Comparing the Efficiency of Different Normalization Methods using C4.5 Algorithm

Khin Nwe Oo
Computer University (Monywa)
khinnweoo5@gmail.com

Abstract

This paper presents three different types of normalization methods. Normalization is particularly useful for classification algorithm. Each normalization method is tested against the C4.5 algorithm methodology using the data sets with accuracy, tree growing time and number of leaf nodes. In this paper, min-max normalization, z-score normalization and normalization by decimal scaling normalization are compared among large amount of data set. From this comparison, the best normalization method is chosen.

1. Introduction

Induction decision tree C4.5 algorithm implements a scheme for top-down induction of decision trees using divide and conquer approach. The input to the algorithm is a tabular set of training objects, each characterized by a fixed number of attributes and a class designation. C4.5 algorithm is one of the most common techniques used in the field of data mining and knowledge discovery [5].

Data usually collected from multiple resources and stored in data warehouse. Resources may include multiple databases, data cubes, or flat files. Different issues could arise during integration of data for mining and discovery. These issues include scheme integration and redundancy. So, data integration must be done carefully to avoid redundancy and inconsistency that in turn improve the accuracy and speed up the mining process. The careful data integration is now acceptable but it needs to be transformed into forms suitable for mining.

Data transformation involves smoothing, generalization of the data, attribute construction and normalization. Data mining seeks to discover unrecognized associations between data items in an existing database. It is the process of extracting valid, previously unseen or unknown, comprehensible information from large databases. The growth of the size of data and number of existing databases exceeds the ability of humans to analyze this data, which creates both a need and an opportunity to extract knowledge from databases. Data transformation such as normalization may improve the accuracy and efficiency of mining algorithms involving neural networks, nearest neighbor and clustering classifiers. Such methods provide better results if the data to be analyzed have been normalized, that is, scaled to specific ranges such as (0.0, 1.0) an attribute is normalized by scaling its values so that they fall within a small-specified range, such as (0.0 to 1.0).

Normalization is particularly useful for classification algorithms involving neural networks, or distance measurements such as nearest neighbor classification and clustering. If using the neural network back propagation algorithm for classification mining, normalizing the input values for each attribute measured in the training samples will help speed up the learning phase. For distanced-based methods, normalization helps prevent attributes with initially large ranges from outweighing attributes with initially smaller ranges. There are many methods for data normalization include min-max normalization, z-score normalization and normalization by decimal scaling [1].

If the data set selected for analysis is huge, it is sure to slow down the mining process. The careful data integration is now acceptable but it needs to be transformed into forms suitable for mining.

For this reason, in this thesis, the system is designed to choose the best normalization methods by analyzing this three methods with C4.5 algorithm.

2. Related Work

Data transformation involves smoothing, generalization of the data, attribute construction and normalization. Data mining seeks to discover unrecognized associations between data items in an existing database and Step in the Process of Data Mining.[1]

Normalization methods that improve the multimodal system performance. They present three well-known normalization methods, and a new method, which we call adaptive normalization. They denote a raw matching score as s from the set S of all scores for that matcher, and the corresponding normalized score as n. Min-Max (MM): This method maps the raw scores to the [0,1] range. [3]

This is emphasized on different types of normalization. Each of which was tested against the ID3 methodology using the HSV dataset. Comparisons between different learning methods were accomplished as they were applied to each normalization method. A new matrix is used to check for the best normalization method based on the factors and their priorities. [2]

3. Theory Background

Databases are highly susceptible to noisy, missing and inconsistent data due to their typical huge size, often several gigabyte or more. “how can the data be preprocessed in order to help improve the quality of the data and consequently, of the mining result?”. The data can be preprocessed so as to improve the efficiency and ease of the mining process.

There are a number of data preprocessing techniques. Data cleaning can be applied to remove and correct inconsistencies in the data. Data integration merges data from multiple sources into a coherence data store, such as a data warehouse or a data cube. Data transformation, such as normalization, may be applied. For example normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements. Data reduction can reduce the data size by aggregation, elimination redundant features, or clustering, for instance. This data preprocessing techniques, when applied to mining, can substantially improve the overall quality to the pattern mined and/or the time require for actual mining.

These methods are organized into the following categories: data cleaning, data integration, data transformation, data reduction [1].

3.1 Normalization Methods

Normalization methods are

1. Min-max normalization
2. Z-score normalization
3. Normalization by decimal scaling

Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are minimum and maximum values of an attribute A. Min-max normalization maps a value v of A to v’ in the range [new_min_A, new_max_A] by computing

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

For example, suppose that minimum and maximum value are $12,000 and $98,000. We would like the range [0.0,1,0]. By min_max normalization a value is 73,600.

v = 73,600
max = 12,000
max = 98,000
new_max = 0
new_min = 1

$$v' = \frac{73{,}000 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0.0) + 0$$
$$= 0.716$$

In z-score normalization (or zero-mean normalization), the values for an attribute A are normalized based on the mean and standard deviation of A. A value v of A is normalized to v’ by computing

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

where $\bar{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalization.

Example: Suppose that the mean and standard deviation of the values for the attribute income are $54,000 and $ 16,000 respectively. With z-score normalization, a value of 73,600 for income is transformed to (73,600-54,000)/16,000=1.225.

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the minmax absolute value of A. A value v of A is normalized to v’ by computing

$$v' = \frac{v}{10^j}$$

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

For example: suppose the range of attribute *X* is -500 to 45. The maximum absolute value of *X* is 500. To normalize by decimal scaling we will divide each value by 1,000 (j = 3). In this case, -500 becomes -0.5 while 45 will become 0.045

3.2 C4.5 Algorithm

In this thesis, C4.5 algorithm is used to calculate the effect of normalization methods. C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, \ldots$ of already classified samples. Each sample $s_i = x_1, x_2, \ldots$ is a vector where $x_1, x_2, \ldots$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1, c_2, \ldots$ where $c_1, c_2, \ldots$ represent the class to which each sample belongs.

At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists [4].

3.3 Estimating Classifier Accuracy

Hold out method estimates the classifier performances in this system. The given data are randomly partitioned into two independent sets, a training set and a test set. Typically, two third of the data are allocated to the training set, and the remaining one third is allocated to the test set.

The training set is used to derive the classifier, whose accuracy is estimated with the test set. The estimate is pessimistic since only a portion of initial data is used to derive the classifier. Random sub sampling is a variation of the holdout method in which the holdout is repeated k time.

The training set is used to derive the classifier, whose accuracy is estimated with the test set. The estimate is pessimistic since only a portion of initial data is used to derive the classifier. Random sub sampling is a variation of the holdout method in which the holdout is repeated k time.

The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration [1].

4. Experimental Result

In this paper, three different normalization methods are considered: min-max normalization, z-score normalization, decimal scaling normalization and five datasets are used. These datasets are taken from UCI repository. [4] They are

1. Blood dataset with number of instances (748) and 5 number of attributes.
2. Iris dataset with number of instances (150) and 5 number of attributes.
3. Hayes-Roth dataset with number of instances (132) and 6 number of attributes.
4. Lenses dataset with number of instances (24) and 5 number of attributes.
5. Glass dataset with number of instances (214) and 11 number of attributes.

4.1 By Using Blood Dataset

Table 1. Comparison of Three Normalization Methods Using Blood Dataset

Approach	Min-Max result	Z-score result	Decimal result
Accuracy (%)	63.45	71.48	80.72
Tree growing time	266	78	94
No: of leaf nodes	263	263	263

While using blood dataset, number of leaf nodes are all the same. When C4.5 algorithm perform on this data set, decimal scaling normalization is the highest accuracy(80.72%) and the next higher accuracy is z-score normalization with (63.45%) and min-max normalization is lowest accuracy.

4.2 By Using Iris Dataset

When C4.5 algorithm is applied in Iris dataset, the number of leaf node is 43 in all methods. Min-max normalization methods generated highest accuracy (100%), and the next higher accuracy is generated by decimal scaling normalization with

(82%) and z-score is lowest accuracy. And then tree growing time is all the same.

Table 2. Comparison of Three Normalization Methods Using Iris Dataset

Approach	Min-Max result	Z-score result	Decimal result
Accuracy (%)	96	82.0	72.0
Tree growing time	16	16	47
No: of leaf nodes	44	44	44

4.3 By Using Hayes-Roth Dataset

In Hayes-Roth dataset, C4.5 algorithm generates highest accuracy of min-max and decimal scaling normalization methods. Z-score normalization method is next higher accuracy. Min-max and z-score have the same tree growing time (16 milliseconds) and decimal scaling normalization has highest tree growing time (32 milliseconds).

Table 3. Comparison of Three Normalization Methods Using Hayes-Roth Dataset

Approach	Min-Max result	Z-score result	Decimal result
Accuracy (%)	64.15	50.94	54.71
Tree growing time	16	16	32
No: of leaf nodes	90	90	90

4.4 By Using Lenses Dataset

When the lenses dataset is used with C4.5 algorithm, min-max normalization method has the highest accuracy (100%) and the second higher accuracy of method is decimal scaling normalization. Z-score normalization has the lowest accuracy. And the tree growing time and number of leaf nodes are all the same.

Table 4. Comparison of Three Normalization Methods Using Blood Dataset

Approach	Min-Max result	Z-score result	Decimal result
Accuracy (%)	100.0	62.0	75.0
Tree growing time	0	0	0
No: of leaf nodes	7	7	7

4.5 By Using Glass Dataset

Table 5. Comparison of Three Normalization Methods Using Blood Dataset

Approach	Min-Max result	Z-score result	Decimal result
Accuracy (%)	87.32	36.61	78.87
Tree growing time	250	250	250
No: of leaf nodes	130	130	130

When glass dataset is used with C4.5 algorithm, min-max normalization has highest accuracy with (87.32%), and decimal scaling normalization is the next higher accuracy. And z-score is the lowest accuracy. Tree growing time and number of leaf nodes are all the same.

According above table, min-max normalization has highest accuracy than other methods. But this normalization method is lowest accuracy when using blood dataset because of these dataset of attribute. These dataset has duplicate data and these data doesn't have the same class.

The experimental results suggest in order to choose the min-max normalization as the best design for training data set. In all experiments, the min-max normalization data set has the highest priority. Although all data set has the same number of leaf node the min-max normalization data set has the lowest tree growing time, so it is faster than the other two methods of normalization.

5. System Flow Diagram

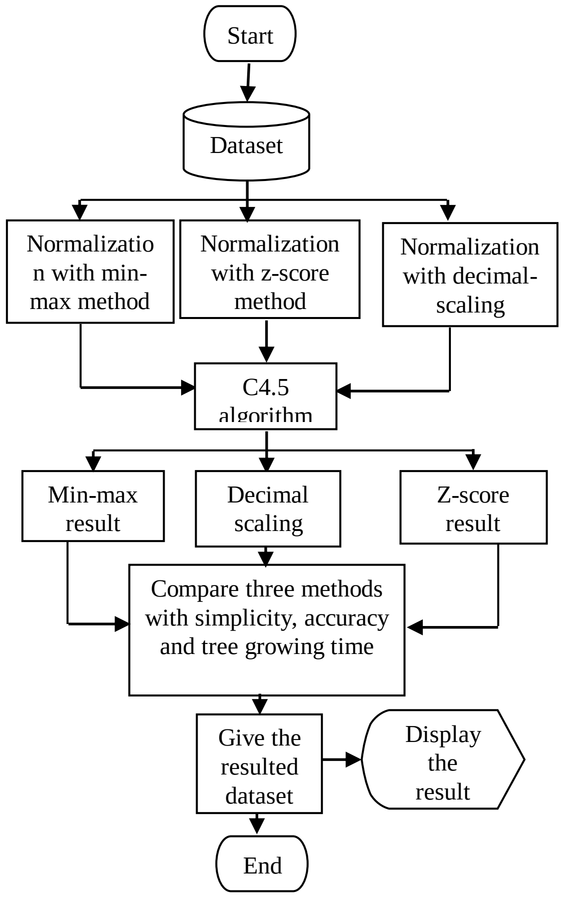

Figure 1. System Flow Diagram

Details of the system design are load data set from the database first. Normalize this dataset with min-max, z-score and decimal scaling normalization method. Analyze this three normalized results with C4.5 algorithm. And then compare these result from C4.5 algorithm with tree growing time, number of leaf nodes and accuracy and display the resulted dataset.

6. Conclusion

In this paper, three different normalization methods are presented. When applying data mining to real world, learning from data that falls within a large specific range in an evitable situation. Trying to normalize data is an obvious solution where data is scaled so as to fall within a small specific range This is designed to test the effect of different normalization methods on accuracy, simplicity and tree growing time factors. According to above experimental result, min-max normalization is one of the best of other two normalization methods in every dataset. But this method has lower accuracy than z-score and decimal scaling normalization methods when using blood dataset. Because this data set has same data of different class. If dataset is used actually not in order to be data mistaken, data must be collected correctly, must use real data and careful data integration. In this experiment, there are three datasets with 5 number of attributes, one dataset with 6 number of attributes and 11 number of attribute respectively. When the dataset with 5 number of attribute is used, min-max accuracy are 63.45%, 96% and 100%. Min-max accuracy is 64.45% when the dataset with 6 number of attribute is used. There is 87% min-max accuracy when the dataset with 11 number of attributes is used. So, min-max normalization does not depend on the number of attributes.

References

[1] J.Han, M.Kamber, Data Mining Concepts and Techniques, Morgan Kaufman Publishers, 2001.

[2] L.AIShalabi, Z.Shaaban and B.kasasbeh, Data Mining: A Preprocessing Engine.

[3] R.Snelick, U.Uludag, A.Mink, M.Idovia and A.Jain normalization and fusion methods that improve the multimodel system performance.

[4] S.Ruggieri, Efficient C4.5, Dipartimento di Inofrmatica, Universita di Pisa, Corso Italia 40, 56125 Pisa Italy.

[5] http://en.wikipedia.org/wiki/C4.5_algorithm

[6] http://www.ics.uci.edu/~mlearn/MLRepository.html, university of California