WEKA FOR REDUCING HIGH - DIMENSIONAL BIG TEXT DATA

Kotonko Lumanga Manga Tresor

Central South University Department of Computer science & Engineering, Address
City Changsha, State ZIP 410083/Zone, China
Kotonko1313@gmail.com

Professor Xu Dezhi

Department of Computer science & Engineering, Address
City Changsha, State ZIP 410083/Zone, China
WechatID: wxid_lclnqurzg3pj21

Abstract

In the current era, data usually has a high volume, variety, velocity, and veracity, these are known as 4 V's of Big Data. Social media is considered as one of the main causes of Big Data which get the 4 V's of Big Data beside that it has high dimensionality. To manipulate Big Data efficiently; its dimensionality should be decreased. Reducing dimensionality converts the data with high dimensionality into an expressive representation of data with lower dimensions.

This research work deals with efficient Dimension Reduction processes to reduce the original dimension aimed at improving the speed of data mining. Spam-WEKA dataset; which entails twitter user information. The modified J48 classifier is applied to reduce the dimension of the data thereby increasing the accuracy rate of data mining. The data mining tool WEKA is used as an API of MATLAB to generate the J48 classifiers. Experimental results indicated a significant improvement over the existing J48 algorithm.

Keywords: Dimension Reduction; J48; WEKA; MATLAB

1. Introduction

In the current era, data usually has a high volume, variety, velocity, and veracity, these are known as 4 V's of Big Data. Social media is considered as one of the main causes of Big Data which get the 4 V's of Big Data beside that it has high dimensionality. To manipulate Big Data efficiently; its dimensionality should be decreased. Reducing dimensionality converts the data with high dimensionality into an expressive representation of data with lower dimensions.

Reducing high dimensional text is really hard, problem-specific, and full of tradeoffs. Simpler text data, simpler models, smaller vocabularies. You can always make things more complex later to see if it results in better model skill. Machine learning is frequently characterized by a singular focus on model selection. Be it logistic regression, random forests, Bayesian methods, or artificial neural networks, machine learning practitioners are often quick to express their preference. The reason for this is mostly historical. Though modern third-party machine learning libraries have made the deployment of multiple models appear nearly trivial.

Dimension reduction (DR) is a per processing step for reducing the original dimensions. The aims of dimension reduction strategies are to improve the speed and precision of data mining. The four main strategies for DR are: Supervised-Feature Selection (SFS), Unsupervised Feature Selection (UFS), Supervised Feature Transformation (SFT), and Unsupervised Feature Transformation (UFT). Feature selection emphases on finding feature subsets that better describes the data, as good as the original dataset, for supervised or unsupervised learning tasks[Kaur & Chhabra, (2014)]. Unsupervised implies there is no trainer, in the form of class labels. It is important to note that DR is but a preprocessing stage of classification.

In terms of performance, having data of high dimensionality is problematic because (a) it can mean high computational cost to perform learning and inference and (b) it often leads to over fitting when learning a model, which means that the model will perform well on the training data but poorly on test data. Dimensionality reduction addresses both of these problems while trying to preserving most of the relevant information in the data needed to learn accurate, predictive models.

Keywords: Dimension Reduction; J48; WEKA; MATLAB

2. J48 Algorithm

Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances, the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help of a tree classification algorithm, the critical distribution of the data is
Easily understandable.

J48 is an extension of ID3. The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open source Java implementation of the C4.5 algorithms. The WEKA tool provides a number of options associated with tree pruning. In case of potential overfitting, pruning can be used as a tool for précising. In other algorithms the classification is performed recursively until every single leaf is pure, that is the classification of the data should be as perfect as possible. This algorithm generates the rules from which particular identity of that data is generated. The objective is progressively generalization of a decision tree until it gains equilibrium of flexibility and accuracy.
The following shows the basic steps in the algorithm

- In case the instances belong to the same class the tree represents a leaf so the leaf is returned by Labeling with the same class.
- The potential information is calculated for every attribute, given by a test on the attribute. Then the gain in information is calculated that would result from a test on the attribute.
- Then the best attribute is found on the basis of the present selection criterion and that attribute selected for branching.

2.1 Counting Gain

This procedure uses the "ENTROPY" which is a measure of the data disorder. Entropy of $\overrightarrow{y}$ is calculated as

$$Entropy\ (\overrightarrow{y}) = -\sum_{j=1}^{n} \frac{|y_t|}{\vec{y}} \log\left(\frac{|y_t|}{|\vec{y}|}\right) \quad (1)$$

$$Entropy\ (j|\overrightarrow{y}) = -\sum_{j=1}^{n} \frac{|y_t|}{\vec{y}} \log\left(\frac{|y_t|}{|\vec{y}|}\right) \quad (2)$$

Making Gain

$$Gain(\overrightarrow{y}, j) = Entropy\ (\vec{y} - Entropy\ (j|\ \overrightarrow{y})) \quad (3)$$

2.2 Pruning

The outliers make this a very significant step to the result. Some occurrences are present in all datasets which are not well defined and also differ from the other occurrences in its neighborhood.

The classification is done on the instances of the training set and tree is formed. The pruning is done for decreasing errors in classification which are produced by specialization in the training set. Pruning is achieved for the generality of the tree.

2.3 Features of the Algorithm

- Both discrete and continuous attributes are handled by this algorithm. A threshold value is decided by C4.5 for managing continuous attributes. This value splits the data list into those who have their attribute value below the threshold and those having more than or equal to it.
- This algorithm also takes care of the missing values in the training data.
- After the tree is fully built, this algorithm does the pruning of the tree. C4.5 after its building drives back through the tree and challenges to eliminate branches that are not helping in reaching the leaf nodes.

3. Related Work

Decision tree classifiers are widely used supervised learning approaches for data exploration, resembling or approximation of a function by piecewise constant regions, also does not necessitate preceding information of the data distribution[Mitra & Acharya, (2003)]. Decision trees models are usually used in data mining to test the data and induce the tree and its rules that will be used to make predictions[Two Crows Corporation, (2005)]. The actual purpose of the decision trees is to categorize the data into distinctive groups that generate the strongest of separations in the values of the reliant variables [Parr Rud (2001)], being superior at identifying segments with the desired compartment such as activation or response, hence providing an easily interpretable solution.

The concept of decision trees was advanced and refined over many years by J. Ross Quinlan starting with ID3 [Interactive Dichotomizer 3 (2001)]. A method based on this approach uses an evidence theoretic measure, such as entropy, for assessing the discriminatory power of every attribute [Mitra & Acharya (2003)]. Major decision tree algorithms are clustered as [Mitra & Acharya (2003)]: (a) classifiers from the machine learning community: IDS, C4.5, CART; and (b) classifiers for large databases: SLIQ, SPRINT, SONAR, Rain Forest.

Weka is a very effective assemblage of machine learning algorithms to ease data mining tasks. It holds tools for data preparation, regression, classification, clustering, association rules mining, as well as visualization. Weka is used in this research to implements the most common decision tree construction algorithm: C4.5 known as J48 in weka. it is one of the more famous Logic Programming methods, developed by Quinlan [Quinlan JR (1986)], an attribute-based machine learning algorithm for creating a decision tree on a training set of data and an entropy measure to build the leaves of the tree. C4.5 algorithms are based on the ID3, with supplementary programming to address ID3 problems.

4. Proposed Technique and Framework

The WEKA tool has emerged with innovatory and effective as well as relatively easiest data mining and machine learning solutions. Since 1994, this tool was developed by the WEKA team. WEKA contains many inbuilt algorithms for data mining and machine learning. It is an open source and freely available platform. People with little knowledge of data mining can also use this software very easily since it provides flexible abilities for scripting experiments. As new algorithms appear in the research literature, these are updated in the software. WEKA has also gained some reputation which makes it one of the favorite tool for data mining research and assisted to progress it by making numerous powerful features available to all.

4.1 The following are steps performed for data mining in WEKA:

- Data pre-processing and visualization
- Attribute selection
- Classification (Decision trees)
- Prediction (Nearest Neighbor)
- Model evaluation
- Clustering (Cobweb, K-means)
- Association rules

4.2 J48 Improvement

The proposed algorithm uses MATLAB to refine the dataset for WEKA to apply the J48 algorithm for improved results.

Providing a basic MATLAB interface to WEKA to allow transformation of data for better output. Hence, increasing the accuracy rate of the J48 algorithm when applied to the same dataset. The proposed algorithm works by loading the arff data file from WEKA into MATLAB. The sanitizing of the dataset is then performed and the J48 classifier is applied after which the accuracy and error rate calculated. Fig 1, shows the flowchart of the proposed technique employed.

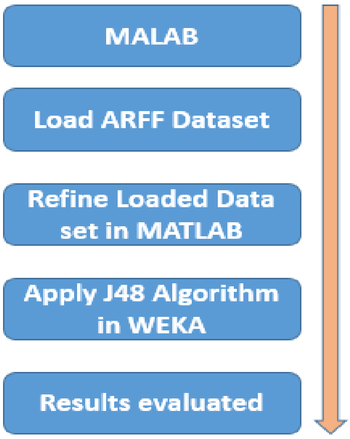


Fig 1: Flow Chart and Set-Up

5. Experimentation

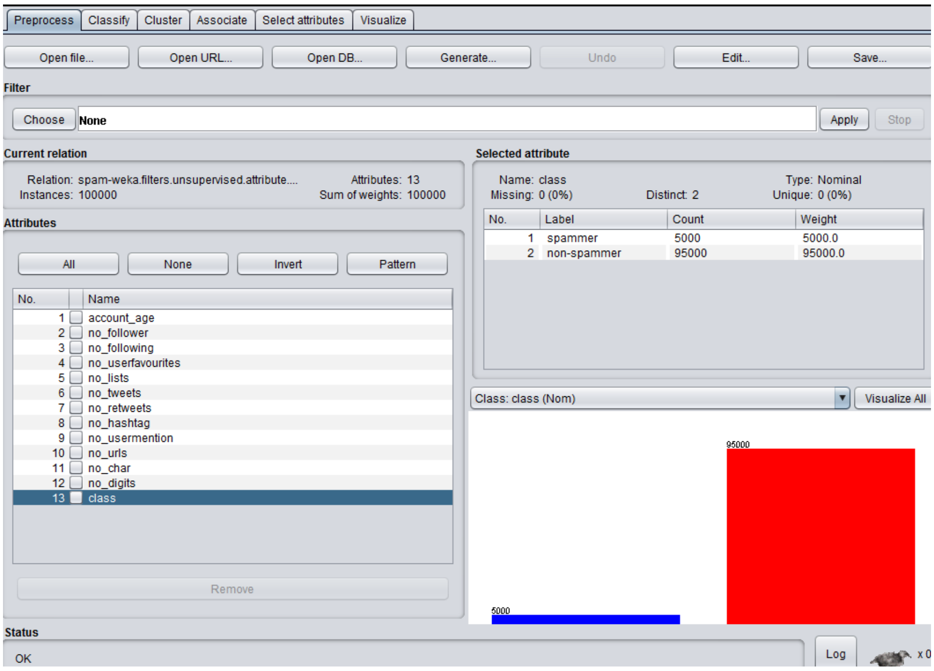
This section shows results and how performance was evaluated, the J48 algorithm is also compared to other algorithms.

The formula employed for calculating the accuracy is

$$TA = \frac{(TP+TN)}{TP+TN+FP+FN} \tag{4}$$

$$RA = \frac{(TP+FP)*(TN+FN)*(FN+TP)*(FP+TP)}{(Total*Total)} \tag{5}$$

In the equation (4) TA = Total Accuracy, TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative. In equation (5) RA represents Random Accuracy.
Fig 2, shows the tested negative and positive values of spammers with respect to the various attributes. It shows the total number of classified spammers and non-spammers per the dataset in WEKA environment.



Filter

None

Current relation

Relation: spam-weka.filters.unsupervised.attribute.... Attributes: 13
Instances: 100000 Sum of weights: 100000

Attributes

No.	Name
1	account_age
2	no_follower
3	no_following
4	no_userfavourites
5	no_lists
6	no_tweets
7	no_retweets
8	no_hashtag
9	no_usermention
10	no_urls
11	no_char
12	no_digits
13	class

Selected attribute

Name: class Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	spammer	5000	5000.0
2	non-spammer	95000	95000.0

Class: class (Nom)

Status

OK

Fig 2: Data representation by class in Weka environment

Table 1, indicates the output of classification represented in the following confusion matrix for spammers and non-spammers.

Table 1: Confusion Matrix

a	**b**	**classified as**
2316	2684	a=spammer
720	94280	b=non-spammer

Table 2 shows the results of various algorithms against the performance of the proposed improved technique.

Table 2: Performance comparison of other algorithms

Algorithm	**Accuracy %**	**Error%**
NaiveBayes	54.46	45.54
MultiClassClassifier	94.999	5.001
Random Tree	94.98	5.02
REPTree	96.347	3.653
Random Forest	96.962	3.038
J48	96.596	3.404
Improved J48	98.607	1.393

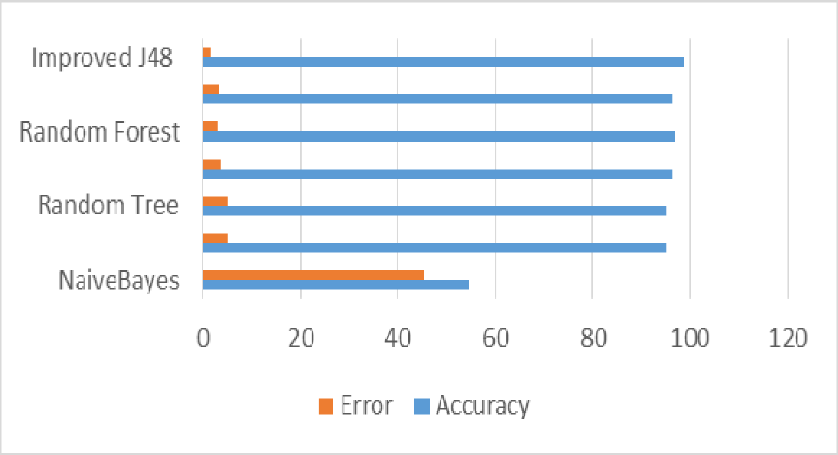


Fig 3: Results of algorithms in percentage

Fig 3 shows the comparison graph of the various algorithms on accuracy and error rate. It clearly shows how the improved technique performs better than the others with its accuracy rate of 98.607 %.

CONCLUSIONS AND FUTURE TREND

This research proposes an approach for efficient prediction of spammers from records of Twitter users. It is able to correctly predict spammers and no-spammers with u to 98.607% accuracy rate. The improved technique makes use of the data mining tool WEKA, which is used together with MATLAB for generating an improved J48 classifier. The experiment results speak for itself.

In the near future, some more datasets will be used to validate the proposed algorithm. Only 100000 instances were used for this research work, a larger and more dynamic dataset should be considered in other to test the effectiveness of this algorithm.

Acknowledgment

With deep appreciation, the author would like to thank Eng. Terry mudumbi and Professor Xu Dezhi for her support to this publication.

References

[1] Acharya, T; Mitra, S. (2003). Data Mining: Concept and Algorithms from Multimedia to Bioinformatics, John Wiley & Sons, Inc. New York.

[2] Aggarwal, C. C; Zhai C. (2012): An introduction to text mining in Mining Text Data, Springer, pp. 1–10.

[3] Cunningham, P. (2008): Dimension reduction, Machine learning techniques for multimedia, **13** pp. 91–112.

[4] Council, N. (2016): Future directions for nsf advanced computing infrastructure to support u.s science and engineering in 2017-2020: Interim report, The National Academies Press Washington, DC.

[5] Flach , A.; Wu, S. (2002): Feature selection with labelled and unlabelled data, in ECML/PKDD, **2** pp. 156–167.

[6] Gao, L.; Song, J.; Liu, X.; Shao, J.; Liu, J.; Shao, J. (2017): Learning in high- dimensional multimedia data: the state of the art, Multimedia Systems, vol. 23, pp. 303–313.

[7] Hϋllermeier, E. (2011): Fuzzy sets in machine learning and data mining, Applied Soft Computing, vol. 11, no. 2, pp. 1493–1505.

[8] Karami, A. (2015): Fuzzy Topic Modeling for Medical Corpora, University of Maryland, Baltimore County.

[9] Karami, A.; Gangopadhyay, A. (2014): A fuzzy feature transformation method for medical documents, in Proceedings of the Conference of the Association for Computational Linguistics (ACL), vol. 128, pp 16-22.

[10] Karami, A.; Gangopadhyay, A; Zhou, B. and H. Kharrazi, “Flatm: A fuzzy logic approach topic model for medical documents,” in Proceed- ings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS). IEEE, 2015.

[11] Karami, A.; Yazdani, H.; Beiryaie, R.; Hosseinzadeh, N. (2010): A risk based model for is outsourcing vendor selection, in Proceedings of the IEEE International Conference on Information and Financial Engineering (ICIFE). IEEE, pp. 250–254.

[12] Karami, A.; Guo, Z. (2012): A fuzzy logic multi-criteria decision frame- work for selecting it service providers, in Proceedings of the Hawaii International Conference on System Science (HICSS). IEEE, pp. 1118–1127.

[13] Lee D. D.; Seung, H. S. (1999): Learning the parts of objects by non- negative matrix factorization, Nature, vol. 401 **6755**, pp. 788.

[14] Liu, H.; Motoda, H. (2007): Computational methods of feature selection, CRC Press.

[15] Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K. R. (1999): Fisher discriminant analysis with kernels, in Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop. IEEE, pp. 41–48.

[16] Pedersen, J. O.; Yang Y. ; (1997) : A comparative study on feature selection in text categorization, in Icml, vol. 97, pp. 412–420.