Fast Feature Selection via Sparse ℓ_2 and ℓ_1 Center Classifiers

DISMA Special Session on Neural Systems and Learning

G.C. Calafiore and G. Fracastoro
DET – Politecnico di Torino

Outline

Classifiers

Preliminaries

- Input data matrix:
$$X = \left[x^{(1)} \ \cdots \ x^{(n)} \right] \in \mathbb{R}^{m,n},$$
columns $x^{(j)} \in \mathbb{R}^m$, $j = 1, \ldots, n$, contain feature vectors from n observations.
- Output data vector: $\mathbf{y} \in \mathbb{R}^n$ 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 $y = -1$).
- A parametric classifier is thus a function $G_\theta : \mathbb{R}^m \to \{-1, 1\}$.
- θ represents the *parameters* of the classifier, which are *learned* from data.

Classifiers

Example: the Bernoulli Naive Bayes classifier

- In the Bernoulli Naive Bayes (BNB) model the features are represented by boolean values (e.g., 0 or 1). For instance, $x_i = 1$ if the i-th term of a dictionary is present in a document and $x_i = 0$ otherwise.
- Given the class $C_\pm$, each x_i is an independent Bernoulli variable with success probability $\theta_i^\pm$, that is, for $i = 1, \ldots, m$,

$$\mathrm{Prob}\{x_i = 1 | C_\pm\} = \theta_i^\pm, \quad \text{and} \quad \mathrm{Prob}\{x_i = 0 | C_\pm\} = 1 - \theta_i^\pm.$$

- Using Bayes' rule one obtains

$$\begin{aligned} \log p(C_\pm | x) &\propto \log p(C_\pm) + \sum_{i=1}^{m} \log p(x_i | C_\pm) \\ &= \log p(C_\pm) + x^\top \log \theta^\pm + (\mathbf{1} - x)^\top \log(\mathbf{1} - \theta^\pm). \end{aligned}$$

Classifiers

Example: the Bernoulli Naive Bayes classifier

- Classify x in C_+ if $\log p(C_+|x) > \log p(C_-|x)$, and in C_- otherwise.
- Classification is based in the sign of the discrimination function

$$
\begin{aligned}
\Delta_B(x) &= \log \frac{p(C_+)}{p(C_-)} + \mathbf{1}^\top(\log(\mathbf{1}-\theta^+) - \log(\mathbf{1}-\theta^-)) \\
&\quad + x^\top \left(\log \theta^+ - \log(\mathbf{1}-\theta^+) - \log \theta^- + \log(\mathbf{1}-\theta^-)\right) \\
&= v_B + x^\top w_B,
\end{aligned}
$$

where

$$
\begin{aligned}
v_B &\doteq \log \frac{p(C_+)}{p(C_-)} + \mathbf{1}^\top(\log(\mathbf{1}-\theta^+) - \log(\mathbf{1}-\theta^-)) \\
w_B &\doteq \log \frac{\theta^+ \odot (\mathbf{1}-\theta^-)}{\theta^- \odot (\mathbf{1}-\theta^+)},
\end{aligned}
$$

and $\odot$ denotes element-wise vector product.

Classifiers

Example: the Bernoulli Naive Bayes classifier

- The discrimination function is *linear*.
- Feature x_i has no influence on the classification iff the corresponding coefficient in w_B is zero.
- This happens if and only if $\theta_i^+ = \theta_i^-$.
- A sparse classifier is obtained iff w_B is sparse $\Leftrightarrow$ $(\theta^+ - \theta^-)$ is sparse.

- Training a Sparse Naive Bayes classifier in the general Multinomial case is a computationally complex problem.
- Approximation schemes exist [e.g., Askari, d'Aspremont, El Ghaoui, 2019].
- We next discuss sparse classifiers that are trainable exactly and efficiently.

Center-based classifiers

Preliminaries

- The *nearest centroid classifier* 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

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

- $\mathcal{J}^{+} \doteq \{j \in \{1, \ldots, n\} : y_j = +1\}$ and $\mathcal{J}^{-} \doteq \{j \in \{1, \ldots, n\} : y_j = -1\}$ contain the indices of the observations in the positive and negative class, respectively, and n_{+}, n_{-} are the corresponding cardinalities.

Center-based classifiers

Nearest centroid classifier

- A new observation vector x is classified as positive or negative according to the sign of

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

- The discrimination surface for the centroid classifier is linear w.r.t. x, since

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

- The coefficient in the linear term of the classifier is $w \doteq \bar{x}^+ - \bar{x}^-$.
- 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.

Nearest ℓ_2 center classifier
Minimization form

- The ℓ_2 centroids can be seen as the optimal solutions to the following optimization problem:

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

- That is, the centroids are the points that minimize the average squared distance to the samples within each class.

Nearest ℓ_1 center classifier

Minimization form

- The minimization form suggests considering different types of metrics for computing centers.
- In particular, there exist an extensive literature on the favorable properties of the ℓ_1 norm criterion, which is well known to provide center estimates that are robust to outliers.
- The natural ℓ_1 version of the centering problem is

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

which we shall call the (plain) ℓ_1-center classifier training problem.

Nearest ℓ_1 center classifier

- It is known that an optimal solution to ℓ_1-center classifier is obtained by taking $\theta^\pm$ to be the (entry-wise) *median* of the values in each class:

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

- The classification is made according to the sign of

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

- The discrimination $\Delta_1(x)$ is not linear in x.
- However, the contribution to $\Delta_1(x)$ from the i-th feature x_i is identically zero whenever $\theta_i^- = \theta_i^+$.

Sparse ℓ_1 and ℓ_2 center classifiers

- For both the ℓ_2 and the ℓ_1 distance criteria, the discrimination is insensitive to the i-th feature whenever $\theta_i^+ - \theta_i^- = 0$.
- The *sparse classifiers* that we introduce next are aimed precisely at computing optimal class centers such that the center difference $\theta^+ - \theta^-$ is k-sparse.
- Formally, we impose 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.

Sparse ℓ_1 and ℓ_2 center classifiers

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

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

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

$$\begin{array}{ll}\displaystyle\min_{\theta^+,\theta^-\in\mathbb{R}^m} & \frac{1}{n_+}\sum_{j\in\mathcal{J}^+}\|x^{(j)} - \theta^+\|_2^2 + \frac{1}{n_-}\sum_{j\in\mathcal{J}^-}\|x^{(j)} - \theta^-\|_2^2 \\ \text{subject to:} & \|\theta^+ - \theta^-\|_0 \le k,\end{array}$$

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

Sparse ℓ_1 and ℓ_2 center classifiers

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

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

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

$$\begin{array}{ll} \min\limits_{\theta^+, \theta^- \in \mathbb{R}^m} & \frac{1}{n_+} \sum_{j \in \mathcal{J}^+} \|x^{(j)} - \theta^+\|_1 + \frac{1}{n_-} \sum_{j \in \mathcal{J}^-} \|x^{(j)} - \theta^-\|_1 \\ \text{subject to:} & \|\theta^+ - \theta^-\|_0 \le k, \end{array}$$

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

Training the sparse ℓ_2-center classifier

Notation

- We let $\mathcal{E}$ 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 write $x_{\mathcal{D}}$ 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

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

- If $\mathcal{E}$ is the set of the indices where $\theta^+ - \theta^-$ is zero, so that $\theta^+_{\mathcal{E}} - \theta^-_{\mathcal{E}} = 0$, then

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

  whence

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

Training the sparse ℓ_2-center classifier

Result

Proposition 1

An optimal solution of the sparse ℓ_2-center problem is obtained as follows:

1. *Compute the standard class centroids $\bar{x}^+$, $\bar{x}^-$;*
2. *Compute the centroids midpoint $\tilde{x} = (\bar{x}^+ + \bar{x}^-)/2$, 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*

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

Sparse ℓ_2-center classifier

Numerical complexity

- 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).
- The whole procedure thus 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), adding exact sparsity comes at the quite moderate extra cost of $O(m \log k)$ operations.

Sparse ℓ_2-center classifier

Online implementation

- 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

$$\bar{x}(\nu+1) = \frac{\nu}{\nu+1}\bar{x}(\nu) + \frac{1}{\nu+1}\xi^{(\nu+1)}.$$

- Only the current centroids need be kept into memory.
- 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).

Sparse ℓ_2-center classifier

Sparsity-accuracy tradeoff

- In practice, a whole sequence of training problems need be solved at different levels of sparsity, say from $k = 1$ (only one feature selected) to $k = m$ (all features selected).
- At each k accuracy is evaluated 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, and the sparse Naive Bayes method, 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 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.

Sparse ℓ_2-center classifier

The Mahalanobis variant

- 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

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

- Maps to the standard ℓ_2-center case in transformed variable space

$$\xi \doteq S^{-1/2} x$$

  where $S^{-1/2}$ is the matrix square root of S^{-1}.
- One relevant special case arises when $S = \mathrm{diag}\left(\sigma_1^2, \ldots, \sigma_m^2\right)$, 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 $\xi_i = x_i / \sigma_i$, $i = 1, \ldots, m$.

Training the sparse ℓ_1-center classifier

Preliminary fact

Proposition 2 (Weighted ℓ_1 center)

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

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

Let $W(\zeta) \doteq \sum_{\{i:\, z_i \leq \zeta\}} w_i$, $\quad \bar{W} \doteq \sum_{i=1}^{p} w_i$, and $\bar{\zeta} \doteq \inf\{\zeta : \; W(\zeta) \geq \bar{W}/2\}$. Then, an optimal solution is given by

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

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 $\bar{\zeta}$.

Weighted median and dispersion

Notation

- 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 the weighted ℓ_1-centering problem, and we denote it by $\mathrm{med}_w(z)$.
- We define as the *weighted median dispersion* the optimal value $d_w(z)$ of weighted ℓ_1-centering problem.
- 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.

Training the sparse ℓ_1-center classifier

Result

Proposition 3

The optimal solution of the ℓ_1-centering problem is obtained as follows:

1. *Compute the plain class medians*

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

2. *Define a 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 $j \in \mathcal{J}^-$.*

3. *Compute the weighted median of all observations*

$$\mu \doteq \underset{w}{\mathrm{med}}(\{x_i^{(j)}\}_{j=1,\ldots,n}).$$

Training the sparse ℓ_1-center classifier

Result

Proposition 3 (Contd.)

4. *Compute the median dispersion vectors* d^+, d^- *according to*

$$\begin{aligned} d_i^+ &\doteq \tfrac{1}{n_+}\sum_{j\in\mathcal{J}^+} |x_i^{(j)} - \mu_i^+| \\ d_i^- &\doteq \tfrac{1}{n_-}\sum_{j\in\mathcal{J}^-} |x_i^{(j)} - \mu_i^-|. \end{aligned}$$

5. *Compute the weighted median dispersion vector* d *according to*

$$d_i \doteq \sum_{j=1}^{n} w_j |x_i^{(j)} - \mu_i| = \frac{1}{n_+}\sum_{j\in\mathcal{J}^+} |x_i^{(j)} - \mu_i| + \frac{1}{n_-}\sum_{j\in\mathcal{J}^-} |x_i^{(j)} - \mu_i|.$$

Training the sparse ℓ_1-center classifier

Result

Proposition 3 (Contd.)

6. *Compute the difference vector* $e \doteq (d^+ + d^-) - d$.
7. *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.*
8. *The optimal parameters θ^+, θ^- are given by*

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

Sparse ℓ_1-center classifier

Numerical complexity

- Computation of the medians in Proposition 3 can be performed with in $O(m)$ operations.
- Computation of the median dispersions requires $O(mn)$ operations.
- Finding the k smallest elements in vector e can be performed in $O(m \log k)$ operations.
- The whole procedure in Proposition 3 is thus performed in $O(mn) + O(m \log k)$ operations.

- Similar to the ℓ_2 case, also in the sparse ℓ_1 center classifier one need to do a full ordering of an m-vector only once in order to obtain all the sparse classifiers for any sparsity level k.

Numerical Tests

Sparse ℓ_2-center classifiers

Numerical experiments

- We compared the proposed sparse ℓ_2-center classifier with other feature selection methods for sentiment classification on text datasets.
- We considered three different datasets:
  - TwitterSentiment140 (TWTR) dataset
  - MPQA Opinion Corpus Dataset
  - Stanford Sentiment Treebank (SST).
- All datasets are labeled with binary labels indicating the polarity of the text.

Table: Text dataset sizes

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

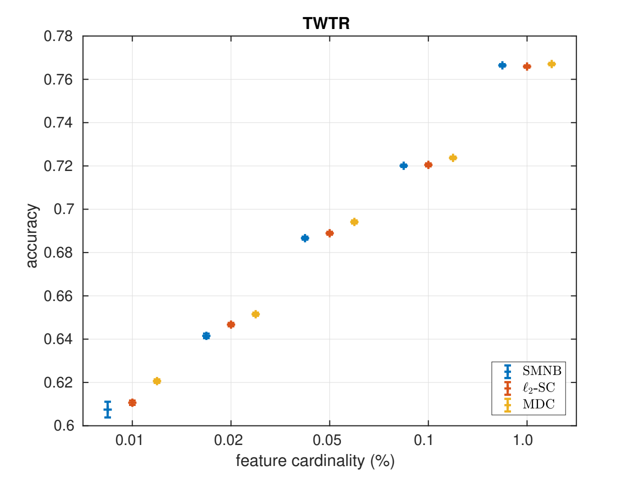
Sparse ℓ_2-center classifiers

Numerical experiments

- For each dataset, we performed a two-stage classification procedure.
- In the first stage, we apply a feature selection method in order to reduce the number of features. Then, in the second stage we train a classifier method employing only the selected features.
- We compared different feature selection methods: sparse ℓ_2-centers (ℓ_2-SC), Mahalanobis distance classifier (MDC), and sparse multinomial naive Bayes (SMNB).
- Other well-known feature selection methods, such as logistic regression, support vector machine, and LASSO, are not considered due to their high computational cost that makes them not feasible with large dataset.
- Using the selected features, we train a linear support vector machine classifier.

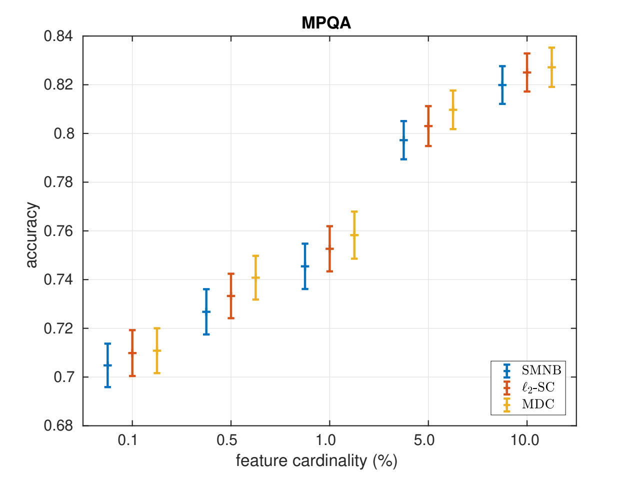
Sparse ℓ_2-center classifiers

Numerical experiments



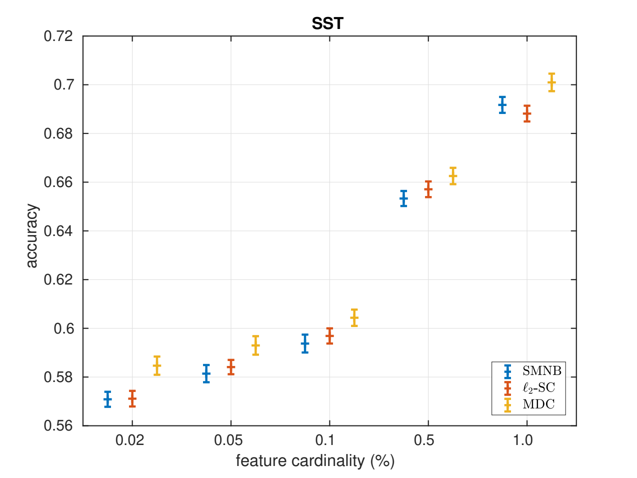
Sparse ℓ_2-center classifiers

Numerical experiments



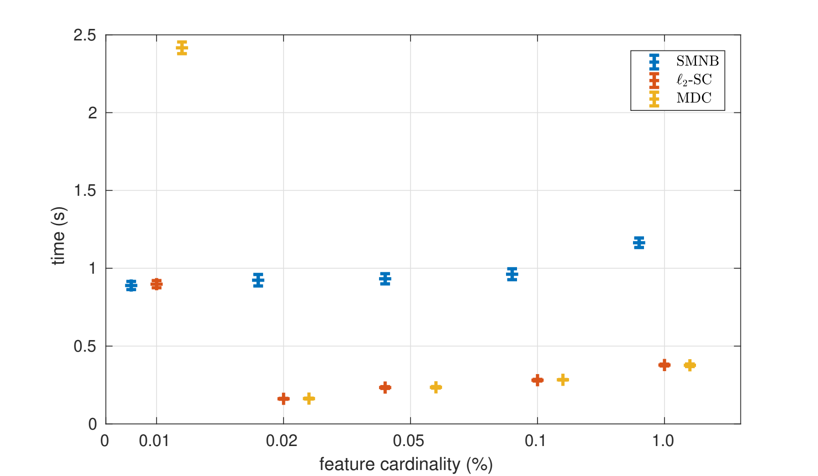
Sparse ℓ_2-center classifiers

Numerical experiments



Sparse ℓ_2-center classifiers

Runtimes



Sparse ℓ_1-center classifiers

Numerical experiments

- We compared the proposed sparse ℓ_1-center classifier with other feature selection methods for RNA gene expression classification.
- We considered the Leukemia dataset, and Breast Cancer dataset .

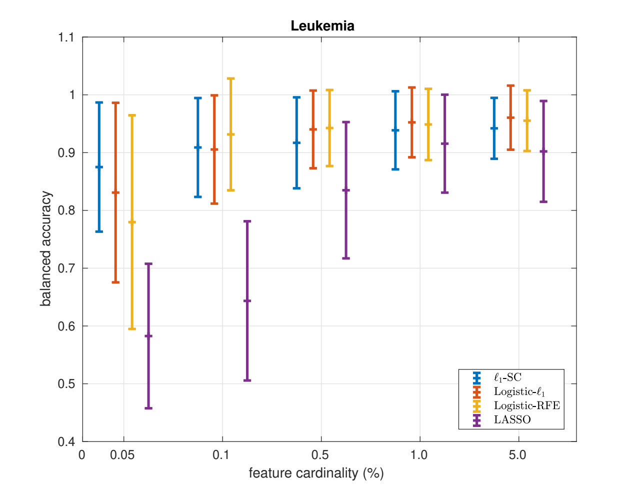
Table: RNA gene expression dataset sizes

	Leukemia	Breast Cancer
Number of features	7129	22215
Number of samples	72	118

- We compared four feature selection methods: sparse ℓ_1-centers (ℓ_1-SC), ℓ_1-regularized logistic regression, logistic regression with recursive feature elimination (RFE), and LASSO.

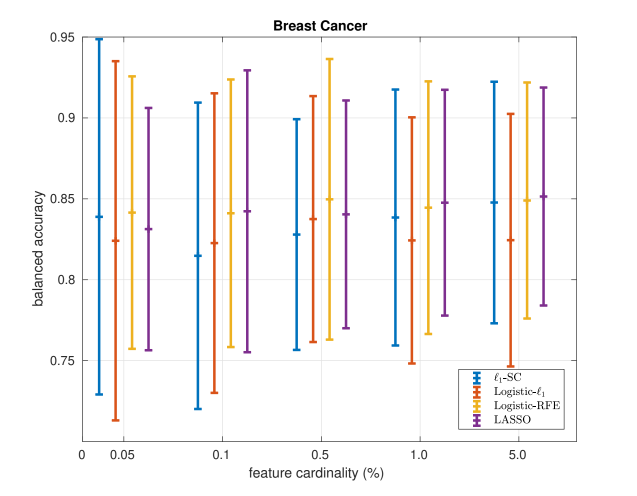
Sparse ℓ_1-center classifiers

Numerical experiments



Sparse ℓ_1-center classifiers

Numerical experiments



Sparse ℓ_1-center classifiers

Runtimes