Information Sciences 566 (2021) 178–194

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Adaptive discriminant analysis for semi-supervised feature selection

Weichan Zhong [a], Xiaojun Chen [a,*], Feiping Nie [b], Joshua Zhexue Huang [a]

[a] College of Computer Science and Software, Shenzhen University, Shenzhen, 518060, PR China
[b] School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University, Xi'an 710072, Shanxi, PR China

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 30 December 2020
Accepted 13 February 2021
Available online 27 February 2021

Keywords:
Feature selection
Semi-supervised feature selection
Discriminant analysis

ABSTRACT

As semi-supervised feature selection is becoming much more popular among researchers, many related methods have been proposed in recent years. However, many of these methods first compute a similarity matrix prior to feature selection, and the matrix is then fixed during the subsequent feature selection process. Clearly, the similarity matrix generated from the original dataset is susceptible to the noise features. In this paper, we propose a novel adaptive discriminant analysis for semi-supervised feature selection, namely, SADA. Instead of computing a similarity matrix first, SADA simultaneously learns an adaptive similarity matrix S and a projection matrix W with an iterative process. Moreover. we introduce the $\ell_{2,p}$ norm to control the sparsity of S by adjusting p. Experimental results show that S will become sparser with the decrease of p. The experimental results for synthetic datasets and nine benchmark datasets demonstrate the superiority of SADA, in comparison with 6 semi-supervised feature selection methods.

1. Introduction

Feature selection is very important for high-dimensional data analysis because it can remove irrelevant features with slight performance deterioration [1]. With the rapid increase of the data size, obtaining labeled data is often costly [2]. Therefore, to free us from the laborious and tedious data labeling work, only a small set of data samples are expected to be marked with ground truth. At the same time, it is desirable to exploit unlabeled samples during training to ensure the effectiveness of the learned models. The research topics related to this problem such as image annotations and categorizations have become hot spots in many machine learning fields [3,4]. Thus, it is desirable to develop feature selection methods that can exploit both labeled and unlabeled data. Therefore, the study of "semi-supervised feature selection" has gained increasing attention [5–7].

Due to the advantages of semi-supervised feature selection, related methods have sprung up in recent years. However, these methods have a shortcoming of measuring features with a ranking criterion without considering the models [8–11]. Ren et al. proposed a wrapper-type forward semi-supervised feature selection framework [12] that exploits labeled data and unlabeled data for the supervised sequential forward semi-supervised feature selection (SFFS). Xu et al. introduced a discriminative semi-supervised feature selection method based on the idea of manifold regularization, making use of

[*] Corresponding author.
E-mail addresses: zhongweichan@email.szu.edu.cn (W. Zhong), xjchen@szu.edu.cn (X. Chen), feipingnie@gmail.com (F. Nie), zx.huang@szu.edu.cn (J.Z. Huang).

https://doi.org/10.1016/j.ins.2021.02.035
0020-0255/© 2021 Elsevier Inc. All rights reserved.

classification margin and geometry of the probability distribution to select the features. However, because of its computational complexity of $O(n^{2.5}/\epsilon)$, where n is the number of objects and ϵ is a fairly small stopping criterion, their method is time-consuming [13]. To choose the "best so far" feature subset from the streaming features, Wu et al. proposed a novel feature selection method called online streaming feature selection (OSFS) [14]. However, domain knowledge is required in OSFS, and thus, Eskandari et al. chose to use rough sets (RS) to optimize the former model (OS-NRRSAR-SA) [15]. Recently, Zhou et al. further improved the new OS-NRRSAR-SA model with the adapted neighborhood rough set [16]. Chen et al. have also used the rough set to perform feature selection on imbalanced data [17]. Embedded semi-supervised methods are superior to other feature selection methods in many ways because feature selection is set as part of the model training process. Chen proposed a semi-supervised feature selection method RLSR [5] in which a rescaled linear square regression is added to extend the least-squares regression for feature selection. Yuan et al. improved RLSR by introducing an ϵ-dragging technique to enlarge the distances between different classes [18]. In addition to the methods mentioned above, other methods such as semi-supervised feature selection via spline regression [19], ensemble feature selection [20–22], and parallel feature selection [23,24] have also been developed. For most feature selection methods, a pair-wise similarity matrix is constructed from the original data and then set as fixed during the subsequent feature selection process. Researchers have designed several metrics to quantify the similarity between features based on some powerful tools like mutual information and graphs [25,26]. However, as pointed in [27], such a similarity matrix may lose the inner class structure on the *multimodal* data in which samples in some classes form several separate clusters, and often mislead the feature selection methods into recovering the wrong local structure since it is easily affected by noise features.

Sparsity commonly exists in real-world data, thus, sparse learning has become a key component in feature selection. Shi et al. consider the superiority of $l_{2,p}$-norm as well as its non-Lipschitz continuity and claim the effectiveness of $l_{2,1-2}$-norm. Moreover, they apply CCCP and ADMM to solve the non-convex problem [28]. Zhang et al. draw the conclusion that for $l_{2,p}$-norm, a smaller p leads to higher performance. They also discuss the situation where $p \rightarrow 0$ and proposed two algorithms to optimize the discrete feature selection problem [29].

Recently, Chen et al. have proposed a LAP framework for both labeled and unlabeled data [30]. Instead of computing a fixed similarity matrix prior to performing feature selection, LAP learns an adaptive similarity matrix **S** and a projection matrix **W** simultaneously with an iterative method. Based on this method, we extend it to semi-supervised feature selection tasks by proposing a semi-supervised adaptive discriminant analysis (SADA). As an extension to LAP, SADA can learn better a similarity matrix by weakening the effect of noisy features on the similarity computing and deal with *multimodal* data [27] by investigating local structure in the data. The main contributions of our work include:

1. We rewrite $||W^T(x_i - x_j)||_2$ as $||W^T(x_i - x_j)||_{2,p}^p$ by introducing the $\ell_{2,p}$ norm to control the sparsity of S by adjusting p, which can be used to adaptively preserve the locality. Experimental results show that S will become sparser with the decrease of p.
2. We take both labeled and unlabeled data into account to better preserve the locality in the semi-supervised scenario.
3. Comprehensive experiments on 9 benchmark datasets show the superior performance of the proposed approach in comparison with 6 semi-supervised feature selection methods.

The rest of this paper is organized as follows. Section 2 presents the notations, and Section 3 surveys the existing semi-supervised feature selection methods. The semi-supervised feature selection method, SADA, is proposed in Section 4. We present the experimental results and analysis in Section 5. The conclusions and directions for future work are provided in Section 6.

2. Notations and definitions

We now summarize the notations and the definition of the norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its i-th row is denoted as $\mathbf{m}^i$, and its j-th column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is defined as $||\mathbf{M}||_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m m_{ij}^2}$. The $\ell_{2,1}$-norm of matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is defined as $||\mathbf{M}||_{2,p} = \left(\sum_{i=1}^n \left(\sum_{j=1}^m m_{ij}^2\right)^{\frac{p}{2}}\right)^{\frac{1}{p}}$.

3. Semi-supervised feature selection

The early semi-supervised feature selection methods are filter-based which score the features with a ranking criterion regardless of the model [8–11]. For example, Zhao et al. proposed a semi-supervised feature selection algorithm named sSelect based on spectral analysis [8]. Consider a dataset $\mathbf{X} \in \mathbb{R}^{d\times n}$ consisting of two subsets: a set of l labeled objects $\mathbf{X}_L = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ that are associated with class labels $\mathbf{y}_L \in \mathbb{R}^l$ and a set of $u = n - l$ unlabeled objects $\mathbf{X}_U = (\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u})^T$ for which the labels $\mathbf{y}_U \in \mathbb{R}^u$ are unknown. In term of feature, $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_d\}$, where $\mathbf{f}_i$ denotes the i-th feature vectors in **X**. **Select** that computes a score s_j for the j-th feature as follows

$$s_j = \lambda \frac{\sum_{i,h=l+1}^{l+u} (g_i - g_h)^2 \times a_{ih}}{2\sum\limits_{i=l+1}^{l+u} g_i^2 \times \mathbf{d}_i} + (1-\lambda)(1 - NMI(\hat{g}, \mathbf{Y}_L)) \tag{1}$$

where λ is a parameter, $\mathbf{g} \in \mathbb{R}^{n\times 1}$ is the cluster indicator generated from the j-th feature vector $\mathbf{f}_j \in \mathbb{R}^{n\times 1}, \hat{g} \in \mathbb{R}^{l\times c}$ is the cluster labels obtained from $\mathbf{g}, a_{ij}$ is the similarity between the i-th feature and the j-th feature and $\mathbf{d}_i = \sum_{j=1}^{n} a_{ij}$. Through analysis we know that the computational complexity of sSelect is $O\left(dn^2\right)$.

Zhao et al. proposed a locality sensitive semi-supervised feature selection named **LSDF**. [9]. In this method, the importance score for the j-th feature is computed as

$$\mathbf{L}_j = \frac{\mathbf{f}_j^T \mathbf{L}_b \mathbf{f}_j}{\mathbf{f}_j^T \mathbf{L}_w \mathbf{f}_j} \tag{2}$$

where $\mathbf{f}_j \in \mathbb{R}^{n\times 1}$ is the j-th feature vector, $\mathbf{L}_w = \mathbf{D}_w - \mathbf{W}\,(\mathbf{D}_w = diag(\mathbf{W1})), \mathbf{L}_b = \mathbf{D}_b - \mathbf{B}\,(\mathbf{D}_b = diag(\mathbf{B1}))$ are graph Laplacians, $\mathbf{W}$ is a within-class affinity matrix and $\mathbf{B}$ is a between-class affinity matrix. LSDF has the same computational complexity of $O\left(dn^2\right)$ as sSelect.

Doquire et al. proposed a semi-supervised Laplacian score (**SSLS**) for semi-supervised feature selection [10]. In their method, they first construct an affinity matrix $\mathbf{S}^{sup}$ with the element s_{ij}^{sup} defining as

$$s_{ij}^{sup} = \begin{cases} e^{\frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{t}} & \text{if } \mathbf{x}_i \in \mathbf{N}_k(\mathbf{x}_j) \text{or} \mathbf{x}_j \in \mathbf{N}_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where t is a suitable positive constant and an affinity matrix $\mathbf{S}^{semi}$ in which s_{ij}^{semi} is defined as follows

$$s_{ij}^{semi} = \begin{cases} e^{\frac{-d_{ij}^2}{t}} & \text{if } \mathbf{x}_i \in \mathbf{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathbf{N}_k(i) \\ & \text{and} \mathbf{y}_i \text{ or } \mathbf{y}_j \text{is unknown} \\ C \times e^{\frac{-d_{ij}^2}{t}} & \text{if } \mathbf{x}_i \in \mathbf{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathbf{N}_k(i) \\ & \text{and } \mathbf{y}_i \text{ and } \mathbf{y}_j \text{are known} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where C is a positive constant and d_{ij}^2 is defined as

$$d_{ij}^2 = \begin{cases} (y_i - y_j)^2 & \text{if } \mathbf{y}_i \text{ and } \mathbf{y}_j \text{ are known} \\ \frac{\sum\limits_{l=1}^{n} (f_{li} - f_{lj})^2}{n} & \text{otherwise} \end{cases} \tag{5}$$

Then, SSLS measures the importance of the j-th feature according to

$$SSLS_j = \frac{\tilde{\mathbf{f}}_j^T \mathbf{L}^{\mathbf{semi}} \tilde{\mathbf{f}}_j}{\tilde{\mathbf{f}}_j^T \mathbf{D}^{\mathbf{semi}} \tilde{\mathbf{f}}_j} \times \frac{\tilde{\mathbf{f}}_j^T \mathbf{L}^{\text{sup}} \tilde{\mathbf{f}}_j}{\tilde{\mathbf{f}}_j^T \mathbf{D}^{\text{sup}} \tilde{\mathbf{f}}_j} \tag{6}$$

where $\tilde{\mathbf{f}}_j = \mathbf{f}_j - \frac{\mathbf{f}_j^T \mathbf{D}^{\mathbf{semi}} \mathbf{1}}{\mathbf{1}_j^T \mathbf{D}^{\mathbf{semi}} \mathbf{1}} \mathbf{1}, \mathbf{L}^{semi} = \mathbf{D}^{\mathbf{semi}} - \mathbf{S}^{semi}$.

It can be verified that SSLS also has the computational complexity of $O\left(dn^2\right)$.

Xu et al. proposed semi-supervised feature selection based on relevance and redundancy criteria (**RRPC**) [11]. According to the algorithm, we assumed that $\mathbf{F}_{k-1}$ have been obtained, which consist of $k-1$ selected features from $\mathbf{F}$. The k-th feature will be selected from feature subsets $\{\mathbf{F} - \mathbf{F}_{k-1}\}$ in the next step using the following equation

$$\mathbf{F}_k = \mathop{\arg\min}_{\mathbf{F}_j \in \mathbf{F} - \mathbf{F}_{k-1}} \left[\mathbf{P}(\mathbf{F}_j, \mathbf{y}_L) - \frac{1}{k-1} \sum_{\mathbf{F}_i \in\ \mathbf{F}_{k-1}} \mathbf{P}(\mathbf{F}_j, \mathbf{F}_i) \right] \tag{7}$$

where $\mathbf{P}(\mathbf{F}_j, \mathbf{Y}_L)$ is Pearson's correlation coefficient between two vectors $\mathbf{F}_j$ and $\mathbf{y}_L$. The computational complexity of RRPC is $O\left(nd^2\right)$.

Ren et al. have extended SFFS and proposed an iterative "wrapper-type" forward semi-supervised feature selection framework [12]. Within every single iteration, this method first chooses the labeled data to train a classifier, then the classifier will be used to predict the unlabeled samples. Afterward, some "labeled" data, which are originally unlabeled, will be added to the training data for the next iteration.

Embedded semi-supervised methods that include feature selection as part of the training process have also attracted increasing attention in the research community. With the concept of manifold regularization, Xu et al. have proposed FS-Manifold [13]. They construct an affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ at first and then use the objective function below to obtain the projection matrix $\mathbf{W}$

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\xi} \quad & \left(\frac{\|\mathbf{w}\|_2^2}{2} + C\sum_{i=1}^{l}\xi_i + \frac{\rho}{2}\mathbf{w}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{w} \right) \\ s.t. \quad & \mathbf{y}_i\left(\mathbf{w}^T\mathbf{x}_i - b\right) \geqslant 1 - \xi_i (1 \leqslant i \leqslant l) \\ & \xi_i \geqslant 0 (1 \leqslant i \leqslant l) \end{aligned} \tag{8}$$

where C and ρ are two parameters, and $\mathbf{L} = diag(\mathbf{A1}) - \mathbf{A}$ is the graph Laplacian. They proposed to use the level method to optimize the above objective function; however, this approach has high computational complexity bounded by $O(n^{2.5}/\epsilon)$, where ϵ is a small stopping criterion.

By improving the least square regression for feature selection with a rescaled linear square regression, Chen et al. proposed a new semi-supervised feature selection **RLSR** [5]. In order to make full use of both labeled and unlabeled data, **RLSR** introduces d scale factors θ to rank the d features of $\mathbf{X} \in \mathbb{R}^{d\times n}$, which contains labeled data $\mathbf{X}_L$ and unlabeled data $\mathbf{X}_U$ associated with class labels $\mathbf{Y}_L$ and $\mathbf{Y}_U$ respectively. The element $\theta_j > 0 (1 \leqslant j \leqslant d)$ of θ measures the importance of the j-th feature. A rescale matrix $\Theta \in \mathbb{R}^{d\times d}$ which is a diagonal matrix with its element defining as $\Theta_{jj} = \theta_j^{1/2}$ is used to rescales the regression coefficients $\mathbf{W} \in \mathbb{R}^{d\times c}$ to simultaneously learn θ and $\mathbf{Y}_U$. The objective function of **RLSR** is defined as

$$\begin{aligned} & \min \left(\|\mathbf{X}^T\Theta\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{W}\|_F^2 \right) \\ & s.t.\ \mathbf{W}, \mathbf{b}, \theta > 0, \mathbf{1}^T\theta = 1, \mathbf{Y}_U \geqslant 0, \mathbf{Y}_U\mathbf{1} = \mathbf{1} \end{aligned} \tag{9}$$

where $\mathbf{Y}_U$ are relaxed as continuous values in $[0, 1]$. $\mathbf{b} \in \mathbb{R}^c$ is the bias and $\gamma > 0$ is the regularized parameter to control the trade-off between the bias and variance of the estimate. Yuan et al. improved RLSR by introducing a ϵ-dragging technique to enlarge the distances between different classes [18].

4. Proposed method

In this section, we propose the new feature selection method conducted in a semi-supervised manner.

4.1. Optimization model

In semi-supervised learning, a dataset $\mathbf{X} \in \mathbb{R}^{d\times n}$ with c classes consists of two subsets: a set of l labeled objects $\mathbf{X}_L = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ that are associated with class labels $\mathbf{Y}_L = \{\mathbf{y}_1, \ldots, \mathbf{y}_l\}^T \in \mathbb{R}^{l\times c}$ and a set of $u = n - l$ unlabeled objects $\mathbf{X}_U = (\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u})^T$ for which labels $\mathbf{Y}_U = \{\mathbf{y}_{l+1}, \ldots, \mathbf{y}_{l+u}\}^T \in \mathbb{R}^{u\times c}$ are unknown. We seek to find a linear combination of the original features to obtain the best approximation of the low-dimensional manifold. Let $\mathbf{W} \in R^{d\times m}$ be a projection matrix with m denoting the projection dimension. The objective function of **LAP** is defined as [30]

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2 + \gamma\|\mathbf{W}\|_{2,1} \tag{10}$$

It has been proved that $\mathbf{W}$ can be solved by the following equation

$$\min_{\mathbf{W},\mathbf{W}^T\mathbf{W}=\mathbf{I}} \left[Tr\left(\mathbf{W}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\mathbf{W}\right) + \gamma Tr\left(\mathbf{W}^T\mathbf{Q}\mathbf{W}\right) \right] \tag{11}$$

where $\mathbf{L}_S = \mathbf{D}_s - \mathbf{S}$ is the Laplacian matrix of $\mathbf{S}$ and $\mathbf{D}_s \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the i-th diagonal element as $\sum_{j=1}^{n} s_{ij}$. And the element of $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is defined as

$$q_{ll} = \frac{1}{2\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}} \tag{12}$$

and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is defined as

$$s_{ij} = \frac{1}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon}} \tag{13}$$

To preserve the locality in feature selection task, a commonly-used approach is to model the k nearest neighbors of each object. For the semi-supervised task, given an object, we can simultaneously consider its k nearest labeled neighbors and k nearest unlabeled neighbors. However, if we simply select the nearest unlabeled neighbors according to the similarities computed from the original data, we may get the wrong result because the similarities are easily affected by noise features. In this paper, we propose to learn a sparse S in order to adaptively preserve the locality from the following problem

$$\min_{\mathbf{W},\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i=1}^{n} \sum_{j \in \mathcal{M}_i^L \bigcup \mathcal{M}_{\rangle}^{\mathcal{U}}} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_{2,p}^p + \gamma\|\mathbf{W}\|_{2,1} \tag{14}$$

where $\mathcal{M}_i^L$ consists of k nearest neighbors of $\mathbf{x}_i$ in $\mathbf{X}_L$ and $\mathcal{M}_i^U$ consists of k nearest neighbors of $\mathbf{x}_i$ in $\mathbf{X}_U$. Both of them are obtained from **YL** and **YU** respectively and the values remain unchanged during the following steps. Specifically, if $\mathbf{x}_i$ is labeled, $\mathcal{M}_i^L$ consists of $\min\{k, nc_i\}$ nearest neighbors that are in the same class as $\mathbf{x}_i$, and nc_i is the number of objects in the class to which $\mathbf{x}_i$ belongs.

We further introduce a parameter p to extend the factor $||W^T(x_i - x_j)||_2^2$ in problem(10) to $||W^T(x_i - x_j)||_{2,p}^p$ which we will discuss in the next subsection. Theoretically speaking, $\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_{2,p}^p$ and $\|\mathbf{W}\|_{2,1}$ can be zero; however, such zero values will make Eq. (15) non-differentiable. This condition can be avoided by introducing a sufficiently small constant ϵ, e.g., 10^{-10}, to obtain a nonzero denominator and rewrite $\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_{2,p}^p$ as $\left(\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon\right)^{\frac{p}{2}}$, and $\|\mathbf{W}\|_{2,1}$ as $\sum_{l=1}^{d}\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}$. Then we can rewrite Eq. (14) as

$$\begin{aligned}\min_{\mathbf{W},\mathbf{W}^T\mathbf{W}=\mathbf{I}} \quad & \sum_{i=1}^{n} \sum_{j \in \mathcal{M}_i^L \bigcup \mathcal{M}_{\rangle}^{\mathcal{U}}} \left(\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon\right)^{\frac{p}{2}} \\ & + \gamma\sum_{l=1}^{d}\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}\end{aligned} \tag{15}$$

The Lagrangian functional $\mathcal{L}(\mathbf{W}, \Lambda)$ of problem (15) is defined as

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \Lambda) = & \sum_{i=1}^{n} \sum_{j \in \mathcal{M}_i^L \bigcup \mathcal{M}_{\rangle}^{\mathcal{U}}} \left(\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon\right)^{\frac{p}{2}} \\ & + \gamma\sum_{l=1}^{d}\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon} + Tr\left(\Lambda\left(\mathbf{W}^T\mathbf{W} - \mathbf{I}\right)\right)\end{aligned} \tag{16}$$

where $\Lambda \in \mathbb{R}^{m \times m}$ is the Lagrangian multiplier. We can obtain Eq. (17) after taking the derivative of $\mathcal{L}(\mathbf{W}, \Lambda)$ w.r.t $\mathbf{W}$ and setting the derivative to zero.

$$\begin{aligned}\tfrac{\partial\mathcal{L}(\mathbf{W},\Lambda)}{\partial\mathbf{W}} = & \sum_{i,j=1}^{n} s_{ij}\tfrac{\partial\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2}{\partial\mathbf{W}} \\ & + 2\gamma\mathbf{QW} + \mathbf{W}\Lambda = 0\end{aligned} \tag{17}$$

where $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is a diagonal matrix and its l-th element is defined as Eq. (12) and $\mathbf{S} \in \mathbb{R}^{n \times n}$ is defined as

$$s_{ij} = \begin{cases} \frac{p}{2\left(\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon\right)^{1-\frac{p}{2}}} & \text{if } j \in \mathcal{M}_i^L \bigcup \mathcal{M}_i^U \\ 0 & \textit{otherwise} \end{cases} \tag{18}$$

The Eq. (18) shows that parameter p is used to solve s_{ij}. Furthermore, as p increases, the value of s_{ij} increases as well. Due to the dependency of $\mathbf{W}$ while computing $\mathbf{Q}$ and $\mathbf{S}$, Eq. (17) is difficult to solve. Thus, in this paper, we propose a novel iterative method to solve $\mathbf{W}$. In the new method, we first fix $\mathbf{Q}$ and $\mathbf{S}$ to obtain $\mathbf{W}$ by solving the following problem

$$\min_{\mathbf{W},\mathbf{W}^T\mathbf{W}=\mathbf{I}} Tr\left[\mathbf{W}^T\left(\mathbf{X}\mathbf{L}_S\mathbf{X}^T + \gamma\mathbf{Q}\right)\mathbf{W})\right] \tag{19}$$

According to the Rayleigh–Ritz theorem [31], it can be verified that the optimal solution to Problem (19) consists of the m eigenvectors of $\mathbf{X}\mathbf{L}_S\mathbf{X}^T + \gamma\mathbf{Q}$ corresponding to its m smallest eigenvalues.

Then, with the newly learned $\mathbf{W}$, we update $\mathbf{Q}$ and $\mathbf{S}$ according to Eqs. (12) and (18). The description of the above algorithm, named as the semi-supervised adaptive discriminant analysis (SADA), is shown in Algorithm 1. In the new algorithm,

$\mathbf{W}$ is solved by performing eigendecomposition of $\boldsymbol{X}\boldsymbol{L}_S\boldsymbol{X}^T + \gamma\boldsymbol{Q}$, and then $\mathbf{Q}$ and $\mathbf{S}$ are directly computed. The three variables $\mathbf{W}, \mathbf{Q}$ and $\mathbf{S}$ are automatically updated until the algorithm converges. Finally, θ are computed from the learned $\mathbf{W}$ and the r most important features are selected according to θ.

Algorithm 1. SADA: the algorithm to solve problem (14)

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, class labels $\mathbf{Y} \in \mathbb{R}^{n\times c}$, regularization parameter $\gamma$, parameter $p \in (0,2)$ and number of selected features $r$.
Initialize $\mathbf{Q}$ as the identity matrix.
Set $t = 0$.
repeat
    Update $\mathbf{W}_{t+1}$ as the $m$ eigenvectors of $\left(\mathbf{X}\mathbf{L}_{S_t}\mathbf{X}^T + \gamma\mathbf{Q}_t\right)$ corresponding to its $m$ smallest eigenvalues.
    Update the diagonal matrix $\mathbf{Q}_{t+1}$, where the $l$-th diagonal element is $\frac{1}{2\sqrt{\mathbf{w}^l_{t+1}\left(\mathbf{w}^l_{t+1}\right)^T+\epsilon}}$.
    Update the matrix $\mathbf{S}_{t+1}$, where $s_{ij}$ is defined in Eq. (18).
    Update the matrix $\mathbf{L}_S = \mathbf{D}_S - S$, where $\mathbf{D}_S$ is defined as $\sum_{j=1}^n s_{ij}$.
    Set $t = t + 1$.
until Converges
Compute $\theta = \frac{\|\mathbf{w}^i\|_2}{\sum_{h=1}^d \|\mathbf{w}^h\|_2}$.
Sort $\theta$ in descending order, and select top $r$-ranked features as the ultimate result.
```

In this algorithm, we need $O\left(d^2(n+m)\right)$ to update $\mathbf{W}, O(dm)$ to update $\mathbf{Q}, O(dnm)$ to update $\mathbf{S}$ and $O(n)$ to update $\mathbf{L_s}$, where n is the number of samples, d is the number of features and m is the number of projection dimension. Thus, the time complexity of Algorithm 1 is $O\left(t\left(d^2(n+m)\right)\right)$, where t is the number of iterations. Moreover, the space complexity is $O\left(d^2 + dn + n^2\right)$.

4.2. Analysis of relationship between S and p

To analyze the relationship between the similarity matrix S and the parameter p, we first define $f(g_{ij})$ as

$$f(g_{ij}) = \frac{p}{2}(g_{ij})^{\frac{p}{2}-1} \tag{20}$$

where $\mathbf{g}_{ij}$ is the projected distance between the $\mathbf{x}_i$ and $\mathbf{x}_j$ defined as

$$g_{ij} = \|W^T(x_i - x_j)\|_2^2 + \epsilon \tag{21}$$

Since $p \in (0,2)$, we have $p/2 - 1 \in (-1,0)$. We plot the function $f(g_{ij})$ in Fig. 1. An examination of Fig. 1 reveals that a smaller p forces $f(g_{ij})$ to be close to zero faster with the increase of g_{ij}. Therefore, we can set a smaller p to obtain sparser similarity matrix S in order to adaptively preserve the locality.

4.3. Convergence proof

To prove the convergence of Algorithm 1, we need the following lemma:

Lemma 1. *For any two positive real vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{n\times 1}$ *and* $0 < p \leqslant 2$, *the following inequality holds*

$$\sum_{i=1}^n \left[a_i^{\frac{p}{2}} - \frac{p}{2}\frac{a_i}{b_i^{\frac{2-p}{2}}}\right] \leqslant \sum_{i=1}^n \left[b_i^{\frac{p}{2}} - \frac{p}{2}\frac{b_i}{b_i^{\frac{2-p}{2}}}\right] \tag{22}$$

Proof. Since $\ln x (x > 0)$ is a concave function, we have the following inequality according to the definition of a concave function

$$\ln((1-\theta)d + \theta c) \geqslant (1-\theta)\ln(d) + \theta\ln(c). \tag{23}$$

for any $\theta \in [0,1]$ and two positive numbers c and d.

Problem (23) can be rewritten as

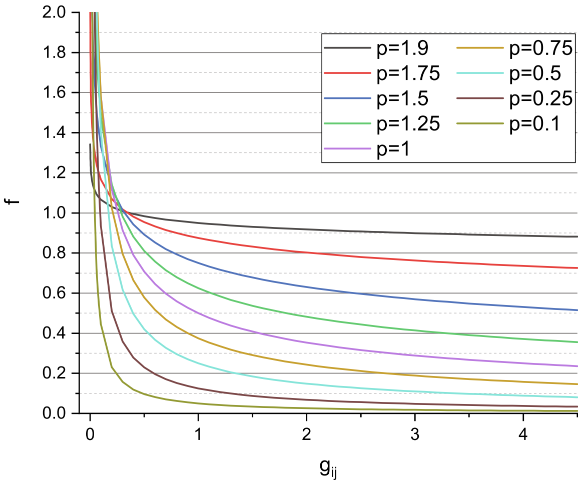


Fig. 1. Relationship between f and g_{ij} with different p.

$$c^{\theta}d^{1-\theta} - \theta c \leqslant d - \theta d \tag{24}$$

which is equivalent to

$$c^{\theta} - \theta\frac{c}{d^{1-\theta}} \leqslant d^{\theta} - \theta\frac{d}{d^{1-\theta}} \tag{25}$$

Substituting $\theta = \frac{p}{2}, c = a_i$ and $d = b_i$ into Eq. (25) obtains

$$a_i^{\frac{p}{2}} - \frac{p}{2}\frac{a_i}{b_i^{\frac{2-p}{2}}} \leqslant b_i^{\frac{p}{2}} - \frac{p}{2}\frac{b_i}{b_i^{\frac{2-p}{2}}}. \tag{26}$$

Summing over $i = 1$ to n of Eq. (26) obtains

$$\sum_{i=1}^{n}\left[a_i^{\frac{p}{2}} - \frac{p}{2}\frac{a_i}{b_i^{\frac{2-p}{2}}}\right] \leqslant \sum_{i=1}^{n}\left[b_i^{\frac{p}{2}} - \frac{p}{2}\frac{b}{b^{\frac{2-p}{2}}}\right]. \tag{27}$$

completing the proof. □

Theorem 1. *The iteration process of Algorithm 1 will monotonically decreases the objective function of problem (14) in each iteration.*

Proof. Suppose the updated **W** by solving problem (19) is $\widetilde{\mathbf{W}}$. It is easy to see that

$$\begin{aligned} & Tr\left(\widetilde{\mathbf{W}}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\widetilde{\mathbf{W}}\right) + \gamma Tr\left(\widetilde{\mathbf{W}}^T\mathbf{Q}\widetilde{\mathbf{W}}\right) \\ \leqslant\ & Tr\left(\mathbf{W}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\mathbf{W}\right) + \gamma Tr\left(\mathbf{W}^T\mathbf{Q}\mathbf{W}\right) \end{aligned} \tag{28}$$

Since $\mathcal{M}_i^L$ and $\mathcal{M}_i^U$ remain unchanged during the optimization process, we denote $\mathcal{M}_i^L \bigcup \mathcal{M}_i^U$ as $\mathcal{M}_i$ for simplicity. We add $\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\frac{\epsilon}{2\left(\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}\right)}$ and $\gamma\sum_{l=1}^{d}\frac{\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}$ to both sides of Eq. (28), and substitute the definition of **Q** in Eq. (12). Then, Eq. (28) can be rewritten as

$$\begin{aligned} & \frac{p}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\frac{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{1-\frac{p}{2}}} + \gamma\sum_{l=1}^{d}\frac{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}} \\ \leqslant\ & \frac{p}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\frac{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{1-\frac{p}{2}}} + \gamma\sum_{l=1}^{d}\frac{\|\mathbf{w}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}} \end{aligned} \tag{29}$$

According to Lemma 1, we have

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\left(\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{\frac{p}{2}}\\&-\frac{p}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\frac{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{1-\frac{p}{2}}}\\\leqslant\;&\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{\frac{p}{2}}\\&-\frac{p}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\frac{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{1-\frac{p}{2}}}\end{aligned}\tag{30}$$

and

$$\begin{aligned}&\gamma\sum_{l=1}^{d}\sqrt{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}-\gamma\sum_{l=1}^{d}\frac{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}\\\leqslant\;&\gamma\sum_{l=1}^{d}\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}-\gamma\sum_{l=1}^{d}\frac{\|\mathbf{w}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}\end{aligned}\tag{31}$$

Summing over (29)–(31), we arrive at

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\left(\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{\frac{p}{2}}+\gamma\sum_{l=1}^{d}\sqrt{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}\\\leqslant\;&\sum_{i=1}^{n}\sum_{j\in\mathcal{M}_i}\left(\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon\right)^{\frac{p}{2}}+\gamma\sum_{l=1}^{d}\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}\end{aligned}\tag{32}$$

Therefore, the iteration process of Algorithm 1 will monotonically decreases the objective function of problem (14) in each iteration. □

Thus, the alternating optimization process will converge as the number of iterations goes to infinity. Let $\hat{\mathbf{W}}$ be the converged solution. It can be verified that the derivative of Eq. (19) with respect to $\mathbf{W}$ is exactly Eq. (16), so the converged $\hat{\mathbf{W}}$ will satisfy Eq. (17), which is the KKT condition of Problem (14). Therefore, Algorithm 1 can obtain a solution that is close enough to a local solution of Problem (14).

5. Experimental results and analysis

5.1. Experiments on synthetic datasets

We generated a synthetic dataset D_1 to test the projection ability of the proposed method for feature selection. D_1 consists of 12 dimensions, where the data in the first two dimensions are distributed in three Gaussian shapes, while the data in the other dimensions are uniformly distributed noise features. Fig. 2a shows the dataset in the first two dimensions in which two small Gaussian clusters are buried in one class. In this experiment, our goal is to find a good projection direction that can be used for feature selection. We compared SADA with five other methods including sSelect [8], LSDF [9], PRPC [11], RLSR [5] and DSFFS [18]. In this experiment, the projection dimension was set as 1, and the nearest neighborhoods k was set as 5. The regularization parameters in RLSR, DSFFS, and SADA were set as 1 for fair comparisons. The neighborhood parameters were set to 10 to SADA and 5 to LSDF for all datasets. For SADA, we set $p = 0.6$. The projection direction results are displayed in Fig. 2b, demonstrating that if we consider separating only the red class from the blue class, the direction of the projection revealed by LSDF is good. However, SADA achieves the best direction of projection for separating the two small classes contained within the red class.

5.2. Experiments on benchmark datasets

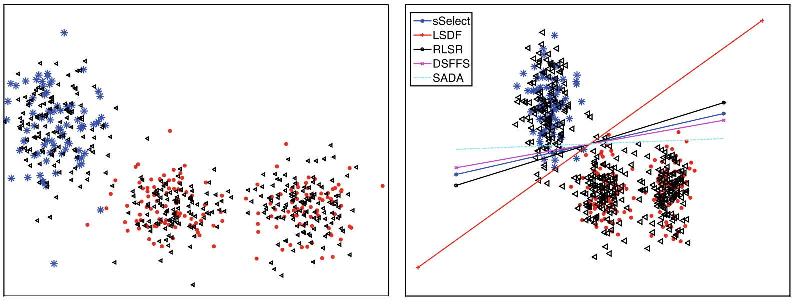
5.2.1. Benchmark datasets

We selected 9 benchmark datasets from the UCI Machine Learning Repository.[1] *Leukemia*3 is collected from St Jude Research website.[2] Table 1 summarizes the characteristics of these 9 datasets.

To validate the effectiveness of SADA, we compared it with 6 state-of-the-art semi-supervised feature selection methods including sSelect [8], LSDF [9], PRPC [11], RLSR [5] and DSFFS [18]. We also consider ULAP as a comparative method because

[1] http://archive.ics.uci.edu/ml/index.php

[2] https://www.stjuderesearch.org/site/data/ALL1



(a) First two dimensions of D_1.

(b) Projection directions in the first two dimensions.

Fig. 2. Projection direction results of five methods on D_1. In each figure, the blue points and red points indicate two different classes, while the black points indicate unlabeled objects.

SADA is proposed to improve ULAP with the help of some labels [30]. We set the parameters of all methods in the same strategy to make the experiments fair, i.e., $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. The neighborhood parameters in LSDF and SADA were set to 10 for all of the datasets. For SADA, the parameter p was set to 10 values from 0.1 to 1.9, and the projection dimensions for different datasets are given in Table 1.

- **Colon:** Gene dataset that contains colon cancer data collected from 62 samples for 2000 genes.
- **Srbct:** Gene dataset that contains the small round blue cell tumor (srbct) cancer data collected from 63 samples for 2308 genes. It contains four clusters.
- **Breast2:** Gene dataset that contains breast cancer data collected from 77 samples for 4869 genes.
- **Prostate:** Gene dataset that contains prostate cancer data collected from 102 samples for 6033 genes.
- **Glass:** Glass dataset that was generated by a criminal reconnaissance that identifies a type of glass. The sixth cluster is marked as outliers.
- **BinAlpha:** Image dataset that contains binary data of handwritten images of 26 letters and ten (0–9) digits (36 in total)
- **Segment:** Image dataset that contains sample data randomly extracted from a database of 7 outdoor images. The image is segmented, and the nineteen attributes of each pixel are counted.
- **Isolet5:** Voice dataset that contains the voice data from 150 volunteers divided into 5 groups, reading the alphabet twice (26 letters in total, corresponding to 26 clusters in the dataset). Isole5 is the dataset of the fifth group.

Table 1
Characteristics of 9 benchmark datasets, where n is the number of samples, d is the number of features, c is the number of classes, m is the number of projection dimensions, and k is the number of selected features.

Name	n	d	c	m	k	Type
Colon	62	2000	2	[50 100 200]	[20,40,. . .,200]	Gene
Srbct	63	2308	2	[50 100 200]	[20,40,. . .,200]	Gene
Breast2	77	4869	2	[50 100 200]	[20,40,. . .,200]	Gene
Postrate	102	6033	5	[50 100 200]	[20,40,. . .,200]	Gene
Leukemia3	248	12625	6	[50 100 200]	[20,40,. . .,200]	Gene
Glass	214	9	6	[2–9]	[2–9]	Physical
BinAlpha	1404	320	36	[40,80,. . .,200]	[20,40,. . .,200]	Image
Isolet5	7797	617	26	[40,80,. . .,200]	[20,40,. . .,200]	Voice
Segment	2310	19	7	[2–19]	[2–19]	Other

Table 2
Average accuracies of 7 feature selection methods on 9 benchmark datasets (the best two results on each dataset are highlighted in bold).

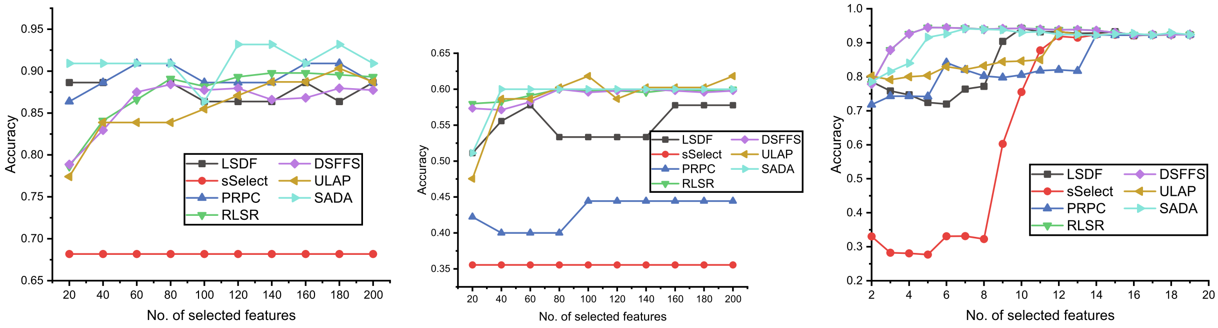
Name	LSDF	sSelect	PRPC	RLSR	DSFFS	ULAP	SADA
Binalpha	0.411 ± 0.018	0.204 ± 0.001	0.316 ± 0.000	**0.457** ± 0.035	**0.455** ± 0.029	0.403 ± 0.015	0.421 ± 0.003
Colon	0.877 ± 0.012	0.682 ± 0.000	**0.893** ± 0.023	0.841 ± 0.027	0.841 ± 0.053	0.887 ± 0.017	**0.923** ± 0.019
Segment	0.859 ± 0.089	0.654 ± 0.295	0.823 ± 0.060	**0.923** ± 0.032	**0.923** ± 0.046	0.873 ± 0.054	0.910 ± 0.044
Srbct	0.551 ± 0.025	0.356 ± 0.000	0.429 ± 0.037	0.591 ± 0.004	**0.593** ± 0.018	0.588 ± 0.015	**0.598** ± 0.027
Glass	**0.502** ± 0.023	0.434 ± 0.082	0.479 ± 0.019	0.492 ± 0.020	0.492 ± 0.027	0.486 ± 0.023	**0.503** ± 0.023
Isolet5	0.775 ± 0.023	0.471 ± 0.008	0.692 ± 0.000	**0.896** ± 0.046	**0.897** ± 0.028	0.706 ± 0.003	0.708 ± 0.034
Breast2	0.598 ± 0.000	0.564 ± 0.008	0.587 ± 0.001	0.591 ± 0.006	0.589 ± 0.005	**0.622** ± 0.011	**0.618** ± 0.004
Prostate	**0.853** ± 0.000	0.635 ± 0.008	0.842 ± 0.001	0.817 ± 0.006	0.827 ± 0.005	0.803 ± 0.001	**0.843** ± 0.020
Leukemia3	0.299 ± 0.011	0.301 ± 0.024	0.311 ± 0.010	0.317 ± 0.060	**0.318** ± 0.057	0.311 ± 0.000	**0.328** ± 0.028

- **Leukemia3:** Gene dataset that contains data collected from 248 leukemia patients, which can be divided into 6 categories: BCR, BCR_2, Hyperdip50, MLL, T and TEL.

5.2.2. Results and analysis

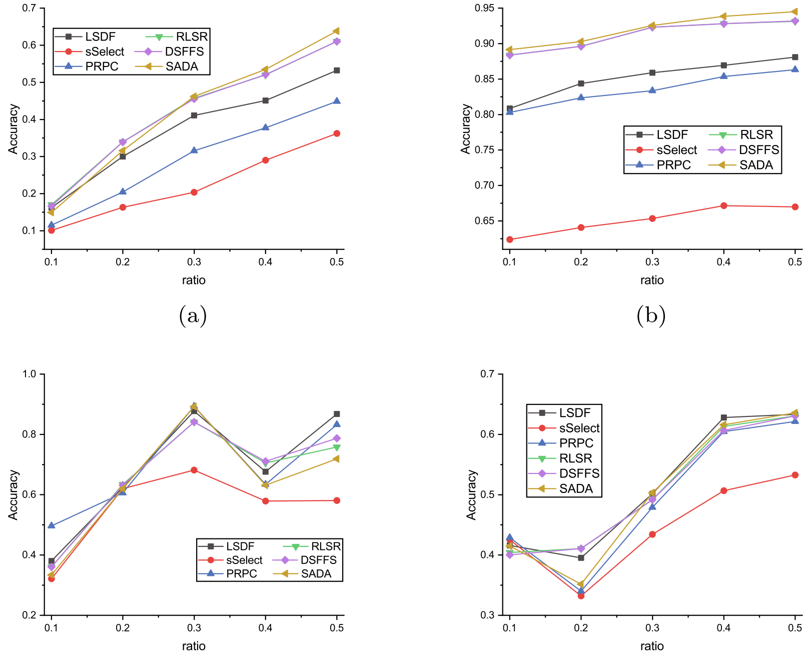
The average accuracies of 6 methods on 9 datasets are reported in Table 2, in which we used 30% data as the labeled data and 70% data as the unlabeled data and test data. Overall, our proposed SADA method outperformed other methods on most datasets, in particular, on the *Colon* and *Breast2* datasets. Specifically, SADA achieves a greater than 3% average improvement on the *Colon* dataset compared to the second-best method, PRPC. We can also observe a similar result on *Breast2*. SADA also achieved good performance on the rest of the datasets on average. This indicates that the learned implicit adaptive local structure learning indeed improves the feature selection performance.

A closer examination of the data presented in Table 2 shows that SADA achieves better performance than most of the other methods on the *Colon*, *Segment*, *Srbct* and *glass* datasets. Specifically, the accuracy of SADA on *Colon* is 5% higher than those of the other methods on average, and the accuracy on *Srbct* is 11% higher than those of the other methods on average. Additionally, we observed that SADA shows excellent performance on most gene datasets, including the *Colon*, *Srbct*, *Breast2* and *Prostrate* gene datasets. However, SADA performs badly on some datasets which contain a large number of classes, e.g., the *Isolet5* and *Binalpha* datasets. This may be because the new method focuses on preserving local structure but ignores the



(a) (b) (c)

Fig. 3. Comparison of accuracy results obtained by six methods with different numbers of selected features. (a) Results on the Colon dataset. (b) Results on the Srbct dataset. (c) Results on the Segment dataset.



(a) (b)

(c) (d)

Fig. 4. Accuracy versus ratio. (a) Results on the Binalpha dataset. (b) Results on the Segment dataset. (c) Results on the Colon dataset. (d) Results on the Glass dataset.

inter-class structure which is important for separating multiple classes. We also notice that our method performs well on high-dimensional datasets, e.g., the *Colon* and *Srbct* datasets. However, our method does not outperform all other methods on large datasets, e.g., the *Segment* and *Isolet5* datasets. This indicates that our method is better at dealing with high-dimensional data than dealing with large data.

Moreover, we set ULAP as a comparative method, which is an unsupervised model designed based on LAP. We can see from Table 2 that SADA achieves better performance than ULAP on most datasets, which means that semi-supervised learning helps to improve the performance of unsupervised feature selection.

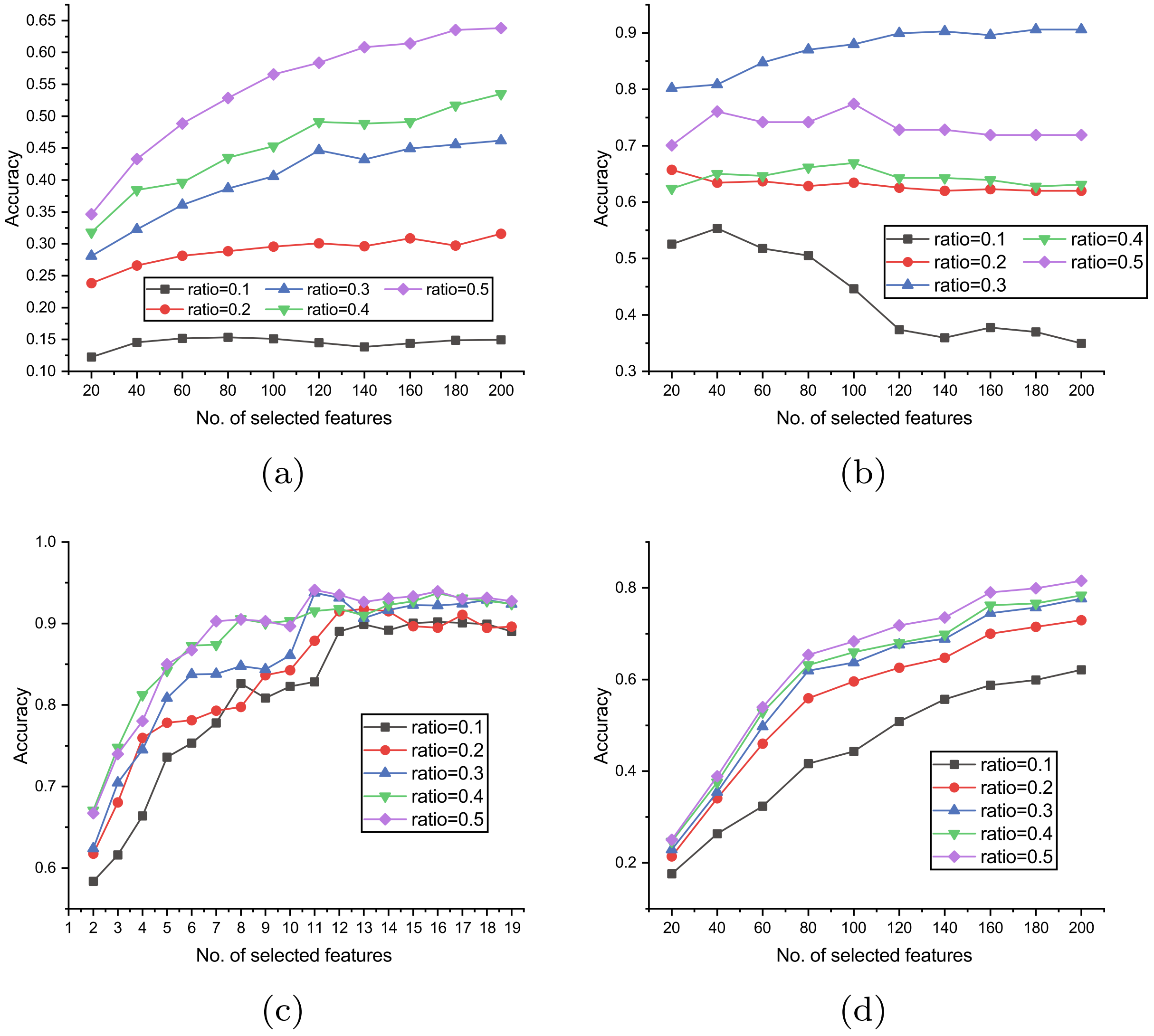
The performances of 7 methods on 3 typical datasets versus the number of selected features r are shown in Fig. 3, which shows that the number of selected features does not affect the performance of most methods too much when the number of selected features is not too small. We also notice that SADA outperforms all other methods with most r on these datasets.

5.2.3. Parameter analysis

In this section, we investigate the parameters of our proposed method.

A. Performance versus the ratio of labeled data (ratio).

We first set the ratio of labeled data to $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ to show the average accuracy of 6 methods versus *ratio* in Fig. 4. We find that for almost all datasets, the methods show similar trends. On the *Colon* dataset, all methods reach their



(a) (b)

(c) (d)

Fig. 5. Accuracy versus the numbers of selected features by SADA. (a) Results on the Binalpha dataset. (b) Results on the Colon dataset. (c) Results on the Segment dataset. (d) Results on the Isolet5 dataset.

best results when the ratio is 0.3, suddenly fall at 0.4, and finally raise at 0.5. SimilarlLAP y, on the *Glass* dataset, the lowest result was obtained at *ratio* $= 0.2$. We can observe increasing trends on the *Binalpha* and *Segment* datasets.

In the second experiment, we show the performance versus both ratio and the number of selected features r in Fig. 5. When *ratio* is small, e.g., *ratio* $= 0.1$, we can observe that the performance of SADA does not change too much with the increase of r on the Binalpha dataset, even decreases with the increase of r on the Colon dataset. With the increase of *ratio*, the performance of SADA increases on most datasets. Intuitively, the increase of *ratio* will improve the performance of SADA, which is verified on most datasets. However, on the *Colon* dataset, we observe that the best performances of SADA are obtained with *ratio* $= 0.3$, indicating that the increase of labeled data may cause performance decline on some datasets.

B. Impact of parameter p

I. Projection results versus p

We set $p = \{0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2\}$ to draw the projection directions by SADA in the first two dimensions in Fig. 6,7, indicating that SADA produces the best result with $p = 1.25$, and the worst result with $p = 0.1$. It can be observed that SADA produces good results with $p = \{0.75, 1, 1.25\}$, and bad results with too small or too large p. In real applications, we should carefully chose proper p for better results.

II. S versus p

We selected the *Colon* dataset to show the learned implicit similarity matrix $\mathbf{S}$ versus p, in which the similarity matrix $\mathbf{S}$ is normalized as $D^{-1/2}SD^{-1/2}$ where $\mathbf{D}$ is the degree matrix of $\mathbf{S}$. From this figure, we can observe that S will be sparser and the local structure will be better preserved with the decrease of p, which is consistent with the analysis in Section 4.2.

III. Performance versus p

In this experiment, we selected the *Colon* and *Srbct* datasets to show the accuracy with different numbers of the selected features versus p in Fig. 8, indicating that p affects the performance too much, especially when we only select a small subset of features.

C. Performance versus the regularization parameter γ

We selected the *Srbct* and *Colon* datasets to investigate the performance versus γ and the number of selected features r in Fig. 9. A careful examination of the two results in Fig. 9 shows that γ affects the performance too much when $r \leqslant 80$ and does not affect the performance too much when $r > 80$. Intuitively, the performance usually increases with an increase of r. However, on the *Colon* dataset, the performance does not change too much with the increase of r, when r is small.

D. Convergence Analysis

We have proved the convergence of the proposed algorithm in Section 4. In this section, we show the convergence curve of Algorithm 1 on real-life dataset in Fig. 10. It can be observed from this figure that our proposed method is stable and con-

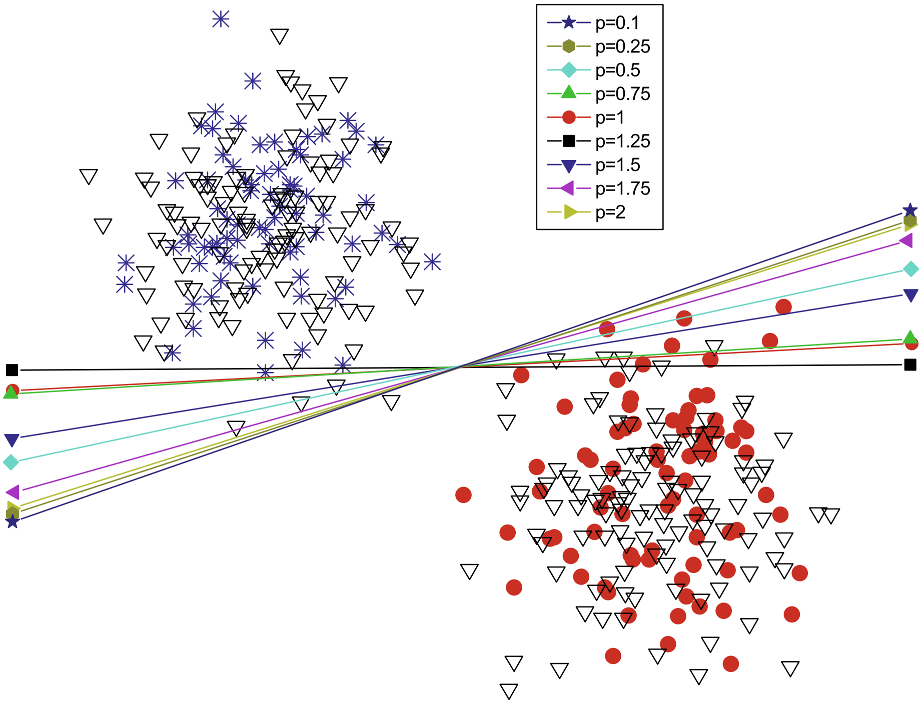


Fig. 6. Projection directions by SADA in the first two dimensions versus p on D_1.

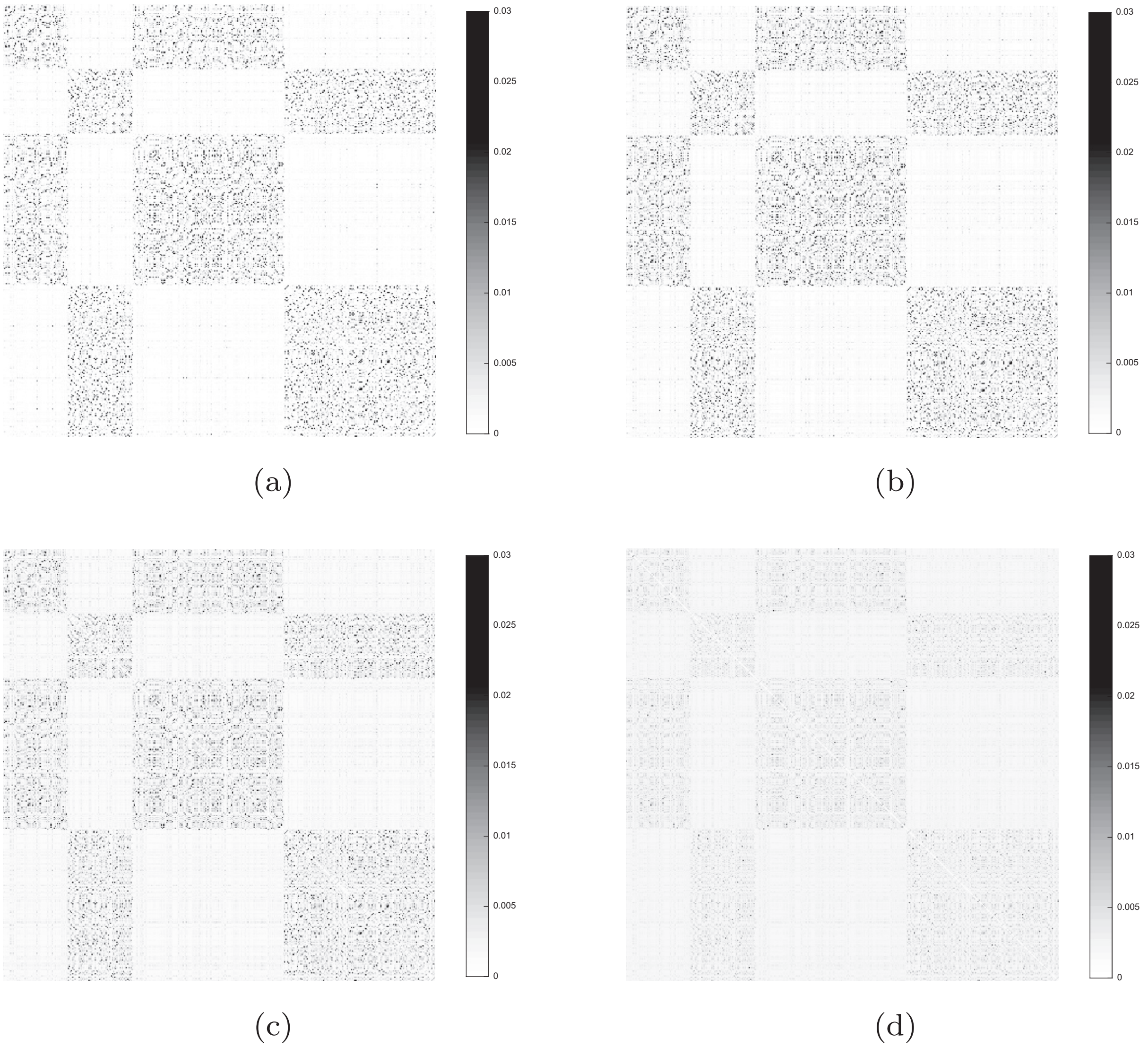


Fig. 7. Normalized implicit similarity matrix **S** learned by SADA on the *Colon* dataset. (a) p = 0.1, (b) p = 0.5, (c) p = 1.0, and (d) p = 1.5.

verges in twenty iterations. We also seek to explore the relationship between convergence and p. Fig. 11 shows that as p decreases, our algorithm converges in less iterations. For example, the number of iterations is 6 when $p = 0.1$ and 10 when $p = 0.75$. An explanation of this conclusion is that SADA can learn a sparser S with a smaller p.

So far, we have discussed how the parameters may affect the performance, now we describe how to set proper parameters in real applications. According to the analysis in Section 5.2.3 B and C, too large or too small γ and p will deteriorate the performance. Thus, we recommend taking value of γ in $(0.01, 10)$ and p in $(0.5, 1.25)$.

5.2.4. Statistical test

In order to analyze the performance of all methods statistically, we conduct Wilcoxon signed-rank test on the results of 7 methods on 9 datasets and the results are compared in Table 3 [32].

Numbers below the diagonal show whether the null-hypothesis H_0 that assumes the mean of two samples is equal to zero will be rejected. The value 1 indicates that we are confident to reject H_0 while 0 means that H_0 cannot be rejected. Numbers above the diagonal line are p-values, which measure the probabilities of H_0 and a smaller p-value indicates strong evidence against H_0. Observing the results in Tables 2,3, SADA performs better than sSelect, PRPC and ULAP but fails to show significant superiority to LSDF, RLSR and DSFFS. Note that the computational complexities of SADA, LSDF, RLSR and DSFFS are $O\left(d^2(n+m)\right), O\left(d^2n^2\right), O\left(nd^3+dnc\right), O\left(d^3+nd^2+dnc\right)$, respectively, where d is the number of features, n is the number of samples, m is the projection dimensions and c is the number of classes. Obviously, our proposed method has lower com-

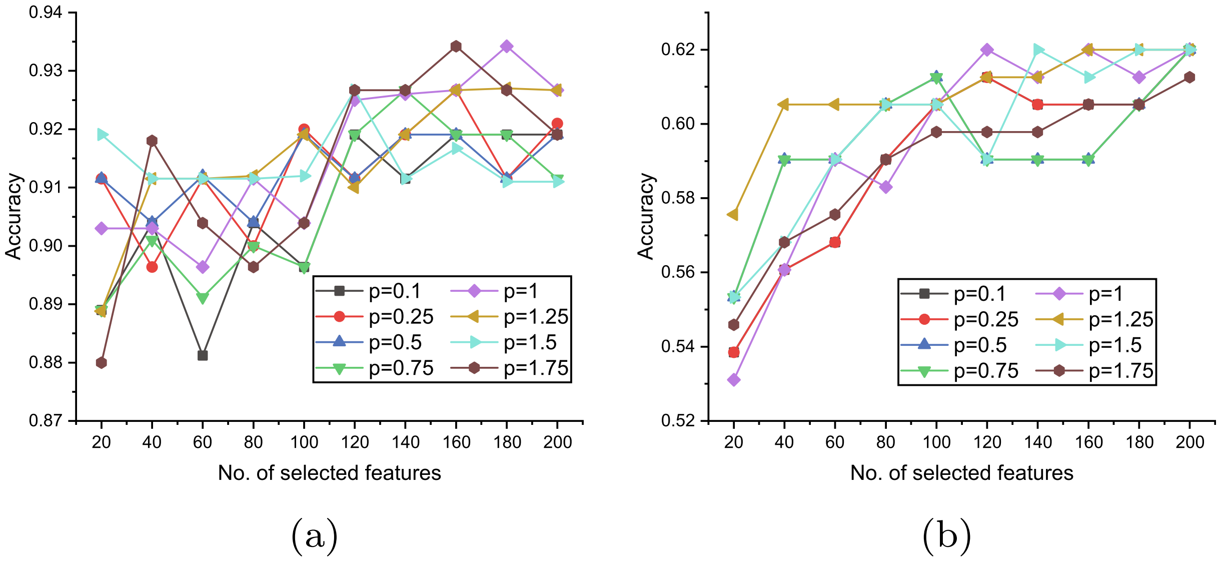


Fig. 8. Accuracy versus p. (a) Results on the *Srbct* dataset. (b) Results on the *Colon* dataset.

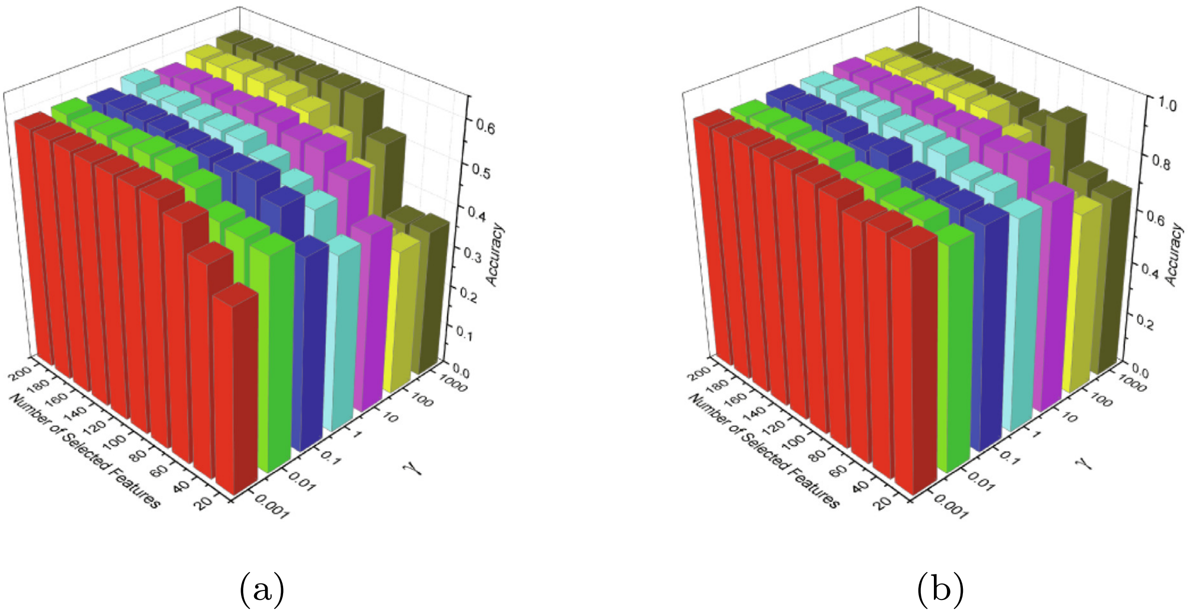


Fig. 9. Accuracy versus γ and the number of selected features. (a) Results on the *Srbct* dataset. (b) Results on the *Colon* dataset.

putational complexity than LSDF. For high-dimensional data with a large d (larger than n), our proposed method also shows superiority to RLSR and DSFFS in computational complexity.

6. Conclusions

In this paper, we have proposed a novel semi-supervised feature selection method, named SADA, that performs feature selection and implicit adaptive local structure learning simultaneously. The new method simultaneously learns a projection matrix $\mathbf{W}$ and an implicit adaptive similarity matrix $\mathbf{S}$ from both labeled and unlabeled data. In the new objective function, the $\ell_{2,p}$ norm is imposed on the pair-wise projected distances and experimental results show that learned implicit similarity matrix S will become sparser with the decrease of p. An iterative optimization algorithm with proved convergence is proposed to optimize the new model in which $\mathbf{W}$ is computed from the learned $\mathbf{S}$. Therefore, we can use it to adaptively preserve the locality and select high-quality features from the implicit similarity matrix. The empirical results for both synthetic datasets and benchmark datasets demonstrate the superior performance of SADA. In future work, we will focus on improving the speed of SADA.

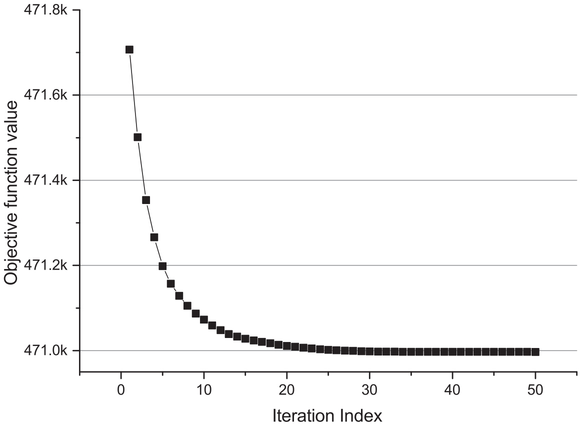


Fig. 10. Convergence curve of SADA on the Colon dataset.

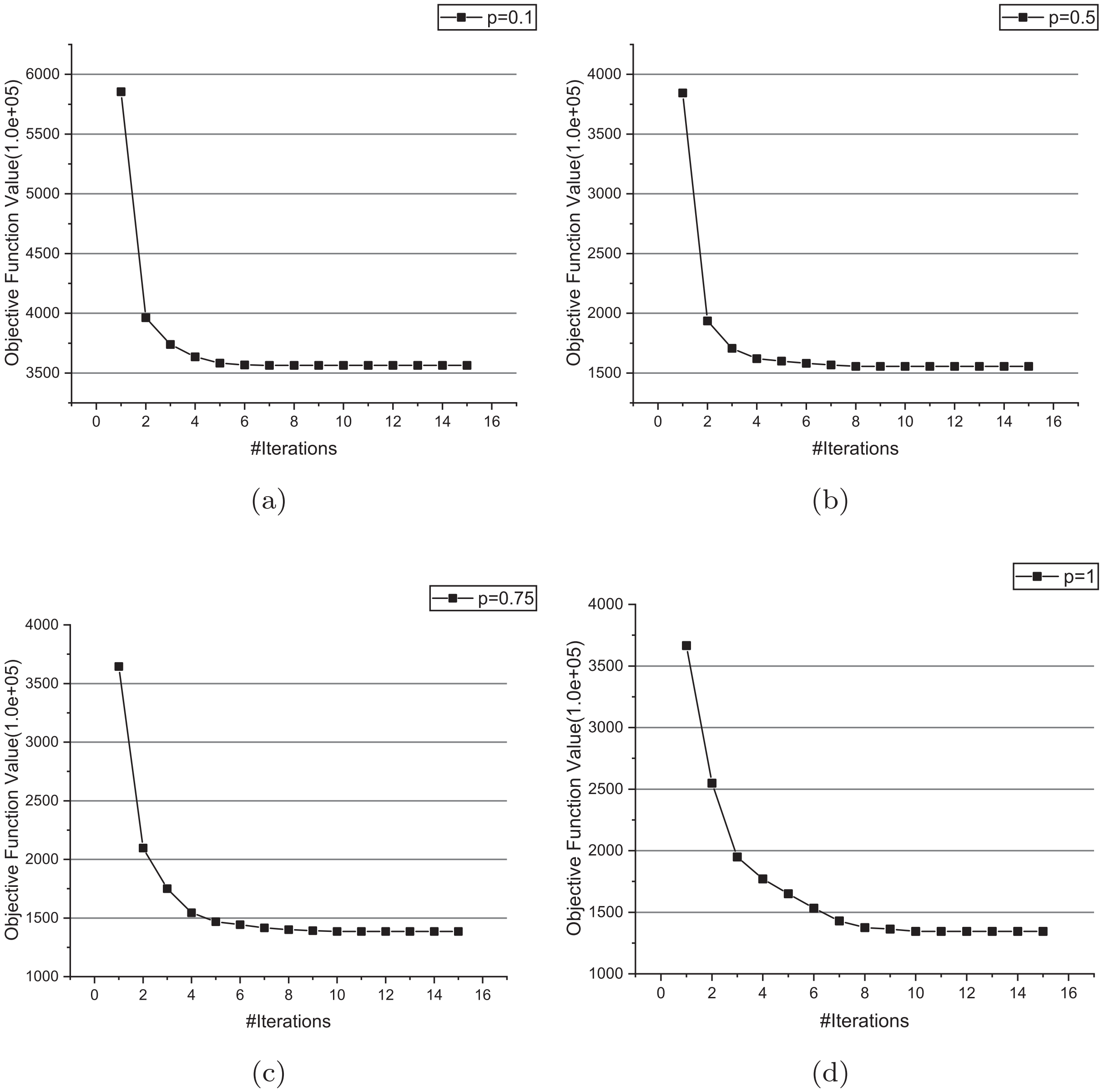


Fig. 11. Convergence comparison results for different values of the p parameter on the *Srbct* dataset ($\gamma = 1, m = 100$).

Table 3
Wilcoxon signed-ranks test of 7 methods, p-value(above diagonal), and rejections of null-hypothesis (below diagonal).

	LSDF	sSelect	PRPC	RLSR	DSFFS	ULAP	SADA
LSDF	0	.008	.070	.238	.250	1	.215
sSelect	1	0	.004	.004	.004	.004	.004
PRPC	0	1	0	.129	.129	.109	.004
RLSR	0	1	0	0	.688	.191	.796
DSFFS	0	1	0	0	0	.203	.820
ULAP	0	1	0	0	0	0	.011
SADA	0	1	1	0	0	1	0

CRediT authorship contribution statement

Weichan Zhong: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft. **Xiaojun Chen:** Conceptualization, Methodology, Validation, Investigation, Writing - original draft. **Feiping Nie:** Conceptualization, Project administration, Writing - original draft. **Joshua Zhexue Huang:** Project administration, Writing - original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by Major Project of the New Generation of Artificial Intelligence (No. 2018AAA0102900), NSFC under Grant No. 61773268 Natural Science Foundation of SZU (Grant No. 000346) and the Shenzhen Research Foundation for Basic Research, China (Nos. JCYJ20180305124149387).

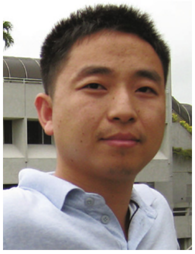
References

[1] S.H. Huang, Supervised feature selection: a tutorial, Artif. Intell. Res. 4 (2) (2015) 22.
[2] Y. Luo, D. Tao, C. Xu, D. Li, C. Xu, Vector-valued multi-view semi-supervised learning for multi-label image classification, in: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, AAAI Press, 2013, pp. 647–653.
[3] C. Tang, X. Liu, P. Wang, C. Zhang, M. Li, L. Wang, Adaptive hypergraph embedded semi-supervised multi-label image annotation, IEEE Trans. Multimedia 21 (11) (2019) 2837–2849.
[4] R. Zhu, F. Dornaika, Y. Ruichek, Learning a discriminant graph-based embedding with feature selection for image categorization, Neural Networks 111 (2019) 35–46.
[5] X. Chen, G. Yuan, F. Nie, J.Z. Huang, Semi-supervised feature selection via rescaled linear regression, in: Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 1525–1531.
[6] C. Shi, Z. Gu, C. Duan, Q. Tian, Multi-view adaptive semi-supervised feature selection with the self-paced learning, Signal Process. 168 (2020) 107332.
[7] J. Li, X. Liang, P. Li, W. Zhang, Q. Du, H. Yuan, Two-dimensional semi-supervised feature selection, in: 2020 10th International Conference on Information Science and Technology (ICIST), IEEE, 2020, pp. 280–287.
[8] Z. Zhao, H. Liu, Semi-supervised feature selection via spectral analysis, in, in: Proceedings of the 2007 SIAM International Conference on Data Mining, 2007, pp. 641–646.
[9] J. Zhao, K. Lu, X. H, Locality sensitive semi-supervised feature selection, Neurocomputing 71(10) (2008) 1842–1849.
[10] G. Doquire, M. Verleysen, A graph laplacian based approach to semi-supervised feature selection for regression problems, Neurocomputing 121 (2013) 5–13.
[11] J. Xu, B. Tang, H. He, H. Man, Semisupervised feature selection based on relevance and redundancy criteria, IEEE Trans. Neural Networks Learn. Syst. 99 (2016) 1–11.
[12] J. Ren, Z. Qiu, W. Fan, H. Cheng, P.S. Yu, Forward semi-supervised feature selection, in: Proceedings of the 12th Pacific-Asia Conference in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 970–976.
[13] Z. Xu, I. King, M.R.T. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Networks 21 (7) (2010) 1033–1047.
[14] X. Wu, K. Yu, H. Wang, W. Ding, Online streaming feature selection, in: Proceedings of the 27th international conference on machine learning (ICML-10), Citeseer, 2010, pp. 1159–1166.
[15] S. Eskandari, M.M. Javidi, Online streaming feature selection using rough sets, Int. J. Approximate Reasoning 69 (2016) 35–57.
[16] P. Zhou, X. Hu, P. Li, X. Wu, Online streaming feature selection using adapted neighborhood rough set, Inf. Sci. 481 (2019) 258–279.
[17] H. Chen, T. Li, X. Fan, C. Luo, Feature selection for imbalanced data based on neighborhood rough sets, Inf. Sci. 483 (2019) 1–20.
[18] G. Yuan, X. Chen, C. Wang, F. Nie, L. Jing, Discriminative semi-supervised feature selection via rescaled least squares regression-supplement, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI Press, New Orleans, Louisiana, USA, 2018.
[19] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, X. Zhou, Semisupervised feature selection via spline regression for video semantic recognition, IEEE Trans. Neural Netw. Learn. Syst. 26 (2) (2014) 252–264.
[20] Y. Saeys, T. Abeel, Y. Van de Peer, Robust feature selection using ensemble feature selection techniques, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 313–325.
[21] P. Drotár, M. Gazda, L. Vokorokos, Ensemble feature selection using election methods and ranker clustering, Inf. Sci. 480 (2019) 365–380.
[22] C.-F. Tsai, Y.-T. Sung, Ensemble feature selection in high dimension, low sample size datasets: parallel and serial combination approaches, Knowl.-Based Syst. 106097 (2020).
[23] J. González-Domínguez, V. Bolón-Canedo, B. Freire, J. Touriño, Parallel feature selection for distributed-memory clusters, Inf. Sci. (2019).
[24] L. Venkataramana, S.G. Jacob, R. Ramadoss, A parallel multilevel feature selection algorithm for improved cancer classification, J. Parallel Distrib. Comput. 138 (2020) 78–98.

[25] L. Zheng, F. Chao, N. Mac Parthaláin, D. Zhang, Q. Shen, Feature grouping and selection: a graph-based approach, Inf. Sci. 546 (2021) 1256–1272.
[26] H. Wang, Y. Zhang, J. Zhang, T. Li, L. Peng, A factor graph model for unsupervised feature selection, Inf. Sci. 480 (2019) 144–159.
[27] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press Professional Inc, San Diego, CA, USA, 1990.
[28] Y. Shi, J. Miao, Z. Wang, P. Zhang, L. Niu, Feature selection with $\ell_{2,1-2}$ regularization, IEEE Trans. Neural Networks Learn. Syst. 29 (10) (2018) 4967–4982.
[29] M. Zhang, C. Ding, Y. Zhang, F. Nie, Feature selection at the discrete limit, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.
[30] X. Chen, G. Yuan, W. Wang, F. Nie, X. Chang, J.Z. Huang, Local adaptive projection framework for feature selection of labeled and unlabeled data, IEEE Trans. Neural Networks Learn. Syst. 29 (12) (2018) 6362–6373.
[31] H. Lütkepohl, Handbook of Matrices, vol. 1, Wiley Chichester, 1996.
[32] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Weichan Zhong is a student for Master degree of the College of Computer Science and Software, Shenzhen University, Shenzhen, China. Her current research interests include clustering and feature selection.

Xiaojun Chen (M'16) received a Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2011. He is currently an Associate Professor of the College of Computer Science and Software, Shenzhen University, Shenzhen, China. His current research interests include subspace clustering, topic model, feature selection, and massive data mining.



Feiping Nie received a Ph.D. degree in Computer Science from Tsinghua University, China, in 2009. His research interests are machine learning and its applications such as pattern recognition, data mining, computer vision, image processing and information retrieval. He has published more than 100 papers in the following top journals and conferences: TPAMI, IJCV, TIP, TNNLS/TNN, TKDE, TKDD, Bioinformatics, ICML, NIPS, KDD, IJCAI, AAAI et al. His papers have been cited more than 5000 times (Google scholar). He is now serving as Associate Editor or PC member for several prestigious journals and conferences in the related fields.

Joshua Zhexue Huang received a Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden. He is currently a Professor with the College of Computer Science and Software, Shenzhen University, Shenzhen, China, a Professor and a Chief Scientist of the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China, and an Honorary Professor with the Department of Mathematics, The University of Hong Kong, Hong Kong. His current research interests include data mining, machine learning, and clustering algorithms.