

FSAVD : Feature Selection Method Applying Variation of Data

Hyun jin Kim and Sanghyun Park

Department of Computer Science, Yonsei University
134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, South Korea
{chriskim, sanghyun}@cs.yonsei.ac.kr

Feature selection is method for choosing meaningful and good features among a large number of features. Typical feature selection methods assume that if a feature is useful, samples in same class have similar values and samples in other classes have different values. Or if a quantity of information between feature and class is large, the feature is thought to be useful. However, samples in same classes do not always indicate that the samples have values in just one similar range. It is possible that samples in same classes have different values in various ranges. If a data set has this various characteristic, reflecting it to a method might be helpful. FSAVD(Feature Selection Method Applying Variation of Data) is a novel approach that uses wavelet transform to find different clustered values of the data which has various characteristic, and based on this measure, FSAVD can choose useful features.

Key Words: feature selection, data mining, wavelet transform, data variation, bioinformatics

1. INTRODUCTION

As the medical technology advances, the average age has risen all over the world. The management of disease gets accordingly important. Among those diseases there are cancers. Since biological data like ones related to cancer have quite a lot of attributes and many worthless attributes, feature selection method is often used to reduce the number of attributes and to remove worthless ones. The typical feature selection methods are relief-A and symmetrical uncertainty. Relief-A is to first select attributes that differ much in attributes' values between classes, and symmetrical uncertainty chooses attributes that have most information on between attributes and classes. However, these methods do not consider the data which have varieties. The data which have varieties mean that though the samples belong to the same class, they do not have the same value (Figure 1). Normally, it is impossible to draw fine attributes from them with the ordinary feature selection method. Hence, we propose FSAVD which can reflect the varieties of data. FSAVD is to select good features through 4 steps. The first step is to arrange the gene expression levels in each class according to size. It is a pre-step to collect samples banded with wavelet transform. In the second step, Wavelet transform is applied to 2-level. The status of data can be checked as applying Wavelet transform. After Wavelet transform, it has a function that removes the sections which have more values than the threshold. And then, third step is to calculate the averages of sets in each class and to find the degree of dispersion. The bigger the degree of dispersion, the more various aspects the attribute can reflect. The last fourth step is to extract k number of attributes according to the degree of dispersion. FSAVD is a new approach method that has not been used, and it can reflect variation of data.

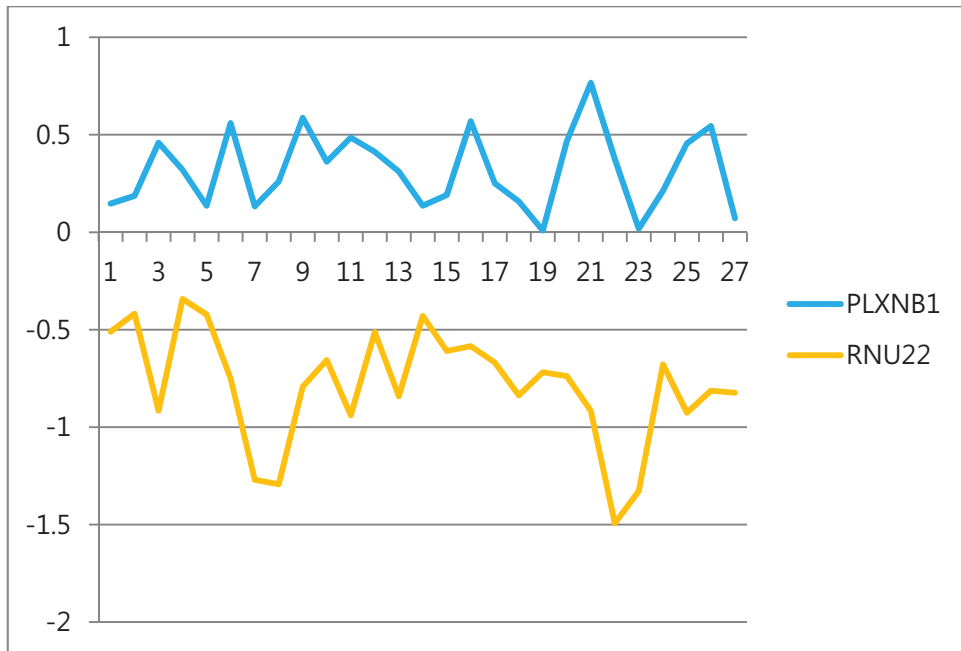


Figure 1. Gene expression levels in PLXNB1 : RNU22 gene fusion from GSE 15484 data.

In this figure, x-axis represents each sample and y-axis represents gene expression levels. The figure shows gene expression levels from gene fusion in one class (aggressive cancer).

2. RELATED WORKS

2.1 Relief-A

Let's suppose that there are data sets which consist of two classes and a number of attributes. If a certain attribute in the data sets is useful, the samples in that same class have the similar values and the samples in other classes have different values. Relief is a feature selection method based on attribute estimation in this way [2]. The pseudo code is shown below to explain the logic of this method:

```

FOREACH Randomly selected sample S DO
    Find nearest hit H and nearest miss M
    FOREACH Attributes DO
         $W[A] = W[A] - \text{diff}(A, S, H)/n + \text{diff}(A, S, M)/n$ 
    END
END

```

In this method, first we randomly extract one sample S from data sets, and then search for the nearest hit H from the same class and the nearest miss M from the other class by use of Euclidean distance method. The difference between S and H is subtracted from the weight value of each attribute and the difference between S and M is added to the weight value of each attribute. This process is for selecting the attributes which have less difference from the sample in the same class and also have bigger difference from the samples in the other classes. Therefore, the higher the final weighted value, the more useful the attribute is.

If, however, the samples tend to have the same values or have many noisy attributes, selecting only one nearest neighbor may produce a wrong result. Relief-A is the method that applies K-Nearest Neighbor method to the searching process to improve this issue [1]. While Relief searches for one nearest hit and one miss, Relief-A searches for k number of neighbors, which a user sets, in each class, and then the weighted value is calculated by use of averaging the attribute values of neighbors. It helps to reduce noise in data.

2.2 Symmetrical uncertainty

The useful attribute in classification algorithm is the attribute that has necessary information on classifying classes. If an attribute has necessary information on classifying classes, that attribute is strongly correlated with the classes: the interdependency between the attribute and the classes is high. The useful attribute should also be distinguished from other attributes. If there exist a number of attributes which appear to have similar aspects, the usefulness of the attribute gets lowered.

Here we can use information gain to measure the correlations between an attribute and a class, and between attributes. Information gain can represent the amount of the correlated information between two variables based on the entropy theory of a measure of the uncertainty associated with a random variable [4]. The entropy of variable X can be written as

$$H(X) = - \sum_i P(x_i) \log_2(P(x_i))$$

Once we notice that there exists another variable Y, we can write the entropy of variable X as

$$H(X|Y) = \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2(P(x_i|y_j))$$

Information gain between variable X and variable Y is the value of the entropy of variable X that is offset by the entropy of variable X conditional on Y. That is, $IG(X|Y) = H(X) - H(X|Y)$. At this time, the information gain is symmetrical because $IG(X|Y) = IG(Y|X)$ holds true. Therefore, mutual information is another word for information gain. This kind of symmetry is essential for calculating the inter-correlation between attributes.

However, the information gain has a weak point that the information gain can grow big because more instances means bigger entropy the variable has: when a variable X has more instances x_i , then the variable X has bigger entropy. Hence, there is a problem that a variable can be given higher rank only because it has more instances. Also, the fact that the information gain varies according to the number of instance, means that any normalization hasn't been done in order to help to compare information gains of each different variable.

Symmetrical Uncertainty is a measure that can overcome the weakness of information gain [3]. The problem caused by being not normalized can be solved if we divide the information gain by the sum of the entropy of each variable. Symmetrical Uncertainty can be expressed as

$$SU(X, Y) = 2 \left\{ \frac{IG(X|Y)}{H(X) + H(Y)} \right\}$$


This method can deal with the unbalanced factors because it divides the entropy of that variable even if the number of variables varies, and since the value of Symmetrical Uncertainty becomes normalized into a range of 0 and 1, it is possible to compare between information gains.

3. METHODS

3.1 Sorting in ascending order

Sorting values in each feature is needed to search clustered parts of sample values. In this paper, we sorted the values in ascending order (Figure 2). After the sorting, the clustered parts might be revealed which have great differences among the sample values. Sorting is preprocessing phase to know the clustered parts using wavelet transform.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
F	26	1	3	53	4	17	16	17	25	25	17	1	51	53	4	54



	S2	S12	S3	S5	S15	S7	S6	S8	S11	S9	S10	S1	S13	S4	S14	S16
F	1	1	3	4	4	16	17	17	17	25	25	26	51	53	53	54

Figure 2. Example of ascending sort

F indicates feature and S indicates sample.

3.2 Applying wavelet transform

In the first phase, we sorted the sample values of feature in ascending order. In this phase, we can find the clustered parts of sample values. Applying 2-level wavelet transform to sorted sequence. Then variable $a1$, $d1$, $a2$, and $d2$ are produced (Figure 3). In this paper, we used $d2$ to find the clustered parts. $d2$ represents how large differences among the separated parts. Therefore, if we calculate absolute values of $d2$ and discarding values which are bigger than threshold, we can get only the clustered parts we wanted to know (Figure 4). Threshold can be calculated as $(\text{maximum } |d2| \text{ value} - \text{minimum } |d2| \text{ value}) / 2$.

	S2	S12	S3	S5	S15	S7	S6	S8	S11	S9	S10	S1	S13	S4	S14	S16
G1	1	1	3	4	4	16	17	17	17	25	25	26	51	53	53	54
a1	$\sqrt{2}$		$\frac{7\sqrt{2}}{2}$		$10\sqrt{2}$		$17\sqrt{2}$		$21\sqrt{2}$		$\frac{51\sqrt{2}}{2}$		$52\sqrt{2}$		$\frac{107\sqrt{2}}{2}$	
d1	0		$-\frac{\sqrt{2}}{2}$		$-6\sqrt{2}$		0		$-4\sqrt{2}$		$-\frac{\sqrt{2}}{2}$		$-\sqrt{2}$		$-\frac{\sqrt{2}}{2}$	
a2		$\frac{9}{2}$				27				$\frac{93}{2}$				$\frac{211}{2}$		
d2		$-\frac{5}{2}$				-7				$-\frac{9}{2}$				$-\frac{3}{2}$		

Figure 3. Example of applying 2-level wavelet transform

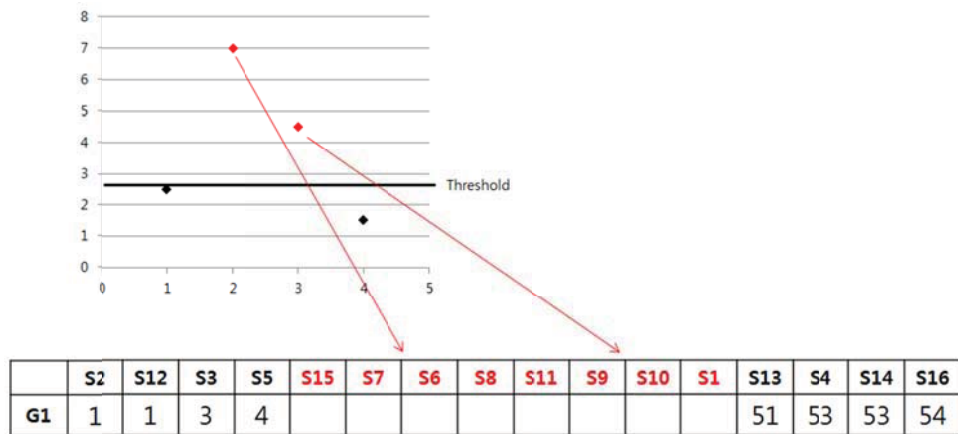


Figure 4. Example of discarding values which are bigger than threshold

3.3 Calculating degree of dispersion

After discarding outlier parts, the others are believed to reflect the variation of data. In the third phase, we have to determine which feature has large differences among the clustered parts. We developed the measure to calculate this problem which is called degree of dispersion.

$$\text{Degree of dispersion} = \sum_{a=1}^{CN} \sum_{b=1}^{L(a)} \sum_{c=a+1}^{CN} \sum_{d=1}^{L(c)} \text{Distance}[\text{Mean}(a, b), \text{Mean}(c, d)]$$

CN = the number of classes

$L(x)$ = the number of clustered parts in x th class

$\text{Distance}[x, y]$ = Euclidean distance between x and y

$\text{Mean}(x, y)$ = average of x th class and y th clustered part

Degree of dispersion is sum of differences among the clustered parts of different classes. If the differences among clustered parts of different classes are larger, the variation of data can be easy to be discovered. Therefore, the larger the difference among the clustered parts, the better the feature is.

3.4 Choosing k -top features

If the degrees of dispersion are clearly calculated, choosing k -top features which have high degrees of dispersion is needed. k is user-dependent variable and the k -top features that have high degrees of dispersion are meaningful and useful features which reflect variation characteristic of the data.

4. CONCLUSION

The world is rapidly changing now and data which are influenced by real world is becoming more complex. In this manner, variations of data are also revealed in many situations. Almost of these complex data need feature selection but most methods do not reflect the variation characteristic. We proposed novel approach FSAVD which can reflect the variation characteristic. FSAVD can find the clustered parts of data and discard the other useless pieces.

REFERENCES

- [1] Igor Kononenko. 1994. Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning. pp. 171-182. 1994.
- [2] Kenji Kira, Larry A. Rendell. 1992. A Practical Approach to Feature Selection. In: Ninth International Workshop on Machine Learning. pp. 249-256. 1992
- [3] Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. 1988. Numerical recipes in C. Cambridge University Press.
- [4] Quinlan, J. 1993. C4.5: Programs for machine learning. Morgan Kaufmann.