ISSN:0975-9646

Classification of Text Data by Fuzzy Self-Constructing Feature Clustering Algorithm

Y. Ratna Kumari, R. Siva Ranjani

CSE Department, GMR Institute of Technology,
JNTU Kakinada, Rajam, AP,INDIA

Abstract— **Feature clustering is a powerful method to reduce the dimensionality of feature vectors for text classification. In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The words in the feature vector of a document set are grouped into clusters, based on similarity test. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature, corresponding to a cluster, is a weighted combination of the words contained in the cluster. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided. Experimental results show that our method can run faster and obtain better extracted features than other methods.**

Keywords— **Feature Reduction, Feature Extraction, Feature Selection.**

I. INTRODUCTION

In text classification, the dimensionality of the feature vector is usually huge. For example, 20 Newsgroups and Reuters21578 top 10, which are two real-world data sets, both have more than 15,000 features. Such high dimensionality can be a severe obstacle for classification algorithms. To alleviate this difficulty, feature reduction approaches are applied before document classification tasks are performed. Two major approaches, feature selection and feature extraction have been proposed for feature reduction. In general, feature extraction approaches are more effective than feature selection techniques, but are more computationally expensive. Therefore, developing scalable and efficient feature extraction algorithms is highly demanded for dealing with high-dimensional document data sets.

Classical feature extraction methods aim to convert the representation of the original high-dimensional data set into a lower-dimensional data set by a projecting process through algebraic transformations. For example, Principal Component Analysis, Linear, Discriminant Analysis Maximum Margin Criterion and Orthogonal Centroid algorithm perform the projection by linear transformations, while Locally Linear Embedding , ISOMAP , and Laplacian Eigen maps do feature extraction by nonlinear transformations. In practice, linear algorithms are in wider use due to their efficiency. Several scalable online linear feature extraction algorithms have been proposed to improve the computational complexity. However, the complexity of these approaches is still high. Feature clustering, is one of effective techniques for feature reduction in text classification. The idea of feature clustering is to group the original features into clusters with a high degree of pair wise semantic relatedness. Each cluster is treated as a single new feature, and, thus, feature dimensionality can be drastically reduced.

The first feature extraction method based on feature clustering was proposed by Baker and McCallum which was derived from the "distributional clustering" idea of Pereira et al. Al-Mubaid and Umair used distributional clustering to generate an efficient representation of documents and applied a learning logic approach for training text classifiers. The Agglomerative Information Bottleneck approach was proposed by Tishby et al. The divisive information-theoretic feature clustering algorithm was proposed by Dhillon et al. which is an information-theoretic feature clustering approach, and is more effective than other feature clustering methods. In these feature clustering methods, each new feature I generated by combining a subset of the original words. However, difficulties are associated with these methods. A word is exactly assigned to a subset, i.e., hard-clustering, based on the similarity magnitudes between the word and the existing subsets, even if the differences among these magnitudes are small. Also, the mean and the variance of a cluster are not considered when similarity with respect to the cluster is computed. Furthermore, these methods require the number of new features be specified in advance by the user.

In text classification, the dimensionality of the feature vector is usually huge

- The current problem of the existing feature clustering methods
- The desired number of extracted features has to be specified in advance
- When calculating similarities, the variance of the underlying cluster is not considered
- How to reduce the dimensionality of feature vectors for text classification and run faster?

In this paper we propose a fuzzy similarity-based self-constructing feature clustering algorithm, which is an incremental feature clustering approach to reduce the

number of features for the text classification task. The words in the feature vector of a document set are represented as distributions, and processed one after another. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. If a word is not similar to any existing cluster, a new cluster is created for this word. Similarity between a word and a cluster is defined by considering both the mean and the variance of the cluster. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster. Three ways of weighting, hard, soft, and mixed, are introduced. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data.

II. ADVANTAGE

A fuzzy self-constructing feature clustering (FFC) algorithm which is an incremental clustering approach to reduce the dimensionality of the features in text classification

- Determine the number of features automatically
- Match membership functions closely with the real distribution of the training data
- Runs faster
- Better extracted features than other methods.

III. ARCHITECTURE DESIGN

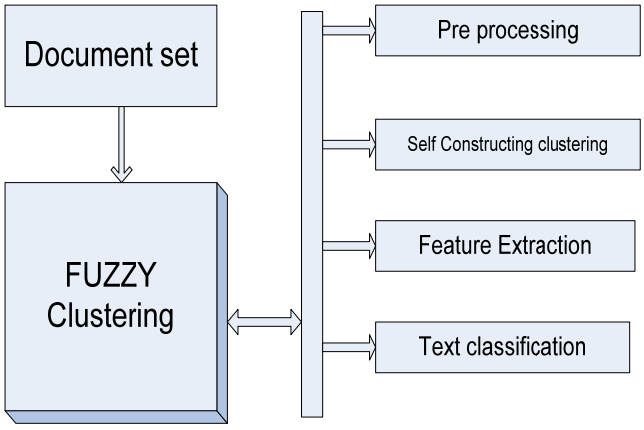


Fig. 1 Module Architecture diagram

IV. FEATURE REDUCTION

In general, there are two ways of doing feature reduction, feature selection, and feature extraction. By feature selection approaches, a new feature set $\mathbf{W}' = \{w'_1, w'_2, \ldots, w'_k\}$ is obtained, which is a subset of the original feature set W. Then W' is used as inputs for classification tasks. Information Gain (IG) is frequently employed in the feature selection approach. It measures the reduced uncertainty by an information-theoretic measure and gives each word a weight. The weight of a word w_j is calculated as follows:

$$IG(w_j) = -\sum_{l=1}^{p} P(c_l)logP(c_l) + P(w_j)\sum_{l=1}^{p} P(c_l|w_j)logP(c_l|w_j) + P(\overline{w}_j)\sum_{l=1}^{p} P(c_l|\overline{w}_j)logP(c_l|\overline{w}_j),$$

where P(cl) denotes the prior probability for class cl, P(wj) denotes the prior probability for feature wj, P(wj) is identical to 1 _ P(wj), and P(cl|wj) and P(cl|wj) denote the probability for class cl with the presence and absence, respectively, of wj. The words of top k weights in W are selected as the features in W'.

In feature extraction approaches, extracted features are obtained by a projecting process through algebraic transformations. An incremental orthogonal centroid (IOC) algorithm. Let a corpus of documents be represented as an m*n matrix $\mathbf{X} \in \mathbf{R}^{m \times n}$, where m is the number of features in the feature set and n is the number of documents in the document set. IOC tries to find an optimal transformation matrix $\mathbf{F}^* \in \mathbf{R}^{m \times k}$, where k is the desired number of extracted features, according to the following criterion:

$$\mathbf{F}^* = \arg\max trace(\mathbf{F}^T \mathbf{S}_b \mathbf{F}),$$

where $\mathbf{F} \in \mathbf{R}^{m \times k}$ and $\mathbf{F}^T\mathbf{F} = \mathbf{I}$, and

$$\mathbf{S}_b = \sum_{q=1}^{p} P(c_q)(\mathbf{M}_q - \mathbf{M}_{all})(\mathbf{M}_q - \mathbf{M}_{all})^T$$

with $P(c_q)$ being the prior probability for a pattern belonging to class c_q, M_q being the mean vector of class c_q, and Mall being the mean vector of all patterns.

V. FEATURE CLUSTERING

Feature clustering is an efficient approach for feature reduction, which groups all features into some clusters, where features in a cluster are similar to each other. The feature clustering methods proposed in are "hard" clustering methods, where each word of the original features belongs to exactly one word cluster. Therefore each word contributes to the synthesis of only one new feature. Each new feature is obtained by summing up the words belonging to one cluster. Let D be the matrix consisting of all the original documents with m features and D' be the matrix consisting of the converted documents with new k features. The new feature set $\mathbf{W}' = \{w'_1, w'_2, \ldots, w'_k\}$ corresponds to a partition

$\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_k\}$ of the original feature set W, i.e., $\mathbf{W}_t \cap \mathbf{W}_q = \emptyset$, where $1 \le q,t \le k$ and $t \ne q$. Note that a cluster corresponds to an element in the partition. Then, the tth feature value of the converted document d'$_i$ is calculated as follows:

$$d'_{it} = \sum_{w_j \in \mathbf{W}_t} d_{ij},$$

which is a linear sum of the feature values in Wt.
The divisive information-theoretic feature clustering (DC) algorithm, proposed by Dhillon et al.calculates the distributions of words over classes,

$$P(\mathbf{C}|w_j), 1 \le j \le m,$$

where $\mathbf{C} = \{c_1, c_2, \ldots\ldots, c_p\},$ and uses Kullback-Leibler divergence to measure the dissimilarity between two distributions. The distribution of a cluster Wt is calculated as follows:

$$P(\mathbf{C}|\mathbf{W}_t) = \sum_{w_j \in \mathbf{W}_t} \frac{P(w_j)}{\sum_{w_j \in \mathbf{W}_t} P(w_j)} P(\mathbf{C}|w_j).$$

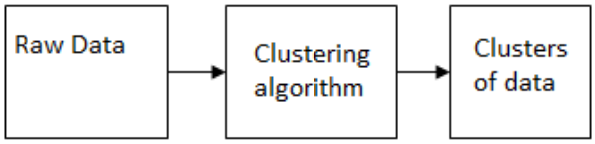
The goal of DC is to minimize the following objective function:

$$\sum_{t=1}^{k} \sum_{w_j \in \mathbf{W}_t} P(w_j) KL(P(\mathbf{C}|w_j), P(\mathbf{C}|\mathbf{W}_t)),$$

This takes the sum over all the k clusters, where k is specified by the user in advance.

VI. CLUSTERING ALGORITHM

Clustering is a tool for data analysis, which solves classification problems. Its object is to distribute cases (people, objects , events etc.) into groups, so that the degree of association to be strong between members of the same cluster and weak between members of different clusters. This way each cluster describes, in terms of data collected, the class to which its members belong. Clustering is discovery tool. It may reveal associations and structure in data which, though not previously evident, nevertheless are sensible and useful once found. The results of cluster analysis may contribute to the definition of a formal classification scheme, such as a taxonomy for related animals, insects or plants; or suggest statistical models with which to describe populations; or indicate rules for assigning new cases to classes for identification and diagnostic purposes; or provide measures of definition, size and change in what previously were only broad concepts; or find exemplars to represent classes. Whatever business you're in, the chances are that sooner or later you will run into a classification problem. Cluster analysis might provide the methodology to help you solve it.
In short: The algorithm Clustering attempts to find natural groups of components, based on some similarity.



The example below demonstrates the **clustering** of padlocks of same kind. There are a total of 10 padlocks which are of three different colors. We are interested in clustering of padlocks of the three different kind into three different groups.

The padlocks of same kind are clustered into a group as shown below:

Thus, we see clustering means grouping of data or dividing a large data set into smaller data sets of some similarity.

VII. RESULTS

	office (w_1)	building (w_2)	line (w_3)	floor (w_4)	bedroom (w_5)	kitchen (w_6)	apartment (w_7)	internet (w_8)	WC (w_9)	fridge (w_{10})	class
d_1	0	1	0	0	1	1	0	0	0	1	c_1
d_2	0	0	0	0	0	2	1	1	0	0	c_1
d_3	0	0	0	0	0	0	1	0	0	0	c_1
d_4	0	0	1	0	2	1	2	1	0	1	c_1
d_5	0	0	0	1	0	1	0	0	1	0	c_2
d_6	2	1	1	0	0	1	0	0	1	0	c_2
d_7	3	2	1	3	0	1	0	1	1	0	c_2
d_8	1	0	1	1	0	1	0	0	0	0	c_2
d_9	1	1	1	1	0	0	0	0	0	0	c_2

Table1: A Simple Document Set D

Fig1: Select the appropriate document from Document Set

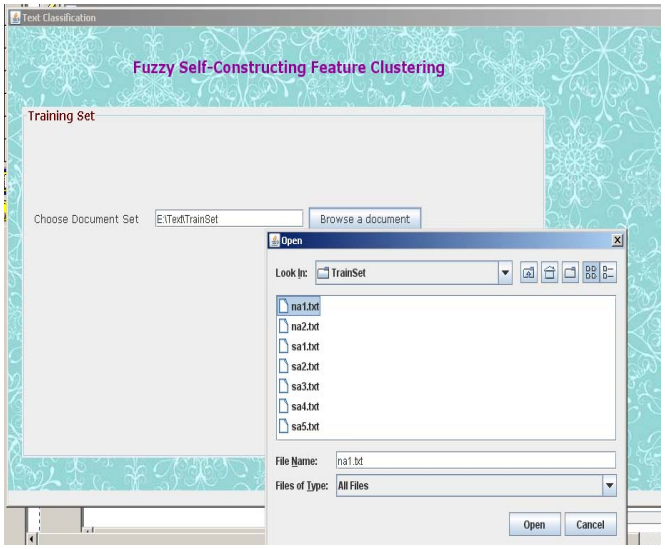


Fig2: select a document

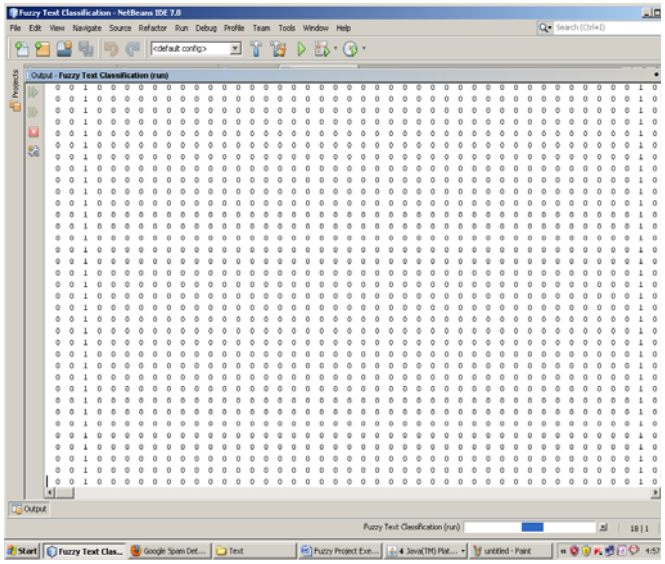


Fig3: Boolean matrix generation

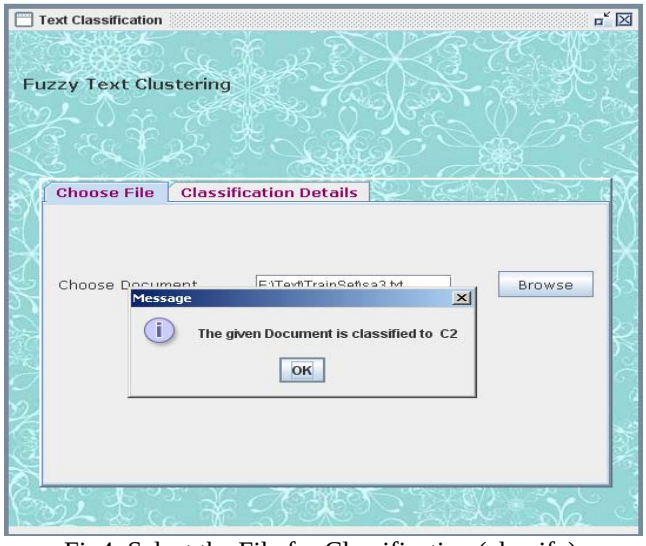


Fig4: Select the File for Classification (classify)

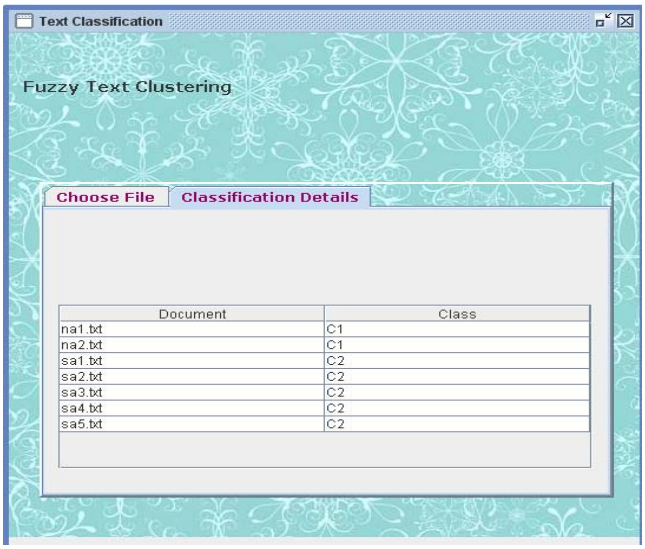


Fig5: Classification details on particular Document

VIII. CONCLUSION

We have presented a fuzzy self-constructing feature clustering (FFC) algorithm, which is an incremental clustering approach to reduce the dimensionality of the features in text classification. Features that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. If a word is not similar to any existing cluster, a new cluster is created for this word. Similarity between a word and a cluster is defined by considering both the mean and the variance of the cluster. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided. Experiments on three real-world data sets have demonstrated that our method can run faster and obtain better extracted features than other methods.

IX. FUTURE ENHANCEMENTS

Similarity-based clustering is one of the techniques we have developed in our machine learning research. In this paper, we apply this clustering technique to text categorization problems. We are also applying it to other problems, such as image segmentation, data sampling, fuzzy modeling, web mining, etc. The work of this paper was motivated by distributional word clustering. We found that when a document set is transformed to a collection of word patterns, as by, the relevance among word patterns can be measured, and the word patterns can be grouped by applying our similarity based clustering algorithm. Our method is good for text categorization problems due to the suitability of the distributional word clustering concept.

REFERENCES

[1] Http://people.csail.mit.edu/jrennie/20Newsgroups/, 2010.
[2] Http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html. 2010.
[3] H. Kim, P. Howland, and H. Park, "Dimension Reduction in Text Classification with Support Vector Machines," J. Machine Learning Research, vol. 6, pp. 37-53, 2005.
[4] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.
[5] B.Y. Ricardo and R.N. Berthier, Modern Information Retrieval. Addison Wesley Longman, 1999.
[6] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," Aritficial Intelligence, vol. 97, nos. 1/2, pp. 245-271, 1997.
[7] E.F. Combarro, E. Montan˜ e´s, I. Dı´az, J. Ranilla, and R. Mones, "Introducing a Family of Linear Measures for Feature Selection in Text Categorization," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 9, pp. 1223-1232, Sept. 2005.