International Journal of Machine Learning and Cybernetics (2020) 11:2607–2624
https://doi.org/10.1007/s13042-020-01131-5

ORIGINAL ARTICLE

Fast feature selection for interval-valued data through kernel density estimation entropy

Jianhua Dai[1] · Ye Liu[1] · Jiaolong Chen[1] · Xiaofeng Liu[1]

Received: 29 November 2019 / Accepted: 10 April 2020 / Published online: 7 May 2020

Abstract
Kernel density estimation, which is a non-parametric method about estimating probability density distribution of random variables, has been used in feature selection. However, existing feature selection methods based on kernel density estimation seldom consider interval-valued data. Actually, interval-valued data exist widely. In this paper, a feature selection method based on kernel density estimation for interval-valued data is proposed. Firstly, the kernel function in kernel density estimation is defined for interval-valued data. Secondly, the interval-valued kernel density estimation probability structure is constructed by the defined kernel function, including kernel density estimation conditional probability, kernel density estimation joint probability and kernel density estimation posterior probability. Thirdly, kernel density estimation entropies for interval-valued data are proposed by the constructed probability structure, including information entropy, conditional entropy and joint entropy of kernel density estimation. Fourthly, we propose a feature selection approach based on kernel density estimation entropy. Moreover, we improve the proposed feature selection algorithm and propose a fast feature selection algorithm based on kernel density estimation entropy. Finally, comparative experiments are conducted from three perspectives of computing time, intuitive identifiability and classification performance to show the feasibility and the effectiveness of the proposed method.

Keywords Kernel density estimation · Entropy · Feature selection · Kernel function · Interval-valued decision table

1 Introduction

Feature selection is of great practical significance in real life. The purpose of feature selection is to select feature subset that can most effectively represent the decision from feature set of original data. Therefore, we can eliminate some attributes that are not related to decision, reduce the dimension of data, reduce over fitting, and improve the generalization ability of the model. Thus, feature selection has attracted the attentions of many researchers [1–9]. Especially in feature selection in numerical data, some researchers [10, 11] use discrete operation to preprocess numerical data. However, it is worth noting that discretization will lead to the loss of information in data. In order to avoid the discretization of numerical features, we can catch the distribution characteristics of numerical data and estimate the probability density of numerical data.

There are two types of probability density estimation: parametric estimation and non-parametric estimation. As for parametric estimation, it is necessary to assume the probability density model of the data. Then, the parameters in the model are solved by using the given data, and the probability density estimation can be obtained. It ought to note that the probability density function can not well reflect the rules of the experimental data if the hypothetical model does not conform to experimental data. However, the above situation will not occur in non-parametric estimation. Non-parametric estimation does not need to assume the model of experimental data in advance, but directly fits the probability density function in line with the law of the distribution. There are several common methods of nonparametric density estimation, including Histogram estimation [12], Kernel density estimation [13] (shortly KDE), Rosenblatt estimation [14] and so on. Kernel density estimation overcomes discontinuous disadvantage of probability density function in Histogram

✉ Jianhua Dai
jhdai@hunnu.edu.cn

1 Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China

estimation and Rosenblatt estimation, so it has been widely used in many areas [15–24].

Among the applications of kernel density estimation, feature selection is an interesting and successful application. The reason why it is so widely used in feature selection is that it can overcome information loss caused by discretization. Therefore, Kwak et al. [24] proposed a feature selection method on basis of mutual information defined by kernel density estimation. Recently, Xu et al. [25] proposed a semi-supervised feature selection method with kernel purity and kernel density estimation. Zhang et al. [26] proposed a feature selection method in line with kernel density estimation for mixed data. It ought to notice that the above methods don't consider interval-valued data and they can't be used in feature selection in interval-valued data.

As a matter of fact, interval-valued data exist widely in real applications to describe uncertainty [27, 28]. Many scholars have studied interval-valued data from different perspectives. Especially in feature selection, many researchers have studied feature selection for inter-val-valued data. Dai et al. [27, 28] constructed uncertainty measurement and feature selection in interval-valued data. Du et al. [29] put forward an approximation distribution reduct in interval-valued ordered decision tables. Yang et al. [30] proposed an attribute reduction based on α−dominance relation in inter-val-valued information systems. Dai et al. [31] constructed dominance-based fuzzy rough set model via probability approach in interval-valued decision systems and used the model to perform approximation distribution reduct. Guru et al. [32] constructed a novel feature selection model for supervised interval-valued data on basis of K-means clustering. Li [33] put forward multi-level attribute reductions in an interval-valued fuzzy formal context. Dai et al. [34] proposed a heuristic feature selection for interval-valued data based on conditional entropy. Dai et al. [35] introduced a feature selection method in incomplete interval-valued decision systems. Guru et al. [36] presented a feature selection of interval-valued data based on Interval Chi-Square Score.

However, so far, there are very few feature selection methods on basis of kernel density estimation entropy for interval-valued data. Focusing on handling interval-valued data by kernel density estimation entropy, a feature selection method based on kernel density estimation for interval-valued data is proposed in this paper. We first raise the kernel density estimation of interval-valued data, and then propose kernel density estimation probability structure. Based on the structure, kernel density estimation entropies are proposed and used in feature selection for interval-valued data. In addition, we improve the feature selection method and propose a fast feature selection method. Experiments indicate the effectiveness of the proposed feature selection methods for interval-valued data.

The rest of this paper is organized as below. In Sect. 2, the basic concepts of information theory and kernel density estimation are introduced. In Sect. 3, a kernel function for interval-valued data is proposed, and its theoretical properties are studied. In Sect. 4, the interval-valued kernel density estimation probability structure is raised with the proposed kernel function. In Sect. 5, the kernel density estimation information entropy, kernel density estimation conditional entropy and kernel density estimation joint entropy for interval-valued data are constructed by using the raised structure. In Sect. 6, we propose a feature selection method based on kernel density estimation conditional entropy. For improving efficiency of the feature selection method, a fast feature selection algorithm is further presented via the incremental expressions of the kernel function and the inverse of the covariance matrix. In Sect. 7, the validity of the fast feature selection method is verified from aspects of computing time, intuitional identifiability and classification performance by experiments. Section 8 summarizes the paper.

2 Preliminary knowledge

2.1 Basic concepts in information theory

Let X be a discrete random variable with a range of $\mathbb{X}$. $p(x) = p(X = x)$ denotes the probability of occurrence of $X = x$. Information entropy $H(X)$ is defined as below [37]:

$$H(X) = -\sum_{x\in\mathbb{X}} p(x)\log p(x) \tag{1}$$

Information entropy can measure the amount of information needed to eliminate uncertainty. The greater the uncertainty of discrete random variable X is, the greater its information entropy is.

Let X and Y be discrete random variables with ranges of $\mathbb{X}$ and $\mathbb{Y}$, respectively. $p(x, y) = p(X = x, Y = y)$ denotes the joint probability of x and y, then the joint entropy is defined as follows:

$$H(X, Y) = -\sum_{x\in\mathbb{X}}\sum_{y\in\mathbb{Y}} p(x, y)\log p(x, y) \tag{2}$$

Joint entropy can measure the amount of information needed to eliminate the uncertainty in the joint distribution of X and Y. The greater the uncertainty in X and Y is, the greater the joint entropy is.

Let X and Y be discrete random variables with ranges of $\mathbb{X}$ and $\mathbb{Y}$. $p(y|x) = p(Y = y|X = x)$ denotes the probability of $Y = y$ under $X = x$. The definition of conditional entropy is shown as below:

$$H(Y|X) = -\sum_{x\in\mathbb{X}} p(x) \sum_{y\in\mathbb{Y}} p(y|x)\log p(y|x) \tag{3}$$

Conditional entropy can measure the amount of information needed to eliminate uncertainty in Y under condition of X. The more information can X provide about Y, the less uncertainty Y has and the less the conditional entropy is.

The above forms of entropy are all for discrete features. Entropy of continuous features without discrete processing can be written in the form of integral:

$$H(X) = -\int_{x\in\mathbb{X}} p(x)\log p(x)dx \tag{4}$$

Here X is a continuous random variable. $p(x)$ represents the probability density function of a random variable X, and $\mathbb{X}$ denotes the range of X. From Eq. (4), we can see that the key to obtain the entropy of continuous features lies in the probability density function.

2.2 Feature selection

The curse of dimensionality is a problem which occurs in the applications of data mining, pattern recognition and machine learning [38–40]. In most cases, data sets coming from real life have many features in which there may exist irrelevant or redundant features that can consume a lot of computing time and storage space. Feature selection can deal with the problem effectively. Feature selection is to get rid of features which are irrelevant to decision and to select the features which are relevant to decision. In this way, the performances of learning algorithms can be improved.

In this paper, we mainly study the feature selection approach based information theory. In most cases, feature selection method based on information theory use condition entropy $H(D|\boldsymbol{F})$ to evaluate the degree of relevance between features and decision. In condition entropy $H(D|\boldsymbol{F})$, $\boldsymbol{F}$ is a feature set and D denotes the decision. The smaller the value of $H(D|\boldsymbol{F})$ is, the greater relevance between $\boldsymbol{F}$ and D is. Then, we intend to select feature set which have minimum $H(D|\boldsymbol{F})$ in the process of feature selection.

Definition 1 [34] In an information table $< \boldsymbol{U},\boldsymbol{C} >$, $\boldsymbol{U}$ is the nonempty sample set and $\boldsymbol{C}$ is the nonempty feature set. Let $\boldsymbol{F}$ be a selected feature set. For $\forall a, b \in \boldsymbol{C} - \boldsymbol{F}$ and $a \neq b$, if $H(D|\boldsymbol{F} \cup a) < H(D|\boldsymbol{F} \cup b)$, then a is more significant than b relative to decision D.

The detailed process about feature selection based on condition entropy $H(D|\boldsymbol{F})$ can be shown as follows.

Algorithm 1 Feature selection based on conditional entropy

Input: Complete data set $\boldsymbol{U}$, feature set $\boldsymbol{C}$, decision D;maximum number of selected features K; threshold T.
Output: The selected feature subset $\boldsymbol{F}$

```
Set F to an empty set;
min_H = ∞;
while (|F| < K)&&(|ΔH| > T) do;
    Q* = arg min_{Q∈C−F} H  D(F   );
    ΔH = H(D|F) − H(D|F ∪ Q*);
    F = F ∪ Q*;
end while
```

2.3 Kernel density estimation

In one-dimensional continuous real data, the definition of kernel density estimation is as follows:

$$\hat{f}_{h_n}(x) = \frac{1}{nh_n}\sum_{i=1}^{n} K_{h_n}\left(x - x_i\right) = \frac{1}{nh_n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h_n}\right) \tag{5}$$

where h_n denotes the window width; $\lim_{n\to\infty} h_n = 0$; n represents the number of samples; K(.) denotes a kernel function; x_i denotes the ith sample. The common kernel functions are Uniform kernel, Gauss kernel, Epanechnikov kernel and Quadric kernel. Gauss kernel $\Phi(x - x_i, h) = \frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{(x-x_i)^2}{2h^2}\right)$ is most commonly used in kernel density estimation. According to the properties of probability density function, it is realized that the integration of probability density function in definition domain is 1, that is to say, the integration of kernel function in its definition domain is equal to 1:$\int_{x\in\mathbb{X}} K(x)dx = 1$. Bandwidth h plays a smooth role in probability density function. The larger h is, the smoother the curve estimated by kernel density is. On the contrary, the steeper the curve is. From the definition of kernel function, we can see that the kernel density estimation actually calculates the average effect of all sample points on the point x probability density based on the distance. The closer the sample points are to the point x, the greater the contribution to the point x is. On the contrary, the farther the distance is, the smaller the contribution will be.

In m-dimensional continuous real data, the Gauss kernel function is defined as follows:

$$\Phi\left(x - x_i, h\right) = \frac{1}{\left(\sqrt{2\pi}h\right)^m |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{\left(x - x_i\right)^T \Sigma^{-1}\left(x - x_i\right)}{2h^2}\right) \tag{6}$$

where x_i represents ith sample; $\sum$ denotes the m-dimensional sample covariance; $\sum^{-1}$ represents the inverse of the covariance matrix; $|\sum|$ denotes the determinant of the covariance.

Definition 2 [41] In a $t \times t$ dimension covariance matrix

$$\sum_t = \begin{pmatrix} \sum_{t-1} & \boldsymbol{r}_t \\ \boldsymbol{r}_t^T & 1 \end{pmatrix}$$

$\sum_{t-1}$ is the first $t-1$ dimensional matrix of $\sum_t$, $\boldsymbol{r}_t$ is the first $t-1$ row of the t column element. If $\sum_t$ is reversible, the inverse matrix $\sum_t^{-1}$ of $\sum_t$ can be expressed as the following incremental formula:

$$\sum_t^{-1} = \begin{pmatrix} \sum_{t-1}^{-1} & \boldsymbol{0}_t \\ \boldsymbol{0}_t^T & 0 \end{pmatrix} + \frac{1}{\beta_t}\begin{pmatrix} \boldsymbol{b}_t\boldsymbol{b}_t^T & \boldsymbol{b}_t \\ \boldsymbol{b}_t^T & 1 \end{pmatrix} \tag{7}$$

where

$$\begin{cases} \boldsymbol{b}_t = -\sum_{t-1}^{-1}\boldsymbol{r}_t \\ \beta_t = 1 - \boldsymbol{r}_t^T \sum_{t-1}^{-1}\boldsymbol{r}_t = 1 + \boldsymbol{r}_t\boldsymbol{b}_t \end{cases}$$

Lemma 1 *The determinant of the covariance matrix satisfies the following property:*

$$|\sum_t| = \beta_t |\sum_{t-1}| \tag{8}$$

Definition 3 [26] In data set U, the feature set $\boldsymbol{X}$ contains $t-1$ features, where $t \geqslant 2$ and its inverse matrix is expressed as: $\sum_{t-1}^{-1}$. The $\boldsymbol{X}$-feature part of sample x is represented as column vector $\boldsymbol{x}$. When the feature set $\boldsymbol{Z} = \boldsymbol{X} \cup Y$ is obtained by adding feature Y to the feature set $\boldsymbol{X}$, its inverse matrix is expressed as $\sum_t^{-1}$. The $\boldsymbol{Z}$-feature part of sample z is expressed as column vector $z = (\boldsymbol{x}, y) = \left(x_1, x_2, ..., x_{t-1}, y\right)^T$, and the incremental expression of each element in the kernel matrix is expressed as:

$$\phi\left(z_i - z_j, h\right) = \frac{\phi\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right)}{\sqrt{2\pi\beta_t}h \exp\left(\frac{\left((\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{b}_t + (y_i - y_j)\right)^2}{2h^2\beta_t}\right)} \tag{9}$$

3 Kernel density estimation for interval-valued data

Real-valued data can be regarded as a special form of interval-valued data, where the left and right boundaries of the interval form of real-valued data are equal. Inspired by the large contribution of close samples and the small contribution of far samples, the interval-valued Gauss kernel can be constructed.

Definition 4 In an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{U}$ denotes the sample set, $|\boldsymbol{U}| = n$ denotes the base of $\boldsymbol{U}$ is n; $\boldsymbol{C}$ represents the conditional feature set; D denotes the decision feature. Feature values on conditional features are interval values and feature values on decision features are real values. Let $\boldsymbol{A} \subseteq \boldsymbol{C}$ and $|\boldsymbol{A}| = m$, the interval Gaussian kernel function of random interval variable x is defined as follows:

$$\begin{aligned} \boldsymbol{\Phi}(x - x_i, h_n, \boldsymbol{A}) = & \frac{1}{2(\sqrt{2\pi}h_n)^m |\sum_{L,A}|^{\frac{1}{2}}} \\ & \exp\left(-\frac{(x - x_{i,A}^-)^T \sum_{L,A}^{-1}(x - x_{i,A}^-)}{2h_n^2}\right) \\ & + \frac{1}{2(\sqrt{2\pi}h_n)^m |\sum_{R,A}|^{\frac{1}{2}}} \\ & \exp\left(-\frac{(x - x_{i,A}^+)^T \sum_{R,A}^{-1}(x - x_{i,A}^+)}{2h_n^2}\right). \end{aligned} \tag{10}$$

where h_n denotes the window width, $h_n > 0$ and $\lim_{n\to\infty} h_n = 0$; $x_{i,A}^-$ represents the m-dimensional vector formed by the left bound of interval values of the ith sample on the feature set $\boldsymbol{A}$; $x_{i,A}^+$ represents the m-dimensional vector formed by the right bound of interval values of the ith sample in the feature set $\boldsymbol{A}$; $\sum_{L,A}$ is the left-bound covariance of m-dimensional on feature set $\boldsymbol{A}$; $\sum_{R,A}$ is the right-bound covariance of m-dimensional on feature set $\boldsymbol{A}$; $\sum_{L,A}^{-1}$ and $|\sum_{L,A}|$ denote the inverse and the determinant of the left-bounded covariance matrix on feature set $\boldsymbol{A}$; $\sum_{R,A}^{-1}$ and $|\sum_{R,A}|$ denote the inverse and the determinant of the right-bounded covariance matrix on feature set $\boldsymbol{A}$.

We can rewrite Eq. 5 to $\boldsymbol{\Phi}(x - x_i, h_n, \boldsymbol{A}) = L(x - x_i, h_n, \boldsymbol{A}) + R(x - x_i, h_n, \boldsymbol{A})$ where $L(x - x_i, h_n, \boldsymbol{A}) = \frac{1}{(\sqrt{2\pi}h_n)^m |\sum_{L,A}|^{\frac{1}{2}}}$ $\exp(-\frac{(x - x_{i,A}^-)^T \sum_{L,A}^{-1}(x - x_{i,A}^-)}{2h_n^2})$ and $R(x - x_i, h_n, \boldsymbol{A}) = \frac{\exp(-\frac{(x - x_{i,A}^+)^T \sum_{R,A}^{-1}(x - x_{i,A}^+)}{2h_n^2})}{(\sqrt{2\pi}h_n)^m |\sum_{R,A}|^{\frac{1}{2}}}$

Example 1 An interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$ is presented in Table 1 where $\boldsymbol{U} = \{x_1, x_2, x_3, x_4\}$, $\boldsymbol{C} = \{a, b, c\}$. In this example, we set $h = 1/log2(4) = 0.5$ and $A = \{a\}$. Then we can get the following results: $L\left(x_1 - x_2, \frac{1}{2}, A\right) = 0.1080$ $R\left(x_1 - x_2, \frac{1}{2}, A\right) = 0.0003$.

Table 1 An interval-valued decision table

	a	b	c	D
x_1	[1,2]	[2,4]	[1,4]	1
x_2	[2,4]	[2,3]	[3,6]	2
x_3	[2,3]	[2,3]	[3,6]	2
x_4	[1,2]	[2,4]	[1,4]	1

Similarly, we can get the following matrixes:

$$L\left(\frac{1}{2},A\right)=\begin{pmatrix}0.7981 & 0.1080 & 0.1080 & 0.7981\\ 0.1080 & 0.7981 & 0.7981 & 0.1080\\ 0.1080 & 0.7981 & 0.7981 & 0.1080\\ 0.7981 & 0.1080 & 0.1080 & 0.7981\end{pmatrix}$$ and

$$R\left(\frac{1}{2},A\right)=\begin{pmatrix}0.7981 & 0.0003 & 0.1080 & 0.7981\\ 0.0003 & 0.7981 & 0.1080 & 0.0003\\ 0.1080 & 0.1080 & 0.7981 & 0.1080\\ 0.7981 & 0.0003 & 0.1080 & 0.7981\end{pmatrix}$$

Proposition 1 *Interval Gaussian kernel function Eq. 10 has the following properties:*

(1) *Continuity;*
(2) $\boldsymbol{\Phi}\left(x-x_i,h_n,\boldsymbol{A}\right)>0, \forall \boldsymbol{A}\subseteq \boldsymbol{C}$;
(3) *Symmetry:* $\phi\left(x-y,h_n,\boldsymbol{A}\right)=\phi\left(y-x,h_n,\boldsymbol{A}\right)$, $\forall x,y\in U, \forall \boldsymbol{A}\subseteq \boldsymbol{C}$;
(4) $\int \phi\left(x-x_i,h_n,\boldsymbol{A}\right)dx=1, \forall \boldsymbol{A}\subseteq \boldsymbol{C}$;
(5) *Semi-positive definiteness.*

We can notice that the interval-valued Gaussian kernel raised in this paper will be reduced to real-valued Gaussian kernel when the interval values are reduced to real values and $\sum_{L,A}$ and $\sum_{R,A}$ are reversible. From this aspect, we can see that interval kernel function is an extension of classical Gaussian kernel.

Theorem 1 $\forall \boldsymbol{A}\subseteq \boldsymbol{B}\subseteq \boldsymbol{C}$, $\exists \delta>0$, *if* $h_n \geqslant \delta$ *and* $\sum_{L,\boldsymbol{B}}$ *and* $\sum_{R,\boldsymbol{B}}$ *are reversible, then* $\phi(x-x_i,h_n,\boldsymbol{A})\geqslant \phi(x-x_i,h_n,\boldsymbol{B})$.

Proof Let $\boldsymbol{A}\subseteq \boldsymbol{B}\subseteq \boldsymbol{C}$, $\boldsymbol{E}=\boldsymbol{A}+b$, $b\in \boldsymbol{B}$. We can get $\boldsymbol{\Phi}\left(x-x_i,h_n,\boldsymbol{A}\right)=L\left(x-x_i,h_n,\boldsymbol{A}\right)+R\left(x-x_i,h_n,\boldsymbol{A}\right)$. We can first prove the properties on $L(.)$.

Suppose $\sum_{L,\boldsymbol{E}}$ is reversible, we can get $\sum_{L,\boldsymbol{A}}$ is reversible and $\beta_{L,\boldsymbol{E}}>0$ by Eq. 8 and the semi-positive definiteness of covariance matrix. Similarly, when $\sum_{L,\boldsymbol{B}}$ is reversible, we can get $\sum_{L,\boldsymbol{F}}$ is reversible and $\beta_{L,\boldsymbol{F}}>0$ for $\forall \boldsymbol{F}\subseteq \boldsymbol{B}$.

$$L(x-x_i,h_n,\boldsymbol{A})=\frac{\exp(-\frac{(x-x_{i,\boldsymbol{A}}^-)^T\sum_{L,\boldsymbol{A}}^{-1}(x-x_{i,\boldsymbol{A}}^-)}{2h_n^2})}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{A}}|^{\frac{1}{2}}}$$

$$L(x-x_i,h_n,\boldsymbol{E})=\frac{\exp(-\frac{(x-x_{i,\boldsymbol{E}}^-)^T\sum_{L,\boldsymbol{E}}^{-1}(x-x_{i,\boldsymbol{E}}^-)}{2h_n^2})}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{E}}|^{\frac{1}{2}}}$$

Based on Definitions 2, 1 and 3, we can get:

$$L(x-x_i,h_n,\boldsymbol{E})=\frac{\exp\left(-\frac{(x-x_{i,\boldsymbol{E}}^-)^T\sum_{L,\boldsymbol{E}}^{-1}(x-x_{i,\boldsymbol{E}}^-)}{2h_n^2}\right)}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{E}}|^{\frac{1}{2}}}$$
$$=\frac{\exp\left(-\frac{(x-x_{i,\boldsymbol{A}}^-)^T\sum_{L,\boldsymbol{A}}^{-1}(x-x_{i,\boldsymbol{A}}^-))^2}{2h_n^2}\right)}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{A}}|^{\frac{1}{2}}\beta_{L,\boldsymbol{E}}^{\frac{1}{2}}}$$
$$*\exp\left(-\frac{\frac{1}{\beta_{L,\boldsymbol{E}}}((x-x_{i,\boldsymbol{A}}^-)^Tb_{L,\boldsymbol{E}}+(x-x_{i,b}^-))^2}{2h_n^2}\right)$$

$$L(x-x_i,h_n,\boldsymbol{A})-L(x-x_i,h_n,\boldsymbol{E})=\frac{1}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{A}}|^{\frac{1}{2}}}\exp(-\frac{(x-x_{i,\boldsymbol{A}}^-)^T\sum_{L,\boldsymbol{A}}^{-1}(x-x_{i,\boldsymbol{A}}^-)}{2h_n^2})$$
$$\times(1-\frac{1}{\sqrt{2\pi}h_n\beta_{L,\boldsymbol{E}}^{\frac{1}{2}}}\exp(-\frac{((x-x_{i,\boldsymbol{A}}^-)^Tb_{L,\boldsymbol{E}}+(x-x_{i,b}^-))^2}{2h_n^2\beta_{L,\boldsymbol{E}}})).$$

We can see that $\frac{1}{(\sqrt{2\pi}h_n)^m|\sum_{L,\boldsymbol{A}}|^{\frac{1}{2}}}\exp(-\frac{(x-x_{i,\boldsymbol{A}}^-)^T\sum_{L,\boldsymbol{A}}^{-1}(x-x_{i,\boldsymbol{A}}^-)}{2h_n^2})>0$. $\max\{\exp(-\frac{((x-x_{i,\boldsymbol{A}}^-)^Tb_{L,\boldsymbol{E}}+(x-x_{i,b}^-))^2}{2h_n^2\beta_{L,\boldsymbol{E}}})\}=1$ for $\beta_{L,\boldsymbol{E}}>0$ and $h_n>0$. So when $h_n\geqslant\frac{1}{\sqrt{2\pi}\beta_{L,\boldsymbol{E}}^{\frac{1}{2}}}$, $L(x-x_i,h_n,\boldsymbol{A})\geqslant L(x-x_i,h_n,\boldsymbol{E})$. Let $\delta_L=\max\{\frac{1}{\sqrt{2\pi}\beta_{L,\boldsymbol{E}}^{\frac{1}{2}}},\frac{1}{\sqrt{2\pi}\beta_{L,F}^{\frac{1}{2}}},\dots,\frac{1}{\sqrt{2\pi}\beta_{L,\boldsymbol{B}}^{\frac{1}{2}}}\}$ where $\boldsymbol{F}=\boldsymbol{E}+f$, $f\in\boldsymbol{B}$. When $h_n\geqslant\delta_L$, $L(x-x_i,h_n,\boldsymbol{A})\geqslant L(x-x_i,h_n,\boldsymbol{B})$.

Similarly, let $\delta_R=\max\{\frac{1}{\sqrt{2\pi}\beta_{R,\boldsymbol{E}}^{\frac{1}{2}}},\frac{1}{\sqrt{2\pi}\beta_{R,\boldsymbol{F}}^{\frac{1}{2}}},\cdots,\frac{1}{\sqrt{2\pi}\beta_{R,\boldsymbol{B}}^{\frac{1}{2}}}\}$. If $h_n\geqslant\delta_R$, we have $R(x-x_i,h_n,\boldsymbol{A})\geqslant R(x-x_i,h_n,\boldsymbol{B})$.

In summary, when $\delta=max\{\delta_L,\delta_R\}$ and $h_n\geqslant\delta$, $\phi(x-x_i,h_n,\boldsymbol{A})\geqslant\phi(x-x_i,h_n,\boldsymbol{B})$ holds. □

Definition 5 Given an interval-valued decision table $IVDT=<\boldsymbol{U},\boldsymbol{C}\cup D>$, $\boldsymbol{A}\subseteq\boldsymbol{C}$ and $|\boldsymbol{A}|=m$, the probability density function estimation of interval values on $\boldsymbol{A}$ is defined as below:

$$\hat{p}_{\boldsymbol{A}}(x)=\frac{1}{n}\sum_{i\in\boldsymbol{U}}\phi\left(x-x_i,h_n,\boldsymbol{A}\right) \tag{11}$$

Proposition 2 (1)$\int\hat{p}_{\boldsymbol{A}}(x)dx=1$.

Proof It can be proved by Proposition 1. □

4 Kernel density estimation probability structure for interval values

Enlightened by Kwak et al. [24], the conditional probability and joint probability of kernel density estimation for interval values can be defined based on the interval kernel function.

Definition 6 Given an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{A} \subseteq \boldsymbol{C}$ and $|\boldsymbol{A}| = m$. In feature set $\boldsymbol{A}$, the conditional probability of kernel density estimation under $D = d$ is defined as below

$$\hat{p}_{\boldsymbol{A}}(x|d) = \frac{1}{n_d} \sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right) \tag{12}$$

where $I_d = \left\{x_i | \forall x_i \in \boldsymbol{U},\ D(i) = d\right\}$ in which $D(i)$ denotes decision value of ith sample; $n_d = |I_d|$ represents the number of elements in set I_d.

Definition 7 Given an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{A} \subseteq \boldsymbol{C}$ and $|\boldsymbol{A}| = m$. In feature set $\boldsymbol{A}$, the joint probability is defined as follows by Eq. 12:

$$\begin{aligned}\hat{p}_{\boldsymbol{A}}(x, d) &= \hat{p}_{\boldsymbol{A}}(d)\hat{p}_{\boldsymbol{A}}(x|d) \\ &= \frac{n_d}{n}\frac{1}{n_d} \sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right) \\ &= \frac{1}{n} \sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right)\end{aligned} \tag{13}$$

where n denotes the number of samples in sample set $\boldsymbol{U}$.

Definition 8 Given an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{A} \subseteq \boldsymbol{C}$ and $|\boldsymbol{A}| = m$. In feature set $\boldsymbol{A}$, the posterior probability is defined as follows by Eqs. 12 and 13:

$$\begin{aligned}\hat{p}_{\boldsymbol{A}}(d|x) &= \frac{\hat{p}_{\boldsymbol{A}}(x, d)}{\hat{p}_{\boldsymbol{A}}(x)} = \frac{\frac{1}{n}\sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right)}{\frac{1}{n}\sum_{i \in \boldsymbol{U}} \phi\left(x - x_i, h, \boldsymbol{A}\right)} \\ &= \frac{\sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right)}{\sum_{i \in \boldsymbol{U}} \phi\left(x - x_i, h, \boldsymbol{A}\right)}\end{aligned} \tag{14}$$

Proposition 3

(1) $\hat{p}_{\boldsymbol{A}}(x) = \frac{n_d}{n} \sum_{d \in D} \hat{p}_{\boldsymbol{A}}(x|d)$;
(2) $\sum_{d \in D} \hat{p}_{\boldsymbol{A}}(d|x) = 1$;
(3) $\hat{p}_{\boldsymbol{A}}(d) = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{\boldsymbol{A}}\left(d|x_i\right)$;
(4) $\hat{p}_{\boldsymbol{A}}(x) \geqslant \hat{p}_{\boldsymbol{A}}(x, d)$;
(5) $\hat{p}_{\boldsymbol{A}}(x, d) \leqslant \hat{p}_{\boldsymbol{A}}(x|d)$.

Proof (1)

$$\begin{aligned}\frac{n_d}{n} \sum_{d \in D} \hat{p}_{\boldsymbol{A}}(x|d) &= \frac{n_d}{n}\frac{1}{n_d} \sum_{d \in D} \sum_{i \in I_d} \phi(x - x_i, h, \boldsymbol{A}) \\ &= \frac{1}{n} \sum_{i \in U} \phi(x - x_i, h, \boldsymbol{A}).\end{aligned}$$

(2)

$$\begin{aligned}\sum_{d \in D} \hat{p}_{\boldsymbol{A}}(d|x) &= \sum_{d \in D} \frac{\sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right)}{\sum_{i \in U} \phi\left(x - x_i, h, \boldsymbol{A}\right)} \\ &= \frac{\sum_{d \in D}\sum_{i \in I_d} \phi\left(x - x_i, h, \boldsymbol{A}\right)}{\sum_{i \in \boldsymbol{U}} \phi\left(x - x_i, h, \boldsymbol{A}\right)} \\ &= \frac{\sum_{i \in \boldsymbol{U}} \phi\left(x - x_i, h, \boldsymbol{A}\right)}{\sum_{i \in \boldsymbol{U}} \phi\left(x - x_i, h, \boldsymbol{A}\right)}.\end{aligned}$$

(3)

$$\begin{aligned}\hat{p}_{\boldsymbol{A}}(d) &= \int \hat{p}_{\boldsymbol{A}}(d, x)dx \\ &= \int \hat{p}_{\boldsymbol{A}}(x)\hat{p}_{\boldsymbol{A}}(d|x)dx \\ &= \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\end{aligned}$$

(3)It can be proven by Eqs. 13 and 11. (4)It can be proven by Eqs. 13 and 12. □

Theorem 2 *$\exists \delta > 0$, $\forall \boldsymbol{A} \subseteq \boldsymbol{B} \subseteq \boldsymbol{C}$, if $h_n \geqslant \delta$ and $\sum_{L, \boldsymbol{B}}$ and $\sum_{R, \boldsymbol{B}}$ are reversible, then:*

(1) $\hat{p}_{\boldsymbol{A}}(x) \geqslant \hat{p}_{\boldsymbol{B}}(x)$;
(2) $\hat{p}_{\boldsymbol{A}}(x, d) \geqslant \hat{p}_{\boldsymbol{B}}(x, d)$;
(3) $\hat{p}_{\boldsymbol{A}}(x|d) \geqslant \hat{p}_{\boldsymbol{B}}(x|d)$.

Proof It can be proved according to Theorem 1. □

5 Kernel density estimation entropy of interval values

According to the law of large numbers, the information entropy, joint entropy and conditional entropy of kernel density estimation for interval values can be defined.

Given an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{U}$ denotes the sample set. Suppose the samples are independent and subject to the same distribution. $\boldsymbol{A} \subseteq \boldsymbol{C}$ denotes feature subset. $\mathbb{A}$ denotes the value domain of $\boldsymbol{A}$.

Definition 9 The information entropy of interval values is defined as below:

$$\begin{aligned}\hat{H}(\boldsymbol{A}) &= -\int_{x \in \mathbb{A}} \hat{p}_{\boldsymbol{A}}(x) \log \hat{p}_{\boldsymbol{A}}(x)dx \\ &= -\frac{1}{n} \sum_{i \in U} \log \hat{p}_{\boldsymbol{A}}\left(x_i\right)\end{aligned} \tag{15}$$

Theorem 3 $\exists\delta > 0, \forall \boldsymbol{A} \subseteq \boldsymbol{B} \subseteq \boldsymbol{C}$, *if* $h_n \geqslant \delta$ *and* $\sum_{L,B}$ *and* $\sum_{R,B}$ *are reversible, then* $\hat{H}(\boldsymbol{A}) \geqslant \hat{H}(\boldsymbol{B})$.

Proof It can be proved according to Theorem 2. □

Definition 10 The joint entropy of interval values is defined as:

$$\begin{aligned}\hat{H}(\boldsymbol{A},D) &= -\int_{x\in\mathbb{A}} \sum_{d\in D} \hat{p}_{\boldsymbol{A}}(x,d)\log\hat{p}_{\boldsymbol{A}}(x,d)dx \\ &= -\int_{x\in\mathbb{A}} \sum_{d\in D} \hat{p}_{\boldsymbol{A}}(x)\hat{p}_{\boldsymbol{A}}(d|x)\log\hat{p}_{\boldsymbol{A}}(x,d)dx \\ &= -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i,d\right)\end{aligned} \tag{16}$$

Definition 11 The entropy of D under the condition $\boldsymbol{A}$ is defined as follows:

$$\begin{aligned}\hat{H}(D|\boldsymbol{A}) &= \int_{x\in\mathbb{A}} \hat{p}_{\boldsymbol{A}}(x)\hat{H}(D|\boldsymbol{A}=x)dx \\ &= -\int_{x\in\mathbb{A}} \hat{p}_{\boldsymbol{A}}(x)\sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d|x)\log\hat{p}_{\boldsymbol{A}}(d|x)dx \\ &= -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\end{aligned} \tag{17}$$

Conditional entropy $\hat{H}(D|\boldsymbol{A})$ can reflect the correlation between conditional feature set $\boldsymbol{A}$ and decision feature D. The larger the condition entropy is, the smaller the correlation between $\boldsymbol{A}$ and D is. Otherwise, the greater the correlation between $\boldsymbol{A}$ and D is.

Definition 12 The entropy of $\boldsymbol{A}$ under the condition D is defined as follows:

$$\begin{aligned}\hat{H}(\boldsymbol{A}|D) &= \sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d)\hat{H}(\boldsymbol{A}|D=d) \\ &= -\sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d)\int_{x\in\mathbb{A}}\hat{p}_{\boldsymbol{A}}(x|d)\log\hat{p}_{\boldsymbol{A}}(x|d)dx \\ &= -\sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d)\int_{x\in\mathbb{A}}\frac{\hat{p}_{\boldsymbol{A}}(x)\hat{p}_{\boldsymbol{A}}(d|x)}{\hat{p}_{\boldsymbol{A}}(d)}\log\hat{p}_{\boldsymbol{A}}(x|d)dx \\ &= -\sum_{d\in D}\int_{x\in\mathbb{A}}\hat{p}_{\boldsymbol{A}}(x)\hat{p}_{\boldsymbol{A}}(d|x)\log\hat{p}_{\boldsymbol{A}}(x|d)dx \\ &= -\frac{1}{n}\sum_{d\in D}\sum_{i\in U}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i|d\right)\end{aligned} \tag{18}$$

Theorem 4

$$\begin{aligned}\hat{H}(\boldsymbol{A},D) &= \hat{H}(\boldsymbol{A}|D) + H(D) \\ &= \hat{H}(D|\boldsymbol{A}) + \hat{H}(\boldsymbol{A}).\end{aligned}$$

Proof Since $\sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d|x) = 1$ and $\frac{1}{n}\sum_{i\in U}\hat{p}_{\boldsymbol{A}}(d|x_i) = \hat{p}_{\boldsymbol{A}}(d)$, we have:

$$\begin{aligned}\hat{H}(\boldsymbol{A}|D) + H(D) &= -\frac{1}{n}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}(d)\log\hat{p}_{\boldsymbol{A}}(d) \\ &\quad -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i|d\right) \\ &= -\frac{1}{n}\sum_{d\in D}\sum_{i\in U}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}(d) \\ &\quad -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i|d\right) \\ &= -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i,d\right) \\ &= \hat{H}(\boldsymbol{A},D) \\ \hat{H}(D|\boldsymbol{A}) + \hat{H}(\boldsymbol{A}) &= -\frac{1}{n}\sum_{i\in U}\log\hat{p}_{\boldsymbol{A}}\left(x_i\right) \\ &\quad -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(d|x_i\right) \\ &= -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(x_i\right) \\ &\quad -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(d|x_i\right) \\ &= -\frac{1}{n}\sum_{i\in U}\sum_{d\in D}\hat{p}_{\boldsymbol{A}}\left(d|x_i\right)\log\hat{p}_{\boldsymbol{A}}\left(d,x_i\right) \\ &= \hat{H}(\boldsymbol{A},D)\end{aligned}$$

In summary, $\hat{H}(\boldsymbol{A},D) = \hat{H}(\boldsymbol{A}|D) + H(D) = \hat{H}(D|\boldsymbol{A}) + \hat{H}(\boldsymbol{A})$. □

6 Feature selection on basis of kernel density estimation entropy

6.1 Feature selection on basis of kernel density estimation entropy

Based on the definition (see Definition 11) of conditional entropy via interval kernel density estimation, we construct the original algorithm (see Algorithm 2) to calculate conditional entropy. In Step 3, we calculate the inverse of covariance matrix by gaussian elimination [42, 43] whose time complexity is $O(|\boldsymbol{A}|^3)$; the time complexity of the kernel matrix from Step 1 to Step 5 is $O(n^2 * |\boldsymbol{A}|^3)$; from Step 6 to Step 11, the time complexity of conditional entropy is $O(n^2 + n * N_d)$. To sum up, the time complexity of Algorithm 2 is $O(n^2 * |\boldsymbol{A}|^3 + n^2 + n * N_d)$.

Algorithm 2 Original Conditional Entropy calculation for Interval-Valued Data (OCE_IVD)

Input: An interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $|\boldsymbol{U}| = n$; $[1,N_d]$, the value domain of decision feature D; the conditional feature set $\boldsymbol{A}$.
Output: $\hat{H}(D|\boldsymbol{A})$.
1: **for** $i = 1$ to n **do**
2: **for** $j = 1$ to n **do**
3: Based on Eq. 10, we can get $\Phi_{\boldsymbol{A},ij}$;//Computing kernel matrix $\Phi_{\boldsymbol{A}}$ on conditional feature set $\boldsymbol{A}$.
4: **end for**
5: **end for**
6: **for** $k = 1$ to n **do**
7: **for** $l = 1$ to n **do**
p(k)=0;
p(k)=p(k)+$\Phi_{\boldsymbol{A},kl}$;
8: **end for**
9: **for** $d = 1$ to N_d **do**
$p'(k,d) = p'(k,d) + \sum_{b \in I_d} \Phi_{\boldsymbol{A},kb}$;
10: **end for**
H(k)=$\frac{p'(k)}{p(k)} log_2 \frac{p'(k)}{p(k)}$
11: **end for**
$\hat{H}(D|\boldsymbol{A}) = -\sum H(:)/n$
12: return $\hat{H}(D|\boldsymbol{A})$.

Then, we construct a feature selection algorithm (see Algorithm 3) based on conditional entropy of kernel density estimation. The time complexity of Algorithm 3 is $O(K * |\boldsymbol{C}| * (|\boldsymbol{A}|^3 * n^2 + n^2 + n * N_d))$.

Algorithm 3 Original Feature Selection based on Interval-Valued Kernel Density Estimation entropy (OFS_IVKDE)

Input: An interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $|\boldsymbol{U}| = n$; the value domain of decision feature D is $[1,N_d]$; number of features K and stop threshold T.
Output: The selected feature $\boldsymbol{S}$
1: Set $\boldsymbol{S\_X}$, $\boldsymbol{S}$ to an empty set;
2: $min\_H = \infty$;
3: **while** $(|\boldsymbol{S}| < K) \&\& (|\Delta H| > T)$ **do**;
4: $pre\_H = min\_H$;
5: **for** $Q = \boldsymbol{C} - \boldsymbol{S}$ **do**
$\boldsymbol{S\_X} = \boldsymbol{S} \cup Q$;
$new\_H = CE\_IVD(IVDT, \boldsymbol{S\_X})$;
6: **if** $new\_H < \min\_H$ **then**
7: $min\_H = new\_H$;
8: $min\_Q = Q$;
9: **end if**
10: **end for**
11: $\Delta H = min\_H - pre\_H$;
12: $\boldsymbol{S}(end+1) = min\_Q$;
13: **end while**

6.2 Fast feature selection on basis of kernel density estimation entropy

The computation of kernel matrix has high time complexity and low efficiency. Therefore, in this section, we first propose an incremental algorithm for interval-valued data (see Algorithm 4) to calculate kernel matrix and the inverse of covariance matrix. Secondly, we propose a concept of kernel partition matrix and an algorithm (see Algorithm 5) of calculating conditional entropy based on kernel partition matrix for interval-valued data. Finally, based on the above two algorithms, a fast feature selection algorithm (see Algorithm 6) is proposed by interval-valued kernel density estimation entropy.

Algorithm 4 Incremental algorithm for Interval-Valued Data (I_IVD)

Input: An interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $|\boldsymbol{U}| = n$; the selected conditional feature set $\boldsymbol{S}$; a candidate conditional feature Q; left-bound covariance matrix $\sigma^L_{\boldsymbol{S}Q}$ on $\boldsymbol{S}$ and Q; right-bound covariance matrix $\sigma^R_{\boldsymbol{S}Q}$ on $\boldsymbol{S}$ and Q; the left-bound kernel matrix $\boldsymbol{\Phi}^L_{\boldsymbol{S}}$ on $\boldsymbol{S}$; the right-bound kernel matrix $\boldsymbol{\Phi}^R_{\boldsymbol{S}}$ on $\boldsymbol{S}$; inverse of left-bound covariance matrix $\sum^{-1}_{L,\boldsymbol{S}}$ on $\boldsymbol{S}$; inverse of right-bound covariance matrix $\sum^{-1}_{R,\boldsymbol{S}}$ on $\boldsymbol{S}$; width parameter h
Output: $\boldsymbol{\Phi}^L_{\boldsymbol{S\_X}}$; $\boldsymbol{\Phi}^R_{\boldsymbol{S\_X}}$;$\sum^{-1}_{L,\boldsymbol{S\_X}}$; $\sum^{-1}_{R,\boldsymbol{S\_X}}$
1: $\boldsymbol{S\_X} = \boldsymbol{S} + Q$;
2: **if** $|\boldsymbol{S\_X}| == 1$ **then** //Q is the first candidate feature.
3: $\sum^{-1}_{L,\boldsymbol{S\_X}} = 1$;
4: $\sum^{-1}_{R,\boldsymbol{S\_X}} = 1$;
5: Calculate $\boldsymbol{\Phi}^L_{\boldsymbol{S\_X}}$,
$\Phi^L_{\boldsymbol{S\_X},ij} = L(x_i - x_j, h, \boldsymbol{S\_X}) = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{(x_i^- - x_j^-)^2}{2h^2})$, $\forall i, j \in [1,n]$;
6: Calculate $\boldsymbol{\Phi}^R_{\boldsymbol{S\_X}}$,
$\Phi^R_{\boldsymbol{S\_X},ij} = R(x_i - x_j, h, \boldsymbol{S\_X}) = \frac{1}{\sqrt{2\pi}h} \exp(-\frac{(x_i^+ - x_j^+)^2}{2h^2})$, $\forall i, j \in [1,n]$;
7: **else** //Q is not the first candidate feature.
8: $\boldsymbol{rl} = \left(\sigma^L_{\boldsymbol{S}Q}\right)$;
9: $\boldsymbol{rr} = \left(\sigma^R_{\boldsymbol{S}Q}\right)$;
10: $\boldsymbol{bl} = -\sum^{-1}_{L,\boldsymbol{S}} \boldsymbol{rl}$;
11: $\boldsymbol{br} = -\sum^{-1}_{R,\boldsymbol{S}} \boldsymbol{rr}$;
12: $\beta l = 1 + \boldsymbol{rl}^T \boldsymbol{bl}$;
13: $\beta r = 1 + \boldsymbol{rr}^T \boldsymbol{br}$;
14: **if** $\beta l \neq 0$ **then**
15: $\sum^{-1}_{L,\boldsymbol{S\_X}} = \left(\sum^{-1}_{L,\boldsymbol{S}} + \frac{\boldsymbol{bl}\boldsymbol{bl}^T}{\beta l}, \frac{\boldsymbol{bl}}{\beta l}; \frac{\boldsymbol{bl}^T}{\beta l}, \frac{1}{\beta l}\right)$;
16: $\boldsymbol{\Phi}^L_{\boldsymbol{S\_X}}$ can be obtained through $\boldsymbol{\Phi}^L_{\boldsymbol{S}}$, $\forall i, j \in [1,n]$,
$L(x_i - x_j, h, \boldsymbol{S\_X}) = \frac{L(x_i - x_j, h, \boldsymbol{S})}{\sqrt{2\pi}h \cdot \beta l^{\frac{1}{2}} \cdot \exp(\frac{((\boldsymbol{x}^-_{i,\boldsymbol{S}} - \boldsymbol{x}^-_{j,\boldsymbol{S}})^{\boldsymbol{T}} \boldsymbol{bl} + (x^-_{i,Q} - x^-_{j,Q}))^2}{2h^2 \beta l})}$;
17: **else**
18: $\sum^{-1}_{L,\boldsymbol{S\_X}} = 1$;
19: All the elements in the kernel matrix $\boldsymbol{\Phi}^L_{\boldsymbol{S\_X}}$ are 1.
20: **end if**
21: **if** $\beta r \neq 0$ **then**
22: $\sum^{-1}_{R,\boldsymbol{S\_X}} = \left(\sum^{-1}_{R,\boldsymbol{S}} + \frac{\boldsymbol{br}\boldsymbol{br}^T}{\beta r}, \frac{\boldsymbol{br}}{\beta r}; \frac{\boldsymbol{br}^T}{\beta r}, \frac{1}{\beta r}\right)$;
23: $\boldsymbol{\Phi}^R_{\boldsymbol{S\_X}}$ can be obtained through $\boldsymbol{\Phi}^R_{\boldsymbol{S}}$, $\forall i, j \in [1,n]$,
$R(x_i - x_j, h, \boldsymbol{S\_X}) = \frac{R(x_i - x_j, h, \boldsymbol{S})}{\sqrt{2\pi}h \cdot \beta r^{\frac{1}{2}} \cdot \exp(\frac{((x^+_{i,\boldsymbol{S}} - x^+_{j,\boldsymbol{S}})^{\boldsymbol{T}} \boldsymbol{br} + (x^+_{i,Q} - x^+_{j,Q}))^2}{2h^2 \beta r})}$;
24: **else**
25: $\sum^{-1}_{R,\boldsymbol{S\_X}} = 1$;
26: All the elements in the kernel matrix $\boldsymbol{\Phi}^R_{\boldsymbol{S\_X}}$ are 1 ;
27: **end if**
28: **end if**
29: return $\boldsymbol{\Phi}^L_{\boldsymbol{S\_X}}$;$\boldsymbol{\Phi}^R_{\boldsymbol{S\_X}}$;$\sum^{-1}_{L,\boldsymbol{S\_X}}$;$\sum^{-1}_{R,\boldsymbol{S\_X}}$.

In Algorithm 4, if conditional feature Q is the first candidate feature, then the inverse of left bound covariance matrix and right bound covariance matrix are both 1 and the kernel matrix $\mathbf{\Phi}_{\boldsymbol{S}}^{L}$ and $\mathbf{\Phi}_{\boldsymbol{S}}^{R}$ are calculated based on Eq. 10 where $\boldsymbol{S}$ denotes the selected conditional features. If conditional feature Q is not the first candidate feature, then we need to calculate the inverse of covariance matrix based on Eq. 7 and two kernel matrices $\mathbf{\Phi}_{\boldsymbol{S}}^{L}$, $\mathbf{\Phi}_{\boldsymbol{S}}^{R}$ based on Eqs. 9 and 10, respectively. The main cost of Algorithm 4 is the calculation of the kernel matrix, so the time complexity of the algorithm is $O(n^2)$.

Algorithm 5 Conditional Entropy calculation for Interval-Valued Data (CE_IVD)

Input: An interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $|\boldsymbol{U}| = n$; $[1, N_d]$, the value domain of decision feature D; the conditional feature subset $\boldsymbol{A}$; kernel matrix $\mathbf{\Phi}_{\boldsymbol{A}}^{L}$ consisting of left bounds of interval values; kernel matrix $\mathbf{\Phi}_{\boldsymbol{A}}^{R}$ consisting of right bounds of interval values

Output: -$\hat{H}(D|\boldsymbol{A})/n$

1: Create a kernel partition matrix $\mathbf{\Upsilon}(\boldsymbol{A},D)$ and set the element value of the matrix to 0; $\mathbf{\Upsilon}(\boldsymbol{A},D) = \left(\Upsilon_{i,D(j)}(\boldsymbol{A},D)\right)_{n\times N_d}, where \Upsilon_{i,D(j)}(\boldsymbol{A},D) = \Upsilon_{i,D(j)} + \Phi_{\boldsymbol{A},ij}^{L} + \Phi_{\boldsymbol{A},ij}^{R}, \forall i,j \in [1,n]$

2: $\hat{H}(D|\boldsymbol{A}) = 0$;

3: **for** $K = 1$ to n **do**

4: $\quad \hat{p}_{\boldsymbol{A}}(d|x_k) = \frac{\Upsilon_{k,d}(\boldsymbol{A},D)}{|\Upsilon(\boldsymbol{A},D)|_k}$;

5: $\quad \hat{H}(D|\boldsymbol{A}) = \hat{H}(D|\boldsymbol{A}) + \hat{P}_{\boldsymbol{A}}(d|x_k)\log \hat{P}_{\boldsymbol{A}}(d|x_k)$;

6: **end for**

7: return -$\hat{H}(D|\boldsymbol{A})/n$.

In Algorithm 5, the time complexity of Step 1 is $O(n^2)$; the time complexity of Step 3 to Step 6 is $O(n)$. So the total time complexity of the algorithm is $O(n^2 + n)$.

Definition 13 In an interval-valued decision table $IVDT =< \boldsymbol{U}, \boldsymbol{C} \cup D >$, $\boldsymbol{U}$ denotes the sample set and $|\boldsymbol{U}| = n$, $\boldsymbol{A} \subseteq C$. $\boldsymbol{\Phi}(\boldsymbol{A}) = \left(\phi\left(x_i - x_j, h, \boldsymbol{A}\right)\right)_{n\times n} = \left(L\left(x_i - x_j, h, \boldsymbol{A}\right) + R\left(x_i - x_j, h, \boldsymbol{A}\right)\right)_{n\times n}$ is a kernel matrix. The range of decision D is the integer of $\left[1, N_d\right]$. Then the definition of the kernel partition matrix $\boldsymbol{\Upsilon}(\boldsymbol{A},D)$ is as follows:

$$\begin{aligned}\mathbf{\Upsilon}(\boldsymbol{A},D) &= \left(\Upsilon_{i,d}(\boldsymbol{A},D)\right)_{n\times N_d} = \left(\sum_{j=1}^{n}\phi_{ij}m_{jd}\right)_{n\times N_d} \\ &= \left(\sum_{j=1}^{n}\phi\left(x_i - x_j, h, \boldsymbol{A}\right)m_{jd}\right)_{n\times N_d} \\ &= \left(\sum_{j=1}^{n}L\left(x_i - x_j, h, \boldsymbol{A}\right)m_{jd}\right)_{n\times N_d} \\ &\quad + \left(\sum_{j=1}^{n}R\left(x_i - x_j, h, \boldsymbol{A}\right)m_{jd}\right)_{n\times N_d}\end{aligned} \tag{19}$$

where $m_{jd} = \begin{cases} 1 & D(j) = d \\ 0 & D(j) \neq d \end{cases}$.

Example 2 *(Continued from Example 1)* Based on Table 1, we can get $M(D) = (m_{jd})_{4\times 2}$: $M(D) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$. $\left(Y_{1,1}(\boldsymbol{A},D)\right) = 1.5962 + 1.5962 = 3.1924$. Similarly, we can get: $\mathbf{\Upsilon}(\boldsymbol{A},D) = \begin{pmatrix} 3.1924 & 0.3243 \\ 0.2166 & 2.5023 \\ 0.432 & 2.5023 \\ 3.1924 & 0.3243 \end{pmatrix}$.

Re mark 1 $Y_{i,d}(\boldsymbol{A},D) = \sum_{j=1}^{n}\phi\left(x_i - x_j, h, \boldsymbol{A}\right)m_{jd} = \sum_{i\in I_d}\phi\left(x - x_i, h, \boldsymbol{A}\right) = n * \hat{p}_{\boldsymbol{A}}\left(x_i, d\right)$.

Theorem 5 *$|\boldsymbol{\Upsilon}(\boldsymbol{A},D)|_i$ represents the addition of the* ith *row elements of the kernel partition matrix. It satisfies the following property:*

$$|\boldsymbol{\Upsilon}(\boldsymbol{A},D)|_i = \sum_{j\in U}\phi\left(x_i - x_j, h, \boldsymbol{A}\right) = n * \hat{p}_{\boldsymbol{A}}\left(x_i\right)$$

Proof $|\boldsymbol{\Upsilon}(\boldsymbol{A},D)|_i = \sum_{d\in[1,N_d]} \Upsilon_{id}(\boldsymbol{A},D) = \sum_{d\in[1,N_d]}\sum_{j=1}^{n}\phi\left(x_i - x_j, h, \boldsymbol{A}\right)m_{jd} = \sum_{j=1}^{n}\phi\left(x_i - x_j, h, \boldsymbol{A}\right) = n * \frac{1}{n} * \sum_{j=1}^{n}\phi\left(x_i - x_j, h, \boldsymbol{A}\right) = n * \hat{p}_{\boldsymbol{A}}\left(x_i\right)$ □

Remark 2 $\hat{p}_{\boldsymbol{A}}\left(d|x_i\right) = \frac{Y_{i,d}(\boldsymbol{A},D)}{|\boldsymbol{\Upsilon}(\boldsymbol{A},D)|_i}$

Algorithm 6 Fast Feature Selection based on Interval-Valued Kernel Density Estimation entropy (FFS_IVKDE)

Input: An interval-valued decision table $IVDT =< U, C \cup D >, |U| = n$; $[1, N_d]$, the value field of decision feature D; the maximum upper limit of the number of selected features K; threshold T; width parameter h
Output: Selected feature set $\boldsymbol{S}$.
1: Set $\boldsymbol{S\_X}$ to an empty set
2: $\Delta H = inf$
3: **while** $|\boldsymbol{S}| < K \&\& |\Delta H| > T$ **do**
4: $\min\_H = \infty$;
5: $pre\_H = \min\_H$;
6: **for** $Q \in \boldsymbol{C} - \boldsymbol{S}$ **do**
7: $\boldsymbol{S\_X} = \boldsymbol{S} + Q$;
8: $\left(\sum_{L,\boldsymbol{S\_X}}^{-1}, \sum_{R,\boldsymbol{S\_X}}^{-1}, \boldsymbol{\Phi}_{\boldsymbol{S\_X}}^{L}, \boldsymbol{\Phi}_{\boldsymbol{S\_X}}^{R}\right) = I\_IVD\left(Q, \boldsymbol{S}, \sum_{L,\boldsymbol{S}}^{-1}, \sum_{R,\boldsymbol{S}}^{-1}, \boldsymbol{\Phi}_{\boldsymbol{S}}^{L}, \boldsymbol{\Phi}_{\boldsymbol{S}}^{R}, h\right)$;
9: $new\_H = CE\_IVD\left(IVDT, \boldsymbol{\Phi}^{L}, \boldsymbol{\Phi}^{R}, h, N_d\right)$
10: **if** $new\_H < \min\_H$ **then**
11: $\min\_H = new\_H$;
12: $\min\_Q = Q$;
13: $\min\_ivscl = \sum_{L,\boldsymbol{S\_X}}^{-1}$;
14: $\min\_\boldsymbol{\Phi}^{L} = \boldsymbol{\Phi}_{\boldsymbol{S\_X}}^{L}$;
15: $\min\_ivscr = \sum_{R,\boldsymbol{S\_X}}^{-1}$;
16: $\min\_\boldsymbol{\Phi}^{R} = \boldsymbol{\Phi}_{\boldsymbol{S\_X}}^{R}$;
17: **end if**
18: **end for**
19: $\Delta H = \min\_H - pre\_H$;
20: $\sum_{L,\boldsymbol{S}}^{-1} = \min\_ivscl$;
21: $\boldsymbol{\Phi}_{\boldsymbol{S}}^{L} = \min\_\boldsymbol{\Phi}^{L}$;
22: $\sum_{R,\boldsymbol{S}}^{-1} = \min\_ivscr$;
23: $\boldsymbol{\Phi}_{\boldsymbol{S}}^{R} = \min\_\boldsymbol{\Phi}^{R}$;
24: $\boldsymbol{S} = \boldsymbol{S\_X}$;
25: **end while**
26: return $\boldsymbol{S}$.

Algorithm 6 describes a Fast Feature Selection based on Interval-Valued Kernel Density Estimation entropy (shortly *FFS_IVKDE*). Step 8 calculates the inverse of the left bound covariance matrix $\sum_{L,S\_X}^{-1}$ and the right bound covariance $\sum_{R,S\_X}^{-1}$ on conditional feature $\boldsymbol{S\_X}$ where $\boldsymbol{S\_X}$ is the conditional feature set after adding a candidate conditional feature Q. In addition, the kernel matrix $\boldsymbol{\Phi}_{S\_X}^{L}$ about the left bound of interval values and the kernel matrix $\boldsymbol{\Phi}_{S\_X}^{R}$ about the right bound of interval values on the feature set $\boldsymbol{S\_X}$ are calculated. Step 9 calculates the conditional entropy *new_H* on the conditional feature set $\boldsymbol{S\_X}$. Steps 10-17 determine whether the conditional entropy *new_H* on the conditional feature set $\boldsymbol{S\_X}$ is smaller than the conditional entropy min *_H* on the original feature set $\boldsymbol{S}$. If *new_H* is less than min *_H*, then candidate feature Q in $\boldsymbol{S\_X}$ can provide feature information about decision feature D and put Q in the selected feature set $\boldsymbol{S}$. And, based on the time complexity analysis of the above Algorithms 5 and 4, we can get that the time complexity of Algorithm 6 is $O(K * |\boldsymbol{C}| * (n^2 + n))$.

7 Experiments

In order to test the effectiveness of the proposed method, experiments are carried out on 14 data sets. The details of these 14 data sets are shown in Table 2. The first four of them are real-life interval-valued data sets [28, 44, 45]. SRBCT is real-valued data from [46]. Glioma is real-valued data from [47]. The other data sets are real-valued data from UCI [48].

Since the last ten data sets are real-valued data, we need to convert the real-valued data into interval-valued data. The specific operation about above converting is designed as follows: $a_i^- = a_i - \sigma_d$, $a_i^+ = a_i + \sigma_d$ where a_i denotes the ith sample's value on feature $a \in C$, σ_d denotes the standard variance of feature values about samples whose labels are the same as ith sample [49].

In the experiment, we evaluate the effectiveness of the fast feature selection method proposed in this paper from three perspectives: (1). Feature selection via interval-valued kernel density estimation entropy mainly includes three aspects: computing of kernel matrix and the inverse of covariance matrix, computing of conditional entropy, feature selection. In order to test wheth-er the fast feature selection algorithm is faster than the original feature selection algorithm, we compare the running time of two methods from three aspects: the computing time of kernel matrix and the inverse of covariance matrix, the computing time of conditional entropy, and the computing time of feature selection. (2). Sample distribution by first two features selected by our method are compared with that of two features selected randomly. (3). Compare the classification performance of our method with other six comparative methods. Due to the limited number of samples in Fish, Face, Car and Glioma, leave-one-out cross validation is used. Other data sets use 10 fold cross validation.

7.1 Comparison of running time

In order to test wether the fast feature selection algorithm is faster than the original feature selection method, we conduct experiments on three representative data, including Face data of 9 label classes, Hill data of 1212 samples and Colon data of 2000 features. Here we set window width parameter h as $1/log2(n)$ where n denotes the number of samples. In addition, the hardware platform for our experiments is a PC equipped with 12 G main memory and 3.41 GHZ CPU. The software is Matlab (Version R2019a).

The calculation of the kernel matrix and the inverse of covariance matrix in Steps 1-5 of Algorithm 2 is denoted as the Non Incremental strategy of Interval-Valued Data (shortly *NI_IVD*). We compare *NI_IVD* with Algorithm 4, in which the calculation of the kernel matrix and the inverse of covariance matrix is denoted Incremental algorithm of Interval-Valued Data (shortly *I_IVD*). In Fig. 1, we show the time comparison results of *NI_IVD* method and *I_IVD* method on Face, Hill and Colon, where the red line represents *I_IVD* and the black line represents *NI_IVD*. In each node (x, y), x represents the number of features, and y represents time of the method calculating the kernel matrix and inverse of covariance matrix under feature subset $\{a_1, a_2, \cdots, a_x\}$. From Fig. 1, we can find *I_IVD* is much faster than *IN_IVD* on the three data sets. Therefore, our incremental algorithm *I_IVD* can greatly speed up the speed of calculating kernel matrix and the inverse of covariance matrix.

Fig. 2 shows the time comparison results of conditional entropy on Face, Hill and Colon. In Fig. 2, the black line represents the original conditional entropy calculation method (see Algorithm 2), and the red line represents the improved conditional entropy calculation (see Algorithm 5). What's more, in each node (x, y), x represents the number of features, and y represents the running time of the method calculating condition entropy under feature subset $\{a_1, a_2, \cdots, a_x\}$. From the figure, we can see that the calculation time of *CE_IVD* is significantly less than that of *OCE_IVD*. It indicates that algorithm *CE_IVD* is faster than algorithm *OCE_IVD*.

Figure 3 shows the time comparison results of feature selection on Face, Hill and Colon data. Furthermore, in each node (x, y), x represents the number of features and y represents running time of the method selecting feature x. In Fig. 3, the black line represents the original feature selection method (see Algorithm 3), and the red line represents the fast feature selection method (see Algorithm 6). We can see that the red line is much lower than the black line. Therefore, the speed of *FFS_IVKDE* is faster than that of *OFS_IVKDE* in feature selection.

7.2 Intuitive effect

In order to intuitively display the effectiveness of the proposed fast feature selection, we select Face, Iris and Colon to show the intuitive effect of the method. We compare scatter plot constructed by the first two features selected through Algorithm 6 with scatter plot constructed by two random features from original data. Here we set window width parameter h as $3/log2(n)$ where n denotes the number of samples. Figs 4, 5, 6 show the comparison of scatter plots, where each rectangle represents a sample and different colors represent different classes.

Sub-figures (a) in Figs. 4–6 show sample distribution under the first two features selected by our method, while sub-figures (b) in Figs. 4-6 show sample distribution under two random features. The x-axis denotes the first selected feature and the y-axis denotes the second selected feature. We can find that the sample distribution of the first two features selected by our method is clear, while the sample distribution of two random features has many intersections. It suggests that the top two features selected by our method have higher identifiability than the two random features, visually. Hence, the proposed feature selection does be effective.

7.3 Classification performance

Most of the traditional classifiers are for real-valued data. To classify interval-valued data, Dai et al. [28] proposed the extensions of K-Nearest Neighbor(KNN) method and Probabilistic Neural Network(PNN).

Table 2 Interval-valued data sets

Data	Abbreviation	Instances	Features	Classes	Data type
Fish	*Fish*	12	13+1	4	Interval-valued data
Face	*Face*	27	6+1	9	Interval-valued data
Car	*Car*	33	7+1	4	Interval-valued data
Water	*Water*	316	48+1	2	Interval-valued data
Iris	*Iris*	150	4+1	3	Real-valued data
Glass	*Glass*	214	9+1	7	Real-valued data
Wine	*Wine*	178	13+1	3	Real-valued data
Waveform	*Wave*	500	40+1	3	Real-valued data
Hillvalley	*Hill*	1212	100+1	2	Real-valued data
ColonTumor	*Colon*	62	2000+1	2	Real-valued data
SRBCT	*Srbct*	83	2308+1	4	Real-valued data
Glioma	*Glioma*	50	4434+1	4	Real-valued data
TumorsC	*TumorsC*	60	7130+1	2	Real-valued data
LungCancer	*Lung*	96	7129+1	2	Real-valued data

Definition 14 [28] u_i and u_j are two objects of interval-valued information table. $u_i = \left[u_i^{k,-}, u_i^{k,+}\right]$ and $u_j = \left[u_j^{k,-}, u_j^{k,+}\right]$ represent the interval values of object u_i and u_j in kth feature. The distance between u_i and u_j is defined as follows:

$$Dis\left(u_i, u_j\right) = \sqrt{\sum_{k=1}^{m} \left(P_{\left(u_i^k \geqslant u_j^k\right)} - P_{\left(u_i^k \leqslant u_j^k\right)}\right)^2}$$

Where m denotes the number of conditional features and $P_{\left(u_i^k \geqslant u_j^k\right)}$ denotes the possible degree of the interval value u_i

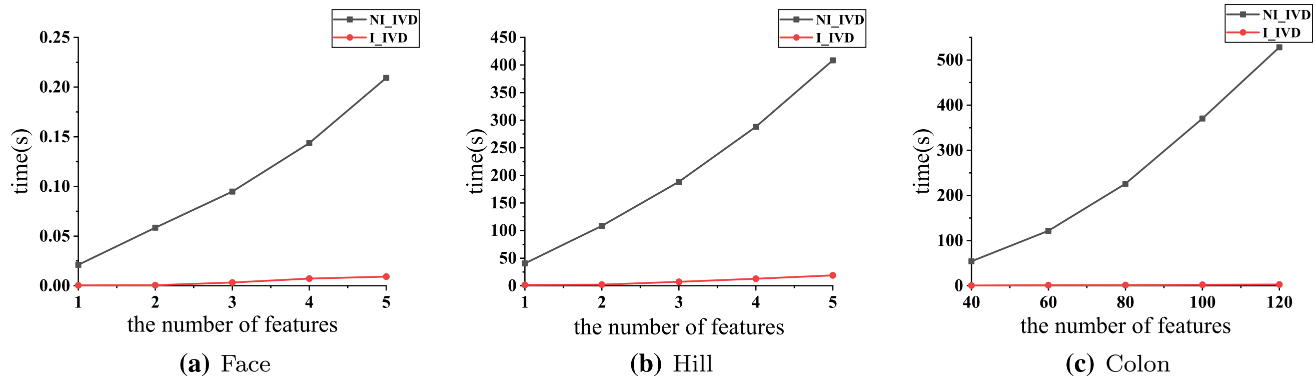

Fig. 1 Comparison of computing time of kernel matrix and inverse on covariance matrix on Face, Hill, Colon

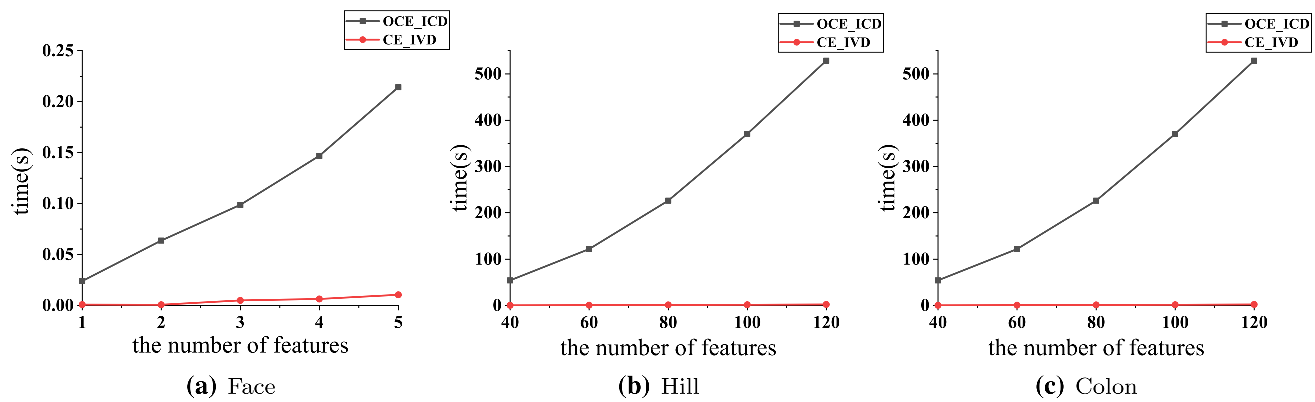

Fig. 2 Comparison of computing time of conditional entropy on Face, Hill, Colon

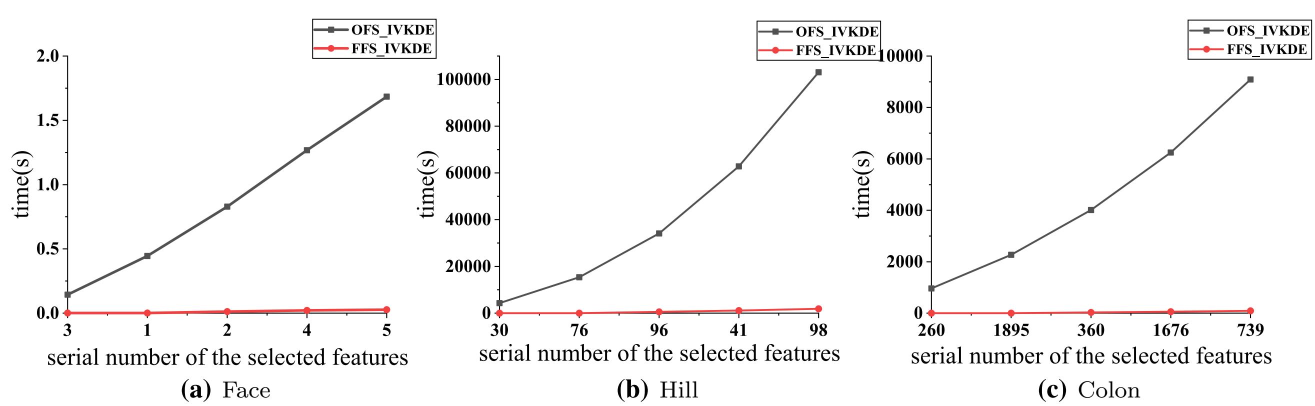

Fig. 3 Comparison of computing time of feature selection on Face, Hill, Colon

relative to the interval value u_j, which is designed as follows: $P_{(u_i^k \geqslant u_j^k)} = \min\{1, \max\{\frac{u_i^{k,+}-u_j^{k,-}}{(u_i^{k,+}-u_i^{k,-})+(u_j^{k,+}-u_j^{b,-})}, 0\}\}$.

In this paper, we compare our method with six representive methods. In [27], a similarity relation between two interval values based on the possible degree of interval value A relative to interval value B was proposed. In [30], the α dominance relation was presented. In [34], the relative bound difference similarity degree between two interval values was proposed. In [50], the α-weak similarity relation between two interval values was proposed. Then we use these four kinds of relations to define conditional entropy similar to [27] for feature selection, and obtain four feature selection methods, namely Feature Selection of the Similarity Relation (FSSR), Feature Selection of α Dominance Relation (FSDR), Feature Selection of the Relative Bound Difference similarity degree (FSRBD) and Feature Selection of the α-Weak Similarity relation (FSWS). Attribute reduction using conditional entropy based on dominance fuzzy rough sets was proposed by [35] , called Attribute Reduction of Dominance Relation (ARDR). Recently, feature selection based on Interval Chi-Square Score was presented by [36], called Feature Selection of Interval Chi-Square Score (FSICSS). We proposed Fast Feature Selection method of the Kernel Density Estimation entropy (FFSKDE) in this paper. The range of parameter θ involved in FSSR, FSDR, FSRBD, FSWS is set to $\{0.4, 0.5, 0.6, 0.7, 0.8\}$. According to literatures [24, 51], the range of parameter h involved in the proposed FFSKDE is set to $\{\frac{1}{log2(n)}, \frac{2}{log2(n)}, \frac{3}{log2(n)}\}$ where n denotes the number of samples. In this experiment, FSSR, FSDR, FSRBD, FSWS, FFSKDE feature selection methods select the optimal classification results within its corresponding parameter range.

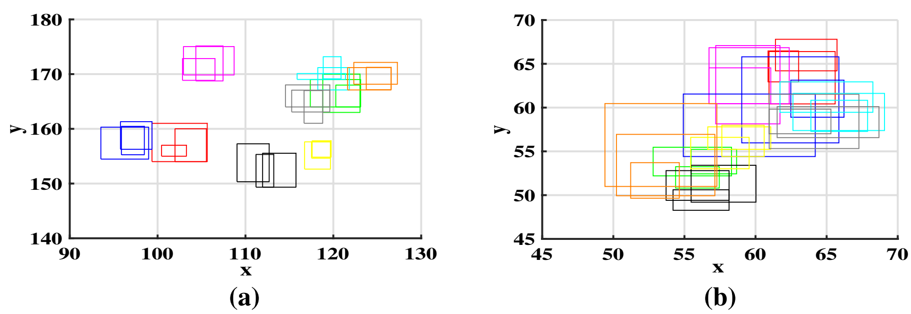

Fig. 4 Comparison of sample distribution on two features from Face Dataset

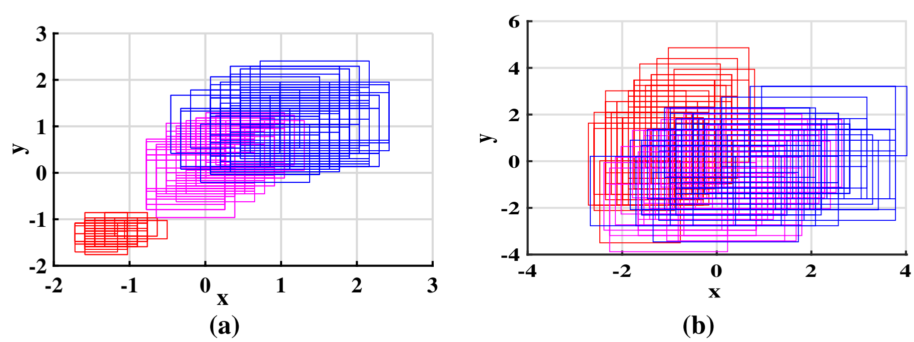

Fig. 5 Comparison of sample distribution on two features from Iris Data set

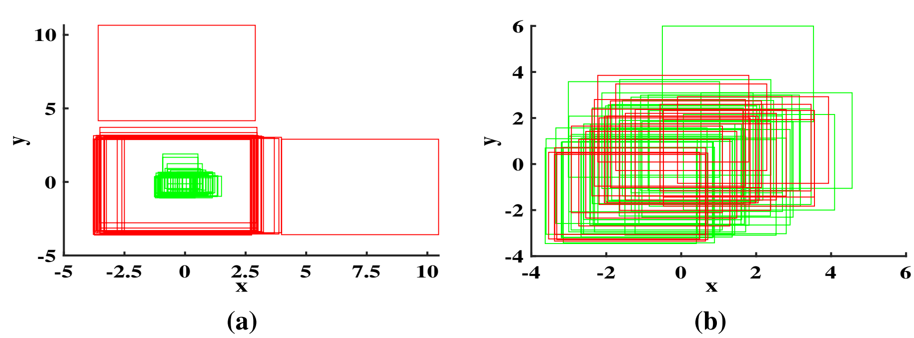

Fig. 6 Comparison of sample distribution on two features from Colon Data set

In the following, we will compare FFSKDE with FSSR,FSDR, FSRBD, FSWS, FSCISS, ARDR and All features on the indexes of accuracy, precision, recall which can comprehensively and well reflect the classification performance of these methods.

First, the accuracy results are shown in Table 3 and Table 4 where the optimal classification accuracies of the data among the seven feature selection methods are represented in bold. From Table 3 and Table 4, the times of being the optimal of FFSKDE method is higher than other six methods on both KNN classifier and PNN classifier. Moreover, in terms of the average classification accuracy on the data sets, FFSKDE method is not only higher than the other six comparative methods, but also higher than the average classification accuracy of ALL features. Especially, in KNN classifier and PNN classifier, only our method has an average classification accuracy of more than 80%. By KNN classifier, our method's average classification accuracy is about 6% higher than the sub-optimal method's average classification accuracy. By PNN classifier, our method's average classification accuracy is about 4% higher than the sub-optimal method's average classification accuracy.

Second, the precision results are displayed in Table 5 and Table 6 where the optimal classification precision results among the seven feature selection methods are denoted in bold. By KNN classifier, we can observe the times when FFSKDE achieves the optimal results is higher than other comparative methods. By PNN classifier, although the times that FFSKDE achieves the best equal to FFSR, the average classification precision of FFSKDE is obviously higher than FFSS. What's more, the average value of classification precision on FFSKDE is not only higher that of other comparative methods, but also higher ALL features.

Finally, the classification recall results are shown in Table 7 and Table 8 where the optimal classification recall results are represented in bold.We can get the same conclusion as classification precision. Then, we can get result that the FFSKDE proposed in this paper performs better than other methods and All features in accuracy, precision and recall. Therefore, the fast feature selection method proposed in this paper is effective.

8 Conclusion

Kernel density estimation technology has been applied in feature selection to avoid discretization for real-valued data. However, there are few studies on feature selection based on kernel density estimation for interval-valued data. Therefore, a feature selection method based on kernel density estimation entropy for interval-valued data is proposed in this paper. Firstly, we raise kernel density estimation of interval-valued data and study its' properties.

Table 3 The accuracy results of KNN

Dataset	FFSKDE	FSSR	FSDR	FSRBD	FSWS	FSICSS	ARDR	All
	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$	$Acc \pm std$
Fish	0.6667 ±0.0000	0.4167 ±0.0000	0.5000 ±0.0000	0.5833 ±0.0000	**0.7500 ±0.0000**	0.4167 ±0.0000	0.3333 ±0.0000	0.5833 ±0.0000
Face	**1.0000 ±0.0000**	**1.0000 ±0.0000**	**1.0000 ±0.0000**	**1.0000 ±0.0000**	**1.0000 ±0.0000**	0.9633 ±0.0000	0.9630 ±0.0000	1.0000 ±0.0000
Car	0.7273 ±0.0000	0.5455 ±0.0000	0.6970 ±0.0000	0.5152 ±0.0000	0.6061 ±0.0000	**0.7656 ±0.0000**	0.6970 ±0.0000	0.6364 ±0.0000
Water	0.7184 ±0.0090	0.7032 ±0.0107	**0.7468 ±0.0060**	0.7180 ±0.0125	0.6832 ±0.0094	0.7405 ±0.0151	0.7142 ±0.0091	0.7633 ±0.0077
Iris	**0.9667 ±0.0032**	0.9493 ±0.0034	0.9600 ±0.0064	0.9533 ±0.0032	0.9487 ±0.0032	0.9533 ±0.0028	0.9447 ±0.0083	0.9547 ±0.0069
Glass	**0.7336 ±0.0163**	0.6136 ±0.0172	0.6308 ±0.0153	0.6822 ±0.0138	0.6794 ±0.0089	0.5981 ±0.0115	0.6818 ±0.0051	0.6710 ±0.0131
Wine	**0.9831 ±0.0055**	0.9736 ±0.0038	0.9494 ±0.0070	0.9691 ±0.0040	0.9736 ±0.0060	0.9405 ±0.0167	0.9612 ±0.0041	0.9792 ±0.0027
Waveform	**0.7640 ±0.0101**	0.7310 ±0.0100	0.5640 ±0.0076	0.7376 ±0.0111	0.5516 ±0.0080	0.7120 ±0.0098	0.5344 ±0.0118	0.7246 ±0.0089
Hill	**0.5743 ±0.0098**	0.5437 ±0.0078	0.5503 ±0.0084	0.5215 ±0.0089	0.5391 ±0.0064	0.5000 ±0.0073	0.5363 ±0.0064	0.5396 ±0.0036
Colon	**0.7742 ±0.0068**	0.7581 ±0.0251	0.7097 ±0.0141	0.6710 ±0.0254	0.6758 ±0.0160	0.6290 ±0.0000	0.6000 ±0.0212	0.6452 ±0.0228
Srbct	0.8313 ±0.0208	0.8313 ±0.0244	0.6988 ±0.0248	0.7229 ±0.0175	0.7108 ±0.0226	0.4458 ±0.0133	**0.9398 ±0.0062**	0.8554 ±0.0213
Clioma	**0.8600 ±0.0000**	0.6400 ±0.0000	0.5800 ±0.0000	0.6600 ±0.0000	0.6600 ±0.0000	0.4800 ±0.0000	0.6800 ±0.0000	0.7200 ±0.0000
Tumors	**1.0000 ±0.0000**	0.9950 ±0.0081	0.6667 ±0.0141	0.6667 ±0.0209	0.6500 ±0.0209	0.8967 ±0.0070	0.6333 ±0.0189	0.5700 ±0.0205
Lung	0.9896 ±0.0073	**1.0000 ±0.0000**	0.8958 ±0.0000	0.9896 ±0.0000	0.8958 ±0.0000	0.8958 ±0.0089	0.9792 ±0.0069	0.8958 ±0.0000
Avg.	**0.8278 ±0.0063**	0.7644 ±0.0079	0.7250 ±0.0074	0.7422 ±0.0084	0.7374 ±0.0072	0.7098 ±0.0066	0.7284 ±0.0070	0.7528 ±0.0077

Table 4 The accuracy results on PNN

Dataset	FFSKDE	FSSR	FSDR	FSRBD	FSWS	FSICSS	ARDR	All
	Acc ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*
Fish	**0.7500 ± 0.0000**	0.5000 ± 0.0000	0.5833 ± 0.0000	0.5833 ± 0.0000	**0.7500 ± 0.0000**	0.5833 ± 0.0000	0.4167 ± 0.0000	0.6667 ± 0.0000
Face	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9630 ± 0.0000	1.0000 ± 0.0000
Car	0.7576 ± 0.0000	0.5758 ± 0.0075	**0.7879 ± 0.0000**	0.5455 ± 0.0000	0.6667 ± 0.0000	0.6970 ± 0.0000	**0.7879 ± 0.0000**	0.6970 ± 0.0000
Water	**0.7405 ± 0.0072**	0.6899 ± 0.0075	0.7152 ± 0.0104	0.7158 ± 0.0070	0.6877 ± 0.0093	0.7278 ± 0.0075	0.6864 ± 0.0106	0.6737 ± 0.1040
Iris	**0.9667 ± 0.0038**	0.9600 ± 0.0054	0.9600 ± 0.0000	0.9647 ± 0.0032	0.9607 ± 0.0049	0.9600 ± 0.0035	0.9600 ± 0.0035	0.9593 ± 0.0021
Glass	0.4159 ± 0.0147	**0.4505 ± 0.0187**	0.4299 ± 0.0229	0.3318 ± 0.0146	0.3720 ± 0.0171	0.2477 ± 0.0083	0.3374 ± 0.0128	0.4514 ± 0.0127
Wine	**0.9719 ± 0.0039**	0.9713 ± 0.0041	0.9607 ± 0.0018	0.9646 ± 0.0027	0.9663 ± 0.0026	0.8202 ± 0.0046	0.9455 ± 0.0027	0.9787 ± 0.0024
Waveform	**0.8200 ± 0.0029**	0.8132 ± 0.0038	0.6160 ± 0.0057	0.8040 ± 0.0041	0.6262 ± 0.0020	0.8040 ± 0.0036	0.6124 ± 0.0065	0.8372 ± 0.0063
Hill	**0.5182 ± 0.0026**	0.5140 ± 0.0025	0.5173 ± 0.0025	0.5124 ± 0.0019	0.5107 ± 0.0034	0.5140 ± 0.0026	0.5097 ± 0.0022	0.5132 ± 0.0045
Colon	**0.7903 ± 0.0127**	0.7709 ± 0.0153	0.7258 ± 0.1080	0.6048 ± 0.0114	0.7177 ± 0.0287	0.1774 ± 0.0156	0.6032 ± 0.0218	0.6565 ± 0.0153
Srbct	0.8916 ± 0.0144	0.8675 ± 0.0128	0.7590 ± 0.0203	0.7373 ± 0.0124	0.7952 ± 0.0161	0.5060 ± 0.0140	**0.9639 ± 0.0062**	0.8273 ± 0.0172
Clioma	**0.8400 ± 0.0000**	0.7400 ± 0.0000	0.6800 ± 0.0000	0.7600 ± 0.0000	0.7600 ± 0.0000	0.4800 ± 0.0000	0.7600 ± 0.0000	0.7200 ± 0.0000
Tumors	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.5333 ± 0.0261	0.5333 ± 0.0261	0.5167 ± 0.0166	0.8671 ± 0.0081	0.5000 ± 0.0294	0.5500 ± 0.0355
Lung	0.9688 ± 0.0033	**1.0000 ± 0.0000**	0.8021 ± 0.0139	0.9896 ± 0.0000	0.4344 ± 0.0860	0.6042 ± 0.0130	0.9698 ± 0.0059	0.8958 ± 0.0000
Avg.	**0.8165 ± 0.0047**	0.7752 ± 0.0055	0.7193 ± 0.0151	0.7177 ± 0.0060	0.6975 ± 0.0133	0.6421 ± 0.0058	0.7154 ± 0.0073	0.7448 ± 0.0143

Table 5 The precision results on KNN

Dataset	FFSKDE	FSSR	FSDR	FSRBD	FSWS	FSICSS	ARDR	All
	Acc ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*
Fish	0.3958 ± 0.0000	0.3750 ± 0.0000	0.3000 ± 0.0000	0.4000 ± 0.0000	**0.8375 ± 0.0000**	0.3631 ± 0.0000	0.3750 ± 0.0000	0.4821 ± 0.0000
Face	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9722 ± 0.0000	0.9722 ± 0.0000	1.0000 ± 0.0000
Car	0.7429 ± 0.0000	0.5545 ± 0.0000	0.7345 ± 0.0000	0.4944 ± 0.0000	0.7121 ± 0.0000	**0.7980 ± 0.0000**	0.7345 ± 0.0000	0.6753 ± 0.0000
Water	0.6877 ± 0.0206	0.6519 ± 0.0173	0.6875 ± 0.0234	0.6321 ± 0.0164	0.5726 ± 0.0207	**0.6980 ± 0.0208**	0.6255 ± 0.0144	0.7118 ± 0.0155
Iris	**0.9738 ± 0.0046**	0.9662 ± 0.0038	0.9544 ± 0.0050	0.9600 ± 0.0035	**0.9738 ± 0.0042**	0.9648 ± 0.0033	0.9544 ± 0.0031	0.9668 ± 0.0029
Glass	0.4626 ± 0.0440	0.3834 ± 0.0168	**0.5480 ± 0.0279**	0.5290 ± 0.0705	0.4274 ± 0.0626	0.3168 ± 0.0243	0.4284 ± 0.0327	0.4692 ± 0.0124
Wine	0.9614 ± 0.0074	**0.9780 ± 0.0045**	0.9498 ± 0.0065	0.9651 ± 0.0035	0.9653 ± 0.0038	0.8579 ± 0.0093	0.9653 ± 0.0035	0.9734 ± 0.0031
Waveform	**0.7702 ± 0.0076**	0.7496 ± 0.0097	0.5606 ± 0.0103	0.7506 ± 0.0084	0.6923 ± 0.0101	0.7309 ± 0.0093	0.5310 ± 0.0069	0.7522 ± 0.0114
Hill	**0.5537 ± 0.0119**	0.5110 ± 0.0083	0.4496 ± 0.0099	0.5075 ± 0.0090	0.4855 ± 0.0108	0.4957 ± 0.0087	0.4665 ± 0.0052	0.5185 ± 0.0089
Colon	0.7418 ± 0.0195	**0.7967 ± 0.0256**	0.4887 ± 0.0839	0.7264 ± 0.0314	0.3197 ± 0.0014	0.3226 ± 0.0000	0.3167 ± 0.0813	0.8279 ± 0.2609
Srbct	0.8393 ± 0.0219	0.7983 ± 0.0125	0.6139 ± 0.0397	0.6961 ± 0.0400	0.7048 ± 0.0399	0.3401 ± 0.0469	**0.9641 ± 0.0058**	0.8610 ± 0.0146
Clioma	**0.8848 ± 0.0000**	0.6068 ± 0.0000	0.5298 ± 0.0000	0.6061 ± 0.0000	0.6061 ± 0.0000	0.6574 ± 0.0000	0.6027 ± 0.0000	0.7733 ± 0.0000
Tumors	**1.0000 ± 0.0000**	0.9875 ± 0.0000	0.3738 ± 0.0921	0.5416 ± 0.2129	0.4831 ± 0.1675	0.9194 ± 0.0048	0.3178 ± 0.0061	0.4121 ± 0.0588
Lung	0.9886 ± 0.0017	**1.0000 ± 0.0000**	0.4479 ± 0.0000	0.9545 ± 0.0192	0.4468 ± 0.0000	0.4479 ± 0.0000	0.9624 ± 0.0040	0.4479 ± 0.0000
Avg.	**0.7859 ± 0.0099**	0.7399 ± 0.0070	0.6170 ± 0.0213	0.6974 ± 0.0296	0.6591 ± 0.0229	0.6346 ± 0.0091	0.6583 ± 0.0116	0.7051 ± 0.0278

Table 6 The precision results on PNN

Dataset	FFSKDE *Acc* ± *std*	FSSR *Acc* ± *std*	FSDR *Acc* ± *std*	FSRBD *Acc* ± *std*	FSWS *Acc* ± *std*	FSICSS *Acc* ± *std*	ARDR *Acc* ± *std*	All *Acc* ± *std*
Fish	0.4917 ± 0.0000	0.4167 ± 0.0000	0.4750 ± 0.0000	0.4208 ± 0.0000	**0.7500 ± 0.0000**	0.4542 ± 0.0000	0.4167 ± 0.0000	0.5417 ± 0.0000
Face	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9722 ± 0.0000	1.0000 ± 0.0000
Car	0.7694 ± 0.0000	0.5871 ± 0.0000	**0.8244 ± 0.0000**	0.6037 ± 0.0000	0.7462 ± 0.0000	0.7631 ± 0.0000	**0.8244 ± 0.0000**	0.7586 ± 0.0000
Water	**0.6733 ± 0.0089**	0.6534 ± 0.0114	0.6387 ± 0.0099	0.6534 ± 0.0037	0.6114 ± 0.0135	0.6358 ± 0.0140	0.6269 ± 0.0057	0.6118 ± 0.0059
Iris	**0.9668 ± 0.0049**	0.9601 ± 0.0054	0.9534 ± 0.0000	**0.9668 ± 0.0046**	0.9600 ± 0.0031	0.9623 ± 0.0044	0.9534 ± 0.0022	0.9668 ± 0.0045
Glass	0.3122 ± 0.0100	**0.4241 ± 0.0110**	0.3382 ± 0.0109	0.3397 ± 0.0136	0.3468 ± 0.0080	0.2996 ± 0.0147	0.2909 ± 0.0120	0.4282 ± 0.0093
Wine	0.9637 ± 0.0026	**0.9712 ± 0.0048**	0.9418 ± 0.0044	0.9607 ± 0.0020	0.9634 ± 0.0034	0.8153 ± 0.0135	0.9470 ± 0.0035	0.9762 ± 0.0020
Waveform	**0.8102 ± 0.0046**	0.8070 ± 0.0037	0.5821 ± 0.0097	0.8035 ± 0.0040	0.7692 ± 0.0092	0.8024 ± 0.0046	0.5772 ± 0.0134	0.8369 ± 0.0058
Hill	0.5210 ± 0.0064	0.5184 ± 0.0043	0.5189 ± 0.0028	0.5160 ± 0.0041	**0.5234 ± 0.0043**	0.5135 ± 0.0029	0.5145 ± 0.0040	0.5112 ± 0.0117
Colon	0.7754 ± 0.0295	**0.7821 ± 0.0232**	0.7038 ± 0.0122	0.7300 ± 0.0293	0.5737 ± 0.0211	0.1643 ± 0.1030	0.6068 ± 0.0155	0.6221 ± 0.0438
Srbct	0.8666 ± 0.0130	0.8343 ± 0.0131	0.7121 ± 0.0190	0.7804 ± 0.0296	0.7837 ± 0.0324	0.3601 ± 0.0182	**0.9585 ± 0.0039**	0.8785 ± 0.0200
Clioma	**0.8711 ± 0.0000**	0.7504 ± 0.0000	0.6939 ± 0.0000	0.7731 ± 0.0000	0.7731 ± 0.0000	0.6782 ± 0.0000	0.7470 ± 0.0000	0.7733 ± 0.0000
Tumors	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.3611 ± 0.0348	0.3435 ± 0.0251	0.3407 ± 0.0271	0.9159 ± 0.0065	0.3851 ± 0.0515	0.4137 ± 0.0321
Lung	0.8401 ± 0.0090	**1.0000 ± 0.0000**	0.5804 ± 0.0147	0.9545 ± 0.0000	0.4254 ± 0.0032	0.4216 ± 0.0018	0.9167 ± 0.0184	0.4479 ± 0.0000
Avg.	**0.7758 ± 0.0064**	0.7646 ± 0.0055	0.6660 ± 0.0085	0.7033 ± 0.0083	0.6834 ± 0.0090	0.6276 ± 0.0131	0.6955 ± 0.0093	0.6976 ± 0.0097

Table 7 The recall results on KNN

Dataset	FFSKDE *Acc* ± *std*	FSSR *Acc* ± *std*	FSDR *Acc* ± *std*	FSRBD *Acc* ± *std*	FSWS *Acc* ± *std*	FSICSS *Acc* ± *std*	ARDR *Acc* ± *std*	All *Acc* ± *std*
Fish	0.4375 ± 0.0000	0.3750 ± 0.0000	0.3750 ± 0.0000	0.4375 ± 0.0000	**0.7500 ± 0.0000**	0.3750 ± 0.0000	0.3125 ± 0.0000	0.5000 ± 0.0000
Face	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9630 ± 0.0000	0.9630 ± 0.0000	1.0000 ± 0.0000
Car	0.7179 ± 0.0000	0.5170 ± 0.0000	0.7018 ± 0.0000	0.4857 ± 0.0000	0.5902 ± 0.0000	**0.7491 ± 0.0000**	0.7018 ± 0.0000	0.6241 ± 0.0000
Water	0.6460 ± 0.0095	0.6312 ± 0.0148	**0.6491 ± 0.0173**	0.6097 ± 0.0121	0.5488 ± 0.0146	0.6302 ± 0.0117	0.6083 ± 0.0117	0.6487 ± 0.0094
Iris	**0.9733 ± 0.0045**	0.9660 ± 0.0038	0.9533 ± 0.0047	0.9600 ± 0.0034	**0.9733 ± 0.0042**	0.9647 ± 0.0032	0.9533 ± 0.0028	0.9667 ± 0.0028
Glass	0.4671 ± 0.0140	0.3907 ± 0.0167	**0.4804 ± 0.0127**	0.3789 ± 0.0174	0.3667 ± 0.0231	0.3059 ± 0.0147	0.4565 ± 0.0147	0.4709 ± 0.0184
Wine	0.9581 ± 0.0097	**0.9767 ± 0.0045**	0.9536 ± 0.0066	0.9673 ± 0.0050	0.9696 ± 0.0042	0.8536 ± 0.0096	0.9709 ± 0.0027	0.9724 ± 0.0041
Waveform	**0.7594 ± 0.0073**	0.7478 ± 0.0107	0.5630 ± 0.0094	0.7487 ± 0.0086	0.6897 ± 0.0116	0.7033 ± 0.0091	0.5374 ± 0.0072	0.7331 ± 0.0113
Hill	**0.5289 ± 0.0104**	0.5099 ± 0.0075	0.4563 ± 0.0087	0.5006 ± 0.0081	0.4876 ± 0.0096	0.4674 ± 0.0073	0.4711 ± 0.0047	0.5149 ± 0.0074
Colon	**0.7102 ± 0.0241**	0.7023 ± 0.0143	0.4977 ± 0.0166	0.6443 ± 0.0214	0.4875 ± 0.0060	0.5000 ± 0.0000	0.4750 ± 0.0127	0.5227 ± 0.0117
Srbct	0.7241 ± 0.0224	0.7836 ± 0.0195	0.5196 ± 0.0336	0.6097 ± 0.0303	0.6102 ± 0.0339	0.3196 ± 0.0183	**0.9361 ± 0.0166**	0.7937 ± 0.0167
Clioma	**0.8810 ± 0.0000**	0.5786 ± 0.0000	0.5274 ± 0.0000	0.6131 ± 0.0000	0.6131 ± 0.0000	0.5048 ± 0.0000	0.6107 ± 0.0000	0.7036 ± 0.0000
Tumors	**1.0000 ± 0.0000**	0.9762 ± 0.0000	0.0489 ± 0.0109	0.5000 ± 0.0198	0.4989 ± 0.0152	0.8214 ± 0.0125	0.4705 ± 0.0242	0.4714 ± 0.0204
Lung	0.9000 ± 0.0158	**1.0000 ± 0.0000**	0.5000 ± 0.0000	0.9942 ± 0.0025	0.4884 ± 0.0000	0.0125 ± 0.0000	0.6500 ± 0.0394	0.5000 ± 0.0000
Avg.	**0.7645 ± 0.0084**	0.7254 ± 0.0066	0.5878 ± 0.0086	0.6750 ± 0.0092	0.6481 ± 0.0087	0.5836 ± 0.0062	0.6512 ± 0.0098	0.6730 ± 0.0073

Table 8 The recall results of PNN

Dataset	FFSKDE	FSSR	FSDR	FSRBD	FSWS	FSICSS	ARDR	All
	Acc ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*	*Acc* ± *std*
Fish	0.6250 ± 0.0000	0.5000 ± 0.0000	0.5000 ± 0.0000	0.5000 ± 0.0000	**0.8125 ± 0.0000**	0.5625 ± 0.0000	0.4375 ± 0.0000	0.6250 ± 0.0000
Face	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9630 ± 0.0000	1.0000 ± 0.0000
Car	0.7580 ± 0.0000	0.5464 ± 0.0000	**0.7893 ± 0.0000**	0.5214 ± 0.0000	0.6402 ± 0.0000	0.6821 ± 0.0000	**0.7893 ± 0.0000**	0.6821 ± 0.0000
Water	**0.6652 ± 0.0065**	0.6629 ± 0.0136	0.6437 ± 0.0090	0.6590 ± 0.0039	0.6388 ± 0.0162	0.6378 ± 0.0144	0.6307 ± 0.0053	0.6280 ± 0.0076
Iris	**0.9667 ± 0.0049**	0.9600 ± 0.0054	0.9533 ± 0.0000	**0.9667 ± 0.0047**	0.9600 ± 0.0031	0.9620 ± 0.0045	0.9533 ± 0.0021	0.9667 ± 0.0044
Glass	0.3630 ± 0.0127	**0.4662 ± 0.0135**	0.4075 ± 0.0102	0.3849 ± 0.0163	0.3858 ± 0.0151	0.3368 ± 0.0237	0.3477 ± 0.0088	0.4914 ± 0.0127
Wine	0.9709 ± 0.0020	**0.9733 ± 0.0047**	0.9521 ± 0.0039	0.9607 ± 0.0024	0.9718 ± 0.0027	0.8239 ± 0.0126	0.9568 ± 0.0030	0.9803 ± 0.0024
Waveform	**0.8120 ± 0.0046**	0.8084 ± 0.0037	0.5948 ± 0.0059	0.8051 ± 0.0040	0.7624 ± 0.0059	0.8027 ± 0.0045	0.5924 ± 0.0090	0.8379 ± 0.0058
Hill	0.5116 ± 0.0035	0.5099 ± 0.0023	0.5107 ± 0.0016	0.5091 ± 0.0023	**0.5784 ± 0.0023**	0.5048 ± 0.0018	0.5083 ± 0.0023	0.5074 ± 0.0078
Colon	0.7557 ± 0.0238	**0.8068 ± 0.0253**	0.7216 ± 0.0134	0.7498 ± 0.0314	0.5132 ± 0.0225	0.1350 ± 0.0129	0.6136 ± 0.0154	0.5534 ± 0.0248
Srbct	0.8775 ± 0.0188	0.8495 ± 0.0151	0.7466 ± 0.0147	0.7885 ± 0.0251	0.7863 ± 0.0222	0.3277 ± 0.0105	**0.9703 ± 0.0059**	0.8662 ± 0.0229
Clioma	**0.8595 ± 0.0000**	0.7202 ± 0.0000	0.6681 ± 0.0000	0.7560 ± 0.0000	0.7560 ± 0.0000	0.5378 ± 0.0000	0.7537 ± 0.0000	0.7036 ± 0.0000
Tumors	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.3927 ± 0.0275	0.3821 ± 0.0251	0.3835 ± 0.0167	0.8119 ± 0.0176	0.3870 ± 0.0461	0.4269 ± 0.0283
Lung	0.9267 ± 0.0264	**1.0000 ± 0.0000**	0.6410 ± 0.0258	0.9942 ± 0.0000	0.3314 ± 0.0155	0.3128 ± 0.0086	0.9884 ± 0.0037	0.5000 ± 0.0000
Avg.	**0.7923 ± 0.0074**	0.7717 ± 0.0060	0.6801 ± 0.0080	0.7127 ± 0.0082	0.6800 ± 0.0087	0.6027 ± 0.0079	0.7066 ± 0.0073	0.6978 ± 0.0083

The kernel density estimation probability structure is constructed. By the constructed structure, a series of kernel density estimation entropies are defined. Further we present a fast feature selection method by kernel partition matrix, incremental expressions of kernel matrix and inverse of covariance matrix. Experiments are conducted to verify the proposed approach. The results show that the proposed fast feature selection method is efficient.

It is worth noting that the proposed fast feature selection algorithm doesn't consider the correlation among the selected features. Therefore, in the future work, we will construct an improved feature selection method via introducing the concept of mutual information which can not only evaluate the correlation between the selected features and decision feature, but also evaluate the correlation among the selected features. However, how to construct the mutual information by kernel density estimation is challenging. In the future, we intend to study this issue.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 61976089, No. 61473259, No. 61070074, No. 60703038), and the Hunan Provincial Science & Technology Project Foundation (2018TP1018, 2018RS3065).

References

1. Javidi MM, Eskandari S (2018) Streamwise feature selection: a rough set method. Int J Mach Learn Cybernet 9(4):667–676
2. Li JZ, Yang XB, Song XN, Wang PX, Yu DJ (2019) Neighborhood attribute reduction: a multi-criterion approach. Int J Mach Learn Cybernet 10(4):731–742
3. Dai JH, Hu QH, Hu H, Huang DB (2018) Neighbor inconsistent pair selection for attribute reduction by rough set approach. IEEE Trans Fuzzy Syst 26(2):937–950
4. Shang RH, Chang JW, Jiao LC, Xue Y (2019) Unsupervised feature selection based on self-representation sparse regression and local similarity preserving. Int J Mach Learn Cybernet 10(4):757–770
5. Dai JH, Hu QH, Zhang JH, Hu H, Zheng NG (2017) Attribute selection for partially labeled categorical data by rough set approach. IEEE Trans Cybernet 47(9):2460–2471
6. Dai JH (2013) Rough set approach to incomplete numerical data. Inf Sci 240:43–57
7. Wang CZ, Qi YL, Shao MW, Hu QH, Chen DG, Qian YH, Lin YJ (2017) A fitting model for feature selection with fuzzy rough sets. IEEE Trans Fuzzy Syst 25(4):741–753
8. Dai JH, Hu H, Wu WZ, Qian YH, Huang DB (2018) Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets. IEEE Trans Fuzzy Syst 26(4):2174–2187
9. Zhang X, Mei CL, Chen DG, Li JH (2016) Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy. Pattern Recogn 56:1–15
10. Dai JH, Xu Q (2013) Attribute selection based on information gain ratio in fuzzy rough set theory with application to tumor classification. Appl Soft Comput 13(1):211–221
11. Dai JH, Han HF, Hu QH, Liu MF (2016) Discrete particle swarm optimization approach for cost sensitive attribute reduction. Knowl-Based Syst 102:116–126

12. Ashour AS, Guo Y, Kucukkulahli E, Erdogmus P, Polat K (2018) A hybrid dermoscopy images segmentation approach based on neutrosophic clustering and histogram estimation. Appl Soft Comput 69:426–434
13. Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 3(33):1065–1076
14. Rosenblatt M (1956) Remarks on some nonparametric estimates of a density function. Ann Math Stat, pp 832–837
15. Banerjee A, Burlina P (2010) Efficient particle filtering via sparse kernel density estimation. IEEE Trans Image Process 19(9):2480–2490
16. Cai XJ, Wu ZF, Cheng J (2012) Using kernel density estimation to assess the spatial pattern of road density and its impact on landscape fragmentation. Int J Geogr Inf Sci 27:1–9
17. Qian PJ, Wang ST, Deng ZH (2011) Fast adaptive similarity-based clustering using sparse parzen window density estimation. Acta Autom Sin 37(2):179–187
18. Rouhani M, Mohammadi M, Kargarian A (2016) Parzen window density estimator-based probabilistic power flow with correlated uncertainties. IEEE Trans Sustain Energy 7(3):1170–1181
19. Schller H, Hartmann U (1992) Mapping neural network derived from the parzen window estimator. Neural Netw 5(6):903–909
20. Wang S, Chung F, Xiong F (2008) A novel image thresholding method based on parzen window estimate. Pattern Recogn 41(1):117–129
21. Wang SC, Gao R, Wang LM (2016) Bayesian network classifiers based on gaussian kernel density. Expert Syst Appl 51:207–217
22. Yang SS, Zheng F, Luo X, Cai SX, Wu YF, Liu KZ, Wu MH, Chen J, Krishnan S (2014) Effective dysphonia detection using feature dimension reduction and kernel density estimation for patients with parkinsons disease. PLoS ONE 9(2):e88825
23. Yu WH, Ai TH, Shao SW (2015) The analysis and delimitation of central business district using network kernel density estimation. J Transp Geogr 45:32–47
24. Kwak N, Choi CH (2002) Input feature selection by mutual information based on parzen window. IEEE Trans Pattern Anal Mach Intell 24(12):1667–1671
25. Xu SQ, Dai JH, Shi H (2018) Semi-supervised feature selection by mutual information based on kernel density estimation. In: 24th international conference on pattern recognition (ICPR), pp 818–823
26. Zhang JH (2017) Kernel density estimation entropy for mixed data and fast greedy feature selection algorithms. Master's thesis, Zhejiang university
27. Dai JH, Wang WT, Xu Q, Tian HW (2012) Uncertainty measurement for interval-valued decision systems based on extended conditional entropy. Knowl-Based Syst 27:443–450
28. Dai JH, Wang WT, Mi JS (2013) Uncertainty measurement for interval-valued information systems. Inf Sci 251:63–78
29. Du WS, Hu BQ (2014) Approximate distribution reducts in inconsistent interval-valued ordered decision tables. Inf Sci 271:93–114
30. Yang XB, Qi Yong YDJ, Yu HL, Yang JY (2015) α-Dominance relation and rough sets in interval-valued information systems. Inf Sci 294:334–347
31. Dai JH, Zheng GJ, Han HF, Hu QH, Zheng NG, Liu J, Zhang QL (2017) Probability approach for interval-valued ordered decision systems in dominance-based fuzzy rough set theory. J Intell Fuzzy Syst 32(1):701–703
32. Guru DS, Kumar NV, Suhil M (2017) Feature selection of interval valued data through interval K-means clustering. Int J Comput Vis Image Process 7:64–80
33. Li LF (2017) Multi-level interval-valued fuzzy concept lattices and their attribute reduction. Int J Mach Learn Cybernet 8(1):45–56
34. Dai JH, Hu H, Zheng GJ, Hu QH, Han HF, Shi H (2016) Attribute reduction in interval-valued information systems based on information entropies. Front Inf Technol Electron Eng 17(9):919–928
35. Dai JH, Yan YJ, Li ZW, Liao BS (2018) Dominance-based fuzzy rough set approach for incomplete interval-valued data. J Intell Fuzzy Syst 34:423–436
36. Guru DS, Kumar NV (2020) Interval chi-square score (ICSS): feature selection of interval valued data. Adv Intell Syst Comput 941:686–698
37. Gatenby RA, Frieden BR (2008) Inf Theory and Entropy. Springer, New York
38. Wang XZ, Xing HJ, Li Y, Hua Q, Dong CR, Pedrycz W (2015) A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning. IEEE Trans Fuzzy Syst 23(5):1638–1654
39. Wang R, Wang XZ, Kwong S, Xu C (2017) Incorporating diversity and informativeness in multiple-instance active learning. IEEE Trans Fuzzy Syst 25(6):1460–1475
40. Wang XZ, Wang R, Xu C (2018) Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. IEEE Trans Cybernet 48(2):703–715
41. Zhang GL, Shen H, Shi F, Huo YQ (2015) Block iterative inversion algorithms for large real symmetric matrix. Wirel Interconnect Technol 6:127–129
42. Grcar J (2011) Mathematicians of Gaussian elimination. Not Am Math Soc 58(6):782–792
43. Stanimirović PS, Petković MD (2013) Gauss-Jordan elimination method for computing outer inverses. Appl Math Comput 219(9):4667–4679
44. Hedjazi L, Aguilar MJ, Lann MVL (2011) Similarity-margin based feature selection for symbolic interval data. Pattern Recogn Lett 32(4):578–585
45. Quevedo J, Puig V, Cembrano G, Blanch J, Aguilar J, Saporta D, Benito G, Hedo M, Molina A (2010) Validation and reconstruction of flow meter data in the barcelona water distribution network. Control Eng Pract 18(6):640–651
46. Khan J, Wei JS, Ringnér M, Lao HS, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nat Med 7(6):673–679
47. Li JD, Cheng KW, Wang SH, Morstatter F, Trevino RP, Tang JL, Liu H (2018) Feature selection: a data perspective. ACM Comput Surv 9(4):1–45
48. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
49. Zhang YY, Li TR, Luo C, Zhang JB, Chen HM (2016) Incremental updating of rough approximations in interval-valued information systems under attribute generalization. Inf Sci 373:461–475
50. Dai JH, Wei BJ, Zhang XH, Zhang QL (2017) Uncertainty measurement for incomplete interval-valued information systems based on α-weak similarity. Knowl-Based Syst 136:159–171
51. He DC, Zhang HJ, Hao WN, Zhang R (2015) A robust parzen window mutual information estimator for feature selection with label noise. Intell Data Anal 19:1199–1212