CORRELATION -BASED OUTLIER DETECTION

Kumari Puja
PG Student
Information Technology
Institute of Information
Technology and Management
(IITM)
Delhi ,India

Priyanka
PG Student
Information Technology
Institute of Information
Technology and Management
(IITM)
Delhi ,India

Ms. Ruby Dahiya
Associate Professor
Information Technology
Institute of information
Technology and Management
(IITM)
Delhi ,India

ABSTRACT

Outlier detection with correlation is an Integral part of calculating various outlier detection. In this paper, we propose many outlier detection methods which help to detect outlier in an easy manner.
In addition each method should have some advantage and disadvantage. To remove outlier detection we have used many practical approach and comparing methods with different approach. The original outlier detection methods are arbitrary but now, principled and systematic techniques are used for outlier detection.

Index Terms: correlation, outlier, Methods of outlier, Algorithms, Experimental, comparison of efficiency of algorithms.

I. INTRODUCTION

An outlier is a data point. It is significantly different from the remaining data. According to Hawkins formally defined [205] the concept of an outlier as follows: [1]

"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

Outliers are also referred to abnormalities, discordant, deviants, or anomalies in the data mining and statistics. The data is created by one or more generating processes, which could either reflect activity in the system or observations collected about entities. When the generating process behaves in an unusual way, its results in the creation of outliers. Therefore, an outlier often contains useful information about abnormal characteristics of the systems and entities, which affect the data generation process.[2] The recognition of such unusual characteristics provides useful application-specific insights.

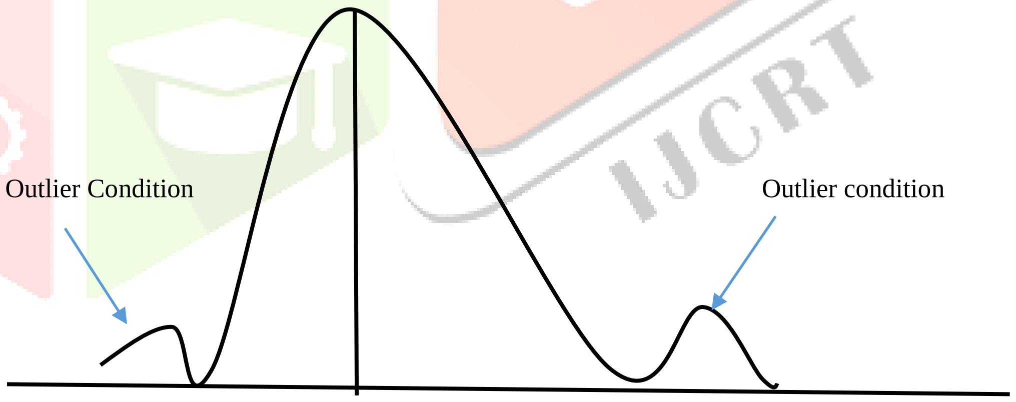


Fig1- shows outlier condition

Outliers are an extreme condition that deviates from other observations on data, they may indicate a Variation in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.
1.1Types of outlier

There are two types of outlier
1). Univariate
Univariate is define as distribution of value in single feature space.
2).Multivariate
Multivariate is define as distribution of value in N-dimensional feature space.

N-dimensional feature space is very difficult to measure for the human brain. That is why we need some to train a model.

An outlier can be in different form, it depends on the environment of data. According to data environment outliers are is three types:
1) Point outliers

2) Contextual outliers
3) Collective outliers

1) Point outliers - Point outliers is a single data point that data is different from other data set.
Example in data set A = {203,209,215,201,204,206,599}. '599' is a point dataset because 599 is different from the other data set all data's are lies in the range of 200 -220 range but 599 lies beyond the range.

2) Contextual outliers: A contextual outlier is also called conditional outliers. A data point deliberates a contextual outlier the value significantly differs from the rest of the data points in the same context. Note that same value is may not be considered an outlier if it occurs in a different context. Example- Is 10C in Vancouver an outlier. Solution – it depends on the session that is winter session or summer session.

3) Collective outliers: A subset of data points within the data set is considered anomalous behaviour is called the collective outliers. A data point behaviour is anomalous in that data point is not considered collective outliers. Example – Intrusion detection when a number of computers keep sending denial- of – service packages to each other.

1.2 Outlier Detection

Outlier detection is the process of detecting outliers from the given set of data. There is no standardize and rigid mathematical method for determining the outliers because it is depending upon the set of data or data population. Its determination and detection become subjective. There are model-based methods for detecting the outliers they assumed that all data are taken in a normal form distribution and identify observations or points, which are based on the mean or standard deviation, an outlier.

1.3 correlation

Correlation is a statistical technique that shows how a pair of variables is related. In finance and investment industries, correlation is a statistic that measures the degree to which two Securities move in relation to each other. Correlation is used in advance portfolio management. The correlation is one of the most common and most useful statistics. A correlation describes the degree of relation between the variable.
The equation of the correlation is:

$$r = \frac{N \Sigma x y - (\Sigma x)(\Sigma y)}{([N\Sigma x^2]-(\Sigma x)^2][N\Sigma y^2-(\Sigma y)^2])^{1/2}} \qquad (1)$$

II. Outliers Detection techniques:

2.1. Z-Score:

The z-score or standard score of an observation is a metric that indicates how many standard deviations a data point is from the sample's mean, assuming a Gaussian distribution. This makes z-score a parametric method. Very frequently data points are not to described by a Gaussian distribution, this problem can be solved by applying transformations to data i.e. scaling it. [2]
After making the appropriate transformations to the selected feature space of the dataset, the z-score of any data point can be calculated with the following expression:

$$\mathbf{Z= \frac{(x - \mu)}{\sigma}} \qquad (2)$$

When computing the z-score for each sample on the data set a threshold must be specified. Some good 'thumb-rule' thresholds can be: 2.5, 3, 3.5 or more standard deviations.

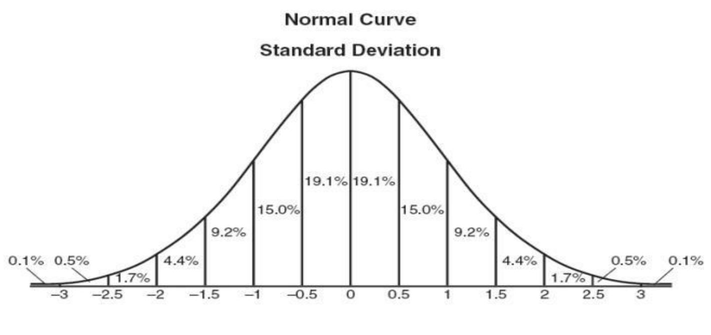


Fig2: curve shows the standard deviation of normal curve [2]

By 'tagging' or removing the data points that lay beyond a given threshold we are classifying data into outliers and not outliers

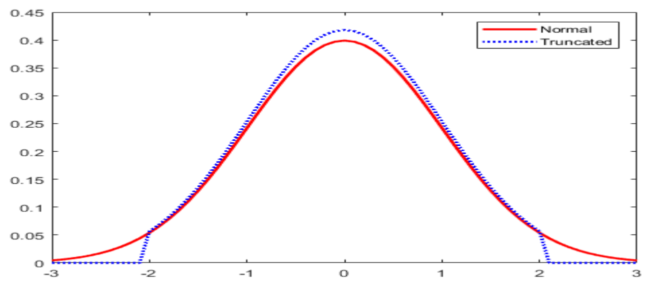


Fig3: Relation between normal curve and Truncated curve [2]

Z-score is a simple, yet powerful method to get rid of outliers in data if you are dealing with parametric distributions in a low dimensional feature space

Z-score Pros:

- It is a very effective method if you can describe the values in the feature space with a Gaussian distribution. (Parametric)
- The implementation is very easy using pandas and scipy stats libraries.

Z-Score Cons:

- It is only convenient to use in a low dimensional feature space, in a small to the medium-sized dataset.
- Is not recommended when distributions cannot be assumed to be parametric

2.2 DBSCAN:

A natural way to group together hosts that are behaving similarly is to use a clustering algorithm. We use DBSCAN, a popular Density-based clustering algorithm, for this purpose. DBSCAN works by greedily agglomerating points that are close to each other. Clusters with few points in they are considered outliers. [2]

Traditionally DBSCAN takes the following description as under below:

1). A parameter Ɛ that specifies a distance between two points is considered to close.

2). the minimum number of points that have to be within a points Ɛ- radius before that point can start agglomerating. The image shows. There are two clusters. The large points had enough close neighbours to agglomerate those points, while the small large point. The points which are blacks are an outlier.

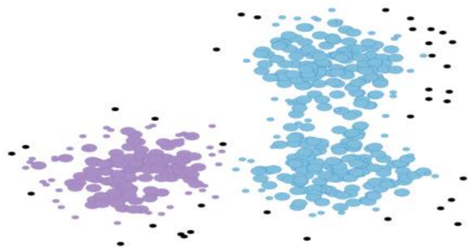


Fig4: shows the clustering of outlier's block of data [2]

DBSCAN Pros:

- It is a super effective method when the distribution of values in the feature space cannot be assumed.
- An If outlier is multidimensional in this situation DBSCAN is very effective.
- Sci-kit lean's implementation is easy to use and the documentation is superb.
- Visualizing the results in DBSCAN is very easy and the method itself is very intuitive.

DBSCAN Cons:

- The values in the DBSCAN feature space need to be scaled accordingly.
- Selecting the optimal parameters eps, MinPts and metric can be difficult since it is very sensitive to any of the three prams.
- DBSCAN unsupervised model and needs to be re-calibrated each time a new batch of data is analysed.
- DBSCAN can predict once calibrated but is strongly not recommended.

2.3 High Dimensional outlier Detection:

In real-world data set are very high dimensional. In some observation real data set may contain hundreds or thousands of dimensions. With increasing dimensionally, many of the traditional outlier detection methods do not work very effectively. That was why the concept of dimensionality arises. In high dimensional space, the data becomes sparse, and the true outliers become masked by the noise effect of multiple irrelevant dimension when analysed in full dimensional.[3]

In high dimensional following steps are required:

1).The first step it is finds the high contrast subspace using the comparison of marginal pdf and conditional pdf for each subspace.

2). Next, it calculates outlier score for each point based on each of high contrast subspaces

3). finally, it calculates the average of scores generated from the previous step.

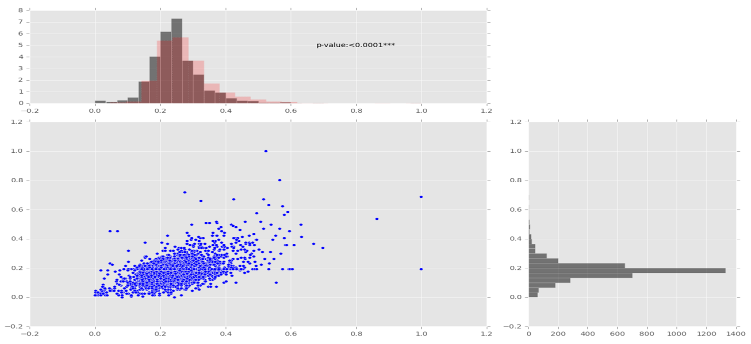
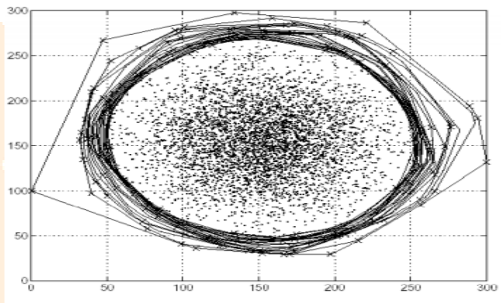


Fig-5 High Contrast Subspace [3]

III CORRELATION –BASED DETECTION TECHNIQUES DE

3.1 Depth-based Approaches

Depth-based approaches are a search for outliers at the border the data space but independent of statistical distribution. It organizes data object in convex hull layers. The outliers are objects on outer layers. [6]

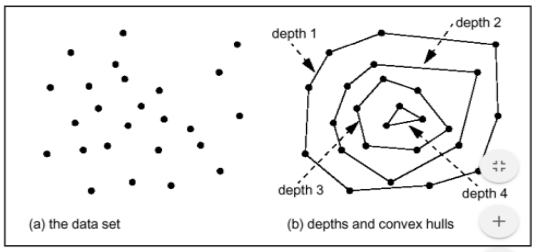


Picture taken from [Johnson et al. 1998]

Fig6: shows the outer layer of the outlier

Outlier are located at the border of the data space. Normal objects are in the centre of the data space. Points on the convex hull of the full data space have depth=1, 2, 3…….

Points having a depth <= k are reported as outliers



Picture taken from [Preparata and Shamos 1988]

Fig7: shows the data set and depth of the convex hulls

3.2. Distance-based Approaches

Distance-based outlier analysis is one of the most used and widely accepted techniques that used in data mining and machine learning. It completely depends on the concept of the local neighbourhood of data points. This concept is also termed as Nearest Neighbour analysis and it can also be applied for different purposes such as classification, clustering and most importantly outlier analysis.[7] Distance-based Approaches has judged a point based on the distance to its neighbours and several variants proposed.

The normal data objects have a dense neighbourhood. The outliers are far apart from their neighbours, i.e. have a less dense neighbourhood.

In distance-based outlier, two things are required Ɛ and π. Where Ɛ is the given radius and π is the percentage. A point p is considered an outlier if at most π percent of all other points have a distance top less than Ɛ.

$$\text{OutlierSet}(\varepsilon,\pi) = \{p \mid \frac{\text{Card}(\{q \in DB \mid dist(p,q)<\varepsilon\})}{\text{Card}(DB)} <= \pi \qquad (3)$$

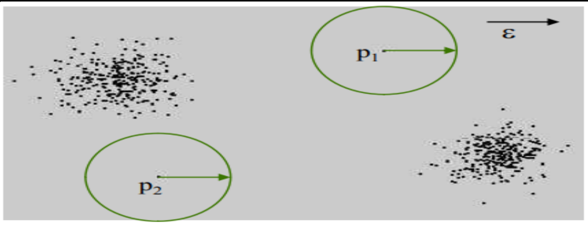


Fig8: shows distance- based Approaches

3.3 .Density –based Approaches

The density-based approach is a classic outlier finds the density allocation of the details and identify outliers as those present in the low-density region. Partitioning methods are designed to find spherically shaped clusters but it is difficult to find clusters of arbitrary shape such as ‘S’ and oval. Density-based methods overcome this limitation.

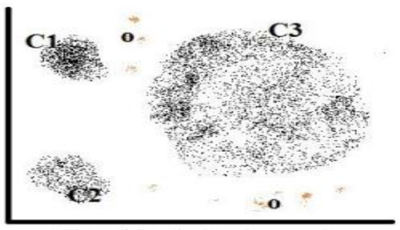


Fig9: density based approaches

This method is used compares the density around a point with its local neighbour’s densities. The relative density of a point compared to its neighbours is computed as an outlier score. Density based outlier detection method uses density distribution of data points within data set. Brewing et al. [7] allocate a local outlier factor (LOF) to every point based on the neighbouring density of its environs. LOF value of an object is based on the average of the ratios of the local reachability density of the area around the object and the local reachability densities of its neighbours. The size of the neighbourhood of the object is determined by the area containing a user-supplied minimum number of points (MinPts). DBSCAN, OPTICS, DENCLUE methods are used for density based Approach. In this figure C1, C2, C3 stands for clusters and O for outliers.

IV. Experimental Setup

In Experimental Setup compare three algorithm of datamining. These algorithm is statistical datamining algorithm. For getting the experimental result using weka software version 3.9.2. Weka version 3 is datamining software in java. It is also used in machine learning process. [4] Weka is data mining software that uses a collection of machine learning algorithms. These algorithms can be applied directly to the data or called from the Java code.

Weka is a collection of tools for:

- Regression
- Clustering
- Association
- Data pre-processing
- Classification
- Visualisation [5]

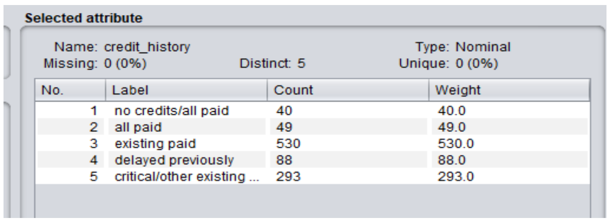
The purpose of the paper is comparison of different Algorithm and find out which algorithm is more efficient for data mining. In this paper comparison between three algorithm:-

(1) meta.AdaBoostM1
(2) meta.Bagging
(3) Random Forest

These three algorithm is applied on the credit data. In credit data it means a person have how many saving data, how many transection was occurred, how many available data, etc. Find out details of credit data using weka tool.

First of all downloaded the weka software. Go to the explorer and select the data name and go to attribute block and click on credit history then we see the selected attribute the history of data are shown in tabular format.

Table 1: show the credit history



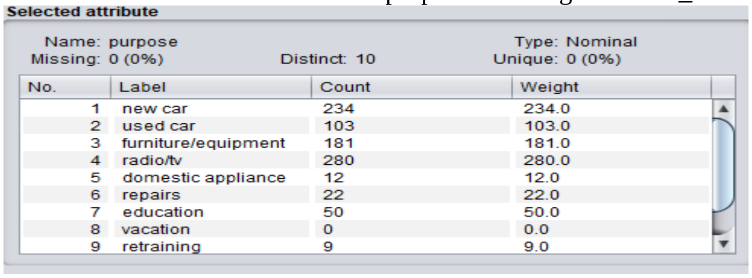
Selected attribute

Name: credit_history Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	no credits/all paid	40	40.0
2	all paid	49	49.0
3	existing paid	530	530.0
4	delayed previously	88	88.0
5	critical/other existing ...	293	293.0

Also find out what is the purpose for use and also find the percentile of data, find the mean and standard deviation of the data and find out other many details related to the credit data.

Table 2: show the details of which purpose for using the credit_data

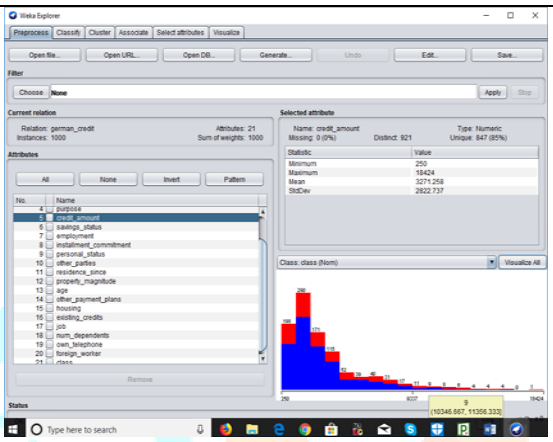


Selected attribute

Name: purpose Type: Nominal
Missing: 0 (0%) Distinct: 10 Unique: 0 (0%)

No.	Label	Count	Weight
1	new car	234	234.0
2	used car	103	103.0
3	furniture/equipment	181	181.0
4	radio/tv	280	280.0
5	domestic appliance	12	12.0
6	repairs	22	22.0
7	education	50	50.0
8	vacation	0	0.0
9	retraining	9	9.0

4.1 Visualization of Data

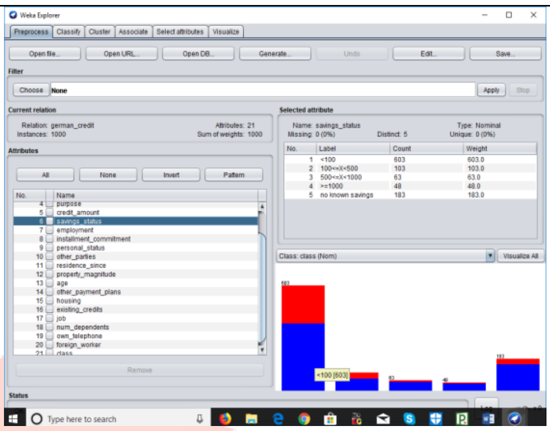


Relation: german_credit Attributes: 21
Instances: 1000 Sum of weights: 1000

Name: credit_amount Type: Numeric
Missing: 0 (0%) Distinct: 921 Unique: 847 (85%)

Statistic	Value
Minimum	250
Maximum	18424
Mean	3271.258
StdDev	2822.737

Fig10: Graph of Credit _amount



Relation: german_credit Attributes: 21
Instances: 1000 Sum of weights: 1000

Name: savings_status Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

No.	Label	Count	Weight
1	<100	603	603.0
2	100<=X<500	103	103.0
3	500<=X<1000	63	63.0
4	>=1000	48	48.0
5	no known savings	183	183.0

Fig11: Graph of saving_data

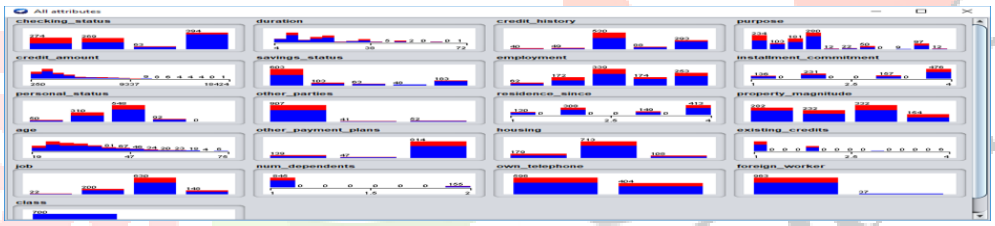
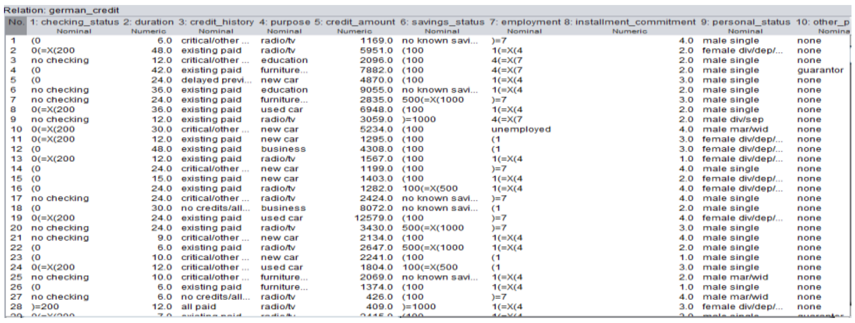


Fig12: Visualize all data

All data are store in column and row format i.e. data are store in table. We click on edit option then all data are shown in tabular form.

Table 3 shows all data record



Relation: german_credit

No.	1: checking_status Nominal	2: duration Numeric	3: credit_history Nominal	4: purpose Nominal	5: credit_amount Numeric	6: savings_status Nominal	7: employment Nominal	8: installment_commitment Numeric	9: personal_status Nominal	10: other_p Nomina
1	(0	6.0	critical/other ...	radio/tv	1169.0	no known savi...	)=7	4.0	male single	none
2	0(=X(200	48.0	existing paid	radio/tv	5951.0	(100	1(=X(4	2.0	female div/dep/...	none
3	no checking	12.0	critical/other ...	education	2096.0	(100	4(=X(7	2.0	male single	none
4	(0	42.0	existing paid	furniture...	7882.0	(100	4(=X(7	2.0	male single	guarantor
5	(0	24.0	delayed previ...	new car	4870.0	(100	1(=X(4	3.0	male single	none
6	no checking	36.0	existing paid	education	9055.0	no known savi...	1(=X(4	2.0	male single	none
7	no checking	24.0	existing paid	furniture...	2835.0	500(=X(1000	)=7	3.0	male single	none
8	0(=X(200	36.0	existing paid	used car	6948.0	(100	1(=X(4	2.0	male single	none
9	no checking	12.0	existing paid	radio/tv	3059.0	)=1000	4(=X(7	2.0	male div/sep	none
10	0(=X(200	30.0	critical/other ...	new car	5234.0	(100	unemployed	4.0	male mar/wid	none
11	0(=X(200	12.0	existing paid	new car	1295.0	(100	(1	3.0	female div/dep/...	none
12	(0	48.0	existing paid	business	4308.0	(100	(1	3.0	female div/dep/...	none
13	0(=X(200	12.0	existing paid	radio/tv	1567.0	(100	1(=X(4	1.0	female div/dep/...	none
14	(0	24.0	critical/other ...	new car	1199.0	(100	)=7	4.0	male single	none
15	(0	15.0	existing paid	new car	1403.0	(100	1(=X(4	2.0	female div/dep/...	none
16	(0	24.0	existing paid	radio/tv	1282.0	100(=X(500	1(=X(4	4.0	female div/dep/...	none
17	no checking	24.0	critical/other ...	radio/tv	2424.0	no known savi...	)=7	4.0	male single	none
18	(0	30.0	no credits/all...	business	8072.0	no known savi...	(1	2.0	male single	none
19	0(=X(200	24.0	existing paid	used car	12579.0	(100	)=7	4.0	female div/dep/...	none
20	no checking	24.0	existing paid	radio/tv	3430.0	500(=X(1000	)=7	3.0	male single	none
21	no checking	9.0	critical/other ...	new car	2134.0	(100	1(=X(4	4.0	male single	none
22	(0	6.0	existing paid	radio/tv	2647.0	500(=X(1000	1(=X(4	2.0	male single	none
23	(0	10.0	critical/other ...	new car	2241.0	(100	(1	1.0	male single	none
24	0(=X(200	12.0	critical/other ...	used car	1804.0	100(=X(500	(1	3.0	male single	none
25	no checking	10.0	critical/other ...	furniture...	2069.0	no known savi...	1(=X(4	2.0	male mar/wid	none
26	(0	6.0	existing paid	furniture...	1374.0	(100	1(=X(4	1.0	male single	none
27	no checking	6.0	no credits/all...	radio/tv	426.0	(100	)=7	4.0	male mar/wid	none
28	)=200	12.0	all paid	radio/tv	409.0	)=1000	1(=X(4	3.0	female div/dep/...	none

4.2 Algorithms

4.2.1 meta.AdaBoostM1

AdaBoost is the first recognition of boosting algorithms in 1996 by Freund and Schapire. This boosting algorithm is designed for only binary classification and its base classifier is decision stamp. Remember that underlying classifier in boosting algorithm is called 'base classifier'. This Underlying base classifier for AdaBoost is decision Stamp which is a decision tree with one level. A major goal of AdaBoost algorithm is to combine these really weak one level decision Stamp classifiers and generate a strong classifier at the end.

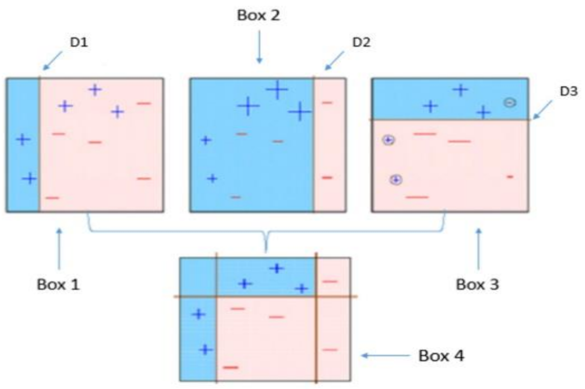


Fig13: Shows the Bossing

Remember that boosting algorithms give more weights to instances that are misclassified by previous model and try to correct predictions on that instances. Giving more weights to some instances means that applied misclassification penalty in the training loss are much higher for that instances. In above figure at the right, each time a decision stamp D1, D2 and D3 is added to the system, it focuses on misclassified instances at the previous stage. These instances are shown in the bigger figure and each decision stamp line is drawn to correct predictions of these instances in current round of boosting algorithm. At the end, all decision stamps (Box1, Box2 and Box3) are collected and final model is formed. (Box4).

The Pseudo code of the algorithm and explain the technical details:

Algorithm

```
1. Initialize the observation Weights $w_i$=1/N, i=1, 2, 3………..N.
2. For m=1 to M repeat Steps 2.1 to 2.4:
    2.1. Fit a classifier $C_m(x)$ to the training data using weights $w_i$.
    2.2). Compute Weight error of newest tree
        $Err_m = \frac{\sum_{i=1}^{N} w_I I(y_i \neq C_m(X_i))}{\sum_{i=1}^{N} W_i}$
    2.3). Compute $\alpha=\log[(1-err_m)/err_m]$.
    2.4). Update Weight for =1, 2,……………… ,N:
        $w_i \longleftarrow W_i . \exp[\alpha_m . I(y_i \neq C_m(x_i))]$
        and renormalize to $w_i$ to sum to 1.
3. Output $C(x) = \mathrm{Sign}[\sum_{m=1}^{M} \alpha_m C_m(x)]$.
```

In the step 1 of the pseudo code, initially all the weights of instances are set equal. This weight is equal to 1/N where N is number of instances in our training data set.

All learning is done in the step 2 of pseudo code. The loop generates each tree stage-wise by using weighted instances from previous stage. In step 2.1, during fitting a classifier to weighted training data, a decision stamp divides data into two class, focusing on previously misclassified instances. In step 2.2, weighted error of the newest tree is calculated over misclassified instances. It is because of function 'I' in the formula. It gives 1 for misclassified instances and returns 0 for correctly classified ones.

In step 2.3, we compute another weight value but this weight is not for instances, it is actually a weight of the newest tree in the last step formula which is for the combination of classifiers. It is calculated using weighted error from step 2.2. if classification error of a tree is high, then it gets lower weight. This is a more reasonable way of combining votes of multiple classifiers for the final result. Finally, since AdaBoost is an algorithm just designed for binary classification (1 or -1), sign of a weighted sum of classifiers' votes is calculated in step 3. Predictions are made by calculating weighted sum of weak classifiers.

2. Bagging

Bagging is the Extension of the Bootstrap algorithm. It is powerful Statistical method for estimating a quality from data sample. This is easiest to understand if the quantity is a descriptive statistic such as a mean or a standard deviation. Using the formula:

$$\text{Mean } (x) = 1/100 * \text{sum}(x). \quad (4)$$

Bagging is also called Bootstrap Aggregation (or bagging for short), is a simple and very powerful ensemble Method. Bagging is the application of the Bootstrap.

Bootstrap Aggregation is a general procedure that can be used to the reduce the variance for those algorithm that have high variance. An algorithm that has high variance are decision trees, like classification and regression trees (CART). Decision trees are sensitive to the specific data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree can be quite different and in turn the predictions can be quite different.

Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, like decision trees.

Let’s assume a sample dataset of 1000 instances (x) and using the CART algorithm. Bagging of the CART algorithm would work as follows.

Step1. Create many (e.g. 100) random sub-samples of our dataset with replacement.
Step2. Train a CART model on each sample.
Step3. Given a new dataset, calculate the average prediction from each model.

Bagging Algorithm

```
Input: S: Training set: T: Number of iterations;
          N: Bootstrap Size; I: weak learner
Output: Bagged classifier: H(x) = sign ($\sum_{t=0}^{T} h_t$(X)) where
          [-1, 1] are the induced classifiers
1. for t= 1 to T do
2. $S_t$ $\leftarrow$ RandomSampleRepacement(n,S)
3. $H_i$ $\leftarrow$ I($S_t$)
4. End for.
```

4.2.3 Random Forest

Random Forests are an improvement over bagged decision trees.

A problem with decision trees like CART is that they are greedy. They choose which variable to split on using a greedy algorithm that minimizes error. Even with Bagging, the decision trees have a lot of structural similarities and have high correlation in their assumptions.

Combining assumptions from multiple models in ensembles works better if the assumptions from the sub-models are uncorrelated or at best weakly correlated.

Random forest changes the algorithm for the way that the sub-trees are learned so that the resulting assumptions from all of the subtrees have less correlation.

It is a simple tweak. In CART, when selecting a split point, the learning algorithm is allowed to look through all variables and all variable values in order to select the most optimal split-point. The random forest algorithm changes this procedure so that the learning algorithm is limited to a random sample of features of which to search.

The number of features that can be searched at each split point (p) must be specified as a parameter to the algorithm. You can try different values and tune it using cross validation.

- For classification a good default is: p = sqrt(q)
- For regression a good default is: p = q/3

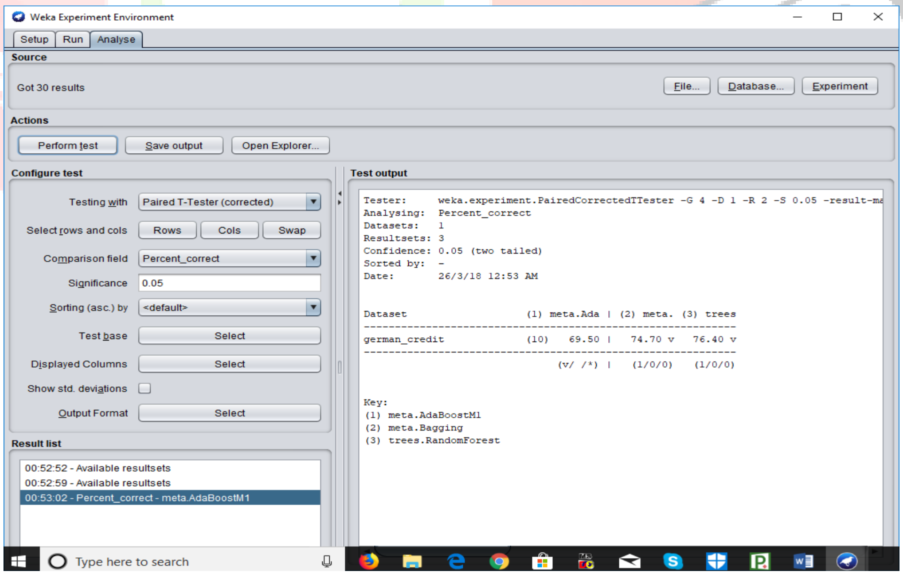
Where m is the number of randomly selected features that can be searched at a split point and q is the number of input variables. For example, if a dataset had 25 input variables for a classification problem ,then:

- p = sqrt(25)
- p = 5

Pseudo code of Random Forest:

```
1.  Randomly Select "K" features from total "M" features.
1.1 where K<<M
2.  Among the "K" features, calculated the node "d" using the best split point.
3.  Split the node into children nodes using the best Split.
4.  Repeat 1 to 3 steps until "I" number of nodes has been reached.
5.  Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.
```

V. Result



Got 30 results

Testing with: Paired T-Tester (corrected)
Comparison field: Percent_correct
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations: ☐

00:52:52 - Available resultsets
00:52:59 - Available resultsets
00:53:02 - Percent_correct - meta.AdaBoostM1

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4 -D 1 -R 2 -S 0.05 -result-ma
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       26/3/18 12:53 AM


Dataset                   (1) meta.Ada | (2) meta. (3) trees
-------------------------------------------------------------
german_credit             (10)   69.50 |   74.70 v   76.40 v
-------------------------------------------------------------
                               (v/ /*) |   (1/0/0)   (1/0/0)


Key:
(1) meta.AdaBoostM1
(2) meta.Bagging
(3) trees.RandomForest
```

Fig14: comparison of three algorithm

Comparison between above three algorithm bagging algorithm is best other AdaBoostM1 and Random Tree. Fig-6 shows that accuracy of Ada is 69.50 |, accuracy of Bagging 74.70v and accuracy of Random Tree is 76.40v. The meaning on V is significant good '*' means significant bad '|' blank. Random Tree is 76.40 good as compare to the AdaBoostM1. Similarly, Bagging is 74.70 is good as compare to the Random Tree. This show statically significant.

VI. CONCLUSION

In this paper, the state of the art on ensemble methodologies to deal with class credit data problem has been reviewed. This issue hinders the performance of standard classifier learning algorithms that assume relatively credit data class distributions, and classic ensemble learning algorithms are not an exception. In recent years, several methodologies integrating solutions to enhance the induced classifiers in the presence of class credit data by the usage of ensemble learning algorithms have been presented. However, there was a lack of framework where each one of them could be classified; for this reason, a taxonomy where they can be placed has been presented. We divided these methods into four families depending on their base ensemble learning algorithm and the way in which they address the class credit problem.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their valuable comments and suggestions that contributed to the improvement of this work.

REFERENCES

[1] *Hawkins (1980)* https://www.investopedia.com/terms/c/correlation.asp
[2] [2] https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561
[3] https://pdfs.semanticscholar.org/29cf/2221afbf02e81c87c5f91d7b2e4e4dfae545.pdf
[4] https://www.cs.waikato.ac.nz/ml/weka/
[5] http://opensourceforu.com/2017/01/an-introduction-to-weka/
[6] http://imada.sdu.dk/~zimek/publications/KDD2010/kdd10-outlier-tutorial.pdf
[7] https://pdfs.semanticscholar.org/526d/2be3a950b9d60edf87d9cbb6ffcc8f5d4a08.pdf
[8] https://www.datasciencecentral.com/profiles/blogs/introduction-to-outlier-detection-methods
[9] file:///C:/Users/LENOVO/Downloads/A_Review_on_Ensembles_for_the_Class_Imba.pdf
[10] https://www.anodot.com/blog/quick-guide-different-types-outliers/
[11] https://www.siam.org/meetings/sdm10/tutorial3.pdf
[12] https://machinelearningmastery.com/how-to-identify-outliers-in-your-data/
[13] https://www.kdnuggets.com/2017/01/3-methods-deal-outliers.html
[14] https://machinelearningmastery.com/how-to-identify-outliers-in-your-data/
[15] https://www.datasciencecentral.com/profiles/blogs/introduction-to-outlier-detection-methods
[16] file:///C:/Users/LENOVO/Downloads/A_Review_on_Ensembles_for_the_Class_Imba%20.pdf
[17] file:///C:/Users/LENOVO/Downloads/Scopus.pdf
[18] file:///C:/Users/LENOVO/Downloads/petrovskiy_pcs03%20.pdf
[19] file:///C:/Users/LENOVO/Downloads/Outlier_Detection_for_High_Dimensional_Data%20.pdf
[20] file:///C:/Users/LENOVO/Downloads/hodgevj4%20.pdf
[21] file:///C:/Users/LENOVO/Downloads/V15-01-02-Zimek%20.pdf
[22] file:///C:/Users/LENOVO/Downloads/V15-01-02-Zimek%20.pdf
[23] file:///C:/Users/LENOVO/Downloads/1007.1268%20.pdf
[24] file:///C:/Users/LENOVO/Downloads/TKDEauthorVersion%20.pdf
[25] Kamber and Han, "Data Mining Concepts and Techniques", Hartcourt India P. Ltd.,2001.
[26] G. K. Gupta, "Introduction to Data Mining with Case Studies", PHI, 2006.
[27] A. B. M. Shawkat Ali, Saleh A. Wasimi, "Data Mining Methods and Techniques",Cengage Learning, 2009
[28] Pang - Ning, Michael- Steinbach, "Introduction to Data Mining", Pearson, 4th Ed.,2009.

IJCRT