CONCEPTION OF DATA PREPROCESSING AND PARTITIONING PROCEDURE FOR MACHINE LEARNING ALGORITHM

Himanshu Shrivastava, Srivatsan Sridharan

P.G. Student - M.Tech
IIIT – Bangalore, Bangalore, India.
himanshu17shrivastava@gmail.com, vatsan.s@rediff.com

Abstract- This paper mainly deals with the preprocessing of the data used as an input for any machine learning algorithm. The main success behind any machine learning algorithm is based on the quality of the input data used. Though many factors affect the success of Machine Learning (ML) on a given task, still the representation and quality of the instance data attributes for the main success of algorithm. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. It is well known that data preparation and filtering steps take considerable amount of processing time in ML problems. The product of data pre-processing is the final training set. Thus, this paper presents the algorithms for each step of data pre-processing so that one achieves the best performance for their data set.

Index Terms: Data Mining, Data Cleaning and Feature Selection.

I. INTRODUCTION

The data preprocessing can often have a significant impact on generalization performance of a supervised ML algorithm. The elimination of noise[1] instances is one of the most difficult problems in any machine learning. The excessively deviating features are also referred to as outliers. An approach that is being handled for the infeasibility of learning from very large data sets is to select a single sample from the large data set. Missing data handling is another issue often dealt with in the data preparation steps. Any real-world problems involve both symbolic and numerical features. Therefore, there is an important issue of discretizing numerical or continuous features. The features with too many values are overestimated in the process of selecting the most informative features, both for inducing decision trees and for deriving decision rules. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modeling.

Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. The accuracy on classification can be improved. The result can be easily interpreted representation of the target concept. Furthermore, the problem of feature interaction can be addressed by constructing new features from the basic feature set. Transformed features generated by feature construction may provide a better discriminative ability than the best subset of given features. This paper addresses issues of data pre-processing that can have a significant impact on generalization performance of a Supervised Machine Learning algorithm.

II. HANDLING MISSING DATA

There are some techniques to handle specifically the missing data that would be most appropriate for the Machine Learning Algorithm[2]. Method of Ignoring Instances with Unknown Feature Values. This method is the simplest by ignoring the instances, which have at least one unknown feature value. Most Common Feature Value. The value of the feature that occurs most often is selected to be the value for all the unknown values of the feature. This time the value of the feature, which occurs the most common within the same class is selected to be the value for all the unknown values of the feature. This could not be a relevant preprocessing technique for any machine learning algorithm as the problem of accuracy arises. For instance the, consider performing an classification algorithm on any sensitive data like cancerous data, filling of the most recent values may lead to incorrect classification. Some important data sets may be shown as even outlier in a classification. So this could not be a suitable solution.

Mean substitution. Substitute a feature's mean value computed from available cases to fill in missing data

values on the remaining cases. A smarter solution than using the "general" feature mean is to use the feature mean for all samples belonging to the same class to fill in the missing value. Regression or classification methods. Develop a regression or classification model based on complete case data for a given feature, treating it as the outcome and using all other relevant features as predictors. Hot deck imputation. Identify the most similar case to the case with a missing value and substitute the most similar case's Y value for the missing case's Y value. Method of Treating Missing Feature Values as Special Values. Treating unknown itself as a new value for the features that contain missing values. These would be classified as outliers in the later stage.

III. DISCRETIZATION

Discretization should significantly reduce the number of possible values of the continuous feature since large number of possible feature values contributes to slow and ineffective process of inductive ML[3],[4]. The problem of choosing the interval borders for the discretization of a numerical value range remains an open problem in numerical feature handling.

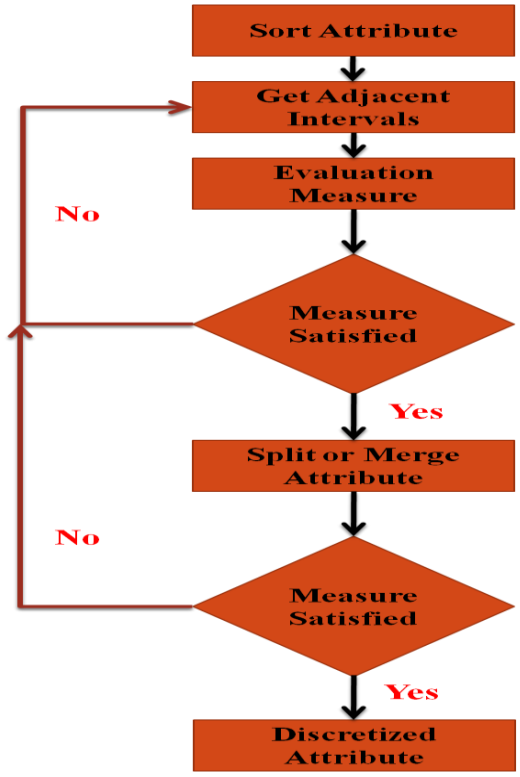

Fig. 1. Discretization Process.

Most discretization methods are divided into top-down and bottom-up methods. Top down methods start from the initial interval and recursively split it into smaller intervals. Bottom up methods start from the set of single value intervals and iteratively merge neighboring intervals. Some of these methods require user parameters to modify the behavior of the discretization criterion or to set up a threshold for the stopping rule.

IV. NORMALIZATION

Normalization[7] is a "scaling down" transformation of the features. Within a feature there is often a large difference between the maximum and minimum values, e.g. 0.01 and 1000. When normalization is performed the value magnitudes and scaled to appreciably low values. This is important for many neural network and k-Nearest Neighborhood algorithms. The two most common methods for this scope are min-max normalization and z-score normalization. This is applied rarely for input data of ML algorithms - especially for spatial data.

$$v' = (v - min)(new\_max - new\_min)/(max - min) + new\_min.$$

$$v' = (v - mean)/std\_devn.$$

Where v' is the new feature value and v is the old feature value.

V. FEATURE SELECTION

Feature subset selection is the process of identifying and removing possible irrelevant and redundant features[5]. This reduces the dimensionality of the data and enables learning algorithms to operate faster and more effectively. Generally, features are characterized as, Relevant. These are features have an influence on the output and their role cannot be assumed by the rest. Irrelevant. Irrelevant features are defined as those features not having any influence on the output, and whose values are generated at random for each example. Redundant. A redundancy exists whenever a feature can take the role of another.

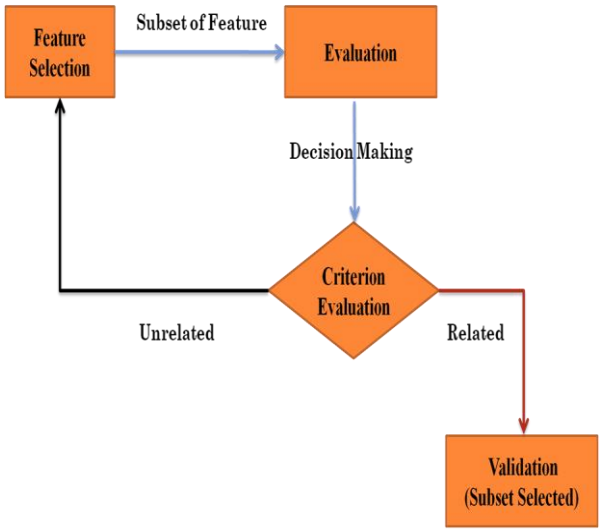

Fig. 2. Feature Selection Process.

Feature Selection algorithms in general have two components[6], a selection algorithm that generates proposed subsets of features and attempts to find an optimal subset and an evaluation algorithm that determines how good a proposed feature subset is, returning some measure of goodness to the selection algorithm. However, without a suitable stopping criterion the Feature Selection process may run

exhaustively or forever through the space of subsets. Stopping criteria can be whether addition or deletion of any feature does not produce a better subset and whether an optimal subset according to some evaluation function is obtained. Ideally, feature selection methods search through the subsets of features, and try to find the best one among the competing 2N candidate subsets according to some evaluation function.

However, this procedure is exhaustive as it tries to find only the best one. It may be too costly and practically prohibitive, even for a medium-sized feature set size (N). Other methods based on heuristic or random search methods attempt to reduce computational complexity by compromising performance. Filter methods are independent of the inductive algorithm, whereas wrapper methods use the inductive algorithm as the evaluation function. The filter evaluation functions can be divided into four categories: distance, information, dependence and consistency.

Distance. For a two-class problem, a feature X is preferred to another feature Y if X induces a greater difference between the two-class conditional probabilities than Y. Information. Feature X is preferred to feature Y if the information gain from feature X is greater than that from feature Y. Dependence. The coefficient is a classical dependence measure and can be used to find the correlation between a feature and a class. If the correlation of feature X with class C is higher than the correlation of feature Y with C, then feature X is preferred to Y. Consistency. Two samples are in conflict if they have the same values for a subset of features but disagree in the class they represent.

VI. FEATURE CONSTRUCTION

The problem of feature interaction can be also addressed by constructing new features from the basic feature set. This technique is called feature construction/transformation[7]. The new generated features may lead to the creation of more concise and accurate classifiers. In addition, the discovery of meaningful features contributes to better comprehensibility of the produced classifier, and better understanding of the learned concept.

Assuming the original set A of features consists of a1, a2... an, some variants of feature transformation. Feature transformation process can augment the space of features by inferring or creating additional features. After feature construction, we may have additional m features a_{n+1}, a_{n+2}... a_{n+m}. For example, a new feature a_k ($n < k \leq n + m$) could be constructed by performing a logical operation of a_i and a_j from the original set of features.

VII. DATA PARTITIONING

Any machine learning algorithms input data will be divided into two types of data sets namely - training and test data sets. Training Dataset. The training dataset contains the input data together with correct or expected output. This dataset is usually prepared by collecting some data in semi-automated way. It is important that there is expected output label associated with the dataset, because it is used for supervised learning. Approximately 60% of the original dataset constitutes the training dataset. Test Dataset. The remaining data is taken as the test dataset. This is the data that validates the underlying model. For cross-validation purpose, the test dataset is in-turn divided into subsets which can then be used to the test the accuracy of the chosen model effectively. Finally the ML algorithm is applied in such a way that after completion of all the iterations in the algorithm the same data set is used as training and test data at different instance of iterations of the same algorithm.

Algorithm 1: FEATURE SUBSET SELECTION ALGORITHM

```
Data: dataFileFeatures[ ], thresholdFileFeatures[ ]
Result: Common feature set features[ ]
k=0;
features[ ]=null ;
for i from 0 to dataFileFeatures.size()-1 do
    for j from 0 to thresholdFileFeatures.size()-1 do
        if (dataFileFeatures[i]=thresholdFileFeatures[j]) and (get-
        DataType(dataFileFeatures[i])=getDataType(thresholdFileFeatures[j]))
        then
            features[k++] ← cols1[i];
            break;
        end
    end
end
```

Fig. 3. Feature Selection Algorithm.

The following are the steps involved in data partitioning. Centroid Generation. This part of the data pre-processing algorithm calculates the Centroid among all the distributed dataset points and stores the coordinates as (Centroid x, Centroid y) points in the space. Intercept Generation. This part of the data pre-processing algorithm generates a random point in either x-axis or y-axis, and correspondingly generates its intercept on the alternate axis using the principle of parallelogram, thereby dividing the dataset into training and testing dataset. Training and Testing Dataset Division. This part of the data pre-processing algorithm counts the number of training and test datasets on either side of the intercept.

VIII. IMPLEMENTATION ISSUES

The available raw sets of data are subjected to preprocessing before the analysis. Data cleansing is carried out in order to make sure that the dataset has relevant features i.e. common attributes among the different types of data sets, following which the data set is divided into training and testing datasets. The process of cleansing is carried out by feature subset algorithm where the attributes are analyzed for data type as well as the value each data sets holds. The division of the given

dataset into training and testing dataset is done using Centroid algorithm where the Centroid of all the given dataset points calculated and an intercept is generated. Based on the position of each dataset point on the hyperplane, it is classified as either training dataset or test dataset. Approximately sixty percent of the dataset points are grouped as training dataset and the remaining forty percent is grouped as test dataset.

Feature Subset Selection Algorithm. For each features the feature names and data types are compared from the data file and the threshold file. If they are same then it is added to the feature set. After this filtering step, the relevant features are selected and written to a file along with the associated output label. The relevant features is only thus extracted finally.

Centroid Generation Algorithm. Random coordinates are generated for each sample row data, and starting values are assigned to arrays max x[ ], min x[ ], max y[ ], min y[ ]. For each Sample data, assign maximum value of random coordinates to max x and max y and minimum values to min x and min y. Assign the average of min x, max x and min y, max y to Centroid x and Centroid y respectively. This algorithm provides the input for the algorithm that divides the datasets into two divisions - training and test data sets.

Algorithm 2: CENTROID GENERATION

```
Data: x_coordinate[ ], y_coordinate[ ], max_x[ ], max_y[ ], min_x,
      min_y
Result: centroid_x, centroid_y
max_x[] ← rand_x_coordinate[0];
max_y[] ← rand_y_coordinate[0];
min_x[] ← rand_x_coordinate[0];
min_y[] ← rand_y_coordinate[0];
begin
    for i=1 to SampleDataRowCount-1 do
        if rand_x_coordinate[i] ≥ max_x and
        rand_y_coordinate[i] ≥ max_y then
            max_x = rand_x_coordinate[i];
            max_y = rand_y_coordinate[i];
        end
        if rand_x_coordinate[i] < min_x and rand_y_coordinate[i] <
        min_y then
            min_x = rand_x_coordinate[i];
            min_y = rand_y_coordinate[i];
        end
    end
    centroid_x = ( max_x + min_x ) / 2 ;
    centroid_y = ( max_y + min_y ) / 2 ;
end
```

Fig. 4. Centroid Generation Algorithm.

Training and Test Data Set Division Algorithm. The line joining the intercepts coordinate1 and coordinate 2 will be the boundary for dividing the dataset into training and test samples. If the count of the coordinates generated during Centroid generation algorithm above this boundary is high, consider the rows corresponding to these coordinates as training data set. The rows corresponding to rest of the coordinates below the line are the test data set and vice-versa.

Algorithm 3: INTERCEPT GENERATION

```
Data: rand_x, rand_y, centroid_x, centroid_y
Result: coordinate1, coordinate2
begin
    begin
        if rand_x > rand_y then
            coordinate1 = rand_x;
            flag0 = 1 ;
        else
            coordinate1 = rand_y;
            flag1 = 1 ;
        end
        if flag0 == 1 then
            coordinate2 = (( coordinate1 * centroid_y ) / centroid_x ) -
            centroid_y;
        end
        if flag1 == 1 then
            coordinate2 = (( coordinate1 * centroid_x ) / centroid_y ) -
            centroid_x;
        end
```

Fig. 5. Intercept Generation Algorithm.

Algorithm 4: TRAINING AND TESTING DATASET DIVISION

```
Data: rand_x_coordinate, rand_y_coordinate , coordinate,
      final_coordinate, SampleDataRowCount
Result: Sampledataset1[ ], Sampledataset2[ ]
begin
    begin
        if rand_x > rand_y then
            for i=0 to SampleDataRowCount-1 do
                if rand_x_coordinate[i] ≤ coordinate1 and
                rand_y_coordinate[i] ≤ coordinate2 then
                    Sampledataset1[j] = i;
                    count1++;
                else
                    Sampledataset2[k] = i;
                    count2++;
                end
            end
        else
            for i=0 to SampleDataRowCount-1 do
                if rand_x_coordinate[i] ≤ coordinate1 and
                rand_y_coordinate[i] ≤ coordinate2 then
                    Sampledataset1[j] = i;
                    count1++;
                else
                    Sampledataset2[k] = i;
                    count2++;
                end
            end
        end
        /* If count1 > count2, then Sampledataset1 will be
           the training data set and Sampledataset2 will be
           the test data set.                              */
    end
end
```

Fig. 6. Training and Test data set Division Algorithm.

IX. CONCLUSION AND FUTURE WORK

Machine learning algorithms automatically extract knowledge from machine readable information. Unfortunately, their success is usually dependant on the quality of the data that they operate on. If the data is inadequate, or contains extraneous and irrelevant information, machine learning algorithms may produce less accurate and less understandable results. Thus, data pre-processing is an important step in the machine learning process. The pre-processing step is necessary to resolve several types of problems include noisy data, redundancy data, missing data values, etc. All the inductive learning algorithms rely heavily on the product of this stage, which is the final training set. By selecting relevant instances, experts can usually remove irrelevant ones as well as noise and/or redundant data. The high quality data will lead to high quality results and reduced costs for data mining. In addition, when a data set is too huge, it may not be possible to run a machine learning algorithm. In this case, instance selection reduces data and enables a machine learning algorithm to function and work effectively with huge data. Thus, this paper presented the most well known algorithms for each step of data pre-processing so that one achieves the best performance for their data set.

X. REFERENCES

[1]. C. M. Teng. Correcting noisy data. In Proc. 16th International Conf. on Machine Learning, pages 239–248. San Francisco, 1999.

[2]. Lakshminarayan K., S. Harp & T. Samad, Imputation of Missing Data in Industrial Databases, Applied Intelligence 11, 259–275, 1999.

[3]. H. Liu, F. Hussain, C. Lim, M. Dash. Discretization: An Enabling Technique. Data Mining and Knowledge Discovery 6:4, 393-423, 2002.

[4]. M. Boulle. Khiops: A Statistical Discretization Method of Continuous Attributes. Machine Learning 55:1, 53-69, 2009.

[5]. Yang J, Honavar V. Feature subset selection using a genetic algorithm. IEEE Int Systems and their Applications; 13(2): 44–49, 1999.

[6]. Somol, P., Pudil, P., Novovicova, J., Paclik, P.. Adaptive floating search methods in feature selection. Pattern Recognition Lett. 20 (11/13), 1157–1163, 1999.

[7]. Hu, Y.-J., & Kibler, D., Generation of attributes for learning algorithms. Proc. 13th International Conference on Machine Learning., 1996.