A Novel Codebook Representation Method and Encoding Strategy For Bag-of-Words Based Acoustic Event Classification

Jia Dai* Chongjia Ni† Wei Xue* and Wenju Liu*
* National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China
E-mail: {jia.dai, wxue, lwj}@nlpr.ia.ac.cn
† School of Mathematic and Quantitative Economics, Shandong University of Finance and Economics, China
E-mail: cjni_sd@sdufe.edu.cn

***Abstract*—The bag-of-words (BoW) model has been widely used for acoustic event classification (AEC). The performance of the BoW based AEC model is much influenced by "codebook construction" and "histogram generation". The common approaches for constructing the codebook and generating the histogram are the K-means and vector quantization encoding (VQE) respectively. However, they have some inherent disadvantages which pose negative effects on the AEC performance. In this paper, for the BoW based AEC problem, we propose a novel method to construct the codebook and generate the histogram. The self-organizing feature map (SOFM) network is utilized for codebook construction, which can ameliorate the local optimization problem. In addition, an N-Competition encoding strategy is proposed for histogram generation, and the robustness to the boundary points is improved. Experimental result shows that, the proposed method can achieve average 2.4% improvement in accuracy over the traditional BoW based method. Experimental analysis denote that our proposed approach can obtain robust boundary points and effective codebook.**

I. Introduction

In the information age, the amount of multimedia is exploding. The effectiveness of analyzing the multimedia data greatly depends on the ability of classifying and retrieving the multimedia data. As the audio data is an important type of multimedia data, classifying the acoustic events plays an important role in many applications [1][2][3][4].

The acoustic event classification (AEC) mainly refers to determining the type of acoustic event which is contained in an audio clip. The representation of audio is essential, since a good and discriminative feature can usually lead to better classification performance. Many methods have been developed to describe or represent the audio. Some methods [5][6][7] propose to describe the audio by a single label ("laughter"for example). Obviously, it fails to capture the nuance characteristics of the audio clip. Some researchers [8] believe that the audio data is made up of some special low level units. If we find out the characteristics of these units, it will be easier to find the nuance of the audio clip and understand the audio clip better. There are various methods to model the audio units [8] [9]. Among these methods, the bag-of-words (BoW) model [10] is the most widely used.

The earliest application of the BoW model is in natural language processing (NLP) and information retrieval (IR) [11]. For the NLP and IR problems, usually, the words have been already defined in the dictionary. However, for AEC, the "words" is not predefined for the audio. Therefore, we have to find a way to generate the "words". In the traditional BoW based AEC method [12] [13], the words are created by using a clustering algorithm (usually K-means) [12][13], then by vector quantization encoding (VQE), the original features are replaced by the indexes of words in the codebook which have nearest distances to the original features[12]. After that, an audio clip is represented by a frequency histogram in which each element represents the amount of a given audio-word in the codebook, and the classification is further conducted by using the histogram. The classification performance of the BoW model depends critically on several stages of the algorithm, which include constructing the codebook, choosing the codebook size, using encoding strategy to form the histogram, and classification.

There are two disadvantages for the traditional BoW model in AEC. First, the traditional K-means method for codebook construction has its native disadvantage of local optimization. This disadvantage may lead to low-quality codebook, thus the classification accuracy may be negatively affected. Second, the encoding strategy used in the traditional model is a hard encoding strategy, and it is not robust enough for boundary points. In this paper, we use a neural network, which is called the self-organizing feature map (SOFM), to represent the audio codebook. By doing this, the problem of local optimization is partly solved, and input features become more discriminative for classification. On the other hand, we propose an N-competition encoding (NCE) strategy instead of VQE, which is more robust to boundary points to some extent. The experiments show that the proposed method improves the performance over the traditional BoW model based method.

II. Database and Feature

The database we use here is obtained from the UPC-TALP database [14]. This database contains a set of isolated acoustic events which occur in a meeting room environment, and

it is recorded for the CHIL acoustic event detection task. The recorded sounds have no temporal overlapping. In our experiment, there have 903 audio clips of isolated acoustic events in the database, and these acoustic events belong to 13 different classes (Knock, Door Open, Door Close, Steps,Chair Moving, Cough and so on). The duration of each audio clip ranges from 1 second to 20 seconds. All audio clips used in the experiments are re-sampled from 44kHz to 16kHz.

We use Mel-Frequency Cepstral Coefficients (MFCC) as the low level feature. We calculate the MFCC with 13 coefficients (including the energy coefficient) using a window of 25ms with 10ms overlap. In order to express the dynamic information, the first and second order derivatives are also computed. Finally, the feature is represented as 39-dimension MFCC.

III. Proposed BoW Model

For solving the method disadvantages mentioned in the section I, we propose a novel BoW model. The structure of proposed model is shown as in Fig. 1. It contains four stages: First, we extract the low level feature. Second, we use SOFM for generating codebook. Third, NCE is used to form the histogram features. At last, the histogram features are used as the input for classification.

A. SOFM for Audio Codebook Generation

In BoW of words model, after we extract the low level feature, we need to cluster the feature vectors to generate the codebook. The performance of BoW is influenced by the quality of the codebook, and the quality of codebook is determined by the clustering result.

The traditional method for clustering to construct codebook is K-means. The reason that we do not use K-means is that K-means depends critically on the choice of initial clustering centers, which may leads to local optimization and low-quality codebook, thus the classification accuracy may be negatively affected. So, instead of K-means, the SOFM is used for codebook construction. It can partly solves the problem of local optimization by training with sufficient times and good optimizing training method.

The SOFM is a special kind of neural network, which is proposed in [15]. It is based on competitive learning, where output nodes compete to become the winner unit. It consists of input layer and competition layers. The competition layer has a two-dimensional grid of map units. The weight of a map unit is represented by w^i, and i is the index of the map unit. The units are connected with adjacent ones by neighborhood relations. During training, data points in the input space are mapped into nearby map units.

After feature extracting, a feature vector x^j is randomly chosen from the frame feature set $\{x^j | j = 1, 2, ..., N\}$, and N is the number of frame vector. Distances between input sample vector x^j and all the weight vectors are computed. The index of winner unit $I(x^j)$ is the map unit which is closest to x^j:

$$I(x^j) = arg\min_i \|x^j - w^i\|, i = 1, 2, ..., M, \quad (1)$$

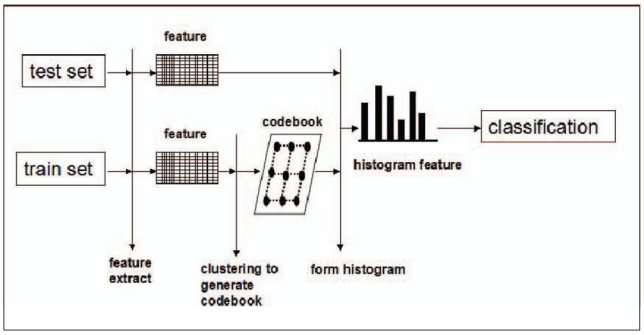

Fig. 1. The structure of proposed BoW model

where M is the total number map units. Next, the weight vectors are updated. The update rule of the tth training epoch for the vector is:

$$w^i(t+1) = w^i(t) + \eta(t)h_{i,I(x^j)}(x^j - w^i(t)), \quad (2)$$

where t is the training epochs, $w^i(t)$ is weight vector of map unit i in training epoch t, $\eta(t)$ is the learning rate coefficient in training epoch t and $h_{i,I(\mathbf{x})}(t)$ is the neighborhood kernel centered on the winner unit:

$$h_{i,I(x^j)}(t) = \exp(-\frac{d^2_{i,I(x^j)}}{2\delta^2(t)}), \quad (3)$$

where $d^2_{i,I(x^j)}$ is the distance between map unit i and winner map unit $I(x^j)$.

In general, the clustering is usually conducted on the whole training set, and large codebook usually leads to high classification accuracy. However, big codebook size always need big memory consumption and causes low clustering speed. Therefore, in practice, the high computational cost of codeword generating discourages the use of a large codebook. Here a strategy for clustering is used. Instead of on the whole training data, we conduct clustering (SOFM and K-means) in each class. Then small codebooks are generated for different classes, and these codebooks are combined to construct the a large codebook. This method contributes a lot to the problem of clustering speed and memory consumption, especially when the codebook size is big.

After SOFM network training, we will find the win neural map unit for every input feature vector x^j. We can judge this as a a kind of clustering, and every frame feature belongs to a winner unit. Then the weights of win neural map unit are used to form the codebook.

B. N-Competition Encoding Strategy

After the codebook is generated, the audio clip can be represented as a frequency distribution over the words in the codebook, which is called the frequency histogram. The common strategy to form histogram is VQE [12] [16]. However, as the VQE is a hard encoding strategy, and it is not robust to the boundary points. For example, the point A and point B in Fig. 2, even though they are very close to each other, they are divided to different clusters, just because they are separated by the boundaries between clusters. On the other

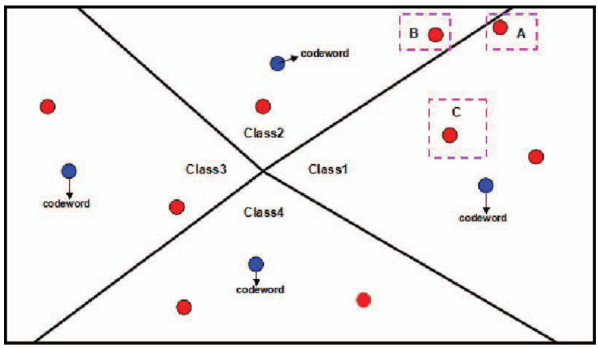

Fig. 2. Clustering. This Fig is used to show the detail of N-Competition Encoding Strategy.

hand, a small fluctuation of the feature point may make them be classified into a different cluster, then the feature points are counted to different bins of the histogram. Therefore, the resulting histogram will be less robust to boundary points.

In this section, NCE strategy is proposed in order to solve this problem. After feature extracting, an audio clip can be represented as $\{x^j|j = 1, 2, ..., n\}$, and n is the number of frame features in a audio clip. For a frame feature x^j in the audio clip, the distance between x^j and the weight of each SOFM unit w^i is computed as $d_i = \|x^j - w^i\|, i = 1, 2, ..., M$, where M is the codebook size, which is also the number of SOFM units.

Then we choose the SOFM units which satisfy $d_i \leq \xi * d_{min}$, where i is the index of SOFM unit, d_{min} is the minimum value in $\{d_i|i = 1, 2, ..., M\}$, and ξ is empirically set to be 1.5 here. Usually we can choose ξ to be larger than 1, and larger ξ makes more units selected. We use a set $S = \{s_i|i = 1, 2, ...\}$ to represent the indexes of selected units.

Finally, we choose l indexes of map units from the set S, and these l indexes are corresponding to the top l smallest distances in $\{d_i|i = 1, 2, ..., m\}$. If the number of indexes in S is smaller than l, then we get l indexes by repeatedly use the indexes in S. Take $l = 5$ for example: if we get the set $S = \{2, 1, 3, 4, 5, 7, 11, 12\}$, then the set of l indexes is $S_l = \{2, 1, 3, 4, 5\}$; if we get the set $S = \{2, 1\}$, then the set of l indexes is $S_l = \{2, 1, 2, 1, 2\}$. S_l is the final set we need.

After these steps, the frame feature x^j is represented by S_l with the indexes of l nearest map units. We compute all n frame features $\{x^j|j = 1, 2, ..., n\}$ in an audio clip as above, and then the audio clip is represented by a set with $n*l$ indexes of map units. Next, we count the number of every index in the set such that a frequency vector is obtained. Then L1 normalization is performed to transform the frequency vector into the probability distribution. We call the normalized vector as histogram.

We finally choose l units for each frame vector x^j. Why we first use $d_j \leq \xi * d_{min}$ to limit the selected units, instead of directly choosing the K nearest ones for x^j? Let us take the Point A and Point C in Fig. 2 for example. In this fig, both A and C belong to Class 1. If we use directly choose l nearest points and take $l = 2$, then both A and C will belong to Class $\{1, 2\}$. However, since C obviously belongs to class 1, this result is not reasonable enough. On the other hand, we can see that by using our NCE, C will belong to the Class $\{1, 1\}$, which fits the fact better.

C. Artificial Neural Network For Classification

After we get the histogram, the training sets and the testing sets will be represented as the resulting histograms. Then the training histograms are used as the input feature of a classifier and testing histograms are used to test the performance. The classifier we use is artificial neural networks (ANN). The structure of the ANN we use is a typical feed-forward neural network. It has one input layer, a hidden layer and an output layer. Back propagation algorithm is a common method to train ANN.

IV. Experiment and Analysis

A. Baseline Model and Experiment Setup

After we get the feature, a randomized five-fold cross-validation repeated five times. Every time, Of which five are used for training and the remaining one is used to test the classification accuracy. In our baseline system, we use the low level feature MFCC as the input feature. The classifier we used for our baseline model is ANN classifier described in III.C. We use a input layer, one hidden layer with 500 nodes and a output layer. The activation function of hidden layers is sigmoid function and the activation function of output layer is soft-max function. The objective function is mean square error cost function. The training epochs we use here is 500. The classification result is shown in Table 1.

B. Experiment Results

After we extract the feature, we get the low level feature MFCC. Then we cluster the training MFCC feature to generate the codebook. The training epochs for SOFM is 400 here. At last we use encoding strategy to form the histogram feature for training and testing. The training histogram is used to train a ANN, and the testing histogram is used to test the performance. The ANN is the same as baseline model.

The first model is the traditional BoW model (K-means+VQE model), it uses traditional K-means method to form the codebook, and then use VQE to form the histogram. The second model (SOFM+VQE model) is conducted for comparing the K-means and SOFM. It has the same histogram forming strategy (VQE) as the first model, but different clustering methods. Table 1 shows the performances of BoW based AEC algorithms using different clustering methods and different histogram forming strategies. In Table 1, the classification which uses SOFM as clustering outperforms K-means. This demonstrates that SOFM can construct a better quality of codebook which makes it more discriminative for classification than K-means, which led to better performance. The third model (SOFM+NCE model) is our final model, which uses the SOFM to generate the codebook, but NCE to form strategy. The SOFM+VQE model and SOFM+NCE model use the the same clustering method, but different encoding strategies: the proposed NCE strategy and the traditional VQE strategy. So from Table 1 we can

know that our NCE strategy outperforms the traditional VQE strategy.

Table 1. the result of different models

model(codebook size)	mean accuracy
baseline model	83.6%
K-means+VQE(78)	88.49%
K-means+VQE(377)	93.64%
SOFM+VQE(78)	93.89%
SOFM+VQE(377)	95.10%
SOFM+NCE(78)	94.10%
SOFM+NCE(377)	95.70%

C. Experimental Analysis

In the BoW model, we should choose the codebook size. In our model, the codebook size is determined by the number of map units of SOFM neural network. The number of map units, which typically varies from a few dozens up to several thousands, influences the accuracy and generalization capability of the SOFM. We choose 16 different codebook sizes ranging from 26 to 689 (that is 26, 52, 78, 104, 130, 182, 221, 260, 325, 377, 429, 481, 533, 585, 637, 689). As we do clustering on each class, the codebook size is the multiple of 13.

From Fig. 3, it is observed that lager codebook size usually produces a higher classification accuracy, and when the codebook size is big enough, the classification accuracy trends to level off. However, from Fig. 3, we can see some violations, meaning that some points having bigger codebook size achieve lower classification accuracy. This may due to some bad codewords produced by clustering, and that the boundary point is sensitive to small changes of codewords when using VQE, so we use NCE to overcome this disadvantage. From Fig. 3, the proposed NCE may not contribute a lot to the improving of the classification accuracy but more robust. In Fig. 3, the model with NCE is more smoothly, which prove that NCE is more robust.

An appropriate size of a codebook usually determined by the size of the database. The performances of different BoW based models are also compared in Fig. 3. It is shown that the proposed model performs better than the traditional codebook model. As it is shown in the right of Fig. 3, we can see that our final SOFM + NCE model (combine our SOFM model and NCE strategy) achieves average 2.4% improvement in accuracy (the average of increase accuracy of 16 different codebook size mentions above and the variance is 0.014%) compared to traditional K-means and VQE model.

V. Conclusions

We present to use the SOFM model for constructing the codebook. It can partly solve the local optimization problem. Furthermore, we introduce the NCE strategy to get the frequency histogram, which is more robust for modeling boundary points When comparing with VQE approach. Experimental results show that our proposed approach outperforms the traditional BoW modeling approach.

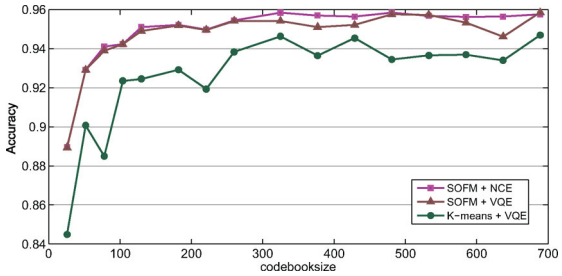

Fig. 3. The performance of different codebook size and different models

VI. Acknowledgment

This research was supported in part by the China National Nature Science Foundation No.91120303, No.61273267, No.61403370, No.90820011 and No. 61305027.

References

[1] P. K. Atrey, M. Maddage, and M. S. Kankanhalli, "Audio based event detection for multimedia surveillance," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[2] J. Ludeña-Choez and A. Gallardo-Antolín, "Nmf-based spectral analysis for acoustic event classification tasks," in *Advances in Nonlinear Speech Processing*, ser. Lecture Notes in Computer Science, T. Drugman and T. Dutoit, Eds. Springer Berlin Heidelberg, 2013, vol. 7911, pp. 9–16.

[3] S. Moncrieff, C. Dorai, and S. Venkatesh, "Detecting indexical signs in film audio for scene interpretation." in *ICME*, 2001.

[4] R. Toldo, U. Castellani, and A. Fusiello, "A bag of words approach for 3d object categorization," in *Computer Vision/Computer Graphics CollaborationTechniques*. Springer, 2009, pp. 116–127.

[5] M. K. I. Molla and K. Hirose, "Audio classification using dominant spatial patterns in time-frequency space," in *INTERSPEECH*, 2013, pp. 2915–2919.

[6] Z. Kons and O. Toledo-Ronen, "Audio event classification using deep neural networks." in *INTERSPEECH*. ISCA, 2013, pp. 1482–1486.

[7] S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 5, pp. 619–625, 2000.

[8] S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification." in *INTERSPEECH*, 2011, pp. 2265–2268.

[9] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo, "Large-scale multimodal semantic concept detection for consumer video," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, pp. 255–264.

[10] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 977–984.

[11] "Bag-of-words model," http://en.wikipedia.org/wiki/ Bag-of-words_model, from Wikipedia, the free encyclopedia, view in September 28,2014.

[12] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification." in *Interspeech*. ISCA, 2012.

[13] S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze, "Robust audio-codebooks for large-scale event detection in consumer videos." in *INTERSPEECH*, 2013, pp. 2929–2933.

[14] "Upc-talp database of isolated meeting-room acoustic events." eLRA Catalog no. S0268.

[15] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[16] Z. Huang, C. Weng, K. Li, Y. C. Cheng, and C. H. Lee, "Deep learning vector quantization for acoustic information retrieval," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 1350–1354.