Applied Intelligence (2024) 54:9987–10006
https://doi.org/10.1007/s10489-024-05691-9

Constrained feature weighting for semi-supervised learning

Xinyi Chen[1] · Li Zhang[1] · Lei Zhao[1] · Xiaofang Zhang[1]

Accepted: 14 July 2024 / Published online: 31 July 2024

Abstract
Semi-supervised feature selection plays a crucial role in semi-supervised classification tasks by identifying the most informative and relevant features while discarding irrelevant or redundant features. Many semi-supervised feature selection approaches take advantage of pairwise constraints. However, these methods either encounter obstacles when attempting to automatically determine the appropriate number of features or cannot make full use of the given pairwise constraints. Thus, we propose a constrained feature weighting (CFW) approach for semi-supervised feature selection. CFW has two goals: maximizing the modified hypothesis margin related to cannot-link constraints and minimizing the must-link preserving regularization related to must-link constraints. The former makes the selected features strongly discriminative, and the latter makes similar samples with selected features more similar in the weighted feature space. In addition, L1-norm regularization is incorporated in the objective function of CFW to automatically determine the number of features. Extensive experiments are conducted on real-world datasets, and experimental results demonstrate the superior effectiveness of CFW compared to that of the existing popular supervised and semi-supervised feature selection methods.

Keywords Semi-supervised learning · Feature selection · Feature weighting · Pairwise constraint · Hypothesis margin

1 Introduction

For decades, semi-supervised learning has yielded promising results in situations where obtaining labeled data is an expensive or time-consuming process. As a pre-processing method in the semi-supervised learning domain, semi-supervised feature selection aims at identifying a subset of relevant and informative features from a large set of input features [1, 2]. By leveraging both labeled and unlabeled data, semi-supervised feature selection methods can enhance the performance of models by incorporating the underlying data structure information furnished by the unlabeled data. Presently, feature selection has some practical applications, such as cancer classification [3, 4], text classification [5, 6], and image classification [7].

Researchers have proposed numerous semi-supervised feature selection methods. From the perspective of the input semi-supervised information, two types of learning frameworks are available for these methods: label-guided and constraint-guided frameworks. A label-guided framework provides a labeled dataset containing a few labeled samples and many unlabeled samples, while a constraint-guided framework addresses a constrained dataset containing some pairwise constraints and numerous unlabeled samples. Note that a labeled dataset can be transformed into a constrained dataset in the meaning of neighborhood, but not vice versa. Thus, the methods utilizing these two learning frameworks intersect. We do not discuss the possible transformations between these two types of frameworks here, we simply describe methods that operate under these two paradigms.

Generally, semi-supervised algorithms are mainly constructed from supervised methods, unsupervised methods, and both types of methods under the label-guided framework [3, 8–12]. Some examples are given as follows. The neighborhood discrimination index (NDI) method is supervised [13], and a Laplacian score (LS) is unsupervised [14]. On the basis of the NDI and LS, Pang and Zhang proposed a

✉ Li Zhang
zhangliml@suda.edu.cn

Xinyi Chen
20224227056@stu.suda.edu.cn

Lei Zhao
zhaol@suda.edu.cn

Xiaofang Zhang
xfzhang@suda.edu.cn

[1] School of Computer Science and Technology, Soochow University, Suzhou 215006, Jiangsu, China

semi-supervised neighborhood discrimination index (SSNDI) [3] and a recursive feature retention (RFR) method [8] for semi-supervised feature selection. The maximum-relevance and minimum-redundancy (MRMR) criterion is supervised [15], while the Pearson correlation coefficient (PCC) is unsupervised [12]. Based on the MRMR criterion and the PCC, a relevance, redundancy and Pearson criterion (RRPC) was proposed [12]. A multi-class semi-supervised LIR (MSLIR) method was proposed based on a supervised method called logistic I-Relief (LIR) [11]. This approach assigns pseudo labels to unlabeled data [9]. Tang and Zhang developed a local preserving LIR (LPLIR) method by incorporating a manifold regularization term into LIR [10]. A locality sensitive discriminant feature (LSDF) algorithm was presented based on the Fisher criterion and two adjacent graphs. This method aims to maximize the margin between samples that belong to different classes and discover the geometric structure of data using both labeled and unlabeled data [16]. It is interesting to note that the LSDF algorithm can be easily converted into a constrained-guided framework [17].

During the data annotation process, we may not know anything about the labels of samples without a priori knowledge, but we may judge whether two samples are similar or not. Constraints are binary annotations that indicate whether two samples are similar (must-link constraints) or not similar (cannot-link constraints) [18, 19]. In the constraint-guided framework, algorithms can utilize constraints and unlabeled information by constructing graphs. Classic supervised constraint scores (CSs) were proposed to evaluate the relevance of features using Laplacian matrices constructed according to pairwise constraints [20]. On the basis of CSs, many graph-based semi-supervised constraint scores have been proposed. Benabdeslem and Hindawi [21] introduced a new semi-supervised method based on CSs, called constrained Laplacian score (CLS). CLS uses the given must-link and cannot-link constraints to construct adjacent graphs and then corresponding Laplacian matrices. Salmi et al. [17] proposed a new constraint score based on a similarity matrix, called the similarity-based constraint score (SCS). The SCS can be implemented in the selected feature subspace, and it constructs similarity graphs to evaluate the relevance of feature subsets. Samah et al. [22] developed a basic Relief with side constraints (Relief-SC) algorithm that adopts only cannot-link constraints to solve a simple convex problem in a closed form. Chen et al. [23] took advantage of CSs and Relief-SC and then explored a new semi-supervised method called an iterative constraint score based on hypothesis margin (HMICS). This method not only considers the relevance between features but also calculates the hypothesis margin of a single feature.

Under the constraint-guided framework, the above semi-supervised methods cannot automatically determine the optimal number of features, which is the main issue faced by these methods. In addition, Relief-SC is unable to fully utilize pairwise constraints, which causes it to miss the information provided by must-link constraints, and it is sensitive to the given cannot-link constraints because of a lack of adequate neighborhood information.

To overcome the drawbacks of Relief-SC, this study proposes a novel semi-supervised feature selection approach, called constrained feature weighting (CFW). By utilizing the hypothesis margin idea derived from Relief-SC, CFW modifies the definition of the hypothesis margin calculated from cannot-link constraints to enrich the available neighborhood information. By adjusting the hypothesis margin in a logarithmic manner and introducing an L1-norm regularization term, the optimization process of CFW tends to produce a sparse solution, effectively selecting only the most informative features with non-zero weights and leading to a more compact and interpretable model. Moreover, CFW designs a must-link preserving regularization that contains the information provided by must-link constraints, which can be used to make similar samples with selected features more similar in the weighted feature space. The main contributions of this study are as follows.

- We redefine the process of calculating the hypothesis margin to reduce its sensitivity to cannot-link constraints. The modified hypothesis margin is derived from multiple nearest neighbors of the cannot-link constraints, whose probabilities are also considered to ensure the robustness of the selection process.
- We design a must-link preserving regularization that contains information provided by must-link constraints. Our goal is to minimize this regularization to maintain the desired relationships between the input samples. In this case, similar (or must-link) samples are more similar in the weighted feature space.
- We propose CFW based on the modified hypothesis margin concept and the must-link preserving regularization. CFW makes full use of pairwise constraints and the given unlabeled data, where the modified hypothesis margin depends on the cannot-link constraints and unlabeled samples, and the must-link preserving regularization considers the information contained in the must-link constraints. In addition, CFW can automatically determine the optimal number of features by incorporating the L1-norm regularization term into its objective function.

The remainder of this paper is organized as follows. Section 2 provides a brief explanation of the pairwise constraints and algorithms related to the hypothesis margin. In Section 3, we describe our proposed method in detail. Furthermore, experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2 Related work

In this section, we briefly review some constraint-guided methods: CS, LSDF, CLS, Relief-SC and HM-ICS. Under a constraint-guided semi-supervised learning framework, assume that we have two sets of pairwise constraints, $\mathcal{M}$ and $\mathcal{C}$, and a set $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ of unlabeled samples, where $\mathbf{x}_i = [x_{i1}, \cdots, x_{id}]^T \in \mathbb{R}^d$ is the i-th unlabeled sample, d is the number of features, n is the number of samples, $\mathcal{M}$ is the set of must-link constraints, and $\mathcal{C}$ is the set of cannot-link constraints.

$$\begin{aligned} \mathcal{M} &= \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are similar}\right\} \\ \mathcal{C} &= \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are dissimilar}\right\} \end{aligned} \tag{1}$$

Let $F = \{f_1, \cdots, f_d\}$ be the set of feature indexes, and $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_d] \in \mathbb{R}^{n \times d}$ be the feature matrix, where $\mathbf{f}_r = [x_{1r}, \cdots, x_{nr}]^T \in \mathbb{R}^n$ is the r-th feature vector.

2.1 CS

Algorithms under the constraint-guided framework are often designed according to the basic CS method. The original CS method is supervised and computes a score for each feature according to both $\mathcal{M}$ and $\mathcal{C}$ [20]. For a feature $f_r \in F$, the score function is as follows:

$$CS(f_r) = \frac{\sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \left(x_{ir} - x_{jr}\right)^2}{\sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} \left(x_{ir} - x_{jr}\right)^2} \tag{2}$$

CS can be expressed in matrix form using similarity matrices $\mathbf{W}^{\mathcal{M}}$ and $\mathbf{W}^{\mathcal{C}}$, which can be constructed by constraint sets $\mathcal{M}$ and $\mathcal{C}$, respectively. That is

$$W_{ij}^{\mathcal{M}} = \begin{cases} 1, & \text{if } \left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{M} \text{ or } \left(\mathbf{x}_j, \mathbf{x}_i\right) \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and

$$W_{ij}^{\mathcal{C}} = \begin{cases} 1, & \text{if } \left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{C} \text{ or } \left(\mathbf{x}_j, \mathbf{x}_i\right) \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $W_{ij}^{\mathcal{M}}$ and $W_{ij}^{\mathcal{C}}$ are the i-th and j-th entries of $\mathbf{W}^{\mathcal{M}}$ and $\mathbf{W}^{\mathcal{C}}$, respectively.

Then CS can be expressed as

$$CS(f_r) = \frac{\mathbf{f}_r^T \mathbf{L}^{\mathcal{M}} \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^{\mathcal{C}} \mathbf{f}_r} \tag{5}$$

According to CS, some semi-supervised CS methods, such as LSDF [16] and CLS [21], have been proposed.

2.2 LSDF

Zhao et al. [16] proposed LSDF, where $\mathbf{W}^{\mathcal{M}}$ is substituted with $\mathbf{W}^{KNN}$. This approach constructs the similarity matrix $\mathbf{W}^{KNN}$ using both must-link constraints and unlabeled data samples. That is

$$W_{ij}^{KNN} = \begin{cases} \gamma, & \text{if } \left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{M} \\ 1, & \text{if } \left(\mathbf{x}_i \in X_U \text{ or } \mathbf{x}_j \in X_U\right) \text{ and} \\ & \left(\mathbf{x}_i \in KNN\left(\mathbf{x}_j\right) \text{ or } \mathbf{x}_j \in KNN\left(\mathbf{x}_i\right)\right) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where γ serves as a constant, $KNN(\mathbf{x}_j)$ denotes the set of k nearest neighbors of the sample $\mathbf{x}_j$, and $X_U \subset X$ is the set of unlabeled samples.

The feature score formula for LSDF is as follows:

$$LSDF(f_r) = \frac{\mathbf{f}_r^T \mathbf{L}^{KNN} \mathbf{f}_r}{\mathbf{f}_r^T \mathbf{L}^{\mathcal{C}} \mathbf{f}_r} \tag{7}$$

where $\mathbf{L}^{KNN} = \mathbf{D}^{KNN} - \mathbf{W}^{KNN}$, $\mathbf{L}^{KNN}$ is the unnormalized constrained Laplacian matrix of $\mathbf{W}^{KNN}$, and $\mathbf{D}^{KNN}$ is the diagonal matrix that is calculated from $\mathbf{W}^{KNN}$.

2.3 CLS

Benabdeslem et al. [21] proposed a CLS method that modifies the similarity matrix (6), ensuring that the k nearest neighbors or must-link constraints are close to each other. The similarity matrix $\mathbf{W}^{CLS}$ of CLS is formulated as follows:

$$W_{ij}^{CLS} = \begin{cases} w_{ij}, & \text{if } \left(\left(x_i, x_j\right) \in \mathcal{M}\right) \text{ or } \left(x_i \in KNN\left(x_j\right) \text{ or } x_j \in KNN\left(x_i\right)\right) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where the similarity value w_{ij} is computed by the heat kernel function and has the form

$$w_{ij} = \exp\left(-\frac{\delta^2\left(\mathbf{x}_i, \mathbf{x}_j\right)}{2\sigma^2}\right), \quad i, j = 1, 2, \ldots, n \tag{9}$$

$\delta\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the Euclidean distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is a scaling parameter.

CLS is defined in terms of Laplacian matrices as follows:

$$CLS\left(f_m\right) = \frac{\mathbf{f}_m^T \mathbf{L}^{CLS} \mathbf{f}_m}{\mathbf{f}_m^T \mathbf{L}^{\mathcal{C}} \mathbf{D}^{CLS} \mathbf{f}_m} \tag{10}$$

where $\mathbf{L}^{CLS} = \mathbf{D}^{CLS} - \mathbf{W}^{CLS}$, $\mathbf{L}^{CLS}$ represents the unnormalized constrained Laplacian matrix of $\mathbf{W}^{CLS}$, and $\mathbf{D}^{CLS}$ is the diagonal matrix derived from $\mathbf{W}^{CLS}$.

2.4 Relief-SC

The hypothesis margin concept is derived from Relief-based methods [9–11, 24]. The hypothesis margin $\boldsymbol{\rho}(\mathbf{x}_i)$ of a sample $\mathbf{x}_i$ is the difference between the distance from $\mathbf{x}_i$ to its nearest miss (or the nearest sample to $\mathbf{x}_i$ in the opposite class) and the distance from $\mathbf{x}_i$ to its nearest hit (or the nearest sample in the same class as that of $\mathbf{x}_i$). That is,

$$\boldsymbol{\rho}(\mathbf{x}_i) = \left|\mathbf{x}_i - \mathbf{M}(\mathbf{x}_i)\right| - \left|\mathbf{x}_i - \mathbf{H}(\mathbf{x}_i)\right| \tag{11}$$

where $\mathbf{H}(\mathbf{x}_i)$ is the nearest hit of $\mathbf{x}_i$, and $\mathbf{M}(\mathbf{x}_i)$ is the nearest miss of $\mathbf{x}_i$. Note that (11) is defined based on label information. Thus, Relief-based methods are mostly supervised [11, 24], and the corresponding semi-supervised approaches are under the label-guided framework [9, 10].

Under the constraint-guided learning framework, Samah et al. [22] proposed Relief-SC using only cannot-link constraints $\mathcal{C}$. In this case, the hypothesis margin is re-designed for a cannot-link constraint and denoted by $\boldsymbol{\rho}\left(\mathbf{x}_i, \mathbf{x}_j\right)$ with pairwise constraint $\left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{C}$. Then, the hypothesis margin vector of $\left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{C}$ has the form shown below:

$$\boldsymbol{\rho}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left|\mathbf{x}_i - \mathbf{H}\left(\mathbf{x}_j\right)\right| - \left|\mathbf{x}_i - \mathbf{H}\left(\mathbf{x}_i\right)\right| \tag{12}$$

In (12), the nearest miss of $\mathbf{x}_i$ is given by $\mathbf{H}(\mathbf{x}_j)$, which is also the nearest hit of $\mathbf{x}_j$. The objective function of Relief-SC can be described as follows:

$$\begin{aligned} &\max_{\mathbf{w}} \ \mathbf{w}^T \left(\sum_{\left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{C}} \boldsymbol{\rho}\left(\mathbf{x}_i, \mathbf{x}_j\right) \right) \\ &\text{s.t.} \ \ \|\mathbf{w}\|_2^2 = 1, \quad \mathbf{w} \geq 0 \end{aligned} \tag{13}$$

where $\|\cdot\|_2$ is the L2-norm of a vector, and $\mathbf{w}$ is the feature weighting vector that reveals the impact of each feature on enlarging the margin. In fact, $\mathbf{w}$ can also be considered feature scores. The higher a weight value is, the more discriminative the corresponding feature is.

2.5 HM-ICS

The above methods all give scores to features according to some criteria related to pairwise constraints. However, these methods ignore the correlation between features.

Based on (2) and (12), Chen et al. [23] proposed the HM-ICS algorithm for semi-supervised feature selection. Let R be the current selected feature subset. For any feature f_r in the candidate feature subset $\bar{R} \subset F$, HM-ICS calculates the score of $R \cup \{f_r\}$, which is defined as follows:

$$J\left(R \cup \{f_r\}\right) = \lambda ICS\left(R \cup \{f_r\}\right) + (1-\lambda)\frac{1}{w_r + \gamma} \tag{14}$$

where $ICS\left(R \cup \{f_r\}\right)$ is the iterative form of the CS for the subset $R \cup \{f_r\}$, $\lambda \in [0, 1]$ is a regularization parameter, $\gamma > 0$ is a constant parameter, and w_r is the weight of f_r, which is calculated by a modified Relief-SC algorithm. In each iteration, HM-ICS selects the feature f_r with the smallest score $J\left(R \cup \{f_r\}\right)$, R is updated by adding this feature, and $\bar{R}$ is updated by deleting it.

The iterative constraint score $ICS\left(R \cup \{f_r\}\right)$ in HM-ICS has the following form:

$$ICS\left(R \cup \{f_r\}\right) = \frac{\sum_{f_t \in R \cup \{f_r\}} \sum_{\left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{M}} \left(x_{it} - x_{jt}\right)^2}{\sum_{f_t \in R \cup \{f_r\}} \sum_{\left(\mathbf{x}_i, \mathbf{x}_j\right) \in \mathcal{C}} \left(x_{it} - x_{jt}\right)^2} \tag{15}$$

The modified Relief-SC in HM-ICS simplifies the hypothesis margin concept and defines a new margin formula as follows [23]:

$$\boldsymbol{\rho}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\left|\mathbf{x}_i - \mathbf{x}_j\right| - \left|\mathbf{x}_i - \mathbf{H}\left(\mathbf{x}_i\right)\right|\right) \tag{16}$$

It can be seen that the difference between (16) and (12) is that the modified Relief-SC strategy replaces $\mathbf{H}\left(\mathbf{x}_j\right)$ with $\mathbf{x}_j$ for simplicity.

3 Proposed method

This section discusses the proposed CFW algorithm in detail. We first provide a new way to calculate the hypothesis margin with cannot-link constraints and then design a must-link preserving regularization that can maintain the underlying relationships between samples. Next, we describe the objective function of CFW and its solution. Finally, we present the algorithm description of CFW. We use the notations mentioned before and assume that $(X, F, \mathcal{M}, \mathcal{C})$ is a semi-supervised information system, where X is a set of unlabeled samples, F is a set of feature indexes, and $\mathcal{M}$ and $\mathcal{C}$ are sets of must-link and cannot-link constraints, respectively.

3.1 Modified hypothesis margin

For the first time, Relief-SC provides a way to calculate the hypothesis margin based on cannot-link constraints [22], as shown in (12). However, Relief-SC is sensitive to the given cannot-link constraints. To remedy this drawback, we redefine the hypothesis margin calculation process with cannot-link constraints.

The sensitivity issue is caused by the lack of adequate neighborhood information. Relief-SC calculates the hypothesis margin with only one nearest hit for each of two dissimilar samples in a cannot-link constraint. Thus, it is possible to enrich the neighborhood information by finding more

near hits. Considering a cannot-link constraint $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$, we define its hypothesis margin as follows:

$$\boldsymbol{\rho}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\mathbf{x}_s \in NH(\mathbf{x}_j)} P(\mathbf{x}_s = H(\mathbf{x}_j) | \mathbf{w}) |\mathbf{x}_i - \mathbf{x}_s| - \sum_{\mathbf{x}_s \in NH(\mathbf{x}_i)} P(\mathbf{x}_s = H(\mathbf{x}_i) | \mathbf{w}) |\mathbf{x}_i - \mathbf{x}_s| \tag{17}$$

where $NH(\mathbf{x}_i)$ represents the set containing the k nearest neighbors of sample $\mathbf{x}_i$ under the weighted feature space, and $P(\mathbf{x}_s = H(\mathbf{x}_i) | \mathbf{w})$ denotes the probability that $\mathbf{x}_s$ is the nearest hit of $\mathbf{x}_i$ under the weighted feature space. One possible way to calculate the probability is

$$P(\mathbf{x}_s = H(\mathbf{x}_i) \mid \mathbf{w}) = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_s\|_{\mathbf{w}} / \sigma)}{\sum_{\mathbf{x}_s \in NH(\mathbf{x}_i)} \exp(-\|\mathbf{x}_i - \mathbf{x}_s\|_{\mathbf{w}} / \sigma)} \tag{18}$$

where $\|\mathbf{x}_i - \mathbf{x}_s\|_{\mathbf{w}} = \sum_{r=1}^{d} w_s |x_{ir} - x_{sr}|$ refers to the weighted distance between $\mathbf{x}_i$ and $\mathbf{x}_s$ under the weighted feature space, and $\sigma > 0$ is a preset parameter.

Equation (17) improves the process of calculating the hypothesis margin (12) from two perspectives. First, (17) considers not only the k nearest neighbors of $\mathbf{x}_i$ and $\mathbf{x}_j$ separately but also their probabilities of being the nearest hit, which enriches the neighborhood information and alleviates the sensitivity of (12) to constraints. Second, the hypothesis margin in (17) is calculated in the weighted feature space, which adaptively adjusts the margin with feature weights during the iteration process and then finds discriminative features.

3.2 Must-link preserving regularization

As mentioned before, the hypothesis margin uses only the information contained in the cannot-link constraints. To incorporate the information provided by must-link constraints, we define a must-link preserving regularization that can maintain the data structure of the must-link constraints and make similar samples more similar. The must-link preserving regularization in the weighted feature space is defined as follows:

$$J_R(\mathbf{w}) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{w}}^2 = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \|\mathbf{w} \odot \mathbf{x}_i - \mathbf{w} \odot \mathbf{x}_j\|^2 \tag{19}$$

where the feature weight $\mathbf{w} \geq 0$, and $\odot$ is the element-wise multiplication operator.

In accordance with (19), the must-link preserving regularization $J_R(\mathbf{w})$ describes the scatter of the similar samples provided by the must-link constraints in the weighted feature space induced by $\mathbf{w}$. Minimizing $J_R(\mathbf{w})$ means that similar samples should be as close as possible in the weighted feature space. In other words, the must-link structure in the original space is maintained in the weighted feature space, as a smaller distance signifies a stronger similarity between the corresponding samples.

Now, we express the must-link preserving regularization in matrix form. First, we need to construct a similarity graph $S^{\mathcal{M}}$ according to the must-link constraints $\mathcal{M}$ by taking samples in the set X as vertices. In this graph, if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$, then an edge exists between them. Thus, the similarity matrix $\mathbf{S}^{\mathcal{M}}$ is represented as follows:

$$S_{ij}^{\mathcal{M}} = \begin{cases} 1, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \text{ or } (\mathbf{x}_j, \mathbf{x}_i) \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

where $S_{ij}^{\mathcal{M}}$ is the i-th and j-th entries of $\mathbf{S}^{\mathcal{M}}$. For simplification, let $\mathbf{m}_i$ be the weighted image of $\mathbf{x}_i$ in the weighted feature space; that is, $\mathbf{m}_i = \mathbf{w} \odot \mathbf{x}_i$, $i = 1, \cdots, n$. By substituting $\mathbf{m}_i$, $i = 1, \cdots, n$ and $S^{\mathcal{M}}$ into (19), we obtain

$$\begin{aligned} J_R(\mathbf{w}) &= \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \|\mathbf{m}_i - \mathbf{m}_j\|^2 \\ &= \sum_{i,j=1}^{n} \|\mathbf{m}_i - \mathbf{m}_j\|^2 S_{ij}^{\mathcal{M}} \\ &= \sum_{i,j=1}^{n} (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) S_{ij}^{\mathcal{M}} \\ &= 2 \sum_{i=1}^{n} D_{ii}^{\mathcal{M}} \mathbf{m}_i^T \mathbf{m}_i - 2 \sum_{i,j=1}^{n} W_{ij}^{\mathcal{M}} \mathbf{m}_i^T \mathbf{m}_j \\ &= \text{trace}\left(2\mathbf{M}\mathbf{L}^{\mathcal{M}}\mathbf{M}^T\right) \end{aligned} \tag{21}$$

where $trace(\cdot)$ denotes the sum of the diagonal elements of a matrix, $\mathbf{M} = [\mathbf{m}_1, \cdots, \mathbf{m}_n] \in \mathbb{R}^{d \times n}$, $D_{ii}^{\mathcal{M}} = \sum_{j=1}^{n} S_{ij}^{\mathcal{M}}$ denotes the diagonal element of $\mathbf{D}^{\mathcal{M}}$, and the Laplacian matrix $\mathbf{L}^{\mathcal{M}}$ is defined as

$$\mathbf{L}^{\mathcal{M}} = \mathbf{D}^{\mathcal{M}} - \mathbf{S}^{\mathcal{M}} \tag{22}$$

Furthermore, $\mathbf{M} = \mathbf{F}\mathbf{W}$ because $\mathbf{m} = \mathbf{w} \odot \mathbf{x}$, where $\mathbf{F} \in \mathbf{R}^{n \times d}$ is the feature matrix, and $\mathbf{W} = diag(\mathbf{w})$ is the diagonal matrix. Obviously, $J_R(\mathbf{w})$ can be rewritten as

$$J_R(\mathbf{w}) = \text{trace}\left(2\mathbf{W}^T\mathbf{F}^T\mathbf{L}^{\mathcal{M}}\mathbf{F}\mathbf{W}\right) \tag{23}$$

3.3 Optimization problem and solution

On the basis of the modified hypothesis margin (17) and the must-link preserving regularization (23), we construct the optimization problem for CFW. That is,

$$\begin{aligned} \min_{\mathbf{w}} \ & J(\mathbf{w}) = \log\left(1+\exp\left(-\mathbf{w}^T\mathbf{z}\right)\right) + \lambda_1\|\mathbf{w}\|_1 + \lambda_2 \text{trace}\left(\mathbf{W}^T\mathbf{Q}\mathbf{W}\right) \\ \text{s.t.} \ & \mathbf{w} \geq 0 \end{aligned} \tag{24}$$

where $\mathbf{z}$ is the margin vector calculated by the cannot-link constraints, $\|\mathbf{w}\|_1$ is the L1-norm of $\mathbf{w}$, $\mathbf{Q} = \mathbf{F}^T\mathbf{L}^{\mathcal{M}}\mathbf{F} \in \mathbb{R}^{d\times d}$, the parameter $\lambda_1 \geq 0$ controls the sparsity level in the weight vector $\mathbf{w}$, and the parameter $\lambda_2 \geq 0$ controls the must-link information of the samples. Here, the margin vector $\mathbf{z}$ is expressed by

$$\mathbf{z} = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \boldsymbol{\rho}\left(\mathbf{x}_i,\mathbf{x}_j\right) \tag{25}$$

where $\boldsymbol{\rho}\left(\mathbf{x}_i,\mathbf{x}_j\right)$ is defined in (17).

The objective function of (24) consists of three terms. The first term in (24) is minimized to maximize the modified hypothesis margin calculated from the cannot-link constraints. In addition, the exponential and logarithmic functions are used to adjust the range of $\mathbf{w}^T\mathbf{x}$ and encourage the model to produce accurate predictions. The second term, $\|\mathbf{w}\|_1$, is the L1-norm regularization term, which encourages sparsity in the weight vector $\mathbf{w}$ and can automatically select features. A larger λ_1 value leads to stronger sparsity enforcement. The third term in (24) focuses on preserving a structure similar to that provided by the must-link constraints in the weighted feature space.

To solve CFW, we first prove that the optimization problem (24) is convex; thus it must have a globally optimal solution. Then, we demonstrate how to use the gradient descent method to obtain the final solution.

Theorem 1 *Given the feature matrix* $\mathbf{F} \in \mathbb{R}^{n\times d}$ *and the Laplacian matrix* $\mathbf{L}^{\mathcal{M}}$*, the must-link preserving regularization* (19) *is a convex function with respect to* $\mathbf{w}$*, where* $\mathbf{w} \geq 0$*.*

Theorem 2 *Given the feature matrix* $\mathbf{F} \in \mathbb{R}^{n\times d}$ *and the Laplacian matrix* $\mathbf{L}^{\mathcal{M}}$*, the optimization problem* (24) *is a convex problem with respect to* $\mathbf{w}$*, where* $\mathbf{w} \geq 0$*.*

Corollary 1 *Given the feature matrix* $\mathbf{F} \in \mathbb{R}^{n\times d}$ *and the Laplacian matrix* $\mathbf{L}^{\mathcal{M}}$*, the optimization problem* (24) *has a global solution with respect to* $\mathbf{w}$*.*

The proofs of Theorems 1 and 2 are given in Appendices A and B, respectively. Theorem 1 indicates that when the margin vector $\mathbf{z}$ is fixed, $J_R(\mathbf{w})$ is a convex function with respect to $\mathbf{w}$. Theorem 2 implies that the problem in (24) is a convex problem. For convex problems, it is known that any locally optimal point is also globally optimal. Thus, Corollary 1 holds true, and its proof can be omitted. Naturally, the convex problem could be solved by applying the gradient descent method.

For the purpose of applying the gradient descent method, when the margin vector $\mathbf{z}$ is fixed, we need to find the first partial derivative of $J(\mathbf{w})$ with respect to $\mathbf{w}$, which can be expressed as follows:

$$\nabla J(\mathbf{w}) = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -\frac{\exp\left(-\mathbf{w}^T\mathbf{z}\right)}{1+\exp\left(-\mathbf{w}^T\mathbf{z}\right)}\mathbf{z} + \lambda_1 + 2\lambda_2\mathbf{w}\odot\mathbf{q} \tag{26}$$

where $\mathbf{q} = [Q_{11},\cdots,Q_{dd}]^T \in \mathbb{R}^d$ is a vector composed of the diagonal elements of $\mathbf{Q}$. After obtaining $\nabla J(\mathbf{w})$, we can iteratively update the weight vector $\mathbf{w}$. Let t be the iteration variable. Then, we have

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \eta\nabla R(\mathbf{w}(t-1)) \tag{27}$$

where $0 < \eta \leq 1$ is the learning rate. Due to the constraint of $\mathbf{w} \geq 0$, $\mathbf{w}$ should abide by the following rules in each iteration:

$$w_r(t) = \begin{cases} w_r(t), & \text{if } w_r(t) > 0 \\ 0, & \text{otherwise} \end{cases}, \quad r = 1,\ldots,d \tag{28}$$

3.4 Algorithm description

CFW implements semi-supervised feature selection under the constraint-guided learning framework. CFW maximizes the hypothesis margin to magnify the discriminative features using cannot-link constraints and minimizes the must-link preserving regularization to strengthen the local structure of the similar samples.

A detailed description of the proposed algorithm is given in Algorithm 1. In step 1, CFW starts by initializing the feature weight vector $\mathbf{w}(0) = [1,\cdots,1]^T \in \mathbb{R}^d$ and the margin vector $\mathbf{z}(0) = [0,\cdots,0]^T \in \mathbb{R}^d$, where d is the number of features. Step 2 constructs a Laplacian matrix $\mathbf{L}^{\mathcal{M}}$ using (22). Steps 3–7 iteratively update the weight vector $\mathbf{w}$ based on the calculated margin vector until one of the preset convergence conditions is satisfied. The final weight vector $\mathbf{w}$ is returned in step 8.

Subsequently, we analyze the computational complexity of CFW. The computational complexity of constructing the Laplacian matrix $\mathbf{L}^{\mathcal{M}}$ by (22) is $O\left(n^2d\right)$, where n is the number of samples, and d is the number of features. Step 4 computes the margin vector $\mathbf{z}(t)$ via (25), which has a computational complexity level of $O\left(|\mathcal{C}|nd\right)$, where $|\mathcal{C}|$ is the number of cannot-link constraints. Step 5 updates $\mathbf{w}(t)$

Algorithm 1 CFW.

Input: Semi-supervised information system $\{X, F, \mathcal{M}, \mathcal{C}\}$, learning rate η, regularization parameters λ_1 and λ_2, the maximum iterative time T, and the permissive error θ;
Output: Feature weight $\mathbf{w}$;
1: Initialize $\mathbf{w}(0) = [1, \cdots, 1]^T \in \mathbb{R}^d$, and hypothesis margin vector $\mathbf{z}(0) = [0, \cdots, 0]^T \in \mathbb{R}^d$, and $t = 1$;
2: Construct Laplacian matrix $\mathbf{L}^{\mathcal{M}}$ according to (22);
3: **while** $t \leq T$ and $\|\mathbf{w}(t) - \mathbf{w}(t-1)\|_2 > \theta$ **do**
4: Calculate margin vector $\mathbf{z}(t)$ by (25);
5: Update weight vector $\mathbf{w}(t)$ by (26)–(28);
6: Let $t \leftarrow t + 1$;
7: **end while**

and has a computational complexity level of $O(nd)$. In each iteration, CFW has a computational complexity level of $O(|\mathcal{C}|nd)$. Then the total computational complexity of CFW is $O\left(T|\mathcal{C}|nd + n^2d\right)$.

Lastly, we delve into the properties of CFW. According to Theorem 2 and Corollary 1, the problem formulated in (24) is convex with respect to $\mathbf{w}$ and ensures a globally optimal solution. CFW is guaranteed to converge if $\mathbf{z}(t)$ remains fixed and the gradient descent method is applied to solve (24). However, $\mathbf{z}(t)$ evolves during the iteration process. The subsequent theorem explores how changes in $\mathbf{z}(t)$ influence on the solution to (24).

Theorem 3 *Given the learning procedure of CFW in Algorithm 1, the following inequalities*

$$J(\mathbf{w}(t) \mid \mathbf{z}(t)) \leq J(\mathbf{w}(t-1) \mid \mathbf{z}(t)) \tag{29}$$

hold true, where $J(\mathbf{w}(t) \mid \mathbf{z}(t))$ *represents the objective function* $J(\mathbf{w}(t))$ *when* $\mathbf{z}(t)$ *is fixed,* $t \geq 0$.

The proof of Theorem 3 is provided in Appendix C. Theorem 3 demonstrates that $J(\mathbf{w}(t))$ represents a better solution than $J(\mathbf{w}(t-1))$ when $\mathbf{z}(t)$ is fixed, which could be attributable to the gradient descent update rule. In essence, regardless of changes in $\mathbf{z}(t)$, the solution derived in the current iteration is better than the one from the previous iteration.

4 Experiments

To validate the feasibility and effectiveness of CFW, we perform extensive experiments on nine public datasets. The information of these datasets is listed in Table 1, including the number of samples (# Sample), the number of features (# Feature), and the number of classes (# Class) in each dataset. The features contained in all datasets are normalized to the interval of [0, 1].

All experiments were carried out in Pycharm 2020 and run in a hardware environment with an Intel Core i9 CPU at 2.50 GHz and 32 GB of RAM.

Table 1 Description of nine datasets used in experiments

No.	Dataset	# Sample	# Feature	# Class
1	CNAE-9	1080	856	9
2	CNS	42	989	5
3	Colon	62	2000	2
4	Glioma	50	4434	4
5	Normal	90	1277	13
6	Novartis	103	1000	4
7	ORL	400	1024	40
8	Prostate-GE	102	5966	2
9	Sj-leukemia	248	985	6

4.1 Analysis of CFW

We analyze our proposed CFW method according to its convergence, sparsity and discriminant ability, parameter sensitivity, and number of constraints. Each of the nine datasets [25–33] in Table 1 is randomly divided into a training set with 2/3 of the total samples and a test set with the remaining samples. In addition, we randomly select samples from the training set to construct a certain number of pairwise constraints, where one half of the constraints are must-link constraints, and the other half are cannot-link constraints.

4.1.1 Convergence

In experiments, we select 100 pairwise constraints, including 50 must-link constraints and 50 cannot-link constraints. Let $\eta = 0.01$ in (27), $k = 5$ in (17), which follows the setting used in [16], and the regularization parameters $\lambda_1 = 1$ and $\lambda_2 = 1$. The convergence of CFW can be validated by observing the variation exhibited by the objective function with respect to the iteration variable. Thus, we consider only the maximum number of iterations as the stop condition of CFW and set $T = 200$.

Figure 1 shows the trend curves yielded by the objective function vs. the number of iterations on the nine datasets. From Fig. 1, we can see that the objective function arrives at its minimum value after a certain number of iterations, which indicates that CFW is convergent. Generally, CFW converges within 50 iterations on most datasets, such as CNS (Fig. 1b) and Glioma (Fig. 1d). Notably, CFW can converge faster or slower, depending on the utilized dataset. For example, CFW converges quickly on the Colon dataset (Fig. 1c) and slowly on the CNAE-9 dataset (Fig. 1a). In short, CFW is convergent.

Based on the experiments conducted above, we find that CFW converges within 100 iterations on all datasets, with many requiring less than 50 iterations. To provide a comprehensive overview, we set the number of iterations to 100 and

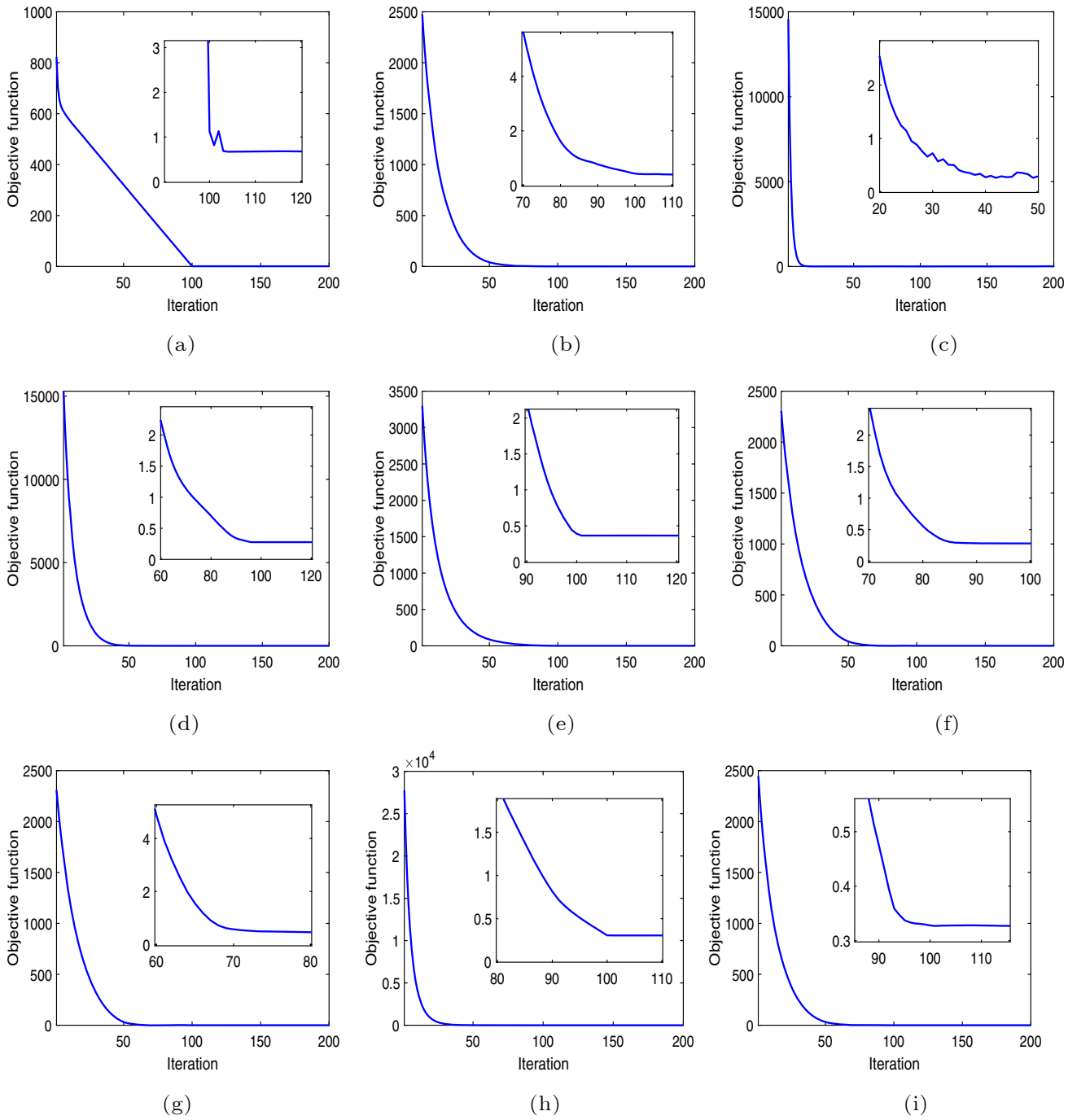

Fig. 1 Curves of objective function vs. iteration obtained by CFW on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

summarize the running times required by the CFW algorithm on these datasets in Table 2.

Referring to Table 2, it is evident that the CFW demonstrates commendable efficiency, with execution times under one minute on five datasets. For instance, a mere 25 seconds running time is required on the CNS dataset. The maximum recorded running time for CFW is 321 seconds on the CNAE-9 dataset. As analyzed in Section 3.4, the computational complexity of CFW is related to only the sample number n and the feature number d when T and $\mathcal{C}$ are given. Thus, CFW will take more time when dealing with the dataset with large number of samples and features, which is supported by running times in Table 2.

Table 2 Running time(s) of CFW on nine datasets

Dataset	CNAE-9	CNS	Colon	Glioma	Normal	Novartis	ORL	Prostate-GE	Sj-leukemia
Running time (s)	321.84	25.45	47.66	108.27	33.38	26.82	81.19	153.10	47.16

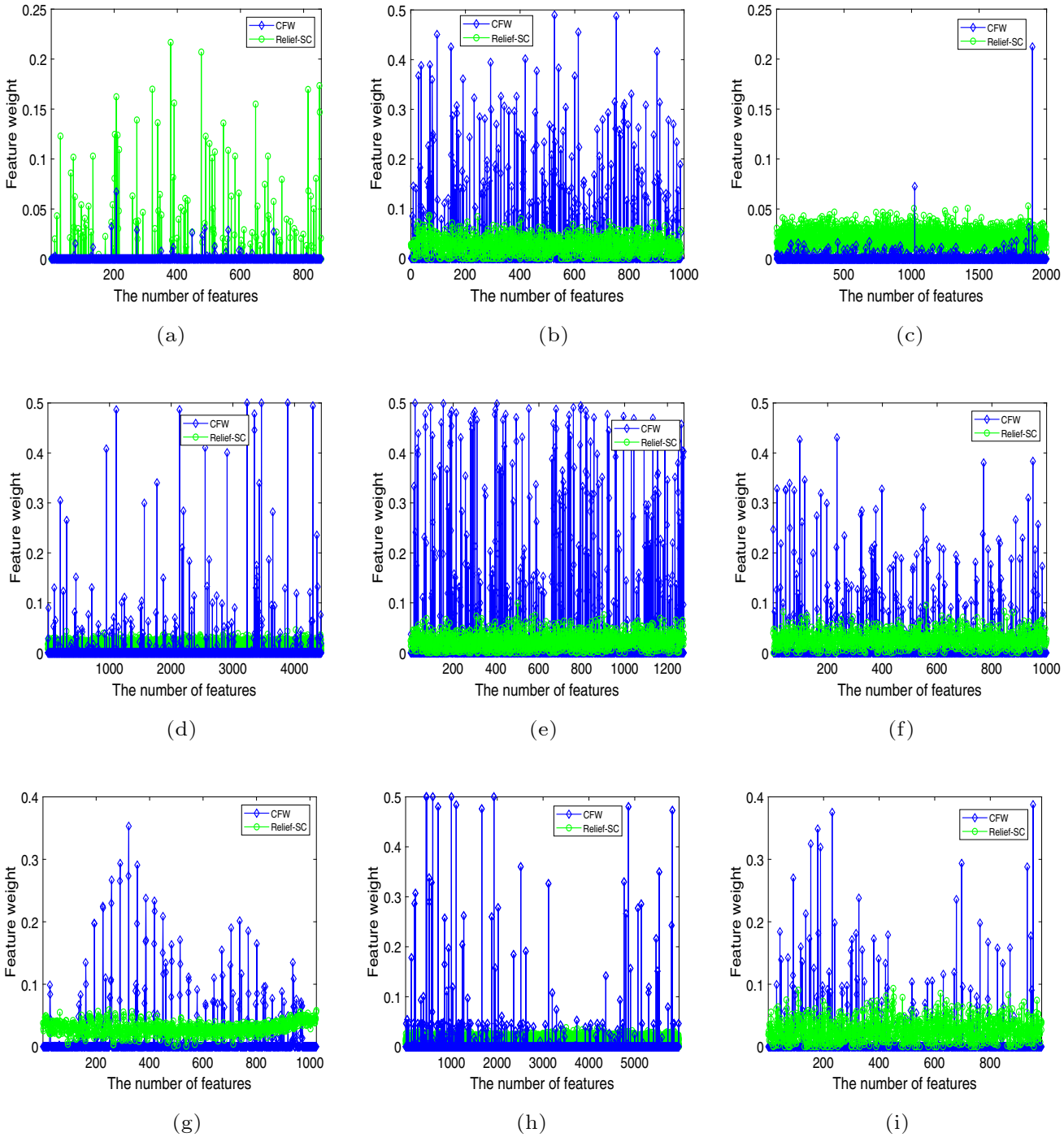

Fig. 2 Feature weight values obtained by CFW and Relief-SC on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

4.1.2 Sparsity and discriminant capability

The experimental settings employed here are the same as those utilized in Section 4.2.1 except the stopping conditions of CFW. Let $T = 100$ and $\theta = 0.001$ in step 3 of Algorithm 1. Because CFW is developed based on Relief-SC, we compare them on their sparsity and discriminant capabilities.

First, we observe the sparsity of CFW by plotting its feature weights on nine datasets, as shown in Fig. 2. Although both methods exhibit certain degrees of sparsity, CFW is much sparser than Relief-SC. We count the numbers of non-zero weights produced by both CFW and Relief-SC for each of these nine datasets and summarize them in Table 3, which further implies the good sparsity of CFW. For example, 2000 features are contained in the Colon dataset. CFW obtains 214 non-zero weights, while Relief-SC generates 1979 non-zero weights. Note that Relief-SC cannot always obtain sparse weights for some datasets, such as ORL. At the same time, we list the accuracies achieved by the nearest neighbor (NN) classifier with the features selected by both methods in Table 3. The findings indicate that CFW has better classification performance and can select more discriminative features than Releif-SC.

To further validate the discriminant capability of the features, we use the idea behind the Fisher criterion. Namely, good features minimize the must-link scatter and maximize the cannot-link scatter. Let $D(X)$ be the ratio of the must-link scatter to the cannot-link scatter with respect to the set X, which can be defined as follows:

$$D(X) = \frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2} \tag{30}$$

Generally, the smaller $D(X)$ is, the stronger the discriminant ability the features in the set X have. Let X_w be the weighted feature space, i.e., $\mathbf{x}_w = \mathbf{w} \odot \mathbf{x}$, where $\mathbf{x}_w \in X_w$ and $\mathbf{x} \in X$. Thus, we hope that $D(X_w)$ is much smaller than $D(X)$. In other words, it is better to make $D(X)/D(X_w)$ large.

Table 4 lists values of $D(X)$, $D(X_w)$ and $D(X)/D(X_w)$ obtained by CFW and Relief-SC. As can be seen from Table 4, the $D(X_w)$ values obtained by CFW are much smaller than the $D(X)$ values for all datasets, while the $D(X_w)$ values obtained by Relief-SC are not always smaller than the $D(X)$ values of all datasets. The $D(X)/D(X_w)$ values obtained by CFW are all greater than 1, which indicates that CFW selects highly discriminant feature subsets. However, the $D(X)/D(X_w)$ values obtained by Relief-SC are near 1 and even less than 1, which suggests that Relief-SC does not significantly improve the discriminant ability of the selected feature subset.

4.1.3 Parameter sensitivity

Here, we investigate the sensitivity of the parameters λ_1 and λ_2 in CFW and keep the other experimental settings unchanged. The value range for these two parameters is set to $\{0.01, 0.1, 1, 10, 100\}$.

The parameter analysis results are given in Fig. 3. As evident from this figure, the regularization parameters have different effects on the classification performance achieved on different datasets. For example, the CNAE-9 dataset (Fig. 3a) is significantly influenced by the parameters λ_1 and λ_2, exhibiting substantial variations. On some datasets (Colon, Glioma, Normal, and Prostate-GE), the accuracy of CFW fluctuates with the parameters. Conversely, the remaining datasets display relatively minimal fluctuations.

Furthermore, Fig. 3(c), (d), (e), (f), and (i) illustrate that the proposed algorithm achieves the highest classification accuracy when λ_1=1 and $\lambda_2 = 1$. Thus, we suggest that λ_1=1 and $\lambda_2 = 1$ in the following experiments.

Table 3 Number of non-zero weights and the corresponding accuracy obtained by CFW and Relief-SC on nine datasets

Dataset	#Non-zero weight		Accuracy (%)	
	Relief-SC	CFW	Relief-SC	CFW
CNAE-9	118	75	63.89	64.80
CNS	925	299	69.05	71.43
Colon	1979	214	82.30	85.63
Glioma	4121	154	69.85	70.59
Normal	1249	367	67.78	70.01
Novartis	984	253	76.69	77.58
ORL	1024	189	58.74	59.51
Prostate-GE	5509	183	78.43	86.27
Sj-leukemia	952	133	67.73	67.76

Table 4 $D(X)$, $D(\mathbf{w} \odot X)$, and $D(X)/D(\mathbf{w} \odot X)$ on nine dataset obtained by Relief-SC and CFW

Dataset	$D(X)$	$D(X_w)$		$D(X)/D(X_w)$	
		Relief-SC	CFW	Relief-SC	CFW
CNAE-9	0.92	0.92	0.61	1.00	1.50
CNS	0.34	0.30	0.11	1.14	3.09
Colon	0.88	0.83	0.32	1.06	2.75
Glioma	0.51	0.48	0.12	1.07	4.24
Normal	0.90	0.86	0.09	1.05	10.00
Novartis	0.46	0.40	0.10	1.15	4.60
ORL	0.46	0.47	0.18	0.99	2.56
Prostate-GE	0.86	0.94	0.64	0.92	1.35
Sj-leukemia	0.82	0.80	0.37	1.02	2.22

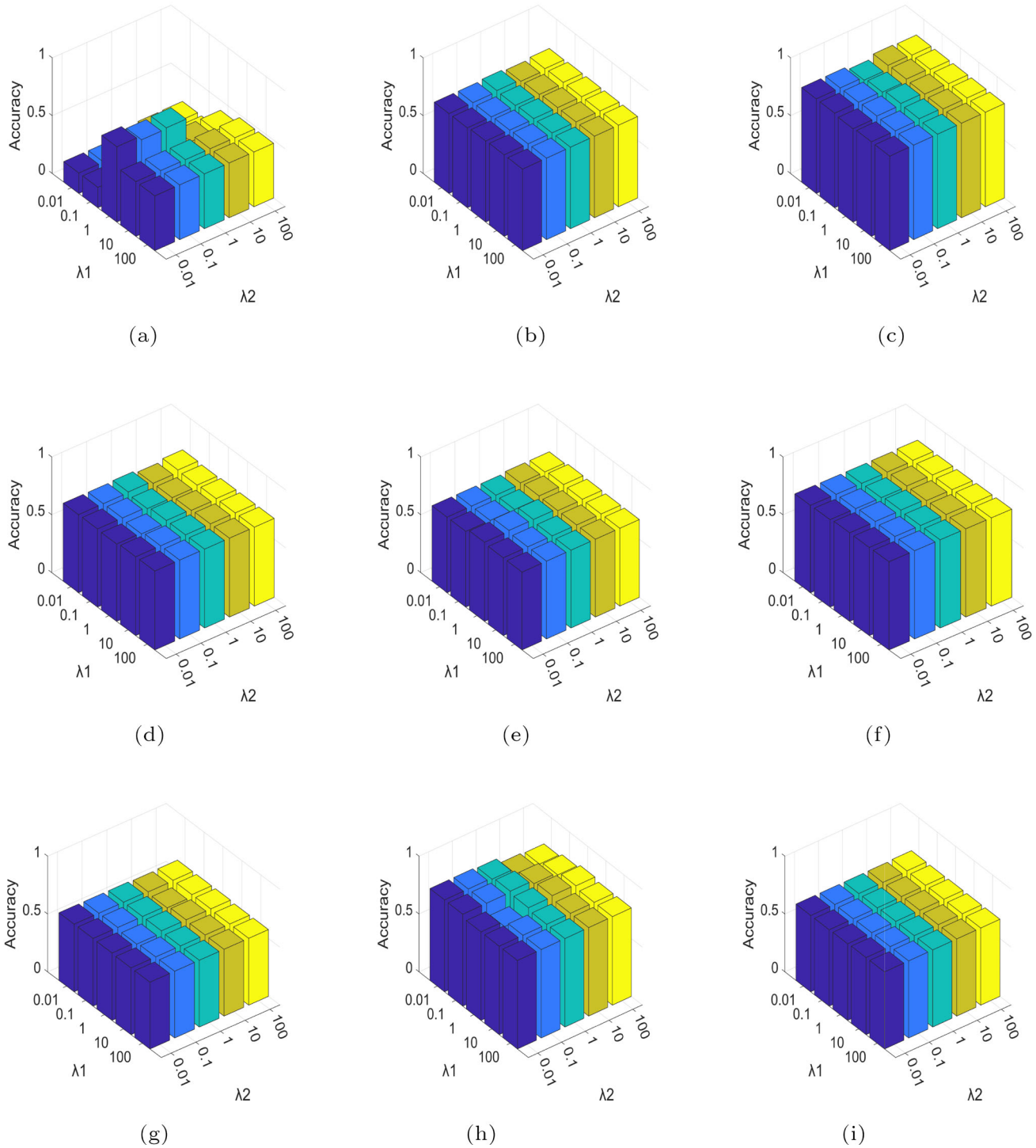

Fig. 3 Classification accuracy vs. both λ_1 and λ_2 on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

4.1.4 Number of constraints

Now, we analyze the impact of the number of pairwise constraints on the classification accuracy of CFW and keep the other experimental settings the same as before. The number of total pairwise constraints varies within the set $\{4, 20, 40, \cdots, 180, 200\}$, where the number of must-link constraints is the same as the number of cannot-link constraints.

Figure 4 plots curves showing the accuracy vs. number of constraints obtained by CFW on each of the nine datasets. Notably, the classification accuracy fluctuates with the number of constraints. At the beginning, the classification accuracy varies significantly when increasing the number of

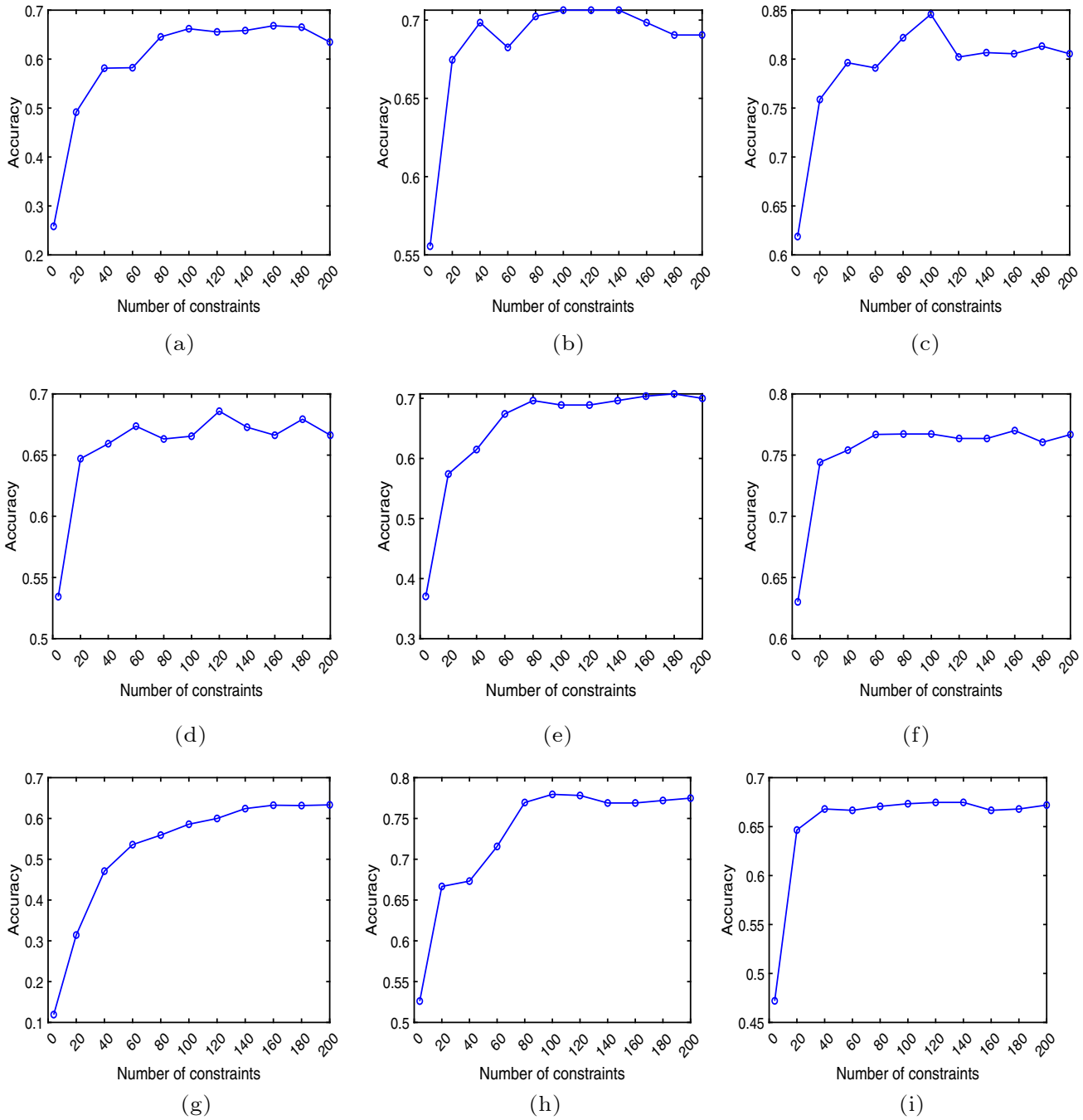

Fig. 4 Curves of classification accuracy vs. constraint number obtained by proposed methods on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

pairwise constraints, this satisfies our expectation that more constraints would induce better performance. However, when the number of pairwise constraints increases beyond a certain value, i.e., 80, the classification performance displays only minor fluctuations, which means that we cannot further improve the performance of the model by increasing the number of constraints. The main reason for this finding is the limitation imposed by supervised information. In the experiments, pairwise constraints are constructed from a limited training set that provides limited supervised information. The small observed classification performance fluctuations may be due to the randomness of constructing constraints.

4.2 Comparison under constraint-guided learning framework

4.2.1 Experimental setting

A 3-fold cross-validation method [34] is employed on the nine datasets. Specifically, each dataset is randomly divided into three subsets, where two subsets are used for training and the third subset is employed for testing. Thus, 3 trials are implemented in the 3-fold cross-validation process. The average results of ten 3-fold cross-validation experiments, totally 30 trials, are reported. In each trial, we construct 50 cannot-link and 50 must-link constraints from the training set.

Six CS-based feature selection methods are compared with CFW, including CS [20], LSDF [16], CLS [21], Relief-SC [22], SCS [17], and HM-ICS [23]. In addition, a part of CFW is related to Relief-SC, and this component is called CFW-SC. In detail, the objective function of CFW-SC contains the first two terms in (24) and can be also solved by the gradient descent method. Here, CFW-SC is included in our list of comparison methods.

The parameter settings of the compared supervised and semi-supervised feature selection algorithms are all derived from their corresponding references. Note that the supervised learning algorithms, e.g., the CS, handle only pairwise constraints. The semi-supervised learning algorithms utilize not only the constraints but also the set of unlabeled training samples. In both CFW and CFW-CS, we set $k = 5$, $\eta = 0.01$, $\theta = 0.001$, $T = 100$, and $\lambda_1 = 1$. Additionally, let $\lambda_2 = 1$ for CFW. Because the compared methods cannot determine the optimal number of features, we assume that the number of optimal features varies within the set $\{20, 40, \cdots, 200\}$.

4.2.2 Experimental results

Figure 5 presents comparisons among the results produced by the different feature selection algorithms on the nine datasets under the constraint-guided learning framework. We first analyze the experimental results as a whole. Observation on Fig. 5 indicates that CFW achieves better classification performance than that of the other compared methods. The curves depicted in Fig. 5(b), (d), (f), and (g) clearly indicate that CFW consistently surpasses the other methods in terms of all 10 feature numbers across the CNS, Glioma, Novartis, and ORL datasets. On the CNAE-9, Colon, and Normal datasets, it is worth noting that CFW achieves the highest classification accuracies, but it does not exhibit superiority with respect to all 10 feature numbers. Next, we analyze CFW, CFW-SC, and Relief-SC. As a variant of Relief-SC, CFW-SC achieves higher accuracies than Relief-SC on most datasets, as demonstrated in Fig. 5(b), (c), (d), (e), (g), and (i). By incorporating the must-link constraints into CFW-SC, CFW can obtain more supervised information and then achieve better performance.

According to Fig. 5, we summarize the highest average accuracies and the corresponding standard deviations produced by the compared methods in Table 5, where the bold values represent the best results among the compared methods, and the numbers in brackets represent the optimal number of features. It can be seen that CFW-SC performs much better than Relief-SC on all datasets except CNAE-9 and Novartis, which validates the efficiency of CFW-SC in terms of improving Relief-SC by enriching the neighborhood information. Additionally, CFW is superior to CFW-SC on all datasets, which suggests that it is necessary to introduce the supervised information provided by must-link constraints. As evident from the results presented in Table 5, CFW consistently achieves the best performance across all datasets. For example, on the CNAE-9 dataset, CFW achieves a 1.92% higher accuracy rate than that of HM-ICS (the second best method) and a 2.58% improvement over Relief-SC (the third best method). These findings demonstrate the superiority of CFW with respect to selecting discriminative features in comparison with other methods.

The superiority of CFW can be ascribed to two key factors. First, the innovative hypothesis margin calculation formula developed for CFW enhances the discriminative power of the selected features. Second, the incorporation of the must-link preserving regularization term ensures that the chosen features effectively preserve the information embedded within must-link constraints.

4.3 Comparison under label-guided learning framework

4.3.1 Experimental setup

Under the label-guided learning framework, we can construct pairwise constraints from the given labeled samples. Thus, we compare our CFW and CFW-SC approaches with some feature selection methods under a label-guided learning framework. The compared methods are described as follows.

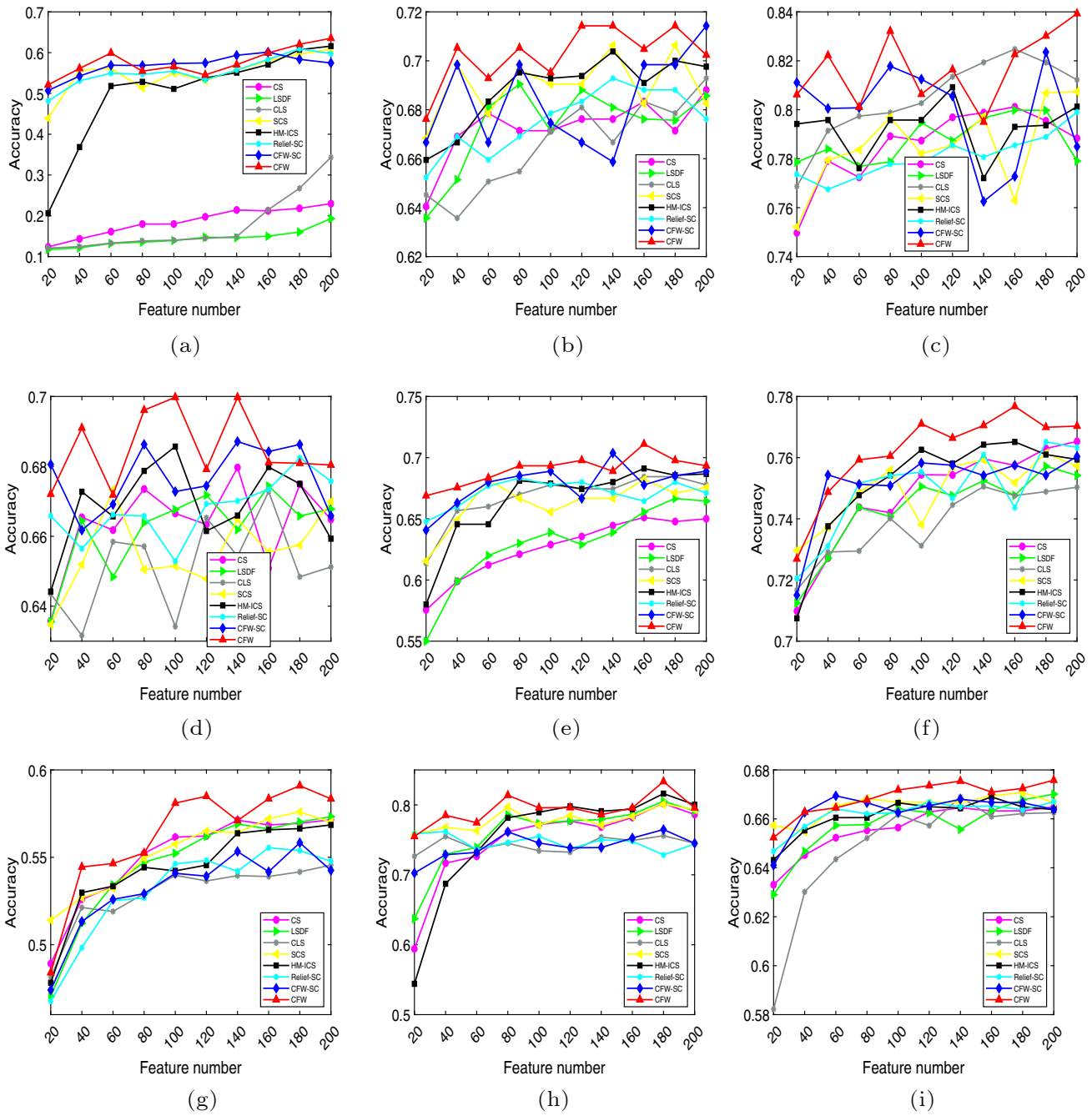

Fig. 5 Curve of classification accuracy vs. the number of selected features obtained by eight methods under the constraint-guided learning framework on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

- **LPLIR** [16]. A local preserving logistic I-Relief (LPLIR) algorithm is a semi-supervised feature selection method that aims to maximize the expected margin of the given labeled data and retain the local structural information of all data.
- **S2LFS** [7]. A semi-supervised local feature selection (S2LFS) method selects different discriminative feature subsets to represent samples from different classes.
- **ASLCGLFS** [35]. A semi-supervised feature selection via adaptive structure learning and constrained graph learning (ASLCGLFS) algorithm introduces adaptive structure learning and graph learning to select features.
- **SFS-AGGL** [36]. A semi-supervised feature selection method based on an adaptive graph with global and local information (SFS-AGGL) effectively leverages the structural distribution information from labeled data to derive label information for unlabeled samples.

Table 5 Mean accuracy and standard deviation of compared methods under the constraint-guided learning framework on nine datasets

Dataset	CFW	CFW-CS	CS	LSDF
CNAE-9	**63.52**±5.69(200)	60.12±2.04(160)	22.97±1.56(200)	19.31±2.77(200)
CNS	**71.43**±12.40(120)	**71.43**±1.37(200)	68.81±2.37(10)	69.05±1.87(80)
Colon	**83.93**±5.44(200)	82.35±7.39(180)	80.12±4.12(160)	79.98±4.89(160)
Glioma	**69.98**±9.21(140)	68.71±1.10(140)	67.97±1.51(140)	67.44±1.93(160)
Normal	**71.11**±5.30(160)	70.37±1.70(140)	65.11±3.06(160)	66.67±1.81(180)
Novartis	**77.68**±6.62(160)	76.04±1.11(200)	76.53±1.10(200)	75.72±1.29(180)
ORL	**59.10**±2.96(180)	55.83±2.44(180)	57.15±1.84(200)	57.32±1.66(200)
Prostate-GE	**83.33**±4.49(180)	76.47±4.27(180)	80.20±4.88(180)	80.51±7.70(180)
Sj-leukemia	**67.58**±4.55(200)	66.94±0.42(60)	66.49±0.77(200)	67.01±0.60(200)
	CLS	Relief-SC	SCS	HM-ICS
CNAE-9	34.33±3.87(200)	60.94±1.79(180)	60.85±1.20(200)	61.60±2.89(200)
CNS	69.29±1.76(200)	69.29±2.08(140)	70.63±1.37(140)	70.38±2.26(140)
Colon	82.48±3.57(160)	79.90±2.76(200)	80.74±5.61(200)	80.92±3.98(120)
Glioma	67.28±2.39(160)	68.25±2.02(180)	67.33±4.08(60)	68.57±1.66(100)
Normal	68.44±1.67(160)	68.33±2.11(80)	68.22±1.86(160)	69.11±2.41(160)
Novartis	75.05±1.20(140)	76.52±1.11(180)	76.15±1.16(180)	76.51±1.46(160)
ORL	54.55±1.87(200)	55.55±1.48(160)	57.60±1.10(180)	56.85±1.74(200)
Prostate-GE	75.59±3.35(180)	76.18±4.48(40)	80.10±3.07(180)	81.61±4.87(180)
Sj-leukemia	66.91±0.82(140)	66.69±0.61(200)	67.08±0.22(180)	66.90±0.54(160)

*Numbers in parentheses are optimal feature numbers

As before, 3-fold cross-validation experiments are repeated ten times. We report the average results obtained across the 30 trials. In each trial, 40% of the training samples are treated as labeled data, and the remaining samples are taken as unlabeled data. For both CFW and CFW-CS, we use the labeled data to construct the must-link constraint set $\mathcal{M}$ and the cannot-link constraint set $\mathcal{C}$ separately. The number of selected features is also set within the range of [20, 200] with an interval of 20.

4.3.2 Outcome of experiments

We compare the methods described above under the label-guided learning framework. The curves demonstrating the accuracy vs. the number of features are shown in Figure 6. From Figure 6, we can see that CFW is better than the other methods.

Table 6 presents a summary of Figure 6, where the highest average accuracy of each method and the corresponding standard deviation are listed, the bold values are the best results obtained among the compared methods, and the numbers in parentheses represent the optimal feature numbers. Table 6 indicates that CFW is superior to the other methods on eight out of the nine datasets. For example, CFW achieves the highest accuracy of 77.34% on the Colon dataset, LPLIR yields the second best accuracy of 76.38%. Only on the ORL dataset did CFW fail to achieve the best results.

CFW can mostly stand out when compared to these label-guided methods due to its comprehensive utilization of both must-link and cannot-link constraints. By employing constraints, CFW gains deep insights into the constraint structure of the training data, enabling the identification of features that not only effectively differentiate between classes but also respect the intrinsic relationships indicated by the constraints.

4.4 Statistical tests

To conduct a thorough comparison, we perform the Friedman test [34] and the corresponding Bonferroni-Dunn test [37] on the experimental results described above. The Bonferroni-Dunn test results indicate significant differences between CFW and the other algorithms. The critical difference between any two methods is defined as follows:

$$CD_\alpha = q_\alpha \sqrt{\frac{l(l+1)}{6N}} \tag{31}$$

where α corresponds to the preset threshold value, l denotes the number of methods, N represents the number of datasets, and q_α is the critical value.

In this study, we set $\alpha = 0.1$ by following the guidelines outlined in Refs. [38, 39]. Therefore, the statistical tests are conducted at a confidence level of 90%. Referring to Ref. [37], we obtain a critical value of $q_\alpha = 2.241$. In

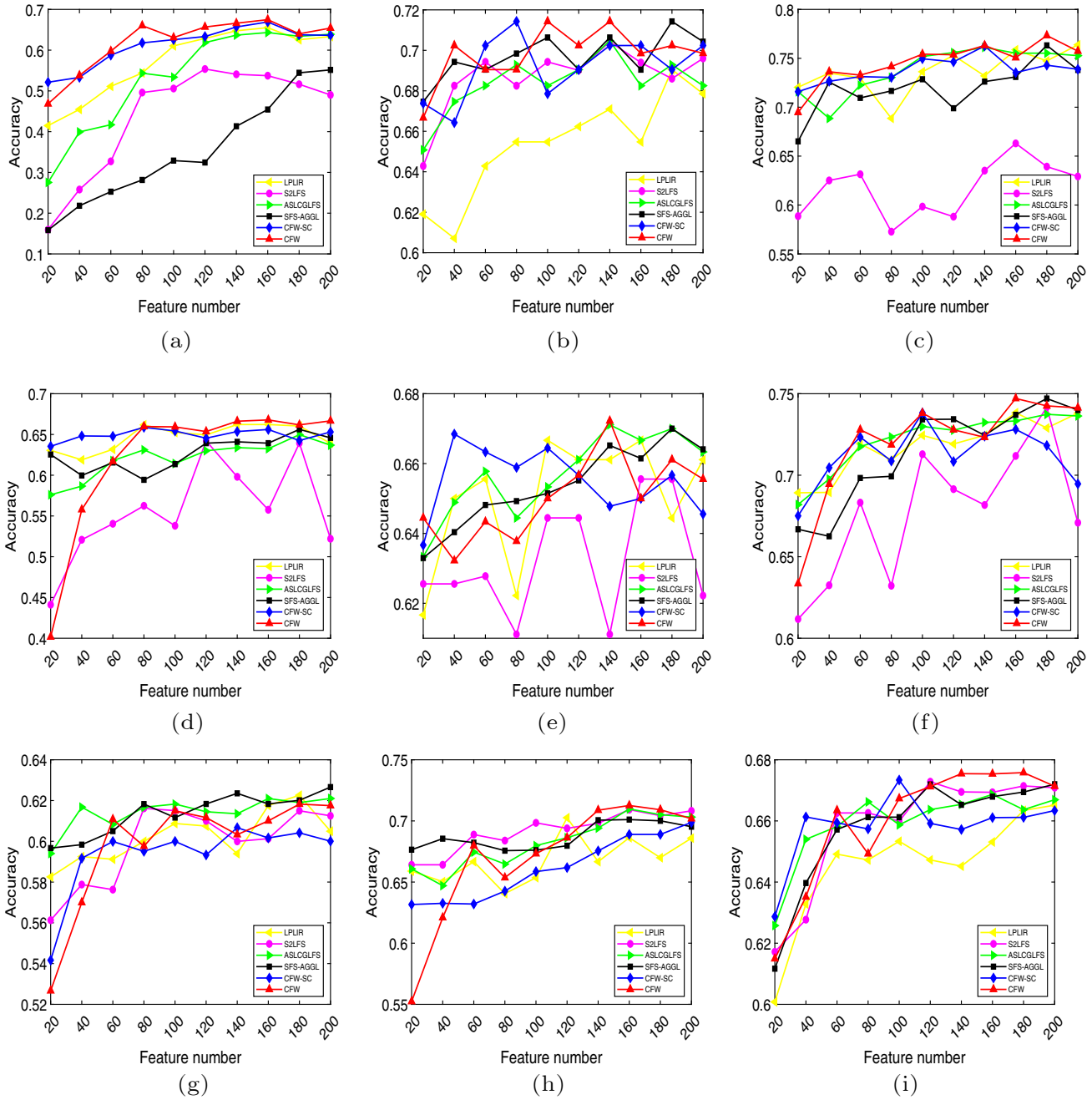

Fig. 6 Curve of classification accuracy vs. the number of selected features obtained by by six methods under the label-guided learning framework on nine datasets, (a) CNAE-9, (b) CNS, (c) Colon, (d) Glioma, (e) Normal, (f) Novartis, (g) ORL, (h) Prostate-GE, and (i) Sj-leukemia

the comparison experiments conducted under the constraint-guided learning framework, $l = 8$ and $N = 9$. Consequently, the critical difference is $CD_{0.1} = 2.59$. If the difference between two algorithms is greater than 2.59, then there is a significant distinction between them. In the comparison experiments conducted under the label-guided framework, $l = 6$ and $N = 9$. Then, the corresponding critical difference is $CD_{0.1} = 1.89$.

Table 7 displays the rank differences obtained through the Friedman test with the Bonferroni-Dunn test when comparing CFW with the other methods. All rank differences, which are represented by the values contained in the second row of Table 7, are found to be greater than the critical difference threshold of 2.59, which suggests that CFW is significantly superior to the seven compared methods. Similarly, the rank differences presented in the fourth row of Table 7 also imply

Table 6 Mean accuracy and standard deviation of compared methods under the label-guided learning framework on nine datasets

Dataset	CFW	CFW-SC	LPLIR
CNAE-9	**67.48** ± 5.45(160)	66.91 ± 1.72(160)	65.52 ± 4.06(160)
CNS	**71.43** ± 2.03(100)	**71.43** ± 1.15(80)	69.05 ± 1.68(180)
Colon	**77.34** ± 8.31(180)	76.26 ± 5.43 (140)	76.38 ± 4.05(200)
Glioma	**66.78** ± 7.81(160)	65.87 ± 2.62(80)	66.22 ± 1.29(140)
Normal	**67.23** ± 6.47(140)	66.84 ± 3.31(40)	66.67 ± 1.57(100)
Novartis	**74.71** ± 6.61(160)	73.82 ± 5.43(100)	73.83 ± 2.71(160)
ORL	61.82 ± 4.57(180)	60.67 ± 1.70(140)	62.26 ± 0.70(180)
Prostate-GE	**71.26** ± 3.10(160)	69.87 ± 1.13(200)	70.26 ± 8.82(120)
Sj-leukemia	**67.58** ± 7.72(180)	67.34 ± 1.73(100)	66.51 ± 0.54(200)
	S2LFS	ASLCGLFS	SFS-AGGL
CNAE-9	55.37 ± 12.09(120)	64.32 ± 9.92(160)	55.15 ± 2.12(200)
CNS	70.39 ± 12.37(140)	70.48 ± 13.18(140)	**71.43** ± 13.67(180)
Colon	66.30 ± 11.44(160)	76.11 ± 13.08(140)	76.32 ± 8.06(180)
Glioma	64.22 ± 10.50(120)	64.98 ± 13.70(180)	65.63 ± 9.39(180)
Normal	65.56 ± 11.71(180)	67.11 ± 8.93(140)	67.00 ± 6.94(180)
Novartis	74.17 ± 5.88(180)	73.73 ± 6.97(180)	74.50 ± 5.53(180)
ORL	61.62 ± 4.34(80)	62.10 ± 4.19(200)	**62.66** ± 5.56(200)
Prostate-GE	70.92 ± 6.38(160)	70.98 ± 7.30(160)	70.11 ± 14.71(160)
Sj-leukemia	67.28 ± 3.47(120)	66.87 ± 6.09(160)	67.20 ± 5.74(200)

*Numbers in parentheses are optimal feature numbers

that CFW performs significantly better than the other five methods. In short, CFW has excellent performance regardless of the employed learning framework.

5 Conclusions

This paper focuses on the task of semi-supervised feature selection under the constraint-guided learning framework and proposes a novel method called, CFW. The proposed CFW integrates the hypothesis margin concept and the constraint information provided by pairwise constraints. CFW first allocates probabilities to neighbor samples, thereby modifying the calculation formula of the hypothesis margin. The modified hypothesis margin term aims to identify features with significant discriminant capabilities. Subsequently, the L1 regularization term is incorporated into the model to guarantee the sparsity of CFW, thereby achieving the purpose of automatic feature selection. Moreover, CFW designs a must-link preserving regularization term, which is aimed at selecting features that have the ability to maintain must-link information.

A comprehensive series of experiments demonstrates the effectiveness of CFW. First, we assess the convergence, sparsity and discriminant ability, parameter sensitivity, and number of constraints of CFW. The experimental results show that CFW can converge quickly while exhibiting good sparsity and discriminant ability. Subsequently, CFW is compared with various supervised and semi-supervised methods on nine high-dimensional datasets under two learning frameworks. The findings show that CFW achieves good classification performance when using an NN as the classifier. Finally, to statistically compare the performance of CFW with that of other algorithms, the Friedman test is performed on the experimental results. The statistical test results suggest that CFW significantly outperforms the other compared algorithms.

However, our method is not without its limitations. The determination of the parameters, λ_1 and λ_2, requires careful tuning. Although we provide guidelines for the parameter

Table 7 Rank differences obtained by Friedman test with Bonferroni-Dunn test for comparing CFW and other methods

$CD_{0.1}$ (Constraint-guided)	CFW-SC	CS	LSDF	CLS	Relief-SC	SCS	HM-ICS
2.59	2.88	4.72	4.61	5.00	4.44	3.27	2.61
$CD_{0.1}$ (Label-guided)	CFW-SC	LPLIR	S2LFS	ASLCGLFS	SFS-AGGL		
1.89	2.34	2.45	3.34	2.67	1.90		

settings based on our experiments, the optimal settings may vary across different datasets and application scenarios. Though CFW exhibits robust performance across a variety of datasets, scaling this method to extremely large datasets is a challenge. The computational complexity of CFW significantly increases for datasets with vast numbers of samples and features. Future efforts will be dedicated to overcoming these challenges, with the aim of improving the scalability of CFW to efficiently accommodate larger datasets.

Appendices

Appendix A: Proof of Theorem 1

It is well known that a function defined on an open set is convex if and only if its Hessian matrix is positive semi-definite. Therefore, to prove the convexity of the function $J_R(\mathbf{w})$ in (19), we must demonstrate that its Hessian matrix is positive semi-definite.

Because the Laplacian matrix $\mathbf{L}^{\mathcal{M}}$ is symmetric and positive semi-definite, $\mathbf{Q} = \mathbf{F}^T\mathbf{L}^{\mathcal{M}}\mathbf{F}$ is also symmetric and positive semi-definite. Therefore, we can express the must-link preserving regularization as $J_R(\mathbf{w}) = \text{trace}\left(2\mathbf{M}^T\mathbf{QM}\right)$, as shown in (21).

Then, the first partial derivative of $J_R(\mathbf{w})$ with respect to w_r can be calculated as follows:

$$\frac{\partial J_R(\mathbf{w})}{\partial w_r} = 4w_r Q_{rr}, \quad r = 1, ..., d \tag{A1}$$

where Q_{rr} is the element in the r-th row and r-th column of $\mathbf{Q}$. The second partial derivative of $J_R(\mathbf{w})$ with respect to w_s can be expressed as

$$\frac{\partial^2 J_R(\mathbf{w})}{\partial w_r \partial w_s} = \begin{cases} 4Q_{rr}, & \text{if } r = s \\ 0, & \text{otherwise} \end{cases} \tag{A2}$$

Therefore, the Hessian matrix $\mathbf{H}$ of $J_R(\mathbf{w})$ is a diagonal matrix, where the diagonal elements are $H_{rr} = 4Q_{rr}$.

Since $\mathbf{Q}$ is positive semi-definite, we have that $Q_{rr} \geq 0$. As a result, the Hessian matrix $\mathbf{H}$ of $J_R(\mathbf{w})$ is positive semi-definite. In other words, $J_R(\mathbf{w})$ is a convex function of $\mathbf{w}$ when $\mathbf{w} \geq 0$. This concludes the proof.

Appendix B: Proof of Theorem 2

Let $J_1(\mathbf{w}) = \log\left(1 + \exp\left(-\mathbf{w}^T\mathbf{z}\right)\right)$ and $J_2(\mathbf{w}) = \lambda_1\|\mathbf{w}\|_1$, then the objective function (24) can be rewritten as:

$$J(\mathbf{w}) = J_1(\mathbf{w}) + \lambda_1 J_2(\mathbf{w}) + \lambda_2 J_R(\mathbf{w}) \tag{B3}$$

According to the properties of convex functions, $J(\mathbf{w})$ is a convex function if and only if $J_1(\mathbf{w})$, $J_2(\mathbf{w})$, and $J_R(\mathbf{w})$ are convex functions. Theorem 1 states that $J_R(\mathbf{w})$ is a convex function. Now, we need to prove that the other two functions are also convex. Following the approach used to prove Theorem 1, we simply need to demonstrate that the Hessian matrices of both $J_1(\mathbf{w})$ and $J_2(\mathbf{w})$ are positive semi-definite.

We start by calculating the first and second partial derivatives of $J_1(\mathbf{w})$ with respect to $\mathbf{w}$, as shown below

$$\frac{\partial J_1(\mathbf{w})}{\partial \mathbf{w}} = -\frac{\exp\left(-\mathbf{w}^T\mathbf{z}\right)}{1 + \exp\left(-\mathbf{w}^T\mathbf{z}\right)}\mathbf{z} \tag{B4}$$

and

$$\frac{\partial^2 J_1(\mathbf{w})}{\partial^2 \mathbf{w}} = \frac{\exp\left(-\mathbf{w}^T\mathbf{z}\right)}{\left(1 + \exp\left(-\mathbf{w}^T\mathbf{z}\right)\right)^2}\mathbf{z}\mathbf{z}^T \tag{B5}$$

Without loss of generality, let $c = \sqrt{\frac{\exp(-\mathbf{w}^T\mathbf{z})}{(1+\exp(-\mathbf{w}^T\mathbf{z}))^2}}$. Substituting c into (B5), we have

$$\frac{\partial^2 J_1(\mathbf{w})}{\partial^2 \mathbf{w}} = (c\mathbf{z})(c\mathbf{z})^T = \mathbf{H}_1 \tag{B6}$$

where $\mathbf{H}_1$ is the Hessian matrix of $J_1(\mathbf{w})$. Because $\mathbf{H}_1$ can be regarded as the outer product of a column vector $c\mathbf{z}$ and its own transpose vector $(c\mathbf{z})^T$. So $\mathbf{H}_1$ is a matrix of rank 1 with only one non-zero eigenvalue. It can be calculated that the non-zero eigenvalue of matrix $\mathbf{H}_1$ is $c^2\|\mathbf{z}\|^2$, which is greater than 0. In this case, the Hessian matrix of (B6) is positive semi-definite. Therefore, $J_1(\mathbf{w})$ is a convex function.

As for $J_2(\mathbf{w})$, we have

$$\frac{\partial J_2(\mathbf{w})}{\partial w_r} = 1, \quad r = 1, ..., d \tag{B7}$$

and

$$\frac{\partial^2 J_2(\mathbf{w})}{\partial w_r \partial w_s} = 0, \quad r, s = 1, ..., d. \tag{B8}$$

Thus, the Hessian matrix of $J_2(\mathbf{w})$ is a matrix with all zeros, which means that the Hessian matrix of $J_2(\mathbf{w})$ is positive semi-definite. Hence, $J_2(\mathbf{w})$ is a also convex function.

In summary, $J_1(\mathbf{w})$, $J_2(\mathbf{w})$ and $J_R(\mathbf{w})$ are convex functions. Thus, $J(\mathbf{w})$ is a convex function. This completes the proof.

Appendix C: Proof of Theorem 3

Note that $\mathbf{z}(t)$ is updated by $\mathbf{w}(t-1)$ in the t-th iteration. When $\mathbf{z}(t)$ and $\mathbf{w}(t-1)$ are given, $\mathbf{w}(t)$ is updated using the

gradient descent scheme in (27) and the truncation rule in (28). Consequently, the objective function achieves its minimum $J(\mathbf{w}(t) \mid \mathbf{z}(t))$ for fixed $\mathbf{z}(t)$. In other words,

$$J(\mathbf{w}(t) \mid \mathbf{z}(t)) \leq J(\mathbf{w}(t-1) \mid \mathbf{z}(t)) \tag{C9}$$

which completes the proof of Theorem 3.

Author Contributions Xinyi Chen: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft; Li Zhang: Writing- reviewing & editing, Supervision, Project administration; Lei Zhao: Supervision, Project administration; Xiaofang Zhang: Supervision, Project administration.

Funding This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 19KJA550002, by the Six Talent Peak Project of Jiangsu Province of China under Grant No. XYDXX-054, and by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Data availability and material Data is openly available in public repositories. http://archive.ics.uci.edu/ml/index.php;https://cam-orl.co.uk/facedatabase.html.

Declarations

Conflicts of interest We declare that there have been no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate All authors agreed to participate.

Consent for publication All authors have consented to publication.

References

1. Sheikhpour R, Sarram MA, Gharaghani S, Chahooki MAZ (2017) A survey on semi-supervised feature selection methods. Pattern Recogn 64:141–158
2. Bouchlaghem Y, Akhiat Y, Amjad S (2022) Feature selection: A review and comparative study. E3S Web of Conferences 351:01046
3. Pang Q, Zhang L (2020) Semi-supervised neighborhood discrimination index for feature selection. Knowl-Based Syst 204:106224
4. Chen H, Chen H, Li W, Li T, Luo C, Wan J (2022) Robust dual-graph regularized and minimum redundancy based on self-representation for semi-supervised feature selection. Neurocomputing 490:104–123
5. Jin L, Zhang L, Zhao L (2023) Max-difference maximization criterion: A feature selection method for text categorization. Front Comp Sci 17(1):171337
6. Jin L, Zhang L, Zhao L (2023) Feature selection based on absolute deviation factor for text classification. Inform Process Manag 60(3):103251
7. Li Z, Tang J (2021) Semi-supervised local feature selection for data classification. Inf Sci 64(9):192108
8. Pang Q, Zhang L (2021) A recursive feature retention method for semi-supervised feature selection. Int J Mach Learn Cybern 12(9):2639–2657
9. Tang B, Zhang L (2019) Multi-class semi-supervised logistic I-Relief feature selection based on nearest neighbor. In: Advances in knowledge discovery and data mining. pp 281–292
10. Tang B, Zhang L (2020) Local preserving logistic I-Relief for semi-supervised feature selection. Neurocomputing 399:48–64
11. Sun Y, Todorovic S, Goodison S (2009) Local-learning-based feature selection for high-dimensional data analysis. IEEE Trans Pattern Anal Mach Intell 32(9):1610–1626
12. Xu J, Tang B, He H, Man H (2016) Semi-supervised feature selection based on relevance and redundancy criteria. IEEE Trans Neural Netw Learn Syst 28(9):1974–1984
13. Wang C, Hu Q, Wang X, Chen D, Qian Y, Dong Z (2017) Feature selection based on neighborhood discrimination index. IEEE Trans Neural Netw Learn Syst 29(7):2986–2999
14. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. Adv Neural Inf Process Syst 18:507–514
15. Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
16. Zhao J, Lu K, He X (2008) Locality sensitive semi-supervised feature selection. Neurocomputing 71(10–12):1842–1849
17. Salmi A, Hammouche K, Macaire L (2020) Similarity-based constraint score for feature selection. Knowl-Based Syst 209:106429
18. Kalakech M, Biela P, Macaire L, Hamad D (2011) Constraint scores for semi-supervised feature selection: A comparative study. Pattern Recogn Lett 32(5):656–665
19. Hindawi M, Allab K, Benabdeslem K (2011) Constraint selection-based semi-supervised feature selection. In: 2011 IEEE 11th international conference on data mining. pp 1080–1085
20. Zhang D, Chen S, Zhou Z-H (2008) Constraint score: A new filter method for feature selection with pairwise constraints. Pattern Recogn 41(5):1440–1451
21. Benabdeslem K, Hindawi M: Constrained laplacian score for semi-supervised feature selection. In: Joint European conference on machine learning and knowledge discovery in databases. pp 204–218
22. Hijazi S, Kalakech M, Hamad D, Kalakech A (2018) Feature selection approach based on hypothesis-margin and pairwise constraints. In: 2018 IEEE Middle East and North Africa Communications Conference, pp 1–6
23. Chen X, Zhang L, Zhao L (2023) Iterative constraint score based on hypothesis margin for semi-supervised feature selection. Knowl-Based Syst 271:110577
24. Sun Y (2007) Iterative Relief for feature weighting: Algorithms, theories, and applications. IEEE Trans Pattern Anal Mach Intell 29(6):1035–1051
25. Asuncion A, Newman D (2013) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml
26. Pomeroy S, Tamayo P, Gaasenbeek M, Sturla L, Angelo M, McLaughlin M, Kim J, Goumnerova L, Black P, Lau C (2002) Gene expression-based classification and outcome prediction of central nervous system embryonal tumors. Nature 415(24):436–442
27. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci 96(12):6745–6750
28. Zhao Z, Zhang K-N, Wang Q, Li G, Zeng F, Zhang Y, Wu F, Chai R, Wang Z, Zhang C (2021) Chinese glioma genome atlas (CGGA): a comprehensive resource with functional genomic data from chinese glioma patients. Genom Proteom Bioinf 19(1):1–12
29. Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang C-H, Angelo M, Ladd C, Reich M, Latulippe E, Mesirov JP (2001) Multiclass cancer diagnosis using tumor gene expression signatures. Proc Natl Acad Sci 98(26):15149–15154

30. Su AI, Cooke MP, Ching KA, Hakak Y, Walker JR, Wiltshire T, Orth AP, Vega RG, Sapinoso LM, Moqrich A (2002) Large-scale analysis of the human and mouse transcriptomes. Proc Natl Acad Sci 99(7):4465–4470
31. Gross R (2005) Face databases. In: Handbook of face recognition. Springer, Pittsburgh, USA pp 301–327
32. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP (2002) Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1(2):203–209
33. Yeoh E-J, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell 1(2):133–143
34. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701
35. Lai J, Chen H, Li W, Li T, Wan J (2022) Semi-supervised feature selection via adaptive structure learning and constrained graph learning. Knowl-Based Syst 251:109243
36. Yi Y, Zhang H, Zhang N, Zhou W, Huang X, Xie G, Zheng C (2024) SFS-AGGL: Semi-supervised feature selection integrating adaptive graph with global and local information. Information 15(1):57
37. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64
38. Chen H, Tiňo P, Yao X (2009) Predictive ensemble pruning by expectation propagation. IEEE Trans Knowl Data Eng 21(7):999–1013
39. Huang X, Zhang L, Wang B, Li F, Zhang Z (2018) Feature clustering-based support vector machine recursive feature elimination for gene selection. Appl Intell 48(3):594–607

Xinyi Chen is currently pursuing the Master's degree at the School of Computer Science and Technology, Soochow University, Suzhou, China. Her research interests include machine learning and pattern recognition.

Li Zhang received the B.S. degree in 1997 and the Ph.D. degree in 2002 in electronic engineering from Xidian University, Xi'an, China. Now she is a full professor with the School of Computer Science and Technology, Soochow University, Suzhou, China. She was a postdoctor at the Institute of Automation, Shanghai Jiao Tong University, Shanghai, China, from 2003 to 2005. She worked as an associate professor at the Institute of Intelligent Information Processing, Xidian University, Xi'an, China, from 2005 to 2010. She was a visiting professor at Yuan Ze University, Taiwan, from February to May 2010. She has authored/co-authored more than 10 0 technical papers published in journals and conferences. Her research interests have been in the areas of machine learning, pattern recognition, neural networks and intelligent information processing.

Lei Zhao is a professor with the School of Computer Science and Technology at Soochow University. He received his Ph.D. degree in Computer Science from Soochow University in 2006. His research focuses on graph databases, social media analysis, knowledge graph, machine learning, etc. He has published over 130 research papers including more than 50 papers published in well-known journals and conferences such as TKDE, JCST, WWW, ICDE, CIKM, ICWS, and DASFAA.

Xiaofang Zhang is a professor with the School of Computer Science and Technology, Soochow University, China. Her research interests include the intersection of Software Engineering and Artificial Intelligence, including intelligent software engineering, software testing, and software maintainence.