A Comprehensive Study of Text Classification Algorithms

Vikas K Vijayan, Bindu K.R, Latha Parameswaran

Department of Computer Science and Engineering
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
Amrita University, India
vikaskvijay@gmail.com, j_bindu@cb.amrita.edu, p_latha@cb.amrita.edu

Abstract—**Huge amount of data in today's world are stored in the form of electronic documents. Text mining is the process of extracting the information out of those textual documents. Text classification is the process of classifying text documents into fixed number of predefined classes. The application of text classification includes spam filtering, email routing, sentiment analysis, language identification etc. This paper discusses a detailed survey on the text classification process and various algorithms used in this field.**

Keywords—Text Classification; KNN; Naïve Bayes; Rocchio; SVM; Regression; Neural Network; Rule Based; Decision Tree;

I. INTRODUCTION

Text classification is the process of classifying text documents into a predefined set of classes. It is a supervised learning approach in which a training set of documents $\{D_1, D_2 \ldots D_n\}$ labelled with classes from $\{1 \ldots .m\}$ are used to build a classification model and predicts the class label of a new incoming document based on the training model. Text classification types include single label and multi-label classification. When a document is assigned with only one class it is called single labelled and when more than one class is assigned for a document it becomes multi-label classification. Binary classification which predicts if a document belongs to a particular class or not is the best example of a single label classification. Text categorization has various stages such as pre-processing, indexing and dimensionality reduction, classification and performance evaluation.

In pre-processing, the text documents are split into small tokens which may be words or phrases. These tokens are subjected to stop word removal process which removes the most frequent insignificant words like 'the', 'a' etc. Stemming is also applied to those token streams for converting the words to its root form.

The contents of text documents has to be represented in some form before being fed into a classifier, this is called Indexing. The features or terms are used to represent a document. The terms may be words or phrases. A commonly used scheme is the Bag of the words model which represents the document as set of words or a word vector. The weight assigned to the word will decide the relevance of the word in the document. Binary Indexing assigns a weight 1 if the word is present in the document and 0 if it is not. TF-IDF is another weighting method which assigns a weight taking into account the Term Frequency and Inverse Document Frequency. It gives higher weight for a word in the document if the Term Frequency i.e. number of times the word appearing in the document is high and less weight if the Document Frequency i.e. the no of training documents in which the word appearing is high. TF-IDF score doesn't take the semantics of the document into consideration which is seen as a disadvantage of this method. There are also other Indexing methods including probabilistic methods and the selection of Indexing methods for a classification model are very much dependent on the availability of training samples.

Dimensionality Reduction reduces the feature vector space of the documents by selecting or extracting a subset of terms out of the original one. A classifier yields a better result if the Dimensionality Reduction technique were applied before classification. Document Frequency which selects the highest frequent words across documents is an effective feature selection method. There are also other methods based on Information Theory like Information Gain, Mutual Information, DIA Association Factor, Chi-Square, NGL Coefficient, Odds Ratio etc. These methods are observed to be more efficient and outperforms Document Frequency by fair margin. Another way of Dimensionality Reduction is by feature extraction. Here the extracted features may not be the subset terms but a compact representation of the features. Some of the methods used in this scenario are Term Clustering and Latent Semantic Indexing (LSI). Term Clustering tries to cluster the terms and represent the feature vector space with a cluster member. LSI uses single value decomposition to transform a higher dimensional vector to a lower dimensional one. The feature extraction handles the problems of polysemy and synonymy well, discriminating this from other methods.

Now comes the classification which is applied onto the pre-processed text data. There are several classifiers such as Decision Tree classifiers, Rule Based classifiers, Probabilistic and Naïve Bayes classifiers, Regression Based classifiers, Proximity Based classifiers, Neural Network Classifiers etc. All the classifiers have its own good and bad making them suitable for specific models. We will give an elaborate discussion on classifiers in the coming sections.

The final stage provides a performance analysis of these classifiers. The different metrics used here includes Precision, Recall, Error, Accuracy etc. The coming section will provide a detailed information on the type of classifiers, scenarios in

978-1-5090-6367-3/17/$31.00 ©2017 IEEE

which they can be used and the advantages and disadvantages of each one of them.

II. CLASSIFICATION ALGORITHMS

A. Rocchio Classification

Rocchio method of text classification [1] finds the centroid of each class from the training set of documents and classifies a text document to the nearest centroid class. The document can be represented using the vector space model and the centroid of the class is the vector average of its members. The formula for finding the centroid of a class is given below. [5]

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d) \tag{1}$$

In (1) D_c represents the number of documents in D whose class is c and $\vec{v}(d)$ is the normalized vector of d. The nearness between the vectors can be estimated based on Euclidean distance or other similarity measurements like dot product. The Rocchio classification finds the separating hyperplanes between classes based on the centroid. The algorithm finds classes with spherical shapes having nearly same radius making it difficult for two class classification problems. The classification can become inappropriate when the centroid of the class fail to represent the class structure for example, when the class contain small clusters of samples [2]. The algorithm gains in terms of its simplicity, linearly complex training phase and a small computation testing phase. Some variations of Rocchio can be found in [3].

B. KNN Classification

K Nearest Neighbour is an instance-based, non-parametric text classifier which uses similarity measurement (dot product, cosine similarity) as a criteria for classifying the documents. The training documents are represented using the vector space model and the nearest k neighbours of incoming test document are found out by comparing the cosine similarity of it with each training document. The test document is classified into the majority class among the k nearest neighbours. Equation (2) is used for calculating the cosine similarity between test document q and training document d_j. [6]

$$\cos(d_j, q) = \frac{\sum_{i=1}^{|V|} d_{ij} q_i}{\sqrt{\sum_{i=1}^{|V|} d_{ij}^2} \sqrt{\sum_{i=1}^{|V|} q_i^2}} \tag{2}$$

Here d_{ij} and q_i represents the tf-idf weight of the term i in in training document d_j and test document q respectively. The KNN approach simply stores the training data and learning happens only at the arrival of test data resulting in higher computational time for prediction. There are many improvements tried out on the traditional KNN approach. In reference [4] employs KNN algorithm for text classification with μ-Cooccurrence(A method based on feature interaction) as the feature selection method. The technique showed better results by selecting less number of relevant features making it suitable for classification containing large data sets. In reference [5] talks about a density based KNN approach to handle the dataset which are not evenly distributed. The distance between the test data and k nearest neighbour's are adjusted based on their density difference in order to cope with the non-uniform distribution of training data. Experimental results have shown that DBKNN is an improved and stabilized KNN algorithm. In [6] the authors introduce an eager learning approach over the store and process method of KNN algorithm by constructing a model of the feature weights from the training samples. The proposed algorithm reduces the computational time of the traditional KNN and improves the classification accuracy of sensitive information.

C. Naïve Bayes Classification

Naïve Bayes is a simple probabilistic classifier which works on the assumption of conditional independence between the features of a text document. Given a text document, the Naïve Bayes classifier find out the class with maximum posterior probability. It is based on the Bayes rule which says [7]

$$p(c/d) = \frac{p(d/c)p(c)}{p(d)} \tag{3}$$

$p(c/d)$, is the probability of document d to belong to class c called the posterior probability, $p(d/c)$ is the likelihood and $p(c)$ is the prior.

Naïve Bayes classifies the document to the class which maximizes the posterior probability.

$$C_{map} = \underbrace{\arg\max}_{c \in C} P(c/d) \tag{4}$$

$$C_{map} = \underbrace{\arg\max}_{c \in C} P(c/d)p(c)$$

Here $p(d)$ can be ignored as it is a constant.

$$C_{map} = \underbrace{\arg\max}_{c \in C} P(X_1, X_2, X_3 ..... X_n / c)p(c)$$

$$C_{map} = \underbrace{\arg\max}_{c \in C} p(X_1/c) \times p(X_2/c) \times p(X_3/c) ..... p(X_n/c)p(c) \tag{5}$$

Equation (5) is derived from the Naïve or conditional independence assumption. Here document d is assigned to the class with highest C_{map} . There are mainly two models for Naïve Bayes classification, they are Multivariate Bernoulli Model and the Multinomial Model [2]. The Multivariate model has binary representation of features while the later represents the features with term frequency. Naïve Bayes classifier seem to work well even with the conditional independence assumption. In [8] authors uses Naïve Bayes to perform spam email detection and reveals the necessity of increased training samples for an accurate classification. The author suggests a dynamic tuning of the word probabilities during classification for getting an improved model for prediction. In [9] authors give more importance to the terms present in the title of the text document by assigning more weight while computing the posterior probability. The technique shows more accuracy in prediction as compared to the standard Naïve Bayes algorithm. In [10] authors apply Naïve Bayes classification on documents using CHIR algorithm as a feature selection method, which unveils the type of relationship between terms and classes. The performance report showed an improved result with the proposed method over Naïve Bayes classification employing Chi-square statistic as feature selection strategy. Naïve Bayes can easily accommodate any domain specific knowledge and also works well with hierarchical classification scenario [2].

D. SVM Classification

Support Vector Machines are linear classifiers suitable for classifying high dimensional data. SVM performs well in text classification scenario as it involves a high dimensional feature space. SVM tries to find out the maximum separating hyper plane between different classes.

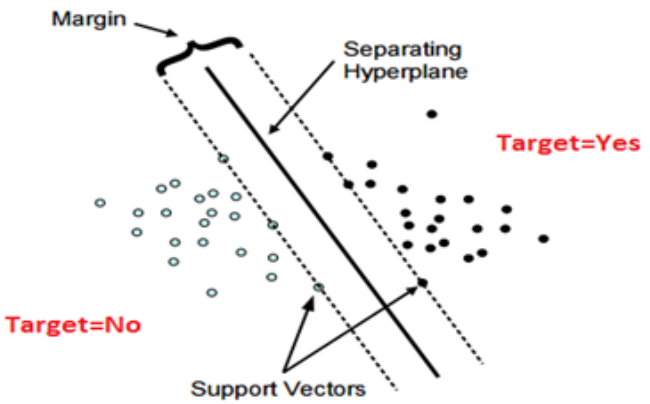

Fig. 1. Figure representing SVM [25]

In Fig.1 [25] among the three lines the bold line best separates the two classes and it is the maximum marginal hyper plane. The linear equation describing this hyper plane will be of the form Y = AX + b. Here X is the feature vector representation, A is the coefficient vector and b is a constant. This predictor hyper plane will classify the documents to different classes. The points on the dotted lines are the support vectors and they are the deciding factors in categorization. SVM can also model nonlinear decision boundaries in base feature space by applying kernel trick. In paper [11] authors say that most text classification problem are linearly separable making SVM a promising choice for text classification. The reference [12] talks about an optimal SVM classification which takes into account the frequencies of terms in individual documents, a feature selection measure using likelihood ratio for binomial distribution and specific parameter tuning. The method results in improved performance over KNN, Naïve Bayes and decision tree classification on Reuter-21578 and TREC-AP. Reference [13] applied improved local linear embedding algorithm with two loss functions for dimensionality reduction and also combined the supervised learning technique along with SVM for classification.

E. Regression Based Classification

Regression is applied in text classification even though it normally addresses the case of real valued attributes. It is a linear classifier and the most common linear regression estimate is the least squares algorithm. Linear regression models a function of the form $f(x)=W^T X$ to match the training data [14]. The weight vector w is estimated by minimizing the squared difference between the predicted value and the true value(y). i.e.

$$\hat{W} = \underbrace{\arg\min}_{c \in C} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - W^T X_i \right)^2 \right\} \qquad (6)$$

There may be many solutions for this function and the ridge regression computes the unique solution for the objective function by adding a component $\lambda W^T W$ to it [15] where λ is the regularization coefficient.

$$p(C = y_i / X_i) = \frac{\exp(\overline{A}.\overline{X_i} + b)}{1 + \exp(\overline{A}.\overline{X_i} + b)} \qquad (7)$$

(7) Represents the logistic regression where X is the term matrix, A is the regression coefficient vector and b is a constant. Logistic regression estimates the regression parameters $\overline{A}$ which maximizes the conditional likelihood $\prod_{i=1}^{n} p(y_i / X_i)$. The regression based classification has fairly expensive training phase which includes the parameter modelling with optimization techniques like single value decomposition.

F. Neural Network Based Classification

Neural networks can be easily used to model linearly separable classification scenarios. Neurons, the smallest processing elements in the neural network takes a set of input,

performs computations and produce an output. The term frequency vector X_i of the document i is given as input to the neuron, the neuron computes a linear function $P_i = W.X_i$ and output a result. Here P_i is the predicted output and W is the weight vector. The weight vector will have the same dimension as of the input vector. The classifier tries to find out the weight vectors which results in the correct classification of the training data. The perceptron learning model for binary classification tries to see if the predicted label P_i is matching the true label y_i of the training document, if there is a mismatch the neuron will adjust its weight vectors in accordance with a predefined learning rate μ to make the predicted and true label same. The neural network approach can also be applied for nonlinear decision boundaries by adding multiple layers in its design where the output of one neuron will be the input for another one. The major performance degrader for this approach is the high computational cost in the training phase.

G. Rule Based Classification

Rule based classifier constructs a rule set from the training set of documents and classifies the test instance based on the modelled rules. The difference between the rule based classifier and a decision tree classifier is on the hierarchical approach of the later. The rules have a left hand side which contains a conjunction of document features and the occurrence of those features in the test document implies the adherence of the test document to the corresponding rule. Then the document is classified to the class which is on the right hand side of the rule. When a test document satisfies many such rules and the right hand side have many classes then ranking of the rules are commonly done to select the class which is most frequent among the top ranked rules. Ripper rule learning algorithm is a commonly used approach in rule based classification whose focus is on positive training samples for learning words. The work in [16] is an improved RIPPER learning algorithm by bringing the hierarchical categories in rule set development. The RIPPER method prune the words which occur in more than one category. This problem is addressed in the prescribed work which brings a new category to accommodate words or rules that spans more than one category. The algorithm makes the rules strong and give better results. The advantage of rule based learning is that the rule set modification is comparatively easy for a new set of training documents. It is also possible to formulate rules which best describes a class content making it suitable for specific cases.

H. Decision Tree Classification

The decision tree classification is an inductive learning approach which performs a hierarchical partitioning of the training document space. There are both internal and leaf nodes in a decision tree classifier. The internal nodes contain rules which raises questions on the incoming document and place the document to a path in the sub tree matching test predicate outcome. The document traversing the tree in a top down fashion will finally end up in a leaf node which represents a class. The document is then classified to the class represented by the leaf node. The decision predicates in the internal nodes can be the presence or absence of the terms in text documents. Pruning of nodes are done commonly to avoid data overfitting. It is done by separating a portion of data from the training documents during the tree construction and later using it to validate the constructed tree. ID3 [17] is a common decision tree algorithm which uses information gain as an attribute selection criteria. Authors of paper [18] compares the performance of SVM, Naïve Bayes and Decision Tree and finds SVM to be the accurate out of the three. It uses C4.5 algorithm for decision tree which is a follower of ID3. Boosting techniques are also employed with decision trees for performance gain. The work in [19] is a comparison between a Bayesian classifier and decision tree algorithm and it reveals the effectiveness of a stepwise feature selection approach with decision tree classifier for large feature sets.

I. Other Techniques

The bag of the words model does not capture the synonyms or the semantics between the terms of the document. The application of topic modeling techniques [20] [24] [26] in text categorization to group related words into clusters can address these problems. Paper [20] proves the use of LDA as a feature representation model provides improved result over the Bag of the words model. LDA, Latent Dirichlet Allocation [21] is a generative probabilistic model with document as a random mixture of hidden topics and topics distributed over terms. The work in [22] uses LDA for dimension reduction with SVM classifier for Chinese news text and compares it with the TF-IDF method. There are other cross domain text categorization algorithms like Expectation-maximization, Latent Semantic Analysis and Probabilistic Latent Semantic Analysis etc. [23]

The Table I. provides a comparison at a glance between the above mentioned algorithms.

TABLE I. COMPARISON BETWEEN CLASSIFICATION ALGORITHMS

Classifier	Characteristics	Limitations
Rocchio Classification	Simple and efficient, Training and testing phase are linear.	Becomes inaccurate when the centroid of the classes does not represent its behaviour well.
KNN Classification	Proximity-based classifier, Training data is stored, Classes may not be linearly separable.	Higher time and space complexity as it stores all the instance, Noisy features degrades the classification accuracy.
Naïve Bayes classification	Simple and efficient, Linear computational time, Domain specific knowledge can be easily included,	Independence assumption of features.

Classifier	Characteristics	Limitations
	Insensitive to noisy features.	
SVM Classification	Linear classifiers, Handles high dimensional data well, can handle nonlinear decision boundaries, works with large size unlabeled and small size labelled data.	High time and space complexity during training and testing.
Regression Based classification	Linear classifier, domain information can be encoded.	Training phase is costly.
Neural Network Based Classification	Model linear and nonlinear decision boundaries.	Higher Computational cost in training.
Rule Based Classification	Rule set modification is easy for new data, can make rules for specific cases.	Computational cost is high.
Decision Tree Classification	Hierarchically partition the training space, Non parametric.	Noise handling is bad, no online learning, overfit.

III. CONCLUSION

Text classification has gained its importance in recent years, resulting in the application of various data mining algorithms for text domain. The high dimensional features and hidden semantics in the text data are the performance limiting factors of many such algorithms. All the algorithms stated in the survey has its own pros and cons and a good performance on classification demands the right choice of classifier for the right problem. A right choice of classifier combined with an appropriate dimensionality reduction technique would definitely improve the expected outcome of classification.

REFERENCES

[1] "Rocchio classification", Nlp.stanford.edu, 2017. [Online]. Available: : http://nlp.stanford.edu/IR-book/html/htmledition/rocchio-classification-1 html. [Accessed: 29- Jan- 2017].

[2] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," Springer, 2012, pp. 163-222.

[3] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," DTIC Document, 1996.

[4] P. Soucy and G. W. Mineau, "A simple KNN algorithm for text categorization," ICDM, 2001, pp. 647-648.

[5] K. Shi, L. Li, H. Liu, J. He, N. Zhang, W. Song, "An improved KNN text classification algorithm based on density," IEEE International Conference on CCIS, pp. 113-117, 2011.

[6] T. Dong, W.Cheng and W. Shang, "The research of KNN text categorization algorithm based on eager learning," International Conference on ICICEE, IEEE, pp. 1120-1123, 2012

[7] G. Aghila and others, "A survey of naive bayes machine learning approach in text document classification," arXiv preprint arXiv: 1003.1795, 2010.

[8] H. Zhang and D. Li, "Naïve Bayes text classifier," IEEE International conference on GRC 2007, pp. 708-708, 2007.

[9] P. Bai, and J. Li, "The improved Naïve Bayesian WEB text classification algorithm," International Symposium on CNMT 2009, IEEE, pp. 1-4, 2009.

[10] M. J. Meena and K. R. Chandran, "Naive Bayes text classification with positive features selected by statistical method," First Inter. Conf. on Advan Comp. ICAC, IEEE, pp. 28-33, 2009.

[11] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features" in Machine learning ECML-98, Springer, pp. 137-142, 1998.

[12] Z. Wang, X. Sun and D. Zhang, "An optimal text categorization algorithm based on svm," Inter. Conf. on Comm., Circuits and Systems Proceedings, IEEE, vol. 3, pp. 2137-2140, 2006.

[13] L. Youwen, X. Shixiong and Z.Yong, "A supervised local linear embedding based SVM text classification algorithm," Sixth WISA 2009, IEEE, pp. 21-26, 2009.

[14] J. Zhang and Y. Yang, "Robustness of regularized linear classification methods in text categorization," Proc. Of the 26th annual inter. ACM SIGIR conf. on Research and Dev. in information retrieval, ACM, pp. 190-197, 2003.

[15] T. Zhang and F. J. Oles, "Text categorization based on regularized linear classification methods," in Information retrieval, vol. 4(1), Springer, pp. 5-31, 2001.

[16] M. Sasaki and K. Kita, "Rule-based text categorization using hierarchical categories," Inter. Conf. on Systems, Man and Cybernetics, IEEE, vol. 3, pp. 2827-2830, 1998.

[17] J. R. Quinlan, "Induction of decision trees," in Machine learning, vol. 1(1), Springer, pp. 81-106, 1986.

[18] G. S. Chanvan, S. Manjare, P. Hedge and A. Sankhe, "A Survey of Various Machine Learning Techniques for Text Classification," in IJETT, vol. 15, pp. 288-292, 2014.

[19] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," third annual symposium on document analysis and information retrieval, vol. 33, pp. 81-93, 1994.

[20] W. Sriurai, "Improving text categorization by using a topic model," in Advanced Computing, vol. 2(6), AIRCC, pp. 21, 2011.

[21] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation," in Journal of machine learning research, vol. 3(Jan), pp. 993-1022, 2003.

[22] X. Wu, L. Fang, P. Wang and N. Yu, "Performance of using lda for chinese news text classification," 28th CCECE, IEEE, pp. 1260-1264, 2015.

[23] M. R. Murty, J. V. R. Murthy, P. P. Reddy and S. C. Satapathy, "A survey of cross-domain text categorization techniques," 1st inter. Conf. on RAIT, IEEE, pp. 499-504, 2012.

[24] K. R. Bindu, L. Parameswaran, K. V. Soumya, "Performance Evaluation of Topic Modelling Algorithms with an application of Q & A Dataset, " International Journal of Applied Engineering Research, vol. 10, pp. 23-27, 2015.

[25] D. Institute, "Building Predictive Model using SVM and R-Dnl Institute ", Dni-institute.in, 2017. [Online]. Available: http://dni-institute.in/blogs/building-predictive-model-using-svm-and-r. [Accessed: 19- May- 2017].

[26] K. R. Bindu, L. Parameswaran, *Sandeep R Nambiar, Jithin Chandran*, "Performance Evaluation of Algorithms for Expert finding on an Open Email dataset, " *International Journal of Applied Engineering Research, vol. 10, pp. 71-75, 2015.*