Worst-Case Discriminative Feature Selection

Shuangli Liao[1], **Quanxue Gao**[1*], **Feiping Nie**[2†], **Yang Liu**[1] and **Xiangdong Zhang**[1]

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China
[2]School of Computer Science, OPTIMAL, Northwestern Polytechnical University, Xi'an, 710072, China
15829069358@163.com, qxgao@xidian.edu.cn, {feipingnie,liuyangxidian}@gmail.com, xdchen@mail.xidian.edu.cn

Abstract

Feature selection plays a critical role in data mining, driven by increasing feature dimensionality in target problems. In this paper, we propose a new criterion for discriminative feature selection, worst-case discriminative feature selection (WDFS). Unlike Fisher Score and other methods based on the discriminative criteria considering the overall (or average) separation of data, WDFS adopts a new perspective called worst-case view which arguably is more suitable for classification applications. Specifically, WDFS directly maximizes the ratio of the minimum of between-class variance of all class pairs over the maximum of within-class variance, and thus it duly considers the separation of all classes. Otherwise, we take a greedy strategy by finding one feature at a time, but it is very easy to implement and effective. Moreover, we utilize the correlation between features to help reduce the redundancy, and then WDFS is extended to uncorrelated WDFS (UWDFS). To evaluate the effectiveness of the proposed algorithm, we conduct classification experiments on many real data sets. In the experiment, we respectively use the original features and the score vectors of features over all class pairs to calculate the correlation coefficients, and analyze the experimental results in these two ways. Experimental results demonstrate the effectiveness of WDFS and UWDFS.

1 Introduction

Nowadays, the dimensionality of the data involved many real-world applications has increased explosively. The presence of many irrelevant and redundant features [Ben-Bassat, 1982] tends to make a learning model overfitting, resulting in its performance degenerates. Dimensionality reduction is one of the most popular techniques to help address this problem. Feature selection is a widely employed technique for reducing dimensionality among practitioners. It aims to select a small number of significant and useful original features without any transformation. Thus, feature selection could maintain the original representation of variables, which leads to better readability and interpretability. Many efforts have been devoted to the research in feature selection during the past few years [Romero and Sopena, 2008] [Pena and Nilsson, 2010].

According to how the learning algorithm is integrated into the process of evaluating and selecting features, feature selection methods can be roughly categorized as wrapper methods [Kohavi and John, 1997] [Wang *et al.*, 2008], embedded methods [Nie *et al.*, 2010] [Cai *et al.*, 2013] and filter methods. The wrapper methods and embedded methods are tightly coupled with the specific classifier, they always have better performance than filter methods on the particular classifier but not on the other classifiers. Different from wrapper and embedded methods, the filter-type methods, which evaluate features based on a certain criterion, are absolutely independent on any classifier. Most existing filter-type feature selection methods conducted the selection process in the manner of feature ranking [Robnik *et al.*, 2003] [Raileanu and Stoffel, 2004] or feature subset evaluation [Nie *et al.*, 2008]. Methods based on the feature ranking compute the relevance of a feature with respect to the class label distribution of data. In this paper, we propose a new filter-type feature selection method in the manner of feature ranking.

For the classification problem, feature selection aims to select subset of highly discriminant features. In other words, it selects features that are capable of discriminating samples that belong to different classes. Fisher discriminant criterion [Fisher, 1936] is probably the most widely used one. One of the most representative algorithms is Fisher score [He *et al.*, 2005], which optimizes the so-called Fisher criterion that maximizes the ratio of between-class scatter over within-class scatter in a individual feature level. On this basis, the Laplacian score [He *et al.*, 2005] also consider the feature ability of preserving the local manifold structure. Then Trace Ratio [Nie *et al.*, 2008] was proposed to improve the computational efficiency of Fisher score, which selects a feature subset based on the corresponding subset-level score that is calculated in a Trace Ratio form. Moreover, there are many feature selection method, combining the popular transformation-based dimensionality reduction method linear discriminant analysis (LDA) and sparsity regularization [Masaeli *et al.*, 2010]. For example, Discriminative Feature Selection (DF-

*Contact Author: Q. Gao. (qxgao@xidian.edu.cn)
†Contact Author: F. Nie. (feipingnie@gmail.com)

S) [Tao *et al.*, 2016] imposes the row sparsity on the transformation matrix of LDA through $\ell_{2,1}$-norm regularization to achieve feature selection. Zhang et al. proposed a self-weighted supervised discriminative feature selection (SSD-FS) method [Zhang *et al.*, 2018], which constrains the transformation matrix to be orthogonal and introduces the $\ell_{2,1}$-norm regularization to select features.

However, according to the definition of between-class scatter, the aforementioned methods actually maximize the average of all pairwise distances between classes [Bian and Tao, 2011] [Su *et al.*, 2018]. This may cause the so-called class separation problem [Loog *et al.*, 2001]. Specifically, these methods tend to pay close attention to classes with larger distances, but ignore those with smaller distances, resulting in the overlap of neighboring classes in the lower-dimensional space. An example to illustrate the class separation problem is shown in Figure 1, where class 1 and class 2 locate closely to each other while class 3 is far away from them, and all classes have the same covariance. We can directly see that the Feature 2 could easily separates three class, while the Feature 1 would result in a complete confusion of class 1 and class 2. But according to the muti-class Fisher certeria [Rao, 1948], the average of pairwise distances between classes is maximized in the Feature 1.

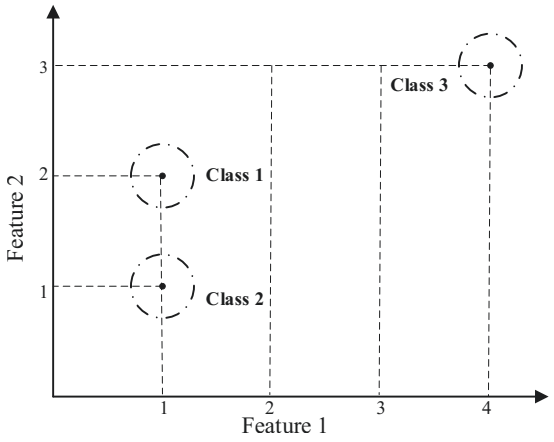


Figure 1: An illustration of the class separation problem.

Due to the fact that many feature selection methods based on the discriminative criteria usually consider the overall (or average) separation of data, thus they cannot guarantee the separation (as best as possible) of any class pairs. Thus, we propose a new criterion for discriminative feature selection, namely worst-case discriminative feature selection (WDFS), which adopt a worst-case view [Yu and Yeung, 2010] but the average view. Specifically, WDFS directly maximizes the ratio of the minimum of between-class variance of all class pairs over the maximum of within-class variance, and thus it duly considers the separation of all classes. For the solution, we take a greedy strategy by finding one feature at a time, but that is very easy to implement. Moreover, in order to reduce the redundancy of selected feature subset, we extend WDFS to uncorrelated WDFS (UWDFS) with the help of the correlation between features. Finally, in the experiment, we respectively use the original features and the score vectors of features over all class pairs to calculate the correlation coefficients, and analyze the experimental results in these two ways. **Our contribution are summarized as follows**:

1. Our method WDFS duly considers the separation of all classes, which adopts a new view called worst-case view different from the conventional average view.
2. Considering the feature evaluation mechanism of the WDFS model, we propose a method to help reduce the redundancy, which only needs to calculate the correlation coefficients between features or feature score vectors.
3. Although we take a greedy strategy by finding one feature at a time, that is very easy to implement with low computational complexity.

2 Related Works

2.1 Fisher Score

Given a set of N data points $\left\{\mathbf{x}_k \in \mathbb{R}^{d\times 1} | k = 1, \cdots, N\right\}$ which are sampled from C classes. The within-class matrix $\mathbf{S}_w$ and between-class matrix $\mathbf{S}_b$ are defined as

$$\mathbf{S}_b = \sum_{i=1}^{c} n_i \left(\mu_i - \mu\right)\left(\mu_i - \mu\right)^T \tag{1}$$

$$\mathbf{S}_w = \sum_{i=1}^{c}\sum_{p=1}^{n_i} \left(\mathbf{x}_p^i - \mu_i\right)\left(\mathbf{x}_p^i - \mu_i\right)^T \tag{2}$$

where n_i denotes the number of samples in the $i-th$ class, μ_i denotes the mean of the $i-th$ class, and $\mathbf{x}_p^i$ denotes the $p-th$ sample in the $i-th$ class.

To specify the selected features in the procedure of feature selection, we equip the conventional transportation matrix with an explainable structure to specify the selected features, namely selective matrix. Next, we will go into details. First, concatenate the dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to a matrix $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$. Then let ${f_r}^T \in \mathbb{R}^{1\times N}$ denotes the $r-th$ row of $\mathbf{X}$, i.e. the $r-th$ feature of the dataset, and then let $\mathcal{F} = \mathbf{X}^T = [f_1, f_2 \ldots, f_d] \in \mathbb{R}^{N\times d}$. Formally, for a feature subset $\mathcal{F}_\mathcal{I} \in \mathcal{F}$, we could define the corresponding selective matrix as

$$\mathbf{W}_\mathcal{I} = \left[\mathbf{w}_{\mathcal{I}(1)}, \mathbf{w}_{\mathcal{I}(2)}, \ldots, \mathbf{w}_{\mathcal{I}(m)}\right] \in \{0,1\}^{d\times m} \tag{3}$$

where $\mathbf{w}_{\mathcal{I}(k)} \in \mathbb{R}^{d\times 1}\,(k = 1, 2, \ldots, m \leq d)$ refers to a column vector whose components are all 0 expect 1 for the $\mathcal{I}(k) - th$ one. Note that this $\mathbf{W}_\mathcal{I}$ is indeed a column-full-rank transformation matrix. With the selective matrix $\mathbf{W}_\mathcal{I}$, the process of feature selection could be expressed as

$$\mathcal{F}_\mathcal{I} = \mathcal{F}\mathbf{W}_\mathcal{I}. \tag{4}$$

Then the model of Fisher Score could be formulated as

$$\max_{\mathbf{W}_\mathcal{I}\in\mathbb{R}^{d\times m}} \sum_{k=1}^{m} \frac{\mathbf{w}_{\mathcal{I}(k)}^T \mathbf{S}_b \mathbf{w}_{\mathcal{I}(k)}}{\mathbf{w}_{\mathcal{I}(k)}^T \mathbf{S}_w \mathbf{w}_{\mathcal{I}(k)}}. \tag{5}$$

It should be noted that each feature is evaluated individually. It is easy to see that the model in the Eq. (5) consider the overall (or average) separation of data, thus it cannot guarantee the separation (as best as possible) of any class pairs.

2.2 Redundancy

The Redundancy (feature−feature) analysis has been another concerned part of supervised filter models, which has many similarities to the Relevance (feature−class) in term of measurement. Yu and Liu [Yu and Liu, 2004] explored further feature redundancy, and more restrictively defined relevant features as any feature that is neither irrelevant nor redundant to the target concept. The independence or complementarity of one feature can be defined based on correlation coefficient, mutual information, or any criterion characterizing feature redundancy. Many models use Euclidean distance, Pearson correlation [Battiti, 1994], and information measures for redundancy analysis. Some feature selection methods seek to remove redundant features while others do not. A classical criterion for feature selection based on relevance and redundancy analysis is Max-Relevance and Min-Redundancy (mRMR) [Bian and Tao, 2011], which maximizes the correlation between features and categorical variables while minimizing the correlation between features by measuring the mutual information.

3 Approach

In order to guarantee the separation (as best as possible) of any class pairs, we propose a new criterion for discriminative feature selection, worst-case discriminative feature selection (WDFS) in the subsection 3.1. Unlike Fisher Score and other methods based on the discriminative criteria considering the overall (or average) separation of data, WDFS adopts a worst-case view which duly considers the separation of all classes. Specifically, WDFS directly maximizes the ratio of the minimum of between-class variance of all class pairs over the maximum of within-class variance of all classes.

Apart from the analysis of relevance (feature−class), the analysis of redundancy (feature−feature) is another crucial part of supervised filter models. This work simply uses the correlation coefficient to help reduce redundancy among features. Then WDFS is extended to uncorrelated WDFS (UWDFS) in the subsection 3.2.

3.1 Worst-Case Discriminative Feature Selection

In the Eq. (1), due to the fact that $\mu = \frac{1}{N}\sum_{i=1}^{c} n_i\mu_i$, thus the equation (1) would be equal to the following equation by simple derivation.

$$\mathbf{S}_b = \frac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c} n_i n_j \left(\mu_i - \mu_j\right)\left(\mu_i - \mu_j\right)^T. \tag{6}$$

Next, we define the between-class matrix $\mathbf{S}_b^{i,j}$ for the class pair (i, j) and the within-class matrix $\mathbf{S}_w^i$ as

$$\mathbf{S}_b^{i,j} = \left(\mu_i - \mu_j\right)\left(\mu_i - \mu_j\right)^T, 1 \le i < j \le c, \tag{7}$$

$$\mathbf{S}_w^i = \frac{1}{n_i}\sum_{p=1}^{n_i}\left(\mathbf{x}_p^i - \mu_i\right)\left(\mathbf{x}_p^i - \mu_i\right)^T, 1 \le i \le c. \tag{8}$$

where μ_i denotes the mean of the $i-th$ class, and denotes the sample in the class. Similarly, based on the Fisher discriminative criterion, we formulate the model of worst-case discriminative feature selection as

$$\max_{\mathbf{W}_{\mathcal{I}}\in\mathbb{R}^{d\times m}} \sum_{k=1}^{m} \frac{\min\limits_{1\le i<j\le c} \mathbf{w}_{\mathcal{I}(k)}^T \mathbf{S}_b^{i,j} \mathbf{w}_{\mathcal{I}(k)}}{\max\limits_{1\le i\le c} \mathbf{w}_{\mathcal{I}(k)}^T \mathbf{S}_w^{i} \mathbf{w}_{\mathcal{I}(k)}}. \tag{9}$$

It should be noted that each feature is evaluated individually. Thus, we could a greedy strategy by finding one feature at a time to solve this optimization problem. Define the score of the $r-th$ feature in $\mathcal{F}$ as

$$score(f_r) = \frac{\min\limits_{1\le i<j\le c} \mathbf{w}_r^T \mathbf{S}_b^{i,j} \mathbf{w}_r}{\max\limits_{1\le i\le c} \mathbf{w}_r^T \mathbf{S}_w^{i} \mathbf{w}_r}, \quad r = 1, 2, \ldots, d, \tag{10}$$

where $\mathbf{w}_r$ is a selective vector corresponding to the $r-th$ feature in $\mathcal{F}$. For clarity, **Algorithm 1** lists the pseudo code of solving the model WDFS.

Algorithm 1 WDFS

Input: Train dataset $\{(\mathbf{x}_i, y_i)\,|i = 1, 2, \ldots, N\}$, wherein $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ and $y_i \in \{1, 2, \cdots, C\}$, the selected number m.
Initialize: Concatenate the dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to a matrix $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$, and let $\mathcal{F} = \mathbf{X}^T = [f_1, f_2 \ldots, f_d]$.
Output: Selective matrix $\mathbf{W}_{\mathcal{I}} \in \mathbb{R}^{d\times m}$.

1: Calculate the within-class matrix $\mathbf{S}_w^i$ for all class by the Eq. (7), and between-class matrix $\mathbf{S}_b^{i,j}$ for all class pair $(\mathrm{i}, \mathrm{j})\,, 1 \le i < j \le C$ by the Eq. (8).
2: For each feature, calculate the score $score(f_r)$,$r = 1, 2, \ldots, d$ by the Eq. (10).
3: Sort the scores of all features.
4: Select the m feature indexes corresponding to the first m largest feature scores and construct the corresponding selective matrix $\mathbf{W}_{\mathcal{I}}$.

3.2 Uncorrelated WDFS

In this subsection, we extend WDFS to Uncorrelated WDFS (UWDFS) by calculating correlation coefficients between features to help reduce the redundancy. For a feature pair (f_i, f_j), we define the correlation coefficient as

$$0 \le \sigma\left(f_i, f_j\right) \le 1, \tag{11}$$

where $\sigma(\bullet)$ is a scalar function, which calculate the similarity (correlation) between two vectors. It is worth noting that the larger correlation coefficient means a greater redundancy between features.

Moreover, according to the aforementioned analysis, we have that WDFS evaluates feature individually. Then we calculate the correlation coefficient between each feature of the rest features (current candidate feature subset except for the feature with the largest score) and the feature with the largest score. These correlation coefficients can further be used to calculate the weights, indicating the importance difference among the rest features.

Specifically, suppose the current selected feature subset be $\mathcal{F}_k = \left\{f_{\mathcal{I}(1)}, f_{\mathcal{I}(2)}, \cdots, f_{\mathcal{I}(k)}\right\} \subset \mathcal{F}$, and its complement, namely the candidate feature set, could be defined as

$$\begin{aligned}\mathcal{F}_k^c &= \mathcal{F} - \left\{f_{\mathcal{I}(1)}, f_{\mathcal{I}(2)}, \cdots, f_{\mathcal{I}(k)}\right\} \\ &= \left\{\tilde{f}_1, \tilde{f}_2, \cdots, \tilde{f}_{d-k}\right\}.\end{aligned} \tag{12}$$

Algorithm 2 UWDFS

Input: Train dataset $\{(\mathbf{x}_i, y_i)\,|i = 1, 2, \ldots, N\}$, wherein $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ and $y_i \in \{1, 2, \cdots, C\}$, the selected number m.
Initialize: Concatenate the dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to a matrix $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$, and let $\mathcal{F} = \mathbf{X}^T = [f_1, f_2 \ldots, f_d]$.
Output: Selective matrix $\mathbf{W}_{\mathcal{I}} \in \mathbb{R}^{d\times m}$.

1: Calculate the score vector $\mathbf{s}^{(0)}$ by the Eq. (10).
2: **for** $k = 1$ **to** m **do**
3: *Find* the feature $f_{I(k)} = f_{\max}$ and remove it from the current candidate feature subset $\mathcal{F}_k^c$, then obtain the new candidate feature subset $\mathcal{F}_{k+1}^c$.
4: *Calculate* the correlation coefficient vector
5: $\rho = [\rho_1, \rho_2, \cdots, \rho_{d-k-1}]$ between $\mathcal{F}_{k+1}^c$ and current $f_{\max}$ by the Eq. (11).
6: *Update* the score vector by the Eq. (16), then $\mathbf{s}^{(k)} = \left[s_1^{(k)}, s_2^{(k)}, \cdots, s_{d-k-1}^{(k)}\right]$
7: **end for**
8: Construct the selective matrix $\mathbf{W}_{\mathcal{I}}$ corresponding to the selected feature subset $\left\{f_{\mathcal{I}(1)}, f_{\mathcal{I}(2)}, \cdots, f_{\mathcal{I}(m)}\right\}$.

Here, we take the process of selecting the first two features as an example. The main steps are summarized as follows.
Step 1 : When $k = 0$, i.e. $\mathcal{F}_0 = \emptyset$, we calculate the score for $\tilde{f}_r\,(r = 1, \cdots, d)$ by the Eq. (10), then we get a score vector

$$\mathbf{s}^{(0)} = \left[s_1^{(0)}, s_2^{(0)}, \cdots, s_d^{(0)}\right], \tag{13}$$

where $s_r\,(r = 1, \cdots, d)$ is a scalar. Then we find the feature $f_{\max}$ with the largest score as

$$f_{\max} = f_{\mathcal{I}(1)} \leftrightarrow \max\left\{s_1^{(0)}, s_2^{(0)}, \cdots, s_d^{(0)}\right\}. \tag{14}$$

Then we obtain $\mathcal{F}_1^c = \mathcal{F} - \left\{f_{\mathcal{I}(1)}\right\} = \left\{\tilde{f}_1, \tilde{f}_2, \cdots, \tilde{f}_{d-1}\right\}$.
Step 2 : We calculate the correlation coefficient between $\tilde{f}_r\,(r = 1, \cdots, d-1)$ and $f_{\max} = f_{\mathcal{I}(1)}$ by the Eq. (11). Then we get the correlation coefficient vector

$$\rho = [\rho_1, \rho_2, \cdots, \rho_{d-1}]. \tag{15}$$

Then we can use the $1 - \rho_r\,(r = 1, \cdots, d-1)$ as the weight of $\tilde{f}_r$. This is consistent with the assumption that a less redundant feature should be assigned a larger weight, while a more redundant feature should be assigned a smaller weight. Finally, we update the score of $\tilde{f}_r\,(r = 1, \cdots, d-1)$ as

$$s_r^{(1)} = (1 - \rho_r) \times s_r^{(0)}, (r = 1, \cdots, d-1). \tag{16}$$

Then the new score vector is $\mathbf{s}^{(1)} = \left[s_1^{(1)}, s_2^{(1)}, \cdots, s_{d-1}^{(1)}\right]$ and the new $f_{\max}$ is

$$f_{\max} = f_{I(2)} \leftrightarrow \max\left\{s_1^{(1)}, s_2^{(1)}, \cdots, s_{d-1}^{(1)}\right\}. \tag{17}$$

So far, we have described a complete operation of selecting a feature in UWDFS. For clarity, **Algorithm 2** lists the pseudo code of solving the model UWDFS.

Dataset	Instances	Features	Classes	Type
COIL20	1440	1024	20	Object Image
USPS	7000	256	10	Hand Written Image
ORL	400	1024	40	Face Image
UMIST	575	1024	20	Face Image
LUNG	203	3312	5	Biological

Table 2: Details of the selected benchmark data sets

4 Experiments

In this section, we will validate our proposed methods compared with other state-of-the-art methods on five datasets.

4.1 Datasets Descriptions

These datasets include one object image dataset, COIL20, USPS, two face image datasets ORL[1] and UMIST, and one biological gene expression microarray dataset, lung cancer (LUNG). (The COIL20, USPS and LUNG datasets are download from the Internet[2]). We summarize the statistics of the data sets in Table 2 and briefly introduce them as follows.

(1) COIL20 contains 1440 images of 20 objects. The images of each object were taken 5 degrees apart as the object is rotated on a turntable, and each object has 72 images.

(2) The original USPS handwritten digit database contains 9298 images. In this paper, we select a balanced random sample of the original data set.

(3) ORL consists of 400 face images. There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times with varying lighting, different facial expressions, and facial details.

(4) UMIST contains 20 persons and totally 575 images. The number of data in each class is between 10 and 24.

(5) LUNG is composed of 203 samples in five classes with 139, 21, 20, 6, and 17 samples, respectively. Each sample has 12600 genes. The genes with standard deviations smaller than 50 expression units were removed and the remaining data set contains 203 samples with 3312 genes.

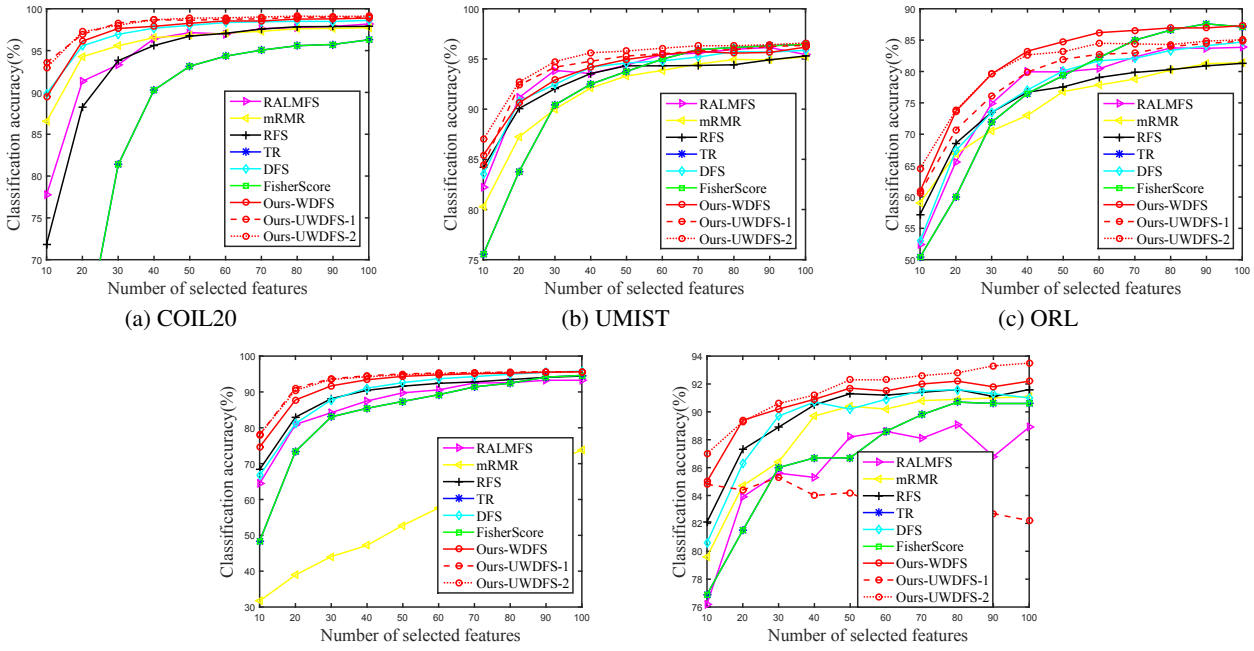
4.2 Reducing Feature Redundancy

A redundant feature does not add anything new to the target concept [Dash and Liu, 1997]. Thus, removing redundancy is straightforward when we require the number of selected features is as small as possible. Actually, we want the preserved features to be informative and complemental to each other. Considering that our proposed method selects one feature at a time, so we simply uses the correlation criteria to help reduce redundancy among features. The Pearson correlation coefficient [Battiti, 1994] is one of the simplest criteria. Moreover, in the experiment, we also try to use the score vectors over all class pairs instead of original feature as the input for calculating the correlation coefficients. The corresponding calculation formulas are as follows.

[1]http://www.zjucadcg.cn/dengcai/Data/FaceData.html
[2]http://featureselection.asu.edu/datasets.php

Dataset	Average accuracy(%) of top 20 features					Average accuracy(%) of top 40 features				
Methods	COIL20	USPS	ORL	UMIST	LUNG	COIL20	USPS	ORL	UMIST	LUNG
RALMFS	91.33 ±3.97	80.93 ±4.02	65.60 ±6.04	91.16 ±2.57	83.90 ±5.47	96.42 ±0.88	87.47 ±2.65	79.95 ±2.58	93.54 ±1.84	85.30 ±2.83
mRMR	94.26 ±1.67	38.94 ±2.12	66.85 ±4.31	87.23 ±3.62	84.70 ±2.06	96.56 ±0.95	47.19 ±1.78	73.00 ±2.47	92.11 ±1.89	89.70 ±3.02
RFS	88.25 ±5.17	82.95 ±3.06	68.55 ±2.01	90.07 ±2.99	87.30 ±4.64	95.61 ±2.70	90.45 ±1.92	76.70 ±3.28	93.54 ±1.60	90.50 ±2.32
TR	58.83 ±15.37	73.38 ±1.00	60.05 ±3.26	83.79 ±4.88	81.50 ±4.20	90.28 ±1.95	85.44 ±0.83	76.50 ±3.73	92.49 ±2.21	86.70 ±2.41
DFS	95.57 ±1.47	81.19 ±5.00	67.40 ±3.65	90.81 ±3.62	86.30 ±5.36	97.68 ±0.62	91.07 ±1.88	77.05 ±3.17	94.18 ±1.81	90.70 ±4.65
FisherScore	58.83 ±15.37	73.38 ±1.00	60.05 ±3.26	83.79 ±4.88	81.50 ±4.20	90.28 ±1.95	85.44 ±0.83	76.50 ±3.73	92.49 ±2.21	86.70 ±2.41
Ours-WDFS	96.13 ±1.50	87.75 ±1.63	73.65 ±2.37	90.63 ±3.38	**89.40 ±3.95**	97.90 ±1.02	93.40 ±0.90	**83.25 ±3.44**	94.18 ±2.01	90.90 ±2.60
Ours-UWDFS-1	96.96 ±0.54	**91.09 ±1.19**	70.70 ±4.87	92.39 ±1.53	84.40 ±3.17	**98.68 ±0.44**	**94.52 ±0.53**	79.90 ±3.59	94.77 ±1.80	84.00 ±3.33
Ours-UWDFS-2	**97.25 ±0.94**	90.47 ±1.24	**73.80 ±4.84**	**92.70 ±1.55**	89.30 ±2.67	**98.68 ±0.54**	94.24 ±0.43	82.65 ±3.49	**95.61 ±1.66**	**91.20 ±2.35**

Table 1: The average classification accuracy(%) and corresponding standard deviation under the selected Top 20 and 40 features.



(a) COIL20 (b) UMIST (c) ORL

(d) USPS (e) LUNG

Figure 2: Average classification accuracy *vs*. Number of selected features of different methods on five datasets.

(1) Pearson correlation coefficient

$$\sigma(\mathbf{a},\mathbf{b}) = \frac{N\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{N\sum {a_i}^2 - \left(\sum a_i\right)^2}\sqrt{N\sum {b_i}^2 - \left(\sum b_i\right)^2}}, \tag{18}$$

where $\mathbf{a}$ and $\mathbf{b}$ are two vectors.

(2) The score vector over all class pairs

For the feature f_r, similar to its score in the Eq. (10), we could define the corresponding score vector $\mathbf{v}_r \in \mathbb{R}^{\frac{C(C-1)}{2}}$ over all class pairs as

$$\mathbf{v}_r = \left[\frac{\mathbf{w}_r^T\mathbf{S}_b^{1,2}\mathbf{w}_r}{\max\limits_{1\le i\le c}\mathbf{w}_r^T\mathbf{S}_w^i\mathbf{w}_r}, \frac{\mathbf{w}_r^T\mathbf{S}_b^{1,3}\mathbf{w}_r}{\max\limits_{1\le i\le c}\mathbf{w}_r^T\mathbf{S}_w^i\mathbf{w}_r}, \cdots, \frac{\mathbf{w}_r^T\mathbf{S}_b^{C-1,C}\mathbf{w}_r}{\max\limits_{1\le i\le c}\mathbf{w}_r^T\mathbf{S}_w^i\mathbf{w}_r}\right] \tag{19}$$

Dataset	Average accuracy(%) of top 60 features					Average accuracy(%) of top 80 features				
Methods	COIL20	USPS	ORL	UMIST	LUNG	COIL20	USPS	ORL	UMIST	LUNG
RALMFS	96.97 ±0.92	90.57 ±1.98	80.45 ±3.30	95.44 ±0.95	88.60 ±3.20	97.75 ±0.64	92.58 ±1.34	83.75 ±3.16	95.96 ±1.22	89.10 ±3.14
mRMR	97.10 ±0.76	57.63 ±1.88	77.85 ±2.15	93.86 ±1.74	90.20 ±1.75	97.61 ±0.57	66.35 ±1.04	80.20 ±3.55	94.95 ±1.54	90.90 ±1.85
RFS	97.07 ±1.10	92.40 ±1.31	79.05 ±3.28	94.32 ±1.76	91.20 ±1.69	97.86 ±0.81	93.45 ±1.02	80.30 ±3.34	94.42 ±1.78	91.60 ±2.12
TR	94.33 ±0.88	89.22 ±0.79	82.20 ±2.66	94.98 ±2.40	88.60 ±3.17	95.58 ±0.63	92.45 ±0.80	86.60 ±1.43	96.14 ±1.96	90.70 ±2.50
DFS	98.36 ±0.72	93.69 ±0.87	81.70 ±2.55	94.81 ±1.89	90.90 ±3.33	98.50 ±0.81	94.95 ±0.52	83.35 ±2.31	95.72 ±1.77	91.60 ±3.50
FisherScore	94.33 ±0.88	89.22 ±0.79	82.20 ±2.66	94.98 ±2.40	88.60 ±3.17	95.58 ±0.63	92.45 ±0.80	86.60 ±1.43	96.14 ±1.96	90.70 ±2.50
Ours-WDFS	98.54 ±0.90	94.76 ±0.84	**86.20** ±**2.36**	95.44 ±1.34	91.50 ±3.27	98.76 ±0.81	95.26 ±0.60	**86.95** ±**3.62**	95.58 ±1.27	92.20 ±2.04
Ours-UWDFS-1	98.75 ±0.57	**95.28** ±**0.36**	82.75 ±3.16	95.51 ±1.26	83.00 ±2.31	98.99 ±0.40	**95.55** ±**0.43**	84.00 ±3.90	95.93 ±1.26	82.60 ±2.72
Ours-UWDFS-2	**98.92** ±**0.28**	95.11 ±0.32	84.50 ±3.70	**96.07** ±**1.70**	**92.30** ±**2.63**	**99.13** ±**0.36**	95.42 ±0.22	84.30 ±3.50	**96.32** ±**1.35**	**92.80** ±**2.20**

Table 3: The average classification accuracy(%) and corresponding standard deviation under the selected Top 60 and 80 features.

4.3 Comparison Algorithms and Parameter Setting

In our experiments, we compare our methods (WDFS and UWDFS) with other five methods, including RALMFS [Cai *et al.*, 2013], mRMR [Peng *et al.*, 2005], RFS [Nie *et al.*, 2010], DFS [Tao *et al.*, 2016], TR[Nie *et al.*, 2008] and FisherScore [He *et al.*, 2005]. (The code of mRMR, TR and FisherScore are downloaded from the ASU feature selection repository[3] and the code of DFS is implemented by ourselves in Python2.7.) Among them, RALMFS and RFS algorithms are embedded methods based on the linear regression model. The other methods belong to filter-type methods and except mRMR are all based on the Fisher discriminant criterion. Moreover, we respectively use the original features and the score vectors of features over all class pairs to calculate the correlation coefficients for UWDFS. For clarity, we call them UWDFS-1 and UWDFS-1 respectively.

We randomly divide each dataset into two parts that are approximately equal, one for training and the other for testing, and we repeat each group experiment for ten times. For the sake of simplicity and justice, we use the 1NN calssifier for classification and Euclidean distance as the metric and let the number of selected features be between 10 and 100 with an interval of 10.(There is no ideal method for choosing the dimension of the feature space.) In particular, the trade-off parameters in DFS and RFS, the trade-off parameters in DFS and RFS are set α from $[1e-6, 1e-4, 1e-3, 1e-2, 0.1, 1, 10, 100, 1e3, 1e4]$.

4.4 Results Analysis

Figure 2 shows the average classification accuracy vs. the number of selected features on six datasets with five different methods. Tables 1 and 3 list the average classification accuracy and the corresponding standard deviation, using the top 20, 40, 60 and 80 features respectively. As can be seen in the experimental results, we have that

1. As shown in Figure 2, with the increase in the number of selected features, the trends of average classification accuracy of different methods almost have a steady rise on different datasets. With the specific values in Tables1 and 3, we can see that our methods have a better classification performance with minor exceptions. Particularly, our methods trumps all the other algorithms at the small number of features with a noticeable rise.
2. On the ORL dataset, WDFS has better classification results than the other methods in most feature numbers. Our Uncorrelated WDFS is almost inferior to the WDFS algorithm. This may be an indication that the decorrelation process is working in a bad way, resulting in removing some weakly relevant and redundant features.
3. Comparing the model UWDFS-1 with UWDFS-2, we can see that UWDFS-2 always has better performance than UWDFS-1. Especially on the LUNG dataset, UWDFS-2 shows a steady rise but UWDFS-1 has a completely opposite trend.

5 Conclusions

In this paper, we propose a new criterion for discriminative feature selection, worst-case discriminative feature selection (WDFS). Based on a worst-case view, WDFS duly considers the separation of all classes by maximizing the ratio of the minimum of between-class variance of all class pairs over the maximum of within-class variance. Moreover, we also utilize the correlation between features to help reduce the redundancy and extend WDFS to uncorrelated WDFS (UWDFS). Last

[3]http://featureselection.asu.edu/index.php

but not least, the solution process in our model is simple and effective. Experimental results demonstrate the effectiveness of WDFS and UWDFS.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant 61773302, Innovation Fund of Xidian University No.10221150004, China Postdoctoral Science Foundation (Grant 2019M653564) and the Fundamental Research Funds for the Central Universities.

References

[Battiti, 1994] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[Ben-Bassat, 1982] Moshe Ben-Bassat. Pattern recognition and reduction of dimensionality. 1982.

[Bian and Tao, 2011] Wei Bian and Dacheng Tao. Max-min distance analysis by using sequential sdp relaxation for dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1037–1050, 2011.

[Cai *et al.*, 2013] Xiao Cai, Feiping Nie, and Heng Huang. Exact top-k feature selection via $\ell_{2,0}$-norm constraint. In *IJCAI*, pages 1240–1246, 2013.

[Dash and Liu, 1997] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[Fisher, 1936] Ronald Aylmer Fisher. The use of multiple measuremens in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

[He *et al.*, 2005] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems*, pages 507–514, 2005.

[Kohavi and John, 1997] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

[Loog *et al.*, 2001] Marco Loog, Robert P. W. Duin, and Reinhold Haeb-Umbach. Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[Masaeli *et al.*, 2010] Mahdokht Masaeli, Jennifer G. Dy, and Glenn M. Fung. From transformation-based dimensionality reduction to feature selection. In *ICML*, pages 751–758, 2010.

[Nie *et al.*, 2008] Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676, 2008.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *International Conference on Neural Information Processing Systems*, pages 1813–1821, 2010.

[Pena and Nilsson, 2010] Jose M. Pena and Roland Nilsson. On the complexity of discrete feature selection for optimal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1517–1522, 2010.

[Peng *et al.*, 2005] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[Raileanu and Stoffel, 2004] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[Rao, 1948] C. Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the royal statistical society series b-methodological*, 10(2):159–193, 1948.

[Robnik *et al.*, 2003] Robnik, Marko Ikonja, and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1-2):23–69, 2003.

[Romero and Sopena, 2008] Enrique Romero and Josep Maria Sopena. Performing feature selection with multilayer perceptrons. *IEEE Transactions on Neural Networks*, 19(3):431–441, 2008.

[Su *et al.*, 2018] Bing Su, Xiaoqing Ding, Changsong Liu, and Ying Wu. Heteroscedastic max-min distance analysis for dimensionality reduction. *IEEE Transactions on Image Processing*, 27(8):4052–4065, 2018.

[Tao *et al.*, 2016] Hong Tao, Chenping Hou, Feiping Nie, Yuanyuan Jiao, and Dongyun Yi. Effective discriminative feature selection with nontrivial solution. *IEEE Transactions on Neural Networks*, 27(4):796–808, 2016.

[Wang *et al.*, 2008] Lipo Wang, Nina Zhou, and Feng Chu. A general wrapper approach to selection of class-dependent features. *IEEE Transactions on Neural Networks*, 19(7):1267–1278, 2008.

[Yu and Liu, 2004] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[Yu and Yeung, 2010] Zhang Yu and Dit Yan Yeung. Worst-case linear discriminant analysis. In *International Conference on Neural Information Processing Systems*, 2010.

[Zhang *et al.*, 2018] Rui Zhang, Feiping Nie, and Xuelong Li. Self-weighted supervised discriminative feature selection. *IEEE Transactions on Neural Networks*, 29:3913–3918, 2018.