Sparse ℓ_1 and ℓ_2 Center Classifiers

Giuseppe Calafiore, *Fellow, IEEE,* and Giulia Fracastoro, *Member, IEEE*

***Abstract*—In this paper we discuss two novel sparse versions of the classical nearest-centroid classifier. The proposed sparse classifiers are based on ℓ_1 and ℓ_2 distance criteria, respectively, and perform simultaneous feature selection and classification, by detecting the features that are most relevant for the classification purpose. We formally prove that the training of the proposed sparse models, with both distance criteria, can be performed exactly (i.e., the *globally optimal* set of features is selected) at a linear computational cost. Specifically, the proposed sparse classifiers are trained in $O(mn) + O(m \log k)$ operations, where n is the number of samples, m is the total number of features and $k \leq m$ is the number of features to be retained in the classifier. Further, the complexity of testing and classifying a new sample is simply $O(k)$ for both methods.**

The proposed models can be employed either as stand-alone sparse classifiers, or as fast feature-selection techniques for pre-filtering the features to be later fed to other types of classifiers (e.g., SVMs). The experimental results show that the proposed methods are competitive in accuracy with state-of-the-art feature selection and classification techniques, while having a substantially lower computational cost.

***Index Terms*—Nearest-centroid classifiers, Machine learning, Sparse optimization, Feature selection, Text classification.**

I. Introduction

A. Perspective and literature overview

In recent years the technological progress has led to a massive proliferation of large-scale datasets. The processing of these large amounts of data poses many new challenges, and there is a strong need of algorithms that scale mildly (e.g., linearly or quasi-linearly) with the dataset size. For this reason, classification methods with a very low computational cost, such as Naive Bayes [1], [2], linear Support Vector Machines (SVM) [3], and the nearest-centroid classifier [4], [5], are still an appealing choice in this endeavour. In many cases, these methods are the only feasible approaches, since more sophisticated techniques would be too demanding from the computational point of view. The cited simple classifiers have in common a training complexity that scales as $O(mn)$, where m is the number of features and n is the number of examples in the training data set. This type of complexity is usually referred to as *linear*, since the number of operations scales proportionally to the overall dimension mn of the data set. To put things in perspective, for instance, nonlinear (kernel) SVMs have a much higher training complexity of $O(\max(n, m) \min(n, m)^2)$, see, e.g., [6], [7].

Even for efficient classifiers, however, a challenge to face when dealing with large-scale datasets is the so-called *curse of dimensionality*. In fact, in numerous applications datasets may have a very high number of features, which can be orders of magnitude higher than the number of samples. In this situation, traditional classification techniques may perform poorly, or even break down. For this reason, feature or variable selection methods and sparse classifiers have been deeply studied in the literature, see, e.g., [8], [9]. Variable selection refers to the problem of selecting input variables (features) that are most predictive of a given outcome. Such selection can be accomplished by imposing a *sparsity constraint* on the classifier, so that only $k \ll m$ features effectively come into play in the discrimination function. Appropriate variable selection can enhance the effectiveness and domain interpretability of an inference model, [10], [11]. Indeed, besides reducing the dataset size, feature selection has other important advantages. First, it eliminates noisy, redundant or irrelevant features, thus reducing the risk of overfitting and improving the generalization performance of the classifier. Second, a sparse classifier can lead to a simplified decision rule for faster prediction in large-scale problems. Thirdly, a small set of features typically leads to better interpretability of the results.

State-of-the-art feature selection methods are typically based on heuristics that do not provide any guarantee of global optimality. Some of them, such as the Lasso [12] or the ℓ_1-regularized logistic regression [13], are based on solving a convex optimization problem with a ℓ_1-norm penalty on the regression coefficients to promote sparsity. The main drawback of these techniques is that they are computationally expensive (i.e., they scale way above linearly in the problem dimension). Other methods, such as Odds Ratio [14] or Information Gain [15], propose a different approach that employs a feature ranking based on their inherent characteristics. These methods are usually very fast, but often their performance in terms of accuracy is poor. Moreover, they only perform variable selection, whereas the actual classification is deferred to a second-stage algorithm (for instance, a kernel SVM) that elaborates on the selected features.

Another approach used for mitigating the issue of high dimensionality consists in directly defining sparse classifiers, which should be able to deal with a large number of features by performing *simultaneous* variable selection and classification. Several works introduced a sparse decision rule for the SVM [10], [16], [11]. The main shortcoming of such Sparse SVM classifiers is that their training involves solving a minimization problem, which typically requires a high computational cost. For example, when a ℓ_1-norm penalty term is considered in a linear SVM in order to obtain a sparse classifier, the worst-case computational complexity grows to $O(nm \min(n, m)^2)$, see [16]. A further type of sparse classifier is the sparse nearest mean classifier [17], [18], where a weighting factor for each feature is introduced. However, the feature weights are obtained by solving a linear program, which again requires a computational effort that grows way more than linearly in the problem size. Recently, [19] also presented a sparse version of

The authors are with Politecnico di Torino – Department of Electronics and Telecommunications, Italy. Email: {name.surname}@polito.it.

the Naive Bayes classifier. This method provides a sub-optimal solution in quasi-linear time for multinomial features, and it is shown in [19] that it may surpasses many state-of-the-art feature selection methods in terms of accuracy and speed.

Another popular classifier is the k-nearest neighbors classifier [20] and its variants [21], [22], [23]. However, the k-nearest neighbors classifier is not designed for obtaining sparsity or feature selection; it has been shown to be particularly sensitive to the curse of dimensionality and its performance might significantly deteriorate as the number of feature increases [24]. To overcome this issue, several approximate versions of the k-nearest neighbor classifier have been proposed, see for instance [25], [26], [27].

B. Paper contribution

The context of the contribution of the present work is that of efficient methods for joint variable selection and classification. We propose two types of sparse nearest-center classifiers that guarantee both global optimality and numerical efficiency. The proposed discriminative models efficiently perform simultaneous feature selection and classification and, different from Naive Bayes, they are not directly related to a specific generative statistical model. Specifically, we discuss two models with different geometry of the underlying discrimination function. The first model is based on an ℓ_2 metric for computing distances between feature vectors, and it is named the *ℓ_2-sparse center classifier*. The second model is based on an ℓ_1 metric for computing distances between feature vectors, and it is named the *ℓ_1-sparse center classifier*.

The ℓ_2 model is a natural sparse variant of the nearest-centroid classifier [4], [28], which is a widely used classifier, especially in text classification [29], [30]. Instead, the ℓ_1 model is related to the median classifier [31], [32], which has shown to be more robust to outliers than the ℓ_2 version. We prove that both the proposed methods select the optimal subset of features for the corresponding classifier, in linear time. Other works have already proposed feature selection methods targeted for the nearest-center classifier [33], [28], [34]. However, they focus only on the ℓ_2 case, and most of them cannot provide any optimality guarantee or, when they have a theoretical guarantee of optimality, they are computationally expensive. The experimental results that we report in Section VI show that the techniques we propose achieve similar performance as state-of-the-art feature selection methods, but at a substantially lower computational cost.

The remainder of this paper is organized as follows: Section II presents some preliminary notions on center-based classifiers. In Section III we introduce the proposed sparse center classifiers. Section IV describes an efficient and exact method for training the sparse ℓ_2-center classifier, while Section V presents an analogous result for the sparse ℓ_1-center classifier. In Section VI we report numerical experiments comparing the proposed methods with relevant methods existing in the literature. Conclusions are finally drawn in Section VII. One technical result is reported in the appendix Section VIII-A for better readability of the main text.

II. Preliminaries on center-based classifiers

Let

$$X = \left[x^{(1)} \cdots x^{(n)}\right] \in \mathbb{R}^{m,n}, \tag{1}$$

be a given data matrix whose columns $x^{(j)} \in \mathbb{R}^m$, $j = 1, \ldots, n$, contain feature vectors from n observations, and let $\mathbf{y} \in \mathbb{R}^n$ be a given response vector such that $y_j \in \{-1, +1\}$ is the class label corresponding to the j-th observation. We consider a binary classification problem, in which a new observation vector $x \in \mathbb{R}^m$ is to be assigned to the positive class C_+ (corresponding to $y = +1$) or to the negative class C_- (corresponding to $y = -1$). To this purpose, the *nearest centroid classifier* [29], [4], [28] is a well-known classification model, which works by assigning the class label based on the least Euclidean distance from x to the centroids of the classes. The centroids are computed on the basis of the training data as

$$\bar{x}^+ = \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} x^{(j)}, \quad \bar{x}^- = \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} x^{(j)}, \tag{2}$$

where $\mathcal{J}^+ \doteq \{j \in \{1, \ldots, n\} : y_j = +1\}$ contains the indices of the observations in the positive class, $\mathcal{J}^- \doteq \{j \in \{1, \ldots, n\} : y_j = -1\}$ contains the indices of the observations in the negative class, and n_+, n_- are the cardinalities of $\mathcal{J}^+$ and $\mathcal{J}^-$, respectively. A new observation vector x is classified as positive or negative according to the sign of the discrimination function

$$\Delta_2(x) = \|x - \bar{x}^-\|_2^2 - \|x - \bar{x}^+\|_2^2,$$

that is, x is classified in the positive class if its Euclidean distance from the positive centroid is smaller that its distance from the negative centroid, and viceversa for the negative class. The discrimination function for the centroid classifier is linear with respect to x, since

$$\begin{aligned} \Delta_2(x) &= \|x\|_2^2 + \|\bar{x}^-\|_2^2 - 2x^\top \bar{x}^- - \|x\|_2^2 - \|\bar{x}^+\|_2^2 + 2x^\top \bar{x}^+ \\ &= \quad (\|\bar{x}^-\|_2^2 - \|\bar{x}^+\|_2^2) + 2x^\top(\bar{x}^+ - \bar{x}^-), \end{aligned} \tag{3}$$

where the coefficient in the linear term of the classifier is given by vector $w \doteq \bar{x}^+ - \bar{x}^-$. Notice that, whenever $\bar{x}_i^+ = \bar{x}_i^-$ for some component i (i.e., $w_i = 0$), the corresponding feature x_i in x is irrelevant for the purpose of classification.

Remark 1: We observe that the centroids in (2) can be interpreted as the optimal solutions to the following optimization problem:

$$\min_{\theta^+, \theta^- \in \mathbb{R}^m} \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)} - \theta^+\|_2^2 + \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)} - \theta^-\|_2^2. \tag{4}$$

That is, the centroids are the points that minimize the average squared distance to the samples within each class. A proof of this fact is immediate, by taking the gradient of the objective in (4) with respect to θ^+ and equating it to zero, and then doing the same thing for θ^-. The two problems are actually decoupled, so the two coefficients $1/n_+$ and $1/n_-$ play no role here in terms of the optimal solution. However, they have been introduced here for balancing the contribution of the residuals of the two classes. ⋆

We shall call (4) the (plain) ℓ_2-center classifier training problem, and Δ_2 in (3) the corresponding discrimination function. The usual centroids in (2) are thus the points that minimize the average ℓ_2 distance from the respective class representatives. This interpretation opens the way to considering different types of metrics for computing centers. In particular, there exist an extensive literature on the favorable properties of the ℓ_1 norm distance, which is well known to provide center estimates that are robust to outliers. The natural ℓ_1 version of problem (4) is

$$\min_{\theta^+,\theta^-\in\mathbb{R}^m} \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}\|x^{(j)}-\theta^+\|_1+\frac{1}{n_-}\sum_{j\in\mathcal{J}^-}\|x^{(j)}-\theta^-\|_1, \tag{5}$$

which we shall call the (plain) ℓ_1-center classifier training problem. It is known that an optimal solution to problem (5) is obtained, for each $i=1,\ldots,m$, by taking θ_i^+ to be the *median* of the values $x_i^{(j)}$ in the positive class, and θ_i^- to be the median of the values $x_i^{(j)}$ in the negative class (see Proposition 2 for a generalization of this fact). We let

$$\mu^+ \doteq \mathrm{med}(\{x^{(j)}\}_{j\in\mathcal{J}^+}), \quad \mu^- \doteq \mathrm{med}(\{x^{(j)}\}_{j\in\mathcal{J}^-}), \tag{6}$$

where $\mathrm{med}(\cdot)$ computes the median of its input vector sequence along each component, i.e., for each $i=1,\ldots,m$, μ_i^+ is the median of $\{x_i^{(j)}\}_{j\in\mathcal{J}^+}$, and μ_i^- is the median of $\{x_i^{(j)}\}_{j\in\mathcal{J}^-}$. The classification in the ℓ_1-center classifier is made by computing the distances from the new datum x and the ℓ_1 centers of the classes, and assigning x to the closest center, that is, we compute

$$\Delta_1(x) \doteq \|x-\mu^-\|_1 - \|x-\mu^+\|_1,$$

and assign x to the positive or negative class depending on the sign of $\Delta_1(x)$. We observe that, contrary to the ℓ_2 case, the discrimination function $\Delta_1(x)$ is not linear in x. However, expressed more explicitly in its components, $\Delta_1(x)$ is written as

$$\Delta_1(x)=\sum_{i=1}^m\left(|x_i-\mu_i^-|-|x_i-\mu_i^+|\right),$$

and we observe again, like in the ℓ_2 case, that the contribution to $\Delta_1(x)$ from the ith feature x_i is identically zero when $\mu_i^-=\mu_i^+$.

III. Sparse ℓ_1 and ℓ_2 center classifiers

In Section II we observed that, for both the ℓ_2 and the ℓ_1 distance criteria, the discrimination is insensitive to the ith feature whenever $\theta_i^+-\theta_i^-=0$, where θ^+, θ^- are the two class centers. The *sparse* classifiers that we introduce in this section are aimed precisely at computing optimal class centers such that the center difference $(\theta^+-\theta^-)$ is k-sparse, meaning that $\|\theta^+-\theta^-\|_0\leq k$, where $\|\cdot\|_0$ denotes the number of nonzero entries (i.e., the cardinality) of its argument, and $k\leq m$ is a given cardinality bound. Such type of sparse classifiers will thus perform simultaneous classification and feature selection, by detecting which k out of the total m features are relevant for the classification purposes. We next formally define the sparse ℓ_2 and ℓ_1 center classifier training problems.

Definition 1 (Sparse ℓ_2-center classifier): A sparse ℓ_2-center classifier is a model which classifies an input feature vector $x\in\mathbb{R}^m$ into a positive or a negative class, according to the sign of the discrimination function

$$\begin{aligned}\Delta_2(x) &= \|x-\theta^-\|_2^2-\|x-\theta^+\|_2^2\\ &= (\|\theta^-\|_2^2-\|\theta^+\|_2^2)+2x^\top(\theta^+-\theta^-),\end{aligned} \tag{7}$$

where the sparse ℓ_2-centers θ^+, θ^- are learned from a data batch (1) as the optimal solutions of the problem

$$\begin{aligned}\min_{\theta^+,\theta^-\in\mathbb{R}^m} & \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}\|x^{(j)}-\theta^+\|_2^2+\frac{1}{n_-}\sum_{j\in\mathcal{J}^-}\|x^{(j)}-\theta^-\|_2^2\\ \text{s. t.: } & \|\theta^+-\theta^-\|_0\leq k,\end{aligned} \tag{8}$$

where $k\leq m$ is a given upper bound on the cardinality of $\theta^+-\theta^-$.

Definition 2 (Sparse ℓ_1-center classifier): A sparse ℓ_1-center classifier is a model which classifies an input feature vector $x\in\mathbb{R}^m$ into a positive or a negative class, according to the sign of the discrimination function

$$\Delta_1(x)\doteq\|x-\theta^-\|_1-\|x-\theta^+\|_1, \tag{9}$$

where the sparse ℓ_1-centers θ^+, θ^- are learned from a data batch (1) as the optimal solutions of the problem

$$\begin{aligned}\min_{\theta^+,\theta^-\in\mathbb{R}^m} & \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}\|x^{(j)}-\theta^+\|_1+\frac{1}{n_-}\sum_{j\in\mathcal{J}^-}\|x^{(j)}-\theta^-\|_1\\ \text{s. t.: } & \|\theta^+-\theta^-\|_0\leq k,\end{aligned} \tag{10}$$

where $k\leq m$ is a given upper bound on the cardinality of $\theta^+-\theta^-$.

A perhaps notable fact is that both the sparse ℓ_2 and the sparse ℓ_1 classifier training problems can be solved exactly and with almost-linear-time complexity (this fact is proved in the next sections), which also makes them good candidates for efficient feature selection methods in two-phase (feature selection + actual classifier training) classifier training procedures.

Remark 2 (Extension to multi-class classification): The focus of this paper is on binary classification. The multi-class case is not treated here directly. However, we mention that the proposed methods may be adapted to a multi-class context by reducing the multi-class classification task to multiple binary classification tasks. Several well-known strategies can be used for this purpose, such as sequences of one-vs-all comparisons [35] or decision trees [36], [37]. $\star$

IV. Training the sparse ℓ_2-center classifier

We next discuss how to solve the training problem in (8). Let us denote by J the objective to be minimized in (8). By expanding the squares and using (2), we have

$$\begin{aligned}J &= \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}\|x^{(j)}\|_2^2+\frac{1}{n_-}\sum_{j\in\mathcal{J}^-}\|x^{(j)}\|_2^2+\|\theta^+\|_2^2\\ &+ \|\theta^-\|_2^2-2\bar{x}^{+\top}\theta^+-2\bar{x}^{-\top}\theta^-\\ &= \text{cost.}+\|\theta^+\|_2^2+\|\theta^-\|_2^2-2\bar{x}^{+\top}\theta^+-2\bar{x}^{-\top}\theta^-.\end{aligned}$$

Let now $\mathcal{E}$ denote a fixed set of indices of cardinality $m-k$, and $\mathcal{D}$ denote the complementary set, that is, $\mathcal{D} = \{1,\ldots,m\} \setminus \mathcal{E}$. For any vector $x \in \mathbb{R}^m$ we next use the notation $x_{\mathcal{D}}$ to denote a vector of the same dimension as x which coincides with x at the locations in $\mathcal{D}$ and it is zero elsewhere. We define analogously $x_{\mathcal{E}}$, so that $x = x_{\mathcal{D}} + x_{\mathcal{E}}$. We then let

$$\begin{aligned} \theta^+ &= \theta^+_{\mathcal{D}} + \theta^+_{\mathcal{E}} \\ \theta^- &= \theta^-_{\mathcal{D}} + \theta^-_{\mathcal{E}}. \end{aligned}$$

Suppose that we fixed the set $\mathcal{E}$ of the indices where $\theta^+ - \theta^-$ is zero (we shall discuss later how to eventually optimize over this choice of the index set), so that $\theta^+_{\mathcal{E}} - \theta^-_{\mathcal{E}} = 0$. We can therefore set

$$\theta^+_{\mathcal{E}} = \theta^-_{\mathcal{E}} \doteq \theta_{\mathcal{E}},$$

whence

$$\begin{aligned} \theta^+ &= \theta^+_{\mathcal{D}} + \theta_{\mathcal{E}} \\ \theta^- &= \theta^-_{\mathcal{D}} + \theta_{\mathcal{E}}. \end{aligned}$$

With such given choice of the zero index set, and using the above expressions for θ^+, θ^-, the problem objective becomes

$$\begin{aligned} J_{\mathcal{E}} &= \text{cost.} + \|\theta^+\|_2^2 + \|\theta^-\|_2^2 - 2\bar{x}^{+\top}\theta^+ - 2\bar{x}^{-\top}\theta^- \\ &= \text{cost.} + 2\|\theta_{\mathcal{E}}\|_2^2 - 4\tilde{x}^\top\theta_{\mathcal{E}} + \|\theta^+_{\mathcal{D}}\|_2^2 + \|\theta^-_{\mathcal{D}}\|_2^2 \\ &- 2\bar{x}^{+\top}\theta^+_{\mathcal{D}} - 2\bar{x}^{-\top}\theta^-_{\mathcal{D}}, \end{aligned}$$

where we defined

$$\tilde{x} \doteq \frac{\bar{x}^+ + \bar{x}^-}{2}. \tag{11}$$

For given zero index set $\mathcal{E}$ we can therefore minimize $J_{\mathcal{E}}$ with respect to $\theta_{\mathcal{E}}$, $\theta^+_{\mathcal{D}}$, and $\theta^-_{\mathcal{D}}$. By simply equating the respective gradients to zero, we obtain that the optimal parameter values are

$$\theta^*_{\mathcal{E}} = \tilde{x}_{\mathcal{E}}, \quad \theta^{+*}_{\mathcal{D}} = \bar{x}^+_{\mathcal{D}}, \quad \theta^{-*}_{\mathcal{D}} = \bar{x}^-_{\mathcal{D}}.$$

Substituting these optimal values back into $J_{\mathcal{E}}$ we obtain

$$\begin{aligned} J^*_{\mathcal{E}} &= \text{cost.} - 2\|\tilde{x}_{\mathcal{E}}\|_2^2 - \|\bar{x}^+_{\mathcal{D}}\|_2^2 - \|\bar{x}^-_{\mathcal{D}}\|_2^2 \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+_{\mathcal{E}} + \bar{x}^-_{\mathcal{E}}\|_2^2 - \|\bar{x}^+_{\mathcal{D}}\|_2^2 - \|\bar{x}^-_{\mathcal{D}}\|_2^2 \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+_{\mathcal{E}}\|_2^2 - \frac{1}{2}\|\bar{x}^-_{\mathcal{E}}\|_2^2 - \bar{x}^{+\top}_{\mathcal{E}}\bar{x}^-_{\mathcal{E}} - \|\bar{x}^+_{\mathcal{D}}\|_2^2 - \|\bar{x}^-_{\mathcal{D}}\|_2^2 \\ &= \text{cost.} - \frac{1}{2}(\|\bar{x}^+_{\mathcal{E}}\|_2^2 + \|\bar{x}^+_{\mathcal{D}}\|_2^2) - \frac{1}{2}(\|\bar{x}^-_{\mathcal{E}}\|_2^2 + \|\bar{x}^-_{\mathcal{D}}\|_2^2) \\ &- \bar{x}^{+\top}_{\mathcal{E}}\bar{x}^-_{\mathcal{E}} - \frac{1}{2}\left(\|\bar{x}^+_{\mathcal{D}}\|_2^2 + \|\bar{x}^-_{\mathcal{D}}\|_2^2\right) \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+\|_2^2 - \frac{1}{2}\|\bar{x}^-\|_2^2 - \bar{x}^{+\top}_{\mathcal{E}}\bar{x}^-_{\mathcal{E}} \\ &- \frac{1}{2}\left(\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2 + 2\bar{x}^{+\top}_{\mathcal{D}}\bar{x}^-_{\mathcal{D}}\right) \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+\|_2^2 - \frac{1}{2}\|\bar{x}^-\|_2^2 - (\bar{x}^{+\top}_{\mathcal{E}}\bar{x}^-_{\mathcal{E}} + \bar{x}^{+\top}_{\mathcal{D}}\bar{x}^-_{\mathcal{D}}) \\ &- \frac{1}{2}\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2 \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+\|_2^2 - \frac{1}{2}\|\bar{x}^-\|_2^2 - \bar{x}^{+\top}\bar{x}^- - \frac{1}{2}\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2 \\ &= \text{cost.} - \frac{1}{2}\|\bar{x}^+ + \bar{x}^-\|_2^2 - \frac{1}{2}\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2. \end{aligned}$$

This last expression shows that $J^*_{\mathcal{E}}$ depends on the choice of the zero index set $\mathcal{E}$ only via the term $\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2$ involving the complementary set $\mathcal{D}$. Minimizing $J^*_{\mathcal{E}}$ with respect to the index set $\mathcal{E}$ thus amounts to maximizing $\|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2$ with respect to the complementary index set $\mathcal{D}$, that is

$$J^* = \text{cost.}' - \frac{1}{2}\max_{|\mathcal{D}|\leq k} \|\bar{x}^+_{\mathcal{D}} - \bar{x}^-_{\mathcal{D}}\|_2^2.$$

The solution to this problem is immediate: we construct the difference vector $\delta \doteq \bar{x}^+ - \bar{x}^-$ and let $\mathcal{D}^*$ contain the indices of the k largest elements of $|\delta|$. We have therefore proved the following

Proposition 1: The optimal solution of problem (8) is obtained as follows:

1) Compute the standard class centroids $\bar{x}^+$, $\bar{x}^-$ according to (2);
2) Compute the centroids midpoint $\tilde{x}$ according to (11), and the centroids difference $\delta \doteq \bar{x}^+ - \bar{x}^-$;
3) Let $\mathcal{D}$ be the set of the indices of the k largest absolute value elements in vector δ, and let $\mathcal{E}$ be the complementary index set;
4) The optimal parameters θ^+, θ^- are given by

$$\begin{aligned} \theta^+ &= \bar{x}^+_{\mathcal{D}} + \tilde{x}_{\mathcal{E}} \\ \theta^- &= \bar{x}^-_{\mathcal{D}} + \tilde{x}_{\mathcal{E}}. \end{aligned}$$

This procedure is summarized in Algorithm 1.

Algorithm 1 Training the sparse ℓ_2-center classifier

```
1: Input: $x^{(j)} \in \mathbb{R}^m$: training data, $j = 1,...,n$; $\mathbf{y} \in \mathbb{R}^n$: class labels.
2: Compute the standard class centroids:
   $\bar{x}^+ = \frac{1}{n_+}\sum_{j\in\mathcal{J}^+} x^{(j)}, \quad \bar{x}^- = \frac{1}{n_-}\sum_{j\in\mathcal{J}^-} x^{(j)}$;
3: Compute the centroids midpoint and the centroids difference: $\tilde{x} = \frac{\bar{x}^+ + \bar{x}^-}{2}, \quad \delta = \bar{x}^+ - \bar{x}^-$;
4: Define the set $\mathcal{D}$ of the indices corresponding to the $k$ largest absolute value elements in $\delta$, and the complementary index set $\mathcal{E}$;
5: Return the optimal sparse centroids:
   $\theta^+ = \bar{x}^+_{\mathcal{D}} + \tilde{x}_{\mathcal{E}}, \quad \theta^- = \bar{x}^-_{\mathcal{D}} + \tilde{x}_{\mathcal{E}}$.
```

Remark 3 (Numerical complexity for training the sparse ℓ_2 classifier): Steps 1-2 in Proposition 1 essentially require computing mn sums. Finding the k largest elements in Step 3 takes $O(m\log k)$ operations (using, e.g., min-heap sorting), whence the whole procedure takes $O(mn) + O(m\log k)$ operations. Thus, while training a plain centroid classifier takes $O(mn)$ operations (which, incidentally, is also the complexity figure for training a classical Naive Bayes classifier or a linear SVM), adding exact sparsity comes at the quite moderate extra cost of $O(m\log k)$ operations. ⋆

Remark 4 (Comparison with SVM training complexity): As discussed in Section I, one of the most popular classification methods is the SVM. Training a kernel SVM requires solving a quadratic problem: modern SVM solvers use various decomposition techniques that essentially require $O(mn^2)$ operations (assuming $m > n$), see, e.g., [6]. When both m and n are large, this complexity figure can be dramatically higher than the one of the sparse ℓ_2 classifier. In the case of a *linear* SVM classifier, the computational complexity can indeed be reduced to $O(mn)$, but it is important to observe that neither the linear nor the kernel SVM alone can provide sparsity, which is the key property we are after in this work. Sparsification in nonlinear SVMs is generally considered unaffordable from the computational point of view. For linear SVM, instead, there are several works that aim to include sparsity as an additional property of the classifier, typically by adding an ℓ_1 penalty term to the SVM objective, see, e.g., [10], [38], [16]. In this case, however, the addition of the sparsification term worsens significantly the computational figure to $O(nm\min(n,m)^2)$, see, e.g., [16]. ⋆

Remark 5 (Online recursive training): The sparse ℓ_2 center classifier training procedure is amenable to efficient online implementation, since the class centers are easily updatable as soon as new data comes in. Denote by $\bar{x}(\nu)$ the centroid of one of the two classes when ν observations $\xi^{(1)},\ldots,\xi^{(\nu)}$ in that class are present: $\bar{x}(\nu)=\frac{1}{\nu}\sum_{j=1}^{\nu}\xi^{(j)}$. If a new observation $\xi^{(\nu+1)}$ in the same class becomes available, the new centroid will be

$$\begin{aligned}\bar{x}(\nu+1) &= \frac{1}{\nu+1}\sum_{j=1}^{\nu+1}\xi^{(j)}=\frac{1}{\nu+1}\left(\sum_{j=1}^{\nu}\xi^{(j)}+\xi^{(\nu+1)}\right)\\ &= \frac{\nu}{\nu+1}\bar{x}(\nu)+\frac{1}{\nu+1}\xi^{(\nu+1)}.\end{aligned}$$

This latter formula gives the new centroid as a weighted linear combination of the previous centroid and of the new observation. An online version of the procedure in Proposition 1 is thus readily obtained, in which only the current centroids are kept into memory and, as soon as a new datum is available, the corresponding centroid is updated (this takes $O(m)$ operations, or less if the datum is sparse) and the feature ranking is recomputed (this takes $O(m\log k)$ operations). A sparse ℓ_2 center classifier can therefore be trained online with $O(m)$ memory storage and $O(m\log k)$ operations per update. ⋆

Remark 6 (Sparsity-accuracy tradeoff): As it is customary with sparse methods, in practice a whole sequence of training problems is solved at different levels of sparsity, say from $k=1$ (only one feature selected) to $k=m$ (all features selected), accuracy is evaluated for each model via cross validation, and then the resulting sparsity-accuracy tradeoff curve is examined for the purpose of selection of the most suitable k level. Most feature selection methods, including sparse SVM, the Lasso [12], and the sparse Naive Bayes method [19], require repeatedly solving the training problem for each k, albeit typically warm-starting the optimization procedure with the solution from the previous k value. In the sparse ℓ_2 classifier, instead, one can fully order the vector $|\bar{x}^+-\bar{x}^-|$ only once, at a computational cost of $O(m\log m)$, and then *all* the optimal solutions are obtained, for any k, by simply selecting in Step 3 of Proposition 1 the first k elements of the ordered vector. ⋆

A. Mahalanobis distance classifier

A variant of the ℓ_2 centroid classifier is obtained by considering the Mahalanobis distance instead of the Euclidean distance. Letting S denote an estimated data covariance matrix, the Mahalanobis distance from a point z to a center $\theta^\pm$ is defined by

$$\mathrm{dist}_S(z,\theta^\pm)=(z-\theta^\pm)^\top S^{-1}(z-\theta^\pm).$$

This leads to the Mahalanobis training problem

$$\begin{aligned}\min_{\theta^+,\theta^-\in\mathbb{R}^m}\quad &\tfrac{1}{n_+}\textstyle\sum_{j\in\mathcal{J}_+}(x^{(j)}-\theta^+)^\top S^{-1}(x^{(j)}-\theta^+)\\ &+\tfrac{1}{n_-}\textstyle\sum_{j\in\mathcal{J}_-}(x^{(j)}-\theta^-)^\top S^{-1}(x^{(j)}-\theta^-)\end{aligned}$$

Classification of a new observation x in this setting is performed according to the sign of

$$\begin{aligned}\Delta_M(x)&=(x-\theta^-)^\top S^{-1}(x-\theta^-)-(x-\theta^+)^\top S^{-1}(x-\theta^+)\\ &=(\theta^- S^{-1}\theta^- - \theta^+ S^{-1}\theta^+)+2(\theta^+-\theta^-)^\top S^{-1}x.\end{aligned}$$

By introducing a change of variables of the type

$$\xi^{(j)}\doteq S^{-1/2}x^{(j)},\ j=1,\ldots,n;\quad \omega^\pm\doteq S^{-1/2}\theta^\pm,$$

where $S^{-1/2}$ is the matrix square root of S^{-1}, we see that the Mahalanobis training problem, in the new variables, becomes

$$\min_{\omega^+,\omega^-\in\mathbb{R}^m}\frac{1}{n_+}\sum_{j\in\mathcal{J}_+}\|\xi^{(j)}-\omega^+\|_2^2+\frac{1}{n_-}\sum_{j\in\mathcal{J}_-}\|\xi^{(j)}-\omega^-\|_2^2 \tag{12}$$

and the discrimination function, for $\xi=S^{-1/2}x$, becomes

$$\Delta_M(\xi)\quad=\quad(\|\omega^-\|_2^2-\|\omega^+\|_2^2)+2(\omega^+-\omega^-)^\top\xi.$$

Problem (12) is now a standard ℓ_2 center classifier problem, hence its sparse version can be readily solved by means of the Algorithm 1. It should however be observed that in this case one obtains sparsity in the transformed center difference $\omega^+-\omega^-$, which implies a selection of the transformed features in $\xi=S^{-1/2}x$. One relevant special case arises when $S=\mathrm{diag}(\sigma_1^2,\ldots,\sigma_m^2)$, in which case the data transformation $\xi=S^{-1/2}x$ simply amounts to normalizing each feature x_i by its standard deviation σ_i, that is $\xi_i=x_i/\sigma_i$, $i=1,\ldots,m$.

V. Training the sparse ℓ_1-center classifier

We next present an efficient and exact method for training a sparse ℓ_1-center classifier. We start by stating a preliminary instrumental result, whose proof is reported in the appendix Section VIII-A, and an ensuing definition.

Proposition 2 (Weighted ℓ_1 center): Given a real vector $z=(z_1,z_2,\ldots,z_p)$ and a nonnegative vector $w=(w_1,\ldots,w_p)$, consider the weighted ℓ_1 centering problem

$$d_w(z)\doteq\min_{\vartheta\in\mathbb{R}}\sum_{i=1}^{p}w_i|z_i-\vartheta|. \tag{13}$$

Let

$$W(\zeta)\doteq\sum_{\{i:\,z_i\le\zeta\}}w_i,\quad \bar{W}\doteq\sum_{i=1}^{p}w_i,$$

and

$$\bar{\zeta}\doteq\inf\{\zeta:\ W(\zeta)\ge\bar{W}/2\}. \tag{14}$$

Then, an optimal solution for problem (13) is given by

$$\vartheta^* = \mathrm{med}_w(z) \doteq \begin{cases} \bar{\zeta} & \text{if } W(\bar{\zeta}) > \frac{\bar{W}}{2} \\ \frac{1}{2}(\bar{\zeta} + \bar{\zeta}_+) & \text{if } W(\bar{\zeta}) = \frac{\bar{W}}{2}, \end{cases} \tag{15}$$

where $\bar{\zeta}_+ \doteq \min\{z_i,\, i = 1, \ldots, p \,:\, z_i > \bar{\zeta}\}$ is the smallest element in z that is strictly larger than $\bar{\zeta}$. ⋆

Definition 3 (Weighted median and dispersion): Given a row vector z and a nonnegative vector w of the same size, we define as the *weighted median* of z the optimal solution of problem (13) given in (15), and we denote it by $\mathrm{med}_w(z)$. We define as the *weighted median dispersion* the optimal value $d_w(z)$ of problem (13). We extend this notation to matrices, so that for a matrix $X \in \mathbb{R}^{m,n}$ we denote by $\mathrm{med}_w(X) \in \mathbb{R}^m$ a vector whose ith component is $\mathrm{med}_w(X_{i,:})$, where $X_{i,:}$ is the ith row of X, and we denote by $d_w(X) \in \mathbb{R}^m$ the vector of corresponding dispersions. ⋆

We now let $\mathcal{E}$ and $\mathcal{D}$ be defined as in Section IV, and we use the same notation as before for $\theta^{\pm}_{\mathcal{D}}$, $\theta^{\pm}_{\mathcal{E}}$, $x_{\mathcal{D}}$, $x_{\mathcal{E}}$. Let then J denote the objective to be minimized in (10). For fixed index set $\mathcal{D}$, we have that $J = J_{\mathcal{D}}$, where

$$\begin{aligned} J_{\mathcal{D}} &= \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)} - \theta^+_{\mathcal{D}} - \theta_{\mathcal{E}}\|_1 \\ &+ \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)} - \theta^-_{\mathcal{D}} - \theta_{\mathcal{E}}\|_1 \\ &= \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|(x^{(j)}_{\mathcal{D}} - \theta^+_{\mathcal{D}}) + (x^{(j)}_{\mathcal{E}} - \theta_{\mathcal{E}})\|_1 \\ &+ \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|(x^{(j)}_{\mathcal{D}} - \theta^-_{\mathcal{D}}) + (x^{(j)}_{\mathcal{E}} - \theta_{\mathcal{E}})\|_1 \\ &= \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)}_{\mathcal{D}} - \theta^+_{\mathcal{D}}\|_1 + \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)}_{\mathcal{E}} - \theta_{\mathcal{E}}\|_1 \\ &+ \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)}_{\mathcal{D}} - \theta^-_{\mathcal{D}}\|_1 + \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)}_{\mathcal{E}} - \theta_{\mathcal{E}}\|_1 \\ &= \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)}_{\mathcal{D}} - \theta^+_{\mathcal{D}}\|_1 + \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)}_{\mathcal{D}} - \theta^-_{\mathcal{D}}\|_1 \\ &+ \sum_{j=1}^{n} w_j \|x^{(j)}_{\mathcal{E}} - \theta_{\mathcal{E}}\|_1, \end{aligned}$$

where

$$w_j = \begin{cases} \frac{1}{n_+} & \text{if } j \in \mathcal{J}^+ \\ \frac{1}{n_-} & \text{if } j \in \mathcal{J}^- \end{cases}, \quad j = 1, \ldots, n.$$

We will next find the minimum of $J_{\mathcal{D}}$ with respect to $\theta^+_{\mathcal{D}}$, $\theta^-_{\mathcal{D}}$ and $\theta_{\mathcal{E}}$. To this end, we observe that $J_{\mathcal{D}}$ decouples as $J_{\mathcal{D}} = \sum_{i=1}^m J_{\mathcal{D},i}$, where for $i = 1, \ldots, m$,

$$J_{\mathcal{D},i} \doteq \begin{cases} \dfrac{1}{n_+} \displaystyle\sum_{j \in \mathcal{J}^+} |x^{(j)}_i - \theta^+_i| + \dfrac{1}{n_-} \displaystyle\sum_{j \in \mathcal{J}^-} |x^{(j)}_i - \theta^-_i|, & \text{if } i \in \mathcal{D} \\ \displaystyle\sum_{j=1}^{n} w_j |x^{(j)}_i - \theta_i|, & \text{if } i \notin \mathcal{D}. \end{cases} \tag{16}$$

The minimum of $J_{\mathcal{D}}$ is hence obtained by minimizing separately each component $J_{\mathcal{D},i}$. For $i \in \mathcal{D}$, we have that the optimal θ^+_i, θ^-_i are given by the (plain) medians of the $x^{(j)}_i$ values in the positive and in the negative class, respectively, that is, recalling (6),

$$i \in \mathcal{D} \Rightarrow \begin{array}{l} \theta^{+*}_i = \mu^+_i \doteq \mathrm{med}(\{x^{(j)}_i\}_{j \in \mathcal{J}^+}) \\ \theta^{-*}_i = \mu^-_i \doteq \mathrm{med}(\{x^{(j)}_i\}_{j \in \mathcal{J}^-}) \end{array} \Rightarrow J^*_{\mathcal{D},i} = d^+_i + d^-_i,$$

where d^+, d^- are the vectors of median dispersions in the positive and negative class, respectively, whose components are, for $i = 1, \ldots, m$,

$$\begin{aligned} d^+_i &\doteq \tfrac{1}{n_+} \textstyle\sum_{j \in \mathcal{J}^+} |x^{(j)}_i - \mu^+_i| \\ d^-_i &\doteq \tfrac{1}{n_-} \textstyle\sum_{j \in \mathcal{J}^-} |x^{(j)}_i - \mu^-_i|. \end{aligned} \tag{17}$$

For $i \notin \mathcal{D}$, instead, by observing that the entries of w in (16) are nonnegative, and applying Proposition 2, we obtain that the optimal solution is the weighted median of *all* the observations, that is

$$i \notin \mathcal{D} \Rightarrow \quad \theta^*_i \;=\; \mu_i \doteq \mathrm{med}_w(\{x^{(j)}_i\}_{j=1,\ldots,n}) \;\Rightarrow J^*_{\mathcal{D},i} = d_i,$$

where d is the vector of weighted median dispersions over all the observations, whose components are, for $i = 1, \ldots, m$,

$$\begin{aligned} d_i &\doteq \sum_{j=1}^{n} w_j |x^{(j)}_i - \mu_i| \\ &= \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} |x^{(j)}_i - \mu_i| + \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} |x^{(j)}_i - \mu_i|. \end{aligned} \tag{18}$$

We are now in position to discuss how to optimize over the choice of the set $\mathcal{D}$, that is how to decide which are the k indices that should belong to $\mathcal{D}$. First observe that $(d^+_i + d^-_i) \leq d_i$, for all $i = 1, \ldots, m$, since d_i is the optimal value of a minimization that constrains θ^+_i to be equal to θ^-_i, whereas $d^+_i + d^-_i$ is the optimal value of the same minimization without such constraint, and therefore its optimal objective value is no larger than d_i. Consider then the vector of differences

$$e \doteq (d^+ + d^-) - d \leq 0.$$

The smallest (i.e., most negative) entry in e corresponds to an index i for which it is maximally convenient (in terms of objective J decrease) choosing $i \in \mathcal{D}$ rather than $i \notin \mathcal{D}$; the second smallest entry in e corresponds to the second best choice, and so on. The best k indices to be included in $\mathcal{D}$ are therefore those corresponding to the k smallest entries of vector e. We have therefore proved the following

Proposition 3: The optimal solution of problem (10) is obtained as follows:

1) Compute the plain class medians

$$\begin{aligned} \mu^+ &\doteq \mathrm{med}(\{x^{(j)}\}_{j \in \mathcal{J}^+}) \\ \mu^- &\doteq \mathrm{med}(\{x^{(j)}\}_{j \in \mathcal{J}^-}) \end{aligned}$$

and the weighted median of all observations

$$\mu \doteq \mathrm{med}_w(\{x^{(j)}\}_{j=1,\ldots,n}),$$

where the weight vector w is such that, for $j = 1, \ldots, n$, $w_j = 1/n_+$ if $j \in \mathcal{J}^+$, and $w_j = 1/n_-$ if $j \in \mathcal{J}^-$.

2) Compute the median dispersion vectors d^+, d^- according to (17), and the weighted median dispersion vector d according to (18), and compute the difference vector

$$e \doteq (d^+ + d^-) - d.$$

3) Let $\mathcal{D}$ be the set of the indices of the k smallest elements in vector e, and let $\mathcal{E}$ be the complementary index set.
4) The optimal parameters θ^+, θ^- are given by

$$\begin{aligned} \theta^+ &= \mu^+_{\mathcal{D}} + \mu_{\mathcal{E}} \\ \theta^- &= \mu^-_{\mathcal{D}} + \mu_{\mathcal{E}}. \end{aligned}$$

The above procedure is next summarized in Algorithm 2.

Algorithm 2 Training the sparse ℓ_2 center classifier

```
Input: $x^{(j)} \in \mathbb{R}^m$: training data, $j = 1, ..., n$; $\mathbf{y} \in \mathbb{R}^n$: class label.
Compute the plain class medians:
    $\mu^+ = \text{med}(\{x^{(j)}\}_{j\in\mathcal{J}^+}), \quad \mu^- = \text{med}(\{x^{(j)}\}_{j\in\mathcal{J}^-})$;
Compute the weighted median of all training data:
    $\mu = \text{med}_w(\{x^{(j)}\}_{j=1,\dots,n})$;
Compute the median dispersion vectors: $d^+ = \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}|x^{(j)} - \mu^+|, \quad d^- = \frac{1}{n_-}\sum_{j\in\mathcal{J}^-}|x^{(j)} - \mu^-|$;
Compute the weighted median dispersion vector:
    $d = \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}|x^{(j)} - \mu| + \frac{1}{n_-}\sum_{j\in\mathcal{J}^-}|x^{(j)} - \mu|$;
Compute the difference vector: $e = (d^+ + d^-) - d$;
Define the set $\mathcal{D}$ of the indices corresponding to the $k$ largest absolute value elements in $e$ and the complementary index set $\mathcal{E}$;
Compute the optimal sparse centers:
    $\theta^+ = \mu^+_{\mathcal{D}} + \mu_{\mathcal{E}}, \quad \theta^- = \mu^-_{\mathcal{D}} + \mu_{\mathcal{E}}$.
```

Remark 7 (Numerical complexity for training the sparse ℓ_1 classifier): Computation of the medians in Step 1 of Proposition 3 can be performed with in $O(m)$ operations, see, e.g., [39]. Computation of the median dispersions requires $O(mn)$ operations, and finding the k smallest elements in vector e can be performed in $O(m \log k)$ operations, hence the whole procedure in Proposition 3 is performed in $O(mn) + O(m \log k)$ operations. Similar to the case discussed in Remark 6, also in the sparse ℓ_1 center classifier one needs to perform a full ordering of an m-vector only once in order to obtain all the sparse classifiers, for any requested sparsity level k. ⋆

Remark 8 (Complexity of the testing phase): We have seen that both the sparse ℓ_2-center and the sparse ℓ_1-center classifier have a training complexity of $O(mn) + O(m \log k)$. We next discuss about the complexity of testing and classifying a new sample $x \in \mathbb{R}^m$. By simply looking at the form of the discrimination function Δ_2 in (7) and Δ_1 in (9) we see that the $m-k$ entries of x corresponding to the zero entries of $\theta^+ - \theta^-$ give zero contribution to the function value, hence it suffices to compute the norms difference for the entries corresponding to the k nonzero entries of $\theta^+ - \theta^-$. Overall, we evaluate the discrimination function in $O(k)$ arithmetic operations. ⋆

TABLE I
TEXT DATASET SIZES

	TWTR	MPQA	SST
Number of features	273779	6208	16599
Number of samples	1600000	10606	79654

VI. Numerical Experiments

In this section we document an experimental evaluation of the proposed methods. Since these methods perform simultaneous feature selection and classification, they have a twofold application: they can be used as a nearest-center classifier with an integrated preprocessing step that performs feature selection, or they can be used as a mere feature selection method that acts as a preprocessing stage for a subsequent and perhaps more sophisticated classifier. We therefore evaluated the proposed methods both in terms of feature selection capability and of overall classification performance. We first considered the proposed methods as feature selection techniques, and we compared them with other feature selection methods. Then, we considered the proposed methods as full classifiers and we evaluated their classification performance.

A. Feature selection performance

We here evaluate the performance of the proposed methods for the feature selection task. The sparse ℓ_2-center classifier is tested in the context of sentiment classification on textual datasets. This is one of the most common application fields of the nearest centroid classifier. Instead, the sparse ℓ_1-center classifier is evaluated on gene expression datasets. Since this type of data is usually affected by the presence of many outliers, in this application the classifier with the ℓ_1 distance criteria can be preferred over the ℓ_2 version, see, e.g., [31].

1) Sparse ℓ_2-center classifier: We here compare the proposed sparse ℓ_2-center classifier with other feature selection methods for sentiment classification on text datasets. We considered three different datasets: the TwitterSentiment140 (TWTR) dataset [40], the MPQA Opinion Corpus Dataset [41], and the Stanford Sentiment Treebank (SST) [42]. Dataset sizes are given in Table I. Before classification, the datasets are normalized by dividing each feature by its standard deviation. Each dataset is then randomly split in a training (80% of the dataset) and test (20% of the dataset) set. The results reported in this section are an average of 50 different random splits of the dataset. For each dataset, we performed a two-stage classification procedure. In the first stage, we applied a feature selection method in order to reduce the number of features. Then, in the second stage we trained a classifier method, by employing only the selected features. In order to have a fair comparison, we used the same classifier in the second stage for all experiments, namely a linear support vector machine classifier.

We compared different feature selection methods: sparse ℓ_2-centers (ℓ_2-SC), sparse multinomial naive Bayes (SMNB), logistic regression with recursive feature selection (Logistic-RFE), ℓ_1-regularized logistic regression (Logistic-ℓ_1), Lasso, and Odds Ratio. We remark that the results of Logistic-RFE,

TABLE II
RNA GENE EXPRESSION DATASET SIZES

	Chowdary (Breast Cancer)	Chin (Breast Cancer)	Singh (Prostate Cancer)
N. features	22283	22215	12600
N. samples	104	118	102

Logistic-ℓ_1 and Lasso are not reported on some datasets, since their computing time resulted to be too high. Figure 1 shows the classification accuracy and the average run time of the different feature selection methods. In all plots, the vertical bars represent the variation intervals over the 50 random tests within plus or minus one standard deviation. These plots show that the sparse ℓ_2-centers is competitive with other feature selection methods in terms of accuracy performance of the classifier, while its run time is significantly lower than most of the other feature selection methods. The only method that has a comparable computational time is Odds Ratio, but its performance is poorer in terms of accuracy.

2) Sparse ℓ_1-center classifiers: We compared the proposed sparse ℓ_1-center classifier (ℓ_1-SC) with other feature selection methods for RNA gene expression classification. We used the same two-stage procedure described for the ℓ_2 case in the previous section. In the first stage, we compared the sparse ℓ_1-center classifier with the same feature selection methods considered in the previous section. Then, in the second stage we used a linear SVM classifier, as done in the ℓ_2 case. We considered three datasets: Chin dataset [43], Chowdary dataset [44], and Singh dataset [45]. The details of the datasets are summarized in Table II. Before feature selection, we normalized the datasets by dividing each feature by its standard deviation. In addition, since some features have negative values and the SMNB method is defined only for positive features, we shifted all the features in order to have only positive values. As done in the ℓ_2 case, we split each dataset in a training (80% of the dataset) and test (20% of the dataset) set, and we tested 50 of such random splits. Figure 2 shows the balanced accuracy of the classifier and the average run time of the feature selection methods considered in the evaluation. In this experiment we observe again that the proposed method provides an accuracy performance which is similar to that of state-of-the-art techniques, but with a computational time which can be orders of magnitude lower.

B. Overall classification performance

In the previous experiments we evaluated the performance of the proposed methods only in terms of feature selection, that is, in a setup where the proposed methods are used as a variable selection first stage, to be followed by a second stage constituted by a plain classifier (a linear SVM, in the considered examples). In this section we consider instead the proposed methods as full-fledged classifiers, which can simultaneously perform feature selection and classification. We first evaluate the performance gain provided by the proposed methods compared with a nearest-center classifier without any feature selection or combined with other feature selection methods. Then, we also perform a comparison between the two variants of the sparse center classifiers, highlighting their respective strengths.

1) Sparse ℓ_2-center classifier: We compare the proposed sparse ℓ_2-center classifier with a ℓ_2-center classifier trained on all the features or combined with one of the feature selection methods introduced in the previous section. We tested the classifiers in the context of sentiment classification on textual datasets and we preprocessed the data as explained previously in the experiments on feature selection. Figure 3 shows the results obtained on the SST dataset. The results show that the proposed method provided the best overall performance, being competitive at all sparsity levels. Moreover, we observe that, when the feature cardinality is high, the accuracy of the best classifiers, namely ℓ_2-SC, Logistic-RFE and SMNB, remains constant or slightly decreases as the feature cardinality increases.

Next, we compare the sparse ℓ_2 center classifier against the linear SVM, at various feature cardinality levels. Since linear SVM does not perform feature selection natively, we used the proposed ℓ_2 center classifier as feature selector for the linear SVM. Figure 4 shows the classification accuracy and the average run time of the two methods. We observe that the proposed classifier has competitive classification performance, but with a significantly lower run time. In addition, it is worth noting that the classification performance of linear SVM degrades as the number of features increases. This highlights again the importance of feature selection to avoid the curse of dimensionality.

2) Sparse ℓ_1-center classifier: Analogously to the ℓ_2 case, in this experiment we compared the proposed sparse ℓ_1-center classifier with a ℓ_1-center classifier without any feature selection or combined with one of the feature selection methods previously considered. We tested the classifiers on RNA gene expression datasets, preprocessing the data with the same procedure described in the experiments on feature selection. Figure 5 shows the results obtained on the Singh dataset. We observe that the classifiers provide the best performance at very high sparsity levels, and then the accuracy significantly decreases as the feature cardinality increases. This shows again the importance of feature selection, especially when the dataset is very noisy. Moreover, we observe that Logistic-RFE and Logistic-ℓ_1 are the only two feature selection methods that, combined with a standard nearest-center classifier, outperform the proposed method. However, as we have already shown in the previous section, these two methods are very computationally expensive and may not be viable on large datasets.

3) Comparison between the two sparse center classifiers: We next compare the classification performance of the two proposed sparse center classifiers (ℓ_2 and ℓ_1 based). In order to better appreciate the differences between the two classifiers, we first considered a synthetic example. In this example, each observation $x^{(j)}$ of the dataset is obtained as:

$$x^{(j)} \sim \begin{cases} \mathcal{N}(\mu, \sigma I), & \text{with probality } 1-p \\ \mathcal{U}(0, b), & \text{with probality } p, \end{cases}$$

where we set $\sigma = 1$, $b = 5$, $\mu = \mu^+$ if $j \in \mathcal{J}^+$ and $\mu = \mu^-$ if $j \in \mathcal{J}^-$, and $\mu^+, \mu^- \in \mathbb{R}^m$ are samples of a

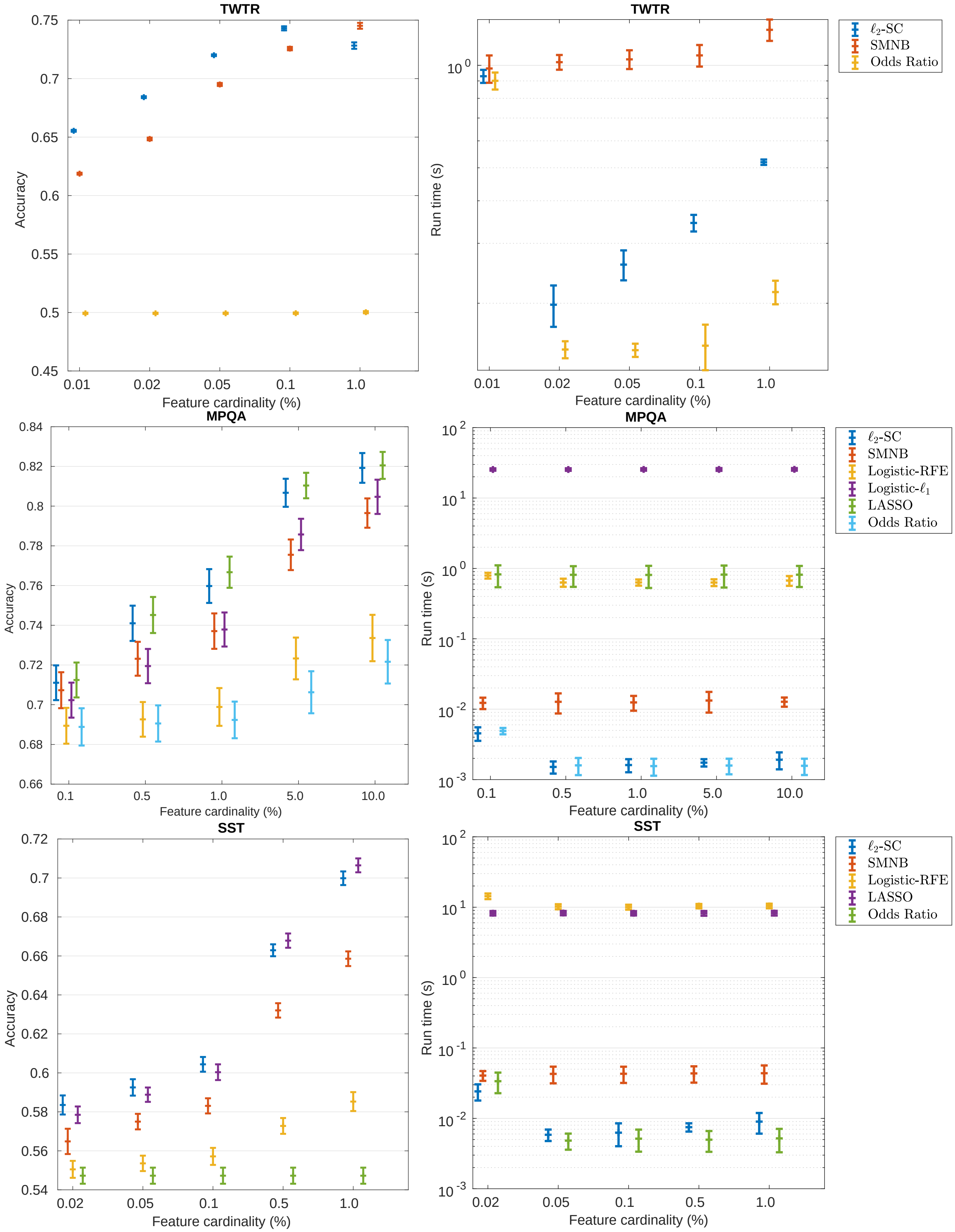

Fig. 1. Comparison of feature selection methods + second stage linear SVM classifier. Panels show the resulting classification accuracy of the linear SVM and average run time of the various feature selection methods on different datasets, as a function of the cardinality of the considered features.

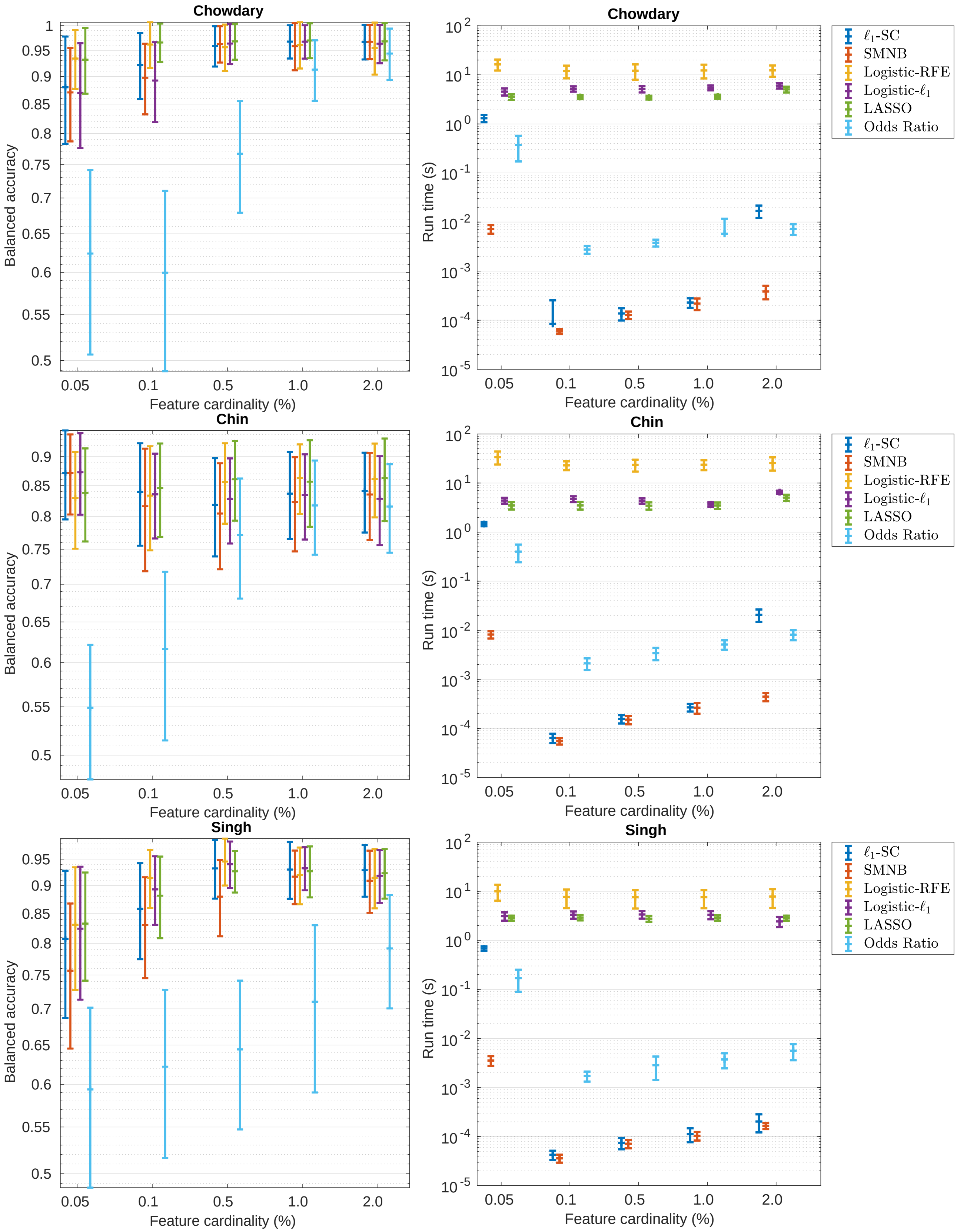

Fig. 2. Comparison of feature selection methods + second stage linear SVM classifier. Panels show the resulting balanced classification accuracy of the linear SVM and average run time of the various feature selection methods on different datasets, as a function of the cardinality of the considered features.

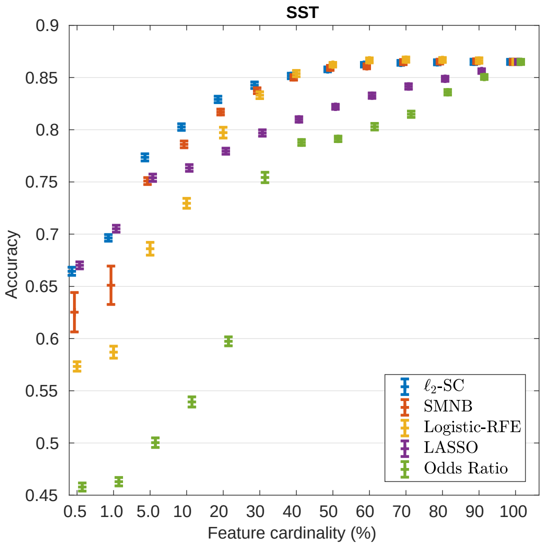

Fig. 3. Comparison between ℓ_2 sparse center classifier and feature selection methods + second stage (plain) ℓ_2-center classifier. The performance of the various methods are evaluated computing the classification accuracy on the SST dataset as a function of the feature cardinality. The case with feature cardinality 100% corresponds to a plain nearest center classifier without any feature selection.

uniform distribution $\mathcal{U}(0,1)$. The parameter p represents the probability of outliers. Figure 6 shows the performance of the two sparse center classifiers as function of p. The results show that if p is zero or close to zero the classifier with the ℓ_2 distance provides slightly better performance, but, as p increases, the accuracy of the sparse ℓ_1-center classifier degrades more slowly, outperforming the ℓ_2 version. This shows that the sparse ℓ_1-center classifier is more robust to outliers. In addition to this experiment on synthetic data, we also show a comparison on a real dataset. We considered a gene expression dataset, since these type of data are known to be usually affected by the presence of many outliers. Figure 7 shows the balanced accuracy of the two sparse center classifiers on the Singh dataset and we observe that the sparse ℓ_1-center classifier outperforms the ℓ_2 variant.

VII. Conclusions

In this paper we proposed two types of sparse center classifiers, based respectively on ℓ_1 and the ℓ_2 distance metrics. The proposed methods perform efficient simultaneous feature selection and classification, and we formally proved that, for both cases, the training algorithm selects the globally optimal set of features and computes the ensuing sparse classifier in $O(mn) + O(m \log k)$ operations. Testing and classification of a new sample is also performed extremely efficiently in $O(k)$ operations. The experimental results show that the proposed methods achieve accuracy levels that are on par with state-of-the-art feature selection methods, while being substantially faster.

VIII. Appendix

A. Proof of Proposition 2

Let $\tilde{w} \doteq w/\bar{W}$. Since $\tilde{w} \geq 0$ and $\sum_{i=1}^{p} \tilde{w}_i = 1$, it can be interpreted as the probability distribution of a discrete random variable Z with support in $z_1, \ldots, z_p$, and corresponding probability mass $\tilde{w}_1, \ldots, \tilde{w}_p$. Note that values in vector z may be repeated, in which case the probability mass relative to a repeated support point is the sum of the corresponding probability values in vector $\tilde{w}$. With such stochastic interpretation, the objective in (13) can be written in terms of the expectation $\mathbb{E}\{|Z - \vartheta|\}$, and then the problem becomes

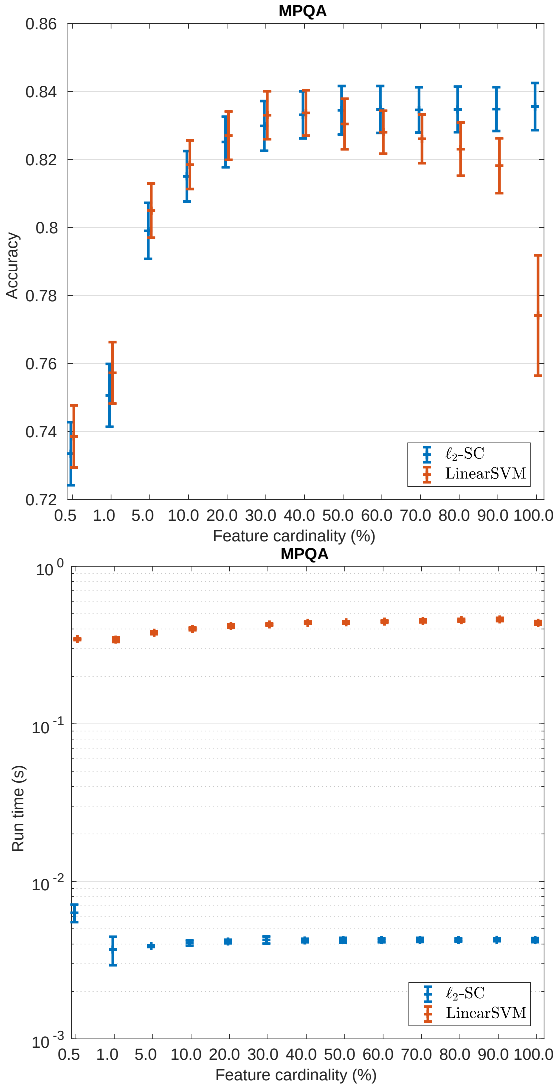

Fig. 4. Classification accuracy and average run time on MPQA dataset.

$$d_w(z) = \bar{W} \min_{\vartheta \in \mathbb{R}} \mathbb{E}\{|Z - \vartheta|\}. \tag{19}$$

When Z has an absolutely continuous distribution, it is well known (see, e.g., [46]) that the value ϑ^* that minimizes the absolute expected loss is the *median* of the probability distribution of Z, that is, the 0.5 quantile of the distribution. In the case of a discrete probability distribution, the definition of median is any value μ such that

$$\text{Prob}\{Z \leq \mu\} \geq \frac{1}{2}, \quad \text{and} \quad \text{Prob}\{Z \geq \mu\} \geq \frac{1}{2}. \tag{20}$$

Now, suppose that μ is a median for our discrete random variable Z, and consider any given $\vartheta > \mu$. If $Z \leq \mu$, then

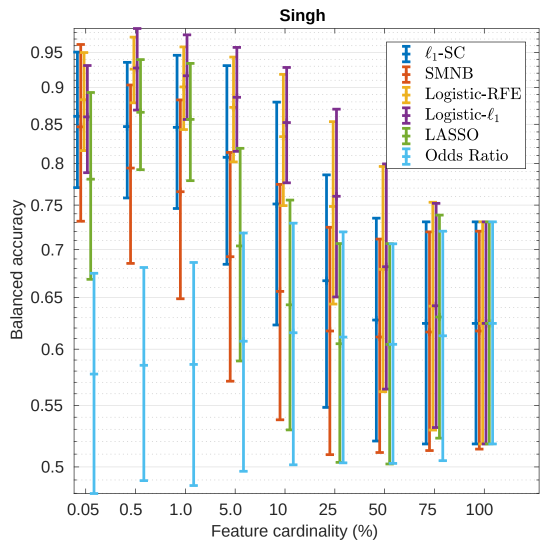

Fig. 5. Comparison between ℓ_1 sparse center classifier and feature selection methods + second stage (plain) ℓ_1-center classifier. The performance of the various methods are evaluated computing the balanced classification accuracy on the Singh dataset as a function of the feature cardinality. The case with feature cardinality equals to 100% corresponds to a plain nearest center classifier without any feature selection.

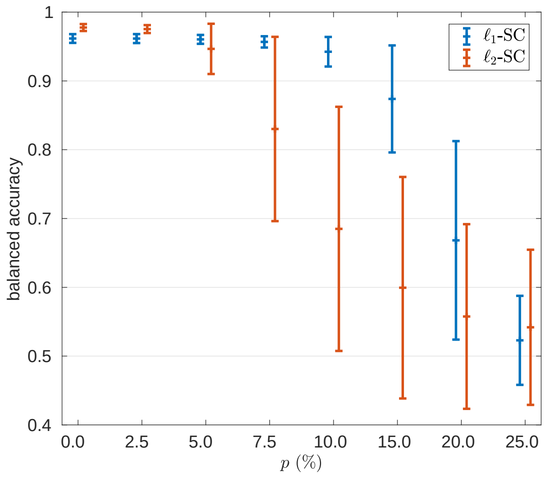

Fig. 6. Classification accuracy on synthetic data (feature cardinality = 2%).

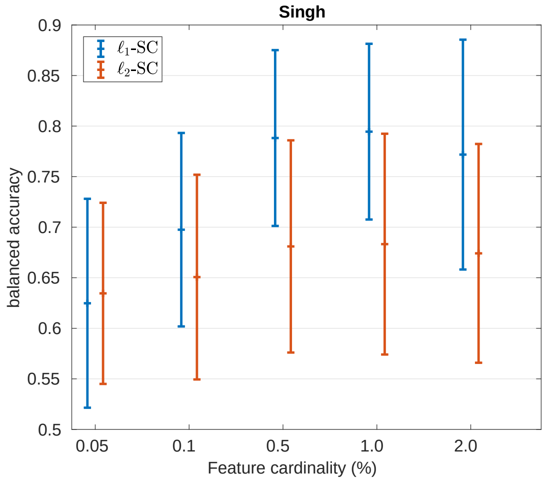

Fig. 7. Classification accuracy on Singh dataset.

$|Z-\mu| = \mu - Z$ and since $\mu < \vartheta$ we also have $Z < \vartheta$ whence $|Z-\vartheta| = \vartheta - Z$, and therefore

$$|Z-\vartheta|-|Z-\mu| = (\vartheta-Z)-(\mu-Z) = \vartheta-\mu, \quad \text{for } Z \le \mu.$$

If instead $Z > \mu$, then

$$\begin{aligned}
|Z-\vartheta|-|Z-\mu| &= |Z-\vartheta|-(Z-\mu)\\
&= |Z-\mu+\mu-\vartheta|-(Z-\mu)\\
&\ge |Z-\mu|-|\mu-\vartheta|-(Z-\mu)\\
&= (Z-\mu)-(\vartheta-\mu)-(Z-\mu)\\
&= -(\vartheta-\mu), \quad \text{for } Z > \mu.
\end{aligned}$$

Therefore, for any given $\vartheta > \mu$, we have that

$$\begin{aligned}
\mathbb{E}\{|Z-\vartheta|-|Z-\mu|\} &\ge (\vartheta-\mu)\text{Prob}\{Z\le\mu\}\\
&- (\vartheta-\mu)\text{Prob}\{Z>\mu\}\\
&= (\vartheta-\mu)(\text{Prob}\{Z\le\mu\}\\
&- \text{Prob}\{Z>\mu\})\\
&= (\vartheta-\mu)\left(2\text{Prob}\{Z\le\mu\}-1\}\right)\\
&\ge 0, \quad \text{for all } \vartheta>\mu.
\end{aligned}$$

where the last inequality follows from the fact that μ is a distribution median and hence from the definition in (20) it holds that $\text{Prob}\{Z \le \mu\} \ge 1/2$. The whole reasoning can be repeated symmetrically for any given $\vartheta < \mu$, obtaining

$$\begin{aligned}
|Z-\vartheta|-|Z-\mu| &\ge -(\mu-\vartheta), \quad \text{for } Z<\mu,\\
|Z-\vartheta|-|Z-\mu| &= (\mu-\vartheta), \quad \text{for } Z\ge\mu.
\end{aligned}$$

Then again

$$\begin{aligned}
\mathbb{E}\{|Z-\vartheta|-|Z-\mu|\} &\ge -(\mu-\vartheta)\text{Prob}\{Z<\mu\}\\
&+ (\mu-\vartheta)\text{Prob}\{Z\ge\mu\}\\
&= (\mu-\vartheta)(\text{Prob}\{Z\ge\mu\}\\
&- \text{Prob}\{Z<\mu\})\\
&= (\mu-\vartheta)\left(2\text{Prob}\{Z\ge\mu\}-1\}\right)\\
&\ge 0, \quad \text{for all } \vartheta<\mu,
\end{aligned}$$

where the last inequality follows from the fact that μ is a distribution median and hence from the definition in (20) it holds that $\text{Prob}\{Z \ge \mu\} \ge 1/2$. Putting things together, we have that, for all ϑ,

$$\mathbb{E}\{|Z-\vartheta|\}-\mathbb{E}\{|Z-\mu|\} = \mathbb{E}\{|Z-\vartheta|-|Z-\mu|\} \ge 0,$$

which implies that the minimum of $\mathbb{E}\{|Z-\mu|\}$ is attained at $\vartheta = \mu$, where μ is a median of the distribution.

We next conclude the proof by showing that ϑ^* in (15) is indeed a median, in the sense of definition (20). Observe first that $W(\zeta) \doteq \sum_{i:z_i\le\zeta} w_i$ is proportional to the cumulative distribution function of Z, that is $W(\zeta) = \bar{W}\tilde{W}(\zeta)$, $\tilde{W}(\zeta) \doteq \text{Prob}\{Z \le \zeta\}$, and that (14) implies that $\tilde{W}(\bar\zeta) \ge 1/2$, and $\tilde{W}(\zeta) < 1/2$ for all $\zeta < \bar\zeta$. Also, since by definition of $\bar\zeta_+$ no probability mass is present in the interior of the interval $[\bar\zeta, \bar\zeta_+]$, we have from (15) that $\tilde{W}(\vartheta^*) \equiv \tilde{W}(\bar\zeta)$. Then, from (15) it follows immediately that $\text{Prob}\{Z \le \vartheta^*\} = \tilde{W}(\vartheta^*) \equiv \tilde{W}(\bar\zeta) \ge 1/2$, which shows that ϑ^* satisfies the condition on the left in (20). We next analyze the condition on the right in (20), which concerns verifying that Prob$\{Z \geq \vartheta^*\} \geq 1/2$. To this purpose, we distinguish two cases: case (a), where $\tilde{W}(\vartheta^*) > 1/2$, and case (b), where $\tilde{W}(\vartheta^*) = 1/2$. In case (a), we have $\vartheta^* \equiv \bar{\zeta}$ and hence, as discussed above, $\tilde{W}(\zeta) < 1/2$ for all $\zeta < \vartheta^*$, which implies that Prob$\{Z < \vartheta^*\} < 1/2$ (while Prob$\{Z \leq \vartheta^*\} \geq 1/2$, since there is a positive probability mass at ϑ^*), and therefore Prob$\{Z \geq \vartheta^*\} = 1 -$ Prob$\{Z < \vartheta^*\} > 1/2$. In case (b), we have instead

$$\begin{aligned}\text{Prob}\{Z \geq \vartheta^*\} &= \text{Prob}\{Z = \vartheta^*\} + \text{Prob}\{Z > \vartheta^*\} \\ &= \text{Prob}\{Z = \vartheta^*\} + 1 - \text{Prob}\{Z \leq \vartheta^*\} \\ &= \text{Prob}\{Z = \vartheta^*\} + 1/2 \\ &= 1/2,\end{aligned}$$

where the last equality follows from the fact that in case (b) we have Prob$\{Z = \vartheta^*\} = 0$, since ϑ^* is the mid point of the interval $[\bar{\zeta}, \bar{\zeta}_+]$, in the interior of which there is no probability mass, by construction. □

References

[1] A. McCallum, K. Nigam *et al.*, "A comparison of event models for Naive Bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48.

[2] L. Jiang, D. Wang, Z. Cai, and X. Yan, "Survey of improving Naive Bayes for classification," in *International Conference on Advanced Data Mining and Applications*. Springer, 2007, pp. 134–145.

[3] V. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artif Intell Rev*, vol. 52, pp. 803–855, 2019.

[4] C. Manning, P. Raghavan, and H. Schütze, "Vector space classification," *Introduction to Information Retrieval*, 2008.

[5] I. Levner, "Feature selection and nearest centroid classification for protein mass spectrometry," *BMC bioinformatics*, vol. 6, no. 1, p. 68, 2005.

[6] S. Shalev-Shwartz and N. Srebro, "SVM optimization: inverse dependence on training set size," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 928–935.

[7] O. Chapelle, "Training a support vector machine in the primal," *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.

[8] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *European Journal of Operational Research*, vol. 265, no. 3, pp. 993–1004, 2018.

[9] H. Zou, "Classification with high dimensional features," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 11, no. 1, p. e1453, 2019.

[10] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1229–1243, 2003.

[11] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse svm for feature selection on very high dimensional datasets," in *International Conference on Machine Learning (ICML)*, 2010.

[12] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[13] A. Y. Ng, "Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[14] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and Naive Bayes," in *ICML*, vol. 99, 1999, pp. 258–267.

[15] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *International Conference on Machine Learning (ICML)*, vol. 97, no. 412-420, 1997, p. 35.

[16] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, "1-norm support vector machines," in *Advances in neural information processing systems*, 2004, pp. 49–56.

[17] C. J. Veenman and D. M. Tax, "Less: a model-based classifier for sparse subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1496–1500, 2005.

[18] C. J. Veenman and A. Bolck, "A sparse nearest mean classifier for high dimensional multi-class problems," *Pattern recognition letters*, vol. 32, no. 6, pp. 854–859, 2011.

[19] A. Askari, A. d'Aspremont, and L. El Ghaoui, "Naive feature selection: Sparsity in naive bayes," *arXiv preprint arXiv:1905.09884*, 2019.

[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[21] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 325–327, 1976.

[22] R. J. Samworth *et al.*, "Optimal weighted nearest neighbour classifiers," *The Annals of Statistics*, vol. 40, no. 5, pp. 2733–2763, 2012.

[23] J. Gou, Z. Yi, L. Du, and T. Xiong, "A local mean-based k-nearest centroid neighbor classifier," *The Computer Journal*, vol. 55, no. 9, pp. 1058–1071, 2012.

[24] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *International conference on database theory*. Springer, 1999, pp. 217–235.

[25] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.

[26] J. M. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, pp. 599–608.

[27] S. Har-Peled, P. Indyk, and R. Motwani, "Approximate nearest neighbor: Towards removing the curse of dimensionality," *Theory of computing*, vol. 8, no. 1, pp. 321–350, 2012.

[28] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6567–6572, 2002.

[29] E.-H. S. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," in *European conference on principles of data mining and knowledge discovery*. Springer, 2000, pp. 424–431.

[30] H. Park, M. Jeon, and J. B. Rosen, "Lower dimensional representation of text data based on centroids and least squares," *BIT Numerical mathematics*, vol. 43, no. 2, pp. 427–448, 2003.

[31] P. Hall, D. Titterington, and J.-H. Xue, "Median-based classifiers for high-dimensional data," *Journal of the American Statistical Association*, vol. 104, no. 488, pp. 1597–1608, 2009.

[32] R. Jörnsten, "Clustering and classification based on the ℓ_1 data depth," *Journal of Multivariate Analysis*, vol. 90, no. 1, pp. 67–89, 2004.

[33] A. R. Dabney and J. D. Storey, "Optimality driven nearest centroid classification from genomic data," *PLoS One*, vol. 2, no. 10, p. e1002, 2007.

[34] R. Tibshirani, T. Hastie, B. Narasimhan, S. Soltys, G. Shi, A. Koong, and Q.-T. Le, "Sample classification from protein mass spectrometry, by 'peak probability contrasts'," *bioinformatics*, vol. 20, no. 17, pp. 3034–3044, 2004.

[35] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[36] E. Frank and S. Kramer, "Ensembles of nested dichotomies for multi-class problems," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 39.

[37] A. C. Lorena and A. C. De Carvalho, "Building binary-tree-based multiclass classifiers using separability measures," *Neurocomputing*, vol. 73, no. 16-18, pp. 2837–2845, 2010.

[38] O. Mangasarian, "Exact 1-norm support vector machines via unconstrained convex differentiable minimization," *Journal of Machine Learning Research*, vol. 7, pp. 1517–1530, 2006.

[39] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan, "Time bounds for selection," *J. Comput. Syst. Sci.*, vol. 7, no. 4, pp. 448–461, 1973.

[40] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[41] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[42] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[43] K. Chin, S. DeVries, J. Fridlyand, P. T. Spellman, R. Roydasgupta, W.-L. Kuo, A. Lapuk, R. M. Neve, Z. Qian, T. Ryder *et al.*, "Genomic and

transcriptional aberrations linked to breast cancer pathophysiologies," *Cancer cell*, vol. 10, no. 6, pp. 529–541, 2006.

[44] D. Chowdary, J. Lathrop, J. Skelton, K. Curtin, T. Briggs, Y. Zhang, J. Yu, Y. Wang, and A. Mazumder, "Prognostic gene expression signatures can be measured in tissues collected in rnalater preservative," *The journal of molecular diagnostics*, vol. 8, no. 1, pp. 31–39, 2006.

[45] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie *et al.*, "Gene expression correlates of clinical prostate cancer behavior," *Cancer cell*, vol. 1, no. 2, pp. 203–209, 2002.

[46] J. Haldane, "Note on the median of a multivariate distribution," *Biometrika*, vol. 35, no. 3-4, pp. 414–417, 1948.

Giuseppe C. Calafiore is a full professor at DET, Politecnico di Torino, where he coordinates the Control Systems and Data Science group, and an associate fellow of the IEIIT-CNR. Dr. Calafiore held visiting positions at the Information Systems Laboratory (ISL), Stanford University, California, in 1995; at the Ecole Nationale Supérieure de Techniques Avancées (ENSTA), Paris, in 1998; and at the University of California at Berkeley, in 1999, 2003, 2007, 2017, 2018 and 2019, where he lately taught a Master course on Financial Data Science. He was a Senior Fellow at the Institute of Pure and Applied Mathematics (IPAM), University of California at Los Angeles, in 2010. Dr. Calafiore is the author of about 200 journal and conference proceedings papers, and of eight books. He is a Fellow of the IEEE. He received the IEEE Control System Society "George S. Axelby" Outstanding Paper Award in 2008. His research interests are in the fields of convex optimization, identification and control of uncertain systems, with applications ranging from finance and economic systems to robust control, machine learning, and data science.

Giulia Fracastoro received the B.Sc. and M.Sc. degrees in applied mathematics from Politecnico di Torino, Italy, in 2011 and 2013, respectively. She received the Ph.D. degree in Electronics and Telecommunications Engineering at Politecnico di Torino in 2017. In 2016, she was a Visiting Student at the Signal Processing Laboratory, EPFL, Lausanne, Switzerland. She is currently an Assistant Professor at the Department of Electronics and Telecommunications (DET), Politecnico di Torino. Her main research interests include machine learning, deep learning, and signal processing.