ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

CLASSIFICATION OF TEXT USING FUZZY BASED INCREMENTAL FEATURE CLUSTERING ALGORITHM

ANILKUMARREDDY TETALI
M.Tech Scholar
Department of CSE ,
B V C Engineering college,
Odalarevu
akr.tetali@gmail.com

B P N MADHUKUMAR
Associate Professor
Department of CSE,
B V C Engineering college,
Odalarevu
bpnmadhukumar@hotmail.com

K.CHANDRAKUMAR
Associate Professor
Department of CSE,
VSL Engineering college,
Kakinada
chandhu_kynm@yahoo.com

Abstract:

The dimensionality of feature vector plays a major in text classification. We can reduce the dimensionality of feature vector by using feature clustering based on fuzzy logic. We propose a fuzzy based incremental feature clustering algorithm. Based on the similarity test we can classify the feature vector of a document set are grouped into clusters following clustering properties and each cluster is characterized by a membership function with statistical mean and deviation .Then a desired number of clusters are formed automatically. We then take one extracted feature from each cluster which is a weighted combination of words contained in a cluster. By using our algorithm the derived membership function match closely with real distribution of training data. By our work we reduce the burden on user in specifying the number of features in advance.

Keywords:

Incremental feature clustering, fuzzy similarity, dimensionality reduction, weighting matrix, text classifier.

Introduction:

A feature vector contains a set of features which are used for the classification of the text. The dimensionality of feature vector plays a major role in classification of text. For example if a document set have 100000 words then it becomes difficult task for the classification of text. To solve this problem, feature reduction approaches are applied before the classification of the text takes place. Feature selection [1] and feature extraction [2][3] approaches have been proposed for feature reduction**.**

Classical feature extraction methods uses algebraic transformations to convert the representation of the original high dimensional data set into a lower-dimensional data by a projecting process. Even though different algebraic transformations are available the complexity of these approaches is still high. Feature clustering is the most effective technique 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. McCallum proposed a first feature extraction algorithm which was derived from the "distributional clustering"[4] idea of Pereira et al. to generate an efficient representation of documents and applied a learning logic approach for training text classifiers. In these feature clustering methods, each new feature is generated by combining a subset of the original words and follows hard clustering, also mean and variance of a cluster are not considered. These methods impose a burden on the user in specifying the number of clusters.

We propose a fuzzy based incremental feature clustering algorithm, which is an incremental feature clustering[[5][6] approach to reduce the number of features for the text classification task. The feature vector of a document are grouped into clusters following clustering properties and each cluster is

characterized by a membership function with statistical mean and deviation This forms the desired number of clusters automatically. We then take one extracted feature from each cluster which is a weighted combination of words contained in a cluster. By using our algorithm the derived membership function match closely with real distribution of training data. Also user need not to specify the number of features in advance.

The main advantages of the proposed work are:

- A fuzzy incremental feature clustering (FIFC) 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 than other methods.
- Better extracted features than other methods.

Background and Related work:

Let D=<d1,d2…dn> be a document set of n documents, where d1,d2…dn are individual documents and each document belongs to one of the classes in the set {c1,c2…cp}. If a two or document belongs to two or more classes, then two or more copies of the document with different classes are included in D. Let the word set W={w1,w2...wn} be the feature vector of the document set. The feature reduction task is to find a new word set W0 such that W and W0 work equally but W0<W well for all the desired properties with D. Based on the new feature vector the documents are classified into corresponding clusters.

Dimensionality Reduction of the Feature Vector:

In general, there are two ways of doing feature reduction, feature selection, and feature extraction. By feature selection approaches, a new feature set W0 is obtained, which is a subset of the original feature set W. Then W0 is used as inputs for classification tasks. Information Gain (IG) is frequently employed in the feature selection approach. 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 before 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.

2(a).Proposed Method:

There are some drawbacks to the existing methods. First up all the user need to specify the number of clusters in advance. Second when calculating the similarities the variance of the underlying cluster are not considered. Third all words in a cluster have the same degree of contribution to the resulting extracted feature. Our fuzzy incremental feature clustering algorithm is proposed to deal with these issues.

Suppose we are given a document set D of n documents d1,d2…dn together with a feature vector W of m words w1,w2…wn, and p classes c1,c2…cp. We then construct one word pattern for each word in W. For word wi, its word pattern xi is defined as

$$\mathbf{x}_i = <x_{i1}, x_{i2}, \ldots, x_{ip}> = <P(c_1|w_i), P(c_2|w_i), \ldots, P(c_p|w_i)>$$

Where

$$P(c_j|w_i) = \frac{\sum_{q=1}^{n} d_{qi} \times \delta_{qj}}{\sum_{q=1}^{n} d_{qi}},$$

For i≤j≤p. Here dqi indicates the number of occurrences of wi in document dq

Also

$$\delta_{qj} = \begin{cases} 1, & \text{if document } \mathbf{d}_q \text{ belongs to class } c_j, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore we have m word patterns in total. It is these word patterns, our clustering algorithm will work on. Our goal is to group the words in W into clusters, based on these word patterns. A cluster contains a certain number of word patterns, and is characterized by the product of P one-dimensional Gaussian functions. Gaussian functions[7][8] are adopted because of their superiority over other functions in performance. Let G be a cluster containing q word patterns Xj=<xj1,xj2…xjp> 1≤j≤p. Then the mean is defined as m=<m1,m2…mp> and the deviation σ =< σ1, σ2… σp> of G are defined

$$m_i = \frac{\sum_{j=1}^{q} x_{ji}}{|G|}$$

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{q} (x_{ji} - m_{ji})^2}{|G|}}$$

For i≤j≤p, where |G| denotes the size of G.the fuzzy similarity of a word pattern X=<x1,x2…xp> to cluster G is defined by the following membership function

$$\mu_G(\mathbf{x}) = \prod_{i=1}^{p} \exp\left[-\left(\frac{x_i - m_i}{\sigma_i}\right)^2\right].$$

Where 0≤μG≤1.

2(b).Fuzzy based Incremental Feature Clustering:

Our clustering algorithm is an incremental, self constructing algorithm. Word patters are considered one by one. No clusters exist the beginning, and clusters are created can be created if necessary. For each word pattern, the similarity of this word pattern to each existing cluster is calculated to decide whether it is combined into an existing cluster or a new cluster is created. Once a new cluster is created, the corresponding membership function should be initialized. On the contrary, when the word pattern is combined into an existing cluster, the membership function of that cluster should be updated accordingly.

Let k be the number of currently existing clusters. The clusters are G1,G2…Gk respectively. Each cluster Gj have the mean mj=<mj1,mj2…mjp> and deviation σj=< σj1, σj2… σjp>. Let Sj be the size of the cluster Gj. Initially we have k=0.so no clusters exist at the beginning. For each word pattern xi=<xi1,xi2…xip>, 1≤i≤m,we calculate the similarity of xi to each existing clusters as

$$\mu_{G_j}(\mathbf{x}_i) = \prod_{q=1}^{p} \exp\left[-\left(\frac{x_{iq} - m_{jq}}{\sigma_{jq}}\right)^2\right]$$

For 1≤j≤k, we sat that xi passes the similarity test on cluster Gj, if

$$\mu_{G_j}(\mathbf{x}_i) \geq \rho,$$

Where ρ, o≤ρ≤1 is a predefined threshold. If the user intends to have larger clusters, then he/she can give a smaller threshold. Otherwise, a bigger threshold can be given. As the threshold increases, the number of clusters also increases. Two cases may occur. First, there are no existing fuzzy clusters on which Xi has passed the similarity test. For this case, we assume that xi is not similar enough to any existing cluster and a new cluster Gh, h=k+1, is created with

$$\mathbf{m}_h = \mathbf{x}_i, \quad \boldsymbol{\sigma}_h = \boldsymbol{\sigma}_0,$$

Where σ0 is a user defined constant vector. The new vector Gh contains only one member i.e., the word pattern xi at this point, since it contains only one member the deviation of a cluster is zero. We cannot use deviation zero in calculating fuzzy similarities. Hence we initialize the deviation of a newly created cluster by and the number of clusters in increased by 1 and the size of cluster Gh, Sh should be initialized i.e.,

$$k = k + 1, \quad S_h = 1.$$

Second, if there are existing clusters on which xi has passed the similarity test, let cluster Gt be the cluster with the largest membership degree, i.e.,

$$t = arg \max_{1 \le j \le k} (\mu_{G_j}(\mathbf{x}_i)).$$

In this case the modification of cluster Gt is describes as follows

$$m_{tj} = \frac{S_t \times m_{tj} + x_{ij}}{S_t + 1},$$

$$\sigma_{tj} = \sqrt{A - B} + \sigma_0,$$

$$A = \frac{(S_t - 1)(\sigma_{tj} - \sigma_0)^2 + S_t \times {m_{tj}}^2 + {x_{ij}}^2}{S_t},$$

$$B = \frac{S_t + 1}{S_t} \left(\frac{S_t \times m_{tj} + x_{ij}}{S_t + 1} \right)^2,$$

for $1 \le j \le p$, and

$$S_t = S_t + 1.$$

We discuss briefly here the computational cost of our method and compare it with DC[9] , IOC[10] , and IG[11]. For an input pattern, we have to calculate the similarity between the input pattern and every existing cluster. Each pattern consists of p components where p is the number of classes in the document set. Therefore, in worst case, the time complexity of our method is O(mkp) where m is the number of original features and k is the number of clusters finally obtained. For DC, the complexity is O(mkp) where t is the number of iterations to be done. The complexity of IG is O(mp+mlogm) and the complexity of IOC is O(mkpn) where n is the number of documents involved. Apparently, IG is the quickest one. Our method is better than DC and IOC.

3. Feature extraction using Weighting Matrix:

Feature Extraction can be expressed in the following form:

$$\mathbf{D}' = \mathbf{D}\mathbf{T},$$

Where

$$\mathbf{D} = [\mathbf{d}_1 \ \mathbf{d}_2 \ \cdots \ \mathbf{d}_n]^T,$$

$$\mathbf{D}' = [\mathbf{d}'_1 \ \mathbf{d}'_2 \ \cdots \ \mathbf{d}'_n]^T,$$

$$\mathbf{T} = \begin{bmatrix} t_{11} & \cdots & t_{1k} \\ t_{21} & \cdots & t_{2k} \\ \vdots & \ddots & \vdots \\ t_{m1} & \cdots & t_{mk} \end{bmatrix},$$

With

$$\mathbf{d}_i = [d_{i1} \ d_{i2} \ \cdots \ d_{im}],$$

$$\mathbf{d}'_i = [d'_{i1} \ d'_{i2} \ \cdots \ d'_{ik}],$$

For $1 \le i \le n$. Where T is a weighting matrix. The goal of feature reduction is achieved by finding appropriate T such that k is smaller than m. In the divisive information theoretic feature clustering algorithm the elements of T are binary and can be defined as follows:

$$t_{ij} = \begin{cases} 1, & \text{if } w_i \in \mathbf{W}_j, \\ 0, & \text{otherwise,} \end{cases}$$

By applying our feature clustering algorithm word patterns have been grouped into clusters, and words in the feature vector W are also clustered accordingly. For one cluster, we have one extracted feature. Since we have k clusters, we have k extracted features. The elements of T are derived based on the obtained clusters, and feature extraction will be done. We propose three weighting approaches: hard, soft, and mixed. In the hard-weighting approach, each word is only allowed to belong to a cluster, and so it only contributes to a new extracted feature. In this case, the elements of T are defined as follows:

$$t_{ij} = \begin{cases} 1, & \text{if } j = arg\max_{1\le\alpha\le k}(\mu_{G_\alpha}(\mathbf{x}_i)), \\ 0, & \text{otherwise.} \end{cases}$$

In the soft-weighting approach, each word is allowed to contribute to all new extracted features, with the degrees depending on the values of the membership functions. The elements of T are defined as follows:

$$t_{ij} = \mu_{G_j}(\mathbf{x}_i).$$

The mixed-weighting approach is a combination of the hard-weighting approach and the soft-weighting approach. In this case, the elements of T are defined as follows:

$$t_{ij} = (\gamma)\times t_{ij}^{H} + (1-\gamma)\times t_{ij,}^{S}$$

By selecting the value of γ, we provide flexibility to the user. When the similarity threshold is small, the number of clusters is small, and each cluster covers more training patterns. In this case, a smaller γ will favor soft-weighting and get a higher accuracy. When the similarity threshold is large, the number of clusters is large, and each cluster covers fewer training patterns which get a higher accuracy.

4. Classification Of Text Data::

Given a set D of training documents, text classification can be done as follows: We specify the similarity threshold ρ, and apply our clustering algorithm. Assume that k clusters are obtained for the words in the feature vector W. Then we find the weighting matrix T and convert D to D' . Using D' as training data, a text classifier based on support vector machines (SVM) is built. SVM is a kernel method, which finds the maximum margin hyperplane in feature space separating the images of the training patterns into two groups[12][13]. A slack variables ξi are introduced to account for misclassifications. The objective function and constraints of the classification problem can be formulated as:

$$\min_{w,b}\frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{l}\xi_i$$

$$s.t. y_i\left(\mathrm{w}^T\phi(\mathrm{x}_i)+b\right) \ge 1-\xi_i, \xi_i \ge 0, i = 1,2,\ldots,l,$$

Where *l* is the number of training patterns, C is a parameter, which gives a tradeoff between maximum margin and classification error, and yi, being +1 or -1, is the target label of pattern xi. Ø:X → F is a mapping from the input space to the feature space F, where patterns are more easily separated, and $W^T\emptyset(X_i) + b = 0$ is the hyper plane to be derived with w, and b being weight vector and offset, respectively.

We follow the idea to construct an SVM-based classifier. Suppose, d is an unknown document. We first convert d to d0 by

$$\mathbf{d}' = \mathbf{d}\mathbf{T}.$$

Then we feed d□ to the classifier. We get p values, one from each SVM. Then d belongs to those classes with 1, appearing at the outputs of their corresponding SVMs.

5. Conclusions:

We have presented a fuzzy based incremental feature clustering (FIFC) algorithm, which is an incremental clustering approach to reduce the dimensionality of the features classification of text. Feature that are similar to each other are placed in the same cluster. New clusters formed automatically, if a word is not similar to any existing cluster. Each cluster so formed is characterized by a membership function with statistical mean and deviation. By our work the derived membership functions match closely with the real distribution of the training data. We reduce the burden on the user in specifying the number of extracted features in advance. Experiments results shows that our method can run faster and obtain better extracted features methods.

6. References:

[1]Y.Yang and J.O.Pedersen, "A Comparative Study on Feature Selection in Text Categorization," Proc. 14th Int'l Conf. Machine Learning, pp. 412-420, 1997.

[2]D.D.Lewis, "Feature Selection and Feature Extraction for Text Categorization," Proc. Workshop Speech and Natural Language, pp. 212-217, 1992.

[3]H.Li,T.Jiang, and K.Zang, "Efficient and Robust Feature Extraction by Maximum Margin Criterion," T.Sebastian, S.Lawrence, and S. Bernhard eds. Advances in Neural Information Processing System, pp. 97-104, Springer, 2004.
.

[4]L.D.Baker and A.McCallum, "Distributional Clustering of Words for Text Classification," Proc. ACM SIGIR, pp. 96-103, 1998.

[5]L.D.Baker and A.McCallum, "Distributional Clustering of Words for Text Classification," Proc. ACM SIGIR, pp. 96-103, 1998.

[6]R.Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "Distributional Word Clusters versus Words for Text Categorization," J. Machine Learning Research, vol. 3, pp. 1183-1208, 2003.

[7J.Yen and RLangari, Fuzzy Logic-Intelligence, Control, and Information. Prentice-Hall, 1999.

[8]J.S.Wang and C.S.G.Lee, "Self-Adaptive Neurofuzzy Inference Systems for Classification Applications," IEEE Trans. Fuzzy Systems, vol. 10, no. 6, pp. 790-802, Dec. 2002.

[9]I.S. Dhillon, S. Mallela, and R. Kumar, "A Divisive Infomation- Theoretic Feature Clustering Algorithm for Text Classification," J. Machine Learning Research, vol. 3, pp. 1265-1287, 2003.

[10]J. Yan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi, and Z. Chen, "Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data Preprocessing," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 3, pp. 320-333, Mar. 2006.

[11]Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," Proc. 14th Int'l Conf. Machine Learning, pp. 412-420, 1997.

[12]B.Scho¨lkopf and A.J.Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2001.

[13]J.Shawe-Taylor and N.Cristianini, Kernel Methods for Pattern Analysis. Cambridge Univ. Press, 2004.

7. About the Authors:

Anil Kumar Reddy Tetali is currently pursuing his M.Tech in Computer Science and Engineering at BVC Engineering College Odalarevu.

B P N Madhu Kumar is currently working as an Associate Professor in Computer Science and Engineering department, BVC Engineering College Odalarevu. His research interests include data mining, web mining.

K.Chandra Kumar is currently working as an Associate Professor in Computer Science and Engineering Department,VSL Engineering College Kakinada. His research interests include data mining and text mining.