Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Audio 120 kbps Classification Based on Feature Clustering Algorithm

J Purna Prakash[1], K.M.V Ramana[2], M. Rama Raju[3]

[1,3]*Computer Science & Engineering, Christu Jyoti Institute of Technology & Science*
[2]*Department of Mathematics, Christu Jyoti Institute of Technology & Science*

Abstract—Feature clustering is a powerful method to reduce the dimensionality of feature vectors for audio 120 kbps classifications. In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The sounds in the feature vector of an audio set a regrouped into clusters, based on 120 kbps similarity test. Sounds that are similar to each other are grouped into the same cluster. Set tempo to the sound 120 kbps for more accuracy results and reliability .When all the sounds 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 sounds 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 feature scan then is avoided.

Index Terms—feature clustering, feature reduction, audio classifications.

I. INTRODUCTION

In text classification, the dimensionality of the feature vector is usually huge. For example, 20 different sounds 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 [3],[4].To alleviate this difficulty, feature reduction approaches are applied before file classification tasks are performed [5].Two major approaches, features selection [6],[7],[8],[9],[10]and feature extraction [11],[12],[13],have been proposed for feature reduction. In general, feature extraction approaches are more effective than feature selection techniques, but are more computationally expensive [11], [12], [14]. Therefore, developing scalable and efficient feature extraction algorithms is highly demanded for dealing with high-dimensional audio data sets.

Classical feature extraction methods aim to convert the representation of the original high-dimensional audio set into a lower-dimensional data set by a projecting process through algebraic transformations For example, Principal Component Analysis [15], Linear Discriminate Analysis [16], Maximum Margin Criterion [12], and Orthogonal Centroid algorithm [17] perform the projection by linear transformations, while Locally Linear Embedding [18], ISOMAP [19], and Laplacian Eigenmaps[20] 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 [14],[21],[22],[23] have been proposed to improve the Computational complexity. However, the complexity of these approaches is still high. Feature clustering [24], [25],[26],[27],[28],[29] is one of effective techniques for feature reduction in audio 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[24],which was derived from the “distributional clustering” idea of Pereira et al.[30].Al-Mubaid and Umair [31]used distributional clustering to generate an efficient representation of files and applied a learning logic approach for training audio classifiers. The Agglomerative Information Bottle neck approach was proposed by Tishbyetal.[25],[29].In these feature clustering methods, each new feature is generated by combining a subset of the original sounds. A sound is exactly assigned to a subset, i.e. hard-clustering, based on the similarity magnitudes between the sound 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.

We propose a similarity-based self-constructing feature clustering algorithm, which is an incremental feature Clustering approach to reduce the number of features for the audio classification task. The sounds in the feature vector of a File set are represented as distributions, and processed one after another. Sounds 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 sound is not similar to any existing cluster, a new cluster is created for this sound. Similarity between a sound and a cluster is defined by considering both the mean and the variance of the cluster. When all the sounds 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 sounds 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. 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 real-world audio sets show that our method can run faster and obtain better extracted features than other methods.

The remainder of this paper is organized as follows: Section2 gives a brief back ground about feature reduction. Presents the proposed similarity-based self-constructing feature clustering algorithm.

II. BACKGROUND AND RELATED WORK

To process files, the bag-of-sounds model [32], [33] is commonly used. Let F = {f1, f2,…..,fn} file set of n files, where f1,f2,…..,fn are individual file sets, and each file set is belongs to one of the classes in the set { c1,c2,………..,cp}. If a file set belongs to two or more classes, then two or more copies of the file with different classes are included in F. Let the sound set S={s1,s2,……..,sm} be the feature vector of the file set. Each file fi, $1 \leq i \leq n$, is represented as fi=<fi1, fi2,……,fim},where each fij denotes the number of occurrences ofSj in the ith file set. The feature reduction task is to find a new sound set S’ = {s’1, s’2,…,s’k} , k<<m, such that S and S’ work equally well for all the desired properties with D. After feature reduction, each file di is converted into a new representation f1’ = < f’i1, f’i2,…,f’ik> and the converted file set is F’ = <{f’1, f’2,…,f’n} . If k is much smaller than m, computation cost with subsequent operations on D’ Can be drastically reduced.

A. Feature Reduction

In general, here are two ways of doing feature reduction, feature selection, and feature extraction. By feature selection approaches, a new feature set S’ = { S’1, S’2,…,S’k} is obtained, which is a subset of the original feature set S. Then S’ is used as inputs for classification tasks. Information Gain (IG) is frequently employed in the feature selection approach [10]. It measures the reduced uncertainty by an information-theoretic measure and gives each sound a weight. The weight of sounds is calculated as follows:

$$IG\,(w'_j) = -\sum_{l=1}^{p} P(cl)\log P(cl)$$

$$+\,P(S_j)\sum_{l=1}^{p} P(Cl\,|\,Sj)\log P(cl\,|\,Wj)$$

$$+P(w_j)\sum_{l=1}^{p} P(Cl\,|\,Sj)\log P(Cl\,|\,Wj).(1)$$

where $P(c_1)$ denotes the prior probability for class c_1 , $P(s_j)$ denotes the prior probability for feature s_j ,$p(s_j)$ is identical to $1\text{-}p(s_j)$, and $p(c_1\backslash s_j)$ and $p(c_1\backslash s_j)$ denote the probability for class c_1 with the presence and absence, respectively, of s_j. The sounds of top k weights in w are selected as the feature in S'.

In feature extraction approaches, extracted features are obtained by a projecting process through algebraic trans-formations. An incremental orthogonal centroid (IOC) algorithm was proposed in[14]. Let a corpus of files be represented as an m×n matrix $X \in R^{m\times n}$, where m is the number of features in the feature set and n is the number of files in the file set. IOC tries to findan optimal transformation matrix $F \in R^{m\times k}$, where k is the desired number of extracted features, according to the following criterion:

$$F^* = \arg\max \operatorname{trace}(F^T S_b F), \qquad (2)$$

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

$$S_b=\sum_{q=1}^{p} P(c_q)(M_q\text{-}M_{all})(M_q\text{-}M_{all})^T \quad (3)$$

With p(cq) being the prior probability for a pattern belonging to class Cq, Mqbeing the mean vector of class Cq, and Mall being the mean vector of all patterns.

B. Feature Clustering

Feature clustering is an efficient approach for feature reduction [25], [29], which groups all features into some clusters, where features in a cluster are similar to each other. The feature clustering methods proposed in[24], [25], [27], [29] are "hard" clustering methods, where each sound of the original features belongs to exactly one sound cluster. Therefore each sound contributes to the synthesis of only one new feature .Each new feature is obtained by summing up the sounds belonging to one cluster. Let D be the matrix consisting of all the original files with m features and D' be the matrix consisting of the converted files With new k features. The new feature set W' = { w1',w2',…wk'} correspondstoa partition {W1,W2,...,Wk} of the originalfeatureset W,i.e., $Wt \cap Wq = \emptyset$, where $1 \le q$, $t \le k$ and $t \neq q$. Notethataclustercorrespondstoanelement in thepartition.Then,the tth featurevalueofthe converted document d' iscalculatedasfollows:

$$d'_{it} = \sum_{Wj \in Wt} d_{ij} \ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Which is a linear sum of the feature values in Wt. The divisive information-theoretic feature clustering(DC)algorithm, proposed by Dhillonetal. [27] calculates the distributions of sounds over classes, $P(C/Wj)$ $1 \le j \le m$, Where C = { c1,c2,….cp}, and uses Kullback-Leibler di- vergence to measure the dissimilarity between two distributions is calculated. The distribution of a cluster Wt as follows:

$$P(C \mid W_t) = \sum_{wj \in Wt}^{n} \frac{P(Wj)}{\sum wj \in Wt\, P(Wj)} P(C \mid w_j) \ldots\ldots\ldots.(5)$$

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

$$\sum_{t=1}^{k} \sum_{Wj \in Wt} P(w_j) KL(P(C \mid w_j), P(C \mid W_t)) \ldots\ldots.(6)$$

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

III. OUR METHOD

There are some issues pertinent to most of the existing feature clustering methods. First, the parameter k, indicating the desired number of extracted features, has to be specified in advance. This gives a burden to the user, since trial-and-error has to be done until the appropriate number of extracted features is found. Second, when calculating similarities, the variance of the underlying cluster is not considered. Intuitively, the distribution of the data in a cluster is an important factor in the calculation of similarity. Third, all sounds in a cluster have the same degree of Contribution to the resulting extracted feature. Sometimes, it May be better if more similar sounds are allowed to have Bigger degrees of contribution. Our feature clustering Algorithm is proposed to deal with these issues.

Suppose, we are given a file set F of n file set f1,f2,…..,fn, together with the feature vector w of m sounds s1,s2,…sm and p classes c1,c2,…..,cp, as specified in section 2. We construct one sound pattern for each sound in w. For sound si its sound pattern xi is defined, similarly as in [27], by

$$X_i = \langle x_{i1}, x_{i2} \ldots\ldots. x_{ip} \rangle$$

$$= \langle P(c_1 \mid w_i), P(c_2 \mid w_i), \ldots.., P(c_p \mid w_i) \rangle \ldots\ldots\ldots\ldots\ldots\ldots(7)$$

Where

$$P(c_j \mid w_i) = \frac{\sum_{q=1}^{n} dqi \times \delta qi}{\sum_{q=1}^{n} dqi} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

for 1 <=j<=p. Note that fqi indicates the number of fi in file fq ,as described in Section2. Also, fqi is defined as

$$\delta_{qj} = \begin{cases} 1, & \text{if file fq belongs to class cj} \\ & 0, otherwise \end{cases}, \ldots\ldots\ldots\ldots\ldots\ldots\ldots.(9)$$

Therefore, we have m sound patterns in total. For example, suppose we have four files f1, f2, f3, and f4 belonging these to c1, c1, c2 and c2, respectively. Let the occurrences of w1 in these files be 1,2,3, and ,4, respectively. Then, the sound pattern x1 of S1 is:

$$P(c1|w1) = \frac{1 \times 1 + 2 \times 1 + 3 \times 1 + 4 \times 1}{1 + 2 + 3 + 4} = 0.3$$

$$P(c2|w1) = \frac{1 \times 0 + 2 \times 0 + 3 \times 1 + 4 \times 1}{1 + 2 + 3 + 4} = 0.7$$

$X_1 = <0.3, 0.7>$. …………………..(10)

It is these sound patterns, our clustering algorithm will work on. Our goal is to group the sounds in S into clusters, based on these sound patterns. A cluster contains a certain number of sound patterns, and is characterized by the product of p one-dimensional Gaussian functions. Gaussian functions Are adopted because of their superiority over other functions in performance [34], [35].Let G be a cluster, containing q sound patterns x1,x2…xq

Let $<x_j=<x_{j1},x_{j2}\ldots x_{jp}>, 1\leq j\leq q$ Then the mean $m =<m_1,m_2\ldots m_p>$and the deviation $\sigma = <\sigma_1, \sigma_2, \ldots \ldots \sigma_p>$of Gare Defined as

$$mi = \frac{\sum_{l=1}^{p} x\,ji}{|G|}\ldots\ldots(11)$$

$$\sigma i= \sqrt{\frac{\sum_{j=1}^{q}(x\,ji - mji)2}{|G|}}\ldots\ldots(12)$$

for 1≤i≤ p, where │G│ denotes the size of G, i.e., the number of sound patterns contained in G. The fuzzy similarity of a sound pattern X=<x1,x2,…xp> to cluster G is defined by the following membership function:

$$\mu_G(x) = \pi \prod_{i=1}^{p} exp\left[-\left(\frac{xi\text{-}mi}{\sigma i}\right)\right]\ldots\ldots(13)$$

Notice that 0<= µG (x) =1.A sound pattern close to the mean of a cluster is regarded to be very similar to this cluster, i.e., µG (x)≈ 1.On the contrary, a sound pattern far distant from. For a cluster is hardly similar to this cluster, i.e., µG (x)≈ 0.

$$\mu_G(x) = \exp\left[-\left(\frac{0.3\text{-}0.4}{0.2}\right)\right] \times exp\left[-\left(\frac{0.7\text{-}0.3}{0.3}\right)\right]$$

$$= 0.7788\times0.8348 = 0.6969\ldots\ldots(14)$$

A. Self-Constructing Clustering

Our clustering algorithm is an incremental, self-constructing learning approach. sound patterns are considered one by one. The user does not need to have any idea about the number of clusters in advance. No clusters exist at the beginning, and clusters can be created if necessary. For each sound pattern, the similarity of this sound 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 sound pattern is combined into an existing cluster, the membership function of that cluster should be updated accordingly.The similarity of xi clusters, i.e.,

$$\mu_{Gj}(xi) = \prod_{i=1}^{p} exp\left[-\left(\frac{xiq\text{-}mjq}{\sigma jq}\right)^2\right]\ldots\ldots(15)$$

for 1≤ j ≤k. We say that x passes the similarity test on I if cluster Gj if

$$\mu Gj(Xi) \geq \rho,\ldots\ldots(16)$$

B. Audio 120kbps Classification

Given a set D of training files, Audio classification can be done as follows: We specify the similarity threshold ρfor (16), and apply our clustering algorithm. Assume that k clusters are obtained for the sounds in the feature vector S. Then we find the weighting matrix T and convert D' By (25).Using D as training data, a classifier based on Support vector machines (SVM) is built .Note that any Classifying technique other than SVM can be applied. Joachims [36] showed that SVM is better than other Methods for audio categorization. SVM is a kernel method,

TABLE 1 A Simple File Set F

	Network (S_1)	Java (S_2)	Datamining (S_3)	Data Structures (S_4)	C (S_5)	Security (S_6)	DBMS (S_7)	Internet (S_8)	Modeling (S_9)	Micro (S_{10})	class
f1	0	1	0	0	1	1	0	0	0	1	C1
f2	0	0	0	0	0	2	1	1	0	0	C1
f3	0	0	0	0	0	0	1	0	0	0	C1
f4	0	0	1	0	2	1	2	1	0	1	C1
f5	0	0	0	1	0	1	0	0	1	0	C2
f6	2	1	1	0	0	1	0	0	1	0	C2
f7	3	2	1	3	0	1	0	1	1	0	C2
f8	1	0	1	1	0	1	0	0	0	0	C2
f9	1	1	1	1	0	0	0	0	0	0	C2

Which finds the maximum margin hyper plane in feature Space separating the images of the training patterns into Two groups [37],[38],[39].To make the method more Flexible and robust, some patterns need not be correctly Classified by the hyper plane, but the misclassified patterns are should be penalized. Therefore, slack variables $\varepsilon$are introduced to account for misclassifications. The objective function and constraints of the classification problem can be formulated as:

$$\min \frac{1}{2} W^T W + C \sum_{i=1}^{l} \varepsilon_i$$

$$\text{s.t.} y_i \ (WT\emptyset\ (x_i) + b\ ) \geq 1 - \varepsilon i,$$

$$\varepsilon i \geq 0, i = 1,2, \ldots\ldots, l, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(17)$$

where l is the number of training patterns, C is a parameter, which gives a tradeoff between maximum margin and classification error, and y_i, being+1or-1,is target label of pattern x_i .Note that 0: 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 off set, respectively.

An SVM described above can only separate a part two classes and y_i=+1 and $y_i = -1$.We follow the idea in[36]to construct an SVM-based classifier. For p classes, we create p, SVMs, one SVM for each class. For the SVM of class c_v are $1 \leq v \leq p$ treated as, having $y_i = +1$, and the training patterns of the other classes are treated as having. $yi = +1$,and the training patterns of the other having yi $= -$ 1.The classifier is then Classes are treated as having yi the aggregation of these SVMs. Now we are ready for classifying unknown files. Suppose, fis an unknown 0 by file. First we convert f to f' by

f' = f T ……………………………………….(18)

IV. AN EXAMPLE

We give an example here to illustrate how our method works. Let D be a simple file set, containing 9 file f1,f2,….fy of two classes c1 and c2 with 10 sounds "Network," "java,"….." ,"Micro" in the feature vector S, as shown in Tables1. For simplicity, we denote the ten sounds as s1,s2,…..s10,respectively. We should considers sound pattern file 120 kilo bytes per seconds per each sound pattern than we are getting more reliable and accuracy of results are out.

We calculate the ten sounds patterns x1,x2,…x10 according to (7) and (8). For example, x6 = <P(c1 | S6),P(c2 | S6)> and P(c2 | w6) is calculated by (35) here in data 1 is indicating in certain file the sound is available . else 0 indicating no data is available. hence following example.

$P(c_2 | w_6) = 1\times0+2\times0+0\times0+1\times0+0\times1+1\times1$

$+1\times1+1\times1+1\times1+0\times1/1+2+0+1+1+1$

$+1+1+0=0.50$ ………………………………………….(19)

The resulting sound patterns are shown in Table 2.

Table 2 Sound patters of S

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0.00	0.20	0.20	0.00	1.00	0.50	1.00	0.67	0.00	1.00
1.00	0.80	0.80	1.00	0.00	0.50	0.00	0.33	1.00	0.00

Note that each sound pattern is a two dimensional vector, since there are two classes involved in F

As we apply same method to all sound pattern we can get these results towards sounds their respective for $x_i = <x_1,\ldots x_m>$ for we taken the example to x6 o.50 and remaining 0.50 as we taken.

V. FUTURE WORK

In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The sounds in the feature vector of an audio set a regrouped into clusters, based on 120 kbps similarity test. Sounds that are similar to each other are grouped into the same cluster. Set tempo to the sound 120 kbps for more accuracy results and reliability .When all the sounds have been fed in, a desired Number of clusters are formed automatically. Further we can apply data sets on different data sets as 360 kbps and so on, in future we can also apply video in different ways as mpeg4,mpeg5, etc., further we can also apply on hd ready and full hd formats.

VI. ACKNOWLEDGEMENTS

I thank my students who have implemented the extension part of this paper under my guidance Also, I would like to thank the management (Rev. Fr. Y. PapiReddy , the director), Principal (Dr.J.B.V.Subrahmanyam) for providing this opportunity and for their constant encouragement and also to Assoc.Professor A. Poorna chander reddy who motivate and suggest few concepts towards publishing the paper. I also thank CSE Staff Srividya Pajjuri and T. Prakash helping to designing the Pictures and hardware for technical support in this paper.

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,andH.Park,"Dimension Reduction in Text Classification with Support Vector Machines," J.MachineLearning Research, vol.6,pp.37-53,2005.
[4] F.Sebastiani,"MachineLearninginAutomatedTextCategorization," ACMComputingSurveys, vol.34,no.1,pp.1-47,2002.
[5] B.Y.RicardoandR.N.Berthier, Modern Information Retrieval.Addison Wesley Longman,1999.
[6] A.L.BlumandP.Langley,"SelectionofRelevantFeaturesand Examples in MachineLearning," AritficialIntelligence, vol.97,nos.1/2,pp.245-271,1997.
[7] E.F.Combarro,E.Montanes,I.Diaz,J.Ranilla,andR.Mones,"Introducinga Family of Linear Measures for Feature Selection in TextCategorization,"IEEETrans .Knowledge and DataEng.,vol.17,no.9,pp.1223-1232,Sept.2005.
[8] K.DaphneandM.Sahami,"TowardOptimalFeatureSelection,"Proc.13thInt'lConf.MachineLearning, pp.284-292,1996.
[9] R.KohaviandG.John,"WrappersforFeatureSubsetSelection,"AritficialIntelligence, vol.97,no.1-2,pp.273-324,1997.
[10] Y.YangandJ.O.Pedersen,"AComparativeStudyonFeatureSelectioninTextCategorization,"Proc.14th Int'l Conf. Machine Learning, pp.412-420,1997.
[11] D.D.Lewis,"FeatureSelectionandFeatureExtractionforTextCategorization," Proc.WorkshopSpeechandNatural Language,pp.212-217,1992.
[12] H.Li,T.Jiang,andK.Zang,"EfficientandRobustFeatureExtractionbyMaximumMarginCriterion,"T.Sebastian,S.Lawrence,andS.Bernhardeds. Advance sin Neural Information Processing System, pp.97-104,Springer,2004.
[13] E.Oja, Subspace Methods of Pattern Recognition. Research Studies Press,1983.
[14] J.Yan,B.Zhang,N.Liu,S.Yan,Q.Cheng,W.Fan,Q.Yang,W.Xi,andZ.Chen,"EffectiveandEfficientDimensionalityReductionforLargeScaleandStreamingDataPreprocessing,"IEEETrans.KnowledgeandDataEng., vol.18,no.3,pp.320-333,Mar.2006.
[15] I.T.Jolliffe, PrincipalComponentAnalysis. Springer-Verlag,1986.
[16] A.M.MartinezandA.C.Kak,"PCAversusLDA," IEEETrans.PatternAnalysisandMachineIntelligence, vol.23,no.2pp.228-233,Feb.2001.
[17] H.Park,M.Jeon,andJ.Rosen,"LowerDimensionalRepresentationofTextDataBasedonCentroidsandLeastSquares," BIT NumericalMath, vol.43,pp.427-448,2003.
[18] S.T.RoweisandL.K.Saul,"NonlinearDimensionalityReductionbyLocallyLinearEmbedding," Science, vol.290,pp.2323-2326,2000.
[19] .J.B.Tenenbaum,V.deSilva,andJ.C.Langford,"AGlobalGeometricFrameworkforNonlinearDimensionalityReduction," Science, vol.290,pp.2319-2323,2000.
[20] M.BelkinandP.Niyogi,"LaplacianEigenmapsandSpectralTechniquesforEmbeddingandClustering,"AdvancesinNeural InformationProcessingSystems,vol.14,pp.585-591,TheMITPress2002.
[21] K.Hiraoka,K.Hidai,M.Hamahira,H.Mizoguchi,T.Mishima,andS.Yoshizawa,"SuccessiveLearningofLinearDiscriminantAnalysis:Sanger-TypeAlgorithm,"Proc.IEEECSInt'lConf.PatternRecognition, pp.2664-2667,2000.
[22] J.Weng,Y.Zhang,andW.S.Hwang,"CandidCovarianceFreeIncrementalPrincipalComponentAnalysis,"IEEETrans.PatternAnalysisandMachineIntelligence,vol.25,no.8,pp.1034-1040,Aug.2003.
[23] J.Yan,B.Y.Zhang,S.C.Yan,Z.Chen,W.G.Fan,Q.Yang,W.Y.Ma,andQ.S.Cheng,"IMMC:IncrementalMaximumMargin Criterion," Proc.10thACMSIGKDD, pp.725-730,2004.
[24] L.D.BakerandA.McCallum,"DistributionalClusteringofWordsforTextClassification," Proc.ACMSIGIR, pp.96-103,1998.
[25] R.Bekkerman,R.ElYaniv,N.Tishby,andY.Winter,"DistributionalWordClustersversusWordsforTextCategorization,"J. MachineLearningResearch, vol.3,pp.1183-1208,2003.
[26] M.C.DalmauandO.W.M.Florez,"ExperimentalResultsoftheSignalProcessingApproachtoDistributionalClusteringofTermsonReuters-21578Collection,"Proc.29thEuropeanConf.IRResearch,pp.678-681,2007.
[27] I.S.Dhillon,S.Mallela,andR.Kumar,"ADivisiveInfomationTheoreticFeatureClusteringAlgorithmforTextClassification,"J.MachineLearningResearch, vol.3,pp.1265-1287,2003.
[28] D.IencoandR.Meo,"ExplorationandReductionoftheFeatureSpacebyHierarchicalClustering," Proc.SIAMConf.DataMining,pp.577-587,2008.
[29] N.SlonimandN.Tishby,"ThePowerofWordClustersforTextClassification," Proc.23rdEuropeanColloquiumonInformationRetrievalResearch(ECIR), 2001.
[30] F.Pereira,N.Tishby,andL.Lee,"DistributionalClusteringofEnglishWords," Proc.31stAnn.MeetingofACL, pp.183-190,1993.
[31] H.AlMubaidandS.A.Umair,"ANewTextCategorizationTechniqueUsingDistributionalClusteringandLearningLogic,"IEEETrans.KnowledgeandDataEng., vol.18,no.9,pp.1156-1165,Sept.2006.
[32] G.SaltonandM.J.McGill,IntroductiontoModernRetrieval.McGraw-HillBookCompany,1983.
[33] T.Joachims,"AProbabilisticAnalysisoftheRocchioAlgorithmwithTFIDFforTextCategorization," Proc.14thInt'lConf.MachineLearning, pp.143-151,1997.

[34] .J.YenandR.Langari, FuzzyLogic-Intelligence,Control,andInformation. Prentice-Hall,1999.
[35] J.S.WangandC.S.G.Lee,"Self-AdaptiveNeurofuzzyInferenceSystemsforClassificationApplications," IEEETrans.FuzzySystems, vol.10,no.6,pp.790-802,Dec.2002.
[36] T.Joachims,"TextCategorizationwithSupportVectorMachine:LearningwithManyRelevantFeatures,"TechnicalReport LS823,Univ.ofDortmund,1998.
[37] C.CortesandV.Vapnik,"Support-VectorNetwork," MachineLearning, vol.20,no.3,pp.273-297,1995.
[38] Learning with Kernels: Support Vector B.Scholkopf and A.J.Smola, Machines, Regularization, Optimization, and Beyond. MITPress,2001.
[39] J.Shawe-TaylorandN.Cristianini, KernelMethodsforPatternAnalysis. CambridgeUniv.Press,2004.
[40] D.D.Lewis,Y.Yang,T.Rose,andF.Li,"RCV1:ANewBenchmarkCollectionforText CategorizationResearch,"J.MachineLearningResearch,vol.5,pp.361397,http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf,2004