[0,1]}$. The problem of interest now becomes

$$\min_{\mathbf{f}_\beta^{(i)}} \mathcal{U}_1\big(\mathbf{f}_\beta^{(i)}\big) \quad \text{s.t.} \begin{cases} f_{m,\beta}^{(i)} \in [0, 1], & m = 1, \ldots, M \\ 1 \leq \mathbf{1}^T\mathbf{f}_\beta^{(i)} \leq \alpha \\ \mathcal{U}_2\big(\mathbf{f}_\beta^{(i)}\big) \geq \beta\epsilon_{\max}^{(i)}. \end{cases} \tag{7}$$

The optimum point obtained from solving (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 (2), i.e., the solution $\mathbf{f}_\beta^{(i)}$ to (7) must be snapped onto a binary grid. This procedure is performed by applying a randomized rounding process [52]–[54] to $\mathbf{f}_\beta^{(i)}$ 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 $m = 1, \ldots, M$. In order to explore the entire region surrounding $\mathbf{f}_\beta^{(i)}$, 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 (7), as well as satisfying all constraints.

The final value $\mathbf{f}^{*(i)}$, corresponding to the best value of β from the set $\{\mathbf{f}_\beta^{*(i)}\}_{\beta\in[0,1]}$, 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 discussed in Section II-C.[1]

Algorithm 1 presents the pseudocode 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 III-A), the initial point does not affect the solution, although it may affect the computational time.

3) 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

$$\frac{1}{1 + \exp(-\sigma^{(i)}\varphi^{(i)})} - 0.5 = 0.47 \tag{8}$$

where

$$\varphi^{(i)} = \max_{j=1:N, j\neq i} \big\{\big(\mathbf{a}_j^T\mathbf{f}_\beta\big)^{(i)}\big\}.$$

The number 0.47 above is chosen to be representative of the knee point of $\mathcal{G}(\cdot)$ (see Fig. 1). The intuition behind (8) is that no sample should fall within the saturation region during the first iteration of the optimization process, so that effectively all the samples are considered by the objective function of (7).

[1]Throughout this paper, our use of the term "optimal" refers to the solution of (7) followed by the randomized rounding process.

Algorithm 1 Pseudocode of the Proposed Feature Selection Algorithm

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 (8) using the initial values;
  Compute $\epsilon_{max}^{(i)}$ through solving (6);
  for $\beta \leftarrow 0$ **to** 1 **do**
    Compute $\mathbf{f}_{\beta}^{(i)}$ through solving (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 $\{\mathbf{f}_{\beta}^{\star(i)}\}_{\beta\in[0,1]}$ which yields the best local clustering performance as explained in Section II-C;
end

The parameter λ controls the contribution of the samples that are in the saturation region (see Fig. 1). The addition of the linear term in (3) allows potentially close samples that are far from $\mathbf{x}_p^{(i)}$ 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 the 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 (8). Hence, as α grows, the slope of the saturation region, i.e., λ, should decrease. Thus, in our experiments, the value of λ is set heuristically according to the value $(0.01/\alpha)$. 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 Section IV were set according to this procedure and were not tuned to improve performance.

C. 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 that is appropriate for the localized scenario. The proposed localized classifier classifies query data $\mathbf{x}^q$ based on measuring distances in the induced feature spaces (i.e., all N frames) 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 hypersphere $\mathcal{Q}^{(i)}$ centered at $\mathbf{x}_p^{(i)}$ (i.e., the projection of $\mathbf{x}^{(i)}$ into the ith frame $\mathcal{S}^{(i)}$) with class label $y^{(i)}$. In this paper, we determine the radius, i.e., $r^{(i)}(\gamma)$, of $\mathcal{Q}^{(i)}$, such that the "impurity level" within the hypersphere $\mathcal{Q}^{(i)}$ is not greater than the user-defined parameter γ. The "impurity" level is the ratio of the number of interclass samples within $\mathcal{Q}^{(i)}$ to the number of intraclass samples within $\mathcal{Q}^{(i)}$. In all our experiments, γ is fixed at the value of 0.2.

The similarity $S_{Y_\ell}(\mathbf{x}^q; \gamma)$ of query data $\mathbf{x}^q$ to class $Y_\ell \in \mathcal{Y}$ is measured based on how many hyperspheres 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:

$$s^{(i)}(\mathbf{x}^q; \gamma) = \text{step}\left[r^{(i)}(\gamma) - \left\|\mathbf{x}_p^{(i)} - \mathbf{x}_p^{(q)}\right\|_2\right] \tag{9}$$

where

$$\text{step}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

$s^{(i)}(\mathbf{x}^q; \gamma)$ may be interpreted as "weak" classifiers that indicate the similarity of $\mathbf{x}^q$ to the corresponding region.[2] The similarity $S_{Y_\ell}(\mathbf{x}^q; \gamma)$ of $\mathbf{x}^q$ to the class Y_ℓ is computed through aggregation of the "weak" classifier results corresponding to the regions whose class labels are the same as Y_ℓ as follows:

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

where $\mathbb{Y}_\ell$ indicates the set of all the 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

$$y^q = \underset{Y_l\in\mathcal{Y}}{\text{argmax}}\{S_{Y_1}, S_{Y_2}, \ldots, S_{Y_c}\}. \tag{12}$$

If $\mathbf{x}^q$ is not situated in any of the hyperspheres $\mathcal{Q}^{(i)}$ $i = 1, \ldots, N$, 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 coordinate 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 coordinate 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 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 0.009%.

In the following, we discuss an approach to determine an appropriate value for β, which results in the selection of a

[2]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 (9) 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.

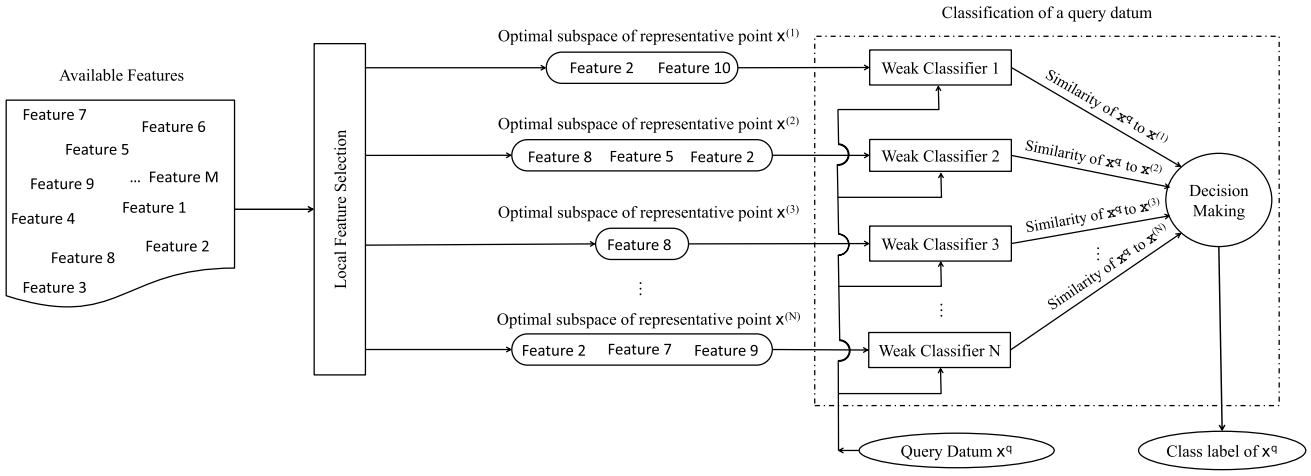

Fig. 3. Block diagram of the proposed algorithm for data classification. The neighboring region of each representative point is modeled by an optimal feature subset selected from the available feature pool. Details of the local feature selection and classification procedures for query data $\mathbf{x}^q$ are presented in Sections II-B and II-C, respectively.

suitable point in the Pareto set. We solve (7) 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 the increments of 0.05. Each candidate binary vector $\mathbf{f}_\beta^{\star(i)}$ defines a local coordinate system and therefore specifies the respective hypersphere $\mathcal{Q}^{(i)}$ 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 $\{\mathbf{f}_\beta^{\star(i)}\}_{\beta\in[0,1]}$, 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 10 of Algorithm 1).

Fig. 3 shows a block diagram of the proposed algorithm.

III. Properties of the Proposed Algorithm

In this section, we present four important properties of the proposed approach defined in Section II. These properties are that: 1) the optimization problems [in (6) and (7)] of the proposed feature selection algorithm are convex and quasi-convex and hence have unique global optimums; 2) the proposed localized classifier defined in Section II-C has a modest VC dimension; 3) the proposed approach is insensitive to the overfitting problem; and 4) the proposed feature selection method may be parallelized.

A. Problem Convexity

In this section, we discuss the convexity property of the optimization problems defined in (6) and (7). By definition, $(\mathbf{a}_j^T\mathbf{f})^{(i)}$ is always positive; hence, the terms $\mathcal{G}((\mathbf{a}_j^T\mathbf{f})^{(i)}; \sigma^{(i)}, \lambda)$ in (1a) and (1b) are always positive. Thus, the function $\mathcal{G}$ is both concave and increasing quasi-convex (see Fig. 1) [52]. Equation (6) defines an optimization problem whose objective function is concave, because it is the summation of $N-n$ concave functions [see (1b)]. The constraint set is linear and hence defines a convex feasible set. Thus, (6) is a convex problem.

The objective function of (7) is a strictly increasing quasi-convex function, since it is the summation of $n-1$ strictly increasing quasi-convex functions [see (1b)]. The constraint set of (7) is convex and feasible. Therefore, (7) defines a quasi-convex problem with a unique global minimum [52]. Since both the 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.

B. Vapnik–Chervonenkis Dimension

The VC dimension [56] 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 overfitting, compared with 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 $\theta = [\theta_1; \theta_2]$. Consider a training set $\mathbf{X}_N = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N \subset \mathbb{R}^{\mathrm{M}} \times \mathcal{Y}$. 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 $M+1$ [57]. The VC dimension plays an important role in establishing bounds on the performance of the classifier.

The VC dimension h for the lLFS classifier is developed in the Appendix, and under certain modest assumptions, is shown to be equal to the value $L(\lceil(1/\gamma)\rceil - 1)$, where L is the number of clusters in the training set and $\lceil\cdot\rceil$ denotes the ceiling function.

The fact that the lLFS 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 [56]. In this respect and under the assumption that all training points are drawn independent and identically distributed (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 draw from the same distribution, we can define an *empirical* risk and an *expected* risk [56], [57], respectively, as follows:

$$\mathcal{R}_N(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}|y^{(i)} - f(\mathbf{x}^{(i)};\theta)| \tag{13}$$

$$\mathcal{R}(\theta) = \int \frac{1}{2}|y - f(\mathbf{x};\theta)|d\mathcal{D}(\mathbf{x}, y). \tag{14}$$

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

$$\mathcal{R}(\theta) \leq \mathcal{R}_N(\theta) + \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right)+1\right) - \log\left(\frac{\xi}{4}\right)}{N}}. \tag{15}$$

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 [56] 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 optimal risk $\mathcal{R}^\star$ as follows:

$$\mathcal{R}^\star(\theta) = \inf_{\theta}\mathcal{R}(\theta). \tag{16}$$

Then, a good training algorithm will generate an $\mathcal{R}_N(\theta)$ close to $\mathcal{R}^\star(\theta)$, or more precisely [58], for a positive real number $\rho \in [0, 1]$, which is prescribed in advance, we have

$$\Pr_{\mathbf{X}_N\sim\mathcal{D}^N}\{\mathcal{R}_N(\theta) < \mathcal{R}^\star(\theta) + \rho\} \geq 1 - \tau \tag{17}$$

where $\tau \in [0, 1]$ 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 [58] as follows:

$$N_o(\rho, \tau) \leq \frac{64}{\rho^2}\left(2h\log\left(\frac{12}{\rho}\right) + \log\left(\frac{4}{\tau}\right)\right). \tag{18}$$

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 [57]. 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. Furthermore, (18) suggests that the number of training samples required to guarantee a certain level of performance varies only logarithmically with the parameters τ and ρ. Both these points suggest that with the lLFS 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.

C. lLFS and the Overfitting Issue

Both the feature selection and classification processes contribute to the overfitting problem. As is discussed in Section III-B and the Appendix, the lLFS 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 overfitting than a method with a high or infinite value of h [57].

We now discuss the lLFS 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 (7). 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 (7) 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 (7) 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 (7). 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 IV-F, the saturation behavior of the number of selected features is clearly evident in real-world scenarios.

In summary, the proposed lLFS feature selection method chooses only relevant features. In this respect, it is less vulnerable to overfitting than methods that 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 lLFS feature selection and classifier procedures are insensitive to the overfitting problem in the sense we have indicated.

D. lLFS 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.

IV. Experimental Results

In real-world applications, obtaining labeled examples to be used as training samples is often very expensive and time consuming, as it can require the effort of human annotators, who must often be quite skilled. However, small sample sizes, and their inherent risk of imprecision and overfitting, pose a significant challenge for many modeling problems [59]–[62]. Hence, most of the real-world data sets used in our experiments have a small value of N, with M varying over a range of values. The performance of the lLFS method on data sets with large N is not the focus of this paper; nevertheless, we discuss this case in Section IV-J.

A. Experimental Setup

In this section, we perform several experiments on 1 synthetic and 11 binary real-world data sets to demonstrate the effectiveness of the proposed feature selection algorithm. The proposed algorithm is compared with the eight state-of-the-art feature selection algorithms: Logo[3] [5], FMS[4] [31], MBEGA[5] [29], Elasticnet[6] (based on LARS-EN) [35], kPLS[7] [37], MetaDistance[8] [49], DEFS[9] [27], and mRMR[10] [10], where the first seven methods are specifically developed for the sparse data case where the number of available training samples is low in comparison with the number of variables. For a fair comparison, the parameters of all these feature selection algorithms as well as those of the proposed lLFS 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 corresponding 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 corresponding to the l_2 and l_1 penalties for use in the test phase (see [63] 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 [10], [12], [27], [29], [31], [41], and [64]–[66], the SVM classifier parameters are set to their default values (in MATLAB). To provide a fair comparison, the 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 Core i7-2600 CPU at 3.4 GHz and 16-GB RAM.

[3]http://plaza.ufl.edu/sunyijun/PAMI2.htm

[4]http://www2.cs.siu.edu/ ~ qcheng/featureselection_pubfolder/index.html

[5]http://csse.szu.edu.cn/staff/zhuzx/MAFS.html

[6]http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3897

[7]https://github.com/sqsun/kernelPLS

[8]http://metadistance.igs.umaryland.edu/

[9]http://www.mathworks.com/matlabcentral/fileexchange/30877-differential-evolution-based-channel-and-feature-selection

[10]http://penglab.janelia.org/proj/mRMR/

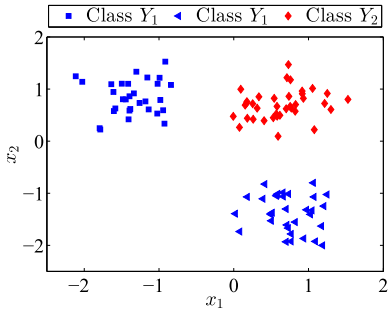

Fig. 4. Illustration of the synthetic data set in terms of its discriminative features x_1 and x_2.

TABLE I

Characteristics of the Real-World Data Sets Used in the Experiments

Data set	# Train	# Test	# Features (M)
Sonar [26]	100	108	60(100)
DNA [12]	100	3086	180(100)
Breast [26]	100	469	30(100)
Adult [67]	100	1505	119(100)
ARR [68]	100	320	278(100)
Prostate [69]	90	12	5966
Duke-breast [70]	30	12	7129
Leukemia [26]	60	12	7070
Colon [68]	50	12	2000
Nervous system [71]	48	12	7129
DNA_l [12]	1000	2186	180(100)

The number of artificially added irrelevant features is indicated in parentheses. The only difference between data set "DNA_l" and "DNA" is the number of training samples assigned.

B. Data Sets

We present our results using both synthetic and real-world data sets. The synthetic, or "toy" data set, as is shown in Fig. 4, is distributed in a 2-D feature space defined by x_1 and x_2 in which class Y_1 has two disjoint subclasses shown by ■ and ◄, whereas samples of class Y_2, shown by ♦, have a unimodal 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 discriminative features x_1 and x_2, following [12], each sample is artificially contaminated by augmenting it with 100 *i.i.d* irrelevant features drawn from a standard normal distribution.

The characteristics of the real-world data sets used for the experiments are summarized in Table I, where the first ten have a small number of training samples, whereas the last one has a relatively large training set. The total number of available labeled samples in each data set is given by the sum of the second and third columns.

To increase the challenge of the classification problems, following [5], the original features of the data sets "Sonar" through "ARR" and "DNA_l" are artificially augmented

TABLE II

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	Logo	FMS	MBEGA	Elasticnet	kPLS	MetaDist	DEFS	mRMR	SVM
Sonar	**22.3**(3.4,5)	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)	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)	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)	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)	33.9(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	**7.5**(8.3,27)	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)	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	**13.8**	20.0	19.0	20.5	21.3	19.5	29.4	23.0	19.4	43.6

by 100 irrelevant features, independently sampled from a standard normal distribution. Data sets "Prostate" through "Nervous system" are microarray data sets, for which M is large in comparison to N. In each case, 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 paper "Logo" [5] is used for pruning, although other approaches may be used.

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

C. Accuracy of Classification

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

In our experiments, the number of selected features t in our comparison feature selection algorithms and the parameter α of 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.

Following [5], 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 ten times on each data set, where for each run, the respective number of available data points, presented in the second column of Table I, are randomly selected as training samples and the remaining data points, the number of which is indicated in the third column of Table I, are used as test samples for that run. The average performance and the standard deviation over all ten 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, are reported in Table II where, for each data set, the best result over all eight algorithms is shown in bold. The average of the classification error rates over all the ten data sets is shown in the last row in Table II.

In order to demonstrate the effectiveness of 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 II, 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.

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 best four comparison feature selection algorithms (on the basis of the last row of Table II) are shown in Fig. 5. This figure, in addition to the results reported in Table II, show that the classification accuracy of the proposed lLFS algorithm has the lowest error rate over all the data sets considered. In particular, it is to be noted that the performance of the lLFS method exceeds that of other comparison methods specifically designed for the small N large M case, for the data sets shown.

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) [72], [73]. These classifiers are again used in conjunction with the best four comparison feature selection algorithms. The average of the minimum classification errors (using these classifiers) over all ten data sets is presented in Table III. We see that the improved performance of the lLFS method persists in this case also.

D. Relevant Feature Identification

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

The synthetic or "toy" data set shown in Fig. 4 is included for the sole purpose of demonstrating that the proposed method

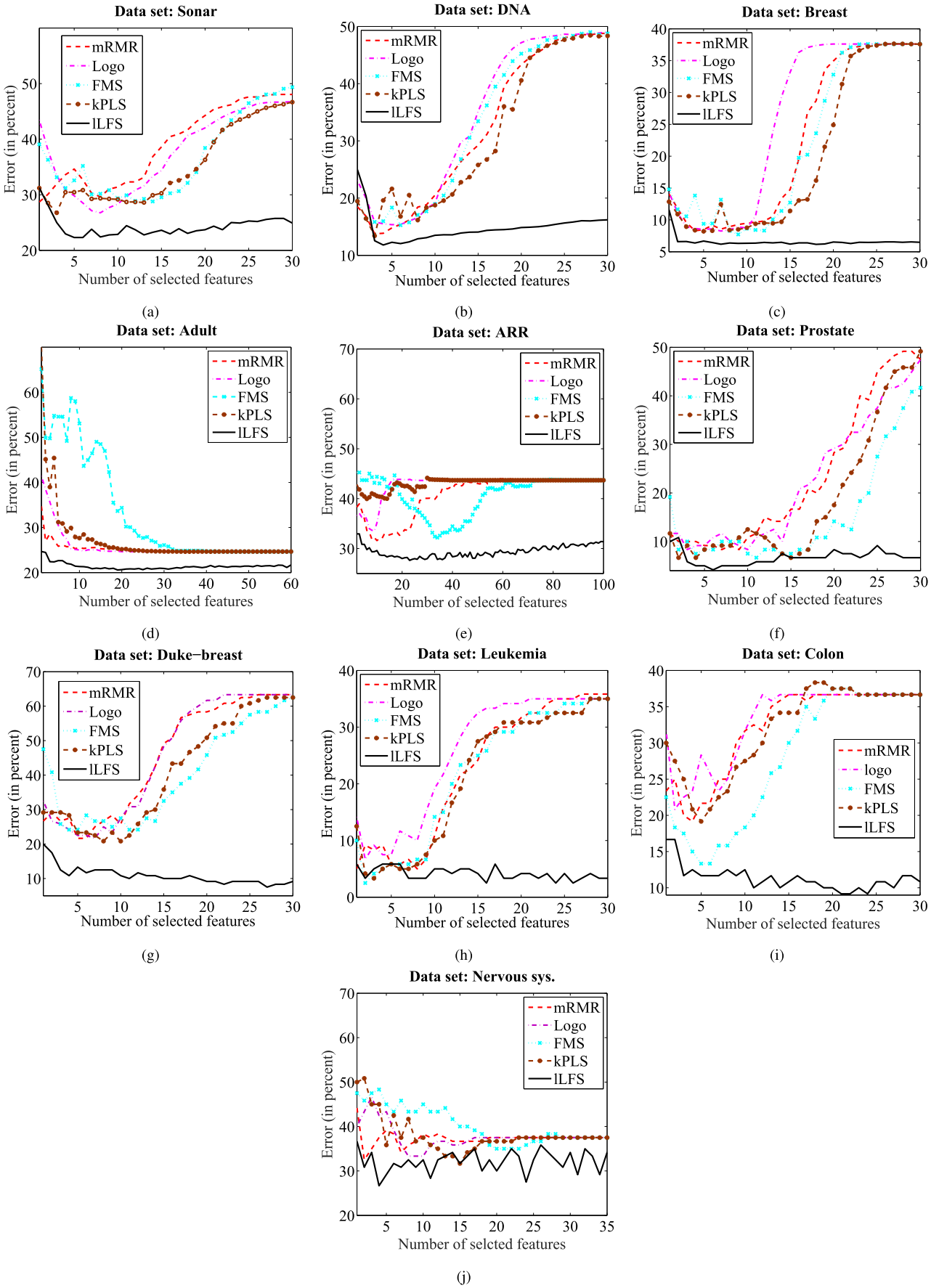

Fig. 5. Classification error (in percent) versus the number of selected features for the proposed lLFS method and the best four comparison feature selection algorithms over all ten real-world data sets.

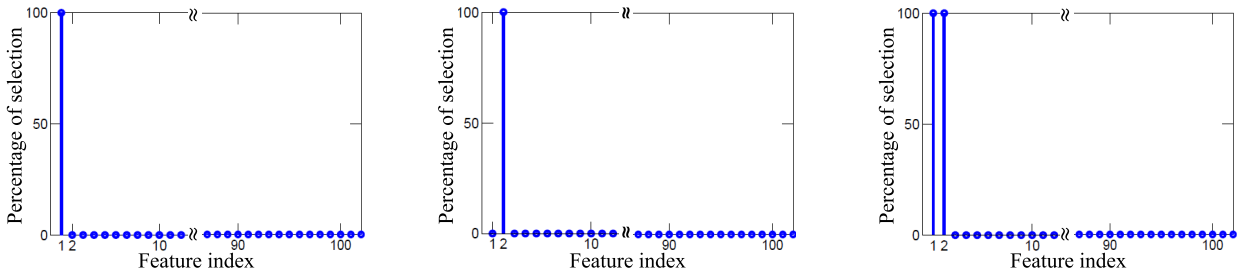

Fig. 6. Selected features for the synthetic data set. The height of each feature index indicates what percentage of the representative points in (a) subclass ■ of class Y_1, (b) subclass ◄ of class Y_1, and (c) class Y_2 shown by ♦ select the respective feature as a member of their optimal feature subset, where α is set to 2.

TABLE III

MINIMUM CLASSIFICATION ERROR (IN PERCENT) OF THE BEST FOUR 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

is capable of identifying discriminative 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 discriminative 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. 6 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 select 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.

The data set "DNA" is generally used for detecting the "presence" or "absence" of a splice junction in a given deoxyribonucleic acid (DNA) sequence [12]. It has been previously shown that improved performance in most cases is observed if the attributes closest to the junctions are used [12], [74]. These attributes correspond to features indexed from 61 to 120. We therefore have a good idea

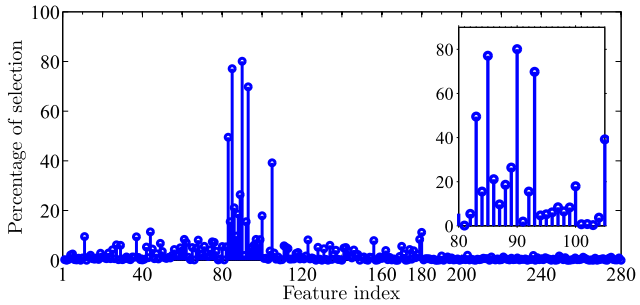

Fig. 7. Selected features for "DNA" data set. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 10.

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. 7, 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 subfeature 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.

E. 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 behavior 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.

1) 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," is shown in Fig. 8. Here, the normalized histogram of within-class samples is shown in red and between-class

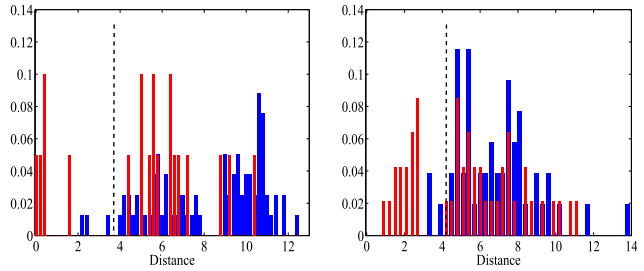

Fig. 8. Distribution of samples around a typical representative point of (a) "Adult" data set and (b) "ARR" data set. In each case, the normalized histogram of within-class distances from the respective representative point is shown in red, and that for between-class distances is in blue. The black dashed line indicates the value of the radius of the respective $\mathcal{Q}^{(i)}$, for the specified level of impurity $\gamma = 0.2$.

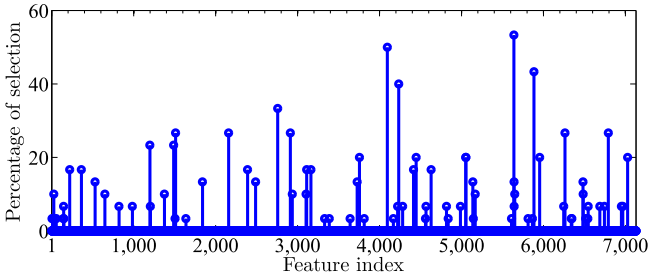

Fig. 9. Histogram of selected features for "Duke-Breast" data set. The height of each feature index indicates what percentage of representative points select the respective feature as a member of their optimal subfeature set. The parameter α is set to the typical value of 10.

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 interclass samples are distributed further from the representative point, as desired. Fig. 8 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 that these samples are heavily contaminated with between-class samples. Therefore, 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 that 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.

2) Overlap of the Optimal Feature Subsets: To what extent do the selected features vary over the representative regions? To address this question, in Fig. 9, we show the normalized

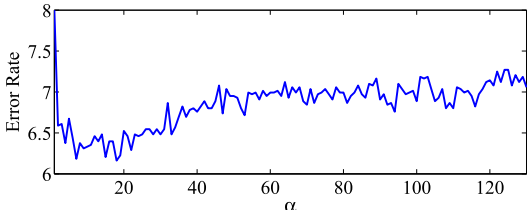

Fig. 10. Classification error rate (in percent) of the proposed method for the data set "Breast" where the parameter α ranges from 1 to the maximum possible value of M, i.e., 130.

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 percentage of the representative points select that respective feature as a member of their selected feature subset. We see that 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 subpopulations 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 subpopulations. 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 ten dominant features, i.e., most informative features, are selected as global features and fed into the SVM classifier with an RBF kernel. The classification error rate using such a subfeature 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 II 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.

F. Sensitivity to the Parameter $\boldsymbol{\alpha}$

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 discriminative features for the respective region, as previously discussed in Section III-C. To demonstrate this point, the classification error rate of the proposed method vs. 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 Figs. 10 and 11, respectively.

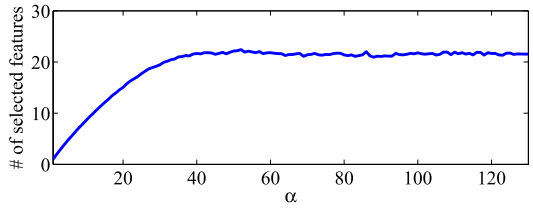

Fig. 11. Averaged number of active features in the optimal feature sets $\mathbf{f}^{*(i)}$ $i = 1, \ldots, N$ versus the parameter α. α ranges from 1 to the maximum possible value of $M = 130$.

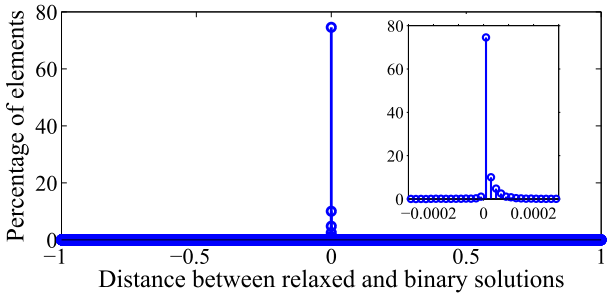

Fig. 12. Normalized histogram of distances of binary elements from the corresponding relaxed elements for data set "Duke-Breast." The parameter α is set to the typical value of 10.

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. 11, the saturation value is 21. This value is the maximum number of features that each local region may require.

G. 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 II-B2, the normalized distribution of the distances of binary elements from the corresponding relaxed elements for data set "Duke-Breast" is shown in Fig. 12. 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.

H. lLFS With a Large Number of Irrelevant Features

A reader may be interested to see performance of the proposed lLFS method in selecting discriminative features in the presence of thousands of irrelevant features. To this end, the performance of the lLFS method on the real-world data set "DNA" (where its "ground truth" is defined in Section IV-D) is shown in Fig. 13 where samples of "DNA" are contaminated with 10^5 *iid* irrelevant features. As is shown, after feature selection, the lLFS algorithm correctly selects 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 100 180. This experiment, as well as the results reported in Table II, confirms the performance of the proposed method for identification of the most discriminative features in the presence of thousands of irrelevant features.

I. CPU Time

The required CPU time for computing optimal feature subsets for lLFS and all our comparison feature selection methods over all data sets are presented in Table IV. Note that these algorithms are implemented in different programming languages—MBEGA is implemented in Java, mRMR, and FMS in C, and the remainder in MATLAB. In comparing computing times in Table IV, the language must be taken into consideration, since MATLAB is significantly slower than Java or C.

The lLFS method is implemented in MATLAB where we use the package "fmincon" for solving the convex and quasi-convex optimization problems defined in (6) and (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 in Table IV. If a computer has K cores, the CPU time of the lLFS method will be $1/K$ of the upper extreme, and 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 single 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 paper has eight cores, the required CPU times are $1/8$ 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.

We do not promote the lLFS as being fast in the training phase. Rather, we submit that its advantage is performance. Regardless, even when lLFS is executed in MATLAB, the required computational times lie between the fastest and slowest of the comparison algorithms. From these data, it is reasonable to say that the lLFS method is feasible with regard to computational time.

Note further that, the feature selection process is performed in the training phase, which is off-line. On the other hand, the more critical online test phase, i.e., classification of query data, is performed much more quickly, once training is complete—the average test phase time over the data sets employed in this paper is 6 ms. This is because the classification process requires no optimization and only involves testing whether the query data are contained within the specified hyperspheres, and determining the class label of its nearest neighbors.

J. Discussion

The CPU time required for computing the optimal feature subset of a single representative point versus the number of the training samples N is shown in Fig. 14, where N is increased up to 10^4. As may be seen, the figure shows approximately linear complexity of CPU time (for one representative point) with respect to the total number of training points. Therefore, the complexity of the proposed lLFS algorithm for computing

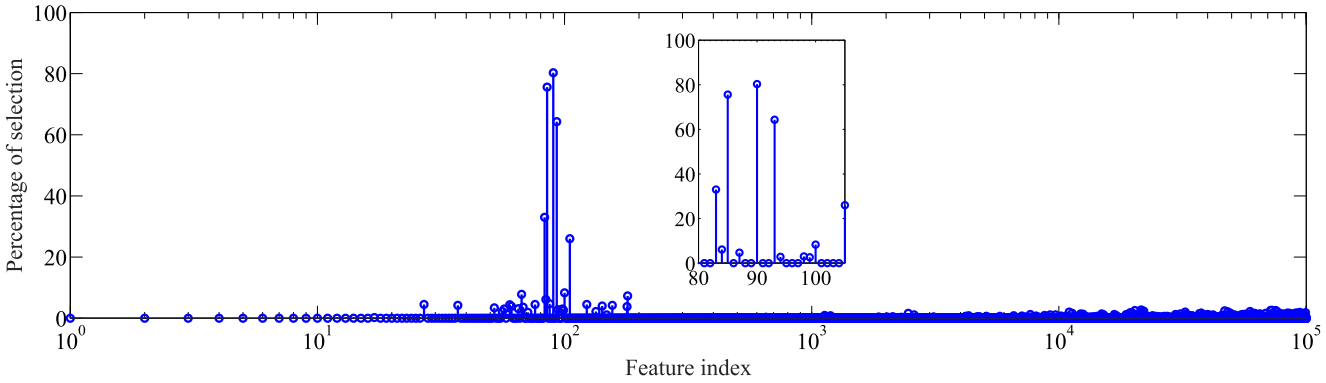

Fig. 13. Selected features for "DNA" data set where each sample is augmented with 10^5 *i.i.d* irrelevant features. The height corresponding to each feature index indicates what percentage of representative points select the respective feature as a discriminative feature, where α is set to a typical value of 10.

TABLE IV

CPU TIME (S) TAKEN FOR FEATURE SELECTION BY DIFFERENT ALGORITHMS WHERE PARAMETER α (t) IS SET TO A TYPICAL VALUE 10. FOR LLFS, THE LEFT-HAND NUMBER INDICATES THE REQUIRED COMPUTING TIME IN MATLAB USING N CORES, (i.e., $K = N$), WHEREAS THE RIGHT-HAND NUMBER IS FOR A SINGLE CORE

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

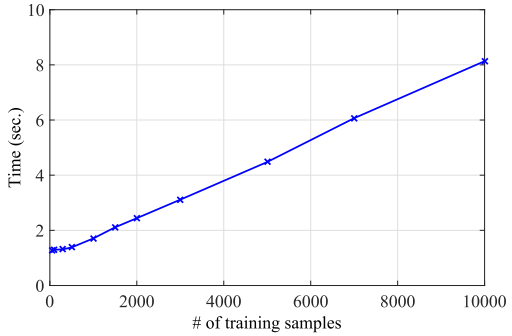

Fig. 14. CPU time taken for computing the optimal feature subset of a representative point versus number of training samples N on a synthetic data set (with a similar distribution as is illustrated in Fig. 4, where all three data clusters have the same number of sample points) where α is set to 2 and the data set is contaminated with 5000 irrelevant features.

all N training points is $\mathcal{O}\{(N^2/K)\}$, where K is the number of available CPU cores (see Section IV-I).

The performance of the proposed lLFS method and the best four comparison algorithms on data set "DNA_l" that has relatively large number of training samples is shown in Table V. In this case, the classification result using lLFS is comparable with the global algorithms. Therefore, it can be said that in classification problems with a large enough training set, the overlap between the selected feature subsets (computed by the lLFS method) would increase. Hence, considering the fact that the global feature selection algorithms usually have lower computational complexity, we recommend that global feature selection algorithms be applied in such applications.

TABLE V

MINIMUM CLASSIFICATION ERROR (IN PERCENT) OF THE DIFFERENT ALGORITHMS APPLIED TO THE DATA SET "DNA_l" WITH RELATIVELY LARGE TRAINING SET WHERE THE NUMBER OF SELECTED FEATURES t (α) RANGES FROM 1 TO 30, SINCE THERE IS NO PERFORMANCE IMPROVEMENT FOR LARGER VALUES. THE LAST COLUMN CORRESPONDS TO THE CLASSIFICATION RESULTS USING SVM (WITH RBF KERNEL) WITHOUT PRIOR FEATURE SELECTION

Data set	lLFS	Logo	FMS	kPLS	mRMR	SVM
DNA_l	**5.8**	41.6	6.8	6.8	6.7	48.3

V. CONCLUSION

In this paper, 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 nonlinear and/or a disjoint manifold. The proposed feature selection is formulated as a joint convex/increasing quasi-convex optimization problem with no local minima. Query data are classified through aggregation of "weak" classifier results, which are based on the selected region-specific feature subsets. The 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 discriminative features, suggest the lLFS method is insensitive to the overfitting problem. In this paper, we specifically considered the challenging case where a small number of observations are available for training. Experimental results demonstrate the superior performance of the proposed algorithm on a large variety of data sets.

APPENDIX

In this section, the VC dimension of the proposed lLFS classifier defined in Section II-C 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 II-C and III-B, the family of functions $\mathcal{F} = \{f(\mathbf{x}^{(i)}; \gamma)\}$ for the lLFS classifier is given, such that the functions $f(\cdot;\cdot)$ are defined as

$$f(\mathbf{x}^{(i)}; \gamma) = \underset{Y_l \in \mathcal{Y}}{\operatorname{argmax}} \{S_{Y_1}(\mathbf{x}^{(i)}; \gamma), S_{Y_2}(\mathbf{x}^{(i)}; \gamma)\} \tag{19}$$

where $S_{Y_l}(\mathbf{x}^{(i)}; \gamma)$, $l = 1, 2$ is defined in (9) to (11). The only parameter which varies in f is the radius of the hyperspheres, controlled through γ.

It is necessary to make assumptions in the derivation of the VC dimension for the lLFS 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 data fall within all hyperspheres. This is not unreasonable since, in the lLFS algorithm, 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 $N \to \infty$, where N is the total number of training points.

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

Proof: Recall the radius of a weak classifier grows until the "impurity level" of the corresponding hypersphere $\mathcal{Q}^{(k)}$ is not greater than the predefined parameter γ, where "impurity level" is the ratio of the number of samples with the 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 misclassification as long as $\lfloor \gamma |\mathcal{L}_l| \rfloor = 0$, i.e., the maximum cardinality of the lth cluster without any classification error is $\lceil (1/\gamma) \rceil - 1$, where $\lceil \cdot \rceil$ denotes the ceiling function. Hence, over L clusters, the lLFS classifier can shatter *at least* $L(\lceil (1/\gamma) \rceil - 1)$ samples.

Now, assume the case where there is an extra training point added, i.e., there are altogether a total of $L(\lceil (1/\gamma) \rceil - 1) + 1$ 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 misclassified, i.e., until the impurity level is not greater than γ. This sample will be misclassified by all η_1 weak classifiers (see the assumptions). Therefore, for this sample, the first term of the argmax function in (19) 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 misclassified.

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

REFERENCES

[1] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, 1962.

[2] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMS," in *Proc. NIPS*, vol. 12. 2000, pp. 668–674.

[3] L. J. van't Veer, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[4] Y. Wang *et al.*, "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer," *Lancet*, vol. 365, no. 9460, pp. 671–679, 2005.

[5] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1610–1626, Sep. 2010.

[6] A. R. Webb, *Statistical Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2003.

[7] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2005.

[8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[9] F. Oveisi, S. Oveisi, A. Erfanian, and I. Patras, "Tree-structured feature extraction using mutual information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 127–137, Jan. 2012.

[10] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[11] H.-L. Wei and S. A. Billings, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 162–166, Jan. 2007.

[12] L. Wang, "Feature selection with kernel class separability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1534–1546, Sep. 2008.

[13] H. Zeng and Y.-M. Cheung, "Feature selection and kernel learning for local learning-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011.

[14] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 143–159, Jan. 2002.

[15] R. Chakraborty and N. R. Pal, "Feature selection using a neural framework with controlled redundancy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 35–50, Jan. 2015.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Hoboken, NJ, USA: Wiley, 2001.

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

[18] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan, "Feature selection based on structured sparsity: A comprehensive study," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2016.2551724.

[19] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[20] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection–a comparative study," in *Intelligent Data Engineering and Automated Learning-IDEAL*. Berlin, Germany: Springer, 2007, pp. 178–187.

[21] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, May 2013.

[22] J. Wang, P. Zhao, S. C. H. Hoi, and R. Jin, "Online feature selection and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 698–710, Mar. 2014.

[23] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *Proc. IEEE Int. Conf. Data Mining*, Mar. 2014, pp. 660–669.

[24] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[25] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. London, U.K.: Chapman & Hall, 2007.

[26] G. Brown, A. Pocock, M.-J. Zhao, and M. Lujan, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, pp. 27–66, Jun. 2012.

[27] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Feature subset selection using differential evolution and a statistical repair mechanism," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11515–11526, 2011.

[28] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[29] Z. Zhu, Y.-S. Ong, and M. Dash, "Markov blanket-embedded genetic algorithm for gene selection," *Pattern Recognit.*, vol. 40, no. 11, pp. 3236–3248, Nov. 2007.

[30] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 171–234, Jan. 2010.

[31] Q. Cheng, H. Zhou, and J. Cheng, "The Fisher-Markov selector: Fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1217–1233, Jun. 2011.

[32] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi, "Effective discriminative feature selection with nontrivial solution," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[33] M. Ramona, G. Richard, and B. David, "Multiclass feature selection with kernel Gram-matrix-based criteria," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1611–1623, Oct. 2012.

[34] S. Xiang, X. Shen, and J. Ye. (2012). "Efficient sparse group feature selection via nonconvex optimization." [Online]. Available: https://arxiv.org/abs/1205.5075

[35] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc., B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[36] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[37] S. Sun, Q. Peng, and A. Shakoor, "A kernel-based multivariate feature selection method for microarray data classification," *PloS ONE*, vol. 9, no. 7, p. e102541, 2014.

[38] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[39] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.

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

[41] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection-theory and algorithms," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 43.

[42] Y. Sun, "Iterative RELIEF for Feature Weighting: Algorithms, Theories, and Applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1035–1051, Jun. 2007.

[43] B. Chen, H. Liu, J. Chai, and Z. Bao, "Large margin feature weighting method via linear programming," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1475–1488, Oct. 2009.

[44] B. Liu, B. Fang, X. Liu, J. Chen, and Z. Huang, "Large Margin subspace learning for feature selection," *Pattern Recognit.*, vol. 46, no. 10, pp. 2798–2806, 2013.

[45] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 1, no. 1, pp. 24–45, Jan./Mar. 2004.

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

[47] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 269–274.

[48] Y. Li, M. Dong, and J. Hua, "Localized feature selection for clustering," *Pattern Recognit. Lett.*, vol. 29, no. 1, pp. 10–18, 2008.

[49] 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.

[50] N. Armanfard and J. P. Reilly, "Classification based on local feature selection via linear programming," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2013, pp. 1–6.

[51] N. Armanfard, J. Reilly, and M. Komeili, "Local feature selection for data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1217–1227, 2016.

[52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[53] M. T. Thai, "Approximation algorithms: Lp relaxation, rounding, and randomized rounding techniques," in *Lecture Notes*, Gainesville, FL, USA: Univ. Florida, 2013.

[54] A. Souza, "Randomized algorithm & probabilistic methods," in *Lecture Notes*, Humboldt, Ed. Berlin, Germany: Univ. Berlin, 2001.

[55] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox for Use With MATLAB: User's Guide, Version 2*. Natick, MA, USA: Math Works, 1999.

[56] V. N. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

[57] C. J. Burges, "A tutorial on support vector machines for pattern recogbition," *Data Mining Knowl. Discovery*, vol. 3, no. 2, pp. 121–167, 1998.

[58] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[59] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[60] C. Sima and E. R. Dougherty, "What should be expected from feature selection in small-sample settings," *Bioinformatics*, vol. 22, no. 19, pp. 2430–2436, 2006.

[61] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[62] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci.*, vol. 282, pp. 111–135, Mar. 2014.

[63] K. Sjôstrand, "Matlab implementation of LASSO, LARS, the elastic net and SPCA," *Richard Petersens Plads, Building*, vol. 321, pp. 1–2, Jun. 2005. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3897

[64] Y. Li, J. Si, G. Zhou, S. Huang, and S. Chen, "FREL: A stable feature selection algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1388–1402, Jul. 2015.

[65] P. Lovato, M. Bicego, M. Kesa, N. Jojic, V. Murino, and A. Perina, "Traveling on discrete embeddings of gene expression," *Artif. Intell. Med.*, vol. 70, pp. 1–11, Oct. 2016.

[66] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, B (Cybernetics)*, vol. 37, no. 1, pp. 70–76, Mar. 2007.

[67] K. Bache and M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[68] U. Alon *et al.*, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 96, no. 12, pp. 6745–6750, 1999.

[69] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint ℓ_2, 1-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[70] M. West *et al.*, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 20, pp. 11462–11467, 2001.

[71] S. L. Pomeroy *et al.*, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, Jan. 2002.

[72] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, vol. 37. Boca Raton, FL, USA: CRC Press, 1989.

[73] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[74] 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