Knowledge-Based Systems 195 (2020) 105600

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

LDA-based data augmentation algorithm for acoustic scene classification☆

Yan Leng [a,*], Weiwei Zhao [a], Chan Lin [a], Chengli Sun [b], Rongyan Wang [c], Qi Yuan [a], Dengwang Li [a]

[a] Shandong Key Laboratory of Medical Physics and Image Processing & Shandong Provincial Engineering and Technical Center of Light Manipulations, School of Physics and Electronics, Shandong Normal University, Jinan, Shandong, 250358, China
[b] School of Information, Nanchang Hangkong University, Nanchang, 330063, China
[c] School of Information Management, Dezhou University, Dezhou, 253023, China

ARTICLE INFO

Article history:
Received 1 September 2019
Received in revised form 29 January 2020
Accepted 31 January 2020
Available online 3 February 2020

Keywords:
Acoustic scene classification
Topic model
LDA
Key audio event
Non-key audio event

ABSTRACT

Deep neural network needs large amount of data for training, to obtain more data, many simple data augmentation algorithms have been proposed. In this paper, we propose a LDA-based data augmentation algorithm to extend the training set. The proposed LDA-based data augmentation algorithm uses the topic model LDA to detect the key audio words in the recordings, and further to detect the key audio events and non-key audio events for each recording; with the detected key-audio-event segments, for each acoustic scene class, the probability distribution of key-audio-event's occurrence numbers, the probability distribution of key-audio-event's locations under each occurrence number and the probability distribution of key-audio-event's durations under each occurrence number is counted, and then the new recordings are generated according to these probability distributions. Experiments are done on the public TUT acoustic scenes 2016 dataset, and the experimental results show that compared with the other simple data augmentation algorithms, the proposed LDA-based data augmentation algorithm is more stable and effective, it can get better generalization ability for different kinds of neural network on different datasets.

1. Introduction

Acoustic scene classification (ASC) is to classify a recording into one of the predefined classes which characterize the environment in which it is recorded, it becomes very popular in the last decade [1]. ASC and acoustic event classification [2,3] are the important tasks in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge series [4], ASC has received the highest number of submissions among all the available tasks in the first three editions of DCASE challenge. ASC has many useful applications, such as human–robot interaction [5], intelligent wearable interface [6] and context-aware computation [7] etc.

With the rapid development of deep learning technology, now more and more research communities choose to use deep neural network to perform ASC. Usually deeper network would have stronger representational ability, and then would obtain better classification performance, but deeper network would also easily cause overfitting problem if the training data is not enough. Acoustic scene class usually has big variability, for example, for the park scene, its acoustic characteristic at the weekend is very different from that on a weekday. To cover the different variants of an acoustic scene, the recordings should be collected in different locations, at different time periods, and with different devices etc., which means that the data collection would cost a lot of time and energy, in that way, the training data is usually not enough compared with the complexity of the deep network.

To increase the generalization ability of the deep network, many efforts have been made, including dropout [8], batch normalization [9] and other regularization technologies; beside these, data augmentation is another good choice, many simple data augmentation algorithms have been proposed. In image recognition field, random rotating, random cropping and flipping are some often used simple data augmentation algorithms, in acoustic field, there are some similar methods, for example, in [10] the authors use pitch shifting, time stretching and dynamic range compression to extend the training set; in [11] the authors create the noisy version of the dataset to enrich the training set; in [12] time shifting is adopted to introduce data augmentation into the training process; in [13], new recordings are generated by adding

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105600.

* Corresponding author.
E-mail address: lengyan@sdnu.edu.cn (Y. Leng).

https://doi.org/10.1016/j.knosys.2020.105600
0950-7051/

Gaussian noise into the original training data and also by re-sampling the original recordings; in [14] the left and right channel of the stereo audio data is used to augment the training set.

In this paper we propose another data augmentation algorithm for ASC, we innovatively use the topic model-LDA (Latent Dirichlet Allocation) to generate new data. Compared with the other often used simple data augmentation methods, the proposed LDA-based data augmentation algorithm is more stable, it can effectively improve the performance of different kinds of neural networks on different datasets.

The rest of the paper is organized as follows: Section 2 introduces LDA briefly, Section 3 explains the proposed LDA-based data augmentation algorithm in detail, Section 4 presents the experimental results and Section 5 gives conclusion.

2. LDA

LDA is a generative probabilistic model [15,16] which is widely used in text field. For a collection of documents D, each document is described as a mixture of latent topics, and each topic is modeled as a discrete distribution over the vocabulary of the documents. Given a LDA model, the generation of the document collection D can be modeled as follows [15]:

For $k = 1, \ldots, K$:
(a) $\emptyset^{(k)} \sim Dirichlet(\beta)$
For each document $d \in D$:
(a) $\theta^{(d)} \sim Dirichlet(\alpha)$
(b) For each word $w_i \in d$:
i. $z_i \sim \text{Discrete}(\theta^{(d)})$
ii. $w_i \sim \text{Discrete}(\emptyset^{(z_i)})$

where K is the number of latent topics, $\emptyset^{(k)}$ is the discrete probability distribution of topic k over the vocabulary, $\theta^{(d)}$ is the discrete probability distribution of document d over topics, z_i is the topic index of word w_i, indicating which topic has generated the word w_i, and α, β are the hyper parameters of *Dirichlet* distribution. The document-topic distribution matrix θ and the topic-word distribution matrix $\emptyset$ which are of interest to us are obtained through learning from the document collection D, Gibbs sampling [15] and variational inference [16] are the two most popular learning algorithms.

3. The proposed LDA-based data augmentation algorithm

3.1. The scheme of the algorithm

The proposed LDA-based data augmentation algorithm is based on two assumptions: first, the audio recording is composed of key audio events and non-key audio events; second, in an audio recording, the occurrence order of key audio events or non-key audio events is random, and the occurrence numbers, occurrence locations and durations of key audio events respectively obey a certain distribution. Key audio events are the events that play a key role in understanding the acoustic scene. Usually the key audio events of an acoustic scene are unique, which means that such key audio events mainly appear in this scene class, but seldom in other scene classes, for example, the tapping of the keyboard and the sound of the working printer are the typical key audio events of the office scene. Non-key audio events are the events that do not play a key role in understanding the acoustic scene. Non-key audio events would usually occur in different scenes, and therefore is not unique to a certain scene class, for example, the speech in the office scene is a non-key audio event.

Based on the above assumptions, the scheme of data augmentation is designed as follows: first, extract the key-audio-event segments and non-key-audio-event segments from each audio recording of the training set, and record the occurrence number, locations and durations of the key audio events in each recording, here we take the distance between the starting point of the event and the beginning of the recording as the location; then for each scene class, count the probability distribution of the occurrence numbers, the probability distribution of locations under each occurrence number and the probability distribution of durations under each occurrence number; finally, generate new recordings based on the above probability distributions. The flowchart of the proposed data augmentation algorithm is shown in Fig. 1.

3.2. Extract key audio events and non-key audio events

For human, perception of acoustic scene is usually done by analyzing the audio events contained in it, the key audio events in the scene then play the critical role in understanding the scene, in that way we think that the topics of a scene are mainly reflected by the key audio events in it, and then the topic distribution of a scene should be similar to the topic distribution of the key audio events. In an audio recording, the duration of the audio events is usually different, to locate the key audio events, we can split the recording into a series of basic unit, such basic units can be called audio words, just like the words in a text document, in that way, the key audio events are composed of a series of key audio words, and then the key audio events can be located through locating the key audio words.

In this paper we propose to use LDA topic model to find out the key audio words, and then further to locate the key audio events. Since a key audio event is composed of key audio words, the topic distribution of a key audio event should be consistent with that of its key audio words, and then the topic distribution of its key audio words would be similar to that of the scene to which the key audio event belongs. The flowchart of extracting key audio events and non-key audio events are shown in Fig. 2.

3.2.1. Audio words

To split the audio recordings into audio words, first, each audio recording is segmented into frames, and for each frame, 40 log mel-spectrogram is extracted by librosa [17], the frame length is set to be 1764 samples (with sampling rate = 44.1 kHz, the frame length is 40 ms) with 50% overlap; then standardization is performed feature-wise; finally, all the frames in the training set are clustered by k-means [18], and the cluster centroids are taken as the audio words, all audio words make up an audio dictionary, the frames in a cluster are then represented as the audio word of that cluster.

3.2.2. Topic distribution and key audio words

With the audio dictionary, each audio recording can be seen as a bag of audio words, count the audio word histogram for each audio recording and concatenate them together, the training set D can then be represented as a recording-word co-occurrence matrix:

$$D = \left[h_{w_i}^{(d)}\right]_{M \times N} \tag{1}$$

where $h_{w_i}^{(d)}$ represents the occurrence number of audio word w_i in audio recording d ($w_i = w_1, \ldots, w_M, d = 1, \ldots, N$), M is the number of audio words in the audio dictionary and N is the number of audio recordings in the training set.

With the recording-word co-occurrence matrix D, LDA is adopted to learn the recording-topic distribution matrix θ and the topic-word distribution matrix $\emptyset$ through the variational EM method [16]:

$$\theta = \left[\theta_k^{(d)}\right]_{K \times N} \tag{2}$$

$$\emptyset = \left[\emptyset_{w_i}^{(k)}\right]_{M \times K} \tag{3}$$

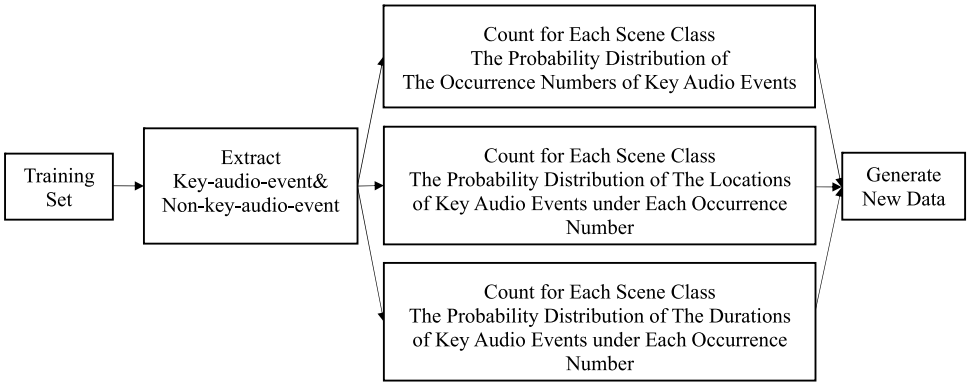


Fig. 1. The flowchart of the proposed data augmentation algorithm.

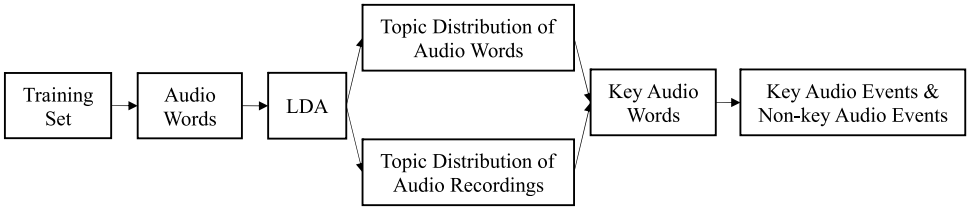


Fig. 2. The flowchart of extracting key audio events and non-key audio events.

where $\theta_k^{(d)}$ represents the probability distribution of audio recording d over topic k and $\emptyset_{w_i}^{(k)}$ represents the probability distribution of topic k ($k = 1, \ldots, K$) over audio word w_i, K is the number of latent topics. With θ and $\emptyset$, for each audio word w_i, the word-topic distribution $\gamma_k^{(w_i)} = p(z_i = k|w_i)$ can be calculated as follows:

$$\begin{aligned}\gamma_k^{(w_i)} &= p\left(z_i = k|w_i\right) = \frac{p\left(z_i = k, w_i\right)}{p\left(w_i\right)} = \frac{p\left(z_i = k, w_i\right)}{\sum_{k=1}^{K} p\left(z_i = k, w_i\right)} \\ &= \frac{p\left(z_i = k\right) p\left(w_i|z_i = k\right)}{\sum_{k=1}^{K} p\left(z_i = k\right) p\left(w_i|z_i = k\right)} = \frac{p\left(z_i = k\right) \emptyset_{w_i}^{(k)}}{\sum_{k=1}^{K} p\left(z_i = k\right) \emptyset_{w_i}^{(k)}} \\ &= \frac{\sum_{d=1}^{N} p\left(z_i = k, d\right) \emptyset_{w_i}^{(k)}}{\sum_{k=1}^{K} \sum_{d=1}^{N} p\left(z_i = k, d\right) \emptyset_{w_i}^{(k)}} \\ &= \frac{\sum_{d=1}^{N} p\left(d\right) p\left(z_i = k|d\right) \emptyset_{w_i}^{(k)}}{\sum_{k=1}^{K} \sum_{d=1}^{N} p\left(d\right) p\left(z_i = k|d\right) \emptyset_{w_i}^{(k)}} \\ &= \frac{\frac{1}{N} \sum_{d=1}^{N} p\left(z_i = k|d\right) \emptyset_{w_i}^{(k)}}{\frac{1}{N} \sum_{k=1}^{K} \sum_{d=1}^{N} p\left(z_i = k|d\right) \emptyset_{w_i}^{(k)}} \\ &= \frac{\sum_{d=1}^{N} \theta_k^{(d)} \emptyset_{w_i}^{(k)}}{\sum_{k=1}^{K} \sum_{d=1}^{N} \theta_k^{(d)} \emptyset_{w_i}^{(k)}} \end{aligned} \tag{4}$$

As that analyzed previously, the topic distribution of the audio words of a key audio event would be similar to that of the scene to which the key audio event belongs, therefore we use the cosine similarity to find out the key audio words:

$$\begin{aligned} cos\left(\gamma^{(w_i)}, \theta^{(d)}\right) &= \frac{\gamma^{(w_i)} \cdot \theta^{(d)}}{\left|\gamma^{(w_i)}\right| \times \left|\theta^{(d)}\right|} \\ &= \frac{\sum_{k=1}^{K} \gamma_k^{(w_i)} \theta_k^{(d)}}{\sqrt{\sum_{k=1}^{K} \left(\gamma_k^{(w_i)}\right)^2} \sqrt{\sum_{k=1}^{K} \left(\theta_k^{(d)}\right)^2}} \end{aligned} \tag{5}$$

For each audio recording d, after calculating its topic distribution similarity with audio words, a threshold T is set to be the product of the maximum similarity value and a constant ρ which is determined through experiments:

$$T = \rho.max\left([cos\left(\gamma^{(w_1)}, \theta^{(d)}\right), \ldots, cos\left(\gamma^{(w_M)}, \theta^{(d)}\right)]\right) \tag{6}$$

With the threshold T, the audio word w_i that satisfies $cos\left(\gamma^{(w_i)}, \theta^{(d)}\right) > T$ is taken as the key audio word of recording d, otherwise is taken as non-key audio word.

3.2.3. Key audio events and non-key audio events

For each audio recording, after its key audio words being detected, label them as "1", and label the other non-key audio words as "0"; a median filter with window size 1×5 is then used to filter out the isolated key audio words; after filtering, the successive key audio words constitute key audio event segments and the successive non-key audio words constitute non-key audio event segments.

3.3. Count the probability distributions

For each acoustic scene class, collect the key audio event segments and non-key audio event segments from each audio recording, and record the occurrence number, locations and durations of the key audio events for each recording; then count the probability distribution of key-audio-event's occurrence numbers through histogram method, denoted as $P(num)$; collect the key audio event segments which come from the recordings with the same occurrence number of key audio events, under each occurrence number, count the probability distribution of key-audio-event's locations, denoted as $P(loc|num)$, and the probability distribution of key-audio-event's durations, denoted as $P(dur|num)$, through histogram method.

$$P\left(num\right) = \text{hist}([num_1, \ldots, num_l]) \tag{7}$$

$$P\left(loc|num\right) = \text{hist}([loc_1, \ldots, loc_r]|num) \tag{8}$$

$$P\left(dur|num\right) = \text{hist}([dur_1, \ldots, dur_r]|num) \tag{9}$$

where hist() indicates the histogram function. $[num_1, \ldots, num_l]$ indicates the occurrence number of key audio events of the audio recordings in an acoustic scene class, l is the number of audio recordings. $[loc_1, \ldots, loc_r]|num$ indicates under the occurrence

number *num*, the locations of the key audio events, r is the number of key audio event segments under the occurrence number *num*, and $[dur_1, \ldots, dur_r]|num$ indicates under the occurrence number *num*, the durations of the key audio events.

3.4. Generate new data

For each acoustic scene class, given the probability distribution of key-audio-event's occurrence numbers $P(num)$, the probability distribution of key-audio-event's locations $P(loc|num)$ and the probability distribution of key-audio-event's durations $P(dur|num)$, a new audio recording i of an acoustic scene class is generated as follows:

(1) generate the key-audio-event's occurrence number num_i according to probability distribution $P(num)$: $num_i \sim P(num)$;
(2) generate the key-audio-event's locations $loc_j (j = 1, \ldots, num_i)$ according to the probability distribution $P(loc|num_i)$: $loc_j \sim P(loc|num_i)$;
(3) generate the key-audio-event's durations $dur_j (j = 1, \ldots, num_i)$ according to the probability distribution $P(dur|num_i)$: $dur_j \sim P(dur|num_i)$;
(4) for each key-audio-event $j (j = 1, \ldots, num_i)$, from the key-audio-event segments of the acoustic scene class, select the one with the duration value that is closest to dur_j, and put it in location loc_j;
(5) for each gap between two successive key-audio-events, from the non-key-audio-event segments of the acoustic scene class, select the one with the duration value that is closest to the gap length, and put it in the gap location. For the first key-audio-event, if its location $loc_1 \neq 0$, then from the non-key-audio-event segments of the acoustic scene class, select the one with the duration value that is closest to loc_1, and put it in front of the first key-audio-event. For the last key-audio-event, if the sum of its location value and its duration value is less than the average length of recordings in the acoustic scene class, that is $loc_{num_i} + dur_{num_i} < ave$, where ave indicates the average length of recordings in the acoustic scene class, then from the non-key-audio-event segments of the acoustic scene class, select the one with the duration value that is closest to $ave - (loc_{num_i} + dur_{num_i})$, and put it behind the last key-audio-event segment.
(6) repeat step (1) $\sim$ (5) to generate more new audio recordings for the acoustic scene class.

4. Experimental results

4.1. Dataset and experimental setting

In this paper, the public TUT acoustic scenes 2016 dataset [19] is adopted for experiments. This dataset contains two parts: the development subdataset and the evaluation subdataset, we use both for experiments. The development subdataset consists of 15 different acoustic scene classes, each class contains 78 audio recordings, each recording is 30 s long, total 1170 recordings and 9.75 h long. Along with the subdataset, a 4-fold cross-validation setup is provided. The evaluation subdataset consists of 15 acoustic scene classes, each class contains 26 30 s-long audio recordings, total 390 recordings and 3.25 h long. All recordings in the dataset are recorded in stereo with 44.1 kHz sampling rate and 24 bit resolution, the recordings are converted into mono channel for experiments.

When using the development subdataset for experiments, the predefined 4-fold cross-validation is adopted, the evaluate set in each fold is used to early-stop the network training process. When using the evaluation subdataset for experiments, the 4-fold cross-validation setup of the development subdataset is still adopted, the training set in each fold is used for training, the evaluate set in each fold is used to early-stop the network training process, all data in the evaluation subdataset is used for test.

For feature extraction, each audio recording is segmented into frames, and for each frame, 40 log mel-spectrogram is extracted by librosa [17], the frame length is set to be 40 ms with 50% overlap; then standardization is performed feature-wise. For experimental evaluation, classification accuracy is used as the metric:

$$accuracy = \frac{the\ number\ of\ correctly\ classified\ audio\ recordings}{the\ total\ amount\ of\ test\ recordings} \tag{10}$$

4.2. Classification model

To detect the effectiveness of the proposed data augmentation algorithm, in this paper we use three different types of neural network as the classification model to perform ASC, that is the MLP (MultiLayer Perceptron) network used in [20], the VGG-16 style network, and the LSTM (Long Short-Term Memory) network used in [21]. All networks are implemented using Keras (v2.1.3) with TensorFlow as backend. The network architecture of the three models are listed in Tables 1–3 respectively.

The MLP network has two hidden fully-connected layers, each containing 150 ReLU (Rectified Linear Units) units, and a 15-way softmax output layer with one unit for each class label. Dropout with probability 40% is applied to each hidden layer. MLP is trained using categorical cross-entropy as the loss function, and using Adam with a fixed learning rate of 0.0001 as the optimizer. The mini-batch size is set to 256. Training is stopped through early stopping if the validation loss is not decreased during 20 epochs, up to a maximum of 300 epochs. The input shape of MLP is (256,120), 256 is the mini-batch size, and 120 is the feature dimension obtained by stacking the current, the previous and the next frames together in order to contain the context information of the current frame. Classification is performed on each frame, and the classification result of the recording is obtained by majority voting.

Different from the standard VGG-16 network, the input of the adopted VGG-16 style network is a single-channel acoustic chunk spectrum with shape of 40 $\times$ 150 which is composed of 150 successive frames, and the output layer is a 15-way softmax layer. The network is trained using categorical cross-entropy as the loss function, and using Adam as the optimizer, the initial learning rate is set to be 0.001, and decayed with decaying factor 0.0001. The mini-batch size is set to 128. The same early stopping criterion as that of MLP is used. Classification is performed on each acoustic chunk, and the classification result of the recording is obtained by majority voting.

The LSTM network has two hidden layers, each containing 256 LSTM units, one hidden fully connected layer with 512 ReLU units, and a 15-way output softmax layer. Dropout with probability 40% is applied to each hidden layer. The LSTM network is trained using categorical cross-entropy as the loss function, and using Adam with a fixed learning rate of 0.0001 as the optimizer. The mini-batch size is set to 256. The same early stopping criterion as that of MLP is used. The input of the LSTM network is a single-channel acoustic chunk spectrum with shape of 40 $\times$ 40 which is composed of 40 successive frames. Classification is performed on each acoustic chunk, and the classification result of the recording is obtained by majority voting.

Table 1
MLP network architecture.

Input (256, 120)
FC-150-ReLU-Drop-Out(0.4) FC-150-ReLU-Drop-Out(0.4) 15-way Softmax

Table 2
VGG-16 style network architecture.

Input 1×40×150
2×Conv3-stride1-64-ReLU Max Pooling
2×Conv3-stride1-128-ReLU Max Pooling
3×Conv3-stride1-256-ReLU Max Pooling
3×Conv3-stride1-512-ReLU Max Pooling
3×Conv3-stride1-512-ReLU Max Pooling
FC-4096 FC-4096
15-way Softmax

Table 3
LSTM network architecture.

Input 1×40×40
LSTM-256-Drop-Out(0.4) LSTM-256-Drop-Out(0.4) FC-512-ReLU-Drop-Out(0.4) 15-way Softmax

4.3. Data augmentation with different size

According to the proposed generation process of new data, it can be seen that for each acoustic scene class, the proposed LDA-based data augmentation algorithm can generate as much new data as you want, in this section, we will do experiments to see the effect of the size of data augmentation on classification performance. For the development subdataset, assuming that in the training set of each fold, the number of audio recordings in each scene class is Ω, then for each scene class, Ω, 2Ω and 3Ω new audio recordings are generated and used for training respectively, in this way, the size of data augmentation is respectively twice, three times and four times that of the original training set. Under different sizes of data augmentation, the three classification models are used to perform ASC, and the classification results are compared with that of using the original training set for experiments, the experimental results are shown in Figs. 3–5. In these figures, the average results over 4-folds are shown; the size of data augmentation is represented as 1,2,3,4 to respectively indicate the original training set, twice, three times and four times the size of the original training set.

From Fig. 3 it can be seen that for MLP, on the development subdataset, its classification performance increases greatly when the size of data augmentation is twice that of the original training set; as the size of data augmentation increases further, its classification performance only increases slightly compared with that of using the original training set for training. On the evaluation subdataset, compared with that of using the original training set for training, the classification performance of MLP increases when the size of data augmentation is twice that of the original training set, but decreases slightly when the size increases further. The reason for performance degradation when the size of data augmentation is larger may be that the MLP model used in this paper is a small network, double size of the original training set is already enough, and more training data may have introduced outliers. From Fig. 4 it can be seen that for VGG-16, on both the development subdataset and the evaluation subdataset, the larger the size of data augmentation, the higher the classification performance. VGG-16 is a much larger network, it needs more training data, and then data augmentation with larger size would achieve better results. From Fig. 5 it can be seen that for LSTM, on the development subdataset, its performance increases with the increase of data augmentation size; while on the evaluation subdataset, its performance is best when the size of data augmentation is twice that of the original training set, as the size increases further, though the performance does not further increases, it is still much better than that of using the original training set for training. According to the proposed generation process of new data, when too much new data is generated, inevitably there would be some overlap between different new recordings, that is to say, after data augmentation, there would be some redundancy in the new training set, maybe it is the redundancy that causes larger size of data augmentation has not obtained better generalization ability on the evaluation subdataset for LSTM. In summary, when the network is much big, the proposed LDA-based data augmentation algorithm can generate effective new data to help to improve the performance; the proposed data augmentation algorithm can generate as much new data as you want, but when too much new data is generated, it should be noticed that there will inevitably be some redundancy in the new training set.

4.4. Performance comparison of different data augmentation algorithms

To test the effectiveness of the proposed LDA-based data augmentation algorithm (denoted as LDA-based for short), in this section, we will compare the performance of the classification models under different data augmentation algorithms, we choose the following data augmentation algorithms for comparison: pitch shifting [10], time shifting [12], adding Gaussian noise of 6db to the original recordings [13] (denoted as Gaussian noise for short), re-sampling the original recordings from 44.1 kHz to 16 kHz [13] (denoted as sr-16k for short), in addition to the mean of the left and right channel, adding the left channel signal to the training set (denoted as left channel for short), and adding the right channel signal to the training set (denoted as right channel for short) [14]. The performance of the classification models trained by the original training set is taken as the baseline (denoted as original for short). To be fair, for the LDA-based data augmentation algorithm, its performance with the size of data augmentation equaling twice that of the original training set is used for comparison.

From the mean results over 4-folds in Tables 4–6, it can be seen that MLP and LSTM obtain the best performance under the proposed LDA-based data augmentation algorithm; although under LDA-based data augmentation algorithm, VGG-16 network has not obtained the best result, its performance is comparable to the best one. Among all the data augmentation algorithms, LDA-based algorithm is more stable, it is the only one that can always obtain improved performance compared with the baseline, while other algorithms are not very stable, for example, the left channel algorithm can improve the performance of MLP and VGG-16 compared with the baseline, but fails for LSTM; compared with the baseline, the pitch shifting, time shifting, Gaussian noise and sr-16k algorithms can effectively improve the performance of VGG-16 on evaluation subdataset, but fails on development subdataset; compared with the baseline, the right channel algorithm can improve the performance of MLP and LSTM on development subdataset, but fails on evaluation subdataset.

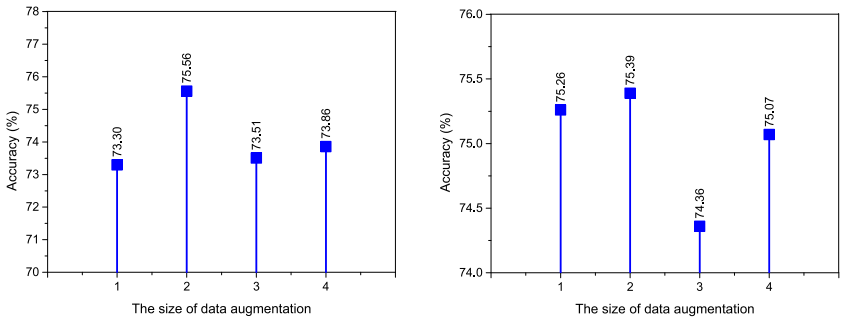


Fig. 3. Classification accuracy of MLP under different sizes of data augmentation. (a) for development subdataset and (b) for evaluation subdataset.

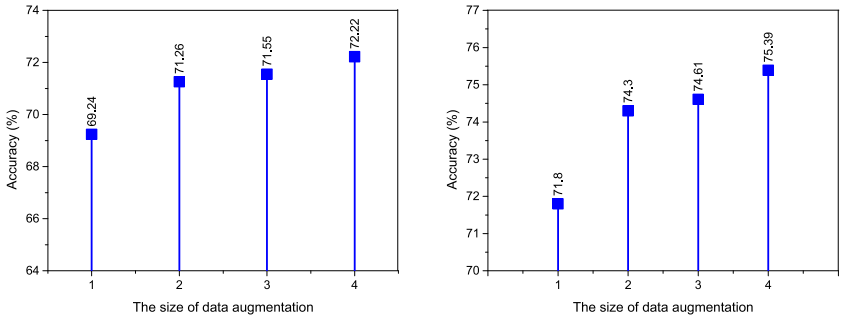


Fig. 4. Classification accuracy of VGG-16 under different sizes of data augmentation. (a) for development subdataset and (b) for evaluation subdataset.

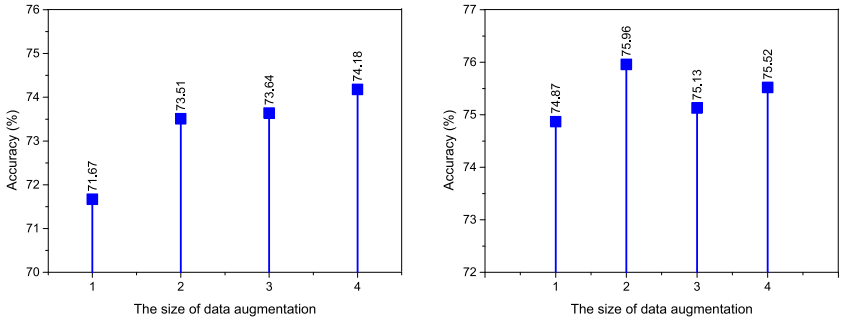


Fig. 5. Classification accuracy of LSTM under different sizes of data augmentation. (a) for development subdataset and (b) for evaluation subdataset.

For the LDA-based data augmentation algorithm, the new generated recordings are composed of key audio events and non-key audio events, and the key audio events are generated according to the statistical probability distributions obtained from the training data, in this way, the new generated recordings could simulate the acoustic scenes in real environment more effectively, and then can effectively help the classification models to improve the performance.

4.5. The characteristics of the augmented data

In order to analyze the usefulness of the augmented data, in this section, some experiments are done to investigate the characteristics of the augmented data. The training dataset in the first fold is adopted for experiments. First, the average occurrence number, the average length and the average location of the key audio events in each acoustic scene class of the original training dataset and the augmented dataset are counted respectively. To be fair, the size of the augmented dataset is set to be the same as that of the original training dataset, that is, the number of audio recordings in each acoustic scene class of the augmented dataset is the same as that of the original training dataset. The statistical results are shown in Table 7, each acoustic scene class is represented by a number: 1 = beach, 2 = bus, 3 = cafe/restaurant, 4 = car, 5 = city_center, 6 = forest_path, 7 = grocery_store, 8 = home, 9 = library, 10 = metro_station, 11 = office, 12 = park, 13 = residential_area, 14 = train, 15 = tram; the length and location is measured in frames, that is, if the length equals 69.9, it means that the length is 69.9 frames long. Second, from the 15 acoustic scene classes, 3 classes are randomly selected which are cafe/restaurant, car and metro_station, and for each of the 3 classes, an audio recording is randomly selected from the original training dataset and the augmented dataset respectively, the audio recordings are then shown in the form of 'key audio event' and 'non-key audio event' sequence as that shown in Fig. 6, where 1 indicates key audio event and 0 indicates non-key audio event.

Table 4
Classification accuracy (%) of MLP under different data augmentation algorithms.

Algorithm	MLP									
	Development subdataset					Evaluation subdataset				
	fold1	fold2	fold3	fold4	mean	fold1	fold2	fold3	fold4	mean
Original	75.72	71.03	72.48	73.97	73.3	74.36	71.54	77.18	77.95	75.26
Pitch shifting	76.55	70.34	72.82	75.68	73.85	72.56	71.54	74.62	75.64	73.59
Time shifting	74.14	67.93	72.15	70.89	71.28	71.79	70.00	72.82	73.85	72.12
Gaussian noise	73.45	69.66	72.48	76.03	72.91	74.62	71.54	73.90	75.90	73.99
sr-16k	74.14	69.31	74.50	75.68	73.41	75.13	72.31	73.08	75.64	74.04
Left channel	76.55	70.69	70.81	76.37	73.61	74.87	74.10	76.67	75.90	**75.39**
Right channel	77.24	72.07	77.18	74.66	75.29	74.62	71.54	74.62	74.10	73.72
LDA-based	77.93	74.14	74.50	75.68	**75.56**	75.38	72.31	76.16	77.69	**75.39**

Table 5
Classification accuracy (%) of VGG-16 under different data augmentation algorithms.

Algorithm	VGG-16									
	Development subdataset					Evaluation subdataset				
	fold1	fold2	fold3	fold4	mean	fold1	fold2	fold3	fold4	mean
Original	71.03	64.48	65.77	75.68	69.24	71.03	73.59	70.77	71.79	71.80
Pitch shifting	71.38	68.28	66.44	67.12	68.31	76.67	76.15	77.95	69.74	75.13
Time shifting	66.90	64.48	62.75	65.75	64.97	71.03	74.36	74.62	71.79	72.95
Gaussian noise	72.76	61.38	69.80	72.26	69.05	77.44	74.36	74.36	73.33	74.87
sr-16k	70.00	68.97	68.12	66.78	68.47	76.15	73.08	75.64	77.18	**75.51**
Left channel	69.66	67.93	71.81	75.68	71.27	73.59	73.08	74.87	74.62	74.04
Right channel	70.34	71.03	71.81	72.60	**71.45**	72.05	73.59	75.90	71.54	73.27
LDA-based	75.86	66.55	67.45	75.17	71.26	72.31	75.64	75.38	73.85	74.30

Table 6
Classification accuracy (%) of LSTM under different data augmentation algorithms.

Algorithm	LSTM									
	Development subdataset					Evaluation subdataset				
	fold1	fold2	fold3	fold4	mean	fold1	fold2	fold3	fold4	mean
Original	73.79	67.24	72.48	73.97	71.87	74.87	72.56	76.67	75.38	74.87
Pitch shifting	72.76	70.69	72.82	74.32	72.65	73.85	70.51	75.13	77.44	74.23
Time shifting	72.76	69.31	66.11	76.71	71.22	71.79	71.28	78.46	76.92	74.61
Gaussian noise	73.79	67.93	68.12	77.05	71.72	71.79	74.10	77.44	76.67	75.00
sr-16k	71.72	67.59	68.12	74.32	70.44	69.74	75.64	76.15	74.62	74.04
Left channel	75.52	68.28	66.44	69.52	69.94	70.26	70.51	76.41	71.79	72.24
Right channel	70.69	69.31	71.14	77.74	72.22	77.18	71.54	75.13	74.10	74.49
LDA-based	76.90	70.69	72.82	73.63	**73.51**	75.90	75.64	76.92	75.38	**75.96**

From Table 7 it can be seen that the new generated data has the statistical characteristics which is very similar to that of the original training data, which means that the new generated data would imitate the original data from the statistical perspective. From Fig. 6 it can be seen that for each acoustic scene class, compared with the original data, the new generated data has redistributed the key audio events in a very different way, it tends to put the key audio events together, in this way, the new generated data can highlight the key audio events, and this is favorable for classification. Just think about how we humans recognize the acoustic scene, if the key audio events which can reflect the topic of the acoustic scene appear continuously, then it would be more easy for us to recognize the acoustic scene, otherwise, if the key audio events are always cut off by the non-key audio events, then it would increase the recognition difficulty, people would need more time to recall all the key audio events contained in the scene, and then to analyze the scene they reflect. In summary, the proposed LDA-based data augmentation algorithm can generate new data according to the statistical characteristics of the original training data, and it can redistribute the key audio events, making the new generated data highlight the key audio events, which is favorable for classification.

5. Conclusion

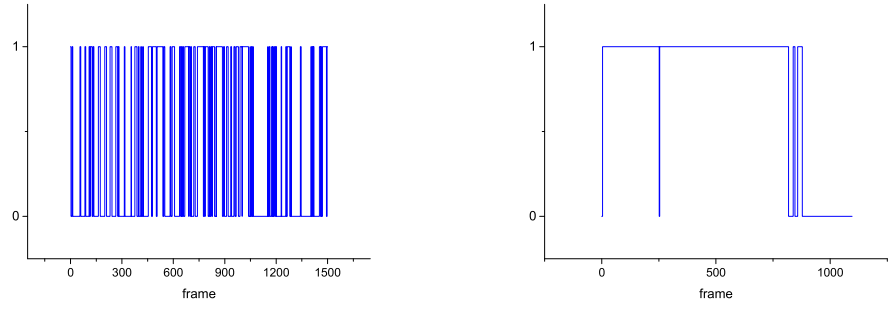
In this paper, we innovatively propose a LDA-based data augmentation algorithm which detects the key audio events by topic model LDA, and generates new recordings according to the distributions of key audio events counted from the original training set. The advantage of LDA-based data augmentation algorithm is that it is more stable and effective, it can improve the performance of different kinds of networks on different test sets, while other simple data augmentation algorithms can either improve the performance of one kind of network, but fails on another kind, or improve the performance on one dataset, but fails on another one. The disadvantage of LDA-based data augmentation algorithm is that when too much new data is generated, inevitably there would be some overlap between different new recordings, which would then introduce some redundancy into the new training set. The innovative application of LDA in this paper provides clues to apply LDA to various research fields, for example, it can be extended and applied to deal with the imbalance classification problem [22,23]. In future work, more efforts will be done to reduce the redundancy in new generated data, and to expand LDA to other research fields.

CRediT authorship contribution statement

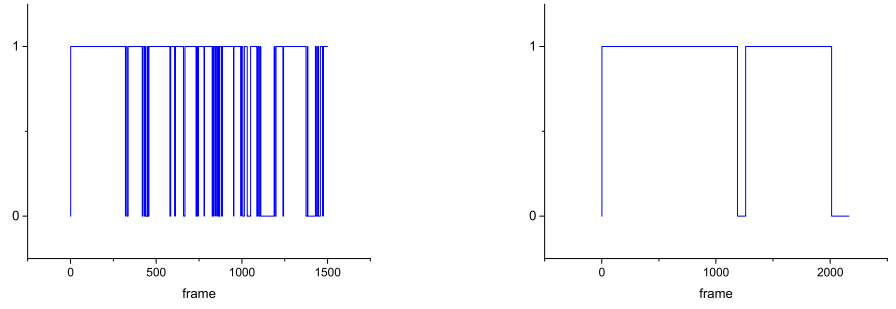
Yan Leng: Conceptualization, Methodology, Writing - original draft, Visualization, Supervision, Project administration, Funding acquisition. **Weiwei Zhao:** Software, Data curation, Writing - review & editing. **Chan Lin:** Software, Data curation, Writing - review & editing. **Chengli Sun:** Investigation, Validation, Funding

Table 7
The average occurrence number, the average length and the average location of the key audio events in each acoustic scene class of the original training dataset and the augmented dataset.

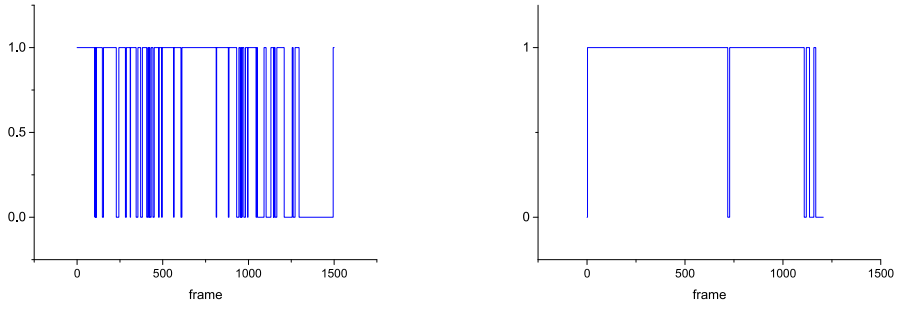
Class	Original dataset			Augmented dataset		
	Occurrence number	Length	Location	Occurrence number	Length	Location
1	28.5	69.9	709.2	28.3	89.7	671.3
2	19.2	122.4	694.4	17.6	133.9	699.0
3	39.0	35.6	736.3	34.5	41.9	719.1
4	14.6	182.8	613.9	14.3	171.0	646.6
5	24.3	96.8	672.6	25.3	78.9	672.3
6	14.6	154.7	651.1	15.7	138.3	688.6
7	33.1	63.1	710.8	33.5	57.0	725.6
8	34.6	34.4	742.8	34.8	35.1	742.9
9	32.1	48.7	710.1	33.8	36.6	717.4
10	32.6	50.7	713.6	31.5	46.2	713.0
11	30.4	63.7	715.5	27.8	69.7	713.4
12	22.9	113.9	694.0	20.0	116.9	669.9
13	23.1	70.3	703.2	20.9	67.1	683.2
14	31.0	97.0	663.7	29.2	92.1	663.3
15	40.5	41.1	730.1	39.1	56.1	723.4



(a)



(b)



(c)

Fig. 6. Audio recording in the form of 'key audio event' and 'non-key audio event' sequence, subfigure (a) is for cafe/restaurant, (b) for car and (c) for metro_station. In each subfigure, the left one shows the audio recording from the original training dataset, and the right one shows the audio recording from the augmented dataset.

acquisition. **Rongyan Wang:** Investigation, Resources, Funding acquisition. **Qi Yuan:** Visualization, Formal analysis. **Dengwang Li:** Resources, Funding acquisition.

Acknowledgments

This work was supported in part by the Project of National Natural Science Foundation of China (grant numbers 61401259, 61861033, 61971271), Research Fund for Young and Middle-aged Scientists of Shandong Province (grant number ZR2016FB25), China's Jiangxi Province Natural Science Foundation (grant number 20181BAB202022), the Taishan Scholars Project of Shandong Province (grant number Tsqn20161023) and the Primary Research and Development Plan of Shandong Province (grant number 2018GGX101018).

References

[1] Y. Leng, N. Zhou, C.L. Sun, X.Y. Xu, Q. Yuan, C.F. Cheng, Y.X. Liu, D.W. Li, Audio scene recognition based on audio events and topic model, Knowl.-Based Syst. 125 (2017) 1–12.
[2] Y. Leng, C.L. Sun, X.Y. Xu, Q. Yuan, S.N. Xing, H.L. Wan, J.J. Wang, D.W. Li, Employing unlabeled data to improve the classification performance of SVM, and its application in audio event classification, Knowl.-Based Syst. 98 (2016) 117–129.
[3] Y. Leng, C.L. Sun, C.F. Cheng, X.Y. Xu, S. Li, H.L. Wan, J. Fang, D.W. Li, Classification of overlapped audio events based on AT, PLSA, and the combination of them, Radioengineering 24 (2) (2015) 593–603.
[4] Detection and classification of acoustic scenes and events 2019 (DCASE2019), 2019, http://dcase.community/.
[5] A. Tsiami, P.P. Filntisis, N. Efthymiou, P. Koutras, G. Potamianos, P. Maragos, Far-field audio-visual scene perception of multi-party human–robot interaction for children and adults, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 6568–6572.
[6] F. Eyben, F. Weninger, F. Gross, B. Schuller, Recent developments in openSMILE, the munich open-source multimedia feature extractor, in: Proceedings of the 21st ACM International Conference on Multimedia (MM '13), 2013, pp. 835–838.
[7] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 414–454.
[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[9] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, arXiv preprint arXiv:1502.03167.
[10] J. Salamon, J. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, IEEE Signal Process. Lett. 24 (3) (2017) 279–283.
[11] J. Abeßer, S.I. Mimilakis, R. Grafe, H. Lukashevich, Acoustic scene classification by combining autoencoder-based dimensionality reduction and convolutional neural networks, in: Proceedings of the 2nd DCASE Workshop on Detection and Classification of Acoustic Scenes and Events, 2017, pp. 7–11.
[12] K.J. Piczak, The details that matter: Frequency resolution of spectrograms in acoustic scene classification, in: IEEE Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2017), 2017, pp. 103–107.
[13] A. Vafeiadis, D. Kalatzis, K. Votis, D. Giakoumis, D. Tzovaras, L.M. Chen, R. Hamzaoui, Acoustic scene classification: From a hybrid classifier to deep learning, in: IEEE Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2017), 2017, pp. 123–127.
[14] N. Moritz, J. Schröder, S. Goetze, Acoustic scene classification using time-delay neural networks and amplitude modulation filter bank features, in: IEEE Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2016), 2016, pp. 70–74.
[15] W.M. Darling, A theoretical and practical implementation tutorial on topic modeling and gibbs sampling, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:, Human Language Technologies, 2011, pp. 642–647.
[16] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[17] B. McFee, C. Raffel, D. Liang, D.P.W. Ellis, M. McVicar, E. Battenberg, O. Nieto, Librosa: Audio and music signal analysis in python, in: Proceedings of the 14th Python in Science Conference (SciPy 2015), 2015.
[18] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[19] A. Mesaros, T. Heittola, T. Virtanen, Tut database for acoustic scene classification and sound event detection, in: 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 1128–1132.
[20] S. Amiriparian, M. Freitag, N. Cummins, B. Schuller, Sequence to sequence autoencoders for unsupervised representation learning from audio, in: IEEE Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2017), 2017, pp. 17–21.
[21] S.H. Bae, I. Choi, N.S. Kim, Acoustic scene classification using parallel combination of LSTM and CNN, in: IEEE Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2016), 2016, pp. 11–15.
[22] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, H. Fujita, Multi-imbalance: An open-source software for multi-class imbalance learning, Knowl.-Based Syst. 174 (2019) 137–143.
[23] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization GAN for unbalanced data, Knowl.-Based Syst. 187 (2020) 104837.