International Journal of Advanced Computer Science and Technology.
ISSN 2249-3123 Volume 8, Number 1 (2018), pp. 1-13

http://www.ripublication.com

Data Reduction Techniques: A Comparative Study for Attribute Selection Methods

Dr. Kadhim B.S. Al Janabi
Department of Computer Science
Faculty of Computer Science and Mathematics
University of Kufa

Rusul Kadhim
Department of Computer Science
Faculty of Computer Science and Mathematics
University of Kufa

Abstract

Feature selection is an important factor in the success of the data mining process through selecting the useful or relevant attributes in the data set. This paper presents a comparison between feature selection methods and their impact on learning algorithms. A decision tree leaner, REPTree is used to measure the effectiveness of selected features. Different search methods were used with each feature selection method to select an optimal subset of features including: forward, backward, floating, branch-and-bound, and randomized. The comparison was conducted on a data set collected from Najaf Education Directorate in Iraq. The results showed that most methods of selecting the features improved the accuracy and the performance of the classification techniques.

Correlation based Feature Selection(CFS), Information Gain (IG) attribute evaluation, Gain ratio(GR) attribute evaluation, Symmetrical Uncertainty, Chi-Squared, Relief F, One-R, Classifier Subset Evaluator and Wrapper Subset Evaluator are the main feature selection techniques discussed in this paper.

Keywords: Data Reduction, Feature Selection, Mining Techniques

1. INTRODUCTION

Attribute /Feature selection methods are used to reduce the dimensionality of the data through removing the redundant and irrelevant attributes in a data set. Feature selection methods are categorized according to the feature evaluation measure, depending on the type into filter and wrapper. Filter methods depend on the general characteristics of the data to select feature subsets and evaluate the quality of these subsets independently of the classification algorithm. The wrapper uses a predefined classification algorithm to evaluate the subsets of selected features. According to the way in which the features are evaluated, Feature selection methods are categorized into single and subset evaluation. In single evaluation, sometimes called feature ranking or feature weighting, each feature is evaluated individually through assigning weight to each feature according to its degrees of relevance while in subset evaluation, evaluates each subset of features instead of the individual features[1]. Wrappers need to invoke the learning algorithm to evaluate each subset of the selected features, so it often gives better results than the filter methods, but this makes them more expensive than the filter methods. Implementing the filter methods is much faster, so it's the best methods for large size databases but these methods ignore the correlation between features that affects performance[2][3].

The feature selection process has many benefits: It allows visualization and understanding of data more easily, reduces the time and storage required for the mining process and improves the performance of the algorithms through avoiding the curse of dimensionality[4].

The feature selection process consists of the following four steps as shown in Figure 1-1:

1- Subset Generation: In this step, subsets of features are generated for the evaluation based on a certain search strategy (starts with either an empty subset of features or with all features or with a random subset of features).

2- Subset Evaluation: To measure the quality of the subset generated by some generation criterion, an evaluation function is used. the subset generated is compared with the previous best and replaced with it, if it is found to be better.

3- Stopping Criterion: The process of feature selection may continue to generate feature subsets therefore, a stop criterion is required for this process. For example, stopping at a predetermined number of features/iterations or when adding/deleting a feature reduces performance or when obtaining an optimal subset of features.

4- Result Validation: At this step, different tests are conducted to verify the validity of the subset of the selected features, in addition to comparing the resulting subset with the previously identified or with the subsets resulting from competing feature selection methods using artificial datasets, real-world datasets, or both[5]

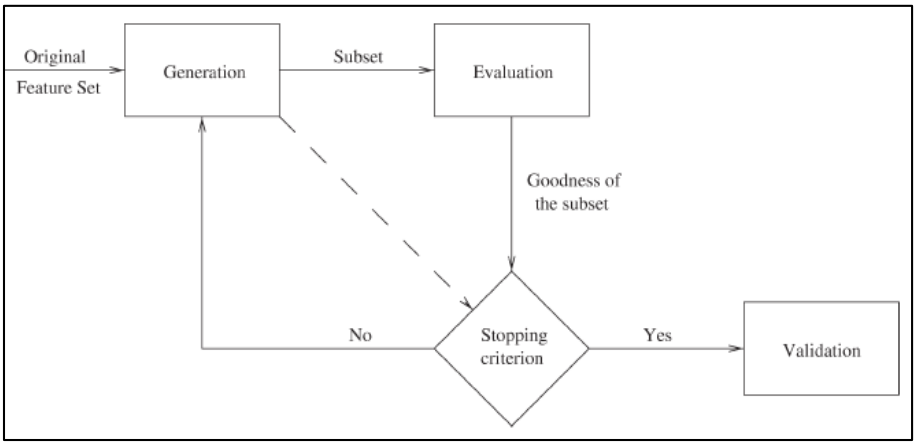

Figure 1-1 Steps of Feature Selection Process[5]

2. RELATED WORKS

In 2015,I. M. El-hasnony and etal[6] conducted a comparative study among five of the data reduction techniques that are gain ratio, principal components analysis, correlation feature selection, rough set attribute selection and fuzzy rough feature selection. such comparison was based on the accuracy of the classification and the results showed that fuzzy rough feature selection surpassed other techniques.

In 2015,O. Villacampa[7]compared a set of the feature selection methods that are Information Gain, Correlation Based Feature Selection, Relief-F, Wrapper, and Hybrid methods used to reduce the number of features in the data sets. Three of the common classification algorithms (Decision Trees, k-Nearest Neighbor and Support Vector Machines) are used as classifiers to evaluate the performance of these methods. The results showed that the Relief-F method outperformed all other feature selection methods.

In 2014,R. Porkodi[8] analyzed five techniques to reduce features and the five techniques were ReliefF, Information Gain, Gain Ratio, Gini Index and Random Forest. The experimental results showed that Random Forest outperforms other techniques.

In 2010, A. G. Karegowda and etal[9] presented two filters for selecting the relevant features were Gain ratio and Correlation based feature selection. The search method used with CFS filter is the genetic algorithm. The selected features were tested by two classification algorithms are Back propagation neural network and Radial basis function network and the results showed that the features selected by CFS filter gave higher classification accuracy of those selected by information gain filter.

In 2004, Y. Wang and F. Makedon[10]applied one of the feature selection methods is ReliefF to reduce the number of features in three cancer classification data sets and to evaluate the performance of the selected features, they used two famous classification algorithms are linear SVM and the k-NN and also compared the performance of the used method with three methods for feature selection are Information Gain, Gain

Ratio, and χ2-statistic. The results showed that the performance of the applied method was similar to the performance of the other three methods.

3. FEATURE SELECTION METHODS

Different techniques and methods are available for feature selection, each has its own advantages, drawbacks and characteristics.

3.1 Correlation based Feature Selection(CFS)

CFS is a feature subset selection algorithm. This algorithm computes the correlation between all features and the output class and selects the best feature subset(i.e the subset with features highly correlated with the class variable and have low correlation with each other features) using correlation based heuristic evaluation function. CFS measures correlation between nominal or categorical features, so discretization is used in the case of numeric features. CFS is given by the equation 1.

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}} \quad \ldots(1)$$

where r_{zc} is the correlation (dependence) between features and the class variable, k is the number of features , $\overline{r_{zi}}$ is the average of the correlation between feature-class and $\overline{r_{ii}}$ is the average inter-correlation between feature-feature[2].

3.2 Information Gain (IG) attribute evaluation

IG is a feature selection measure introduced by J. R. Quinlan[36] and used to select the test attribute at each node of the decision tree in the basic algorithm for building a decision tree(ID3).

Let node *N* represent or hold the records of partition *D* from the dataset. Calculating IG helps in choosing the splitting attribute for node *N*, where the attribute of *N* having maximum IG is the one chosen for splitting. This attribute with higher IG minimizes the information needed to classify the objects in the resulting partitions and reflects the least randomness or "impurity" in these partitions. Such an approach optimizes the performance of DT algorithm and guarantees that a simple tree is found.

The expected information needed to classify an object in partition D is given by :

$$Info(D) = -\sum_{i=1}^{m} p_i \log(p_i) \quad \ldots(2)$$

where p_i represents the probability that an object in *D* belongs to class $C_i$and is calculated by $\left|C_{i,D}\right| / |D|$. As the information is encoded in bits, a log function to the

base 2 is used. *Info*(*D*) represents the average amount of information needed to find out the class label of an object in partition *D*.

Now, if it is required to partition the objects in *D* on some feature (attribute) *A* having *v* distinct values, $\{a_1, a_2, a_3, \dots, a_v\}$ (e.g. for attribute Gender the distinct values are Male and Female), the expected information based on the partitioning into subsets by attribute *A*, is given by :

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad \dots(3)$$

Where $\frac{|D_j|}{|D|}$ acts as the weight of the *j*th partition. IG is defined as the difference between the original information before splitting or partitioning and the new IG obtained after partitioning on *A* as given in equation (4).

$$Gain(A) = Info(D) - Info_A(D) \quad \dots(4)$$

In other words, *Gain(A)* tells us how much would be gained by branching on *A*. It represents the expected reduction in the information requirement caused by knowing the value of feature *A* [28].

3.3 Gain Ratio(GR) attribute evaluations

GR is an extension to IG measure and used in the decision tree based learning algorithm,C4.5[37].This measure overcomes bias of IG toward the features with the large number of values by applying normalization to information gain as follow:

$$SplitInfo_A(D) = -\sum_{j=1}^{v}(|D_j|/|D|)log_2(|D_j|/|D|) \quad \dots(5)$$

The resulting value from the above equation represents the information obtained by splitting the attribute *A* in the training data set *D* into *v* partitions corresponding to v outputs. The gain ratio is defined as[28]

$$GainRatio(A) = Gain(A)/SplitInfo(A) \quad \dots(6)$$

3.4 Symmetrical Uncertainty

symmetrical uncertainty criterion overcomes the bias of information gain towards the features with the more values by normalizing its value to the range[0,1].It is given by the following equation:

$$\mathrm{SU} = 2\frac{\mathrm{Gain(A)}}{\mathrm{I}info(D) + \mathrm{SplitInfo(A)}} \quad \dots(7)$$

If SU value=0, indicates that the two features have no association whereas the value of SU=1 indicates that knowledge of one feature completely predicts the other. This criterion is similar to GR criterion in favor of features with fewer values[14].

3.5 Chi-Squared

Chi-Squared is a simple and general algorithm that uses the x^2 statistic to discretize the numeric features through sorting the features according to their values and calculating the x^2 value as in equation(8) for each pair of adjacent intervals in the feature and merging the pair of adjacent intervals with the lower x^2 value and continuing the process until inconsistencies are found in the data(the attribute with only one value, it removed). Chi-squared is given by the following equation:

$$x^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad ...(8)$$

Where O_{ij} is the observed frequency and E_{ij} is the expected frequency[15].

3.6 One-R

OneR (One Rule) is a simple algorithm that evaluates the features according to error rate. This algorithm generates a set of candidate rules(one rule for each feature in the data set) and selects the rule with the least error ratio. It deals with the categorical features, so if the features have numeric values, it uses a straightforward method to divide the range of values into several separate intervals. It handles missing values by treating "missing" as a legitimate value[16].

3.7 Relief-F

It is extension of Relief algorithm .It is adapted to work with multi-class problems by finding one or more (*k*) neighboring instance *M(C)* from each different class *C* and averages their contribution for upgrading estimates *W[A]* weighting it with the prior probability of each class. The estimation of weight *W* of feature *A* when the sampled instance is *R* (which is sampled m times) and the nearest instance of the same class is *H* is conducted as shown in the following equation

$$W[A] := W[A] - \sum_{j=1}^{k} \frac{diff(A, R, H_j)}{m.k} + \sum_{C \neq class(R)} \frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^{k} \frac{diff(A, R, M_j(C))}{m.k} \quad \ldots(9)$$

The number of the checked neighboring instances is determined by either predefining a number or the maximum distance. If the features are categorical then the difference *diff(A, I1, I2)* is one if the values of the instances are equal and if the values are different then the difference is 0.In the case of numerical features, then the difference is calculated by the following equation[17]:

$$diff(A, I1, I2) = \frac{|value(A, I1) - value(A, I2)|}{\max(A) - \min(A)} \quad ...(10)$$

3.8 Consistency Subset Evaluator (CSE)

It evaluates feature subsets by the degree of consistency in class values when the training instances are projected onto the set, i.e. the prevalence of one class in subsets that the data set is divided into by attribute values. This also means that feature values have to be discretized. Consistency of a subset can never surpass that of the full set so the algorithm searches for the smallest subset which has the same consistency as the full set.

The consistency of a feature subset S in a data set with N instances is calculated using the following equation:

$$C_s = 1 - \frac{\sum_{i=0}^{J} |D_i| - |M_i|}{N} \quad ...(11)$$

where *J* is the number of distinct attribute value combinations,

|*Di*| is the number of occurrences of the *i*-th attribute value combination,

|*Mi*| is the cardinality of the majority class for the *i*-th attribute value combination[17]

3.9 Classifier Subset Evaluator

Evaluates feature subsets by applying a classification training data or a separate hold out testing set and using this classifier to estimate the accuracy of these subsets.

3.10 Wrapper Subset Evaluator

It is a feature subset evaluator that uses the learning algorithm as a function to evaluate these subsets and to estimate the accuracy of the learning algorithm, it uses cross validation[3].

4. SEARCH METHODS

To select an optimal subset of features, different search strategies are used: forward, backward, floating, branch-and-bound, and randomized. The strategy that starts with an empty subset of features and adds the relevant features to the subset, it is called forward selection whereas that starts with all features and deleting the irrelevant features, this strategy is called backward selection, but these exhaustive strategies become costly to work when the number of features increases where have time complexity is exponential, so more efficient heuristic strategies are used (*sequential forward selection and sequential backward selection*). In *sequential forward selection*, one feature is selected at a time even adding another feature to the subset reduces the subset quality and the selected features can't be deleted later while *sequential backward selection*, one feature is deleted at a time and the deleted feature can't be re-selected. Most heuristic search strategies such as Greedy sequential search, best-first search, and genetic algorithm have time complexity is quadratic in terms of data dimensionality but do not ensure finding the optimal features for this

reason, it is replaced by a randomized search strategy[1].

5. REPTree ALGORITHM

REPTree builds a decision/regression tree using information gain/variance and prunes it using reduced error pruning. It splits instances into pieces to handle the missing values. Optimized for speed, REP Tree arranges the numeric values for each numeric attribute only once[18].

6. WEKA TOOL

To implement the attribute selection methods, We used WEKA[19] an open source software written in the Java programming language and developed at the University of Waikato in New Zealand. Weka is a collection of machine learning algorithms for data mining tasks and includes tools for data preprocessing, classification, regression, clustering, association rules, and visualization. And we used learning algorithm, REPTree to evaluate the subset of selected features. Feature selection methods in WEKA consist of two elements: Feature evaluator, the way by which feature subsets are evaluated and Search method, the way by which space of feature subsets are searched.

7. CONCLUSION AND RESULTS

In this paper, feature selection methods are applied on real dataset that contains five attributes. More than (18266 records) about schools, students and teachers were collected. The results showed that three features, **Level, Schooltype and City** are the most significant features for determining the dependent variables, and hence feature reduction can be applied.

Attribute evaluator	**Search method**	***No of selected attributes***	***No of cancelled attributes***	***Selected attributes***		***Time complexity m: number of records n: number of attributes t: number of distinct values for the attribute***	***Type***	***Description***	***10-Fold Cross Validation using REP Tree***	
				Names	***No of distinct values***				***Accuracy before applying the evaluator***	***Accuracy after applying the evaluator***
CfsSubsetEval	***BestFirst, ExhaustiveSearch, GeneticSearch, GreedyStepwise, LinearForwardSelection, RankSearch, Scatter Search, SubsetSizeForwardSelection***	***4***	***1: Year***	***City Schooltype Gender level***	***10 10 2 23***	O(m,n,t) ∝*(2174,4,(10,10,2,23))*	***Filter model***	***Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them where it finds subsets of features that are highly correlated with the class while having low intercorrelation are preferred .***	***77.2309 %***	***78.4729 %***
	RandomSearch	***3***	***2: Year, Schooltype***	***City, Gender, Level***	***10 2 23***	O(m,n,t) ∝*(2174,3,(10,2,23))*				***66.6053 %***
ChiSquaredAttributeEval	***Ranker***	***4***	***1: Year***	***Level Schooltype City Gender***	***23 10 10 2***	O(m,n,t) ∝*(2174,4,(23,10,10,2))*	***Filter model***	***Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class .***	***77.2309 %***	***78.4729 %***
ConsistencySubsetEval	***BestFirst, ExhaustiveSearch, GeneticSearch, GreedyStepwise, LinearForwardSelection, RandomSearch, RankSearch, Scatter Search, SubsetSizeForwardSelection***	***4***	***1: Year***	***City Schooltype Gender Level***	***10 10 2 23***	O(m,n,t) ∝*(2174,4,(10,10,2,23))*	***Filter model***	***Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes .***	***77.2309 %***	***78.4729 %***

GainRatioAttributeEval	***Ranker***	***4***	***1: Year***	***Schooltype Level City Gender***	***10 23 10 2***	O(m,n,t) ∝*(2174,4,(10,23,10,2))*	***Filter model***	***Evaluates the worth of an attribute by measuring the gain ratio with respect to the class . GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute) .***	***77.2309 %***	***78.4729 %***
InfoGainAttributeEval	***Ranker***	***4***	***1: Year***	***Level Schooltype City Gender***	***23 10 10 2***	O(m,n,t) ∝*(2174,4,(23,10,10,2))*	***Filter model***	***Evaluates the worth of an attribute by measuring the information gain with respect to the class . InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute)***	***77.2309 %***	***78.4729 %***
OneRAttributeEval	***Ranker***	***4***	***1: Year***	***Schooltype Level City Gender***	***10 23 10 2***	O(m,n,t) ∝*(2174,4,(10,23,10,2))*	***Filter model***	***Evaluates the worth of an attribute by using the OneR classifier .***	***77.2309 %***	***78.4729 %***
ReliefFAttributeEval	***Ranker***	***4***	***1: Year***	***Level Schooltype City Gender***	***23 10 10 2***	O(m,n,t) ∝*(2174,4,(23,10,10,2))*	***Filter model***	***Evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Can operate on both discrete and continuous class data .***	***77.2309 %***	***78.4729 %***
SymmetricalUncertAttributeEval	***Ranker***	***4***	***1: Year***	***Level Schooltype City Gender***	***23 10 10 2***	O(m,n,t) ∝*(2174,4,(23,10,10,2))*	***Filter model***	***Evaluates the worth of an attribute by measuring the symmetrical***	***77.2309 %***	***78.4729 %***

								uncertainty with respect to the class . ***SymmU(Class, Attribute) = 2 * (H(Class) - H(Class | Attribute)) / H(Class) + H(Attribute) .***		
WrapperSubsetEval	***BestFirst, ExhaustiveSearch,GeneticSearch, GreedyStepwise, LinearForwardSelection, RaceSearch, RandomSearch, RankSearch, SubsetSizeForwardSelection,***	***4***	***1: Year***	***CitySchooltypeGenderLevel***	***10 10 2 23***	O(m,n,t) ∝*(2174,4,(10,10,2,23))*	***Wrapper model***	***Evaluates attribute sets by using a learning scheme. Cross validation is used to estimate the accuracy of the learning scheme for a set of attributes***	***77.2309 %***	***78.4729 %***
	Scatter Search	***5***		***YearCitySchooltypeGenderLevel***	***5 10 10 2 23***	O(m,n,t) ∝*(2174,5,(5,10,10,2,23))*				***77.2309 %***
ClassifierSubsetEval	***BestFirst, ExhaustiveSearch,GeneticSearch, GreedyStepwise, LinearForwardSelection, RaceSearch, RankSearch, Scatter Search, SubsetSizeForwardSelection***	***5***		***YearCitySchooltypeGenderLevel***	***5 10 10 2 23***	O(m,n,t) ∝*(2174,5,(5,10,10,2,23))*	***Wrapper model***	***Evaluates attribute subsets on training data or a seperate hold out testing set. Uses a classifier to estimate the 'merit' of a set of attributes .***	***77.2309 %***	***77.2309 %***
	RandomSearch	***4***	***1: Year***	***CitySchooltypeGenderLevel***	***10 10 2 23***	O(m,n,t) ∝*(2174,4,(10,10,2,23))*			***77.2309 %***	***78.4729 %***

Figure 1.2 Comparison of Feature Selection Methods

REFERENCES

[1] H. Liu and H. Motoda, Computational methods of feature selection. CRC Press, 2008.

[2] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," no. April, 1999.

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artif. Intell., vol. 97, no. 1–2, pp. 273–324, 1997.

[4] A. e E. Isabelle Guyon, "An Introduction to Variable and Feature Selection," J. ofMachine Learn. Res., vol. 3, no. Mar, pp. 1157–1182, 2003.

[5] M. Dash and H. Liu, "Feature Selection for Classification," Intell. data Anal., vol. 1, no. 1–4, pp. 131–156, 1997.

[6] I. M. El-hasnony, H. M. El Bakry, and A. A. Saleh, "Comparative Study among Data Reduction Techniques over Classification Accuracy," vol. 122, no. 2, pp. 8–15, 2015.

[7] O. Villacampa, "Feature Selection and Classification Methods for Decision Making : A Comparative Analysis," Nova Southeastern University, 2015.

[8] R. Porkodi, "COMPARISON OF FILTER BASED FEATURE SELECTION ALGORITHMS : AN OVERVIEW," pp. 108–113, 2014.

[9] A. G. Karegowda, A. S. Manjunath, G. Ratio, and C. F. Evaluation, "COMPARATIVE STUDY OF ATTRIBUTE SELECTION USING GAIN RATIO," vol. 2, no. 2, pp. 271–277, 2010.

[10] Y. Wang and F. Makedon, "Application of Relief-F Feature Filtering Algorithm to Selecting Informative Genes for Cancer Classification using Microarray Data," in Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE, 2004, pp. 497–498.

[11] J. R. QUINLAN, "Induction of Decision Trees,"*Machine Learning*, pp. 81–106, 1986.

[12] J. H. and M. Kamber, "DataMining: Concepts and Techniques., " 2nd Edition, Elsevier, 2006.

[13] J. R. QUINLAN, "C4.5: Programs for machine learning., " Morgan Kaufmann, 1993.

[14] M. A. Hall and L. A. Smith, "Practical Feature Subset Selection for Machine Learning," 1998.

[15] Huan Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence, 1995, pp. 388–391.

[16] R. C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets," Mach. Learn., vol. 11, no. 1, pp. 63–91, 1993.

[17] I. Polaka, "Feature Selection Approaches in Antibody Display Data Analysis," in 8th International Scientific and Practical Conference, 2011, vol. 2, pp. 16–23.

[18] I. H. W. and E. Frank, "Data Mining Practical Machine Learning Tools and Techniques, " 2nd Edition, ISBN: 0-12-088407-0, Elsevier, 2005.

[19] "WEKA: Waikato environment for knowledge analysis." http://www.cs.waikato.ac. nz/ml/weka.