Int J Software Informatics, Vol.2, No.2, December 2008, pp. 125–139 E-mail: ijsi@iscas.ac.cn
International Journal of Software and Informatics, ISSN 1673-7288 http://www.ijsi.org
 Tel: +86-10-62661040

Attribute Selection for Numerical Databases that Contain Correlations*

Taufik Djatna[1] and Yasuhiko Morimoto[2]

[1](Dept. of Information Engineering, Hiroshima University Japan and Dept. of Agroindustrial Technology, Bogor Agricultural University Indonesia, taufikdjatna@ipb.ac.id)

[2](Hiroshima University Japan, morimoto@mis.hiroshima-u.ac.jp)

Abstract There are many correlated attributes in a database. Conventional attribute selection methods are not able to handle such correlations and tend to eliminate important rules that exist in correlated attributes. In this paper, we propose an attribute selection method that preserves important rules on correlated attributes. We first compute a ranking of attributes by using conventional attribute selection methods. In addition, we compute two-dimensional rules for each pair of attributes and evaluate their importance for predicting a target attribute. Then, we evaluate the shapes of important two-dimensional rules to pick up hidden important attributes that are under-estimated by conventional attribute selection methods. After the shape evaluation, we re-calculate the ranking so that we can preserve the important correlations. Intensive experiments show that the proposed method can select important correlated attributes that are eliminated by conventional methods.

Key words: Feature Selection; Correlated Attribute; Two Dimensional Rule; Region Shape

Djatna T, Morimoto Y. Attribute selection for numerical databases that contain correlations. *Int J Software Informatics*, **2008, 2(2): 125–139. http://www.ijsi.org/1673-7288/2/125.pdf**

1 Introduction

Attribute selection, also known as feature selection, is an important technique to reduce computational costs for analyzing a target attribute in a database. For example, assume that we want to know the risks of a certain disease from medical database records. Each record in the database has attributes, say *conditional attributes*, that contains various diagnosis results and also contains an attribute, say *target attribute*, that indicates whether the patient becomes sick or not. If there are many conditional attributes in the database, elimination of conditional attributes that are irrelevant to the target attribute is important for an intensive analysis[2,3].

In ideal subset of attributes, each conditional attribute is highly correlated to the target attribute but is not correlated to other conditional attributes[4,12,14]. Thus conventional attribute selection methods select attributes that are highly correlated to the target attribute and try to eliminate attributes that are correlated to the

* This work was supported by KAKENHI (#19500123). Taufik Djatna was supported by scholarship of MEXT Japan.
Corresponding author: Taufik Djatna, Email: taufikdjatna@ipb.ac.id
Manuscript received 14 Oct., 2008; revised 4 Dec., 2008; accepted 19 Dec., 2008; published online 29 Dec., 2008

selected attributes. Such conventional methods did not pay attention to importance of correlations among conditional attributes.

In real databases, there are many correlated attributes. For example, in a medical diagnosis records, to analyze risks of metabolic diseases, we have to look both "weight" and "height" of patients to find out whether she or he is overweight or not. In this case these two attributes are said to be correlated attributes, as we can not predict the risk by using one of the two attributes. Therefore, it is very important to take into account such correlations.

Conventional attribute selection methods ignore such important correlation and make it difficult to find the important knowledge that may exists in correlated attributes, which also decreases classification or regression accuracy. In this paper, we propose new attribute selection method that preserves such important correlated rules on a pair of conditional attributes.

Figure 1 is an example of an important correlated rule in the diagnosis records. The grey region indicates patients whose risk of disease is low. It means if patient's weight and height lies on this region, his/her risk is much lower than that of patients lie outside the region.

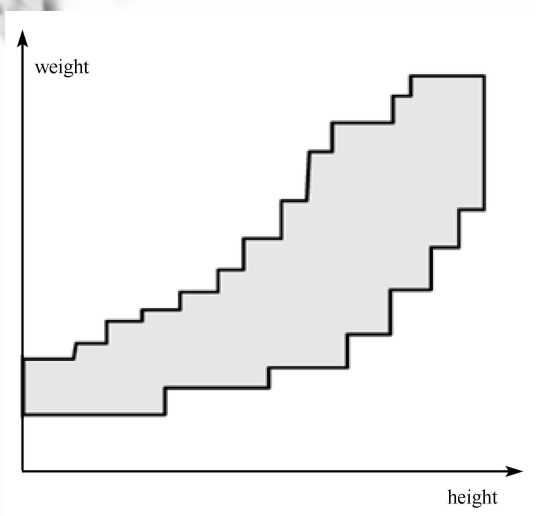

Figure 1. Low risk region

Each attribute is independently considered in conventional methods. In fact, when we observe only "height", it is difficult to decide whether a patient is risky or not, since there are both risky and non-risky area in any height values. Conventional methods tend to eliminate such attributes like "height". However, it is one of the important attribute of the correlated rule for analyzing the risk and should not be eliminated.

Our attribute selection processes as follows:

1. We evaluate each single attribute based on relevance to the target attributes and rank the attributes by using conventional method.

2. We compute two dimensional rules like Fig.1 for each pair of attributes in the database.

3. We also rank the two dimensional rules based on the relevance to the target and select some of the rules as significant.

4. We evaluate the shape of each significant two dimensional rule to find whether both of the axis (attributes) of the region are necessary or not.

5. We re-rank the attributes so that the necessary axis (attributes) of significant two dimensional rules can be preserved.

Figure 2 shows examples of two-dimensional rules. If shape of a rule is like (a), the importance of "att2" is very low since we can judge the rule by "att1" only. On the other hand, if shape of a rule is like (b), both attributes are necessary since we have to look at both values to judge the rule (to judge whether it is inside the region or not). Note that the edge value of "att2" of the rule varies widely. Conventional attribute selection tends to under-estimates the importance of such attributes like "att2" if we look the attribute independently. If both attributes contribute to the two dimensional rule, we prioritize such attributes if they are under-estimated.

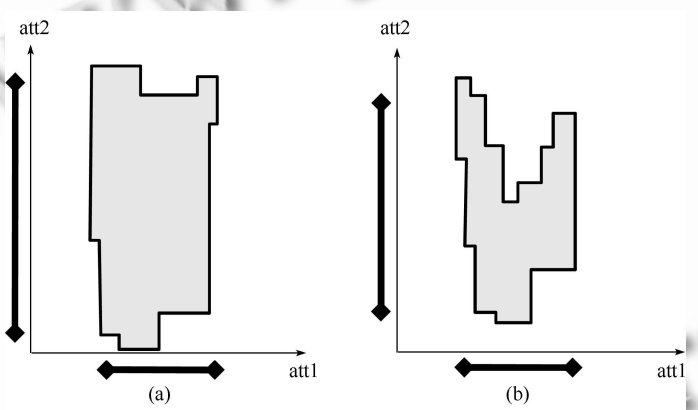

Figure 2. Shape examples

In this paper, we propose a new attribute selection method which can preserve such important correlated attributes that is under-estimated by conventional attribute selections. For example, both of "weight" and "height" in Fig.1 tend to be eliminated in conventional attribute selection, while our method can select those attributes.

This paper is organized as follows: Section 2 defines the attribute selection problem for databases that contain correlated attributes. Section 3 describes how we rank attributes and preserve important correlations in detail. Section 4 discusses related works. Section 5 discusses experimental results and findings. Finally, Section 6 concludes the paper by summarizing our main contribution.

2 Preliminaries

2.1 Attribute selection

Given an initial set of n conditional attributes $f = \{f_1, f_2, ..., f_n\}$ and a target attribute f_{target}. Attribute selection problem is to find a subset $f' \in f$ with n' attributes ($n' \leqslant n$) that has better prediction accuracy for a given classifier[1,3,6].

In general, conventional attribute selection methods proceed as follows[7].

1. Select some conditional attributes based on a certain criterion, such as "mutual information" for categorical target and "mean squared errors" for numerical target, and make a candidate subset.
2. The candidate subset is evaluated based on prediction accuracy.
3. Add or remove attributes from the candidate subset and evaluate the modified subset. During the process, keep the best subset found so far.
4. If a certain stopping condition is satisfied, then output the best subset, otherwise repeat the process of "3".

In the process of selecting conditional attributes, conventional methods evaluate each single attribute to see how each attribute correlated to the target attribute.

As we mentioned in Section 1, there are many correlated attributes among conditional attributes like Fig.1. The single attribute analysis can not handle such correlations well. As a result, conventional methods may eliminate attributes of important correlated rules (two-dimensional rules) like the example and it prevents us from noticing important correlated knowledge about the target. Moreover, if we can use two-dimensional rules in decision trees or regression trees, we can construct accurate and compact prediction model[15,16]. Conventional attribute selections make it difficult to utilize such accurate classifiers that uses two-dimensional rules.

2.2 Two dimensional rules

In order to handle the correlation problem, we use two-dimensional rules[5], which are expressed as grid regions. For each pair of conditional attributes, we define an $N \times N$ cell grid plane G. A *grid region* is a union of cells in G that are connected. By using a grid region, we can divide a database D into D_{in} and D_{out}, where D_{in} is a set of records inside the region and D_{out} is a set of records outside.

A two-dimensional rule is a grid region R that optimizes a certain criterion. For classification problem, where target attribute is categorical, we use *information gain*, $Gain(R)$, defined as follows:

$$Gain(R) = -\frac{|D_{in}|}{|D|}\sum_{c=1}^{k} p_c(D_{in})\log p_c(D_{in}) - \frac{|D_{out}|}{|D|}\sum_{c=1}^{k} p_c(D_{out})\log p_c(D_{out})$$

In the formula, $|D|$ is the number of records in D. k is the number of values (classes) in the target attribute. $p_c(D)$ is the probability of the c-th target value in D. We find R that minimizes $Gain(R)$ as the optimal two-dimensional rule on G.

For regression problem, where target attribute is numerical, we use *mean squared error*, $MSE(R)$, defined as follows:

$$MSE(R) = \frac{1}{|D|}\sum_{r\in D_{in}} (r[f_{target}] - \mu(D_{in}))^2 + \frac{1}{|D|}\sum_{r\in D_{out}} (r[f_{target}] - \mu(D_{out}))^2$$

where r is a record in a database. $r[f]$ is value of attribute f of a record r. $\mu(D)$ is the mean value of the target attribute in D. We find R that minimizes $MSE(R)$ in regression problem.

Though the problem of finding the optimal grid region is NP-hard, if the shape of regions are restricted to be *x-monotone*, we can compute the optimal region efficiently in $O(N^2 \log N)$ expected running time [15,16], which is almost linear to the number of cells in G. An x-monotone region is a grid region whose intersection with any vertical line is undivided. For example, Fig.3(a) shows a region that is not x-monotone, since the intersection of the vertical line `A` and the region is divided. In contrast, Fig.3(b) is an x-monotone region. We can express an x-monotone region by a connected vertical stripes.

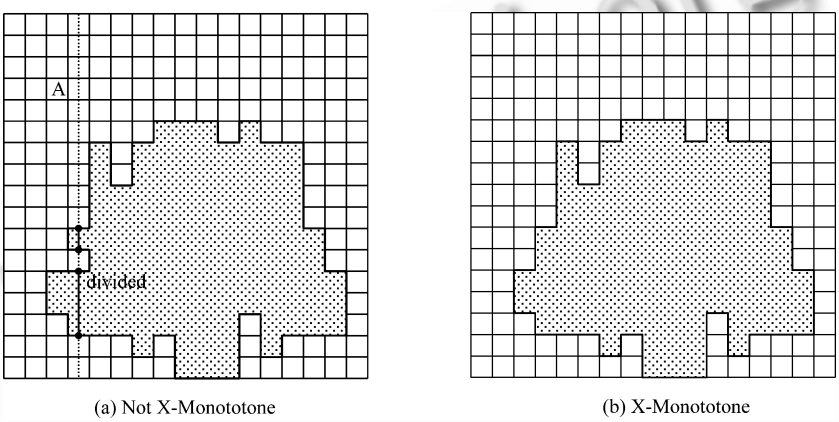

Figure 3. X-Monotone region

2.3 Shape evaluation

We compute the optimal x-monotone region for each cell grid and evaluate the regions by information gain or mean squared error. For some important regions whose evaluation is good, we further examine the shape to evaluate the importance of the regions' attributes. Any x-monotone region R on a grid can be represented by two vectors, say $\tau(R) = \{t_0, t_1, ..., t_{m-1}\}$ and $\beta(R) = \{b_0, b_1, ..., b_{m-1}\}$, and an *xoffset* value.

Figure 4 shows examples of x-monotone regions and their vector representations. As mentioned above, an x-monotone region is a connected m ($m > 0$) vertical stripes. The *xoffset* value is the X-index of the leftmost stripe, of an x-monotone to start. Two vectors, $\tau(R)$ and $\beta(R)$, are sequences of the Y-index of top and bottom of stripes, respectively, from the leftmost to the rightmost (the 0-th stripe to the $(m-1)$-th stripe).

We evaluate a shape of an x-monotone region R by *flatness* function, $Flat(R)$, as follows:

$$Flat(R) = max\left\{\frac{\sum_{i=0}^{m-1}(t_i - \mu(\tau(R)))^2}{m}, \frac{\sum_{i=0}^{m-1}(b_i - \mu(\beta(R)))^2}{m}\right\}$$

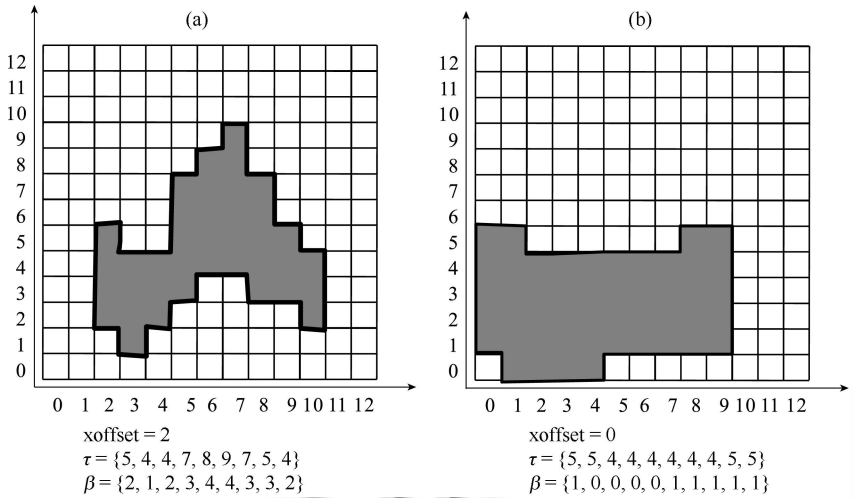

Figure 4. Representation of X-mono. region

where $\mu(\tau(R))$ and $\mu(\beta(R))$ are the mean value of $\tau(R)$ and $\beta(R)$, respectively. Since

$$\frac{\sum_{i=0}^{m-1}(t_i - \mu(\tau(R)))^2}{m} = \frac{m(\sum_{i=0}^{m-1} t_i^2) - (\sum_{i=0}^{m-1} t_i)^2}{m^2},$$

$Flat(R)$ can be computed by single scan of m stripes. The value of the flatness function shows how the sequence of each vector is apart from the mean value.

The larger the value of the flatness function is, the more important the corresponding two attributes are. We specify a threshold value, say min_{flat}, and if the value of the flatness function is larger than min_{flat}, we prioritize the corresponding attributes. An appropriate threshold will filter non-correlated attributes from the correlated ones.

Example: The flatness value of the region of Fig.4(a), whose top and bottom vectors are $\tau = \{5, 4, 4, 7, 8, 9, 7, 5, 4\}$ and $\beta = \{2, 1, 2, 3, 4, 4, 3, 3, 2\}$, is $max\{0.742, 0.889\}$ = 0.889. On the other hand, the flatness value of the region 4(b), whose top and bottom vectors are $\tau = \{5, 5, 4, 4, 4, 4, 4, 5, 5\}$ and $\beta = \{1, 0, 0, 0, 0, 1, 1, 1, 1, 1\}$, is $max\{0.240, 0.240\} = 0.240$. Note that the flatness value of the left region is much higher than that of the right one.

3 Attribute Selection by using 2D Rules

3.1 Single attribute evaluation

Assume we have n conditional numeric attributes, f_1, f_2,..., f_n, in a database. First of all, we compute evaluation score for each single attribute. Let $e(f)$ be the evaluation score of attribute f.

For classification problem where the target attribute is categorical, we used the following formula $e(f)$, which is based on information gain which is widely used in conventional attribute selection[8].

$$e(f_i) \quad = -\sum_{c=1}^{k} p_c \log p_c - \sum_{v=v_{min}}^{v_{max}} p_v \sum_{c=1}^{k} p_{c|v} \log p_{c|v}$$

In the formula, p_c is the probability of the c-th target value (class), v is a value of f_i ($v_{min} \leqslant v \leqslant v_{max}$), p_v is the probability of the value, and $p_{c|v}$ is the conditional probability of p_c given the value v.

For regression problem, we adopted the following evaluation attribute, which is used in the Regressional ReliefF (RReliefF)[17]. In the evaluation, *difference* of an attribute f_i between two records, say k and l, is defined by the following function[13]:

$$d(f_i,k,l) = \begin{cases} 0 & |x_{(l,i)} - x_{(k,i)}| \leqslant t_{eq}; \\ \dfrac{|x_{(l,i)} - x_{(k,i)}| - t_{eq}}{t_{diff} - t_{eq}} & t_{eq} < |x_{(l,i)} - x_{(k,i)}| \leqslant t_{diff}; \\ 1 & t_{diff} < |x_{(l,i)} - x_{(k,i)}| \end{cases}$$

In the function, $x_{(k,i)}$ denotes the (k,i)-th element of the database and t_{eq} and t_{diff} are the threshold values. In our experiments, we use the default threshold values of RReliefF.

We randomly choose a record, say k, and compute the difference of the target attribute from the k-th record and choose the K nearest records. Let S_K be a set of the K nearest records. Using S_K, we compute the average difference of f_{target}, $\bar{d}(f_{target})$, and the average difference of f_i, $\bar{d}(f_i)$. We also compute the average conditional difference which is defined as follows:

$$\bar{d}(f_{target}|f_i) = \sum_{l \in S_K} d(f_{target},k,l) * d(f_i,k,l)/K$$

We iteratively compute these values and evaluate the attribute f_i as follows:

$$e(f_i) = \frac{\bar{d}(f_{target}|f_i) * \bar{d}(f_i)}{\bar{d}(f_{target})} - \frac{(1 - \bar{d}(f_{target}|f_i)) * \bar{d}(f_i)}{1 - \bar{d}(f_{target})}$$

Without losing generality, we normalize the evaluation score so that we can make the value range into $[0-1]$.

3.2 Region evaluation

Next, we compute two dimensional rules to handle correlations among conditional attributes. For each pair of n conditional attributes, we make an $N \times N$ cell grid G. We decide the size of N so that we can make the density (average number of records per one cell) around 5, which we empirically found that the density value of 5 produces relatively accurate prediction models[8,9]. For example, if we have 2000 records, we set $N = \sqrt{2000/5} = 20$.

For each grid G, we compute the optimal x-monotone region and y-monotone region. Since the optimal x-monotone region and the optimal y-monotone region are not the same, we have to examine both. We have developed an efficient algorithm for computing optimal x-monotone region [15,16,5]. Given a grid, we pre-compute candidates of the top and the bottom index for each vertical stripe from left to right. During the dynamic programming of the pre-computation, we maintain the best intermediate value and finally identify the position of the right-most stripe of the optimal region. Then, from the right-most stripe, we construct the optimal x-monotone region from right to left using the candidate indices. The algorithm computes the optimal

x-monotone region in $O(N^2 logN)$. The algorithm is efficient enough to compute all optimal x and y monotone regions for each pair of attributes. Because of the space limitation, we can not describe the detailed algorithms here but you can find the details of the algorithm in Ref.[5].

We compute evaluation score for each optimal region (two-dimensional rule) and let $e(R)$ be the evaluation score of R. We use $Gain(R)$ and $MSE(R)$, defined in Section 2, as $e(R)$ for classification and regression problem, respectively. As same as single attribute evaluation, we also normalize the evaluation score.

We select regions whose evaluation is better than a user specified threshold value, say min_{2D}. We call the selected regions *"significant regions"*. Though significant regions and their corresponding attributes are worth to select, some of significant regions like Fig.2(a) can be judged by single attribute analysis. Therefore, for each significant region, we examine the shape to find whether both attributes of the region are necessary or not. We use the flatness function, $Flat(R)$, which is defined in Section 2, to evaluate the shape of a region R. If the $Flat(R)$ is larger than a threshold value min_{flat}, we call such a region R a *"shapely region"*. We prioritize both x-axis and y-axis attribute of each shapely region.

3.3 Re-Ranking attributes

Let $e(f)$ is the evaluation of an attribute f, $e(R)$ is the evaluation of a region R. Let $G_{(i,j)}$ is a grid whose x attribute and y attribute are f_i and f_j, respectively. Similarly, let $R_{(i,j)}$ and $R_{j,i}$ are the optimal x-monotone region and the optimal y-monotone region, resp., on $G_{(i,j)}$.

A shapely significant region R satisfies $e(R) > min_{2D}$ and $Flat(R) > min_{flat}$ and is considered to be an important correlation. Therefore, we prioritize both x-axis attribute and y-axis attribute of each shapely significant regions.

Let $\mathbf{R^s}$ be the set of all shapely significant regions. The evaluation of attribute f_i is updated to $max\{e(f_i), e_i(\mathbf{R^s})\}$, in which $e_i(\mathbf{R^s})$ is the average evaluation of shapely significant regions whose one of the axis is f_i. In order to find each attribute evaluation value for this re-ranking step, we define $e_i(\mathbf{R^s})$as follows:

$$e_i(\mathbf{R^s}) = \frac{\sum_{R \in R_{*,i} \in \mathbf{R^s}} e(R) + \sum_{R \in R_{i,*} \in \mathbf{R^s}} e(R)}{|R_{*,i} \in \mathbf{R^s}| + |R_{i,*} \in \mathbf{R^s}|}$$

where $R_{*,i} \in \mathbf{R^s}$ is a set of shapely regions whose y attribute is f_i and $|R_{*,i} \in \mathbf{R^s}|$ is the number of shapely regions whose y attribute is f_i. Similarly, $R_{i,*} \in \mathbf{R^s}$ is a set of shapely regions whose x attribute is f_i.

Example: Assume we have four attributes, f_1, f_2, f_3, f_4 and their evaluation score are $e(f_1) = 0.6, e(f_2) = 0.5, e(f_3) = 0.5, e(f_4) = 0.4$. Assume the evaluation scores of regions are as following table. In the table, $e(R_{i,j})$ is in the (i,j)-th element and we underlined the shapely significant regions.

	f_1	f_2	f_3	f_4
f_1		0.2	0.4	0.3
f_2	0.3		<u>0.8</u>	0.2
f_3	0.7	<u>0.6</u>		0.2
f_4	<u>0.5</u>	0.5	0.6	

Since $e_1(\mathbf{R^s}) = 0.5$ and the single attribute evaluation is better, $e(f_1)$ is not updated. Note that 0.3 and 0.7 of the first column are not shapely significant and all the first row are not shapely significant. On the other hand, since $e_2(\mathbf{R^s}) = 0.7$, $e(f_2)$ is updated to 0.7. Shapely significant regions of the second column and row is 0.6 and 0.8. Similarly, $e(f_3)$ is updated to 0.7 and $e(f_4)$ is updated to 0.5.

This re-ranking warrants to revive any important abandoned (discarded or lower ranked) attributes in a correlated numerical database. After the re-evaluation, we sort the attributes according to the updated evaluation. Then, we select the top $n' < n$ attributes. We interactively find adequate n' so that the prediction accuracy of the projected database is maximized. In this process, we used popular classifiers C4.5 for classification problem, and SMOreg for regression problem, that are provided in WEKA environment[18].

4 Related Works

Jakulin and Bratko initiated the study of the correlation problems in which they called *attribute interactions*[10,11]. Attribute interactions are the irreducible dependencies between attributes. They developed the interaction gain measure to identify whether data sets have interaction. This measure can detect 2-ways and 3-ways interactions. In definition, the 2-ways interactions are between one attributes and the target attribute. The 3-ways interactions are between two attributes and the target attribute. Jakulin and Bratko assumed that correlation in database as a special form of attribute interaction[11].

Their result is comparable to our proposed method, which utilizes the *2D rules* on the dataset. Our first advantage is that the *2D rules* ensure the mapping of two attributes correlation characteristic in form of x-monotone regions. The second advantage is that our shape evaluation warrant to detect significant regions as correlated attributes in higher-interdependencies among attributes. Our method also shows clearly the importance of both single and correlated attributes in the selection result.

Using Jakulin and Bratko's work, Zhao and Liu[19] considered the searching for interacting attributes problem with the key issues hindering the use of consistency measure. *Consistency measure* is defined by inconsistency rate which is calculated as the ratio of the inconsistency count on its number of data items element. *Inconsistency count* is the number of the same valued data items in an attribute that have different target attributes. In order to handle attribute interactions, they add consistency measure with consistency contribution indicator. This indicator shows how significantly the elimination of attributes will affect attribute consistency. They use a step called *backward elimination* to evaluate and find each attribute interaction. For the purpose of individual ranking, *symmetrical uncertainty* was applied. *Symmetrical uncertainty* is a fast correlation measure to evaluate the relevance of individual attributes[18].

In comparison to the work of Zhao and Liu[19], we explore the characteristics of 2D rules over all attributes. 2D rules provide us visualization of two attributes relation within x-monotone shapes. Our flatness and shape evaluation effectively find all correlated attributes of the significant region on certain shape threshold. These predefined thresholds are set to prune less important attributes, which will

lead to higher accuracy. Using *re-ranking mechanism*, our approach warrants to give priority for both single and correlated attributes based on their importance to the determination of the target class. We found this capability is missed in other attribute selection methods. Furthermore, the proposed method directly support for regression problem as well, while Zhao and Liu's method did it with additional steps.

5 Experimental Results

We evaluate the effectiveness of the proposed method through intensive experiments by using various real datasets.

5.1 Dataset

We used sixteen datasets, eight of them are for classification problems and the rest are for regression problems. All of those datasets are provided by the UCI ML Repository[1]. Brief statistics about these datasets are summarized in Table 1. In the table, "#attr" shows the number of attributes and "size" shows the number of records in the dataset, and "type" indicates the type of the problem, i.e., classification (c) or regression (r).

Table 1 Summary of datasets

Dataset	#attr	size	type
Credit german	7	1055	c
Heart StatLog	13	351	c
Ionosphere	35	351	c
Pima diabetes	8	768	c
Labor	17	57	c
Hepatitis	20	155	c
Breast-w	10	699	c
Collic	28	368	c
Body fat	15	252	r
Pharynx	12	195	r
Pollution	15	60	r
Sensory	12	579	r
LowBirthWeight	10	189	r
AutoMPG	8	398	r
Wisc.Cancer	33	194	r
ElNino	9	782	r

5.2 Under-Estimated attributes

We examined how the evaluation of attributes are updated if we take into account the correlations. In this experiment, we compared $e(f_i)$ and $e_i(\mathbf{R^s})$ for each attribute of *Pima diabetes* dataset. Table 2 summarizes the difference of the evaluation. We use the threshold value min_{flat}= 0.5 for distinguishing significant regions, which we empirically find to be adequate in separating correlated attributes. Notice that we compare the threshold value with the mean squared error of both top and bottom stripes in an x-monotone shape. However this value is dependent to the data characteristic, hence it is open to explore for different data set.

Table 2 Evaluation of attribute

i	Name	$e(f_i)$	$e_i(\mathbf{R^s})$
1	Plas	0.1901	0.0670
2	Mass	0.0749	0.0746
3	Age	0.0725	0.0975
4	Insu	0.0595	0.0662
5	Skin	0.0443	**0.0752**
6	Preg	0.0392	**0.1446**
7	Pedi	0.0208	**0.4026**
8	Pres	0.0140	**0.0722**

We highlight prominent values that significantly higher than that of conventional single attribute evaluation. This shows some of the attributes are under-estimated if we take into account the correlation among attributes. It also shows the proposed method can select such under-estimated attributes adequately.

5.3 Shapely significant region

Figure 5 is an example of a shapely significant region which we found in Pima diabetes dataset. The x-axis and y-axis of the region are "insu" and "skin", respectively. Note that we have to see both attributes' value in order to judge whether a record is inside the region or not.

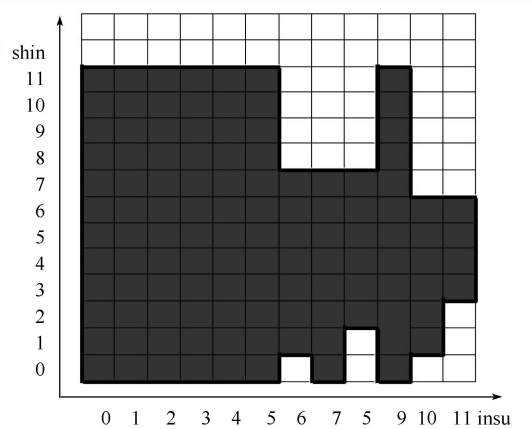

Figure 5. Shapely region from Pima diabetes

5.4. Accuracy

We examine the proposed method by comparing accuracy of selected attributes. For classification cases, we compared the accuracy of each dataset (in %) using C4.5 decision trees, which is one of the most popular classifiers. We performed 10-fold cross-validation method. Similarly, for regression cases, we compared *the root of mean squared error* of SMOreg classifier. We set min_{2D} to 0.5 in these experiments. We empirically found that this value produces adequate shapely regions on both classification and regression problems.

We, then, compute a ranking of attributes and reduce the number of attributes so that the accuracy of the projected datasets is maximized. We also compute the

accuracy of two representative attribute selection methods, Correlation based Feature Selection (CFS)[7] and ReliefF[12]. Both are attribute selection methods that can handle classification and regression cases as our proposed method built. Both are available in the WEKA environment[18].

CFS[7] assumes that useful attribute subsets contain attributes that are predictive of the target variable but uncorrelated with one another. CFS computes the *merit* of a attribute subset from pairwise attribute correlations. A heuristic search is used to traverse the space of attribute subsets in reasonable time; the subset with the highest merit found during the search is reported. CFS thus also determines the number of returned attributes.

The ReliefF[12] algorithm has been designed for *multiclass* problems and is based on the k nearest neighbors from the same attribute class. Relief algorithms represent quite original approach to attribute selection, that is not based on evaluation of one-dimensional probability distributions.

Table 3 and 4 show the average of the 10-fold cross-validation results for classification and regression problem respectively. As for the classification accuracy, the proposed method achieved the best among the three methods if we compared the average accuracy. As for the regression accuracy, CFS's accuracy is the best, while the proposed method is the second, the differences are small and we can conclude the accuracy of the proposed methods is comparable.

Table 3 Classification accuracy

Dataset	Proposed	CFS	ReliefF
Credit german	69.20	71.00	70.30
Heart StatLog	83.33	80.74	76.67
Ionosphere	91.45	89.74	91.17
Pima diabetes	77.80	76.69	77.34
Labor	77.19	78.95	73.68
Hepatitis	91.94	80.65	83.87
Breas-w	94.99	94.56	94.56
Collic	68.75	66.30	66.30
Average	81.83	79.83	79.23

Table 4 Regression accuracy

Dataset	Proposed	CFS	ReliefF
Body fat	1.3083	4.8681	4.4827
Pharynx	352.3265	311.9828	352.3265
Pollution	37.3346	39.3228	40.316
Sensory	0.7838	0.8281	0.8280
LowBirthWeight	387.306	372.64	387.306
AutoMPG	2.2585	32.2585	2.2602
Wisc.Cancer	26.6994	26.9043	27.3683
ElNino	0.2236	0.2291	0.2232
Average	101.0300	98.6292	101.8888

Table 5 is the number of attributes in the reduced datasets. CFS reduces the most, while the reduction size of ReliefF and our method are almost same. For example, in the *Pima diabetes*, the proposed method selects seven attributes and

eliminates the "insu" attribute. This dataset results 77.80% accuracy. On the other hand, CFS selects four attributes and the accuracy is 76.69%.

Table 5 Number of attributes

Dataset	Proposed	CFS	ReliefF
Credit german (21)	5	3	20
Heart StatLog (13)	6	7	13
Ionosphere (34)	32	14	33
Pima diabetes (8)	7	5	8
Labor(17)	4	7	17
Hepatitis (20)	10	10	19
Breast-w (10)	7	9	10
Collic (28)	15	2	27
Body fat (15)	14	2	14
Pharynx (11)	10	4	11
Pollution (16)	14	5	15
Sensory (12)	11	7	11
LowBirthWeight(10)	9	4	9
AutoMPG (8)	7	7	7
Wisc.Cancer(33)	20	16	33
ElNino (9)	8	3	8

We remark that the accuracy results in this experiments are the results of conventional classifiers, i.e., C4.5 and SMOreg. Since conventional attribute selection methods try to find the subset of attributes that maximizes accuracy of such conventional classifiers. If we use two-dimensional rules in decision trees and regression trees, we can construct more accurate trees than conventional classifiers [15,16]. Moreover, as we mentioned, conventional methods tend to eliminate important correlations among attributes, while we can preserve such correlations.

5.5 Performance

In this section, we examine the time to compute the whole attribute selection process. The proposed method is implemented with Java Standard Edition Release 1.6.0x and MySQL5.1. All experiments were performed on a machine of Pentium-4, 3.06GHz CPU and 1.5GB RAM on the Microsoft Windows XP professional.

Table 6 shows the time (in seconds) to compute subset of attributes and find reduced dataset. We compared the time of our method to results of CFS and ReliefF. We find that the proposed method demands for higher computational cost compared to CFS and ReliefF since we have to compute all pairs of conditional attributes. In this literatures, even though ReliefF is known to be suffered by high dimensional data if n becomes large ($n > 20$) [13], the time is faster than our method. Though it takes time compared to conventional methods, the proposed method has demonstrated the ability to preserve important correlations. Notice that reputation of ReliefF is very good even if it takes time compared to other attribute selection methods. We think that the quality of selected attributes is more important than faster computation in the attribute selection problem.

Table 6 Performance

Data set	Proposed	CFS	ReliefF
Credit german	13.01	0.06	0.75
HeartstatLog	17.70	0.22	1.08
Ionosphere	167.43	2.01	11.80
Pima diabetes	27.60	0.04	1.93
Labor	14.70	7.17	11.7
Hepatitis	10.10	1.00	1.19
Breast-w	79.0	9.00	22.10
Collic	15.10	2.27	12.7
Body fat	23.61	0.31	1.55
Pharynx	11.60	0.72	2.33
Pollution	19.58	0.11	2.11
Sensory	38.67	0.33	9.90
LowBirthWeight	2.68	1.30	2.20
AutoMPG	6.62	4.70	6.10
Wisc.Cancer	10.45	1.00	1.3
ElNino	57.68	20.01	56.90
Average	32.22	3.14	9.10

Moreover, by preserving correlations we can increase prediction accuracy and can find important knowledge about the target attribute like an example of Fig.1 that is tend to be eliminated by conventional methods. Thus, the proposed method has demonstrated the ability to increase the accuracy of learning performance by reducing irrelevant attributes within data set and preserve important correlated attributes that are important to the determination of target attribute.

6 Conclusions

In this paper, we proposed new attribute selection method that preserve important correlations among conditional attributes, which are ignored in conventional methods. We demonstrated the effectiveness of our approach using real datasets. We also showed that the proposed method works well in both classification and regression problem.

This work inspires further works on in the attribute selection problem. To handle correlations among more than three attributes are one of our challenging future works. We are also interested in how we can improve the time to solve the attribute selection problem.

Acknowledgements This work was supported by KAKENHI (#19500123). Taufik Djatna was supported by scholarship of MEXT Japan.

References

[1] Asuncion A, Newman D. UCI machine learning repository. Available: http://www.ics.uci.edu/learn/MLRepository.html

[2] Bakus J, Kamel MS. Higher order attribute selection for text classification. Knowledge and Information Systems, 2006, 9: 468–491.

[3] Bhavani SD, Rani TS, Bapi RS. Attribute selection using correlation fractal dimension: Issues

and applications in binary classification problems. Applied Soft Computing, 2008, 8: 555–563.
[4] de Sousa EPM, Traina C, Traina AJM, Wu LJ, Faloutsos C. A fast and effective method to find correlations among attributes in databases. Data Mining and Knowledge Discovery, 2007, 14: 367–407.
[5] Fukuda T, Morimoto Y, Morishita S, Tokuyama T. Data mining with optimized two-dimensional association rules. ACM Trans. on Database Systems, 2001, 26(2): 179–213.
[6] Guyon I, Elisseeff A. An introduction to variable and attribute selection. Journal of Machine Learning Research, 2003, 3: 1157–1182.
[7] Hall MA. Correlation based attribute selection for discrete and numeric class machine learning. In: Proc. of the 17th International Conf. on Mach. Learn. 2000. 359–366.
[8] Hall MA, Holmes G. Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans. on Knowledge and Data Engineering, 2003, 15: 1437–1447.
[9] Harol A, Lai C, Pezkalska E, Duin RPW. Pairwise attribute evaluation for constructing reduced representations. Pattern Analysis and Applications, 2007, 19: 55–68.
[10] Jakulin A, Bratko I. Analyzing attribute dependencies. In Proc. of the 7th Eur. Conf. on Principle and Practice of Knowledge Discovery in Databases (PKDD). 2003: 229–240.
[11] Jakulin A, Bratko I. Testing the significance of attribute interaction. In: Proc. of the Intl. Conf. on Machine Learning (ICML). 2004. 409–416.
[12] Kononenko I, Hong SJ. Attribute selection for modeling. Future Generation Computer Systems, 1997, 13: 181–195.
[13] Kononenko I, Robnik-Sikonja M. Non-myopic feature quality evaluation with (R)ReliefF. Chapman & Hall CRC, Data Mining and Knowledge Discovery Series, Taylor and Francis Group, Boca Raton USA, 2008. 169–191.
[14] Liu WZ, White AP. The importance of attribute selection measures in decision tree induction. Machine Learning, 1994, 15: 25-41.
[15] Morimoto Y, Fukuda T, Morishita S, Tokuyama T. Implementation and evaluation of decision trees with range and region splitting. Constraints, an International Journal, 1997: 401–427.
[16] Morimoto Y, Ishii H, Morishita S. Efficient construction of regression trees with range and region splitting. Machine Learning, 2001, 45: 235–259.
[17] Robnik-Sikonja M, Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning, 2003, 53: 23-69.
[18] Witten IH, Frank E. Data Mining - Practical Machine Learning Tools and Techniques with Java Implementation (2nd Edition). Morgan Kaufmann, San Fransisco-USA, 2005.
[19] Zhao Z, Liu H. Searching for interacting features. In: Proc. of Intl. Joint Conference On AI (IJCAI). 2007: 1156–1161.