Classification of Water Pollution with Feature Selection

Pwint Mar Naing Win, Thandar Aung
Computer University (Mandalay)
pwintmar@gmail.com

Abstract

In the feature subset selection problem, a learning algorithm is faced with the problem of selecting a relevant subset of features upon which to focus its attention to achieve the highest predictive accuracy with the learning algorithm on this domain, a feature subset selection method should consider how the algorithm and the training data interact with filter method. This paper applies the normalization by decimal scaling process before feature selection to speed up the learning phase and prevent attributes with initially smaller ranges. This paper uses sequential forward selection to improve the generalization performance of pattern recognizers for water pollute or not. k-Nearest Neighbor classifier is built with filter approach by using the sequential forward selection. To estimate how accurately a classifier labels future data, this paper evaluates the performance of k-Nearest Neighbor classifier on the complete features and the selected feature subset by using the k-fold cross-validation.

1. Introduction

Classification is one of the fundamental problems in machine learning theory. The problem of classification has been well examined when the numbers of variables is large. Less work has been done as the number of variables increases. Feature selection can solve these problems. Water is the source of life and constitutes 65% of human body and it plays a vital role in most of biologic functions, so need to classify water pollute or not. This paper applies the sequential forward selection with filter approach on the water dataset. To speed up the learning phase, normalization by decimal scaling method is used before feature selection. An attribute is normalized by scaling its range, such as 0.0 to 1.0 [1]. Sequential forward selection is quadratic time, Near-Optimal and Greedy algorithm. Filter approach is independent of the classifier so it can execute fast. This paper finds the "goodness" of feature subsets using the information gain and evaluates the performance of k-Nearest Neighbor classifier on the complete features and selected feature subset.

2. Background Theory

2.1 Normalization by Decimal Scaling

Preprocessing can improve the efficiency and ease of the mining process. Normalization by decimal scaling method moves the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value of A is normalized to v' by computing as follows:

$$v' = v / 10^j \quad (1)$$

where j is the smallest integer such that $Max(v') < 1$ [1].

2.2 Filter Approach

In filter approach, a search strategy is needed to direct the Filter Feature Subset Selection process as it explores the space of all possible combination of features and the objective function evaluates candidate subsets and returns a measure of their "goodness", a feedback signal used by the search strategy to select new candidates. In the classifier design process, filters carry out feature selection prior to classifier training as shown in Figure 1 [2].

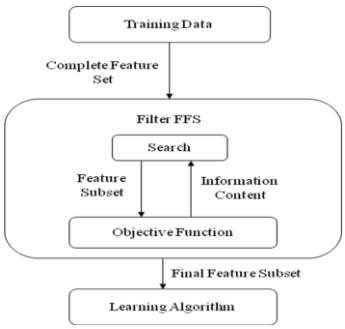

Figure 1. Filter Approach

2.3 Sequential Forward Selection

This paper uses sequential forward selection with information gain measure. The working of sequential forward selection begins with zero attributes and then evaluates all features subsets. After that selects the one with the best performance and adds to these subsets the feature that yields the best performance subsets of next larger size. This process can see in Figure 2.

```
SS=0
BestEval=0
REPEAT
        BestF=None
        For each feature F in Feature Selection
        AND NOT in SS {
                SS'= SS ∪ {F}
        IF EVAL(SS') > BestEval THEN
          BestF = F;
          BestEval = Eval(SS')}
         IF      BestF <> None      THEN
        SS = SS ∪ {BestF}
   UNTIL  BestF = None OR SS = FS
   RETURN SS
```

Figure 2. Sequential Forward Selection Algorithm

Information measures are used to calculate evaluation measures. Expected information is calculated by this equation:

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log(p_i) \qquad (2)$$

where p_i is the probability that an arbitrary sample belongs to C_i and is estimated by S_i/S. Let attribute A have v distinct values, $\{a_1, a_2, ---, a_v\}$. Attribute A can be used to partition S into v subsets, $\{S_1, S_2, ----, S_v\}$, where S_j contains those samples in S that have value a_j of A. If A were selected as the test attribute, then these subsets would correspond to the branches grown from the node containing the set S. Let s_{ij} be the number of samples of class C_i in a subset S_j. The entropy, or expected information based on the partitioning into subsets by A, is given by

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j}, \ldots, s_{mj}}{s} I(s_{1j}, \ldots, s_{mj}) \qquad (3)$$

The term $\frac{S_{1j}, \ldots, S_{mj}}{S}$ acts as the weight of the j^{th} subset and is the number of samples in the subset divided by the total number of samples in S. Note that for a given subset S_i,

$$I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log(p_{ij}) \qquad (4)$$

where $p_{ij} = s_{ij} \Big/ \left|S_j\right|$ and is the probability that a sample in S_j belongs to class C_i. The encoding information that would be gained by branching on A is

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A) \qquad (5)$$

Gain (A) is expected reduction in entropy caused by knowing the value of attribute A [1].

2.4 k-Nearest Neighbor Classifier

In k-Nearest Neighbor classifier, the nearest neighbors of an instance were defined in terms of the distance. More precisely, an arbitrary instance x be described by the $<a_1(x), a_2(x), \ldots, a_n(x)>$ where $a_r(x)$ denotes the value of the r^{th} attribute of the instance x. Then the distance between two instances x_i and x_j was defined to be d (x_i, x_j). As shown there, the value $\hat{f}(x_q)$ returned by this algorithm as its estimate of $f(x_q)$ was just the most common value of f among the k training examples nearest to x_q. If we choose k=1 then the 1 nearest neighbor algorithm assigns to $\hat{f}(x_q)$ the value $f(x_i)$ where x_i was the training instance nearest to (x_q). For large values of k, the algorithm assigns the most common value among the k-Nearest Neighbor examples.

To affect the accuracy of classification, k-Nearest Neighbor classifier is used simple distance-weighted function. That is defined as follows:

$$w_i = 1 \Big/ d_i \qquad (6)$$

where w_i is the weight for i^{th} nearest neighbor and d_i is the distance of i^{ih} nearest neighbor to the classified case. Voting strength is the strength of a case that should be classified into category j is used this equation

$$S_j = \sum_{i=1}^{k} \begin{cases} w_i & if \ c_i = j \\ 0 & otherwise \end{cases} \qquad (7)$$

where c_i is the category of the i^{th} neighbor

Strong facts of k-Nearest Neighbor classifier are it learns quickly $O(n)$ for a training set of n instances. It is guaranteed to learn a consistent training set. It was intuitive and easy to understand, which facilitates implementation and modification. It provides good generalization accuracy on much application [4].

2.5 Performance Evaluation

Not only performance evaluation is important to compare competing algorithms, but is an integral

part of the learning algorithm itself [2]. Performance evaluation is calculated as follow:

$$sensitivity = t - pos/pos \quad (8)$$

$$specificity = t - neg/neg \quad (9)$$

$$accuracy = (sensitivity(pos/(pos + neg))) + (specificity(neg/(pos + neg))) \quad (10)$$

where t-pos are the number of true positives, pos are the number of positive samples, t-neg are the number of true negatives, and neg is the number of negative samples.

This paper uses k-fold cross-validation for estimation of accuracy. In k-fold cross-validation, the initial data are randomly partitioned into k mutually exclusive subsets of folds, S_1, S_2, ---, S_k, each of approximately equal size. Training and testing is performed k times. In iteration i, the subset S_i is reserved as the test set, and the remaining subsets are used to the classifier. The accuracy estimate is the overall number of correct classification from the k iterations, divided by the total number of samples in the initial data [1].

3. Overview of the System

Firstly, the attribute's values are normalized by the normalization by decimal scaling process. In the second stage, features are selected with filter approach by using the sequential forward selection. In the third stage, the k-Nearest Neighbor classifier is built with the selected features and complete features. In this stage, finds the nearest neighbors of new instance by calculating the Euclidean distance measures of weighted function. At fourth stage, calculates the performance of k-Nearest Neighbor classifier with feature selection and without feature selection by using k-fold cross- validation. Figure 3 shows the system flow diagram.

4. Implementation

Water is important for life. This paper applies feature selection on the water dataset to classify water pollute or not. Water dataset is taken from the Department of Chemistry, Mandalay. Water dataset has 143 records with 14 features (attributes) such as turbidity, Total Filterable Residue, pH, total alkalinity, total hardness, calcium, potassium, toxic, iron, chloride, sulphate, nitrogen ammonia, oxygen dissolved and coliform. Maximum absolute value of attribute Total Filterable Residue is 1941. To normalize, the all values of attribute Total Filterable Residue are divided by 10000. Normalization by decimal scaling process normalizes the values of attributes in see Figure 4. Selected features are Total Filterable Residue, Total alkalinity, Total Hardness, Calcium, Chloride, Sulphate, Oxygen Dissolved and

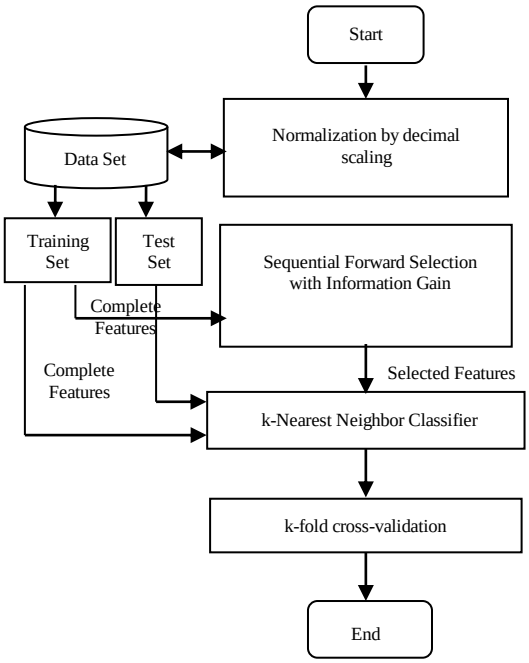

Figure 3. System Flow Diagram

Coliform, these are shown in Figure 5.The instance is classified with selected features subset by using k-Nearest Neighbor classifier and is also classified with the whole features in see Figure 6 and Figure 7. Sample implementation of k-Nearest Neighbor is shown below. Sample data set is shown in Table 1.

Table 1. Sample data set

Instances	Turbidity	Total Hardness	Sulphate	Calcium	Coliform	Class-label
1	5.0	8.5	0.4	0.0	180.0	Yes
2	5.0	8.5	0.2	0.0	180.0	Yes
3	5.0	7.3	1.3	0.3	180.0	Yes
4	2.0	7.3	0.0	0.0	0.0	No
5	5.0	7.3	0.2	0.0	0.0	No

If the attribute's values of new instance are entered, the nearest neighbors of the training instances are found by using the Euclidean distance. Let the attribute's values of new instance are turbidity=4.0, Total Hardness=8.5, Sulphate=0.4, Calcium=0.1 and Coliform=180. Distance between new instance and training instances are calculated as below:

$$d(N,1) = \sqrt{(4.0-5.0)^2 + (8.5-8.5)^2 + (0.4-0.4)^2 + (0.1-0.0)^2 + (180.0-180.0)^2} = 1.00$$

$$d(N,2) = \sqrt{(4.0-5.0)^2 + (8.5-8.5)^2 + (0.4-0.2)^2 + (0.1-0.0)^2 + (180.0-180.0)^2} = 1.02$$

$$d(N,3) = \sqrt{(4.0-5.0)^2 + (8.5-7.3)^2 + (0.4-1.3)^2 + (0.1-0.3)^2 + (180.0-180.0)^2} = 1.8$$

$$d(N,4) = \sqrt{(4.0-2.0)^2 + (8.5-7.3)^2 + (0.4-0.0)^2 + (0.1-0.0)^2 + (180.0-0.0)^2} = 180$$

$$d(N,5) = \sqrt{(4.0-5.0)^2 + (8.5-7.3)^2 + (0.4-0.2)^2 + (0.1-0.0)^2 + (180.0-0.0)^2} = 180$$

The weight measures are calculated as below:

$$w_1 = \frac{1}{d(N,1)} = \frac{1}{1.00} = 1.0$$

$$w_2 = \frac{1}{d(N,2)} = \frac{1}{1.02} = 0.98$$

$$w_3 = \frac{1}{d(N,3)} = \frac{1}{1.8} = 0.56$$

$$w_4 = \frac{1}{d(N,4)} = \frac{1}{180} = 0.0$$

$$w_5 = \frac{1}{d(N,5)} = \frac{1}{180} = 0.0$$

$$S_{yes} = 1.0 + 0.98 + 0.56 + 0.0 + 0.0 = 2.54$$

$$S_{no} = 0.0 + 0.0 + 0.0 + 0.0 + 0.0 = 0.0$$

The voting strength of class-label=yes is greater than the voting strength of class-label=no. So, the new instance is water pollute. The number of nearest neighbors of the instance showed depends on the value of k in k-Nearest Neighbor classifier.

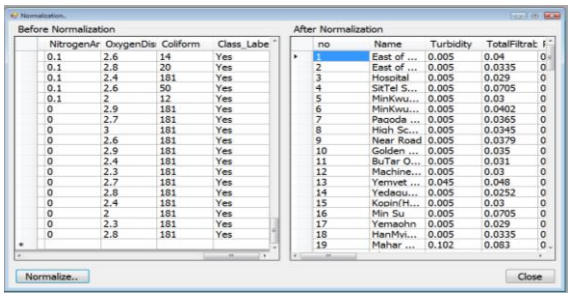

Figure 4. Normalization Form

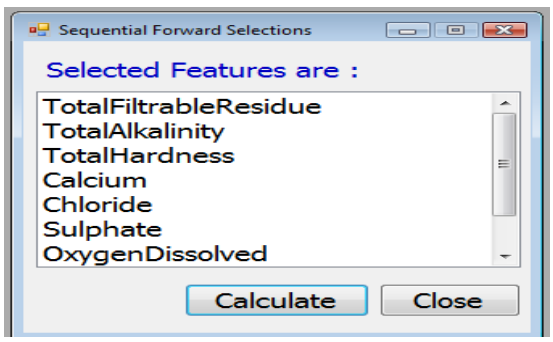

Figure 5. Sequential Forward Selection Form

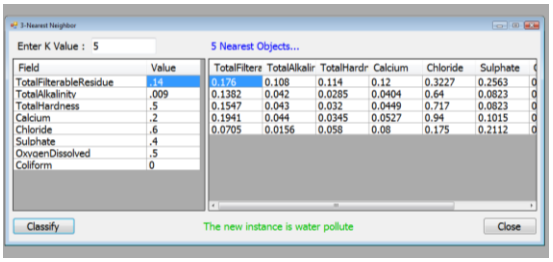

Figure 6. k-Nearest Neighbor Classifier Form with Selected Features

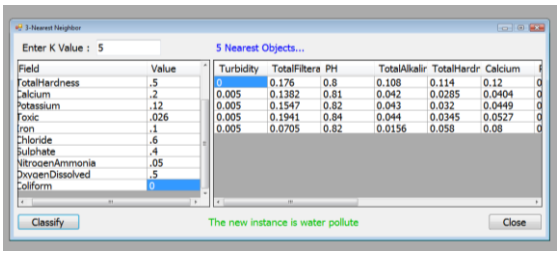

Figure 7. k-Nearest Neighbor Classifier Form with All Features

5. Experimental Result

Table 3 and Table 4 show the performance on the whole features and the selected features by using the accuracy, sensitivity and specificity according to k values. Water dataset has 143 records and the total numbers of attributes are 14, the selected features are 8 and the possible classes are 2 classes such as yes and no in see Table 2. The objective of this section is to minimize number of selected features and maximize learning accuracy on selected features. Feature selection can provide high degree of dimensionality reduction and enhance classification accuracy with predominant features. This paper calculates k-fold cross-validation with different values of k. Accuracy varies depends on the value of k. The result shows that the classifier with feature selection gives more efficient accuracy rate than the classifier without feature selection.

Table 2. Dataset used in experiment

Instances	# of complete features	# of selected features	Feature Type	Classes
143	14	8	Numeric	2

Table 3 . Performance of k-Nearest Neighbor classifier on the whole features

Cross-validation k-values	Sensitivity	Specificity	Accuracy
2	0.6306	0.0625	50%
3	0.6306	0.8125	67%
4	0.8739	0.3438	76%
5	0.8739	0.5938	81%
6	0.8829	0.6875	84%
7	0.8649	0.75	84%
8	0.9640	0.5938	88%
9	0.9430	0.625	89%
10	0.9009	0.8125	88%

Table 4. Performance of k-Nearest Neighbor classifier on the selected features

Cross-validation k-values	Sensitivity	Specificity	Accuracy
2	0.6036	0.1563	50%
3	0.6126	0.875	67%
4	0.8739	0.3438	76%
5	0.8829	0.5625	81%
6	0.9009	0.625	84%
7	0.8649	0.75	84%
8	0.9369	0.6875	88%
9	0.9550	0.6875	90%
10	0.8829	0.875	88%

6. Conclusion

This paper emphasizes a concept of feature selection, introduces an efficient way of analyzing feature redundancy. The main advantage may be that fewer features required for classification can be important for application such as water pollute or not where computational cost for selecting features may be high but it gives higher predictive accuracy and needs less time to test attributes. To classify water pollution or not needs many attributes, but by using feature selection reduces the dimensionality of the feature space and removed the redundant, irrelevant data. So, feature selection needs small amount of memory to store data set, reduces time and improves performance and accuracy. In future work, feature selection methods can apply on more data set, more classifiers such as Genetic Algorithm and more approach such as wrapper.

7. References

[1] H. John, Micheline Kamber, “Data Mining: Concepts and Techniques”, Simon Fraser University.

[2] K. YongSeog , W.Nick Street, and Filippo Menczer, "Feature Selection in Data Mining", University of Lowa, USA.

[3] M. C. Lius , Lluis Belanche, Angela Nebot, “Feature Selection Algorithms: A Survey and Experimental Evaluation”, Universital Politecnica de Catalunya, Departent de Llenguatges I Sistemes Informatics, Jordi Girona 1-3, C6, 08034, Barcelona, {Spain, lcmolina,belanche,angela}@lsi.upc.es

[4] P. Sabai, “A Hybrid Approach to Genetic Algorithm and k-Nearest Neighbor Classifier on Information Based Distance Metric”, University of Computer Studies, Yangon, 2004.

[5] S. Martin , “Feature Selection”, 2007.

[6] S.Thiri, "Chemical Investigation on Drinking Water in Myit-Thar Area", University of Mandalay, June, 1998.

[7] T. Larose Damiel, “k-Nearest Neighbor Algorithm”, Discovering Knowledge in Data: An Introduction to Data Mining, ISBN 0-471-66657-2 Copyright 2005.