Machine Learning (2025) 114:138
https://doi.org/10.1007/s10994-025-06773-6

Group feature selection using non-class data

Chunna Li[1] · Yuangang Pan[2] · Weijie Chen[3] · Ivor W. Tsang[2] · Yuanhai Shao[1]

Received: 3 January 2024 / Revised: 4 January 2025 / Accepted: 20 March 2025

Abstract
Existing embedded feature selection methods barely let non-class data contribute to feature selection. However, in some learning tasks, when non-class data have contribution to classification, they should also have an influence to the selection of useful features. For instance, F_∞-norm support vector machine is an effective embedded group feature selection method that performs classification simultaneously. In this paper, we find out that it implicitly uses a kind of non-class data formulated as coordinate Universum when implementing group feature selection, and the information contained in this non-class data could be a meaningful group-wise F_∞-norm penalization. As far as we know, this is the first time that F_∞-norm penalization is understood from this angle. We prove that useful features can be identified through this non-class data that contribute to classifier construction. In addition, to fully explore the classification information provided by this non-class data, we improve F_∞-norm support vector machine by deeming the non-class data as a middle class to better classify positive and negative classes. Experiments show that the non-class data in the proposed method help reduce the labelled data in some sense. Furthermore, it improves F_∞-norm support vector machine in terms of both classification and group feature selection.

Keywords Group feature selection · Non-class data · Universum · Support vector machine · F_∞-norm

1 Introduction

Extremely high dimensionality data have produced serious challenging to learning methods (Tang et al., 2014; Li et al., 2020), such as the curse of dimensionality (Hastie et al., 2009). Feature selection is an important and effective technique to cope with this problem, which selects a subset of relevant features from the original feature set without any transformation, meanwhile maintains physical meanings of the original features. In many real-world applications, features may not come individually but form group structures (Zhai et al., 2016). For example, a typical cause of cancer is the mutation on a gene pathway or a group of structured genes (Yuan et al., 2011; Diez et al., 2021);

Editors: Longbing Cao, David Anastasiu, Qi Zhang, Xiaolin Huang.

Extended author information available on the last page of the article

different frequency bands can be represented by groups in signal processing (Zhang et al., 2018). Therefore, when performing feature selection, one tends to select or not select features belonging to the same group simultaneously. For classification, features are selected with discriminative ability that can discriminate samples from different classes well.

In estimating support vector machine (SVM) (Cortes & Vapnik, 1995), one of the most widely applied classifiers, for simultaneously group feature selection, the feature selection phase can be independent of or closely involved with SVMs. According to this criterion, group feature selection SVMs can be categorized into three ways: filter, wrapper and embedded methods (Guyon & Elisseeff, 2003). Filter SVMs (Bradley & Mangasarian, 1998; Pisner & Schnyer, 2020) first select features independently and then perform SVMs as a second stage. Wrapper SVMs (Guyon et al., 2002; Guo et al., 2021) use a predetermined SVM to evaluate the quality of selected group features. However, they have to run the predefined SVM many times to achieve features quality assessment, which is very computationally expensive. Due to the shortcomings of the above two types of group feature selection SVMs, embedded SVMs (Lal et al., 2006; Jiménez-Cordero et al., 2021) were studied to bridge the gap. They incorporate the statistical criteria as in filter SVMs to select subsets of features of a given cardinality with the highest classification accuracy. Embedded SVMs achieve model fitting and feature selection simultaneously, while including the interaction with the SVM classifiers and having less computational cost than wrapper methods. This kind of SVMs learns classifiers and features simultaneously in one union model, where the subspace learning and model learning complement each other, such as latent subspace learning for image classification (Fang et al., 2018), subspace support vector data description (Sohrab et al., 2018) and discriminative sparse subspace learning (Feng et al., 2024). For embedded group feature SVMs, adding an extra group-wise penalization is considered as the most common and effective technique. Representative group-wise penalization SVMs are group lasso penalized SVM (GLasso-SVM) (Yang & Zou, 2015), combined L_2-L_1-norm based doubly regularized SVM (DrSVM) (Neumann et al., 2005; Wang et al., 2006) and F_∞-norm SVM (F_∞-SVM) (Zou & Yuan, 2008), where F_∞-SVM penalizes the empirical hinge loss as well as the sum of the factor-wise L_∞-norm penalty. The group feature penalization drives coefficients in one group to zero together, and therefore realizes group feature selection.

Most of the existing embedded feature selection techniques in supervised classification only depend on the data of current task. However, data that are not directly belong to the current task may also have an influence to a classifier's performance. For instance, additional comments regarding the characteristics of samples in a class (Vapnik & Vashist, 2009; Vapnik et al., 2015; Yuan et al., 2020), data in a different but relevant domain (Ding et al., 2022; Khan & Swaroop, 2021), images of digit 6 when distinguishing images of digits 5 and 8 (Weston et al., 2006; Richhariya & Tanveer, 2020), they all affect the target classification task. However, these data are rarely directly used for feature selection, while they should affect selecting features since useful features benefit classification. For example, some detection problems such as acoustic event detection (Butko & Nadeu, 2011) use non-class data to find useful features in a filter style by computing the log-likelihood ratios of the class and the non-class. In embedded SVMs for classification, if some non-class data can affect the decision boundary, they should be considered when selecting features. The question is, which kind of non-class data should be considered and which role they play in embedded classifiers for feature selection? Actually, we observe that if some particular sample corresponds to a feature, then it could reveal the essence of a feature being useful by

observing whether its corresponding sample contributes to classification. These samples may not belong to any of the known classes, but still contribute to identifying useful features.

In fact, a kind of non-class samples formulated as Universum that was first brought up by Vapnik (Vapnik, 2006) and further introduced into SVM (Weston et al., 2006) may connect samples with features under some circumstances. For binary classification, *Universum* is a dataset defined as a collection of unlabeled samples known not to belong to either class. Denote *Universum* as $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{m_u}^*\}$, where $\boldsymbol{x}_j^*$ is the j-th *Universum sample*, $j = 1, \ldots, m_u$. The goal of the Universum SVM (U-SVM) (Weston et al., 2006) is to find a separating hyperplane as SVM. Different from SVM, this hyperplane is obtained not only from the given training labelled data but also with the help of the given Universum. Suppose the optimal separating hyperplane that U-SVM looks for is $\boldsymbol{w}^T\boldsymbol{x} - b = 0$. Then U-SVM penalizes Universum by minimizing $L_\varepsilon(\boldsymbol{w}^T\boldsymbol{x}^* - b)$ to achieve maximal contradiction on Universum principle, where $L_\varepsilon(\cdot)$ is the ε-insensitive loss and $\boldsymbol{x}^*$ is the Universum sample. Suppose the Universm only contains samples that one of their features is taken value 1 and other features are value 0, which are called as *coordinate Universum (c-Universum)* in this paper. Then when the classifier is linear without threshold and $\varepsilon = 0$, this penalization becomes the L_1-norm $\|\boldsymbol{w}\|_1$ of the weight vector $\boldsymbol{w}$, which will produce sparse features. However, though Weston et al. (Weston et al., 2006) pointed out such connection ostensibly, the role and effectiveness of the non-class c-Universum in classification or feature selection are not clear, and subsequent studies are also rare.

To explore the importance of non-class data to feature selection, and motivated by the work in (Weston et al., 2006), in this paper, we implement the idea of group feature selection using non-class data by exploring Universum and F_∞-norm penalization. We first reveal that for F_∞-SVM, its sum of the factor-wise L_∞-norm penalty that used for group feature selection implicitly employs some non-class data, while it does not benefit group feature selection or classification. Then we endeavor to construct a novel F_∞-norm Universum support vector machine (F_∞-USVM) by formulating this special type of non-class data as Universum to identify useful features, and in turn give an explanation for features that are selected. At the same time, F_∞-USVM improves the performance of F_∞-SVM on classification by exploring the discriminative information provided by this non-class data. In specific, the contributions of the paper are as follows:

(i) It is the first time to perform F_∞-norm group feature selection by minimizing the loss on a non-class data. Therefore, it allows us to put this data in use for classification, and further identifies useful features from it.
(ii) Compared to F_∞-SVM, the proposed F_∞-USVM gives an explanation that why a feature should be selected from the non-class data angle, and hence connects classification with feature selection. F_∞-USVM identifies a feature as useful if the corresponding c-Universum sample contributes to the construction of the separating hyperplane. It reveals that the non-class c-Universum helps for better quality of the selected features.
(iii) Further, F_∞-USVM improves the classification performance of F_∞-SVM by putting this non-class data in use for classification, incorporating the idea of ordinal regression. The considered non-class data c-Universum in the proposed F_∞-USVM is deemed as a middle class that is exploited to classify positive and negative classes in F_∞-USVM. This demonstrates that this non-class data in F_∞-SVM helps reduce the labelled data.

(iv) Experiments on two simulated datasets and three real-world datasets show that compared to F_∞-SVM and other state-of-the-art SVM feature selection methods, F_∞-USVM can select better useful features while maintain classification performance. In particular, by performing the experiment on an emotional recognition problem where the data is very sparse, the results show that the proposed idea identifies meaningful features and achieves better classification accuracy.

The rest of the paper is organized as follows. Section 2 briefly reviews F_∞-norm support vector machine. Section 3 proposes F_∞-norm Universum support vector machine for group feature selection. Section 4 makes comparisons of the proposed method with its related methods on simulated data and real-world data. At last, concluding remarks are given in Sect. 5.

2 F_∞-norm support vector machine

This paper considers a binary classification problem in the n-dimensional real vector space $\mathbb{R}^n$. All vectors are column ones shown in bold. Given a training dataset $T = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ is the input and $y_i \in \{-1, 1\}$ is the corresponding output, $i = 1, \ldots, m$. Without loss of generality, we suppose the first m_1 data sample are from Class -1, and the following m_2 data sample are from Class $+1$, where $m_1 + m_2 = m$. The goal of support vector machine is to find a mapping $\mathbb{R}^n \xrightarrow{f} \{-1, +1\}$ so that for each $\boldsymbol{x} \in \mathbb{R}^n$, one can deduce its output by $\text{sign}(\boldsymbol{w}^T\boldsymbol{x} - b)$, where $\boldsymbol{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$, and $\text{sign}(\cdot)$ is the sign function. $\|\cdot\|_1$ denotes the vector L_1-norm, which is defined as the absolute value sum of components of a vector. $\|\cdot\|_\infty$ denotes the vector L_∞-norm, which is defined as the maximum of the absolute values of components of a vector.

Suppose the features are generated by G factors namely $F_1, \ldots, F_G$, and S_g is the index set of features generated by F_g, $g = 1, \ldots, G$. Then $\bigcup_{g=1}^{G} S_g = \{1, \ldots, n\}$. Further, suppose $S_g \cap S_{g'} = \emptyset$ for $g \neq g'$. For this group partition, $\boldsymbol{w}$ can also be composed as $\boldsymbol{w} = (\boldsymbol{w}_{S_1}; \ldots; \boldsymbol{w}_{S_G})$, where $\boldsymbol{w}_{S_g}$ is the subvector of $\boldsymbol{w}$ that corresponding to index set S_g.

F_∞-SVM (Zou & Yuan, 2008) is a natural extension of L_1-norm support vector machine (L_1-SVM) (Zhu et al., 2003) that accounts for feature grouping information. It penalizes the empirical SVM loss as well as the sum of the group-wise L_∞-norm. Due to the nature of the L_∞-norm, F_∞-SVM is able to eliminate a given set of features simultaneously. F_∞-norm SVM solves the following problem

$$\min_{\boldsymbol{w},b} \sum_{i=1}^{m} [1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b)]_+ + C \sum_{g=1}^{G} \|\boldsymbol{w}_{S_g}\|_\infty, \tag{1}$$

where $C > 0$ is the trade-off parameter, $(t)_+ = \max\{0, t\}$. For optimal $\boldsymbol{w}$ and b, the class label of a new coming sample $\boldsymbol{x}$ is assigned as $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} - b)$.

By writing $\boldsymbol{w} = \boldsymbol{w}^+ - \boldsymbol{w}^-$ and $b = b^+ - b^-$, where $\boldsymbol{w}^+ \geq 0$ and $\boldsymbol{w}^- \geq 0$ are the positive and negative parts of $\boldsymbol{w}$ satisfying $\boldsymbol{w}^+ \circ \boldsymbol{w}^- = \boldsymbol{0}$, $\circ$ is the Hadamard product, and $b^+ \geq 0$ and $b^- \geq 0$ are the positive and negative parts of b satisfying $b^+b^- = 0$. F_∞-SVM is equivalent to the following alternative formulation

$$
\begin{aligned}
\min_{\boldsymbol{w}^+,\boldsymbol{w}^-,b^+,b^-,\boldsymbol{\xi},\boldsymbol{\eta}} \quad & \sum_{i=1}^{m}\xi_i + C\sum_{g=1}^{G}\eta_g \\
\text{s.t.} \quad & y_i[(\boldsymbol{w}^+ - \boldsymbol{w}^-)^T\boldsymbol{x}_i - (b^+ - b^-)] \geq 1 - \xi_i, \\
& \boldsymbol{w}_j^+ + \boldsymbol{w}_j^- \leq \eta_g,\ j \in S_g,\ g = 1,\ldots,G, \\
& \boldsymbol{w}^+ \geq \mathbf{0},\ \boldsymbol{w}^- \geq \mathbf{0}, \\
& b^+ \geq 0,\ b^- \geq 0, \\
& \xi_i \geq 0,\ i = 1,2,\ldots,m, \\
& \eta_g \geq 0,\ g = 1,2,\ldots,G,
\end{aligned}
\tag{2}
$$

where $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_n)^T$ and $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_G)^T$ are the vectors of slack variables, and $\mathbf{0}$ is all zero vector.

By observing the formulation of (2), F_∞-SVM can be efficiently solved through a standard linear programming problem. Experiments in (Zou and Yuan, 2008) demonstrate that F_∞-SVM has the ability to select features in groups. In fact, from (2), one sees that when $\eta_g = 0$ for some g, then $|\boldsymbol{w}_j| = \boldsymbol{w}_j^+ + \boldsymbol{w}_j^- = 0$, $j \in S_g$. In this case, the features in the g-the group are deemed as useless for F_∞-SVM. However, it will be pointed out in the next section that F_∞-SVM implicitly uses Universum but neglects its role in construction separating hyperplane and feature selection.

3 F_∞-norm Umiversum SVM for group feature selection

3.1 Group feature Universum generation

To utilize the Universum in group feature selection, we take a glance at F_∞-norm penalization $\sum_{g=1}^{G}\|\boldsymbol{w}_{S_g}\|_\infty$. Suppose the features are pre-grouped as in the beginning of the above section. As can be seen from the definition, F_∞-norm penalization is composed by the sum of L_∞-norm terms. The common way to explain the L_∞-norm is the direct observation in $\boldsymbol{w}$ space, as shown in the left panel of Fig. 1. By projecting $\boldsymbol{w}$ into the i-th

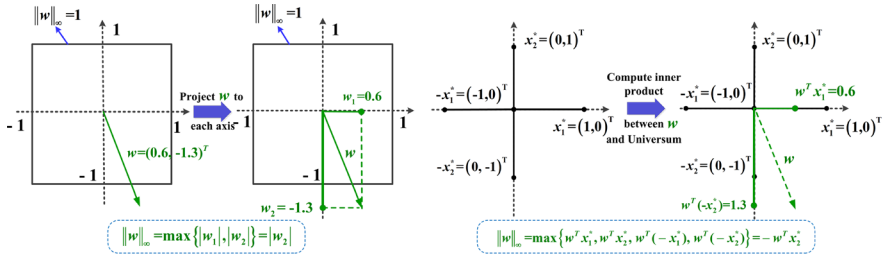


Fig. 1 Illustrations of two explanations of L_∞-norm of $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2)^T = (0.6, -1.3)^T$. **Left Panel:** By projecting $\boldsymbol{w}$ into the two axes, the L_∞-norm of $\boldsymbol{w}$ is computed as $\|\boldsymbol{w}\|_\infty = \max\{|\boldsymbol{w}_1|, |\boldsymbol{w}_2|\} = |\boldsymbol{w}_2| = 1.3$. **Right Panel:** Given Universum $T_u = \{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, -\boldsymbol{x}_1^*, -\boldsymbol{x}_2^*\}$, where $\boldsymbol{x}_1^* = (1,0)^T, \boldsymbol{x}_2^* = (0,1)^T$. Then the L_∞-norm of $\boldsymbol{w}$ is computed as the maximum of the inner product between $\boldsymbol{w}$ and each Universum sample in T_u, i.e., $\|\boldsymbol{w}\|_\infty = \max\{\boldsymbol{w}^T\boldsymbol{x}_1^*, \boldsymbol{w}^T\boldsymbol{x}_2^*, \boldsymbol{w}^T(-\boldsymbol{x}_1^*), \boldsymbol{w}^T(-\boldsymbol{x}_2^*)\} = \boldsymbol{w}^T(-\boldsymbol{x}_2^*) = 1.3$

axis in the $\boldsymbol{w}$ space, one obtains the algebraic length of $\boldsymbol{w}$, say $\boldsymbol{w}_i$. Then the L_∞-norm of $\boldsymbol{w}$ is computed as $\|\boldsymbol{w}\|_\infty = \max\{|\boldsymbol{w}_1|, \ldots, |\boldsymbol{w}_n|\}$. In the case of Fig. 1, $\|\boldsymbol{w}\|_\infty = \max\{|\boldsymbol{w}_1|, |\boldsymbol{w}_2|\} = |\boldsymbol{w}_2|$. The second angle to explain L_∞-norm is in a Universum sample space.

Definition 1 Define coordinate Universum (c-Universum) as the dataset $T_u = \{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*, -\boldsymbol{x}_1^*, \ldots, -\boldsymbol{x}_n^*\}$, where the j-th feature of $\boldsymbol{x}_j^*$ is 1 and other features of $\boldsymbol{x}_j^*$ are 0, $j = 1, \ldots, n$. A sample in c-Universum is called a c-Universum sample.

Property 1 The F_∞-norm penalization $\sum_{g=1}^{G} \|\boldsymbol{w}_{S_g}\|_\infty$ in F_∞-SVM can be realized from the view of inner product between $\boldsymbol{w}$ and c-Universum samples. Further, it can be written as some loss on c-Universum samples.

Proof Without loss of generality, we consider the L_∞-norm of $\boldsymbol{w}$, since the F_∞-norm penalization is the sum of L_∞-norm terms. Then

$$\|\boldsymbol{w}\|_\infty = \max\{\boldsymbol{w}^T\boldsymbol{x}_1^*, \ldots, \boldsymbol{w}^T\boldsymbol{x}_n^*, \boldsymbol{w}^T(-\boldsymbol{x}_1^*), \ldots, \boldsymbol{w}^T(-\boldsymbol{x}_n^*)\}.$$

Therefore, for a given $\boldsymbol{w}$, its L_∞-norm can be computed as the maximum inner product between $\boldsymbol{w}$ and c-Universum samples. Also, $\|\boldsymbol{w}\|_\infty = \max\{|\boldsymbol{w}^T\boldsymbol{x}_1^*|, \ldots, |\boldsymbol{w}^T\boldsymbol{x}_n^*|\}$. Therefore, by defining a loss on c-Universum as $L = \sum_{g=1}^{G} \max_{j \in S_g}\{|\boldsymbol{w}^T\boldsymbol{x}_j^*|\}$, it is exactly the group-wise penalization in F_∞-SVM. □

In the case of Fig. 1, it is obvious that

$$\|\boldsymbol{w}\|_\infty = \max\{\boldsymbol{w}^T\boldsymbol{x}_1^*, \boldsymbol{w}^T\boldsymbol{x}_2^*, \boldsymbol{w}^T(-\boldsymbol{x}_1^*), \boldsymbol{w}^T(-\boldsymbol{x}_2^*)\} = \boldsymbol{w}^T(-\boldsymbol{x}_2^*).$$

As pointed in (Weston et al., 2006), such non-class Universum information should be used to improve the classification performance. In addition, it is expected that this implicitly used Universum could identify which features being selected.

3.2 F_∞-norm Universum SVM formulation

From the above analysis, it is reasonable to introduce the c-Universum

$$T_u = \{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*, -\boldsymbol{x}_1^*, \ldots, -\boldsymbol{x}_n^*\}$$

into F_∞-SVM for classification. In specific, we want to construct two parallel hyperplanes $\boldsymbol{w}^T\boldsymbol{x} - b_1 = 0$ and $\boldsymbol{w}^T\boldsymbol{x} - b_2 = 0$ such that the first hyperplane separates the negative class and c-Universum, while the second hyperplane separates the positive class and c-Universum. To realize this idea, we propose the following F_∞-norm Universum support vector machine (F_∞-USVM) for group feature selection

$$\min_{\boldsymbol{w}, b_1, b_2, \boldsymbol{\xi}, \boldsymbol{\zeta}} \frac{1}{m}\sum_{i=1}^{m}\xi_i + \frac{C_u}{4n}\sum_{j=1}^{n}\sum_{k=1}^{4}\zeta_j^k + \frac{C_r}{G}\sum_{g=1}^{G}\max_{j \in S_g}\{|\boldsymbol{w}^T\boldsymbol{x}_j^*|\} \tag{3}$$

$$\text{s.t. } \boldsymbol{w}^T\boldsymbol{x}_i - b_1 \le -\theta + \xi_i,\ i = 1, \dots, m_1, \tag{4}$$

$$\boldsymbol{w}^T\boldsymbol{x}_i - b_2 \ge \theta - \xi_i,\ i = m_1 + 1, \dots, m, \tag{5}$$

$$\boldsymbol{w}^T\boldsymbol{x}_j^* - b_2 \le -\theta + \zeta_j^1,\ j = 1, \dots, n, \tag{6}$$

$$\boldsymbol{w}^T\boldsymbol{x}_j^* - b_1 \ge \theta - \zeta_j^2,\ j = 1, \dots, n, \tag{7}$$

$$-\boldsymbol{w}^T\boldsymbol{x}_j^* - b_1 \ge \theta - \zeta_j^3,\ j = 1, \dots, n, \tag{8}$$

$$-\boldsymbol{w}^T\boldsymbol{x}_j^* - b_2 \le -\theta + \zeta_j^4,\ j = 1, \dots, n, \tag{9}$$

$$b_1 \le b_2, \tag{10}$$

$$\xi_i \ge 0,\ i = 1, 2, \dots, m, \tag{11}$$

$$\zeta_j^k \ge 0\, j = 1, \dots, n,\ k = 1, 2, 3, 4, \tag{12}$$

where $C_u, C_r > 0$ are trade-off parameters and $\theta > 0$ is the margin, $\boldsymbol{\xi} = (\xi_1, \dots, \xi_{m+2n})^T$, $\boldsymbol{\zeta} = (\zeta_1^1, \dots, \zeta_n^4)^T$ are the vectors of slack variables. After obtaining optimal $\boldsymbol{w}$, b_1 and b_2, for a new coming sample $\boldsymbol{x}$, its class label is assigned as $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} - \frac{1}{2}(b_1 + b_2))$.

From (3)∼(12), we have the following two observations on classification and feature selection.

- For classification, by observing the first two objective terms and combing the constraints of model (3)∼(12), F_∞-USVM treats c-Universum as a middle class, and forces it to lie between the positive and negative classes, which borrows the idea from support vector ordinal regression (Herbrich, 1999; Chu & Keerthi, 2007). In specific, the first, fourth and fifth constraints separate the negative class and c-Universum by the hyperplane $\boldsymbol{w}^T\boldsymbol{x} - b_1 = 0$, and the second, third and sixth constraints separate the positive class and c-Universum by the hyperplane $\boldsymbol{w}^T\boldsymbol{x} - b_2 = 0$. This makes implicit Universum T_u be exploited to provide discriminative information.
- For group feature selection, by minimizing the loss term on c-Universum, then $\max_{j \in S_g}\{|\boldsymbol{w}^T\boldsymbol{x}_j^*|\} = 0$ implies that the inner products of weight vector $\boldsymbol{w}$ and the c-Universum samples $\pm\boldsymbol{x}_j^*$ for all $j \in S_g$ are all 0. In this situation, the g-th group will not be selected. While the c-Universum samples $\pm\boldsymbol{x}_j^*$ are parts of the training data in the construction of (3), we will see later that if the g-th group features have contribution to classification, then the corresponding g-th group features are also useful.

By introducing a nonnegative upper bound η_g for each $\max_{j \in S_g}\{|\boldsymbol{w}^T\boldsymbol{x}_j^*|\}$, model (3)∼(12) can be equivalently written as a linear programming problem by adding constraints $|\boldsymbol{w}^T\boldsymbol{x}_j^*| \le \eta_g$ and $\eta_g \ge 0$, $g = 1, 2, \dots, G$ and minimizing the third term in (3) as $\sum_{g=1}^{G} \eta_g$ instead.

Therefore, the standard interior-point algorithm could be used to solve F_∞-USVM directly. For the above proposed F_∞-USVM, we further discuss the effect of c-Universum in feature selection. Given an optimal solution, for c-Universum samples that strictly lie between the positive class and negative class, they satisfy $\boldsymbol{w}^T\boldsymbol{x} - b_1 > \theta$ and $\boldsymbol{w}^T\boldsymbol{x} - b_2 < -\theta$. These samples have little influence on the classifier construction. Therefore, we only discuss the c-Universum samples satisfying $\boldsymbol{w}^T\boldsymbol{x} - b_1 \leq \theta$ or $\boldsymbol{w}^T\boldsymbol{x} - b_2 \geq -\theta$.

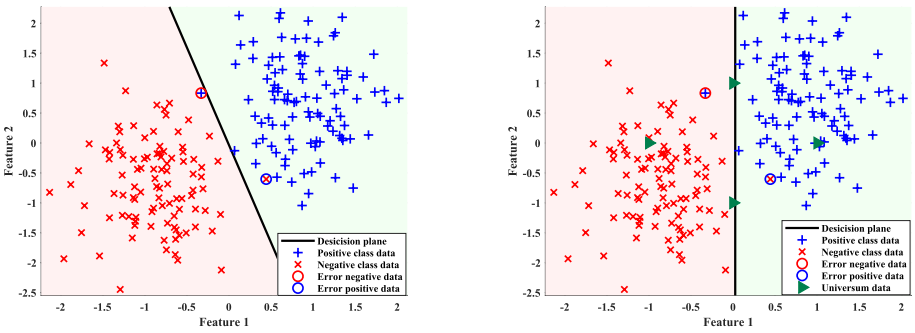
Definition 2 If a c-Universum sample $\boldsymbol{x}^*$ satisfies one of the inequalities $\boldsymbol{w}^T\boldsymbol{x} - b_1 \leq \theta$ or $\boldsymbol{w}^T\boldsymbol{x} - b_2 \geq -\theta$, we say $\boldsymbol{x}^*$ is an influenced c-Universum sample for classification in F_∞ -USVM.

In the following, we show that c-Universum has relation to some useful features, where the j-th feature is said to be useful if $w_j \neq 0$.

Proposition 1 Given an optimal solution of F_∞-USVM, suppose there exist some $j_1, j_2 \in \{1, \ldots, n\}$ such that $\zeta_{j_1}^1 + \zeta_{j_1}^4 = 0$ and $\zeta_{j_2}^2 + \zeta_{j_2}^3 = 0$. Then if $\boldsymbol{x}_j^*$ is an influenced c-Universum, it identifies the corresponding j-th feature as a useful feature.

Proof By adding the constraints (7) and (8) together, $b_1 + \theta \leq \min\limits_{j=1,\ldots,n} \{\zeta_j^2 + \zeta_j^3\}$ is obtained. Under the assumption, it has $\min\limits_{j=1,\ldots,n} \{\zeta_j^2 + \zeta_j^3\} = 0$ and hence $b_1 + \theta \leq 0$. Similarly, by adding the constraints (6) and (9) together, it follows $b_2 - \theta \geq -\min\limits_{j=1,\ldots,n} \{\zeta_j^1 + \zeta_j^4\}$, and the assumption gives $b_2 - \theta \geq 0$. Therefore, $b_1 + \theta \leq 0 \leq b_2 - \theta$. Without loss of generality, we assume $b_1 + \theta < 0 < b_2 - \theta$. If $\boldsymbol{x}_j^*$ is an influenced c-Universum for classification, then $\pm\boldsymbol{w}^T\boldsymbol{x}_j^* - b_1 \leq \theta$ or $\pm\boldsymbol{w}^T\boldsymbol{x}_j^* - b_2 \geq -\theta$. If $\pm\boldsymbol{w}^T\boldsymbol{x}_j^* - b_1 \leq \theta$, then $\pm\boldsymbol{w}^T\boldsymbol{x}_j^* \leq b_1 + \theta < 0$ and $\boldsymbol{w}^T\boldsymbol{x}_j^* \neq 0$. This implies that the j-th feature is deemed useful. Similarity goes when $\pm\boldsymbol{w}^T\boldsymbol{x}_j^* - b_2 \geq -\theta$. The proof is completed. □

Figure 2 gives an illustration of F_∞-SVM and F_∞-USVM on a binary two-dimensional data, where each feature constitutes a feature group. One easily observes that Feature 1 is enough to separate the data well, and thus Feature 1 will be identified as useful if a sparse classifier is applied. It can be seen that with the help of c-Universum data, F_∞-USVM



(a) F_∞-SVM (b) F_∞-USVM

Fig. 2 Influence of c-Universum to decision hyperplane on a two-dimensional data

classifies the data well by just using Feature 1, while F_∞-SVM has to use both of the two features to separate two classes.

In summary, putting the c-Universum samples representing features into the classifier will make F_∞-USVM possess fair group feature selection ability as well as good classification performance.

4 Experiments

In this section, F_∞-USVM is experimentally compared with standard SVM (Cortes & Vapnik, 1995), L_1-norm based feature selection L_1-SVM (Zhu et al., 2003), and representative group feature selection SVMs, including DrSVM (Wang et al., 2006), GLasso-SVM (Yang & Zou, 2015) and F_∞-SVM (Zou & Yuan, 2008). Trade-off parameters for other methods are optimally selected from the set $\{2^{-8}, \ldots, 2^0, \ldots, 2^8\}$ by grid search and five-fold cross validation. Parameters for our F_∞-USVM is optimally selected from $\{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. Experiments are performed on two simulated datasets with different numbers of samples and features, two medical datasets, and an emotional recognition dataset. All the methods are carried out on a PC with Intel i5 1.60 GHz CPU by Matlab 2017b. All data are preprocessed by z-score normalization (Cheadle et al., 2003).

4.1 Simulated datasets

This part considers two simulated datasets similar to those in (Zou and Yuan, 2008), which focus on the situations that features are naturally grouped. The brief information of the simulated datasets is listed in Table 1, while their detailed construction is put in the Appendix for readability. In the table, the i-th simulated dataset is denoted as Si, $i = 1, 2$. S1 considers the first order interaction among features, and features of S2 within a group have the same pairwise correlation.

4.1.1 Classification performance

For each of these two simulated datasets, we first consider the case that when fixing the number of data features to 40 (32), and 1000, 500, 300, 200, 100 and 50 samples are generated respectively. To investigate if the proposed method works on high-dimensional data, we also consider the case that when fixing the number of data samples to 100, then

Table 1 Statistics information of simulated datasets

Data	#features	#groups	#true groups	True groups
S1	40	13	2	$\{F_1, \ldots, F_{10}\}, \{F_{11}, \ldots, F_{20}\}$
	200	53	2	$\{F_1, \ldots, F_{50}\}, \{F_{51}, \ldots, F_{100}\}$
	400	103	2	$\{F_1, \ldots, F_{100}\}, \{F_{101}, \ldots, F_{200}\}$
	800	203	2	$\{F_1, \ldots, F_{200}\}, \{F_{201}, \ldots, F_{400}\}$
S2	32	10	3	$\{F_1, F_2\}, \{F_3, F_4\}, \{F_9, F_{15}, F_{21}, F_{27}\}$
	200	55	3	$\{F_1, F_2\}, \{F_3, F_4\}, \{F_{21}, F_{66}, F_{111}, F_{156}\}$
	450	120	3	$\{F_1, F_2\}, \{F_3, F_4\}, \{F_{31}, F_{136}, F_{241}, F_{346}\}$
	800	210	3	$\{F_1, F_2\}, \{F_3, F_4\}, \{F_{41}, F_{231}, F_{421}, F_{611}\}$

200, 400 (450) and 800 features are generated. The details about how to generate these high-dimensional features can also be found in the Appendix. For each case, random 20% samples of them are used for training, and the rest 80% samples are used for testing. The procedure is repeated 10 times, and the mean classification accuracy (Acc) and standard derivation (Std) averaged over these 10 runs for all the methods are recorded in Tables 2 and 3.

From the results, we see that: (i) As the number of samples decreases, the performance of all the methods drops. Particularly, most of the methods perform much poorer when there are 20 or 10 samples. (ii) When the number of samples is relatively larger than the number of features, F_∞-USVM is not very competitive, and is only slightly better than or comparable to the other methods; however, when the number of samples is small, F_∞-USVM has an advantage; (iii) For high-dimensional case, F_∞-USVM can still select the proper features or groups while maintain fair classification performance. It can be observed that the differences of accuracies between F_∞-USVM and most of the other methods become larger in general. In particular, F_∞-USVM improves F_∞-SVM. It shows that the Universum provides useful discriminative information, and it works well when the number of labelled samples is small.

4.1.2 Feature selection performance

We then investigate the feature selection ability. The last four columns of Tables 2 and 3 report the number of selected truly useful groups, the number of selected features, the number of selected truly useful features, and the number of selected noise features. In the tables, "#" means number, and the feature selection results are observed when a classifier is applied on all data samples. The closer of #selected true groups and #selected true features to the true group and true feature numbers the better, and the less #selected noise features the better.

From the results, one sees that: (i) SVM always can select almost all the feature groups, it is because it selects almost all the features. (ii) Though L_1-SVM can select features, sometimes it can not identify features in groups. For examples, on data S1 with number of samples 20, it selects 21 features, which is more than ground true features. However, it does not identify any feature groups. (iii) For the group feature selection SVMs, F_∞-USVM can select useful group features, and selects less noise features, which shows the feature selection role played by Universum. In addition, though some feature selection SVMs may select useful features and feature groups, they do not possess better classification performance as F_∞-USVM. This again shows the effect of Universum in classification for F_∞-USVM.

4.2 Medical datasets

4.2.1 Cleveland dataset

Cleveland dataset[1] is Dr. Detrano's database modified to be a real dataset of 303 samples with 13 features, including 7 categorical features and 6 numeric features (Detrano et al., 1989). The dataset comes from the Cleveland Clink in Cleveland, Ohio, for the diagnosis

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease..

Table 2 Comparison results on simulated dataset S1. Bold figure shows the best result on each dataset

#samples× #features	Method	Acc±Std	#selected true groups	#selected features	#selected true features	#selected noise features
200×40	SVM	93.19 ± 0.45	2	40	20	20
	L_1-SVM	91.47 ± 0.33	1	30	19	11
	DrSVM	93.89 ± 0.18	2	31	20	11
	GLasso-SVM	91.52 ± 0.45	2	35	20	15
	F_∞-SVM	94.95 ± 0.20	2	30	20	10
	F_∞-USVM (ours)	$\mathbf{95.30 \pm 0.16}$	2	20	20	0
100×40	SVM	91.25 ± 0.89	2	40	20	20
	L_1-SVM	90.32 ± 0.77	1	29	19	10
	DrSVM	92.21 ± 0.60	2	34	20	14
	GLasso-SVM	92.31 ± 0.48	2	37	20	17
	F_∞-SVM	94.55 ± 0.40	2	30	20	10
	F_∞-USVM (ours)	$\mathbf{95.33 \pm 0.32}$	2	20	20	0
60×40	SVM	91.26 ± 0.76	2	40	20	20
	L_1-SVM	88.28 ± 0.97	1	35	18	17
	DrSVM	91.19 ± 1.07	2	23	20	3
	GLasso-SVM	92.95 ± 1.41	2	30	20	10
	F_∞-SVM	92.15 ± 0.70	2	30	20	10
	F_∞-USVM (ours)	$\mathbf{93.13 \pm 1.07}$	2	20	20	0
40×40	SVM	89.00 ± 0.79	2	40	20	20
	L_1-SVM	89.16 ± 1.22	0	25	15	10
	DrSVM	89.29 ± 1.18	2	34	20	14
	GLasso-SVM	88.81 ± 1.13	2	30	20	10
	F_∞-SVM	91.58 ± 1.20	2	30	20	10
	F_∞-USVM (ours)	$\mathbf{92.54 \pm 1.07}$	2	20	20	0
20×40	SVM	89.05 ± 1.74	2	40	20	20
	L_1-SVM	82.03 ± 1.93	0	21	10	11
	DrSVM	87.23 ± 2.47	2	33	20	13
	GLasso-SVM	88.13 ± 4.05	2	20	20	0
	F_∞-SVM	85.95 ± 2.89	2	33	20	13
	F_∞-USVM (ours)	$\mathbf{92.55 \pm 1.56}$	2	20	20	0
10×40	SVM	86.15 ± 2.56	2	40	20	20
	L_1-SVM	75.60 ± 3.01	0	18	10	8
	DrSVM	85.10 ± 4.92	2	39	20	19
	GLasso-SVM	82.90 ± 5.47	2	40	20	20
	F_∞-SVM	82.35 ± 4.40	2	30	20	10
	F_∞-USVM (ours)	$\mathbf{90.70 \pm 2.75}$	2	20	20	0
100×200	SVM	88.47 ± 2.00	2	200	100	100
	L_1-SVM	84.20 ± 3.15	0	35	15	20
	DrSVM	91.23 ± 1.31	0	186	96	90
	GLasso-SVM	92.28 ± 0.98	2	141	100	41
	F_∞-SVM	90.50 ± 1.10	2	150	100	50
	F_∞-USVM (ours)	$\mathbf{94.58 \pm 0.91}$	2	150	100	50

Table 2 (continued)

#samples×#features	Method	Acc±Std	#selected true groups	#selected features	#selected true features	#selected noise features
100×400	SVM	92.88 ± 2.64	2	400	200	200
	L_1-SVM	81.83 ± 2.94	0	42	23	19
	DrSVM	93.15 ± 1.56	0	294	163	131
	GLasso-SVM	87.73 ± 1.87	2	300	200	100
	F_∞-SVM	93.35 ± 1.68	2	300	200	100
	F_∞-USVM (ours)	**94.23 ± 1.26**	2	300	200	100
100×800	SVM	91.30 ± 1.04	2	800	400	400
	L_1-SVM	81.38 ± 2.33	0	44	25	19
	DrSVM	89.38 ± 1.78	0	456	280	176
	GLasso-SVM	90.73 ± 4.24	2	598	400	198
	F_∞-SVM	92.45 ± 2.26	2	600	400	200
	F_∞-USVM (ours)	**93.13 ± 2.16**	2	598	400	198

of coronary artery disease of 303 patients under-going angiography. The class is either healthy (buff) or with heart-disease (sick). This dataset is interesting because it contains mixed features of both continuous and categorical. To verify the group feature section ability, we code these categorical features by dummy variables of 0 and 1. Therefore, some natural group features are generated. Each continuous feature is deemed as an individual group. By using this technique, we have 23 features that form 13 groups. The details of these features and groups are listed in the first two columns of Table 4. During experiments, five-fold cross validation is used for parameter searching, and then ten-fold cross validation averaged classification and feature selection results under optimal parameters are adopted.

To see which groups are selected, we compute the average frequency of features being selected in each group. If the average frequency of a group is 1, then this group is selected, otherwise, it is not selected. We list these frequencies for all methods in Table 4. The selected feature groups are marked by “check" in the bracket. We have the following observations. (i) The table shows that SVM selects all groups of features except the third group, while this group has very high frequency of 0.9900. This demonstrates that SVM does not have have the ability for feature selection. Apart from SVM, DrSVM, GLasso-SVM and our F_∞-USVM can select feature groups on this data. (ii) We also list the classification accuracy along with standard derivation of each method at the bottom of Table 4. The result shows that SVM has the highest classification accuracy, however, it uses all features. Apart from SVM, F_∞-USVM outperforms other methods with relatively stable performance. By combining the feature selection results, it demonstrates that F_∞-USVM classifies and selects useful group features simultaneously on this dataset.

To further compare the classification performance as well as the feature selection ability for each method, we depict their classification accuracies and the frequencies of selected features and feature groups under different values of sparseness regularization parameters, while other parameters are optimally set. As shown in Fig. 3, SVM always selects a very high percentage of features or feature groups, since it does not have feature selection ability. For L_1-SVM and F_∞-SVM, though they can select some features or feature groups under some parameters, they may not possess the best classification performance at the

Table 3 Comparison results on simulated dataset S2. Bold figure shows the best result on each dataset

#samples× #features	Method	Acc±Std	#selected true groups	#selected features	#selected true features	#selected noise features
200×32	SVM	**100.00 ± 0.00**	3	32	8	24
	L_1-SVM	**100.00 ± 0.00**	1	4	4	0
	DrSVM	**100.00 ± 0.00**	1	4	4	0
	GLasso-SVM	98.22 ± 1.70	1	4	4	0
	F_∞-SVM	**100.00 ± 0.00**	1	4	4	0
	F_∞-USVM (ours)	**100.00 ± 0.00**	1	4	4	0
100×32	SVM	**100.00 ± 0.00**	3	32	8	24
	L_1-SVM	99.85 ± 0.47	1	4	4	0
	DrSVM	98.96 ± 0.68	1	4	4	0
	GLasso-SVM	99.05 ± 1.85	1	4	4	0
	F_∞-SVM	99.76 ± 0.51	1	4	4	0
	F_∞-USVM (ours)	**100.00 ± 0.00**	1	4	4	0
60×32	SVM	99.14 ± 0.73	3	32	8	24
	L_1-SVM	99.71 ± 0.92	1	4	4	0
	DrSVM	99.36 ± 0.48	1	4	4	0
	GLasso-SVM	98.52 ± 1.94	1	4	4	0
	F_∞-SVM	99.78 ± 0.46	1	4	4	0
	F_∞-USVM (ours)	**100.00 ± 0.00**	1	4	4	0
40×32	SVM	98.05 ± 1.12	3	32	8	24
	L_1-SVM	98.03 ± 1.55	1	4	4	0
	DrSVM	97.86 ± 2.24	1	4	4	0
	GLasso-SVM	94.59 ± 2.60	1	4	4	0
	F_∞-SVM	98.96 ± 1.22	1	4	4	0
	F_∞-USVM (ours)	**99.65 ± 0.46**	1	4	4	0
20×32	SVM	91.03 ± 3.26	3	32	8	24
	L_1-SVM	88.03 ± 3.43	1	4	4	0
	DrSVM	85.40 ± 4.95	3	29	8	21
	GLasso-SVM	81.55 ± 5.88	1	4	4	0
	F_∞-SVM	95.83 ± 2.25	1	4	4	0
	F_∞-USVM (ours)	**98.43 ± 0.99**	1	4	4	0
10×32	SVM	81.55 ± 3.50	3	31	8	23
	L_1-SVM	77.85 ± 3.16	1	4	4	0
	DrSVM	73.90 ± 3.76	3	25	8	17
	GLasso-SVM	78.65 ± 4.99	3	32	8	24
	F_∞-SVM	81.35 ± 5.15	1	4	4	0
	F_∞-USVM (ours)	**90.40 ± 6.15**	1	4	4	0

Table 3 (continued)

#samples×#features	Method	Acc±Std	#selected true groups	#selected features	#selected true features	#selected noise features
100×200	SVM	71.25 ± 3.23	3	200	8	192
	L_1-SVM	77.43 ± 3.65	1	4	4	0
	DrSVM	76.83 ± 3.95	3	8	8	0
	GLasso-SVM	67.80 ± 2.64	3	196	8	188
	F_∞-SVM	93.65 ± 5.53	1	4	4	0
	F_∞-USVM (ours)	**98.48 ± 1.66**	1	4	4	0
100×450	SVM	67.63 ± 2.17	3	450	8	442
	L_1-SVM	71.60 ± 6.89	1	4	4	0
	DrSVM	70.88 ± 4.94	0	3	3	0
	GLasso-SVM	63.83 ± 1.89	1	76	4	72
	F_∞-SVM	95.20 ± 4.17	1	4	4	0
	F_∞-USVM (ours)	**98.18 ± 1.80**	1	4	4	0
100×800	SVM	64.30 ± 2.85	3	799	8	791
	L_1-SVM	68.43 ± 4.12	1	4	4	0
	DrSVM	64.65 ± 2.41	3	350	8	342
	GLasso-SVM	64.45 ± 5.90	3	800	8	792
	F_∞-SVM	86.03 ± 2.55	1	4	4	0
	F_∞-USVM (ours)	**98.13 ± 2.43**	1	4	4	0

Table 4 The frequency of features being selected in each group on the Cleveland dataset. Here "**1**($\surd$)" means that the frequency of features in the corresponding group being selected is 100% and the group is selected

Group	SVM	L_1-SVM	DrSVM	GLasso-SVM	F_∞-SVM	F_∞-USVM (ours)
1	**1** ($\surd$)	0.6000	0.9800	**1** ($\surd$)	0.2600	0.4000
2	**1** ($\surd$)	0.4000	**1** ($\surd$)	0.9600	0.7400	0.9600
3	0.9900	0.6100	0.9450	**1** ($\surd$)	0.9550	**1** ($\surd$)
4	**1** ($\surd$)	0.7600	**1** ($\surd$)	0.9800	0.3600	0.5000
5	**1** ($\surd$)	0.7400	**1** ($\surd$)	0.9800	0.1400	0.3200
6	**1** ($\surd$)	0.7800	0.9600	0.9600	0.2800	0.2800
7	**1** ($\surd$)	0.3000	0.7800	**1** ($\surd$)	0.5867	0.7733
8	**1** ($\surd$)	0.7000	0.9600	0.9600	0.4400	0.7000
9	**1** ($\surd$)	0.7800	**1** ($\surd$)	**1** ($\surd$)	0.6000	0.7400
10	**1** ($\surd$)	0.7200	**1** ($\surd$)	0.9800	0.3400	0.7400
11	**1** ($\surd$)	0.4733	0.9600	**1** ($\surd$)	0.8133	**1** ($\surd$)
12	**1** ($\surd$)	0.9400	**1** ($\surd$)	**1** ($\surd$)	0.8000	0.9800
13	**1** ($\surd$)	0.5400	0.9133	**1** ($\surd$)	0.9533	**1** ($\surd$)
Acc±Std	**82.63 ± 0.67**	78.22 ± 1.30	81.40 ± 1.20	73.10 ± 2.53	78.40 ± 1.76	81.57 ± 0.65

(a) SVM (b) L_1-SVM

(c) DrSVM (d) GLasso-SVM

(e) F_∞-SVM (f) F_∞-USVM

Fig. 3 The classification accuracy, frequency of selected features, and frequency of selected feature groups for all the methods under different parameters on the Cleveland dataset

same time. For DrSVM, GLasso-SVM and F_∞-USVM, they all perform well and can select some features and feature groups simultaneously under some parameters, while DrSVM and F_∞-USVM outperform GLasso-SVM.

4.2.2 WOBC dataset

Breast cancer Wisconsin (original) dataset[2] from the UCI repository contains 699 breast cancer patients, of which 458 are benign and 241 are malignant. Each instance is described by 9 attributes with integer value in the range 1-10. Similar to the Cleveland data, we represent features by dummy variables of 0 and 1. Therefore, there are 9 groups of features and each group contains 10 features, as displayed in Table 5.

[2] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29.

Table 5 The frequency of features being selected in each group on the WOBC dataset. Here "**1** ($\surd$)" means that the frequency of features in the corresponding group being selected is 100% and the group is selected

Group	SVM	L_1-SVM	DrSVM	GLasso-SVM	F_∞-SVM	F_∞-USVM (ours)
1	**1** ($\surd$)	0.3000	0.3680	0.0000	0.9520	0.8600
2	**1** ($\surd$)	0.3280	0.5400	0.9800	0.9060	**1** ($\surd$)
3	**1** ($\surd$)	0.3320	0.5520	0.8800	0.9600	**1** ($\surd$)
4	0.9980	0.2500	0.3620	0.3200	0.8780	0.6800
5	0.9980	0.2460	0.4400	0.5000	0.8020	**1** ($\surd$)
6	**1** ($\surd$)	0.4260	0.4120	**1** ($\surd$)	0.8620	0.9400
7	**1** ($\surd$)	0.2400	0.4980	0.0800	0.9080	0.9800
8	0.9980	0.3800	0.4680	0.6800	0.9140	0.9400
9	0.9000	0.1360	0.2140	0.0180	0.3400	0.2160
Acc±Std	96.03 ± 0.22	93.32 ± 0.76	95.55 ± 0.41	94.98 ± 0.38	93.35 ± 0.68	**96.12 ± 0.21**

From Table 5, we see that except SVM, F_∞-USVM is the only method can select group features, namely the second, the third, and the fifth groups. In contrast, other methods can not find the same selected group among ten random splits. It also shows that the classification accuracies of other feature selection methods are not as satisfied as F_∞-USVM. In fact, F_∞-USVM has comparable performance as SVM on this data.

As in the Cleveland dataset, we also present the classification accuracies and the frequencies of selected features and feature groups under different values of sparseness regularization parameters for each method, as in Fig. 4. We can see that SVM can not select features, while L_1-SVM can not select the same features for different runs. For DrSVM and F_∞-SVM, they have the ability to select features but fail to select a whole group. In contrast, GLasso-SVM and our F_∞-USVM can select groups of features while F_∞-USVM outperforms GLasso-SVM on classification.

4.3 Emotional recognition dataset

This subsection considers a text dataset regarding the emotions of netizens on COVID-19 that is collected by Beijing government.[3] The data contain the comments related to COVID-19 on weibo.com between 2020/01/01-2020/02/20, including release time, account, content, pictures and videos (optional). Emotional tendency of each data sample is annotated artificially, and we here consider the text content of emotion tendency $-1, +1$, where $-1, +1$ represent the negative and positive emotions, respectively. 400 samples with label $+1$ and 400 samples with label $+1$ are used. The data are processed using term frequency-inverse document frequency (TF-IDF) technique, and the key words of TF-IDF greater than 0.003 are kept, which leaves 1160 features. These features are grouped into 50 clusters using k-means. Fifteen percent samples are randomly selected for training, while the rest samples are for testing.

The test classification accuracy, the number of selected features, the number of selected groups, the number of selected features belong to the selected groups, and the

[3] https://data.beijing.gov.cn/kjzy2020/index.html.

Fig. 4 The classification accuracy, frequency of selected features, and frequency of selected feature groups for all the methods under different parameters on the WOBC dataset

sparsity that represents the ratio of selected features to all features are listed in Table 6. The results show that F_∞-USVM outperforms other methods in terms of accuracy, especially is better than F_∞-SVM.

Table 6 The classification accuracy and the number of selected features on the emotional recognition dataset

Characteristic	SVM	L_1-SVM	DrSVM	GLasso-SVM	F_∞-SVM	F_∞-USVM (ours)
Acc	61.76	59.26	67.65	60.74	63.68	**68.53**
#selected features	1160	62	992	556	104	73
#selected groups	50	2	32	32	6	4
#selected features in groups	1160	2	79	171	16	59
Sparsity	0.00	94.66	14.48	52.07	91.03	93.71

(a) Support and blessing (b) Anxiety (c) Rational attention (d) Resort for help

Fig. 5 Feature groups selected by F_∞-USVM on the emotional recognition dataset. The 1st row is the original Chinese words in different groups, and the 2nd row is the corresponding English translation

Table 7 Overall comparison among F_∞-USVM and related baselines. The results are the summarized performance of each method on all the datasets, the details of which are described in Sect. 4.4

Method/Characteristic	FS	Group FS	Universum	Acc±Std	FS-NFS ratio	Goup FS ratio
SVM	No	No	No	84.77 ± 1.12	–	–
L_1-SVM	Yes	No	No	84.73 ± 1.40	0.0181	0.0057
DrSVM	Yes	Yes	No	87.34 ± 1.08	0.2350	0.1574
Glasso-SVM	Yes	Yes	No	83.71 ± 1.84	0.2802	0.3604
F_∞-SVM	Yes	Yes	No	85.42 ± 1.29	0.1483	0.2790
F_∞-USVM (ours)	Yes	Yes	Yes	**88.60 ± 0.67**	**0.2651**	**0.5682**

On the number of selected features and feature groups, SVM does not have the feature selection ability, while L_1-SVM selects the the least number of features, and F_∞-SVM and F_∞-USVM can select features. In specific, L_1-SVM and F_∞-SVM and F_∞-USVM have SFRatio greater than 90%. However, F_∞-USVM has better classification accuracy than those of L_1-SVM and F_∞-SVM.

As for the selected feature groups, one sees that L_1-SVM selects the least number of feature groups, while each of these two groups only contains one feature. This shows that though L_1-SVM can select few features, it barely has group feature selection ability. In contrast, for the proposed F_∞-USVM, it selects four feature groups that contain 59 features totally, which covers most of its selected features. By observing the SGNo, SGFeaNo and SFeaNo values, we see that the features selected by the proposed F_∞-USVM mostly come from feature groups, while DrSVM, GLasso-SVM and F_∞-SVM select less features from the feature groups. To see if the features and feature groups selected by F_∞-USVM are meaningful, we plot its selected (groups of) words in Fig. 5a–d. In Fig. 5e–h, we give the corresponding English translation. From the figure, one observes that each selected group has its own characteristic. The first group reflects the support to COVID-19 control work and blessing to relevant workers; the second group reflects the anxiety when facing the epidemic situation; the third group reflects the rational attention to the epidemic situation; the fourth group reflects the resort for help on unknown or uncertain information. In summary, the above results confirm the classification and feature selection ability of the proposed F_∞-USVM.

4.4 Overall comparison

To give a clear and overall behavior comparison between F_∞-USVM and its related methods, their characteristics and average experimental results are summarized in Table 7. In the table, FS represents feature selection. The first three columns demonstrate whether a method has the feature selection ability, the group feature selection ability, and whether uses Universum information. The last three columns demonstrate the average experimental results across all datasets. The fourth column shows the average and standard derivation of accuracies. The fifth column first computes the number of true useful features in the selected features and the number of useless features in the selected features, and then compute the ratio of their difference to the number of total features (FS-NFS ratio). The last column computes the ratio of the number of selected useful groups and the number of true groups (Goup FS ratio). For both FS-NFS ratio and Goup FS ratio, they are the higher the better.

From Table 7 and by combining the above observations, one sees that all methods have the feature selection ability except SVM. For methods with feature selection ability, the indicators of the last three columns are obviously the larger the better. It is clear that L_1 -SVM has feature selection ability, but it can not effectively identify group features. DrSVM, GLasso-SVM, F_∞-SVM and our F_∞-USVM have group feature selection ability, but DrSVM, GLasso-SVM, and F_∞-SVM do not employ the group feature selection and classification information provided by implicit Universum, which affects their performance on selecting useful features and classifying samples. In contrast, the proposed F_∞-USVM fully employs this information and has the highest FS-NFS ratio and Goup FS ratio. Combing its recognition accuracy, one sees F_∞-USVM outperforms other methods.

5 Conclusion

This paper implements the idea of group feature selection using non-class data by exploring a special Universum and F_∞-norm penalization, and proposes a novel F_∞-norm Universum support vector machine. This type Universum realizes group feature selection by imposing a new loss, and it also plays a discriminative role in classification. Empirical results show that F_∞-norm Universum support vector machine outperforms related methods, especially F_∞-norm support vector machine. It should be pointed out that F_∞-norm support vector machine is proposed for predefined feature groups. When there is no information on groups, one can choose to cluster features first and then apply the proposed method, as pointed out in (Zou and Yuan, 2008). Of course, obtaining proper clustered features is also not an easy job and hence is worth investigating.

Appendix A The construction of the Simulated data

For simulated data, 5-fold cross validation is used for parameter searching. Different from the common 5-fold cross validation, we here use 20% of the data for training, and 80% of the data for testing. Under the optimal parameters, 10 times 5-fold cross validation results are recorded.

Simulated data 1: Firstly, thee latent variables Z_1, Z_2 and Z_3 and 40 variables $\{\varepsilon_i | i = 1, \ldots, 40\}$ are randomly independently generated from a standard normal distribution. Then the features of this simulated data are defined by

$$\begin{aligned} F_i &= Z_1 + 0.5\varepsilon_i, i = 1, \ldots, 10, \\ F_i &= Z_2 + 0.5\varepsilon_i, i = 11, \ldots, 20, \\ F_i &= Z_3 + 0.5\varepsilon_i, i = 21, \ldots, 30, \\ F_i &= \varepsilon_i, i = 31, \ldots, 40. \end{aligned} \tag{A1}$$

For these fixed 40 features, 1000, 500, 300, 200, 100 and 50 samples are generated respectively, and for each data sample, its label is decided by which sides of the hyperplane $4Z_1 + 3Z_2 + 1 = 0$ it lies in. It can be seen that the first 20 features form two groups in which the pairwise correlation within each group is 0.8. Similarly, the second and third 10 features form the second group and third group with the same correlation. Since the last 10 features are independent noise features, each of them form an individual group of size one. Therefore, this data has 13 groups of 40 features totally. By observing the decision plane, it can be seen that the true groups are the first two groups $\{F_1, \ldots, F_{10}\}, \{F_{11}, \ldots, F_{20}\}$ corresponding to the first 20 features.

To further investigate if the proposed method works on the high-dimensional data, we also generate 200, 400 and 800 variables $\{\varepsilon_i\}$ as above, which correspond to 200, 400 and 800 features.

Simulated data 2: This simulated data first constructs four latent variables $Z_1, \ldots, Z_4$ and then generates two types of discrete features F_{2i-1} and F_{2i} for each Z_i, $i = 1, \ldots, 4$. The first type feature F_{2i-1} satisfies that if $Z_i \geq \Phi^{-1}(\frac{2}{3})$, then the feature value is 1, and 0 otherwise. The second type feature F_{2i} satisfies that if $Z_i \leq \Phi^{-1}(\frac{1}{3})$, then the feature value is 1, and 0 otherwise. On top of the above 8 features, we further construct more features with interactions between them. In specific, features $F_9, \ldots, F_{14}$ take value 1 if $Z_i \geq \Phi^{-1}(\frac{2}{3})$ and $Z_j \geq \Phi^{-1}(\frac{2}{3})$ for $1 \leq i \leq j \leq 4$ and 0 otherwise. For example, feature F_9 takes value 1 if $Z_1 \geq \Phi^{-1}(\frac{2}{3})$ and $Z_2 \geq \Phi^{-1}(\frac{2}{3})$ and 0 otherwise. Similarly, features $F_{15}, \ldots, F_{20}$ take value 1 if $Z_i \geq \Phi^{-1}(\frac{2}{3})$ and $Z_j \leq \Phi^{-1}(\frac{1}{3})$ for $1 \leq i \leq j \leq 4$ and 0 otherwise, features $F_{21}, \ldots, F_{26}$ take value 1 if $Z_i \leq \Phi^{-1}(\frac{1}{3})$ and $Z_j \geq \Phi^{-1}(\frac{2}{3})$ for $1 \leq i \leq j \leq 4$ and 0 otherwise, and features $F_{27}, \ldots, F_{32}$ take value 1 if $Z_i \leq \Phi^{-1}(\frac{1}{3})$ and $Z_j \leq \Phi^{-1}(\frac{1}{3})$ for $1 \leq i \leq j \leq 4$ and 0 otherwise. Therefore, these four latent variables generate 32 features. As simulated data 1, for these fixed 32 features, 1000, 500, 300, 200, 100 and 50 samples are generated as such, and for each of them, its label is decided by which sides of the hyperplane $3F_1 + 2F_2 + 3F_3 + 2F_4 + F_9 + 1.5F_{15} + 2F_{21} + 2.5F_{27} - 4 = 0$ it lies in as above. Clearly, these 32 features form 10 groups, where for $i = 1, \ldots, 4$, the i-th group contains two features $\{F_{2i-1}, F_{2i}\}$, and for each $j = 1, \ldots, 6$, $\{F_{8+6(i-1)+j} | i = 1, \ldots, 4\}$ forms a group. From the decision hyperplane, it can be seen that the true features are $F_1, F_2, F_3, F_4, F_9, F_{15}, F_{21}, F_{27}$, which correspond to tree true groups $\{F_1, F_2\}, \{F_3, F_4\}, \{F_9, F_{15}, F_{21}, F_{27}\}$. Further, we generate 10, 15 and 20 latent variables as above, which correspond to 200, 450 and 800 features.

Acknowledgements The authors thank the editor and the referees for their valuable comments which have largely improved the presentation of this work. This work is supported by the National Natural Science Foundation of China (No.62066012 and No.12271131), the National Natural Science Foundation of China (Major Program) (No. T2293774), the Hainan Provincial Natural Science Foundation of China

(No.624RC483), and the Key Laboratory of Engineering Modeling and Statistical Computation of Hainan Province.

Author Contributions C.L.: Writing, validation; Y.P.: Review, editing; W.C.: Software, visualization; I.T.: Conceptualization, supervision; Y.S.: Methodology, project administration.

Funding This work is supported by the National Natural Science Foundation of China (No. 62466014, No. 62066012 and No. 12271131), the National Natural Science Foundation of China (Major Program) (No. T2293774) and the Hainan Provincial Natural Science Foundation of China (No. 624RC483).

Data Availability The details for generating simulated data are presented in Appendix A. The Cleveland, WOBC and the emotional recognition data are publically available, and their links are given in the paper.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Code availability The code is available on request from the authors.

References

Bradley, P.S., & Mangasarian, O.L. (1998). Feature selection via concave minimization and support vector machines. In: Proceedings of the 15th International Conference on Machine Learning, pp 82–90.

Butko, T., & Nadeu, C. (2011). Audio segmentation of broadcast news: A hierarchical system with feature selection for the albayzin-2010 evaluation. *2011 IEEE International Conference on Acoustics* (pp. 357–360). Speech and Signal Processing (ICASSP): IEEE.

Cheadle, C., Vawter, M. P., Freed, W. J., et al. (2003). Analysis of microarray data using z score transformation. *The Journal of Molecular Diagnostics, 5*(2), 73–81.

Chu, W., & Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation, 19*(3), 792–815.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Detrano, R., Janosi, A., Steinbrunn, W., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology, 64*(5), 304–310.

Diez, I., Larson, A. G., Nakhate, V., et al. (2021). Early-life trauma endophenotypes and brain circuit-gene expression relationships in functional neurological (conversion) disorder. *Molecular Psychiatry, 26*(8), 3817–3828.

Ding, N., Xu, Y., Tang, Y., et al. (2022). Source-free domain adaptation via distribution estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 7212–7222.

Fang, X., Teng, S., Lai, Z., et al. (2018). Robust latent subspace learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems, 29*(6), 2502–2515.

Feng, W., Wang, Z., Cao, X., et al. (2024). Discriminative sparse subspace learning with manifold regularization. *Expert Systems with Applications, 249*, 123831.

Guo, Y., Zhang, Z., & Tang, F. (2021). Feature selection with kernelized multi-class support vector machine. *Pattern Recognition, 117*, 107988.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., et al. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*, 389–422.

Hastie, T., Tibshirani, R., Friedman, J. H., et al. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Vol. 2). Berlin: Springer.

Herbrich, R. (1999). Support vector learning for ordinal regression. In: *9th International Conference on Artificial Neural Networks: ICANN'99*, IEE.
Jiménez-Cordero, A., Morales, J. M., & Pineda, S. (2021). A novel embedded min-max approach for feature selection in nonlinear support vector machine classification. *European Journal of Operational Research, 293*(1), 24–35.
Khan, M. E. E., & Swaroop, S. (2021). Knowledge-adaptation priors. *Advances in Neural Information Processing Systems, 34*, 19757–19770.
Lal, T.N., Chapelle, O., Weston, J., et al. (2006). Embedded methods. In: *Feature extraction: Foundations and Applications*. Springer, p 137–165.
Li, X., Wang, Y., & Ruiz, R. (2020). A survey on sparse learning models for feature selection. *IEEE Transactions on Cybernetics, 52*(3), 1642–1660.
Neumann, J., Schnörr, C., & Steidl, G. (2005). Combined svm-based feature selection and classification. *Machine Learning, 61*, 129–150.
Pisner, D.A., & Schnyer, D.M. (2020). Support vector machine. In: *Machine Learning*. Elsevier, p 101–121.
Richhariya, B., & Tanveer, M. (2020). A reduced Universum twin support vector machine for class imbalance learning. *Pattern Recognition, 102*, 107150.
Sohrab, F., Raitoharju, J., Gabbouj, M., et al. (2018). Subspace support vector data description. In: *2018 24th International Conference on Pattern Recognition (ICPR)*, pp 722–727.
Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data classification: Algorithms and applications p 37.
Vapnik, V. (2006). *Estimation of Dependences Based on Empirical Data*. Berlin: Springer Science & Business Media.
Vapnik, V., & Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural Networks, 22*(5–6), 544–557.
Vapnik, V., Izmailov, R., et al. (2015). Learning using privileged information: similarity control and knowledge transfer. *Journal of Machine Learning Research, 16*(1), 2023–2049.
Wang, L., Zhu, J., & Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica, 16*, 589–615.
Weston, J., Collobert, R., Sinz, F., et al. (2006). Inference with the Universum. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp 1009–1016.
Yang, Y., & Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing, 25*, 1129–1141.
Yuan, L., Liu, J., & Ye, J. (2011). Efficient methods for overlapping group lasso. *Advances in Neural Information Processing Systems, 24,* 28–37.
Yuan, L., Tay, F.E., Li, G., et al. (2020). Revisiting knowledge distillation via label smoothing regularization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 3903–3911.
Zhai, Y., Ong, Y. S., & Tsang, I. W. (2016). Making trillion correlations feasible in feature grouping and selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(12), 2472–2486.
Zhang, Y., Nam, C. S., Zhou, G., et al. (2018). Temporally constrained sparse group spatial patterns for motor imagery BCI. *IEEE Transactions on Cybernetics, 49*(9), 3322–3332.
Zhu, J., Rosset, S., Tibshirani, R., et al. (2003). 1-norm support vector machines. *Advances in Neural Information Processing Systems, 16*, 433–440.
Zou, H., & Yuan, M. (2008). The F_∞-norm support vector machine. *Statistica Sinica, 18*, 379–398.

Authors and Affiliations

Chunna Li[1] · Yuangang Pan[2] · Weijie Chen[3] · Ivor W. Tsang[2] · Yuanhai Shao[1]

✉ Yuanhai Shao
shaoyuanhai21@163.com

Chunna Li
na1013na@163.com

Yuangang Pan
yuangang.pan@gmail.com

Weijie Chen
wjcper2008@126.com

Ivor W. Tsang
ivor.tsang@gmail.com

[1] School of Mathematics and Statistics, Hainan University, Renmin Avenue No. 58, Haikou 570228, Hainan, China

[2] Center for Frontier AI Research, Agency for Science, Technology and Research (A*STAR), Fusionopolis Way, Singapore 138632, Singapore

[3] Zhijiang College, Zhejiang University of Technology, Yuezhou Avenue No.958, Shaoxing 610101, Zhejiang, China