Tampere University

SHUYANG ZHAO

Clustering Analysis and Active Learning for Sound Event Detection and Classification

Tampere University Dissertations 541

Tampere University Dissertations 541

SHUYANG ZHAO

Clustering Analysis and Active Learning for Sound Event Detection and Classification

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty of Information Technology and Communication Sciences
of Tampere University,
for public discussion in the auditorium RG202
of the Rakennustalo, Korkeakoulunkatu 5, Tampere,
on 19 January 2022, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University, Faculty of Information Technology and Communication Sciences
Finland

Responsible supervisor and Custos
Professor
Tuomas Virtanen
Tampere University
Finland

Pre-examiners
Senior researcher
Frederic Font
Universitat Pompeu Fabra
Spain

Associate Professor
Zhiyao Duan
University of Rochester
United States

Opponents
Senior researcher
Frederic Font
Universitat Pompeu Fabra
Spain

Assistant Professor
Karol J. Piczak
Jagiellonian University
Poland

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-2265-6 (print)
ISBN 978-952-03-2266-3 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-2266-3

PunaMusta Oy – Yliopistopaino
Joensuu 2022

ABSTRACT

The objective of the thesis is to develop techniques that optimize the performances of sound event detection and classification systems at minimal supervision cost. The state-of-the-art sound event detection and classification systems use acoustic models developed using machine learning techniques. The training of acoustic models typically relies on a large amount of labeled audio data. Manually assigning labels to audio data is often the most time-consuming part in a model development process. Unlabeled data is abundant in many practical cases, but the amount of annotations that can be made is limited. Thus, the practical problem is optimizing the accuracies of acoustic models with a limited amount of annotations.

In this thesis, we started with the idea of clustering unlabeled audio data. Clustering results can be used to derive propagated labels from a single label assignment; meanwhile, clustering itself does not require labeled data. Based on this idea, an active learning method was proposed and evaluated for sound classification. In the experiments, the proposed active learning method based on k-medoids clustering outperformed reference methods based on random sampling and uncertainty sampling. In order to optimize the sample selection after annotating the k medoids, mismatch-first farthest-traversal was proposed. The active learning performances were further improved according to the experimental results.

The active learning method proposed for sound classification was extended to sound event detection. Sound segments were generated based on change point detection within each recording. The sound segments were selected for annotation based on mismatch-first farthest-traversal. During the training of acoustic models, each recording was used as an input of a recurrent

convolutional neural network. The training loss was derived from frames corresponding to only annotated segments. In the experiments on a dataset where sound events are rare, the proposed active learning method required annotating only 2% of the training data to achieve similar accuracy, with respect to annotating all the training data.

In addition to active learning, we investigated using cluster analysis to group recordings with similar recording conditions. Feature normalization according to cluster statistics was used to bridge the distribution shift due to mismatched recording conditions. The achieved performance clearly outperformed feature normalization based on global statistics and statistics per recording.

The proposed active learning methods enable efficient labeling on large-scale audio datasets, potentially saving a large amount of annotation effort in the development of acoustic models. In addition, core ideas behind the proposed methods are generic and they can be extended to other problems such as natural language processing, as is investigated in [8].

CONTENTS

ACRONYMS

CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DCASE	Detection and Classification for Acoustic Scene and Events
FNN	Feedforward Neural Network
MAL	Medoids-based Active Learning
MAL-MF	Medoids-based Active Learning, Mismatch-first Farthest-traversal as second stage sample selection method
MAL-R	Medoids-based Active Learning, Recursively Clustering on Unlabeled Data
MFCCs	Mel-frequency Cepstral Coefficients
MFFT	Mismatch-first Farthest-traversal
PAM	Partition Around Medoids
RNN	Recurrent Neural Network
SED	Sound Event Detection
SVM	Support Vector Machine

ORIGINAL PUBLICATIONS

Publication I — P. Maijala, Zhao S.Y., T. Heittola and T. Virtanen. Environmental noise monitoring using source classification in sensors. *Applied Acoustics* 129.6 (Jan. 2018), 258–267.

Publication II — Zhao S.Y., T. Heittola and T. Virtanen. Active learning for sound event classification by clustering unlabeled data. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, 751–755.

Publication III — Zhao S.Y., T. Heittola and T. Virtanen. Learning vocal mode classifiers from heterogeneous data sources. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, 16–20.

Publication IV — Zhao S.Y., T. Heittola and T. Virtanen. An Active Learning Method Using Clustering and Committee-Based Sample Selection for Sound Event Classification. *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, 116–120.

Publication V — Zhao S.Y., T. Heittola and T. Virtanen. Active Learning for Sound Event Detection. *IEEE Transactions on Audio, Speech and Language Processing* 28 (2020), 2895–2905.

Author's contribution

Publication I	Zhao Shuyang is the second main author of this work. His contribution includes the data annotation, design and implementation of the experiments on noise source classification. The main author Panu Maijala is responsible for noise measurement system design and piloting the main concept.
Publication II	Zhao Shuyang is the main author of this work. His contribution includes the development of the main idea, the design of the experiments and the implementation of the proposed algorithms. The collaborators, Toni Heittola and Tuomas Virtanen, helped the main author with the choice of experimental datasets, the illustrations of proposed methods, and technical corrections.
Publication III	Same as Pubilcation II.
Publication IV	Same as Pubilcation II.
Publication V	Same as Pubilcation II.

1 INTRODUCTION

The perception of sounds is the fundamental of various human functionalities, such as communication and sensing dangers. Thus, computational analysis of sounds potentially helps automate many real-world tasks. Sound event detection and classification is a research area that deals with the problem of automatically recognizing sounds in audio signals. It has various applications, including health monitoring [34], road monitoring [30, 67] and machine inspection [55, 80].

The state-of-the-art approaches of sound event detection and classification depend on acoustic models developed using machine learning. The performance of a learned acoustic model largely depends on the training material, a collection of annotated audio signals. In most cases, capturing audio is easy, but annotating is time-consuming. Thus, the practical problem is to optimize the performance of learned acoustic models, with a limited amount of annotations being made. From another perspective, the problem is to minimize the supervision effort needed to achieve a reliable model. Reduced labor cost in model development potentially allows sound event detection and classification techniques to be deployed in more applications, where a sufficient number of labels for supervised learning are currently too expensive to obtain.

1.1 Objective of the thesis

The objective of the thesis is to develop techniques that optimize sound event detection and classification performance at minimal supervision cost. The

starting point is to perform cluster analysis on audio data, grouping similar sounds into clusters. The utilization of clustering results is investigated in two approaches. The first approach is active learning, which aims at selecting samples that are most beneficial to annotate. The idea of utilizing clustering results to help labeling large dataset is briefly introduced in Section 3.2, and the developed active learning methods following the idea are presented in detail in Chapter 4 and Chapter 5. The second approach is feature normalization. It is used to learn acoustic models using audio recorded under mismatched recording conditions, and the metadata about the recording conditions is not available. The method is introduced in Section 3.3.

Sound event detection and classification As a broad definition, a sound event refers to an audio segment that can be associated with a concept. The task of automatic sound event detection (SED) is to temporally locate the target sound events from an audio signal and associate a class label with each individual event. The target sound events can be largely different according to the specific applications. For example, clapping, coughing, knocking, and phone ringing are detected for health care monitoring in [34]. Speech recognition and music transcription can be considered as special cases of sound event detection. However, as the convention in the audio signal processing community, recognition of speech and musical notes are not referred to as sound event detection. Compared to SED, sound classification is relatively simple. It associates exactly one class to an audio signal, in case of single-label classification [42]. In case of multi-label classification [11], an audio signal could be associated with multiple classes. Multi-label sound classification is also called audio tagging [31].

Clustering analysis Cluster analysis or clustering is commonly used to find hidden structures in a dataset. This can be achieved with various algorithms such as K-means [51], K-medoids [81] and single-linkage clustering [94]. This thesis investigated using clustering methods to group similar sounds. Based on the clustering results, active learning and feature normal-

ization were investigated to minimize the supervision effort required to learn acoustic models.

Active learning The term active learning [90] is defined from the perspective of the learning algorithm. An active learning algorithm actively participates in the data collection process: the learning algorithm is allowed to select the data it learns from and make queries to a teacher, typically a human annotator. Active learning is typically used for optimizing learned models when unlabeled data is abundant, but the amount of annotations that can be made is limited. When affordable annotation effort can provide labels to only a small portion of unlabeled data, the selection of the subset may largely affect the performance of the learned model. Ideally, the active learning system learns from diverse training material that is relevant to the target problems.

Research questions In order to optimize the accuracy of a learned model with a limited amount of annotations, this study addresses the following research questions:

1. When only a limited number of labels can be assigned to abundant unlabeled data, how to evaluate the effectiveness of a machine learning method that deals with this problem?

2. How to measure audio similarity and perform cluster analysis based on it?

3. What is the most efficient way of utilizing clustering results in the development of acoustic models, in terms of minimizing supervision effort?

4. When a few segments are labeled in each recording, how to utilize the temporal information of the original recordings to learn acoustic models?

1.2 Main results of the thesis

The main results and contributions of the publications leading to this thesis are as follows.

Publication I: Environmental Noise Monitoring Using Source Classification in Sensors In publication [I], sound source classification was introduced to noise measurement sensors. In the first case study, the accuracy and computation time using the Gaussian mixture model and the neural network was analyzed. In the second case study, a clustering method was used to analyze a large amount of industrial noise data from a harbor. K-means clustering was performed on sound segments represented by means of their corresponding MFCC vectors. An annotator was used to examine randomly sampled sound segments within each cluster. When sampled sounds were from different event classes in a cluster, the cluster was split into smaller ones. When the sounds in a cluster belonged to the same class, sounds in the cluster were collectively annotated.

Publication II: Active Learning for Sound Event Classification by Clustering Unlabeled Data In publication [II], an active learning method was proposed for sound event classification based on K-medoids clustering. The distance measurement was based on Kullback-Leibler divergence between segments represented by the Gaussian distribution of MFCCs. The clustering analysis used in publication [I] sometimes requires a considerable amount of annotation effort to verify whether the clusters contain mixed classes of sounds. The active learning algorithm proposed in publication [II] used rather small clusters, and the sounds in a cluster were always assumed to belong to the same class. Medoids, the centers of each cluster, were selected for manual annotation. An annotated label associated with a medoid was propagated to other members of its cluster. Both the annotated labels and propagated labels were used for model training. The experimental results showed that the active learning algorithm proposed in publication [II] saved 50%-60% anno-

tation effort, with respect to the best reference method, on an environmental sound classification dataset.

Publication III: Learning Vocal Mode Classifiers from Heterogeneous Data Sources In publication [III], a feature normalization method was proposed for utilizing multiple external data sources to learn sound classification models when no training data is available for the target problem. The study suggested that the model directly learned from different datasets was not reliable. The distribution of log-mel band energies largely varies among different datasets, due to the difference in audio capturing setups, which has different frequency responses. The performance of the learned model was largely improved by normalizing the features to zero mean and unit variance for each dataset. In some of the datasets, multiple capturing setups were used. The study proposed using clustering to divide a dataset into subdatasets, each of which was normalized according to its distribution.

Publication IV: An Active Learning Method Using Clustering and Committee-based Sample Selection for Sound Event Classification In publication [IV], an active learning algorithm was proposed improving on the basis of publication [II]. After annotating the medoids, the sample selection continued with mismatch-first farthest-traversal. The primary selection criterion was the prediction mismatch on unlabeled sound segments between model prediction and label propagation. Counterexamples were selected from either the predictions of an existing model or the labels propagated from the medoids. The second criterion was the distance to previously selected segments. It aimed at maximizing the diversity of selected data. The experimental results on the same dataset showed that the accuracy of learned models clearly outperforms the methods presented in publication [II], when the labeling budget was larger than the number of medoids.

Publication V: Active Learning for Sound Event Detection In Publication [V], an active learning algorithm was proposed for sound event detection. In comparison to previous publications, the system takes original

recordings as input instead of short sound segments. The system comprises three main parts. Variable-length sound segments are generated as selection candidates based on a change point detection approach. Mismatch-first farthest-traversal proposed in publication [IV] is used to select sound segments for manual annotation. The distance measurement is based on the cosine distance between embedding vectors extracted with a pre-trained model. Weak labels are required in the annotation to indicate the sound event classes present in selected sound segments. During the model training, each original recording is used as an input, and the loss is derived from only the frames corresponding to annotated segments. Experimental results showed that the proposed system effectively saved annotation effort for two datasets. Particularly, the proposed system required annotating only 2% of the training set to achieve the same performance as annotating the whole training set.

1.3 Outline and structure of thesis

The organization of the rest of this thesis is as follows.

Chapter 2 introduces fundamental concepts in sound event detection and classification, including acoustic feature extraction and acoustic models. It also introduces the techniques that minimize the supervision costs in learning acoustic models.

Chapter 3 presents two studies, publication [I] and publication [III], that perform cluster analysis on audio datasets. In publication [I], clustering results are used to explore and annotate a noise monitoring dataset. When no information is initially available about an audio dataset, data exploration is referred to the process that the annotator understands the general characteristics and discovers the existing sound event classes in the dataset. In publication [III], clustering is used to group recordings that are captured in similar recording conditions. Feature normalization, according to cluster statistics, is used to bridge the feature distribution shift caused by mismatched recording conditions.

Chapter 4 presents two proposed methods that address the problem of ac-

tive learning for sound classification. Publication [II] presents an active learning algorithm based on K-medoids clustering. Publication [IV] proposes a mismatch-first farthest-traversal algorithm for active learning.

Chapter 5 presents publication [V] that extends the mismatch-first farthest-traversal algorithm to sound event detection, integrating with a weakly supervised learning method.

The discussion and conclusion of the thesis are given in Chapter 6 and Chapter 7, respectively.

2 BACKGROUND

The main objective of the thesis is to minimize supervision effort in learning acoustic models for sound event detection and classification, utilizing cluster analysis. This chapter gives the background information in three sections. The first section introduces the fundamentals of the machine learning approaches for sound event detection and classification. The second section introduces the techniques that are used to minimize the supervision effort in machine learning. The third section introduces techniques of cluster analysis, since the main objective of the thesis is to investigate using clustering to minimize supervision effort in the development of acoustic models for sound event detection and classification.

2.1 Sound event detection and classification

The task of automatic sound event detection (SED) [73] is to temporally locate sound events from an audio signal and associate each individual sound event to a class. A sound event is a recognizable acoustic activity [72], and sound event classes are terms used to describe sound events, such as “door slam”, “door barking”, and “water drops”.

Compared to SED, the definition of sound classification [42] is relatively simple. In single-label classification, exactly one class is associated with each audio signal. The classes are defined to be mutually exclusive; thus, multiple classes cannot be present simultaneously. In multi-label classificatoin [11], or audio tagging [31], an audio signal is associated with a set of classes, whose cardinality can be zero, one or more. In some cases [78, 85], an audio signal

is assumed to contain only one isolated sound event. The term sound recognition has been previously used for sound classification in [18]. The thesis uses the term sound classification for clarity.

2.1.1 Acoustic feature extraction

An audio signal is captured by a transducer that converts the time-varying pressure of a sound wave into electrical voltages, which are further sampled and quantized. The content analysis of audio signals is rarely performed on raw audio signals. In most cases, audio signals are transformed into compact and interpretable representations, called acoustic features. Commonly used acoustic features used for SED include mel frequency cepstral coefficients (MFCCs) [22], mel band energies [11], fundamental frequency [16], and embedding vectors [60].

Spectrum Spectrum is a frequency domain representation of an audio signal. Most audio content analysis methods are based on time-frequency representations [88], motivated by the structure of the human auditory system. In human ears, sound pressure fluctuation is transduced by hair cells that have different receptive frequency ranges, depending on their positions in cochlea [29]. The starting point of spectrum computation is typically discrete Fourier transform (DFT), which transforms a time series into a frequency domain. Since audio signals change over time in each recording, DFT is performed on the audio signals in short time frames, for example, 23 ms in [54]. As a result, a 1-D times series is transformed to a 2-D time-frequency representation. The outputs of DFT are complex values, and the magnitudes are commonly used for audio content analysis. The choice of the frame length is a trade-off between frequency resolution and time resolution. In a time frame, a windowing function such as Hanning window is typically used to smooth the signal at the ends of both sides. A time frame usually overlaps with the adjacent ones by 50% or 75% in sound event detection and classification.

Log-mel spectrum and mel frequency cepstral coefficients The results of DFT correspond to frequencies on a linear scale. However, according to a subjective listening test [95], the perceptual distance between two pitches is non-linear as the function of the frequency difference in hertz. Mel-scale (mel from melody) is a perceptually determined scale that measures the relative pitch differences. The Bark scale is another non-linear frequency scale based on subjective loudness [109]. It is more recently proposed but less popular in the field of audio content analysis.

In order to convert a linear scale spectrum into mel scale, triangular filterbanks are used. The amplitude of a filter peaks at the central frequency and linearly decreases along with the distance to the central frequency. The central frequencies of the triangular filters increase linearly in the mel-scale from low frequency to high frequency. As a result, mel band energies are a compressed representation of the linear scale spectrum, more densely compressed on a higher frequency. The logarithm operation is performed on mel band energies to reduce the dynamic range. The log-mel band energies are commonly used acoustic features for audio content analysis.

Log-mel band energies can be further processed into mel-frequency Cepstral coefficients (MFCCs) [22], more compact representations widely used in speech recognition [84], and sound classification [18]. To obtain MFCCs, discrete cosine transform (DCT) is performed on log-mel band energies, and the 0th coefficients are often discarded to obtain amplitude-invariant representations.

Embedding vectors Some recent studies [4, 21, 43, 60] show that time-frequency representations can be further processed into more distinctive and noise-robust representations based on machine learning. These types of representations are called embedding vectors. Embedding methods are used to represent data objects in a vector space [45]. One of the examples is word embeddings, representing each word as a vector [61]. Publicly available audio embedding extractors include SoundNet [4], VGGish [43] and OpenL3 [21].

2.1.2 Acoustic models

An acoustic model is a parametric model that maps acoustic features into predicted outputs, such as the class labels of present sound events. Previously, various types of acoustic models have been investigated, including Gaussian mixture model [18], support vector machine (SVM) [4, 85], random forest [85], and neural networks [11, 12, 69]. This section briefly introduces SVM and neural networks since they are used in the publications contributing to this thesis.

2.1.2.1 Support vector machine

SVM has been widely used for classification problems [13, 85, 98]. It consists of a set of binary classifiers, each of which constructs a hyperplane separating a pair of classes [63]. A hyperplane is optimized to separate the two classes and maximize the margin between the hyperplane and its nearest instances. SVM has been investigated for sound classification problems in [4, 78, 85]. It requires each sound to be represented as a high-dimensional vector. In [78, 85], a sound is represented by the statistics of MFCCs, such as mean and variance. In [4], a sound is represented by an embedding vector, extracted using a convolutional neural network.

The training process of SVM is deterministic since the optimization of SVM is a convex problem, and the global optima are unique. Due to this property, SVM is often used as a baseline system for sound classification, such as in [85]. However, SVM does not natively support multi-label classification.

2.1.2.2 Neural networks

A neural network is constructed by connecting artificial neurons, which are conceptually derived from biological synapses. Each artificial neuron has multiple inputs and produces a single output. By connecting artificial neu-

rons, neural networks can be used to approximate non-linear relationships between variables. Due to the flexibility in modeling, neural networks have been used in a wide range of problems, such as regression [10], classification [19], and dimensionality reduction [7]. The state-of-the-art SED models are based on convolutional recurrent neural networks (CRNN) [12]. One of the CRNN networks has been used in [V]. A CRNN consists of three types of neural networks as substructures: fully-connected network [64], convolutional neural network [6], and recurrent neural network [47].

Fully-connected neural networks A fully-connected network (FNN) [64, 69] consists of only fully-connected layers, where each neuron is connected with all the input variables, or hidden variables produced by the previous layer, as

$$\mathbf{h} = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}), \tag{2.1}$$

where the input is a vector $\mathbf{x}$ and the output is a vector $\mathbf{h}$. The matrix $\mathbf{W}$ is called weights and the vector $\mathbf{b}$ is called bias, or offset. A non-linear activation function is denoted as σ, which is an element-wise operation. Typical activation functions include rectified linear unit, sigmoid function and hyperbolic tangent. Sigmoid function is a monotonic function that maps a real number to between 0 and 1.

The learnable parameters, weights and biases, are initialized at the beginning of a training process and are iteratively optimized to match the model outputs with training targets. The number of learnable parameters of a fully connected network is large with high-dimensional input data. As a result, fully connected neural networks easily overfit some types of data, due to the excessively large number of parameters.

Convolutional neural networks A convolutional neural network (CNN) [6] consists of a series of convolutional layers. The input of a convolutional layer can be a vector, a matrix, or a tensor. Assuming the inputs are matrices, a small number of kernels are convolved with the input matrices in each convolutional layer. The convolution output with each kernel is called a channel.

Given an input matrix $\mathbf{X}$ and a kernel $\mathbf{W}$, the output of the corresponding channel is computed as

$$\mathbf{H}_{i,j} = \sigma(b + \sum_m \sum_n \mathbf{W}_{m,n} \cdot \mathbf{X}_{i-m,i-n}) \tag{2.2}$$

where σ is the activation function, typically rectified linear unit, and b is the bias of the convolutional kernel $\mathbf{W}$. Since a small number of convolutional kernels are shared over the entire input matrices, a CNN typically requires much fewer parameters to process a large matrix in comparison to using FNN. Max-pooling is commonly used between two convolutional layers. It downsamples the representation in each channel. This reduces the representation size while the information upstreams from bottom to top layers. Due to the nature of max-pooling, representations learned by CNNs are shift-invariant: delay or pitch shift on a sound event has minimal impact on its CNN representations.

Recurrent neural networks Recurrent neural network (RNN) [17, 47] consists of recurrent units. It is typically used to model sequential data. Given a sequence of N vectors as input $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$, each recurrent unit produces an output of the current time step $\mathbf{h}_t$, based on the input vector of current timestep $\mathbf{x}_t$ and the output of the previous time step $\mathbf{h}_{t-1}$. The output of recurrent units may carry information of previous timesteps and propagate the information forward. This enables RNNs to model the temporal patterns of sound events in long term. RNN has many variations, among which gated recurrent units (GRUs) [17] are used in [V]. With a GRU, the output corresponding to a timestep $\mathbf{h}_t$ is computed as

$$\mathbf{gate}_{reset} = \sigma(\mathbf{W}_0 \cdot \mathbf{x}_t + \mathbf{U}_0 \cdot \mathbf{h}_{t-1} + \mathbf{b}_0) \tag{2.3a}$$

$$\mathbf{gate}_{update} = \sigma(\mathbf{W}_1 \cdot \mathbf{x}_t + \mathbf{U}_1 \cdot \mathbf{h}_{t-1} + \mathbf{b}_1) \tag{2.3b}$$

$$\begin{aligned}\mathbf{h}_t =&(1-\mathbf{gate}_{update})\odot tanh(\mathbf{W}_2\cdot\mathbf{x}_t+\mathbf{gate}_{reset}\odot(\mathbf{U}_2\cdot\mathbf{h}_{t-1})+\mathbf{b}_2)\\ &+\mathbf{gate}_{update}\odot\mathbf{h}_{t-1},\end{aligned} \tag{2.3c}$$

where $\{\mathbf{W}_i|i=0,1,2\}$, $\{\mathbf{U}_i|i=0,1,2\}$ and $\{\mathbf{b}_i|i=0,1,2\}$ are the weight matrices and bias vectors. Element-wise multiplication is denoted with $\odot$. The reset gate and update gate are computed as in Equation 2.3a and Equation 2.3b, respectively. The reset gate decides what information is to be used from the previous timestep. The update gate decides what past information to propagate forward. As can be seen from Equation 2.3c, the information retained with the update gate can be interpreted as long-term memory, and the reset gate decides if the memory is taken into account when processing the current timestep.

Covolutional recurrent neural networks Convolutional recurrent neural networks (CRNNs) have been commonly used for audio tagging and SED in recent studies [12, 14]. CNNs are used to process time-frequency representations, typically log-mel band energies, into one or more sequences of latent representations. The latent representations are processed by RNNs to model long-term temporal patterns. The outputs of RNNs in each time step are further mapped into class probabilities using one or more FNNs.

2.2 Minimizing supervision effort

Acoustic models are typically developed based on supervised learning, where labeled audio data is used as training examples. Labels are typically obtained by annotation, manually assigning desired prediction outputs to audio signals. It is often the most time-consuming part in the model development process. In order to develop acoustic models with minimal cost of annotation effort, various techniques have been studied including domain adaptation [32], semi-supervised learning [108], active learning [II, IV], and weakly

supervised learning [68]. Domain adaptation allows utilizing labeled data that are already available in a source domain, developing acoustic models with none or few annotations made in a target domain. Semi-supervised learning deals with a partially labeled dataset. It utilizes not only labeled data but also unlabeled data in learning acoustic models. Active learning starts from an unlabeled dataset, aiming at selecting the optimal subset of the dataset for annotation, maximizing the learned acoustic model at the cost of a limited annotation effort. Weakly supervised learning utilizes annotations with less detailed information, typically referred to as training SED models with only presence/absence labels in recording level. This chapter describes and discusses these machine learning techniques that aim at minimizing supervision effort. The problem setups of these techniques are summarized in Table 2.1.

	Domain adaptation	Semi-supervised learning	Active learning	Weakly supervised learning
Utilizes data from external sources	Yes	No	No	No
Utilizes unlabeled data	Yes	Yes	Yes	No
Algorithm selects training data	No	No	Yes	No
Annotation with less details	No	No	No	Yes

Table 2.1 A summary of machine learning techniques that minimize annotation effort.

2.2.1 Domain adaptation

Domain adaptation [32] deals with the problem of distribution shift between training and testing datasets (domains). The distribution of a training dataset is called a source domain, whereas the distribution of the test dataset is called a target domain. By utilizing available data in source domains, domain adaptation can largely save the number of annotations required on a target domain. In the DCASE 2020 task 4, domain adaptation methods [104] have been shown effective in exploiting external synthetic data to improve the SED performance on a test dataset containing real-life recordings.

Feature normalization is a traditional approach to bridge the distribution differences. It has previously been investigated for robust speech recognition, for example, in [74], where the mean and variance of each feature variable is

estimated for each dataset, and the features in each recording are normalized to zero-mean and unit-variance according to the statistics of the dataset.

Recent studies [24, 104] perform domain adaptation using adversarial learning, which is widely used to disentangle confounding factors. The core idea is to train separate feature extractors for each domain. Thus the features extracted from different domains follow similar distributions. During the training, the extracted features from different domains are fed to a scene classifier and a domain classifier. The training target is to minimize the prediction loss from the scene classifier and maximize the prediction loss from the domain classifier.

2.2.2 Semi-supervised learning

In many cases, unlabeled data is abundant, but the amount of labeled data is rather limited. Semi-supervised learning techniques are used to optimize learned models by utilizing unlabeled data. An early attempt of semi-supervised learning on sound event detection [108] suggests that the performance of the learned model can be improved by simply using predicted labels as a training target for unlabeled data, although the improvement in performance is rather small. The semi-supervised learning method uses an iterative re-training process. In the first iteration, a teacher model is trained with labeled data, and the teacher model generates predicted labels to unlabeled data in the training set. A student model is trained using both the labeled data and unlabeled data with predicted labels. In the next iteration, the student model is used as the teacher, and a new student model is obtained.

Recently the challenge of DCASE 2018-2019 task 4 has attracted increasing research interest on semi-supervised learning for sound event detection. The top-performing system [103] is based on the mean-teacher method [96]. Compared to [108], the mean-teacher model is not obtained through re-training. It is obtained by taking the exponential moving average weights of previous student models. In addition, the unlabeled data is given a regularization role. A consistency cost is computed between the classification output on origi-

nal unlabeled data and unlabeled data imposed with noise. Since the ground truth labels should not change along with additive noises, the unlabeled data can help the trained model to be robust with noises. The original mean-teacher study [96] on an image classification dataset, CIFAR-10, shows that the performance training using 4000 labeled data can be achieved by using only 1000 labeled data with the mean-teacher method. The mean-teacher method has also been shown to achieve clearly better performance than using only labeled data in audio tagging [103].

2.2.3 Active learning

Similar to semi-supervised learning, active learning [20] also deals with abundant unlabeled data, given a limited number of labels that can be assigned. The main difference compared to semi-supervised learning is that an active learning algorithm is allowed to choose the data to be labeled, whereas the learning algorithm does not select the data to be annotated under a typical semi-supervised learning setup. The selection of the labeled subset may have a big impact on the performance of learned models. As an extreme case, nothing can be learned when select data all belong to the same class.

Previously, various active learning algorithms have been proposed for problems such as text classification [98] and speech recognition [37]. Uncertainty sampling [20] selects the samples that are classified with low confidence according to an existing model. Committee-based sampling [91] selects samples with a low level of prediction agreement among a group of models as a decision committee. These two methods attempt to select the samples where one or more existing models make mistakes, assuming counterexamples are more beneficial to improve an existing model. Diversity-oriented active learning [75] aims at high diversity in selected data points, covering local distributions in the dataset.

2.2.4 Weakly supervised learning

A SED model predicts each individual sound event, including the onset, offset and sound event class in each recording. Ideally, the labels should contain the same level of details. Labels of this type are called strong labels. In practice, precisely annotating the onset and offset of each individual sound event in an audio recording is time-consuming and sometimes difficult. In order to reduce the required annotation effort, weakly supervised learning techniques [58, 68] have been studied to utilize labels with lower levels of details. Two types of weak audio labeling have been identified in [102]. The first type is absence/presence labeling, which indicates the presence of each target sound event in each recording. The second type is sequential labeling, which indicates the order of present sound events without timestamps.

Most recent research has been done on the exploitation of absence/presence labeling since sequential labeling is not common in an existing audio dataset. An attention pooling [58] and an adaptive pooling method [68] based on softmax pooling have been shown effective. Attention pooling [58] methods learn to predict pooling weights, along with predicted probabilities for each frame. The weighted average of the class probabilities is used as the prediction for a sound segment. In [68], a learnable hyper-parameter is introduced to softmax pooling. Both methods achieved similar SED performance using weak labels compared to using strong labels.

2.3 Cluster analysis

Cluster analysis [53] is a general task of generating partitions in data. The aim of clustering is to achieve internal cohesion and external isolation according to a matter of interest [25]. Most clustering algorithms are based on inter-instance similarities, distances, or connectivities. The general term proximity is used for similarities, distances, or connectivities [25]. Cluster analysis has been previously used in a wide range of applications, including bioinformatics [101], astronomy [26], image classification [75], and community de-

tection in social network [28]. The clustering results are used for predicting local modularity structures [101] , building taxonomy [26], and minimizing suprevision effort in training classifiers [75].

A large variety of clustering algorithms have been previously developed. The algorithms have their own advantages and disadvantages, and the choice of the algorithm depends on the type of data and the usage of the clustering results. One category of clustering algorithms is based on optimization, either minimizing or maximizing a numeric criterion. Another category is hierarchical clustering, which generates a series of partitions, with smaller clusters merging into larger ones.

2.3.1 Clustering by optimizing objectives

A large number of clustering algorithms are based on optimizing a criterion, typically in terms of internal cohesion, or external isolation [25]. The clustering criteria are derived from either vector representations of each instance [23, 65] or a proximity matrix containing the dissimilarity values between each inter-instance pair [53, 76, 87].

K-means clustering is a widely used clustering criterion derived from vector representations. Based on the vector representations, each instance is regarded as a data point $\mathbf{x}$ in a vector space. K-means clustering divides the n data points into k clusters. The mean value μ_i of the data points in the ith cluster $\mathcal{S}_i$ is regarded as the cluster centroid. The optimization objective is to minimize the sum of Euclidean distances from each data point to its cluster centroid, as

$$\underset{\mathcal{P}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{S}_i} ||\mathbf{x} - \mu_i||^2 . \tag{2.4}$$

A partition with k clusters is denoted $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 ... \mathcal{S}_k\}$. The time complexity of the stardard K-means algorithm, Lloyd's algorithm [65], is $O(knd)$, where the k, n, d are the number of clusters, the number of instances and the dimensionality of the vector representations.

In many cases, instances are not represented as vectors, and only a proximity matrix, or an adjacency matrix in the case of graph data, is available. K-medoids clustering is a commonly used clustering criterion derived from a proximity matrix. It generates a partition of k clusters, each of which has an examplar instance as the cluster centroid. The examplar instances are called medoids, and each individual instance in the dataset is assigned to the cluster of its nearest medoid. The optimization objective is to minimize the sum of dissimilarities from each individual instance to its nearest medoid, as

$$\underset{\mathscr{M}}{\operatorname{argmin}} \sum_{i=1}^{n} min\{d(x_i, m_j) | m_j \in \{\mathscr{M}\}, \tag{2.5}$$

where $\mathscr{M} = \{m_1, m_2, ...m_k\}$ are the medoids of the k clusters. The time complexity of the standard optimization algorithm [53] is $O(n^2k^2)$. An improved algorithm can reduce the runtime to $O(n^2)$ [87].

The proximity matrix can be derived from vector representations by computing the proximity values based on a distance or a similarity function, such as Euclidean distance, cosine similarity, or a machine-learned similarity metric. This allows the choice of an arbitrary distance or similarity function that works well for a specific problem. Inversely, a proximity matrix can also be used to derive vector representations. In [41], each instance is represented by the corresponding row vector in the proximity matrix.

2.3.2 Hierarchical clustering

A hierarchical clustering algorithm generates a series of partitions instead of a single one. In each step, a partition is generated by either merging or splitting clusters obtained in the previous step. With a typical agglomerative hierarchical clustering, each individual instance is initialized as a cluster, and some of the clusters are merged in each step according to a rule, for example, single linkage [94]. To the opposite of agglomerative clustering, divisive clustering algorithms [35] start with a single cluster containing all the instances, and the clusters are divided into smaller clusters in each step.

Typically, hierarchical clustering is used to derive a dendrogram from analyzing a set of instances [25]. The dendrogram structure can be used for interactive data exploration [49, 89]. In comparison to K-means or K-medoids, hierarchical clustering does not optimize an objective. As a problem, the partition generated in any step of hierarchical clustering may not meet an objective at all, and what was done in previous steps could never be repaired [53].

3 CLUSTERING ANALYSIS FOR AUDIO DATASETS

Unlabeled data can be easily captured in an environment, where a sound classifier aims to operate. However, it requires a large amount of time to understand and annotate the data. Cluster analysis is commonly used for data exploration, or initial data analysis, by which an annotator can understand basic information about the dataset, including the correctness of the data and the general distribution of present sound events. In [I], clustering is used to explore noise monitoring data and to help annotating the noise sources in audio signals.

Clustering is used for sound event classification in a different approach in [II]. It is used to group recordings captured in similar conditions, when the labels of recording setups are unavailable. Then, feature normalization is performed according to cluster statistics to bridge the feature distribution shift caused by different recording conditions.

3.1 Related works

3.1.1 Audio similarity measurement

A clustering algorithm requires a method of similarity measurement. The similarity measurement is essential for the performance of clustering. Similarity measurement methods are commonly evaluated using information retrieval metrics such as mean average precision [70] and area under receiver

operating curve [27]. Three types of methods have been commonly used for the measurement of similarity between two sounds. One of the metrics is to measure the divergence between the distributions of MFCCs within each sound [66]. This similarity metric is referred to as MFCC-Gaussian-KL in this thesis. Another audio similarity measurement is based on dynamic time warping [79], which is an algorithm that measures the similarity between a pair of temporal sequences. Recent studies [50] achieve good audio information retrieval performances using cosine similarities between embeddings extracted by pre-trained models.

The requirement for the similarity measurement is similar to sound information retrieval [105] tasks. Ideally, the dissimilarities between inter-class pairs should be large, while the dissimilarities between intra-class pairs should be relatively small. Kullback-Leibler (KL) divergence between MFCC distributions is used to measure the dissimilarity between two sound segments in [II]. A sequence of MFCCs is extracted for each sound segment, and a sound is represented as a multi-variate Gaussian distribution based on the mean and variance of the corresponding MFCCs and their first and second-order deltas. The KL divergence between two Gaussian distributions $\mathscr{P}_0$ and $\mathscr{P}_1$ is calculated as

$$\begin{aligned} D_{\mathrm{KL}}(\mathscr{P}_0||\mathscr{P}_1) = \frac{1}{2}(&\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) \\ &+ (\mu_1-\mu_0)^\top \Sigma_1^{-1}(\mu_1-\mu_0) \\ &+ \ln(\frac{\det\Sigma_1}{\det\Sigma_0}) - k), \end{aligned} \tag{3.1}$$

where μ_0 and μ_1, Σ_0 and Σ_1 represent the means and the covariance matrices of two distributions, respectively. The sum of diagonal values of a matrix, or called the trace of a matrix, is denoted as tr. KL divergence is not commutative, thus $D_{\mathrm{KL}}(\mathscr{P}_0||\mathscr{P}_1)$ does not equal to $D_{\mathrm{KL}}(\mathscr{P}_1||\mathscr{P}_0)$. The average of both way KL divergence is commonly used to measure the dissimilarity between

two segments as

$$D(\mathscr{P}_0||\mathscr{P}_1) = D(\mathscr{P}_1||\mathscr{P}_0) = \frac{D_{\mathrm{KL}}(\mathscr{P}_0||\mathscr{P}_1) + D_{\mathrm{KL}}(\mathscr{P}_1||\mathscr{P}_0)}{2}. \quad (3.2)$$

The short-term temporal information is kept in the deltas of MFCCs; however, the long-term temporal information is lost using KL divergence. The combination of KL divergence and dynamic time warping (DTW) leads to an improvement in sound information retrieval performance compared to using only KL divergence in [105]. DTW is a method that calculates an optimal match between two sequences with possibly different lengths. It has also been widely used to measure the similarities for speech [38], and satellite images [77]. Recently, using cosine distance between embeddings extracted by a neural network pre-trained using very large-scale YouTube data has been shown effective in general sound retrieval tasks [50]. This similarity measurement is used in [IV].

3.2 Investigating noise monitoring data with cluster analysis

Environmental noise monitoring systems continuously measure sound levels. Publication [I] proposed a concept of automatically assigning measured sound levels to different noise sources, by running sound classification algorithms in a wireless sensor. Two case studies were made monitoring a rock crushing station and a harbor.

3.2.1 Acoustic model development using traditional supervised learning

In one case study, environmental noise was monitored near a rock crushing site using wireless sensors. A binary sound event classifier was used to predict

whether the noise source was from the rock crushing site in each recording, which has a duration of one minute. Within each recording, an annotator is used to mark the onset and offset of rock crushing machine sounds. Acoustic models are trained to predict the probability of the target noise source being active, per frame.

Traditional supervised learning was used to learn acoustic models in this study case. Two days of data were manually annotated. The annotated data was used for two-fold validation, swapping the data of day 1 and day 2 for training and testing. One of the trained acoustic models was deployed in wireless sensors for the concept pilot. The prediction results from the acoustic model well matched with the working schedule of the rock crushing site in general. This case study is not described in detail since it is not relevant to the main topic of this thesis.

3.2.2 Acoustic model development with interactive clustering

In the second case study, environmental noise was monitored at a few locations in a harbor. Continuous recordings had been collected for months, and prior knowledge about the harbor was unavailable to the annotator. The target was to analyze the types of noise sources in the environment and to develop acoustic models for the identified noise sources. A clustering approach was used to analyze the initially unlabeled dataset. The recordings were first split into audio segments based on Bayesian information criterion [59], and the segments were partitioned into ten clusters based on K-means clustering.

In k-means clustering, each instance is represented as a feature vector, and arbitrary points in the feature space can be used as centroids. MFCCs of a sound is a sequence of vectors; thus, it cannot be directly used for k-means clustering. In [41], a dissimilarity matrix is generated based on MFCC-Gaussian-KL [66], and each row vector in a dissimilarity matrix is used as a vector representation of the corresponding instance. This approach has been used in [I] and [II]. The weakness is the computation cost for large datasets. The

time complexity of k-means is $O(knd)$, and it is $O(kn^2)$, when row vectors of dissimilarity matrix are used as the vector representations of the instances.

In each cluster, audio segments were randomly sampled, and the samples were presented to an annotator. The annotator listened to a small fraction of sound segments randomly sampled from each cluster. When the samples in a cluster were found to belong to the same class, the whole cluster was collectively annotated as the samples' majority class. When the samples in a cluster were from multiple classes, the annotator would decide whether the cluster was further partitioned into 10 clusters or skipped. The annotated segments were used to develop acoustic models as was used in traditional supervised learning.

Due to the lack of ground truth labels, careful evaluation could not be made with the noise monitoring data. Following the idea of utilizing clustering results in the development of acoustic models, the problem of active learning for sound classification was formalized and studied in [II] and [IV].

3.3 Cluster analysis for feature normalization

The distribution of acoustic features largely depends on the recording setups, including the recording device, the acoustic space, and the background noises. The distribution shift of acoustic features caused by mismatched recording setups has been previously dealt with various techniques such as domain adaptation [32], which have been studied to apply models trained in a source domain to a target domain. However, a dataset may contain recordings that are captured under multiple recording setups, and the information about the recording setup of each recording is not available. To deal with these cases, [II] proposes to divide a dataset into clusters and perform feature normalization according to the feature statistics of each cluster.

3.3.1 A problem of vocal mode classification

A vocal mode classification problem is studied in [II], where the target is to distinguish singing from speech. It is expensive to collect speech and singing data using the specific recording setup of the target application scenario. Thus, it would be of great importance to utilize a large amount of speech and singing data that are publicly available. Many speech recognition datasets are publicly available, including [5, 36, 56]. Singing recordings are publicly available in the vocal tracks in some multitrack music datasets, including [9, 52]. The problem is to learn vocal model classification models utilizing audio data from multiple datasets and apply the models for target recording setups.

3.3.2 Featrue normalization techniques

The distributions of acoustic features, mel-band energies, of two speech recognition datasets are visualized in Figure 3.1. Both datasets have balanced speaker genders and balanced phonemes used in the utterances. As can be seen, an energy level of a band can be relatively low in one of the datasets but high in another.

In order to bridge the distribution shift, feature normalization is used according to the dataset statistics. Two feature normalization method mean-variance normalization [100] and quantile equalization [44] has been investigated. In mean-variance normalization, the mean μ and variance σ^2 are computed for each dataset. An acoustic feature $\mathbf{x}$ is shifted by the mean and scaled according to the variance, as

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - \mu}{\sigma}. \tag{3.3}$$

Quantile equalization [44] estimates a transformation function for each feature coefficient based on the quantile statistics. Quartiles are a group of quantiles, consisting of the minimum, 25th percentile, median, 75th percentile, and maximum values of a feature coefficient, denoted as Q_0, Q_1, Q_2,

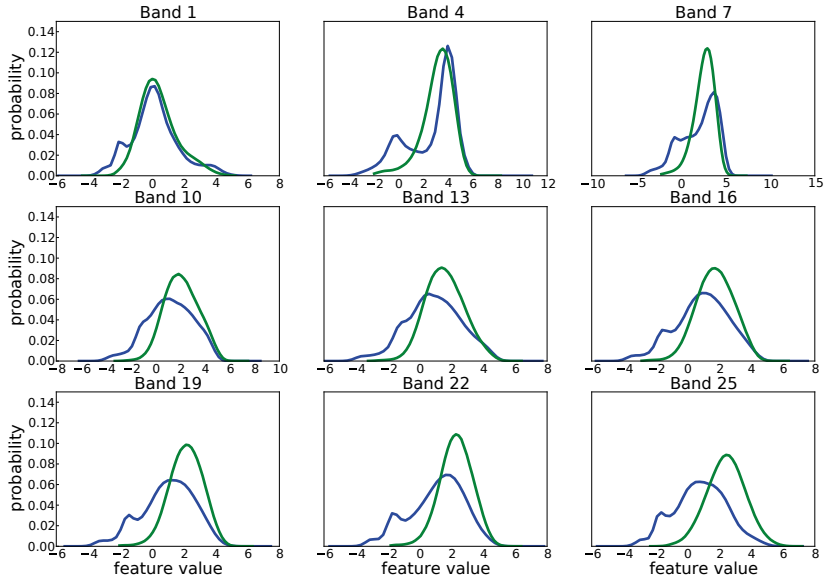

Figure 3.1 Feature distribution in CHiME2010 and Arctic dataset,illustrated in green and blue lines, respectively. The visualized features are log-mel band energies from nine different bands. The histogram plots are obtained by dividing the interval $[-4\sigma, 4\sigma]$ of each feature coefficient into 50 bins. ©2017 IEEE.

Q_3, Q_4, respectively. They divide the range of a feature coefficient value into four bins: $\mathscr{B}_0 = [Q_0, Q_1]$, $\mathscr{B}_1 = (Q_1, Q_2]$, $\mathscr{B}_2 = (Q_2, Q_3]$, $\mathscr{B}_3 = (Q_3, Q_4]$. In [II], quantile equalization is used to normalize the features based on the quartiles, using the transformation function

$$x_{norm} = \hat{Q}_k + (x - Q_k)\frac{\hat{Q}_{k+1} - \hat{Q}_k}{Q_{k+1} - Q_k} \quad x \in \mathscr{B}_k, \tag{3.4}$$

where x is the original feature coefficient value that falls into $\mathscr{B}_k$, and the normalized feature coefficient value is denoted as x_{norm}. The kth quartile of the source distribution is denoted as Q_k, and the kth quartile of the target distribution is denoted as $\hat{Q}_k$. In order to bridge the distribution shift, feature vectors from different source distributions are transformed to have the same target distribution.

3.3.3 Feature normalization according to cluster statistics

In some cases, multiple recording setups might be used in a dataset. As is shown in Figure 3.2, the two recordings from Arctic speech recognition dataset [56] seems to have different cut-off frequencies and different levels of background noises. In these cases, feature normalization according to dataset statistics would be less effective. A straightforward approach is to normalize the features according to the statistics of each recording. However, the diversity of sounds in a single recording is limited, and the feature distribution in a recording is mainly affected by the sound sources instead of the recording setup.

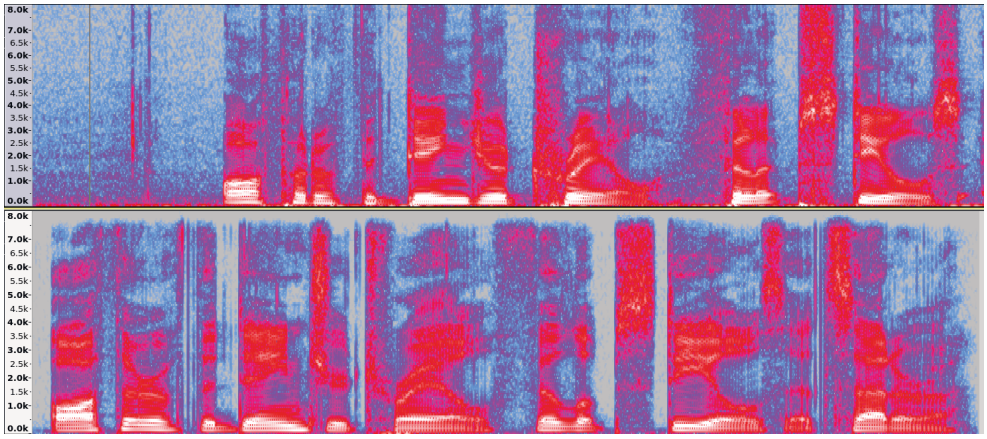

Figure 3.2 The spectrograms of two recordings in Arctic dataset.

In order to have a larger data scope for the feature normalization, K-means clustering [51] is used to divide each dataset into clusters, aiming at grouping the recordings based on the recording setups. Feature normalization is then performed according to cluster statistics. The number of clusters is defined proportionally to the total duration of each dataset, about two hours of non-silent audio per cluster.

3.3.4 Experimental results

A dataset TUT-vocal-2016 containing both speech and singing was collected. The dataset contains 80 pieces of singing recordings, from 20 volunteers. The lyric of each piece of singing is read out by the volunteer. The total duration was three hours and 15 minutes. The test dataset was split into four folds, and cross-validation was performed to evaluate the performance of acoustic models learned with in-domain data, recorded with matched condition with test data. The obtained accuracy across the four folds was 95.5%.

In order to evaluate the performance of acoustic models learned using only external training material, seven publicly available datasets, four speech datasets, and three singing datasets, about 35 hours in total, were used for training. Before learning acoustic models, feature normalization was performed based on the feature statistics. Since the results obtained with mean-variance normalization and quantile equalization were similar, only the results using mean-variance normalization are mentioned below. When the global statistics were used for feature normalization, the accuracy was only 69.6%. The accuracy was improved to 81.1%, when the acoustic features were normalized according to dataset statistics. When datasets were divided into clusters and feature normalization was performed according to cluster statistics, the accuracy was further improved to 96.8%. When the feature normalization was performed according to the feature statistics of each recording, the achieved accuracy was 72.7%. The diversity of a single recording is limited, and the feature statistics can hardly be used to estimate the distribution shift between recording setups.

Compared to 95.5% accuracy obtained by using in-domain data, similar performance (96.8%) is achieved using only external data with the proposed feature normalization method. It shows that, in some cases, the proposed method enables learning acoustic models from different data domains without requiring domain labels. The limitation of the method is that it requires a sufficient amount of data under each recording setup to estimate the distribution shift between recording setups.

4 ACTIVE LEARNING FOR SOUND CLASSIFICATION

Active learning is a special case of machine learning, where the learning algorithm is allowed to choose the data from which it learns [90]. In most cases, active learning targets the situation where unlabeled data is abundant, but the amount of annotations that can be done is limited. Previously, active learning has been studied for text classification [48, 98], speech recognition [37], and image classification [75]. This chapter describes the active learning methods proposed for sound classification in [II, IV], and the evaluation results on the two methods are collectively presented.

4.1 Problem definition

In general, three problem scenarios have been outlined for active learning: membership query synthesis [3], stream-based selective sampling [20] and pool-based sampling [62]. In the scenario of membership query synthesis, a learning algorithm is allowed to query for labels on an arbitrary point in a feature space. The learning algorithm should be able to generate data instances for annotation, corresponding to the selected data point. This is reasonable for some types of data, such as text [107]; however, it is not always possible to generate realistic sounds based on an arbitrary type of feature vector. Stream-based selective sampling assumes that data is continuously collected, and the learning algorithm decides whether to learn from each data instance on the fly. A data instance is either selected for annotation or discarded. A typi-

cal scenario is to improve a model that operates on a large scale of incoming data such as Tweet data streams in [99]. Among the three problem scenarios, pool-based sampling is the most widely studied. In pool-based sampling, a large pool of unlabeled data is provided in the beginning, and no incoming data instance is added afterward. Data instances not being selected at a time are not discarded, and they are used as selection candidates in the future.

In the noise monitoring scenario discussed in Section 3.2, a large amount of unlabeled data is collected before the model development. This situation falls into the category of pool-based sampling. In addition, most of previous active learning studies [37, 48, 75, 98] deal with pool-based sampling. Thus, pool-based sampling is assumed in this thesis. In principle, after a model is developed with the initial dataset and being deployed, stream-based active learning can be performed to further improve the model. However, this scenario is not addressed in this thesis.

In pool-based sample selection, an initially unlabeled dataset containing N sound instances is denoted as $\mathscr{S} = \{s_1, s_2, ...s_N\}$. A data instance is also called a sample, and selecting samples to be annotated is called sampling. An annotator is used to assign labels to the selected samples. The set of classes that can be assigned is pre-defined as $\mathscr{C} = \{c_1, c_2, ...c_C\}$, where C is the total number of classes. A label $l = (s, c) \in \mathscr{S} \times \mathscr{C}$ associates class c to the sound segment s. The maximum number of samples that can be manually annotated is called *labeling budget*. An active learning process finishes when the labeling budget is exhausted. An acoustic model is trained using all the obtained labels. The performance of the model is used to evaluate the outcome of the learning process. The performance is benchmarked as a function of the labeling budget; thus, different active learning algorithms can be compared either by the performances with a given labeling budget or the labeling budget required to achieve a given performance.

In an active learning process, annotated samples are denoted as a set $\mathscr{A}$, and the set of samples not being annotated are denoted as $\mathscr{U}$. Some active learning algorithms [39] incorporate the idea of semi-supervised learning. In addition to the annotated labels, predicted labels are generated to $\mathscr{U}$, and the

predicted labels are used in the training of acoustic models. In these cases, the annotated labels are denoted as $\mathscr{L}_{\mathscr{A}}$, and the predicted labels are denoted as $\mathscr{L}_{\mathscr{U}}$.

4.2 Related works

4.2.1 Uncertainty Sampling

A "query strategy" is used to define a rule of selecting data for annotation. Uncertainty sampling is a simple and commonly used query strategy. The idea is to query for labels on data instances where an existing model predicts with low confidence. The sample selection process is typically in batch mode [48]: a batch of samples is selected for annotation in each iteration, and the existing model is re-trained with the annotated samples $\mathscr{A}$. This idea requires a model to assign a confidence score to each prediction. The confidence measurement is straightforward for probabilistic models. Confidence estimation methods are available for many commonly used classification models such as SVM, decision tree, and neural network.

One of the problems with uncertainty sampling is known as cold start [92]. The estimation of certainty is not reliable unless an existing model is trained with a decent amount of labeled data. In many cases, uncertainty sampling does not outperform random sampling with a relatively low labeling budget [37, 82]. The low diversity within each selection batch is another problem for uncertainty sampling since the samples uncertain to the same model are often similar [86, 92].

4.2.2 Committee-based sampling

Committee-based sampling [91], or query by committee, relies on multiple models as a decision committee. Instances are considered to contain more information value to existing models when the models make mismatched pre-

dictions [91], since the predictions are wrong for at least some of the classifiers in the committee. Typically a classifier benefits more from a counterexample, where it makes mistakes, rather than an example where it succeeds.

The performance of committee-based sampling largely depends on the choice of the committee members. In ideal cases, the committee members should be able to correct each other. Committee-based sampling is not effective when the committee members use similar prediction mechanisms and always make the same decisions, or some committee members are constantly inferior to the others.

Since committee-based sampling involves training multiple classifiers, the computation time is typically larger than uncertainty sampling. Similar to uncertainty sampling, Committee-based sampling also has a cold start problem and the low diversity problem within each selection batch.

4.2.3 Cluster-based sampling

At the very early stage of an active learning process, labeled data is too limited to train a reliable acoustic model. Cluster-based sampling aims at selecting the representatives of each local distribution of a dataset. Since it does not rely on labeled data, cluster-based sampling has no cold start problem. It has been shown effective in image classification [75] when the labeling budget is small.

4.3 Medoid-based active learning for sound classification

In [II], an active learning method is proposed for sound classification. The algorithm is based on clustering results obtained with K-medoids clustering; thus, it is called medoid-based active learning (MAL). K-medoids clustering is performed on an initially unlabeled dataset, producing K instances as medoids, the centroids of K clusters. The medoids, as representatives of each local distribution, are presented to an annotator, ordered by the size of

clusters, largest first. The annotated label assigned to a medoid is propagated to other cluster members. Both annotated labels $\mathscr{L}_{\mathscr{A}}$ and propagated labels $\mathscr{L}_{\mathscr{U}}$ are used for training classifiers.

4.3.1 Sample selection based on k-medoids clustering

K-medoids clustering algorithm is commonly used to find exemplar instances, called medoids, in a dataset. The optimization objective is to minimize the sum of distances from each instance to its nearest medoid. In [II], the medoids are initialized using farthest-first traversal [83], which is a commonly used approximation of k-center problems [46]. In the preliminary study, initializing the medoids with farthest-first traversal leads to more consistent and generally better active learning performance, compared to using randomly initialized medoids. The optimization of the medoids is based on PAM [53].

4.3.1.1 The choice of clustering method

Compared to k-means, which is used in [I, III], k-medoids clustering has two advantages. Firstly, the centroid of each cluster is a real data instance, which is intrinsically the exemplar instance of the whole cluster. In comparison, the cluster centroids with k-means clustering are arbitrary points in the feature space. Secondly, k-medoids clustering directly uses a proximity matrix as an input, whereas k-means requires instances to be represented as vectors. In [I,II,III,IV], the sound segments are not intrinsically represented as vectors. The row vectors in the MFCC-Gaussian-KL dissimilarity matrix are used as vector representations of each data instance in [I, III]. Since the dimensionality of the vector representation is the number of instances, the computational cost of k-means clustering is high for a large dataset.

4.3.1.2 The use of the clustering results

The clusters obtained with k-medoids clustering are sorted based on the size, from the largest to the smallest. The idea is that typical cases of each class are first annotated, and then outliers are annotated only when the labeling budget is large enough. The labels annotated on each medoid are propagated to other members in its cluster to obtain propagated labels $\mathscr{L}_{\mathscr{U}}$. When the labeling budget is more than K, another round of clustering is performed on $\mathscr{U}$, and the labeling process is continued with medoids in the latest round of clustering. After all the labeling budget is consumed, a classifier is trained using both annotated labels $\mathscr{L}_{\mathscr{A}}$ and propagated labels $\mathscr{L}_{\mathscr{U}}$.

An imaginary dataset is given as an example of a pool-based sampling scenario in Figure 4.1. The imaginary dataset has 120 randomly generated instances from two classes. The ground truth labels are initially unknown. A labeling budget allows the active learning algorithm to query for labels of a limited number of samples from an annotator. Figure 4.2 visualizes the selected samples from the imaginary dataset Figure 4.1 using the MAL algorithm proposed in [II].

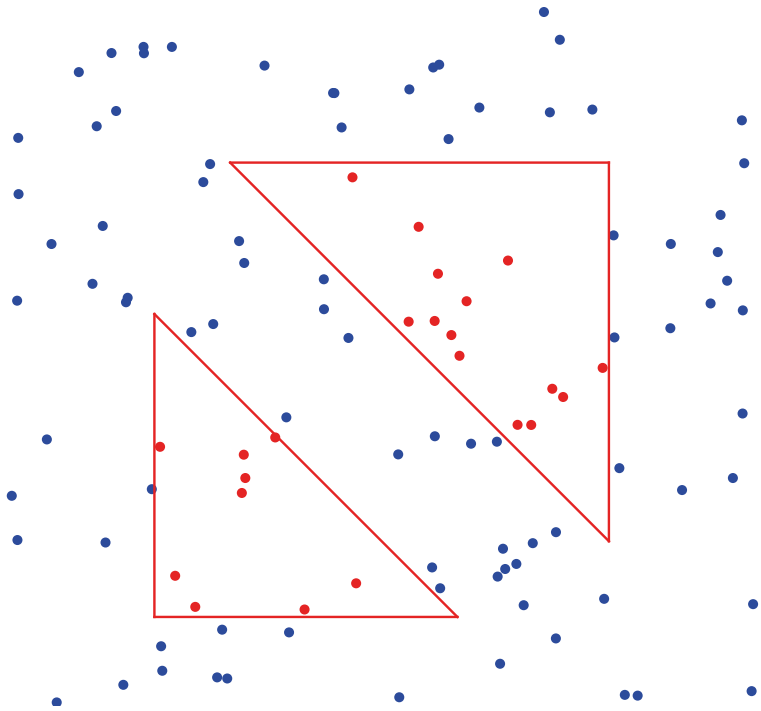

Figure 4.1 An imaginary binary classification problem, with 120 randomly generated instances on a 2-D space. The color represents the ground truth class of each instance. The red border marks the ground truth decision boundary between the two classes.

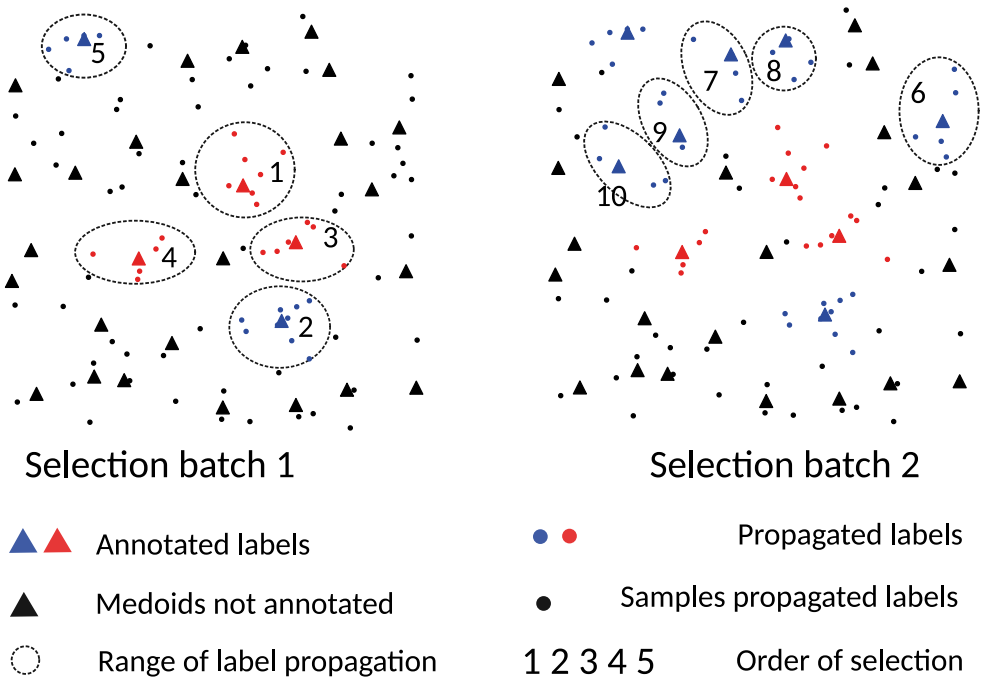

Figure 4.2 An illustration of the medoid-based active learning (MAL) algorithm on an imaginary dataset, shown in Figure 4.1. Medoids, illustrated as triangles, are produced based on k-medoids clustering. In each batch of sample selection, medoids of the largest clusters, the top five in the example, are selected for annotation. The label annotated to a medoid is propagated to other members in its cluster.

4.3.1.3 Choosing the number of clusters

The choice of the number of clusters K controls a trade-off between cluster size and the accuracy of the propagated labels: the bigger the cluster size, the more propagated labels can be derived from a single label assignment, but less accurately. A factor M is defined as the average cluster size, as $M = N/K$. In [II], M is fixed to four, based on a preliminary study on a small-scale dataset. In [IV], the factor M is denoted as KI. Due to possible confusions with K, the notation is changed to M in this thesis.

4.3.2 Limitations

MAL has been shown effective for sound classification under a low labeling budget. However, as the labeling budget grows, the performance is suboptimal since repeating k-medoids clustering on unlabeled data does not utilize previously annotated data. In addition, the number of clusters estimated in the preliminary study is unlikely to be optimal for all the cases.

4.4 Extending medoid-based active learning with mismatch-first farthest-traversal

In order to optimize the sample selection after annotating the medoids, mismatch-first farthest-traversal (MFFT) is proposed in [IV]. The active learning algorithm proposed in [IV] comprises two stages. The first stage is the same as in [II]: k-medoids clustering is performed, and the medoids are selected for annotation. After annotating all the medoids, each instance has a label, either annotated or propagated. The sample selection in the second stage, MFFT, aims at correcting wrong propagated labels, meanwhile maximizing the diversity of the selected samples. For the sake of simplicity, the method proposed in [II] is called MAL-R, which means medoid-based active learning, recursively clustering on unlabeled data. The method proposed in [IV] is called MAL-MF, which means medoid-based active learning, with mismatch-first farthest traversal as second stage sample selection method.

4.4.1 Mismatch-first farthest-traversal

MAL intrinsically involves two classification mechanisms: label propagation based on the clusters and a model-based classifier trained with labeled data. The primary selection criterion of MFFT is the prediction mismatch between the two mechanisms. Let us denote the propagated label of a sample x as $l_p(x)$ and the model-predicted label of x as $l_m(x)$. The set of samples with

prediction mismatch is defined as

$$\mathscr{M} = \{x \in \mathscr{U} | l_p(x) \neq l_m(x)\}, \tag{4.1}$$

where $\mathscr{U}$ is the set of samples not being annotated. The samples with prediction mismatch have either wrong model-predicted labels or propagated labels. As is assumed in committee-based sample selection, [91], a classifier benefits more from a counterexample, where the classifier makes mistakes, rather than an example where the classifier succeeds. Since the two classification mechanisms are fundamentally different, they typically have a decent number of samples with mismatched predictions to correct each other. The secondary criterion is the distance from an instance to its nearest previously selected sample. The selected sample s is defined as

$$s = \underset{x \in \mathscr{M}}{\operatorname{argmax}}\, d(x, \mathscr{S}). \tag{4.2}$$

The distance from a sample x to the set of samples $\mathscr{S}$ is defined as $d(x, \mathscr{S}) = min_{y \in \mathscr{S}} d(x, y)$. It aims at maximizing the diversity of selected samples.

Figure 4.3 visualizes the first selection batch with MFFT. It is based on the imaginary dataset shown in Figure 4.1, and the clustering results are shown in Figure 4.2. After all the medoids are annotated, a batch of five samples is selected among the samples with prediction mismatch.

4.4.2 Estimating the number of clusters

In addition to the MAL-MF algorithm, a method of estimating the cluster number K is proposed in [IV]. Before the clustering, the sample with median nearest neighbor distance among the dataset is selected as a test sample. The annotator is used to check the neighbors of the test sample, from the nearest to the farthest, until finding a neighbor that belongs to a different class. The number of top nearest neighbors that belong to the same class as the test sample is an estimate of the average number of propagated labels that can be reliably derived with a single label assignment. Thus, it is used as the average

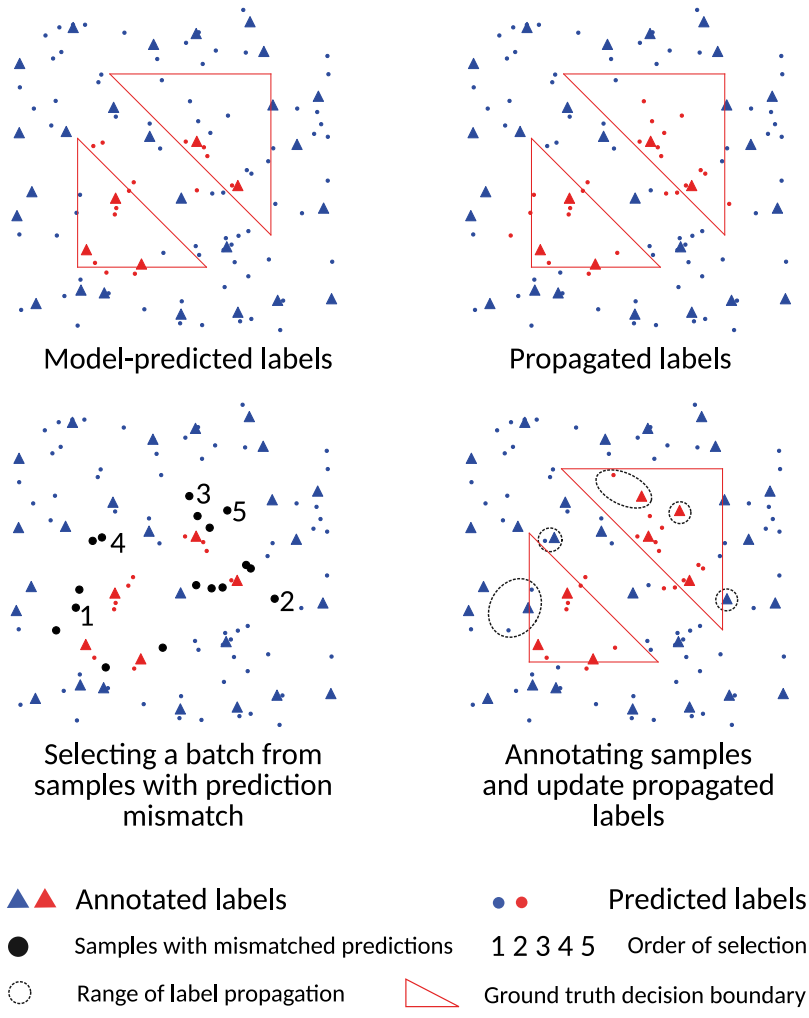

Figure 4.3 An illustration of mismatch-first farthest-traversal (MFFT) after all the medoids in Figure 4.2 are annotated. Model-predicted labels are generated based on a classification model trained with annotated labels, and propagated labels are generated based on nearest-neighbor prediction from annotated labels. A batch of samples with mismatched predictions are selected based on farthest-first traversal. The propagated labels are updated after selected samples are annotated.

cluster size M.

4.4.3 Limitations

The method of estimating K lies in the assumption that the instances of each class are generally balanced. In case that one of the classes comprises a very high proportion of the dataset, for example, 99% of the dataset, the estimated M would be very large. As a result, instances of rare classes can hardly be selected. In addition, the proposed MFFT algorithm is restricted to single-label classification.

4.5 Evaluating active learning algorithms in sound event classification

4.5.1 Dataset and settings

UrbanSound8K dataset [85] is used for evaluation. It is a public environmental sound dataset recorded in real urban environments. It contains 8 732 labeled sound segments, with a total duration of 8.75 hours. The sound segments are manually labeled into ten classes, including air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music.

The evaluation follows the 10-fold cross-validation setup originally proposed in UrbanSound8K. When testing the active learning performance on each fold, a model is trained using the other nine folds. The ground truth labels are initially hidden to the active learning system, which is allowed to query for labels up to the number of a pre-defined labeling budget. The annotated labels on queried segments are simulated according to the ground truth. After consuming all the labeling budget, a model is trained based on both annotated labels and propagated labels. The model is used to perform sound classification on the fold left out for testing. The unweighted accuracy of the predictions averaging the ten folds is used for evaluation. The classi-

fication model and the extraction of acoustic features follow the baseline of UrbanSound8K dataset.

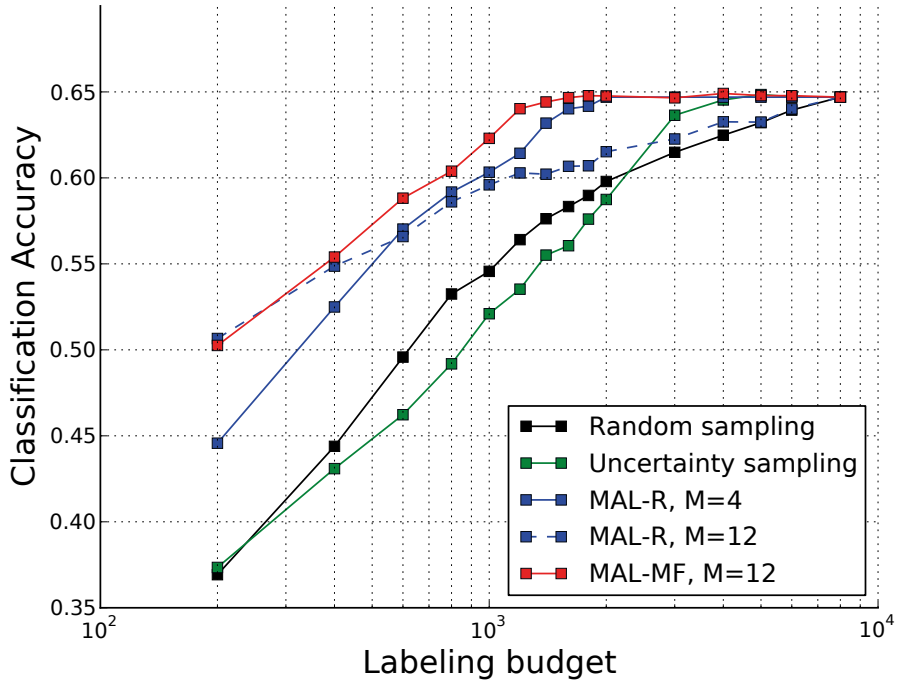

Figure 4.4 Classification accuracy as a function of labeling budget. MAL-R denotes the medoid active learning, recursively running k-medoids clustering after annotating medoids. MAL-MF denotes medoid-based active learning and mismatch-first farthest-traversal. ©2018 IEEE.

In order to evaluate an active learning algorithm with varying labeling budgets, the sound classification accuracy is evaluated as a function of the labeling budget. M is defined as N/K, which can be interpreted as the average size of clusters. The number of M used in [II] is 4, based on a preliminary study on a smaller dataset, whereas the number of M used in [IV] is 12, based on the proposed estimation method of M. In order to compare the performances, MAL-R and MAL-MF are evaluated with both $M = 4$ and $M = 12$. In addition to the proposed methods, random sampling and uncertainty sampling are evaluated as reference methods.

4.5.2 Experimental results

The experimental results are shown in Figure 4.4. The ceiling performance is 64.7%, when all the segments in the training dataset are annotated. MAL-R and MAL-MF outperform random sampling and uncertainty sampling to a large extent. Due to the cold start problem, uncertainty sampling does not outperform random sampling with a labeling budget less than 3000, approximately 38% of the training data. The MAL-R method proposed in [II] can save 50% to 60% annotations to achieve the same accuracy, with respect to random sampling or uncertainty sampling. The MAL-MF method proposed in [IV] can save 50% to 80% annotations to achieve the same accuracy, with respect to random sampling or uncertainty sampling.

Since the first stage is the same in MAL-R and MAL-MF, the performances with a labeling budget under K are almost the same. When $M = 4$ is used, the performances are very close for MAL-R and MAL-MF, since the performance already approximates the ceiling performance when the labeling budget is 2000, roughly the number of K. In order to keep clear looking in the figure, only the results with MAL-R are presented with $M = 4$. When $M = 12$ is used, MAL-MF clearly outperforms MAL-R when the labeling budget is over K, which is around 650.

5 FROM SOUND CLASSIFICATION TO SOUND EVENT DETECTION

Active learning has previously been studied for sound classification problems in [II, IV]. It is extended to SED in [V]. There are two fundamental differences between the active learning setups in [II, IV] and in [V]. Firstly, the learning outcome of [V] is a SED model, which predicts the onset and offset of each individual sound event, whereas exactly one class is associated with each sound segment in [II, IV]. Secondly, the system in [V] deals with relatively long audio signals, containing an arbitrary number of sound events, possibly overlapping in time. In comparison, [II, IV] deals with short segments, with a duration lower than four seconds, assuming each segment to contain only one isolated sound event. However, obtaining sound segments with only isolated events is not easy in real-life environments.

5.1 Basic ideas for minimizing supervision effort in learning SED models

Two aspects of annotation cost are considered in learning SED models: the duration of audio to be annotated and the difficulty of making the annotations. In general, the idea is to minimize the annotation cost from these two aspects. Following this thread, the basic ideas are introduced below.

5.1.1 Annotation unit

An annotation unit is an audio signal, within which an annotator is instructed to annotate target sound events. Traditionally, audio annotation is done on each recording. Since sounds in a recording are often produced from the same sound sources, annotating only representative segments within each recording is sometimes sufficient for acoustic model training. When the labeling budget is limited, annotating selected sound segments from different recordings enables higher diversity in annotated data, compared to annotating a single recording with the same total duration. Maximizing diversity is the pivotal sample election principle in [II, IV], and It has been shown effective according to the experimental results. Therefore, sound segments should be generated as an annotation unit, rather than using each recording as an annotation unit.

Considering the difficulty of annotation, the sound segments should not be too short. Otherwise, the annotation could be rather erroneous according to the listening test made in [18], where reports that manual annotations are clearly less accurate along with the decrease of the duration of sound segments, when the duration is below four seconds.

5.1.2 Preserving contextual information in training

The sound segments are processed independently in [II, IV], regardless of the temporal location in the original recordings. This is hardly optimal, since temporal information across sound segments can be potentially useful. This has been previously discussed in [40]. Firstly, background sounds of an annotated event might be helpful in learning the unique characteristics of an event out of the background. Secondly, contextual information can be used to model the dependencies in a sequence of sounds. As an example, key rattling and is often prior to a door closing sound. In order to preserve the contextual information of annotated segments, each recording is used as a training input, and the training losses are derived from only annotated seg-

ments within it.

5.1.3 Using weak labels

Traditionally, SED models are trained using strong labels, which are the target outputs of SED models: the onset, offset, and class of each individual sound event in each signal. However, strong labels are much more time-consuming to annotate, compared to annotating weak labels, which indicate only the presence of each target sound event in sound segments. According to the experimental results in recent studies on weakly supervised learning [58, 68], similar accuracy can be achieved using weak labels compared to using strong labels. Based on these results, weak labels are used in the proposed system.

5.2 Description of the active learning system for SED

Following the basic ideas described above, an active learning system is proposed in [V]. The overview of the active learning system is visualized in figure 5.1. The active learning system involves three components: generating sound segments from audio recordings, selecting sound segments for annotation, and weakly supervised learning with annotated sound segments within recordings.

5.2.1 Generating sound segments from audio recordings

The most straightforward approach to generating sound segments is to slice each audio recording into fixed-length segments. As a drawback, the boundary between two consecutive segments is quite often in the middle of a sound event, dividing a single event into two separate segments. This sometimes brings difficulties to annotators. Furthermore, a segment that contains a par-

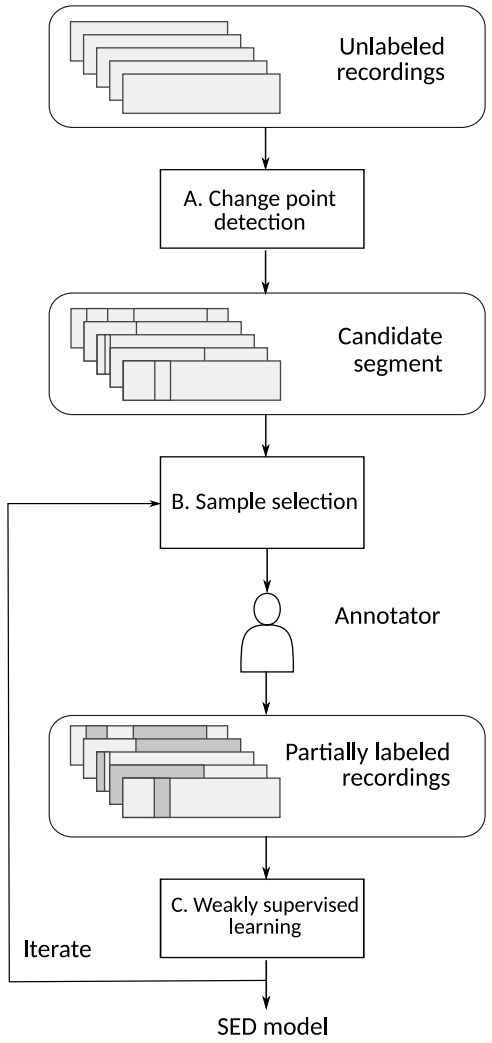

Figure 5.1 The overview of the active learning system proposed in [V]. ©2020 IEEE.

tial event is sometimes rather dissimilar to segments that contain full sound events of the same class. This results in errors when inferring the presence of sound events through similarity analysis. In order to avoid splitting events into separate segments, change point detection is used to generate segments.

Change point detection is a problem of finding abrupt changes in sequen-

tial data, typically caused by internal systematic change or external interference [2]. The purpose of using change point detection is to preserve full sound events between detected change points.

Previously, unsupervised audio segmentation methods were mostly based on MFCCs and Bayesian information criterion (BIC) [15, 59]. Change point detection can be considered as a problem of finding a time point where the signal prior to it has the largest dissimilarity compared to the signal subsequent to it. Recent study [105, 106] suggested that learned audio embeddings largely outperformed MFCCs for audio similarity analysis. Base on these results, the change point detection in [V] is based on embeddings extracted using a model learned from a large-scale sound event detection dataset, Audioset [33]. The embeddings are extracted per frame of mel band energies. The likelihood of change is estimated for each time point based on the cosine distance between the mean of past M frames and the mean of the future M frames. Peaks in the likelihood of change are detected as change points. The mel-spectrum, embeddings, and likelihood of change are visualized in Figure 5.2.

The generated sound segments are used as candidates for sample selection. Each selected segment is presented to an annotator for annotation. The sound event classes present in a segment x are considered as a set, denoted as L_x, which possibly contains zero, single, or multiple sound event classes.

5.2.2 Sample selection criterion for multi-label classification

The sample selection algorithm plays a pivotal role in the performance of an active learning system. In [II], clustering analysis has been used in sample selection, and evaluation shows that the proposed sample selection is effective with a limited labeling budget. K-medoids clustering is performed on an initially unlabeled dataset, and the medoids are selected for annotation. In [IV], after the medoids are annotated, the sample selection proceeds based on mismatch-first farthest-traversal as the second stage. Mismatch-first farthest-traversal aims at the points with wrong label predictions, meanwhile maxi-

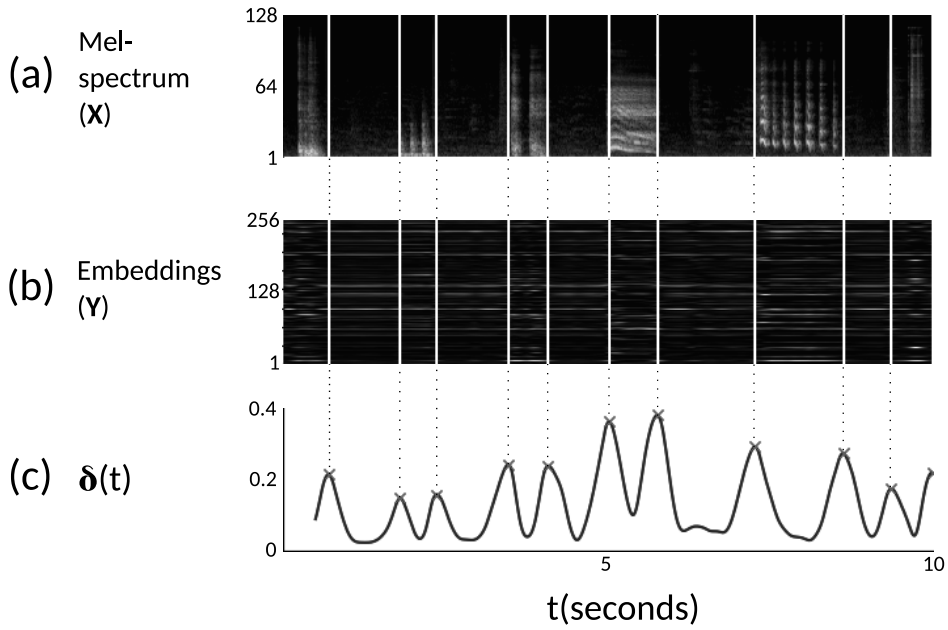

Figure 5.2 An illustration of the audio change point detecion method proposed in [V]. Panel (a) is the log-mel spectrogram of an example audio signal, with the detected change points marked by white vertical lines. Panel (b) visualizes the embeddings extracted using a pre-trained model. Panel (c) illustrates the estimated likelihood of change on each time step. The peaks in the likelihood sequence are detected as change points, which are marked with red crosses. ©2020 IEEE.

mizing the diversity of all selected samples. Evaluation in [IV] shows that mismatch-first farthest-traversal clearly outperforms reference sample selection methods on UrbanSound8K [85].

The K-medoids clustering in [II] and [IV] involves initialization of the K-medoids based on farthest-first traversal and optimization of the medoids based on partition around medoids (PAM) [53]. The time complexity of PAM is $O(k(n-k)^2)$ [53]; thus, the clustering analysis takes a considerable amount of time for a large scale dataset. Thus, the benefit of running PAM is estimated in a primary study. In one setup, PAM is performed to optimize the K-medoids. In another, the K medoids are directly produced using farthest-first traversal, without performing PAM. The K samples selected in each method are then used to train a classifier, respectively. According

to the result, the accuracy of the two classifiers has no significant difference. Therefore, the first stage of sample selection in publication [V] performs only the farthest-first traversal process, excluding the optimization process using PAM. As a result, the sample selection rule is equivalent in the first stage and in the second stage, since mismatch-first farthest-traversal is simplified to farthest-first traversal when predicted labels do not exist in the first stage. The overall sample selection method is then simply mismatch-first farthest-traversal.

As is introduced in Chapter IV, model-predicted labels and propagated labels are generated according to an existing model and nearest neighbor prediction, respectively. The prediction mismatch is used as the primary selection criterion. Publication [IV] deals with sound event classification where exactly one class is associated with each segment. In sound event detection, the number of present sound event classes in a segment can be possibly zero, one or multiple. Following this setup, the labels assigned to each audio segment is regarded as a set in [V]. The model-predicted labels are regarded as $\mathscr{A}_x$, and the propagated labels are regarded as a set $\mathscr{B}_x$. The prediction mismatch between the two sets is measured between two sets, using Jaccard Index as

$$J(x) = \begin{cases} \frac{|\mathscr{A}_x \cap \mathscr{B}_x|}{|\mathscr{A}_x \cup \mathscr{B}_x|} & \text{, if } \mathscr{A}_x \cup \mathscr{B}_x \neq \emptyset \\ 1 & \text{, if } \mathscr{A}_x \cup \mathscr{B}_x = \emptyset \end{cases}. \tag{5.1}$$

Among the sound segments that have the lowest Jaccard Index, farthest-traversal is performed in order to maximize the diversity of selected samples, as is introduced in chapter IV.

5.2.3 Weakly supervised learning

Neural networks are commonly used for SED. The outputs corresponding to each frame are commonly interpreted as probabilities of sound event activities. According to the setup in [V], the presence of sound events is annotated within selected sound segments. Previously, attention pooling [58]

and linear softmax [68] have been shown effective in learning SED models from weakly labeled audio. In the active learning setup, typically, a recording consists of multiple segments, and only a small number of segments are annotated under a small labeling budget assumption. A recording is referred to as a partially labeled recording if annotations exist but do not cover all the segments within it.

Previously, segments generated from a recording are processed independently in weakly supervised learning [50], regardless of the context of annotated segments. The contextual information may benefit the SED performance as is discussed in [40]; thus, publication [V] proposes to use partially labeled recordings as training inputs. Each recording is used as an input, and the training loss only depends on the frames corresponding to annotated segments. The signal not corresponding to an annotated segment may provide the following two types of information. Firstly, the background sounds of an annotated event might be helpful in learning the unique characteristics of an event out of the background. Secondly, contextual information can be used to model the dependencies in sound sequences. As an example, key rattling and latching sound is often prior to a door closing sound event, which is commonly used to describe the impact sound between a door and its frame. When only the door closing is defined as a target event, the key rattling and latching sound can help the door closing event be distinguished from other impact sounds in a home environment.

5.3 Evaluation

Publication [V] extends the active learning algorithm proposed in [IV] to sound event detection. Three novel components in the active learning system are evaluated: sound segments generated based on change point detection, the sample selection method based on mismatch-first farthest-traversal, and the weakly supervised learning method that uses full recording as training input, preserving the context for annotated segments. The evaluation of the components is made reversely, with respect to the order of processing in

the active learning system. Experiment A focuses on the weakly supervised learning; Experiment B focuses on the sample selection method; Experiment C focuses on the segmentation method based on change point detection.

5.3.1 Dataset and settings

Two SED datasets are used in the evaluation. The statistics of the two datasets are shown in Table 5.1. The first dataset is TUT Rare Sound Events 2017 [71], which is used in the challenge of Detection and Classification of Acoustic Scenes and Events (DCASE) 2017, as task 2. The dataset is used to evaluate active learning systems in scenarios where sound events are rare. The second dataset is TAU Spatial Sound Events 2019 - Ambisonic, which is used in the challenge of DCASE2019 [1], as task 3. The dataset is used to evaluate active learning systems in scenarios where sound events are dense.

TUT Rare Sound Events 2017 includes a training/testing split. The training data is regarded as an initially unlabeled dataset, and the annotated labels are generated to selected samples according to the ground truth. When the amount of annotated segments reaches a benchmarked labeling budget, a SED model is trained using annotated labels, and the model is tested using the testing split. The following proportions of the training data as labeling budgets are evaluated: 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 20%, 100%. Segment-based error rate (ER) [73] is used to evaluate the performance of a SED model. The segment length in the segment-based evaluation is one second, which is a common setup in sound event detection studies, such as DCASE 2017 task 3. Spatial Sound Events 2019 includes a four-fold cross-validation setup. In the experiments, the rotation of the training/testing split follows the cross-validation setup, and the average of the segment-based ER across the four folds is reported.

In order to evaluate each component in the proposed active learning system, three experiments have been made. Experiment A focuses on the idea of preserving context for annotated segments that are generated with the proposed change point detection algorithm, and the evaluated systems are sum-

marized in Table 5.1. Random sampling is used in experiment A. Experiment B focuses on the sample selection method, comparing random sampling, MFFT, and uncertainty sampling as is summarized in Table 5.2. The weakly supervised learning setup in experiment B follows system 1 in experiment A. The detailed setup of the experiments are available in the original paper in [V].

	System	Annotation unit	Label type	Training input
Experiment A1	1	segment	weak label	**recordings**
	2	segment	weak label	**segments**
Experiment A2	3	**segment**	strong label	recordings
	4	**recording**	strong label	recordings

Table 5.1 A summary of experiment A. Bold font is used to highlight the investigated aspect in each experiment.

	System	Sampling method
Experiment B	1	Random sampling
	5	Mismatch-first farthest-traversal (proposed)
	6	Uncertainty sampling

Table 5.2 A summary of experiment B on investigated sampling methods.

5.3.2 Experimental results

The experimental results from experiment A are shown in Figure 5.3. By comparing the results obtained with system 1 and system 2, we can see preserving the original recording as the context of each annotated segment clearly improves the performance, compared to training with only annotated segments. By comparing system 3 and system 4, clearly higher accuracy can be achieved with the same amount of annotations by annotating sound segments, compared to annotating recordings. In overall, the best performance

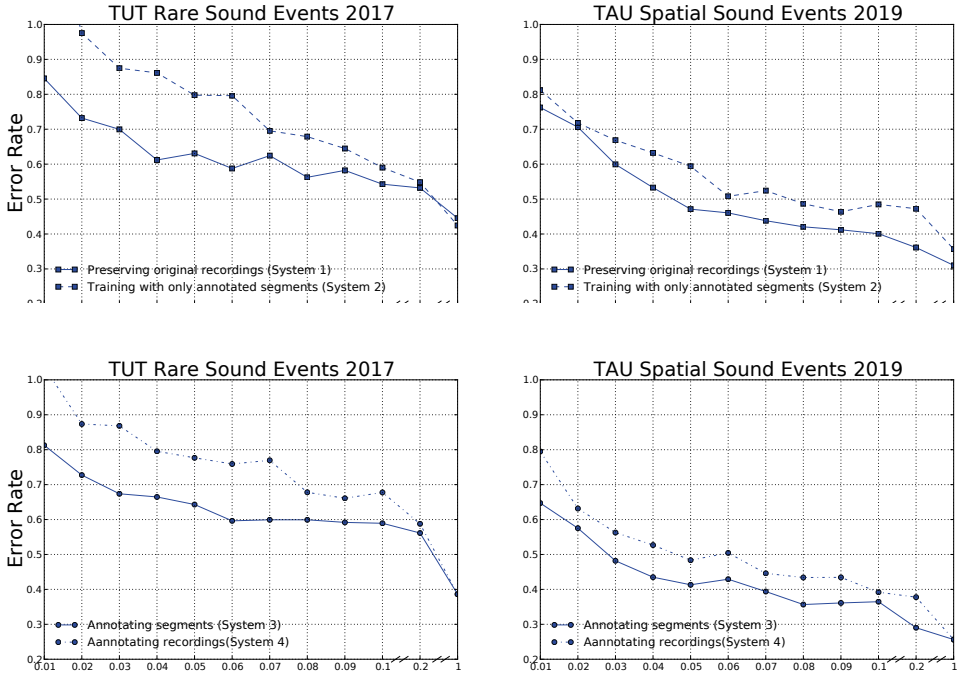

Figure 5.3 Error rate of learned models as the function of labeling budget for methods that use different training inputs and annotation units in experiment A. ©2020 IEEE.

is achieved by annotating sound segments and using original recordings as training inputs.

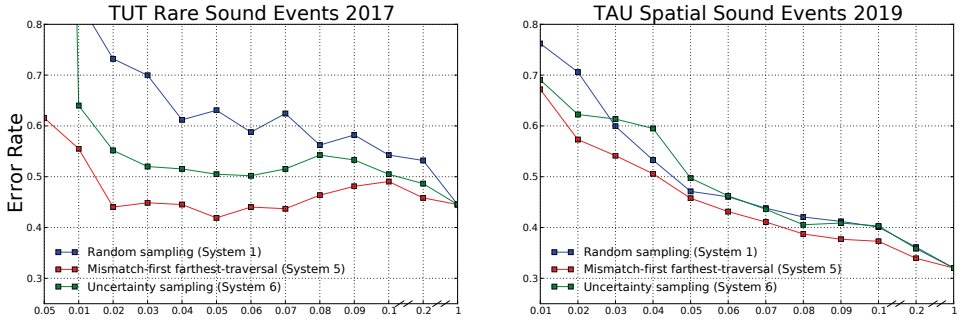

Figure 5.4 Error rate of learned models as the function of labeling budget for methods that use different sampling method in experiment B. ©2020 IEEE.

The experimental results from experiment B are shown in Figure 5.4. The

results show that the proposed sampling method clearly outperforms reference methods. In the experiments on the TUT Rare Sound dataset, the proposed method outperforms reference methods to a large extent. Remarkably, the proposed active learning method requires only 2% of the training data to be annotated to achieve similar performance, compared to annotating all the data. In the experiments on the TAU Spatial Sound dataset, the proposed method slightly outperforms the two reference methods.

6 DISCUSSION

This chapter first discusses the relevant research activities in the audio research community and how this thesis potentially benefits the community. Then, it discusses the situations where the proposed methods might have problems and how to modify the proposed methods to deal with different situations.

The field of SED has gained increasing interest in the audio research community. This can be seen from the number of participants in the DCASE challenge, which started in 2013. The numbers of both the participating teams and submitted systems have increased by every year. However, the number of productized SED applications is still very limited at present, compared to applications in the speech and music domain.

The cost of developing acoustic models is one of the obstacles to productizing some potential SED applications, since a SED application typically requires its own task-specific acoustic model to perform with reliable accuracies. General-purpose SED systems fail in many cases due to the following factors. Firstly, unlike phonemes or musical notes, sound event classes can hardly be universally defined. Different acoustic properties might be associated with the same class name in different tasks. For example, a "door closing" class literally includes sounds produced from oven doors, cupboard doors, sliding doors, electric doors, wooden doors, and many other types of doors. A specific SED application might assume only one type of door, excluding the others. Secondly, the frequency responses vastly vary among recording devices, especially outside the human voice frequency range. As a result, a general-purpose acoustic model may fail to produce reliable predictions without optimization for specific recording devices.

In order to reduce the cost of developing SED models, studies have been made on the direction of minimizing supervision effort since DCASE 2017 task 4 [71]. Various techniques have been introduced to SED, including weakly supervised learning, domain adaptation, and semi-supervised learning. These methods are used to utilize external data or unlabeled data to improve the performances of learned models. However, few studies have been made to deal with the selection of samples to be labeled. Since the selection method can largely affect the performance of learned models in many situations, a thesis that focuses on the selection problem is expected to add good value to the research community. Based on this expectation, the thesis is being conducted, and the contribution to the community is summarized as follows.

A sample selection strategy based on k-medoids clustering was proposed in [II]. It was then improved with mismatch-first farthest-traversal as the sample selection strategy after all the medoids being annotated. The methods were shown effective with respect to random sampling and uncertainty sampling as reference methods. When recordings are long, typically only a small number of representative segments in each recording are selected for annotation. Traditionally, the labeled parts in a recording are regarded as isolated segments. To utilize the contextual information, [V] proposed to keep the original recordings as training inputs, and training losses were derived from only labeled parts of the recordings. The experimental results showed that the proposed method clearly outperformed learning from annotated segments independently. Combining the sample selection strategy proposed in [II] and [IV], and the method of learning from partially labeled recordings in [V], the development cost of SED applications can be potentially reduced to a large extent. Furthermore, since the proposed active learning algorithms are generic, they can be applied to other types of data. For instance, the active learning method proposed in [IV] was modified and applied to a natural language processing problem in [8].

In addition to the contribution, some observations, possible modifications to address different situations are briefly disscussed. Ideally, a cluster-

ing method can directly group sounds according their ground truth classes. Thus, a straightforward idea would be to produce clusters with a similar number of sound event classes. This was investigated in [I]. As an observation, the clusters were rarely reliable to derive accurate propagated labels. Since sound event classes are usually defined by their semantic meaning, a sound class is typically associated with various subtypes of acoustic properties. Taking the "door closing" class discussed above as an example, wooden doors and electric doors may sound totally different, and they can hardly be grouped together. Instead, they may group with subtypes of other classes. As a result, the method proposed in [I] can hardly perform well, unless different classes can be clearly separated by their acoustic properties. In [II] and [IV], the number of clusters is defined as proportional to the total number of sound segments in the training dataset, much larger than the number of sound event classes. This allows subtypes of each class to have their own clusters; thus, clusters typically have high purities. As an observation, the propagated labels were typically more accurate than the model-predicted labels until a large proportion of the training dataset was labeled.

In ESC-10 and Urbansound8k datasets, the sound event classes are generally evenly distributed. With this prior knowledge, the clusters are sorted by cluster size, largest first. The purpose is to maximize the number of obtained propagated labels. With this cluster ranking method, instances of rare classes are unlikely to be selected in the early stages, since an instance of a rare class can hardly be the medoid of a large cluster. In cases that some sound event classes are rare, the active learning performance would be poor since samples of the rare classes are missing until a decent amount of data is labeled. When some of the sound event classes are rare such as in TUT Rare Sound Events 2017, the medoids ranking method should be replaced by farthest-first traversal, which is used in [V].

When the size of the training dataset is large, performing partition around medoids (PAM) algorithm to optimize clustering loss can take a large amount of time. A preliminary in [V] investigated using farthest-first traversal to approximate the medoids, without performing PAM to optimize the clustering

loss. The obtained active learning performance was close compared to the results using PAM. When the scale of the training dataset is too large to perform k-medoids clustering in practical time, using farthest-first traversal to approximate the medoids can be considered.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The development of acoustic models for sound event detection and classification requires labeled audio data. In most cases, audio data is easy to collect, but annotating the data is time-consuming. The research question to deal with these cases is how to evaluate the effectiveness of a machine learning method that requires only limited labels from abundant unlabeled data. Publication [II] proposes to use the classification accuracies of learned models as a function of labeling budget, the maximum number of labels that can be manually assigned. A method is considered more effective if it achieves higher classification accuracy with a given labeling budget or requires less labeling budget to achieve a target classification accuracy.

In most cases, the starting point of acoustic model development is a large amount of audio data, initially unlabeled, such as the harbor noise monitoring case study in [I]. The basic idea developed in [I] was to group similar sounds together using cluster analysis, and thereafter clustering results can be used to derive propagated labels from annotated samples.

One of the research questions raised from this idea is how to measure the similarities between sound pairs; thereafter, cluster analysis can be performed based on it. The propagated labels are reliable only when similar sounds belong to the same class, while dissimilar sounds belong to different classes. In preliminary studies, the audio similarity measurement methods were com-

pared using mean average precision [70] on ESC-50 [78] dataset. Three similarity measurement methods were investigated in this thesis. In [I], the similarity between two sounds was computed as the Euclidean distance between the means of corresponding MFCCs. The similarity measurement method was improved in [II, IV], where the dissimilarity between a sound pair was computed as the KL divergence between the statistical distributions of corresponding MFCCs. In [V], audio embeddings were extracted from a pre-trained convolutional neural network. The similarity between a sound pair was measured by the cosine similarity between their audio embeddings. This method achieved the highest AUC among the studied methods.

Given the proximity matrix of sound pairs, k-medoids clustering is a common choice for performing cluster analysis. The medoids, as representatives of each cluster, are natural choices for sample selection. In conclusion, we propose to use k-medoids clustering based on the similarity matrix generated using cosine similarities between the audio embeddings of corresponding sound pairs. Since optimizing the medoids using PAM algorithm is time-consuming, simply using farthest-first traversal to approximate the k-medoids can be considered when the scale of the dataset is very large.

Another research question is how to efficiently use the clustering results in terms of minimizing supervision effort. The first solution was proposed in [I], and improved solutions were proposed in [II, IV]. In [I], sounds were initially divided into ten clusters. In each cluster, a small number of sounds were randomly sampled and presented to an annotator. The annotator decided whether to collectively label a cluster or to further divide the cluster into ten smaller ones. Statistically, a decent number of samples are needed to test whether the purity of a cluster is high enough for collective labeling. The purity of a cluster is the fraction of instances from the most frequent class in a cluster. In [II], smaller clusters were produced using k-medoids clustering. Only the medoid of each cluster was selected for annotation, assuming the medoid to be the most frequent class in its cluster. The annotated label to a medoid was propagated to other cluster members. The idea was formalized as an active learning algorithm. In general, an active learning algorithm actively

requires labels for selected samples according to its selection criterion, which aims at picking the samples that are the most beneficial for model training. The proposed method was then called medoid-based active learning (MAL). MAL did not optimize the selection of samples after all the medoids were annotated. Mismatch-first farthest-traversal was proposed in [IV] as the sample selection method after annotating all the medoids. The experimental results showed that the active learning method proposed in [IV] required 50%-80% fewer labels to achieve the same accuracy with respect to the reference methods based on random sampling and uncertainty sampling. In conclusion, to minimize supervision effort based on clustering results, the most effective approach we have developed so far is medoid-based active learning followed by mismatch-first farthest-traversal.

Typically, only representative sounds in each recording were selected for annotation for the sake of saving labeling budget. In these cases, the research question was how to utilize the contextual information in the original recordings. In [V], each recording was used as a training input, and the loss was derived from only annotated segments within it. The evaluation results showed that preserving the context information for annotated segments clearly outperformed using each annotated segment independently as training input. In addition, the sample selection method, mismatch-first farthest-traversal, proposed in [IV] was extended to multi-label classification in [V]. In a dataset where sound events were rare, the overall proposed method required annotating only 2% of the training data to achieve the same accuracy, with respect to annotating all the training data.

As another direction of utilizing clustering results, cluster analysis was investigated to group recordings with similar recording conditions. Feature normalization according to cluster statistics was used to bridge the distribution shift when recordings were captured in different conditions. The performance clearly outperformed feature normalization based on dataset-wise statistics and recording-wise statistics.

7.2 Future work

The sound representation is vital to different aspects of the proposed active learning algorithms, including unsupervised segmentation, clustering and supervised detection and classification. The embeddings extracted based on a pre-trained model using Audioset in [V] outperformed approaches based on MFCCs in [II, IV]. In the future, the development of self learning or pre-training techniques potentially results in a big improvement in active learning performances. A recent study [93] showed that a clear improvement in performance was achieved in the experiments of [II, IV] by replacing MFCCs with embeddings extracted using a general-purpose pretrained audio neural network [57].

The scale of unlabeled datasets can be very large in many cases. It is typically time-consuming to perform k-medoids clustering on large-scale sound datasets, since partition around medoid (PAM) algorithm for k-medoids clustering has a high computation complexity, $O(n^2k^2)$. In the preliminary study in [V], when the medoids are initialized using farthest-first traversal, the active learning performance was similar, whether to run PAM to optimize the clustering loss or not. Future studies can be made to investigate if the same observation can be found on other datasets. In addition, The runtime of k-medoids clustering can be largely reduced using FastPAM [87] or BanditPAM [97], slightly comprising the clustering loss compared to PAM. It might be worthy of investigating alternative k-medoids clustering algorithm for active learning.

In [V], propagated labels were used only for sample selection but not for model training. This left a space to combine the active learning method with the semi-supervised learning methods such as mean-teacher [96]. In the future, the optimal combination of active learning and semi-supervised learning could further optimize the accuracy of learned models when the labeling budget is limited.

REFERENCES

[1] S. Adavanne, A. Politis, J. Nikunen and T. Virtanen. Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks. *Journal of Selected Topics Signal Processing* 13.1 (2019), 34–48. DOI: 10.1109/JSTSP.2018.2885636.

[2] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51.2 (2017), 339–367. DOI: 10.1007/s10115-016-0987-z.

[3] D. Angluin. Queries and Concept Learning. *Machine Learning* 2 (1987), 319–342. DOI: 10.1007/BF00116828.

[4] Y. Aytar, C. Vondrick and A. Torralba. SoundNet: Learning Sound Representations from Unlabeled Video. *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 2016, 892–900.

[5] J. Barker, E. Vincent, N. Ma, H. Christensen and P. D. Green. The PASCAL CHiME speech separation and recognition challenge. *Computer Speech and Language* 27.3 (2013), 621–633. DOI: 10.1016/j.csl.2012.10.004.

[6] *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 2012.

[7] M. Becker, J. Lippel, A. Stuhlsatz and T. Zielke. Robust dimensionality reduction for data visualization with deep neural networks. *Graphical Models* 108 (2020), 101060. DOI: 10.1016/j.gmod.2020.101060.

[8] R. Bhambhoria, L. Feng, D. Sepehr, J. Chen, C. Cowling, S. A. Koçak and E. Dolatabadi. A Smart System to Generate and Validate Question Answer Pairs for COVID-19 Literature. *Proceedings of the First Workshop on Scholarly Document Processing*. 2020, 20–30. DOI: 10.18653/v1/2020.sdp-1.4.

[9] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam and J. Bello. MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research. *15th International Society for Music Information Retrieval Conference*. 2014.

[10] Z. Bitvai and T. Cohn. Non-Linear Text Regression with a Deep Convolutional Neural Network. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*. 2015, 180–185. DOI: 10.3115/v1/p15-2030.

[11] E. Cakir, T. Heittola, H. Huttunen and T. Virtanen. Polyphonic sound event detection using multi label deep neural networks. *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*. 2015, 1–7. DOI: 10.1109/IJCNN.2015.7280624.

[12] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen and T. Virtanen. Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection. *IEEE ACM Transaction on Audio, Speech, and Language Processing* 25.6 (2017), 1291–1303. DOI: 10.1109/TASLP.2017.2690575.

[13] S. Chang, N. M. Nasrabadi and L. Carin. Infrared-image classification using support vector machines. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002,*

May 13-17 2002, Orlando, Florida, USA. IEEE, 2002, 4168. DOI: 10.1109/ICASSP.2002.5745606.

[14] C. C. Chatterjee, M. Mulimani and S. G. Koolagudi. Polyphonic Sound Event Detection Using Transposed Convolutional Recurrent Neural Network. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*. 2020, 661–665. DOI: 10.1109/ICASSP40776.2020.9054628.

[15] S.-S. Cheng, H.-M. Wang and H.-C. Fu. BIC-based audio segmentation by divide-and-conquer. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*. 2008, 4841–4844. DOI: 10.1109/ICASSP.2008.4518741.

[16] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of Acoustic Society of America* 111 (2002), 1917–1930.

[17] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 2014, 1724–1734. DOI: 10.3115/v1/d14-1179.

[18] S. Chu, S. S. Narayanan and C.-C. J. Kuo. Environmental Sound Recognition With Time-Frequency Audio Features. *IEEE Transaction on Audio, Speech and Language Processing* 17.6 (2009), 1142–1158. DOI: 10.1109/TASL.2009.2017438.

[19] I. B. Ciocoiu. Hybrid Feedforward Neural Networks for Solving Classification Problems. *Neural Processing Letters* 16.1 (2002), 81–91. DOI: 10.1023/A:1019755726221.

[20] D. A. Cohn, L. E. Atlas and R. E. Ladner. Improving Generalization with Active Learning. *Machine Learning* 15.2 (1994), 201–221. DOI: 10.1007/BF00993277.

[21] J. Cramer, H.-H. Wu, J. Salamon and P. J. Bello. Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings. *International Conference on Acoustics Speech and Signal Processing* (2019).

[22] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), 357–366. DOI: 10.1109/TASSP.1980.1163420.

[23] A. P. Dempster, N. M. Laird and D. B. Rubin. *Journal of the Royal Statistical Society: Series B* 39 (1977), 1–38.

[24] K. Drossos, P. Magron and T. Virtanen. Unsupervised Adversarial Domain Adaptation Based on The Wasserstein Distance For Acoustic Scene Classification. *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2019, New Paltz, NY, USA, October 20-23, 2019*. 2019, 259–263. DOI: 10.1109/WASPAA.2019.8937231.

[25] B. S. Everitt, S. Landau and M. Leese. *Cluster Analysis*. 4th. Wiley Publishing, 2009. ISBN: 0340761199.

[26] M. Faundez-Abans, M. I. Ormeno and M. de Oliveira-Abans. Classification of planetary nebulae by cluster analysis and artificial neural networks. *Astronomy and Astrophysics Supplement Series* 116 (Apr. 1996), 395–402.

[27] T. Fawcett. An introduction to ROC analysis. *Pattern Recognit. Lett.* 27.8 (2006), 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[28] L. N. Ferreira, A. R. Pinto and L. Zhao. QK-Means: A clustering technique based on community detection and K-Means for deployment of cluster head nodes. *The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012*. 2012, 1–7. DOI: 10.1109/IJCNN.2012.6252477.

[29] R. Fettiplace. Hair Cell Transduction, Tuning, and Synaptic Transmission in the Mammalian Cochlea. 7(4) (2017), 1197–1227.

[30] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio and M. Vento. Audio Surveillance of Roads: A System for Detecting Anomalous Sounds. *IEEE Transaction on Intelligent Transportation Systems* 17.1 (2016), 279–288. DOI: 10.1109/TITS.2015.2470216.

[31] E. Fonseca, M. Plakal, F. Font, D. Ellis and X. Serra. Audio Tagging with Noisy Labels and Minimal Supervision. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. 2019.

[32] Y. Ganin and V. S. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. 2015, 1180–1189.

[33] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal and M. Ritter. Audio Set: An ontology and human-labeled dataset for audio events. *Proc. IEEE ICASSP 2017*. New Orleans, LA, 2017.

[34] S. Goetze, J. Schröder, S. Gerlach, D. Hollosi, J.-E. Appell and F. Wallhoff. Acoustic Monitoring and Localization for Social Care. *JCSE* 6.1 (2012), 40–50. DOI: 10.5626/JCSE.2012.6.1.40.

[35] K. C. Gowda and T. V. Ravi. Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity. *Pattern Recognition* 28.8 (1995), 1277–1282. DOI: 10.1016/0031-3203(95)00003-I.

[36] M. Grimaldi and F. Cummins. Speaker Identification Using Instantaneous Frequencies. *IEEE Transactions on Audio, Speech, and Language Processing* 16.6 (2008), 1097–1111. DOI: 10.1109/TASL.2008.2001109.

[37] D. Hakkani-Tür, G. Riccardi and A. L. Gorin. Active learning for automatic speech recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, May 13-17 2002, Orlando, Florida, USA*. 2002, 3904–3907. DOI: 10.1109/ICASSP.2002.5745510.

[38] S. Haltsonen. Improved dynamic time warping methods for discrete utterance recognition. *IEEE Transaction on Acoustic Speech Signal Processing* 33.2 (1985), 449–450. DOI: 10.1109/TASSP.1985.1164559.

[39] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu and X. Zhu. Semi-Supervised Active Learning for Sound Classification in Hybrid Learning Environments. *PLOS ONE* 11.9 (Sept. 2016), 1–23. DOI: 10.1371/journal.pone.0162075.

[40] T. Heittola, A. Mesaros, A. Eronen and T. Virtanen. Context-Dependent Sound Event Detection. *EURASIP Journal on Audio, Speech and Music Processing* (2013).

[41] T. Heittola, A. Mesaros, D. Korpi, A. J. Eronen and T. Virtanen. Method for creating location-specific audio textures. *EURASIP Journal on Audio Speech and Music Processing* 2014 (2014), 9. DOI: 10.1186/1687-4722-2014-9.

[42] T. Heittola, A. Mesaros and T. Virtanen. Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*. 2020.

[43] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss and K. Wilson. CNN Architectures for Large-Scale Audio Classification. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[44] F. Hilger, S. Molau and H. Ney. Quantile based histogram equalization for online applications. *7th International Conference on Spoken Language Processing, ICSLP2002*. 2002.

[45] G. R. Hjaltason and H. Samet. Properties of Embedding Methods for Similarity Searching in Metric Spaces. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 25.5 (2003), 530–549. DOI: 10.1109/TPAMI.2003.1195989.

[46] D. S. Hochbaum and D. B. Shmoys. A Best Possible Heuristic for the *k*-Center Problem. *Mathematics of Operations Research* 10.2 (1985), 180–184. DOI: 10.1287/moor.10.2.180.

[47] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computing* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[48] S. C. H. Hoi, R. Jin and M. R. Lyu. Large-scale text categorization by batch mode active learning. *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*. ACM, 2006, 633–642. DOI: 10.1145/1135777.1135870.

[49] T. Ishibashi, Y. Nakao and Y. Sugano. Investigating audio data visualization for interactive sound recognition. *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17-20, 2020*. ACM, 2020, 67–77. DOI: 10.1145/3377325.3377483.

[50] A. Jansen, J. F. Gemmeke, D. P. W. Ellis, X. Liu, W. Lawrence and D. Freedman. Large-scale audio event discovery in one million YouTube videos. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. 2017, 786–790. DOI: 10.1109/ICASSP.2017.7952263.

[51] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transaction on Pattern Analysis and Machchine Intelligence* 24.7 (2002), 881–892. DOI: 10.1109/TPAMI.2002.1017616.

[52] *Karaoke Version*. www.karaoke-version.com. Accessed: 11.03.2016.

[53] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990. ISBN: 978-0-47187876-6. DOI: 10.1002/9780470316801.

[54] A. Klapuri, A. J. Eronen and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transaction on Speech Language and Audio Processing* 14.1 (2006), 342–355. DOI: 10.1109/TSA.2005.854090.

[55] Y. Koizumi, S. Saito, H. Uematsu, N. Harada and K. Imoto. ToyADMOS: A Dataset of Miniature-machine Operating Sounds for Anomalous Sound Detection. *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. Nov. 2019, 308–312.

[56] J. Kominek, A. W. Black and V. Ver. *CMU Arctic Databases for Speech Synthesis*. Tech. rep. Carnegie Melon University, 2003.

[57] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang and M. D. Plumbley. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE ACM Transaction on Audio Speech and Language Procesing* 28 (2020), 2880–2894. DOI: 10.1109/TASLP.2020.3030497.

[58] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang and M. D. Plumbley. Weakly Labelled AudioSet Tagging With Attention Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.11 (2019), 1791–1802. DOI: 10.1109/TASLP.2019.2930913.

[59] M. Kotti, E. Benetos and C. Kotropoulos. Computationally Efficient and Robust BIC-Based Speaker Segmentation. *IEEE Transaction on Speech Audio and Language Processing* 16.5 (2008), 920–933. DOI: 10.1109/TASL.2008.925152.

[60] A. Kumar, M. Khadkevich and C. Fügen. Knowledge Transfer from Weakly Labeled Audio Using Convolutional Neural Network for Sound Events and Scenes. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. 2018, 326–330. DOI: 10.1109/ICASSP.2018.8462200.

[61] O. Levy and Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization. *Annual Conference on Neural Information Processing Systems 2014*. 2014, 2177–2185.

[62] D. D. Lewis. A Sequential Algorithm for Training Text Classifiers: Corrigendum and Additional Data. *SIGIR Forum* 29.2 (1995), 13–19. DOI: 10.1145/219587.219592.

[63] C.-J. Lin. Errata to "A comparison of methods for multiclass support vector machines". *IEEE Transaction on Neural Networks* 13.4 (2002), 1026–1027. DOI: 10.1109/TNN.2002.1021904.

[64] R. Lippmann. Book Review: Neural Networks, a Comprehensive Foundation, by Simon Haykin. *International Journal of Neural Systems* 5.4 (1994), 363–364. DOI: 10.1142/S0129065794000372.

[65] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transaction on Information Theory* 28.2 (1982), 129–136. DOI: 10.1109/TIT.1982.1056489.

[66] M. I. Mandel and D. Ellis. Song-Level Features and Support Vector Machines for Music Classification. *ISMIR 2005, 6th International Conference on Music Information Retrieval, London, UK, 11-15 September 2005, Proceedings*. 2005, 594–599.

[67] K. Marciniuk, M. Szczodrak and A. Czyzewski. An application of acoustic sensors for the monitoring of road traffic. *Signal Processing: Algorithms, Architectures, Arrangements, and Applications, SPA 2018, Poznan, Poland, September 19-21, 2018*. IEEE, 2018, 208–212. DOI: 10.23919/SPA.2018.8563406.

[68] B. McFee, J. Salamon and J. P. Bello. Adaptive Pooling Operators for Weakly Labeled Sound Event Detection. *IEEE ACM Transaction on Audio Speech and Language Processing* 26.11 (2018), 2180–2193. DOI: 10.1109/TASLP.2018.2858559.

[69] I. V. McLoughlin, H. Zhang, Z. Xie, Y. Song and W. Xiao. Robust Sound Event Classification Using Deep Neural Networks. *IEEE ACM Transaction on Audio, Speech, and Language Processing* 23.3 (2015), 540–552. DOI: 10.1109/TASLP.2015.2389618.

[70] Mean Average Precision. *Encyclopedia of Database Systems*. 2009, 1703. DOI: 10.1007/978-0-387-39940-9_3032.

[71] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj and T. Virtanen. DCASE 2017 Challenge Setup: Tasks, Datasets and Baseline System. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Nov. 2017, 85–92.

[72] A. Mesaros, T. Heittola, A. J. Eronen and T. Virtanen. Acoustic event detection in real life recordings. *18th European Signal Processing Conference, EUSIPCO 2010, Aalborg, Denmark, August 23-27, 2010*. IEEE, 2010, 1267–1271.

[73] A. Mesaros, T. Heittola and T. Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences* 6.6 (2016), 162.

[74] S. Molau, F. Hilger and H. Ney. Feature space normalization in adverse acoustic conditions. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, Hong Kong, April 6-10, 2003*. 2003, 656–659. DOI: 10.1109/ICASSP.2003.1198866.

[75] H. T. Nguyen and A. W. M. Smeulders. Active learning using pre-clustering. *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*. 2004. DOI: 10.1145/1015330.1015349.

[76] H.-S. Park and C.-H. Jun. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications* 36.2 (2009), 3336–3341. DOI: 10.1016/j.eswa.2008.01.039.

[77] F. Petitjean and J. Weber. Efficient Satellite Image Time Series Analysis Under Time Warping. *IEEE Geoscience Remote Sensing Letters* 11.6 (2014), 1143–1147. DOI: 10.1109/LGRS.2013.2288358.

[78] K. J. Piczak. ESC: Dataset for Environmental Sound Classification. *23rd Annual ACM Conference on Multimedia Conference*. 2015, 1015–1018. DOI: 10.1145/2733373.2806390.

[79] A. Pikrakis, S. Theodoridis and D. Kamarotos. Recognition of isolated musical patterns using Context Dependent Dynamic Time Warping. *11th European Signal Processing Conference, EUSIPCO 2002, Toulouse, France, September 3-6, 2002*. 2002, 1–4.

[80] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa and Y. Kawaguchi. MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. Nov. 2019, 209–213.

[81] A. P. Reynolds, G. Richards and V. J. Rayward-Smith. The Application of K-Medoids and PAM to the Clustering of Rules. *Intelligent Data Engineering and Automated Learning - IDEAL 2004, 5th International Conference, Exeter, UK, August 25-27, 2004, Proceedings*. 2004, 173–178. DOI: 10.1007/978-3-540-28651-6_25.

[82] G. Riccardi and D. Hakkani-Tür. Active learning: theory and applications to automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 13.4 (2005), 504–511. DOI: 10.1109/TSA.2005.848882.

[83] D. J. Rosenkrantz, R. E. Stearns and P. M. L. II. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing* 6.3 (1977), 563–581. DOI: 10.1137/0206041.

[84] M. Sahidullah and G. Saha. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication* 54.4 (2012), 543–565. DOI: 10.1016/j.specom.2011.11.004.

[85] J. Salamon, C. Jacoby and J. P. Bello. A Dataset and Taxonomy for Urban Sound Research. *22nd ACM International Conference on Multimedia (ACM-MM'14)*. Orlando, FL, USA, Nov. 2014, 1041–1044.

[86] M. Sassano. An Empirical Study of Active Learning with Support Vector Machines for Japanese Word Segmentation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 2002, 505–512. DOI: 10.3115/1073083.1073168.

[87] E. Schubert and P. J. Rousseeuw. Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms. *Similarity Search and Applications - 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2-4, 2019, Proceedings*. 2019, 171–187. DOI: 10.1007/978-3-030-32047-8_16.

[88] E. Sejdic, I. Djurovic and J. Jiang. Time-frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing* 19.1 (2009), 153–183. DOI: 10.1016/j.dsp.2007.12.004.

[89] J. Seo and B. Shneiderman. Interactively Exploring Hierarchical Clustering Results. *Computer* 35.7 (2002), 80–86. DOI: 10.1109/MC.2002.1016905.

[90] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. DOI: 10.2200/S00429ED1V01Y201207AIM018.

[91] H. S. Seung, M. Opper and H. Sompolinsky. Query by Committee. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh, Pennsylvania, USA, 1992, 287–294. ISBN: 0-89791-497-X. DOI: 10.1145/130385.130417.

[92] J. Shao, Q. Wang and F. Liu. Learning to Sample: An Active Learning Framework. *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*. 2019, 538–547. DOI: 10.1109/ICDM.2019.00064.

[93] S. Shishkin, D. Hollosi, S. Doclo and S. Goetze. Active learning for sound event classification using Monte-Carlo dropout and PANN embeddings. *6th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE*. 2021, 368–374.

[94] R. Sibson. SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method. *The Computer Journal* 16.1 (1973), 30–34. DOI: 10.1093/comjnl/16.1.30.

[95] S. Stevens, J. Volkmann and E. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America* 8 (1937), 185–190.

[96] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. 2017.

[97] M. Tiwari, M. J. Zhang, J. Mayclin, S. Thrun, C. Piech and I. Shomorony. BanditPAM: Almost Linear Time k-medoids Clustering via Multi-Armed Bandits. *Advances in Neural Information Processing Systems*. 2020, 368–374.

[98] S. Tong and D. Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Jouranal of Machine Learning Research* 2 (2001), 45–66.

[99] V. C. Tran, T. T. Nguyen, D. T. Hoang, D. Hwang and N. T. Nguyen. Active Learning-Based Approach for Named Entity Recognition on Short Text Streams. *Multimedia and Network Information Systems - Proceedings of the 10th International Conference MISSI 2016, Wrocław, Poland, 14-16 September 2016*. 2016, 321–330. DOI: 10.1007/978-3-319-43982-2_28.

[100] O. Viikki and K. Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication* 25.1-3 (1998), 133–147. DOI: 10.1016/S0167-6393(98)00033-8.

[101] R. Wang, G. Liu, C. Wang, L. Su and L. Sun. Predicting overlapping protein complexes based on core-attachment and a local modularity structure. *BMC Bioinformics* 19.1 (2018), 305:1–305:15. DOI: 10.1186/s12859-018-2309-9.

[102] Y. Wang. Polyphonic sound event detection with weak labeling. PhD thesis. Carnegie Mellon University, 2018.

[103] J. Yan, Y. Song, L.-R. Dai and I. V. McLoughlin. Task-Aware Mean Teacher Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. 2020, 326–330. DOI: 10.1109/ICASSP40776.2020.9053073.

[104] L. Yang, J. Hao, Z. Hou and W. Peng. Two-Stage Domain Adaptation for Sound Event Detection. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*. Tokyo, Japan, Nov. 2020, 230–234.

[105] Y. Zhang and Z. Duan. IMISOUND: An unsupervised system for sound query by vocal imitation. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. 2016, 2269–2273. DOI: 10.1109/ICASSP.2016.7472081.

[106] Y. Zhang and Z. Duan. IMINET: Convolutional semi-siamese networks for sound search by vocal imitation. *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2017, New Paltz, NY, USA, October 15-18, 2017*. 2017, 304–308. DOI: 10.1109/WASPAA.2017.8170044.

[107] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen and L. Carin. Adversarial Feature Matching for Text Generation. *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, 4006–4015.

[108] Z. Zhang and B. W. Schuller. Semi-supervised learning helps in sound event classification. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*. 2012, 333–336. DOI: 10.1109/ICASSP.2012.6287884.

[109] E. Zwicker. Subdivision of the audible frequency range into critical bands. *The Journal of the Acoustical Society of America* 33 (1961).

PUBLICATIONS

PUBLICATION I

Environmental noise monitoring using source classification in sensors

P. Maijala, Zhao S.Y., T. Heittola and T. Virtanen

Applied Acoustics 129.6 (2018), 258–267

Applied Acoustics 129 (2018) 258–267

Contents lists available at ScienceDirect

Applied Acoustics

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

Environmental noise monitoring using source classification in sensors

Panu Maijala [a,*], Zhao Shuyang [b], Toni Heittola [b], Tuomas Virtanen [b]

[a] VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland
[b] Tampere University of Technology, PO Box 527, FI-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 8 May 2016
Received in revised form 8 July 2017
Accepted 3 August 2017
Available online 12 August 2017

Keywords:
Environmental noise monitoring
Acoustic pattern classification
Wireless sensor network
Cloud service

ABSTRACT

Environmental noise monitoring systems continuously measure sound levels without assigning these measurements to different noise sources in the acoustic scenes, therefore incapable of identifying the main noise source. In this paper a feasibility study is presented on a new monitoring concept in which an acoustic pattern classification algorithm running in a wireless sensor is used to automatically assign the measured sound level to different noise sources. A supervised noise source classifier is learned from a small amount of manually annotated recordings and the learned classifier is used to automatically detect the activity of target noise source in the presence of interfering noise sources. The sensor is based on an inexpensive credit-card-sized single-board computer with a microphone and associated electronics and wireless connectivity. The measurement results and the noise source information are transferred from the sensors scattered around the measurement site to a cloud service and a noise portal is used to visualise the measurements to users. The proposed noise monitoring concept was piloted on a rock crushing site. The system ran reliably over 50 days on site, during which it was able to recognise more than 90% of the noise sources correctly. The pilot study shows that the proposed noise monitoring system can reduce the amount of required human validation of the sound level measurements when the target noise source is clearly defined.

1. Introduction

Environmental noise, defined as unwanted or harmful outdoor sound created by human activities [1, Art. 3], can be generated by traffic, industry, construction, and recreation activities [2, p. 12]. Airports, (wind) power plants, rock-crushing, shooting ranges, and motorsport tracks are examples of noise sources for which sound propagation over several kilometers is relevant.

One challenge in environmental noise monitoring is how to make sufficiently comprehensive measurements both in time domain and spatially. The changes in weather conditions have a significant effect on monitored noise levels [3] and in order to obtain most of the variations the noise has to be monitored for extended periods of time [4–6]. Also, a single point noise measurement is rarely representative for a whole neighbourhood and several sensor locations are needed. Because of high costs of the equipment and the amount of human resources needed, the reliability, validity, and representativeness of environmental data is usually unsatisfactory. Only a few reported scientific experiments with uninterrupted noise data captured from each relevant location over long periods of time exist [7–10].

The typical need for measurements is to monitor the noise caused by a noise source (e.g. an airport or an industrial plant) in a residential area. However, also other noise sources exist and the captured noise level is usually a result of a combination of the target and interfering sound sources: wind-generated, cars, and birds being examples. Sound level meters used for noise monitoring either capture sound levels or time domain noise data and store the data locally – or nowadays more often – on a remote server [11]. The most common method to ensure the noise was caused by the original source is listening through all the samples afterwards. This requires a huge amount of resources because of a large amount of data due to often necessary long-term measurements. Also, if only noise levels are recorded, validation by listening is not possible.

A considerable amount of manual work can be saved by automatically validating sound sources. Furthermore, privacy issues can be avoided and required network load can be largely reduced, if the automatic validation algorithm is performed on the sensor and only the measurement result is transferred. Previous validation algorithms on sensors have been limited to hand-crafted rule-based systems [12]. However, a simple hand-crafted classifi-

* Corresponding author.
E-mail addresses: Panu.Maijala@vtt.fi (P. Maijala), Shuyang.Zhao@tut.fi (Z. Shuyang), Toni.Heittola@tut.fi (T. Heittola), Tuomas.Virtanen@tut.fi (T. Virtanen).

http://dx.doi.org/10.1016/j.apacoust.2017.08.006
0003-682X/

cation rule can hardly provide good accuracy in a complex environment, e.g. monitored target producing several types of sounds. As another drawback, the design of a hand-crafted classifier requires an expert for every noise monitoring scenario. The increased computational capacity has made a sensor possible to classify noise sources using a pattern classification algorithm, which is capable of learning a sophisticated noise source classifier for an arbitrary scenario, simply using relevant annotated recordings as training material.

An pattern classification algorithm typically consists of a feature extractor and a classifier. Mel-frequency cepstral coefficients (MFCCs) [13] are used as common features for a wide range of acoustic pattern classification such as speech recognition [14] and music information retrieval [15]. Gaussian mixture model (GMM) [16] has been traditionally cooperated with MFCCs to model different types of sounds. Specifically, the combination of MFCCs and GMM has been used for various noise monitoring scenarios [17,18]. The use of artificial neural network (ANN) for acoustic pattern classification has been increasing with the development of computing power and new training algorithms that allow utilising large amounts of training data. Some recent studies have shown that ANN outperforms traditional GMM in sound event detection [19–21].

Together with the smaller and cheaper computing capacity, the breakthrough of wireless technology in the very beginning of 2000s have made possible to translate the physical world into information [22] and given reason to define concepts like Internet of Things and ubiquitous sensing [23]. The word "smart" was first used as an attribute to a sensor with an Internet access. Today, it is more closely related to a sensor with own intelligence, some computational capacity for data analysis and decision making [24].

The main objective of this study was to show if it would be possible to automatically capture only the noise from the original source, by adding intelligence and human hearing-like decision algorithms to the sensor. This would free the huge amount of human resources needed to validate the noise data and improve and representativeness of the results in environmental noise measurements. An implementation of a noise classification algorithms in a sensor will be introduced. The general concept of the noise monitoring system is explained in Section 2 and the pattern classification algorithms are given in Section 3. Additionally, an evaluation of the performance of the algorithms in a case study is shown (Section 4) and some discussion the requirements and the future work in Section 5.

2. Noise monitoring

The proposed noise monitoring system comprises of *smart sensors* which are connected through wireless uplink to the *cloud service*. The overview of the system is illustrated in Fig. 1. The smart sensor consist of a measurement microphone and a single-board computer with a wireless transmission unit. To alleviate the privacy issues concerning the continuous audio capturing and storage, the most of the analysis and processing is done already in the sensor and only analysed data is transferred and stored in the default setting. This approach will also lower the amount of transferred data from a sensor to the cloud service, and enables placing sensors to areas with lower quality wireless uplinks. In the sensor, A-weighted 10-min equivalent sound pressure level ($L_{p,\mathrm{A},600\mathrm{s}}$) values are calculated continuously, and predominant noise sources are detected within the measurement time segment. This information is used to decide whether the actual acoustic signal is needed for further inspection in the cloud service. For example, segments exceeding the legal maximum allowed sound level can be saved for manual inspection. All the extracted measurements are transmitted from the smart sensor to the cloud service for further analysis. The cloud service stores the data in the measurement database, and audio segments marked for later inspection are stored in the disk server. End-users access the measurement data and analysis of the measurements through a web-based portal.

2.1. Smart sensor

For the prototype, the credit-card-sized RPi (Raspberry Pi) developed by the Raspberry Pi Foundation was selected mainly due to its excellent support network and general usability. RPi1, the first generation model was used in the prototype because it was the only available model in 2012 when the implementation was made. Additional functionality was added by an audio codec (a 24-bit multi-bit sigma delta AD converter), a smart power

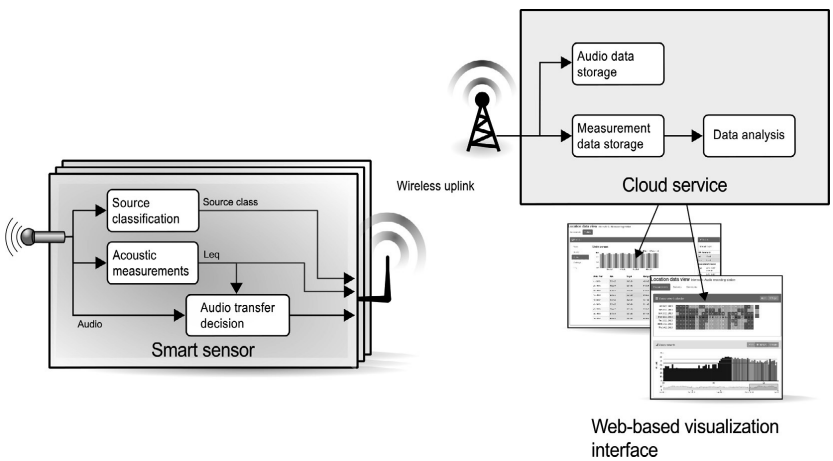

Fig. 1. Block diagram of the noise monitoring system.

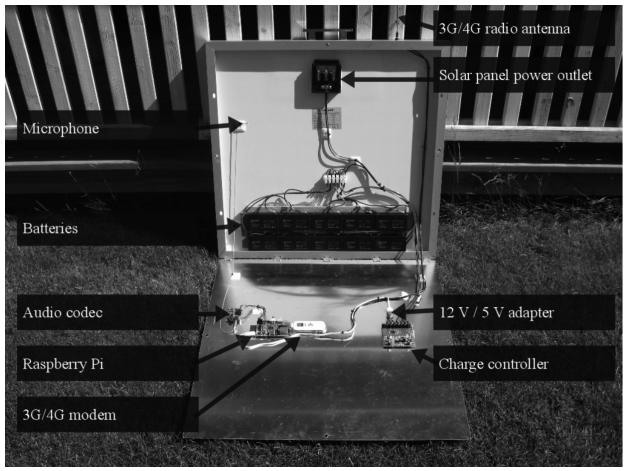

Fig. 2. A prototype version back cover opened.

management board with an uninterruptible power supply feature, and mobile connectivity. The selection of the microphones ended up with two models: one covering the audible range dynamics from 14 dB to 119 dB, and another from 20 dB to 140 dB (A-weighted).

Based on preliminary tests, solar power was selected to allow totally wireless sensors. The electronics and batteries were built inside a solar panel frame (see Fig. 2). Whenever the 60 W panel gets solar energy, the batteries start charging, system is powered up, a secure cloud service connection is established, and pre-processed real-time noise data flow to the online service is initiated. It is also possible to access the sensor unit remotely through the online service. The batteries, when fully charged, will keep the system running during the dark hours. The total cost of the components is about 150 €, the solar panel being the most expensive component, but the price could be reduced in mass production, or using an external power source.

The sensor continuously monitors the noise by capturing 10-min long non-overlapping analysis segments, and the equivalent sound pressure level $L_{p,A,600s}$ values are calculated for each segment. The sound source classification is used to find the noise source likelihoods within the analysis segment. The acoustic measurement values, noise source likelihoods and time-stamps are transmitted to the cloud service. Analysis segments having $L_{p,A,600s}$ value over the set threshold are compressed with a lossy-audio compression (e.g. 32 kbit/s MPEG-1 Audio Layer 3) method and transmitted to the cloud service. These can be later used to verify the noise source more accurately either with automatic methods or by the users.

2.2. Accessing data and visualisation

The measurements are accessible through a web-based user interface, which combines a large amount of measurements in an easily readable format by using data visualisation and data reports.

The sound pressure level (SPL) measurements can be filtered based on the sound source classification results to show measurements for assigned to particular sound source. In the service the measurement data is visualised in multiple ways: calendar heat-maps, graphs, and report tables. Example view from the portal is shown in Fig. 3.

The calendar heat-maps are used to visualise the average SPL values over certain time span (one day, one hour) with a colour of a calendar cell, an example of this is shown as measurement calendar in Fig. 3. The heat-map collapses SPL measurements within one hour into one number and decodes it into colours based on location-specific SPL limits. In preliminary studies, three colours were observed to give sufficient visualisation of measured SPL values. For the case study (see Section 4), colours are defined in following manner: green colour denotes SPL values under 45 dB, yellow denotes SPL values between 45 dB and 55 dB, and red denotes SPL value over 55 dB, the national limit for outdoor noise in residential areas. The limits shall be adjusted in accordance with the national law for each target. Only measurements associated to the targeted sound class are presented in the calendar.

The measurement graph is used to visualise the SPL values against the measurement time-stamp, an example of this is shown in the lower panel in Fig. 3. Three type of graphs are used to visualise measurement with differently assigned data: firstly showing all SPL measurements as such, secondly showing SPL measurements and sound source probability at current time interval denoted with colour intensity under the curve, and thirdly showing only SPL measurements assigned for targeted sound source. The noise monitoring location specific SPL limits (same as in calendar heat-map) are shown in the graph with horizontal lines.

In addition to the calendar and graph based visualisations, numerical measurement reports are used to show more exact values and analysis. The reports are used to show daily, weekly, monthly and yearly averages of the SPL measurements. Reports include also noise descriptors such as the day-evening-night level L_{den} introduced in the END [1], to give comprehensive figure of the noise levels over longer time segments. If needed, some higher level noise values like unbiased annoyance (UBA) [25] can be added to be calculated.

The portal provides different level information depending on the user account type. The monitoring site managers (system clients) can grant access for the people living close to the monitoring site (public users), and the services provides them easily approachable noise measurement summaries, and possibility to add feedback or comments on the measurement time-line, providing direct connection to the monitoring site management. The site manager or *a community liaison officer* can use the feedback from

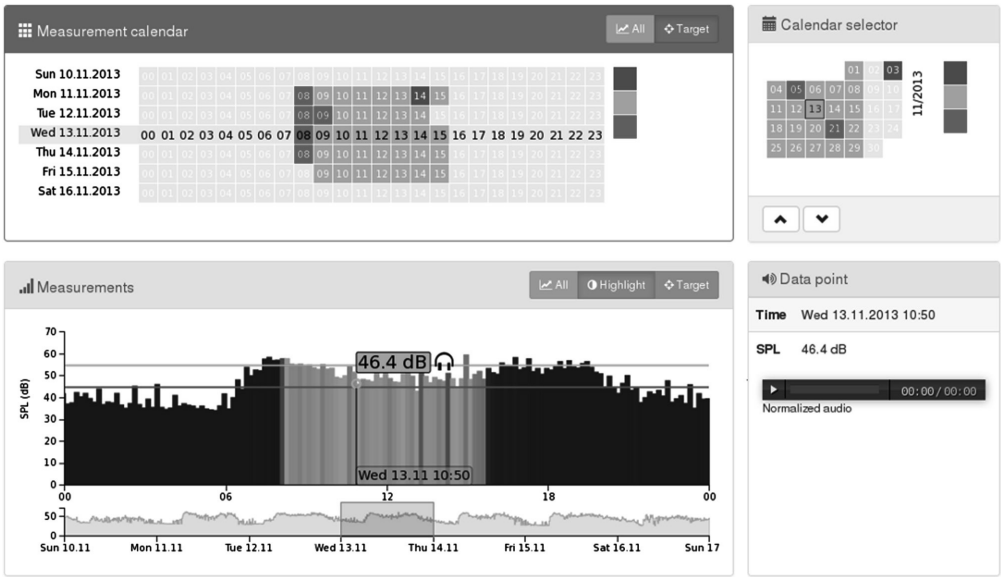

Fig. 3. Example view of the noise monitoring portal. In the upper right corner there is a calender selector with day view. By selecting a day, more detailed data of the day is shown in a calendar view and in a graph view at left. The measurement calendar shows measurements assigned to the target source, and in the graph view target presence is shown with intensity of the colour. An audio playback is shown in the lower right corner.

the public to react noise levels and types, and reply directly to the comments. If the commented time-stamp has a stored audio associated to it, the site manager can also audition it in this stage. Public access is important to make the noise monitoring transparent, and engage the public by giving them more active role how the monitoring results should be interpreted. This should alleviate many negative attitudes related environmental noise and noise monitoring. Administrative users like governmental authorities are presented accurate measurement reports to help to follow the average noise levels over longer time segments often used in official noise management.

2.3. Validity of the results

Standard IEC 61672-1:2002 [26] specifies three kinds of sound measuring instruments in two performance categories. Most of the commercially available sound level meters conform this standard requirements. There have been attempts to integrate sound measuring capabilities also to other instrumentation or devices, like mobile phones [27–29]. The driving force in these studies has been the need for spatially more representative data and fulfilling the accuracy requirements of the instrumentation for standardised measurements has not been addressed. The presented approach balances between these two extremes: the goal for design is to conform at least class 2 requirements, but still to keep the costs low so that the number of units in any implementation may be several times higher than using the conventional sound level meters. The calibration of the unit is performed using a conventional sound level calibrator equipped with a specially manufactured 1" adapter on the microphone of the unit.

Considering the uncertainty of the measurements, the fact is that the influence of instrumentation can be considered low [30] compared to the effect of environmental conditions [5]. The representativeness of data increases validity of an environmental noise measurement and this is achieved by both the increased spatial coverage and classified noise source data.

3. Automatic detection of target sources

In the proposed automatic target source detection system, noises are defined into two classes. Sounds propagating from the target sources belong to a *target* class, whereas interfering noises as well as silence belong to a *background* class. Examples of possible target sounds are plant noise and aircraft noise. Possible background noises may be caused by e.g. traffic, wind, rain, thunder, and birds. The activity of the target sources is detected by analysing continuous audio input and making binary classification between the background and the target. The audio input is the same as the signal used for SPL measurement, but without the A-weighting filter.

The detection system consists of two stages: the training stage and the monitoring stage (see Fig. 4). Acoustic models are learned from training examples, captured audio with manual annotation, in the training stage. The learned acoustic models are used to classify audio captured on a sensor, to detect the activity of target, in the monitoring stage. An example of the system output is given in Fig. 5. The training algorithm needs only annotation of target sounds. Traffic sounds, regarded as background in 5 are annotated to help understand the system output.

3.1. Acoustic features

Feature extraction transforms an audio signal into reduced representation. MFCCs are used as features in the proposed system. Mel-frequency cepstral coefficients (MFCCs) [13] have been originally proposed and widely used in speech recognition [14]. Afterwards, MFCCs have been proved to be effective in a wide range of audio processing applications such as sound event detection

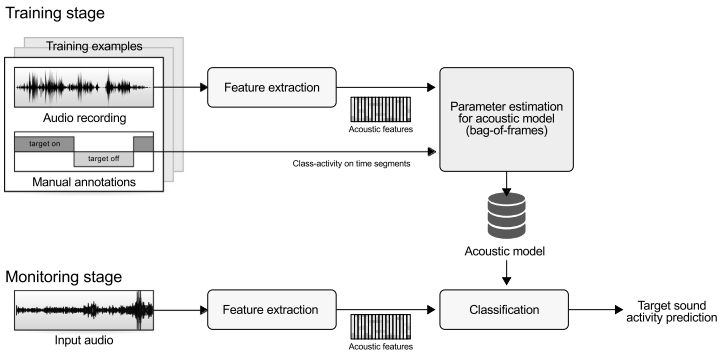

Fig. 4. Block diagram of the automatic target sound detection system.

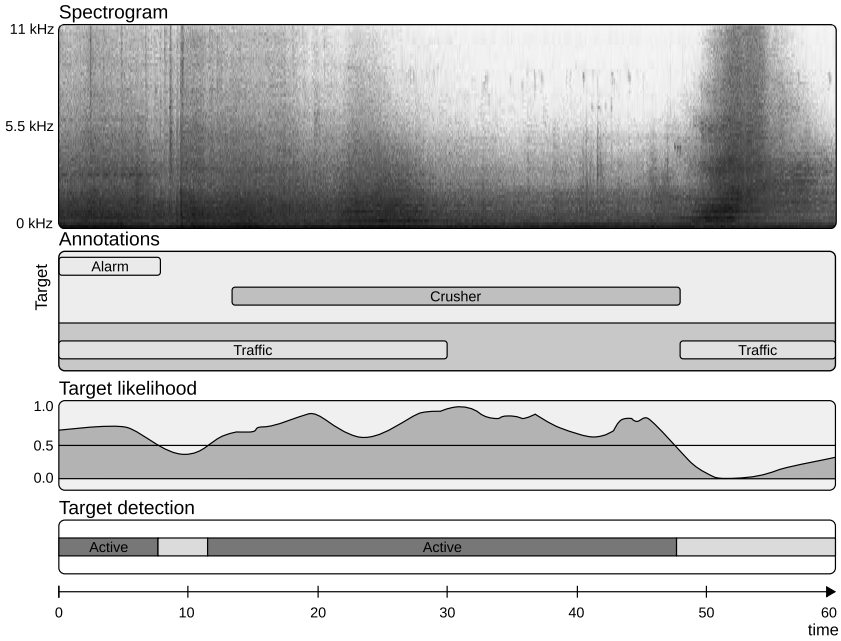

Fig. 5. Example of the target detection output using the GMM classifier. The top panel shows the spectrogram and the second panel illustrates the corresponding annotation. Traffic is regarded as background sound, whereas crusher and alarm are the target sounds. The third panel illustrates the target likelihood and decision threshold. The bottom panel illustrates the detection result as the system output.

[31–33] and speaker verification [16]. An audio signal is analysed within short frames (e.g. 100 ms) with 50% overlap. Every frame of signal is windowed with a Hamming window. Discrete Fourier transformation is performed on the windowed frames to obtain spectrum and the spectrum is wrapped into Mel scale. Logarithm of Mel-spectrum is performed with discrete cosine transformation to obtain Mel-cepstrum. Coefficients taken from Mel-cepstrum are called MFCCs. The proposed method uses the same classifier for sensors in different locations. However, the audio amplitude changes with the distance between a source and a microphone, which is reflected in the 0th coefficient. The 0th coefficient is usually excluded [14] to keep the features amplitude invariant. In order to provide temporal dynamic information across adjacent frames, deltas of MFCCs [34] are used in addition to static MFCCs. The first-order delta (Δ) is called differential of MFCCs and the second-order delta ($\Delta\Delta$) is called acceleration.

3.2. Supervised classifiers

Two types of supervised classifiers are investigated: Gaussian mixture model (GMM) as a representative of generative classifiers and artificial neural networks (ANN) as a representative of discriminative classifiers. A GMM represents a class by a distribution of its correspondent feature vectors [16]. The probability density func-

tion of a GMM for an observation $\mathbf{x}$ is the weighted average of its multi-variate Gaussian distribution components as

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i (2\pi)^{-\frac{k}{2}} |\mathbf{\Sigma}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}, \quad (1)$$

where M is the number of Gaussian components. The parameters of the density model are collectively denoted as $\lambda = \{w_i, \boldsymbol{\mu}_i, \mathbf{\Sigma}_i; i = 1 \dots M\}$. The weight, mean and covariance matrix of i:th Gaussian component are denoted as $w_i, \boldsymbol{\mu}_i, \mathbf{\Sigma}_i$, respectively, satisfying $\sum_{i=1}^{M} w_i = 1$. The GMM parameters of a class are iteratively estimated using the training data with the expectation maximisation (EM) algorithm. Classification can be made using GMMs by outputting the class whose GMM gives the highest likelihood on a input vector $\mathbf{x}$.

An ANN is used to estimate a function that yields desired outputs with given inputs [35]. The parameters of an ANN are estimated using training examples. A training example consists of an input feature vector $\mathbf{x}$ and a target vector $\mathbf{y}$. When an ANN is used as a classifier [21], the target output is typically a vector $\mathbf{y}$ with the size of C, the number of classes. Given the feature vector $\mathbf{x}$ from class i, the target vector entry y_i is set to 1, whereas other elements in target vector $\mathbf{y}$ are set to 0. Thus, the output of an optimised ANN classifier is interpreted as the activity indications of C classes of sound events. The activity indication is later called likelihood, since it is used in the same way as estimated likelihood in the GMM, though the indication is not a probability measurement. In the proposed system the multilayer perceptron (MLP) [36], which is a basic type of ANN, was used.

Let us denote input layer as $\mathbf{h}^1 = \mathbf{x}$ and the values of kth layer as $\mathbf{h}^k$. The values of the next layer $\mathbf{h}^{k+1}$ is calculated as

$$\mathbf{g}^k = \mathbf{W}^k \mathbf{h}^k + \mathbf{b}^k, \quad 2 \leqslant k < L \quad (2)$$

$$\mathbf{h}^{k+1} = \mathcal{F}(\mathbf{g}^k), \quad (3)$$

where $\mathbf{W}^k \in \mathbb{R}^{S_k \times S_{k+1}}$ is the weight matrix between layer k and layer $k+1, S_k$ being the number of neurons in layer k. The bias vector of layer k is denoted as $\mathbf{b}^k$, which can be considered as the weights for an additional all one's input vector. An activation function $\mathcal{F}$ is the applied element-wise on the linear transformation output. L is the total number of layers in the ANN. In the developed system, maxout function as activation function for hidden layers and logistic sigmoid function for output layer was used. Maxout is an unbounded function whereas sigmoid function ranges between 0 and 1. It has been shown that using two maxout layers with enough neurons can approximate any continuous functions [37]. In the optimisation, a cost function is a measure of difference between the obtained neural network outputs and target outputs. Kull-Leibler divergence is used as cost function and the parameters, weight matrices ($\mathbf{W}^k$) and bias vectors ($\mathbf{b}^k$), are optimised using the stochastic gradient descent algorithm.

3.3. Training and monitoring

Supervised learning requires a set of training examples, i.e., audio signals with manual annotations, at the training stage. Feature vectors of target class are derived from the time segments annotated as target sounds, whereas all other frames are used to represent background class. The extracted features (MFCCs) are collected for each class according to the annotations. When GMM is used, the features are used to estimate the feature distributions of each class. When ANN is used, the target outputs are $[1, 0]$ and $[0, 1]$ for feature vectors corresponding to the background class and the target class, respectively.

At the monitoring stage, a detection is made in one second non-overlapping segments. For each class, a score is computed as the sum of log-likelihoods (the logarithm of the likelihoods) of each frame in the corresponding second. The target likelihood in Fig. 5 is calculated as the score of target class divided by the sum score from all classes. The target sound source is detected as being active when the target likelihood is over a threshold (default value 0.5), otherwise inactive. The threshold can be tuned in case that precision is more important than recall, or vice versa. The precision and recall are later introduced in Section 4.3. Fig. 3 illustrates the noise portal that represents the estimated target activity in long term (1 h), taking majority vote from the activity outputs of corresponding seconds.

4. A case study: rock-crushing plant

A case study was made on the noise measurement of a rock crushing plant – a typical environmental noise assessment with nearby habitation. The feasibility of the proposed concept was evaluated with one sensor node next to the plant. The plant has regular working hours, thus the reliability of the target activity detection could be easily verified.

4.1. Measurement setup

The audio data was captured near a rock crushing plant (Fig. 6). The location of the sensor is indicated by a red triangle. The location of the nearest habitation house is indicated by the blue square. The most prominent sound sources in the plant are two rock-rushers denoted as red circles: a fixed rock-crusher and a mobile rock-crusher. The distance between the sensor and the fixed rock-crusher was about 280 m measured from their GPS coordinates and the distance between the sensor and the mobile rock-crusher was about 500 m. Even though the mobile rock-crusher is able to change its position, it was stationary during the case study. Beside the rock-crushers, another significant type of a target sound was made by lift-trucks, which feeded rocks to the crushers and distributed the produced stones. The sensor was located close to a road, near a forest.

4.2. Captured noise data

Three minutes of audio was continuously captured every 10 min, making a total of 432 min for each day. All types of noise generated by the working activity of the plant was collectively defined as the target class, including rock crushing, lifting-truck sounds, and alarm sounds from the machinery. On the contrary, traffic noise coming from the road and the noise generated by the wind and the trees were two significant types of background sound sources. Example sound spectra of rock crushing, a car passing, and wind is given in Fig. 7.

Two days of audio data were annotated and used to develop and evaluate the target detection system. The data was manually annotated (like in Fig. 5). The rock crushing activity is rather continuous and long-lasting, which made the annotation easy in most cases. In a few cases, the onset and offset of the target sound were hard to determine due to overlapping sound sources. In these cases, a 0.4 s uncertainty was associated to the onset or offset.

4.3. Evaluation setup

A quantitative evaluation was made on the target noise detection performance with temporal resolution of one second. A two-folded validation, swapping the data of day 1 and day 2 for training and testing, was used. A detection output, either active or inactive,

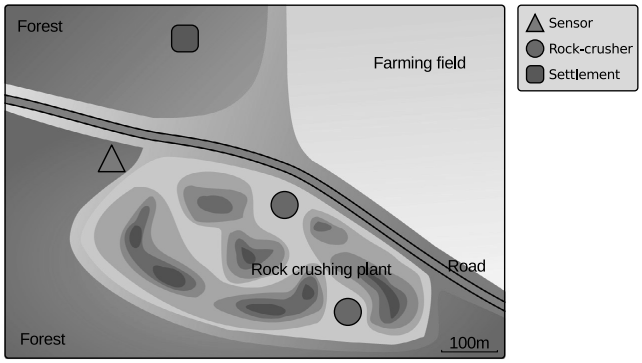

Fig. 6. Map of the rock crushing plant that was the target of the case study.

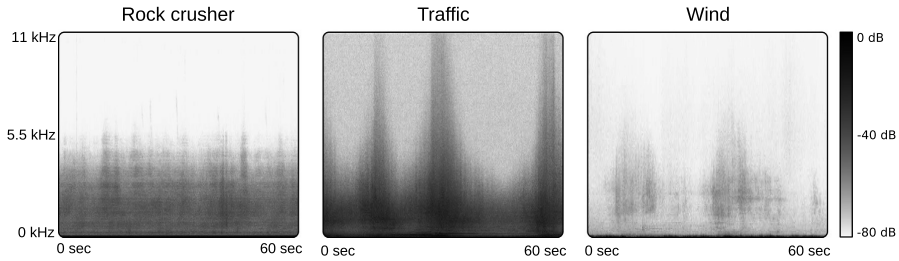

Fig. 7. Example sound spectra for rock crushing, a car passing, and wind from left to right.

was obtained through the proposed system for every one-second segment. The ground truth of a one second segment was seen as a target, when the target sound lasted longer than 0.1 s according to the annotation, otherwise judged as background noise.

The target source detection performance was evaluated using F-score [38], which is often used to evaluate binary classification performance. The F-score is calculated as a harmonic mean of precision and recall. Precision is the fraction of the predicted target activities that are correct, whereas recall is the fraction of the actual target activities that are predicted.

In order to study the feasibility of real-time execution and to find the most relevant factor to computation time, the computation time was evaluated for feature extraction and classification. The target detection algorithm was implemented in C++ and was run in a sensor node. The file read/write, SPL measurement, feature extraction, and classification process takes 51 s for a one minute signal, fast enough for real-time execution (85%). The sensor implementation was used as a benchmark and computation time of other feature extractors and classifiers were estimated using a Python implementation, assuming that the computation time had always the same proportion between the implementation in the sensor and in any other computer.

The developed classifier was imported to the sensor and it continuously performed noise measurement and source classification for 50 days. A reliability evaluation was made by examining the results transmitted to the web portal.

4.4. Evaluated classification systems

The acoustic features (MFCCs) from an audio signal which was sampled at 22,055 Hz. The audio signal was further analysed at a frame length of 100 ms with a 50% overlap between neighbouring frames and windowed with a Hamming window. 4096 point discrete Fourier transform and 40 Mel bands were used. Mel cepstral coefficients from the first to kth were used as features. The number of coefficients (k) was studied as a variable. In addition to static MFCCs, the deltas were calculated using four preceding frames and four succeeding frames to represent temporal dynamics. The features were normalised to zero mean and unit variance was based on the statistics of the training dataset.

A single variable was changed at a time from the default setup to test the variable. Four variables were tested: the number of coefficients, the temporal dynamic features (deltas), the time-domain filters, and the frame length. The variable value that achieved the best performance was used to determine the next variable. To evaluate the performance of the feature extraction variables, a GMM with $M = 16$ components was used as a classifier.

The best achieving feature extraction setup was used to evaluate the classifiers. GMMs with a different number of Gaussian components $M = \{1, 2, 4, 8, 16, 32\}$ in Eq. (1) using diagonal covariance matrices and ANNs with two hidden layers, each having $\{10, 30, 50, 100\}$ neurons, were tested. Python toolboxes scikit-learn and pylearn2 were used in the implementation of the GMM classifier and the ANN classifier, respectively.

4.5. Results

The parts of the system were evaluated to select the features and the classifiers. A quantitative evaluation of the results is shown in Table 1. The selected values are shown in bold font and the computational requirements for the feature extraction is expressed as a time ratio to the estimate of real-time. Besides the detection

Table 1
The results of the evaluation of the acoustic features.

Studied variable	Variable value	F_1-score	Feature extraction time
Number of coefficients	8	0.926	0.51×
	13	**0.927**	0.51×
	20	0.927	0.51×
Temporal dynamics	MFCC	0.927	0.51×
	MFCC+ Δ	**0.931**	0.51×
	MFCC+ Δ + ΔΔ	0.917	0.51×
Time-domain filter	**No filter**	**0.931**	0.51×
	Pre-emphasis	0.885	0.54×
	A-filter	0.930	0.64×
Frame length	50 ms	0.898	0.99×
	100 ms	**0.931**	0.51×
	200 ms	0.914	0.27×

Table 2
The results of the evaluation of the classifiers.

Classifier	Parameters	F_1-score	Classification time
GMM	M = 1	0.795	0.10×
	M = 2	0.870	0.10×
	M = 4	0.925	0.10×
	M = 8	0.928	0.11×
	M = 16	0.931	0.12×
	M = 32	0.934	0.14×
ANN	10 × 2	0.904	0.10×
	30 × 2	0.934	0.10×
	50 × 2	0.938	0.11×
	100 × 2	0.938	0.12×

performance, the estimated computation time (the feature extraction time) is shown for the sensor implementation compared to real time.

A small effect to the classification performance was found by changing the number of the cepstral coefficients. In Table 2, M denotes the number of Gaussian components used in a mixture model and the parameters of ANN marked as $a \times b$ means a neural network with b hidden layers and a neurons per layer. 13 coefficients were selected, because those gave the same performance as using 20 coefficients and a smaller number of coefficients makes classification faster. The best performance among the studied temporal dynamic feature combinations is gained by using MFCCs with only first-order delta. Adding a second-order delta did not give any improvement, perhaps because a rather long frame length (100 ms) was used and the first-order deltas already covered 500 ms temporal dynamics. Based on the results, imposing time-domain filters (a A-weighting filter [39] and a pre-emphasis filter [16]) is not justified. The frame length is clearly the key factor contributing to the computation time. The frame length of 100 ms is the best choice, which leads to the best classification performance and is capable in real-time execution.

ANN achieves the best F-score and takes the least time to compute. However, the difference between ANN and GMM is rather small. The estimated classification time does not largely depend on the number of the model parameters. This suggests that it takes the most time for overhead operations such as copying the features, when compared to computing likelihoods with the classifier. This computation time could be further reduced with a better implementation.

The computation in the sensor includes reading the audio stream, SPL measurement, feature extraction, classification, and transmitting results. A feasible implementation would use less than 70% of real-time for feature extraction and classification. This has to be taken into account when choosing the features and the classifier. The leading factor of the computation time in the source classification algorithm is the audio analysis frame length, which determines the number of frames to process. In comparison, the computation time is not much affected by other the factors such as the neural network topology.

To evaluate the reliability of the proposed system, the hour-level measurement and detection results visualised in the web portal (Fig. 3) were examined. The sensors continuously performed noise measurement and source classification for 50 days and were able to transmit the results of every single hour, though some results were received with a delay of hours. It was assumed that the work in the plant could begin one hour later and end one hour earlier than the regular working hours (Mon-Fri, 8:00–15:00). With this assumption, almost all the target detection results were correct (1198/1200).

4.6. Required amount of annotated recordings

In the training material, about four hours of audio, the total number of feature vectors was about 300 000. A reduced size of the training material was tested by using every second or every fourth recording in time order. The learned classifiers achieved F-scores 0.913 and 0.924. Thus, it is sufficient to use about one hour of annotated recordings to achieve a decent classifier. When using the first half or the second half of a day data for training, the learned classifier achieved less than 0.8 F-score. This suggests that the training material should contain recordings from different times of a day to cover the most of the variability of the environmental sounds.

In environmental sound classification, a training set with a few hours can currently be regarded as a large dataset. For example, UrbanSound8K [40] contains one hour for each of 10 classes. As an example of small datasets, ESC-10 [41] contains at most 200 s audio for each of 10 classes.

In the reliability evaluation, it was shown that the system was able to do accurate hour-level classification in varying weather conditions by using annotated data of one day captured in good weather conditions. However, annotated recordings in more diverse conditions are typically required to achieve a similar accuracy as obtained in the quantitative evaluation with one second temporal resolution.

5. Further analysis and future work

5.1. Selection of classifier

The performance of the classifiers GMM and ANN was practically the same in the evaluation. The selection between GMM and ANN should be based on other aspects. Adding a new class to the GMM classifier is easier, since statistics of the existing classes stay unchanged when a new class is added. In contrast to the GMM, the ANN has to re-estimate all the parameters to introduce a new class. Another benefit of using the GMM is that it is easier to adapt so that the classifier could adapt to small changes of the environment over time, using maximum a posteriori [16] algorithm. Typically ANN outperforms GMM when the number of classes is large. The number of the ANN parameters does not significantly increase with the number of classes, whereas the number of GMMs depends linearly from the number of classes. For example, ANN and GMM used approximately the same time in the computational time test for the binary classification. If there were ten classes in the same setup, the classification with ANN would have been about five times faster than GMM. In the one-minute audio test on a ten class case, it took 0.32 s for the ANN classifier and

1.48 s for the GMM classifier using a Python implementation in a desktop computer.

5.2. Extension to monitoring multiple target classes

The same algorithm could be used in noise measurement scenarios involving multiple noise types. In addition to the rock crushing case study described above, a preliminary study on a set of noise samples from the port of Dublin was made. In this case, a classification system with multiple noise source classes was built. There are many kind of sound sources in the port area, some of them being also present in the neighbouring environment. The noise data was annotated with an interactive clustering method, by which a cluster of sounds were annotated or skipped at once. With this method, the annotation was fast but less accurate.

Ten classes of sound sources were present in the evaluation and the average recognition rate was 81%. The ten classes were alarm sounds, bird chirping, mild fans, strong fans, traffic noise, engine noise, footsteps, musical concert, raining, and wind blocking the microphone. It should be noted that the results might be optimistic since the segments not clearly belonging to any of the classes might have been skipped in the annotation with interactive clustering.

5.3. Sensor network

In the future, all the data from a large number of various networked sources, already available or from the autonomous smart sensors, will be centralised to a cloud service, where the data is accessible to a various groups of people: public, authorities, and to the dedicated users. The data will be made available for all the purposes it is needed: mapping and monitoring of emissions, noise, aerosols etc. It is possible to get accumulated standardised descriptors and conventional reports for various purposes. Also, it is possible to comment the visualised, and, possibly auralised, results on a time line to make feedback possible to the responsible party.

To increase the validity of the classification, multiple sensors could be used to also analyse the direction of arrival of sounds [42]. In future, the final outcome of an environmental noise assessment will be an annoyance map of an area, reported with the level of uncertainty. Further, when the needs go beyond the current legislative values and limits, it is possible to calculate higher level descriptors like unbiased noise annoyance (e.g. UBA [25]), or some other psychoacoustical descriptors at the sensor. The solar powered sensor was optimised for average summer conditions, so that the batteries keep the system running at the night time. However, during a long period when the direct sunlight is limited, or does not exist at all (e.g. winter north of Arctic Circle), external power is needed.

6. Conclusions

It was shown that environmental noise monitoring could be enhanced by separating between the target and interfering noise sources and implementing this approach to the sensor level. Also, an autonomous and a low-cost sensor implementation with a connection to a cloud service was introduced.

A credit-card-sized single-board computer, Raspberry Pi, was found to be powerful enough for automatic source classification. A solar-powered sensor was demonstrated to allow measurements in locations without power outlet.

The activity of the noise source was detected by making a binary classification between the target and the background. Mel-frequency cepstral coefficients were used as acoustic features and the classification was made using a supervised classifier (GMM and ANN), learned from annotated audio recordings.

The performance of the developed methods was evaluated in a rock crushing plant case study. The quantitative evaluation showed that the noise source classification using the proposed approach was accurate enough: on a temporal resolution of one second, F-score of 0.938 with the best investigated classifier was achieved. The system was run for 50 days and the results of the developed classifier matched well with the regular working hours of the rock crushing plant.

Also, a cloud service and a noise portal were introduced. The sensors transmitted the results to the cloud service and the portal was for visualisation of the results, statistical analysis, and data archiving. This approach makes it possible to extend the system towards noise management and, due to the minimal cost per sensor unit, towards real-time noise mapping with real measured data. By using this approach, the reliability, validity, and the spatial coverage of environmental noise monitoring will be increased.

References

[1] Directive, EN, Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 Relating to the Assessment and Management of Environmental Noise, Jun. 2002.
[2] Guarinoni M, Ganzleben C, Murphy E, Jurkiewicz K. Towards a comprehensive noise strategy. In: Environment, public health and food safety, no. IP/A/ENVI/ST/2012-17, PE 492.459 in European Parliament's Committee on Environment, Public Health and Food Safety, Directorate General for Internal Policies, Policy Department A: Economic and Scientific Policy, B-1047 Brussels; 2012, p. 86.
[3] Maijala PP. Excess attenuation and meteorological data in a long term measurement. In: Proceedings of the international conference on noise control engineering, Tampere, Finland, 30 May – 1 June, (Euronoise 2006), no. SS20-392 in euro-noise series, EAA, Tampere, Finland; 2006, p. 1–6.
[4] Maijala PP. A set-up for long term sound propagation measurements. In: Proceedings of the International congress on noise control engineering, Tampere, Finland, 30 May – 1 June, (Euronoise 2006), no. SS20-390 in euro-noise series, EAA, Tampere, Finland; 2006, p. 1–6.
[5] Maijala PP. A measurement-based statistical model to evaluate uncertainty in long-range noise assessments [Doctoral dissertation]. P.O. Box 1000, FI-02044 VTT, Finland: Tampere University of Technology; 2013.
[6] ISO. Standard ISO 1996-2:2007. Acoustics – description, measurement and assessment of environmental noise – Part 2: Determination of environmental noise levels; 2007.
[7] Konishi K, Tanioku Y, Maekawa Z. Long time measurement of long range sound propagation over an ocean surface. Appl Acoust 2000;61(2):149–72.
[8] Hole LR. Sound propagation in the atmospheric boundary layer: an experimental and theoretical study [Ph.D. thesis]. Geophysical Institute, University of Bergen; 1998.
[9] Gauvreau B. Long-term experimental database for environmental acoustics. Appl Acoust 2013;74(7):958–67.
[10] Maijala PP, Ojanen O. Long-term measurements of sound propagation in Finland (invited paper). In: Proceedings of the international conference on noise control engineering, Honolulu, Hawaii, Dec. 3–6 (Inter-noise 2006), no. 326 in INTER-NOISE Series, INCE, Honolulu, Hawaii, USA. p. 1–10.
[11] Manvella D. From noise monitoring to noise management – a better way to deal with noise issues. Inter-noise and noise-con congress and conference proceedings, vol. 250. Institute of Noise Control Engineering; 2015. p. 2473–84.
[12] Leskinen A, Hjort R, Saine K, Gao Z. Aures – the advanced environment noise monitoring system – Leq(A) or new measurement technology? Inter-noise and noise-con congress and conference proceedings, vol. 249. Institute of Noise Control Engineering; 2014. p. 2411–9.
[13] Noll M. Short-time spectrum and cepstrum techniques for vocal-pitch detection. J Acoust Soc Am 1964:296–302.
[14] Rabiner L, Juang B-H. Fundamentals of speech recognition. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.; 1993.
[15] Pampalk E. Computational models of music similarity and their application in music information retrieval [Ph.D. thesis]. Vienna, Austria: Vienna University of Technology; 2006. March.
[16] Reynolds DA, Quatieri TF, Dunn RB. Speaker verification using adapted gaussian mixture models. Digital signal processing, vol. 10. p. 19–41.
[17] Sakurai M, Sakai H, Ando Y. A computational software for noise measurement and towards its identification. J Sound Vib 2001;241(1):19–27.
[18] Fujii K, Sakurai M, Ando Y. Computer software for identification of noise source and automatic noise measurement. J Sound Vib 2004;277(3):573–82. fifth Japanese-Swedish Noise Symposium on Medical Effects.
[19] Mesaros A, Heittola T, Eronen A, Virtanen T. Acoustic event detection in real life recordings. In: 18th European signal processing conference. p. 1267–71.
[20] Oguzhan G, Virtanen T, Huttunen H. Recognition of acoustic events using deep neural networks. In: Proc. 22nd European Signal Processing Conference (EUSIPCO). p. 506–10.

[21] Cakir E, Heittola T, Huttunen H, Virtanen T. Polyphonic sound event detection using multi label deep neural networks. In: The International Joint Conference on Neural Networks 2015 (IJCNN 2015), Cill Airne, Eire.
[22] Culler D, Estrin D, Srivastava M. Overview of sensor networks. Computer 2004;37(8):41–9.
[23] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gen Comput Syst 2013;29(7):1645–60. including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications - Big Data, Scalable Analytics, and Beyond.
[24] El-Bendary N, Fouad MMM, Ramadan RA, Banerjee S, Hassanien AE. Smart environmental monitoring using wireless sensor networks. In: El Emary IMM, Ramakrishnan S, editors. Wireless sensor networks: from theory to applications. CRC Press; 2013. p. 799.
[25] Zwicker E. On the dependence of unbiased annoyance on loudness. Proceedings of inter-noise 1989, Newport Beach, CA, USA II 1989:809–14.
[26] IEC. Standard IEC 61672-1:2002. Electroacoustics - Sound Level Meters - Part 1: Specifications, May 2002.
[27] Kanjo E. NoiseSPY: a real-time mobile phone platform for urban noise monitoring and mapping. Mob Networks Appl 2010;15(4):562–74.
[28] Maisonneuve N, Stevens M, Ochab B. Participatory noise pollution monitoring using mobile phones. Inf Polity 2010;15(1, 2):51–71.
[29] Santini S, Ostermaier B, Adelmann R. On the use of sensor nodes and mobile phones for the assessment of noise pollution levels in urban environments. In: Proceedings of the 6th international conference on Networked sensing systems. IEEE Press; 2009. p. 31–8.
[30] Manvell D, Aflalo E. Uncertainties in environmental noise assessments – ISO 1996, effects of instrument class and residual sound. In: Proceedings of ForumAcusticum 2005, Budapest.
[31] Vuegen L, Van Den Broeck B, Karsmakers P, Gemmeke JF, Van hamme H, et al. An MFCC-GMM approach for event detection and classification. IEEE AASP challenge on detection and classification of acoustic scenes and events 2013:3.
[32] Valenzise G, Gerosa L, Tagliasacchi M, Antonacci F, Sarti A. Scream and gunshot detection and localization for audio-surveillance systems. In: Proceedings of the 2007 IEEE conference on advanced video and signal based surveillance, AVSS '07. Washington, DC, USA: IEEE Computer Society; 2007. p. 21–6.
[33] Ntalampiras S, Potamitis I, Fakotakis N. On acoustic surveillance of hazardous situations. In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP2009. p. 165–8.
[34] Huang X, Acero A, Hon H-W. Spoken language processing: a guide to theory, algorithm, and system development. 1 ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 2001.
[35] Haykin S. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 1998.
[36] Rumelhart DE, Hinton GE, Williams RJ. Neurocomputing: foundations of research. Cambridge, MA, USA: MIT Press; 1988. p. 673–95. Ch. Learning Internal Representations by Error Propagation.
[37] Goodfellow IJ, Warde-Farley D, Mirza M, Courville AC, Bengio Y. Maxout networks. In: International conference of machine learning. Jounal of Manchine Learning Proceedings, vol. 28. p. 1319–27.
[38] Rijsbergen CJV. Information retrieval. 2nd ed. Newton, MA, USA: Butterworth-Heinemann; 1979.
[39] Fletcher H, Munson WA. Loudness, its definition, measurement and calculation. J Acoust Soc Am 1933;5(2):82–108.
[40] Salamon J, Jacoby C, Bello JP. A dataset and taxonomy for urban sound research. In: Proceedings of the ACM international conference on multimedia, MM '14. p. 1041–4.
[41] Piczak KJ. ESC: dataset for environmental sound classification. In: Proceedings of the ACM international conference on multimedia (ACM). p. 1015–8.
[42] Genescá M, Romeu J, Pámies T, Sánchez A. Real time aircraft fly-over noise discrimination. J Sound Vib 2009;323(1–2):112–29.

PUBLICATION II

Active learning for sound event classification by clustering unlabeled data

Zhao S.Y., T. Heittola and T. Virtanen

2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, 751–755

ACTIVE LEARNING FOR SOUND EVENT CLASSIFICATION BY CLUSTERING UNLABELED DATA

Zhao Shuyang Toni Heittola Tuomas Virtanen

Tampere University of Technology, Finland.

ABSTRACT

This paper proposes a novel active learning method to save annotation effort when preparing material to train sound event classifiers. K-medoids clustering is performed on unlabeled sound segments, and medoids of clusters are presented to annotators for labeling. The annotated label for a medoid is used to derive predicted labels for other cluster members. The obtained labels are used to build a classifier using supervised training. The accuracy of the resulted classifier is used to evaluate the performance of the proposed method. The evaluation made on a public environmental sound dataset shows that the proposed method outperforms reference methods (random sampling, certainty-based active learning and semi-supervised learning) with all simulated labeling budgets, the number of available labeling responses. Through all the experiments, the proposed method saves 50%-60% labeling budget to achieve the same accuracy, with respect to the best reference method.

Index Terms: active learning, sound event classification, K-medoids clustering

1. INTRODUCTION

Sound event classification [1] and detection [2] has many applications such as noise monitoring [3, 4, 5], surveillance [6, 7] and home service robots [8]. The development of sound event classification and detection applications requires annotated recordings. Recordings can be made continuously all day around, almost effortlessly. However, reliable annotation takes at least the duration of a recording. As a result, the annotation work is quite often the main cost to build a sound event classifier. To aim at this situation, we attempt a method that optimizes the classification performance with a limited annotation effort, utilizing an abundant amount of audio data that is much more than the amount that can be afforded to annotate.

The maximum number of labels that can be assigned is called a *labeling budget*, which is used to quantify a limited annotation effort. When labeling budget is small, there are two established techniques to utilize the abundant amount of unlabeled data: active learning and semi-supervised learning.

An active learning algorithm actively asks for labeling responses on data selected by the algorithm from a set of unlabeled data. An unlabeled data point is called a sample and the selection of samples to be labeled is called sampling; after labeling, a labeled data point and its label constitutes a training example. Active learning algorithms controls the sampling, in order to avoid redundant examples to optimize the efficiency of labeling effort. Though other types of active learning methods exist, only certainty-based active learning (CRTAL) [9] has been studied in the field of acoustic pattern recognition. It has been proposed to speech recognition in [10]. In certainty-based active learning methods, a small set of samples (selected by the annotator or randomly) are annotated in the beginning. The annotated labels are used to train a classifier and unlabeled samples are classified. A batch of samples with the lowest classification certainties are presented to the annotator for labeling. The classifier is updated after adding new labels to the training material. An experiment on speech recognition has shown that the amount of labels needed to achieve a target word accuracy can be reduced by 60% using CRTAL [11], compared to random sampling.

Semi-supervised learning (SSL) assigns predicted labels to unlabeled data so that unlabeled data is utilized as training examples according to predicted labels. Expectation-maximization based semi-supervised learning has been studied for various acoustic pattern recognition problems such as speaker identification [12] and musical instrument recognition [13]. These methods start by training an initial classifier with labeled data, and they iteratively update predicted labels of either a batch or all unlabeled data. The final classifier is obtained by training with both annotated labels and predicted labels. Gender identification and speaker identification error rates are generally halved using semi-supervised learning with varying proportion of labeled data [12].

All the above-mentioned methods rely on a classifier for uncertainty sampling or label prediction. However, it would require much labeling effort as an overhead to achieve a classifier that produces reasonable classification outputs (predicted class and certainty). As is shown in [11], as long as less than 10% (about 3000) utterances are labeled, performance of CRTAL is behind random sampling. An ideal way to deal with a small labeling budget is to utilize the internal structure of the dataset so that the method starts to outperform random sampling from the very beginning of a labeling process.

We propose a method to optimize the sound event classification performance when labeling budget is limited and only a small portion of data can be annotated. The proposed method is called medoid-based active learning (MAL). K-medoids clustering is performed on sound segments, and the centroids of clusters (medoids) are selected for labeling. The label assigned to a medoid is used to derive predicted labels for other cluster members. An advantage of MAL over traditional SSL and CRTAL is that it does not depend on a model that would require many labels as an overhead to achieve reliable performance on uncerntainty sampling and label prediction. In the evaluation, labels are produced to a training dataset through the proposed method or reference methods, simulating a limited number of labeling responses. A classifier is trained according to the produced labels and its classification accuracy on a test dataset is used to evaluate the performance of the whole process. Selecting cluster representatives for labeling has been originally proposed for text classification in [14], but it does not use representatives to predict labels. Similar studies have not been found in the field of acoustic pattern recognition.

The proposed method is described in Section 2. The evaluation

This work has received funding from the European Research Council under ERC Grant Agreement 637422 EVERYSOUND.

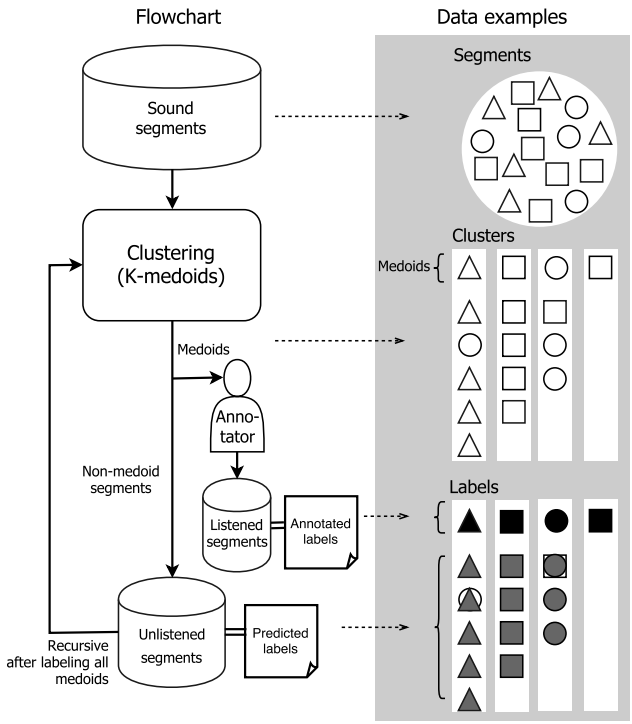

Fig. 1. Overview of the proposed method. The shape of a geometric drawing in data examples represents the ground truth class or label of a segment. Ground truth classes, annotated labels and predicted labels are represented by unfilled drawings, black filled drawings and gray filled drawings, respectively.

of the proposed system and the discussion about the results is given in Section 3. The conclusion is drawn in Section 4.

2. THE PROPOSED METHOD

The procedure of the proposed method is shown in Figure 1. The proposed method takes sound segments as input and labels of segments are produced as the output. Sound segments are typically sliced from audio recordings. The production of labels requires an annotator who listens to presented segments and assign labels for them. The labels are chosen from a closed set of pre-defined classes.

Segments in the dataset are originally unlabeled and marked as unlistened. Each segment in the dataset is represented by a multi-variate Gaussian distribution and the dissimilarity between a pair of segments is measured by Kullback-Leibler (KL) divergence. Segments are clustered using K-medoids algorithm based on the dissimilarity to each other. The medoid of each cluster is presented to annotators for labeling. Medoids are the representatives of local distributions so that they have two useful properties. Firstly, medoids are assured to span different local distributions, thus redundant examples densely distributed within a small area can be avoided. Secondly, a cluster consists of segments around the medoid, thus predicted labels for other cluster members can be derived from the medoid. In case the labeling budget is more than the number of clusters, the annotation proceeds with another round of clustering on unlistened segments. The details of the processing is described in more detail in the following subsections.

2.1. Sound segment representation

Mel-frequency cepstral coefficients (MFCCs), its first-order and second-order derivatives are used as acoustic features. A sound segment is represented by a multi-variate Gaussian distribution, based on the mean and the covariance of the corresponding features. In a preliminary study, using a diagonal covariance matrices gave better performance than using full covariance matrices, thus diagonal covariance matrices are used in this study.

2.2. Segment-to-segment dissimilarity measurement

Dissimilarity measurement between segments is needed to perform clustering. Symmetric KL divergence is a dissimilarity measurement between multi-variate Gaussian pairs, which has been used in various applications such as in music information retrieval [15] and audio texture creation [16]. Symmetric KL divergence is also used in this study to measure the dissimilarity between a pair of sound segments. The KL divergence between two multi-variate Gaussian distributions $\mathcal{P}_0$ and $\mathcal{P}_1$ is calculated as

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{P}_0\|\mathcal{P}_1) =& \frac{1}{2}(\mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) \\ &+ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \\ &+ \ln(\frac{\det \boldsymbol{\Sigma}_1}{\det \boldsymbol{\Sigma}_0}) - k), \end{aligned} \quad (1)$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are mean and covariance of distribution $\mathcal{P}_0$, respectively. The mean and covariance of distribution $\mathcal{P}_1$ are denoted as $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$.

KL divergence is not a commutative operation so that $D_{\mathrm{KL}}(\mathcal{P}_0\|\mathcal{P}_1)$ is different from $D_{\mathrm{KL}}(\mathcal{P}_1\|\mathcal{P}_0)$. In order to obtain a symmetric dissimilarity matrix, the average of both way KL divergence is used to measure the dissimilarity between two segments as

$$D(\mathcal{P}_0\|\mathcal{P}_1) = D(\mathcal{P}_1\|\mathcal{P}_0) = \frac{D_{\mathrm{KL}}(\mathcal{P}_0\|\mathcal{P}_1) + D_{\mathrm{KL}}(\mathcal{P}_1\|\mathcal{P}_0)}{2}. \quad (2)$$

2.3. K-medoids clustering

K-medoids clustering algorithm [17, 18] is performed based on the segment-to-segment distance matrix. K-medoids is a partitioning-based clustering algorithm, similar to K-means. K-medoids uses a data point in the dataset as a centroid whereas K-means uses an arbitrary point in the coordinate space as a centroid. K-medoids typically outperforms K-means, in terms of accuracy, and the advantage increases with the size of the dataset [19]. Furthermore, a medoid, as the centroid of a cluster, is intuitively the best sample to estimate the most frequent class in a cluster if only one sample can be taken.

In a bit more detail, K-medoids is performed by assigning each segment to the nearest medoid among all k medoids. The medoids are initialized and iteratively updated to minimize the total distance of all segments to the nearest medoids until no medoid can be swapped to reduce the total distance.

The initialization of medoids is based on farthest-first traversal [20]. Farthest-first traversal has been proved to give an efficient approximation of k-center problem [21]. A traversed set starts as a singleton of a random segment. The farthest segment to the current traversed set (the distance from a point x to a set $\mathcal{S}$ is defined as $d(x,\mathcal{S}) = min_{y\in\mathcal{S}} d(x,y)$) is added to the traversed set until the traversed set reaches the size of K. The traversed set is then used as the initial medoids.

The choice of the number of clusters k gives a trade-off between bigger cluster size (more predicted labels can be derived from a single label assignment) and better accuracy of predicted labels. Let us denote the number of unlistened segments as n. We choose $k = n/4$, which can be interpreted that the average size of clusters is four.

2.4. Assigning labels

The medoids of clusters are presented to an annotator in a sequence sorted by cluster size in descending order. Only one medoid is played at a time and the annotator assign label to the medoid by selecting a class from a list of pre-defined classes. Assigning a label consumes labeling budget by one. The label assigned to a medoid is seen as an *annotated* label. The label of the medoid is derived as *predicted* labels for the rest of cluster members. Largest clusters are labeled first so that high number of predicted labels can be derived with low listening budget.

2.5. Recursive process

Initially, all the segments are flagged as *unlistened*. Once a medoid segment is annotated, the segment is flagged as *listened*. The target situation is small labeling budget so that we do not aim on an optimal performance when the budget is more than the number of clusters. In case all medoids are annotated, we simply perform another round of clustering on unlistened segments and the annotation process continues with medoids in the latest round of clustering. Annotated labels overrule predicted labels received in previous rounds.

If the listening budget is sufficient so that multiple rounds of clustering have been performed, there would be multiple, possibly different predicted labels given to an unlistened segment. In supervised learning, all the different predicted labels for an unlistened segment are used, by taking the segment as an training example of each labeled class.

3. EVALUATION

The performance of the proposed method is evaluated as the classification accuracy using labels produced with the proposed method.

3.1. Dataset

The goal of the proposed method is to save annotation effort. In order to approximate the target situation, the used dataset has to be large enough so that reducing annotation effort is worthy attempting. In addition, a public dataset designed for sound event classification is preferred.

We use UrbanSound8K dataset [22], a public environmental sound dataset, consisting of 10 classes of sound events: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. All the sounds in the dataset are real field-recordings from urban environments. The dataset includes 8 732 labeled sound segments with maximum duration of 4 seconds, totaling 8.75 hours. A 10-folds division is provided by the dataset for cross validation. The division is made using a random allocation process that keeps segments originating from the same recordings allocated to the same fold, meanwhile trying to balance the number of segments per fold for each sound class.

3.2. Experimental setup

MFCCs are used as frame-wise features. The audio signal is divided into frames with 24 ms length and 50% frame overlap. We compute 1st to 25th MFCCs from 40 Mel bands between 25 Hz and 22 050 Hz. To calculate the segment-to-segment distances, the mean and covariance of MFCCs are used as is discussed in Section 2.2. In supervised learning, the following summary statistics of MFCCs are used as segment-wise features: minimum, maximum, median, mean, variance, skewness, kurtosis and the median and variance of the first and second derivatives.

In each round of evaluation, nine folds are used for training and one fold is used for testing. The labels provided by the dataset are used as ground truth. In a training set, the ground truth labels are initially all hidden. A labeling budget m allows a learning algorithm to query labeling responses for up to m segments. The labels obtained directly through labeling responses are called annotated labels, whereas other labels generated using the proposed method or SSL are called predicted labels.

Two annotators are simulated: an oracle annotator that always answers the ground truth and an artificial weak annotator [23] that produces noisy labels. The labeling accuracy of our artificial weak annotator is set to 75%, which is the lowest reported human sound event recognition rate in found studies [5, 8, 24, 25]. The probabilities that the artificial annotator mislabels a class to any other classes are even.

Obtained labels are used to perform supervised learning. Support vector machine (SVM) with radial basis function as kernel is used as classification model. Since this study does not aim on optimal parametrization, we use default settings of Python Scikit-learn [26]. A training example consists of a segment-wise feature vector and a target class according to the label.

Since the distribution of classes in the dataset is not even, we use unweighted accuracy to weigh different classes the same regardless to the number of instances. The classification accuracy is reported averaging the accuracy across all 10 folds. There are random elements (medoid initialization, random sampling and labeling errors from the weak annotator) in the experiments that affect on the performance. Therefore, all the experiments are repeated five times and the averaged results are reported.

3.3. Reference methods

Random sampling is used as a baseline, where a random subset with the size of labeling budget in the training dataset is annotated. The purpose of random sampling is to simulate the performance of passive learning as a benchmark.

CRTAL [10] is used as the second reference method. Half of the labeling budget is used for the initial samples that are randomly selected. The other half of the labeling budget is used for uncertainty selection. A batch size five is used so that the least confident five samples to the current model, in each iteration, are selected for labeling and the model is updated after adding new labels to training material.

SSL [12] is coupled with random sampling and CRTAL, respectively, as the third and the fourth reference method. The annotated labels are obtained though either random sampling or CRTAL. An initial classifier is trained with annotated labels and all unlabeled segments get predicted labels based on the classification output using the initial classifier. The predicted labels and the classifier are updated with five iterations. This way of combining of CRTAL and SSL is called a serial combined learner [27].

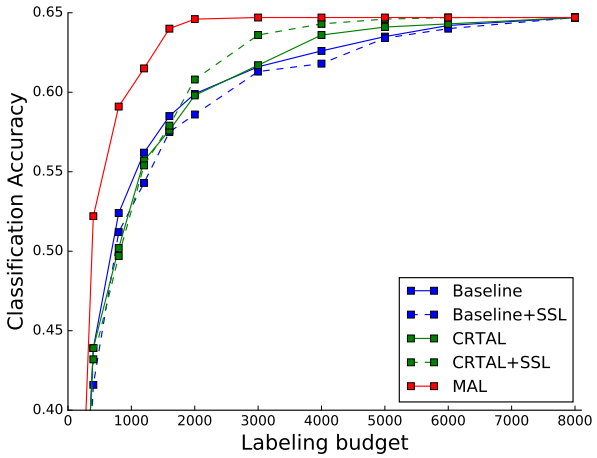

Fig. 2. Classification accuracy as a function of labeling budget, simulated using an oracle annotator.

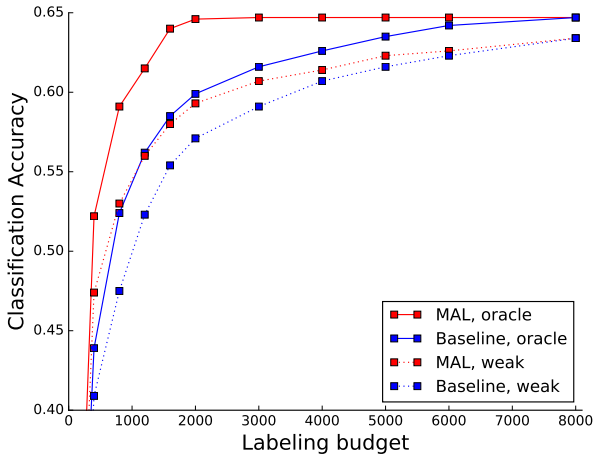

Fig. 3. Classification accuracy as a function of labeling budget, simulated using an oracle annotator (oracle) and an artificial weak annotator (weak).

3.4. Results

Figure 2 illustrates the performance of the proposed method compared with reference methods, simulating oracle labeling responses. All segments in the training set get annotated labels when the labeling budget is 8 000. When all the segments are labeled as ground truth, the obtained classifier achieves an accuracy about 65%, which is the ceiling performance of all compared methods.

The proposed method (MAL) performs the best with all simulated labeling budget until all methods converge to the ceiling performance. Reference methods need 2-4 times of labeling budget, compared to the proposed method, to achieve the same accuracy . An interesting benchmark is listening budget 2 000, where each segment has received a label, either annotated or predicted using the proposed method. We have observed that the accuracy of predicted labels is about 97%. The high labeling accuracy makes the resulted classifier approximates the ceiling performance.

CRTAL does not outperform the baseline until labeling budget of 3 000. An active learning study on speech recognition [11] shows a similar trend. When labeling budget is small, the most uncertain segments selected within a batch are often similar to each other, which makes the selected training material more redundant than when using baseline.

The effect of SSL goes divergent along with baseline and CRTAL. The performance is improved when SSL is used together with CRTAL, but similar improvement is not observed with the baseline. Uncertain segments are labeled out with CRTAL, and there remains confident segments to predict. As a result, the label prediction accuracy is much higher when CRTAL is used compared to baseline.

Figure 3 illustrates the difference in performance between the resulted classifiers using the oracle annotator and the artificial weak annotator. The results show that the proposed method also outperforms the baseline when the weak annotator is used. However, the advantage of the proposed method is smaller compared to using the oracle annotator: the baseline needs less than double sized labeling budget to achieve the same accuracy. Intuitively, this phenomenon is due to the predicted label derivation mechanism of the proposed method. Mislabeling the medoid makes a whole cluster of segments wrongly labeled to another class, which may lead to a strong confusion between the two classes. In comparison, when the same amount of wrong labels are evenly distributed to all classes, the performance of the resulted classifier seems to be affected much less. As a summary, the proposed method might be less effective when using weak annotators.

4. CONCLUSION

We propose a novel method, medoid-based active learning (MAL), to improve sound event classification performance when labeling budget is small, compared to the number of unlabeled data.

In the evaluation using an oracle annotator, when the labeling budget was less than 10% of unlabeled data, the resulted classifier using the proposed method gave about 8% better accuracy than using the best reference method. Furthermore, as the listening budget grew, the proposed method kept to outperform reference methods. Through all the experiments, the proposed method used generally 50%-60% less labeling budget to achieve the same classification accuracy with respect to the best reference method.

In this study, the number of clusters k was set to a rather big number (only four segments per cluster in average). However, the performance of the proposed method could be potentially further improved by tuning k according to the listening budget, e.g. a smaller k for a tight budget. In preliminary experiments, the classification accuracy with a tight listening budget (400) was further improved by 5% when k was halved.

The experiment using an artificial weak annotator shows that the proposed method is less effective if the annotator gives too many wrong labels. This suggests a future study about using weak annotators. In case of very weak annotator, clustering may be used to improve the labeling accuracy (listening to all segments in a cluster and label the whole cluster using majority vote) instead of active learning, which leads to another study.

As a conclusion, the proposed method can effectively improve the sound event classification performance when the labeling budget is small. In future, datasets with different number of segments and possible classes can be studied. Furthermore, it would be helpful to evaluate the performance using realistic human annotators. At last, it would be useful to study alternative acoustic models, e.g. neural network, to compare how they work along with less accurate labels.

5. REFERENCES

[1] Karol J. Piczak, "Environmental sound classification with convolutional neural networks," in *25th IEEE International Workshop on Machine Learning for Signal Processing, (MLSP)*, 2015, pp. 1–6.

[2] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6440–6444.

[3] Antonio J. Torija and Diego P. Ruiz, "Automated classification of urban locations for environmental noise impact assessment on the basis of road-traffic content," *Expert System with Applications*, vol. 53, pp. 1–13, 2016.

[4] Buket Barkana and Burak Uzkent, "Environmental Noise Classifier Using a New Set of Feature Parameters Based on Pitch Range," *Applied Acoustics*, vol. 72, no. 11, pp. 841–848, Nov. 2011.

[5] Paul Gaunard, Corine Ginette Mubikangiey, Christophe Couvreur, and Vincent Fontaine, "Automatic classification of environmental noise events by hidden markov models," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998, pp. 3609–3612.

[6] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE Transaction on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 279–288, 2016.

[7] Sébastien Lecomte, Régis Lengellé, Cédric Richard, François Capman, and Bertrand Ravera, "Abnormal events detection using unsupervised one-class svm - application to audio surveillance and evaluation.," in *International Conference on Advanced Video and Signal-Based Surveillance*, 2011, pp. 124–129.

[8] Ha Manh Do, Weihua Sheng, and Meiqin Liu, "Human-assisted sound event recognition for home service robots," *Robotics and Biomimetics*, vol. 3, no. 1, pp. 1–12, 2016.

[9] David Cohn, Les Atlas, and Richard Ladner, "Improving generalization with active learning," *Machince Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[10] Dilek Z. Hakkani-Tür, Giuseppe Riccardi, and Allen L. Gorin, "Active learning for automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, pp. 3904–3907.

[11] Giuseppe Riccardi and Dilek Hakkani-Tür, "Active learning: theory and applications to automatic speech recognition," *IEEE Transaction on Speech and Audio Processing*, vol. 13, no. 4, pp. 504–511, 2005.

[12] Pedro J. Moreno and Shivani Agarwal, "An experimental study of em-based algorithms for semi-supervised learning in audio classification," in *2003 International Conference on Machine Learning (ICML) Workshop on the Continuum from Labeled to Unlabeled Data*, 2003.

[13] Aleksandr Diment, Toni Heittola, and Tuomas Virtanen, "Semi-supervised learning for musical instrument recognition," in *21st European Signal Processing Conference (EUSIPCO)*, 2013, pp. 1–5.

[14] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang, "Representative sampling for text classification using support vector machines," in *European Conference on Information Retrieval*, 2003, pp. 393–407.

[15] Michael I. Mandel and Dan Ellis, "Song-level features and support vector machines for music classification," in *6th International Conference on Music Information Retrieval*, 2005, pp. 594–599.

[16] Toni Heittola, Annamaria Mesaros, Dani Korpi, Antti J. Eronen, and Tuomas Virtanen, "Method for creating location-specific audio textures," *EURASIP Journal of Audio, Speech and Music Processing*, vol. 2014, no. 9, 2014.

[17] Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley, 1990.

[18] Hae-Sang Park and Chi-Hyuck Jun, "A simple and fast algorithm for k-medoids clustering," *Expert System with Application*, vol. 36, no. 2, pp. 3336–3341, 2009.

[19] Tagaram S. Madhulatha, "Comparison between k-means and k-medoids clustering algorithms," in *1st International Conference in Advances in Computing and Information Technology (ACITY)*, 2011, pp. 472–481.

[20] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *4th SIAM International Conference on Data Mining*, 2004, pp. 333–344.

[21] Dorit S. Hochbaum and David B. Shmoys, "A best possible heuristic for the k-center problem," *Mathematics of Operations Research*, vol. 10, no. 2, pp. 180–184, 1985.

[22] Justin Salamon, Christopher Jacoby, and Juan P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the ACM International Conference on Multimedia, MM*, 2014, pp. 1041–1044.

[23] Shankar Vembu and Sandra Zilles, "Interactive learning from multiple noisy labels," in *European Conference in Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, 2016, pp. 493–508.

[24] Karol J. Piczak, "ESC: dataset for environmental sound classification," in *23rd Annual ACM Conference on Multimedia Conference*, 2015, pp. 1015–1018.

[25] Selina Chu, Shrikanth S. Narayanan, and C.-C. Jay Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Transaction on Audio, Speech & Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] Malte Darnstädt, Hendrik Meutzner, and Dorothea Kolossa, "Reducing the cost of breaking audio captchas by active and semi-supervised learning," in *13th International Conference on Machine Learning and Applications*, 2014, pp. 67–73.

PUBLICATION III

Learning vocal mode classifiers from heterogeneous data sources
Zhao S.Y., T. Heittola and T. Virtanen

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017, 16–20

LEARNING VOCAL MODE CLASSIFIERS FROM HETEROGENEOUS DATA SOURCES

Zhao Shuyang, Toni Heittola, Tuomas Virtanen

Tampere University of Technology
Signal Processing Department
korkeakoulunkatu 1, Tampere 33720, Finland
shuyang.zhao@tut.fi, toni.heittola@tut.fi, tuomas.virtanen@tut.fi

ABSTRACT

This paper targets on a generalized vocal mode classifier (speech/singing) that works on audio data from an arbitrary data source. Previous studies on sound classification are commonly based on cross-validation using a single dataset, without considering training-recognition mismatch. In our study, two experimental setups are used: matched training-recognition condition and mismatched training-recognition condition. In the matched condition setup, the classification performance is evaluated using cross-validation on TUT-vocal-2016. In the mismatched setup, the performance is evaluated using seven other datasets for training and TUT-vocal-2016 for testing. The experimental results demonstrate that the classification accuracy is much lower in mismatched condition (69.6%), compared to that in matched condition (95.5%). Various feature normalization methods were tested to improve the performance in the setup of mismatched training-recognition condition. The best performance (96.8%) was obtained using the proposed subdataset-wise normalization.

Index Terms: sound classification, vocal mode, heterogeneous data sources, feature normalization

1. INTRODUCTION

In this study, we aim at a generalized vocal mode (speech/singing) classifier, working on audio data from arbitrary sources. A generalized vocal mode classifier can potentially save a lot of time when finding interesting parts in a video, along with established vocal activity detection techniques [1, 2]. As an example, the singing part from a talent show episode can be easily found on YouTube.

The captured audio is affected by the recording device, acoustic space and background noises. The acoustic space and the recording device are collectively defined as transmission channel. In practice, the training-recognition mismatch is a significant problem: a classifier often fails when working on audio data captured using a different recording setup. However, majority of previous sound classification studies are based on a single dataset using cross-validation [3, 4, 5], without considering the cases of training-recognition mismatch. We call it a *homogeneous recognition scenario*, when training and testing data are from the same recording setup. We call it a *heterogeneous recognition scenario*, when recognition data is from different recording setups compared to the training data.

In previous studies, feature normalization has been shown effective to cope with training-recognition mismatch in robust speech recognition [6, 7, 8]. Mean-variance normalization (MVN) [9] scales features in each data source to have zero-mean and unit-variance. Histogram equalization (HE) [7, 8] aims at a more sophisticated matching over the histogram from a distribution basis to a distribution target. Notably, there is a significant difference between our study and robust speech recognition. Taking the experimental framework Aurora [10] used in [7, 8] as an example, a single clean speech dataset is used for training. The background noises of different environment are added to clean data to be used as testing material, thus the main mismatch is the background noises. In our study, the training material is from a few different datasets instead of one to cover various speech and singing styles. The main mismatch between the datasets is in channel effect instead of background noise, since all the datasets are recorded in relatively silent environment.

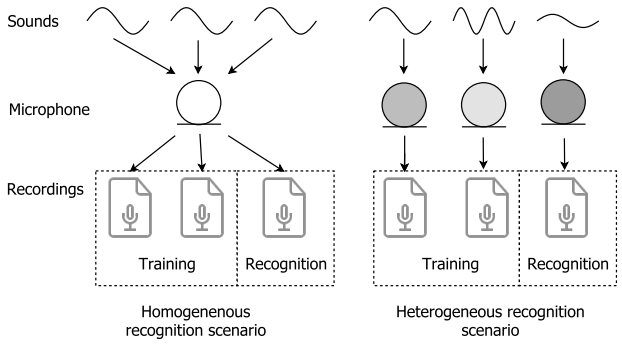

Figure 1: An example of a homogeneous recognition scenario and a heterogeneous recognition scenario.

This study deals with the training-recognition mismatch when learning vocal mode classifiers from heterogeneous data sources. Firstly, we investigate the difference in performance between homogeneous recognition scenario and heterogeneous recognition scenario. Secondly, we evaluate various feature normalization methods to improve the classification performance in heterogeneous recognition scenario. The main focus is the data scope to perform feature normalization, which is seldom investigated in previous studies. Besides the obvious recording-wise and dataset-wise normalization, subdataset-wise normalization is proposed. The normalization data scopes are evaluated along with MVN and HE. A new dataset TUT-vocal-2016 is introduced to evaluate the classification performance.

The organization of this paper is as follows. The method is described in Section 2. The used datasets are discussed in Section 3 and the experimental results for evaluation are given in Section 4. The conclusions are drawn in Section 5.

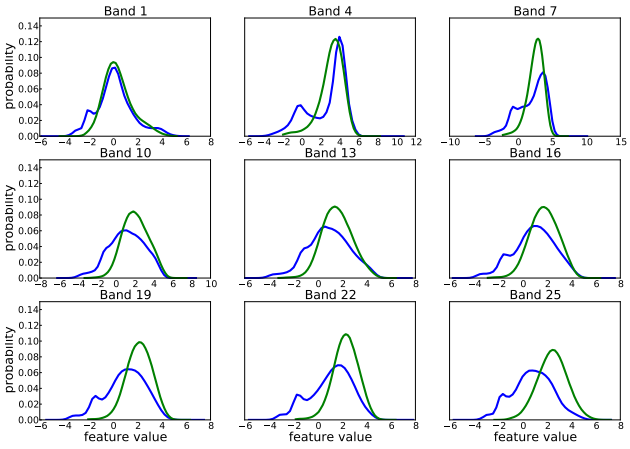

Figure 2: Feature distribution in CHiME2010 and Arctic dataset, illustrated in green and blue lines, respectively. The visualized features are log Mel-band energies from nine different bands.

2. METHOD

A vocal mode classifier takes an audio recording as input and the output is the predicted vocal mode corresponding to every second in the recording. The vocal mode classifier follows an established setup in the domain of sound classification: log-mel band energies as features and a multilayer perceptron as the classifier [11, 12]. In addition to the established setup, feature normalization is performed on the log-mel band energies.

2.1. Acoustic Features

The acoustic features are calculated as follows. The audio amplitude is normalized, scaling the maximum amplitude of a recording to one. The audio signal is divided into frames with length of 30 ms and 50% overlap. The number of Mel filter banks is 30, ranging from 25 Hz to 8000 Hz.

In order to investigate the difference in transmission channel between different data sources, the feature distributions of two speech datasets, CHiME2010 [13] and Arctic [14], are visualized in Figure 2. The histogram plots are obtained by dividing the interval [-4σ, 4σ] of each feature coefficient into 50 bins. Only features from non-silent frames are taken into account. As is shown in Figure 2, each feature coefficient in the CHiME2010 dataset is distributed around a single peak, similar to the normal distribution. In contrast, most coefficients in Arctic dataset are distributed around two peaks. Both CHiME2010 and Arctic are speech datasets containing balanced English utterances recorded in relatively silent environment, however the feature distributions are largely different, which reveals the the difference between the two datasets in terms of channel effect.

2.2. Feature normalization

A transmission channel introduces a time-invariant distortion to the original signal, under the assumption of linear system. It is assumed that there exists an invariant global distribution for voice signal, before transmitting through a channel [6]. If different recording setup is used, the global distribution becomes transformed. The global distribution using a recording setup can be estimated using available data from the source. Feature normalization aims at removing the noise and channel effect by matching the overall feature distributions of different data sources.

Two types of feature normalization techniques are considered: mean-variance normalization (MVN) [9] and quantile equalization (QE) [15]. They are simple and require not too much data from a data source to estimate the feature distribution, compared to more complicated and elaborated methods such as full histogram equalization [7], feature space rotation [16] and vocal tract length normalization [17].

In practice, it is quite often unknown what recordings are from the same recording setup. The audio data inside a recording is surely homogeneous, however the amount of data in a single piece of recording may not be sufficient to estimate the feature distribution of the source. Another solution is dataset-wise normalization, based on the assumption that the audio in the same dataset is recorded under very similar condition. However, this is not always a valid assumption. As an example, some audio datasets are collected in parallel using different recording devices in different environment. We use a term *data scope*, within which the feature distribution is estimated and feature normalization is performed. Global normalization, as a reference, scales all the data the same way, based on the statistics of the whole training material.

In addition, we propose another approach, where datasets are decomposed into sub-datasets based on K-means clustering on recordings. The number of clusters is defined proportionally to the data amount, with two hours of non-silent material in the dataset corresponding to one cluster.

Overall, we consider two feature normalization techniques and three normalization data scopes. In mean-variance normalization, a feature vector $\mathbf{x}$ in a data scope $\mathbf{X}$ is normalized as

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - \mu}{\sigma}, \tag{1}$$

where μ and σ is the mean and standard deviation within the data scope $\mathbf{X}$.

Quantile equalization estimates a transformation function for each feature coefficient based on the quantile statistics of the data scope as basis and the whole training set as target. Five critical values: minimum, 25th-percentile, median, 75-percentile and the maximum are used to divide the range of a feature coefficient into four bins. The value of kth critical value for ith coefficient is denoted as Q_k^i and $\hat{Q}_k^i$, respectively for the basis and target distribution. A feature coefficient x^i is normalized as

$$x_{norm}^i = \hat{Q}_k^i + (x^i - Q_k^i)\frac{\hat{Q}_{k+1}^i - \hat{Q}_k^i}{Q_{k+1}^i - Q_k^i} \tag{2}$$
$$\forall x^i \in Q_k^i < x < Q_{k+1}^i.$$

2.3. Supervised learning

Multilayer perceptron (MLP) [18] is a basic type of artificial neural network, consisting of layers of nodes with each layer fully connected to the next one. Feature vectors are given to the network as input and the output corresponds to target classes. The implementation is based on Keras [19] using Theano [20] as backend.

Let us denote the node values of input layer as $\mathbf{h}^1 = \mathbf{x}$ and the node values of kth layer as $\mathbf{h}^k$. Given the node values of $k-1$th

Name	Class	Duration	Ref
CHiME	Speech	7h 06m	[13]
Arctic	Speech	6h 27m	[14]
CHAINS	Speech	2h 19m	[21]
Multitrack2013	Sing	17h 10m	[22]
Marl	Sing	1h 51m	[23]
Tonas Flamenco	Sing	0h 13m	[24]
TUT-VOX	Sing	0h 48m	-
TUT-vocal-2016	Both	3h 15m	-

Table 1: Datasets used in our experiments. Used length is the non-silent part of used recordings in a dataset. The duration is reported excluding silence.

layer, the node values of kth layer are calculated as

$$\mathbf{g}^{k-1} = \mathbf{W}^k \mathbf{h}^{k-1} + \mathbf{b}^k, 2 \leq k < M \quad (3)$$

$$\mathbf{h}^k = \mathcal{F}(\mathbf{g}^{k-1}). \quad (4)$$

Eq. (3) shows the linear transformation operation on $k-1$th layer of the neural network, where $\mathbf{W}^k \in \mathbb{R}^{S_{k-1} \times S_k}$ is a weight matrix between layer $k-1$ and layer k. S_k is number of neurons in layer k. A bias vector is denoted as $\mathbf{b}^k$. A non-linear activation function $\mathcal{F}$ is applied element-wise on the linear transformation outputs. The total number of hidden layers is two. Sigmoid function is used as activation function for hidden layers and the output layer ($\mathbf{h}^4 = \mathbf{y}$).

Context windowing is used for the neural network input: the consecutive feature frames $[\mathbf{x}[t-R], ..., \mathbf{x}[t], ..., \mathbf{x}[t+R]]$ are stacked together to form a single feature vector $\mathbf{x_c}[t]$ to represent temporal dynamics, where R is number of the past and future frames. We use $N_{cw} = 2R + 1 = 25$ to denote the total number of frames used in a context window. In order to smooth the neural network output, mean filter is used for neural network output as $[\mathbf{y}[t-L], ..., \mathbf{y}[t], ..., \mathbf{y}[t+L]]$. The size of the mean filter is $N_{mf} = 2L + 1 = 35$.

3. DATASETS

There is not any public dataset designed for speech/singing classification. However, there are many speech datasets designed for speech recognition and several singing datasets designed for music research. Three speech datasets and four singing datasets are selected as training material based on the variability and accessibility. The list of used datasets is shown in Table 1. In addition, we collected a new dataset TUT-vocal-2016 that contains both speech and singing to evaluate trained classifiers.

3.1. Datasets for training

All the speech datasets contain English speech from both male and female. CHiME dataset [13] contains speech utterances from 34 speakers with reverberation. Arctic dataset [14] is a clean speech dataset designed for speech synthesis and speech recognition, contributed by 7 speakers. CHAINS dataset is contributed by 36 speakers, including normal, fast and whispered speech. Only the normal speech and whispered speech utterances are used in this study.

Four singing databases are used. Multitrack2013 covers singing styles of pop and pop rock [22]. Tonas Flamenco [24] contains only Flamenco singing. The Marl dataset [23] contains pop singing and rap. Recordings containing rapping have been excluded in our experiments, since it is ambiguous if rapping belongs to speech or singing. TUT-VOX is a proprietary dataset containing acappella singing in English and Finnish.

3.2. TUT-vocal-2016

In order to make a proper evaluation for vocal mode classification, we introduce a new dataset TUT-vocal-2016. The core idea is to have audio where the same person is speaking and singing, preferably the same language content. The dataset is contributed by 20 volunteers, 10 females and 10 males. Each volunteer is required to choose four songs. The volunteer is required to sing from one to one half minutes of each song, thus all recordings weigh similarly in the evaluation. There are 80 pieces of singing collected, from a set of 21 different songs. The lyric of the songs is read out by each volunteer in three types: normal speech, whispered speech and shouted speech. The shouted speech is not used in this study since we have found very little shouted speech as training material.

3.3. Annotation

We use frame-level voice activity annotation, by which the silent parts in recordings are excluded for both training and testing. The frame-level activity annotation was obtained using two automatic approaches. The principle is to exclude all the silent frames in the evaluation and a small part of voices annotated as silence is tolerated.

Speech utterances were mostly short and contained usually only silence at the beginning and at the end of the signal. A simple energy-based scheme was chosen for this type of signals. In the scheme, 10% of the average RMS-energy was used as threshold to detected non-silence (active) segments. This scheme worked best with signals having mostly active segments and most of the energy is also concentrated in these vocal segments.

The acappella singing contains longer silent segments and in some cases added effects like reverberation making it hard to use such a simple threshold. For these type of signals, a binary classifier based approach was used [25]. In this approach, 10% of lowest energy frames within a recordings are used to train Off-class and 10% of highest energy frames is used to train On-class. The classifier was used to get probability of frame belonging to the On-class (active). Classification was done by defining the probability threshold as weighted mean between top 10% and bottom 10% of collected probabilities. After the binary classification, short segments under 200 ms were omitted from output.

4. EVALUATION

Firstly, we evaluate the difference in classification performance between homoogeneous recognition scenario and heterogeneous recognition scenario. Secondly, we try to find the best feature normalization method and data scope in heterogeneous recognition scenario.

4.1. Setup

To evaluate the classification performances in the homogeneous recognition scenario, we perform a 4-fold cross validation on the TUT-vocal-2016 dataset. The evaluation results are reported averaging the four folds.

Scenario	Normalization data scope		MVN	QE
	Training	Testing		
Heterogeneous	Global	Global	69.6	
Homogeneous	Global	Global	95.5	
Heterogeneous	Recording	Recording	72.7	76.2
	Dataset	Dataset	88.1	91.6
	Subdataset	Subdataset	96.8	93.9
	Subdataset	Dataset	90.7	90.4
	Dataset	Recording	81.1	78.3
	Subdataset	Recording	81.2	81.1

Table 2: Evaluation on different data scopes using mean-variance feature normalization (MVN) and quantile equalization (QE).

In the evaluation of the heterogeneous recognition scenario, TUT-vocal-2016 dataset is used for testing, while the rest of the datasets are used for training. The baseline is global feature normalization, where all the feature vectors in training and testing material are operated with the same linear transformation based on the statistics of training material alone. Two feature normalization techniques, MVN and QE are evaluated, along with three feature normalization data scopes, recording-wise, dataset-wise, subdataset-wise. Particularly, we evaluate recording-wise normalization for the testing data, while using all the three normalization data scopes for training data. In many practical cases, the data source is unknown at the recognition stage, or the statistics of the whole recognition dataset are not available.

4.2. Results

The experimental results are reported in unweighted accuracy (average recall), of the two classes. The experimental results are shown in Table 2. MVN and QE give similar performance through all the experiments. In contrast, the feature normalization data scope significantly affects the classification performances. Based on that, we can simply use the results from MVN to discuss different normalization data scopes.

The obtained accuracy using subdataset-wise normalzation was remarkably high. We investigated the clustering results of the TUT-vocal-2016 dataset and found that the speech and singing recordings were clustered to different subdatasets. All of our training datasets consist either speech or singing. The condition is more matched, when training and testing data scope contains only just one class, which leads to a big improvement when the testing dataset is normalized subdataset-wise. When online application is considered (normalization scope is recording-wise at recognition), there is no major difference in performance between dataset-wise and subdataset-wise normalization.

In most robust speech recognition studies [6, 8], QE gives clearly better performance than MVN. However, this conclusion does not hold in our study. In the robust speech recognition studies, the purpose of feature normalization is to improve the noise robustness. In comparison, all the datasets used in our study are recorded in close microphone scenario, thus relatively clean from interfering sounds. Our experimental results suggests that it has no benefit using QE compared to MVN, when the mismatch is mainly on channel effect.

5. CONCLUSION

This paper targets on a generalized vocal mode classifier, which is able to perform classification on signals from arbitrary data sources. A new dataset TUT-vocal-2016, containing both speech and singing from 20 volunteers, was collected for evaluation.

In a homogeneous recognition scenario, a four fold cross-validation is made on TUT-vocal-2016 alone. In a heterogeneous recognition scenario, four speech datasets and three singing datasets are used as training material, and TUT-vocal-2016 is used for testing. In the experiments, the vocal mode classifiers were based on log-Mel band energies as features and multi-layer perceptrons as models. The experimental results showed that the classifier gave clearly higher accuracy 95.5% in the homogeneous recognition scenario compared to heterogeneous recognition scenario (69.6%).

This result shows that the classification performance is severely degraded by training-recognition mismatch. However, we found no public evaluation setup for sound classification targeting on heterogeneous recognition scenario. A new evaluation setup should be established to test the capability of classifiers to work on heterogeneous data sources.

Various feature normalization methods were tested to improve the classification performances in the heterogeneous recognition scenario. Subdataset-wise mean-variance normalization was found to gave the best performance, which achieved a classification accuracy of 96.8%. However, the subdataset-wise normalization relies on the knowledge of recognition data source and a sufficient amount of data from the source is needed to estimate the feature distribution. In case that the feature distribution can only be estimated based on the current signal to be recognized, the best achieved accuracy was 81.2 %.

This suggested that an online application would be much more challenging than an offline application for a heterogeneous recognition scenario. In the future, studies should be made on normalization methods that requires less data to improve on heterogeneous recognition scenario.

6. REFERENCES

[1] T. Drugman, Y. Stylianou, Y. Kida, and M. Akamine, "Voice activity detection: Merging source and filter-based information," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 252–256, 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2015.2495219

[2] F. G. Germain, D. L. Sun, and G. J. Mysore, "Speaker and noise independent voice activity detection," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, 2013, pp. 732–736. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2013/i13_0732.html

[3] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 1041–1044. [Online]. Available: http://doi.acm.org/10.1145/2647868.2655045

[4] K. J. Piczak, "ESC: dataset for environmental sound classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15*, 2015, pp.

1015–1018. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806390

[5] T. Virtanen, A. Mesaros, T. Heittola, M. Plumbley, P. Foster, E. Benetos, and M. Lagrange, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 2016.

[6] S. Molau, F. Hilger, and H. Ney, "Feature space normalization in adverse acoustic conditions," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03*, 2003, pp. 656–659. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2003.1198866

[7] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust large vocabulary speech recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 14, no. 3, pp. 845–854, 2006. [Online]. Available: http://dx.doi.org/10.1109/TSA.2005.857792

[8] Á. de la Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005. [Online]. Available: http://dx.doi.org/10.1109/TSA.2005.845805

[9] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1-3, pp. 133–147, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0167-6393(98)00033-8

[10] D. Pearce and H. Hirsch, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH*, 2000, pp. 29–32. [Online]. Available: http://www.isca-speech.org/archive/icslp_2000/i00_4029.html

[11] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks, IJCNN*, 2015, pp. 1–7.

[12] J. Li, W. Dai, F. Metze, S. Qu, and S. Das, "Comparison of deep learning methods for environmental sound detection," in *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '17*, 2017, pp. 126–130.

[13] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. D. Green, "The PASCAL chime speech separation and recognition challenge," *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.csl.2012.10.004

[14] J. Kominek, A. W. Black, and V. Ver, "CMU arctic databases for speech synthesis," Carnegie Melon University, Tech. Rep., 2003.

[15] F. Hilger, S. Molau, and H. Ney, "Quantile based histogram equalization for online applications," in *7th International Conference on Spoken Language Processing, ICSLP2002*, 2002. [Online]. Available: http://www.isca-speech.org/archive/icslp_2002/i02_0237.html

[16] S. Molau, F. Hilger, D. Keysers, and H. Ney, "Enhanced histogram normalization in the acoustic feature space," in *7th International Conference on Spoken Language Processing, ICSLP2002*, 2002. [Online]. Available: http://www.isca-speech.org/archive/icslp_2002/i02_1421.html

[17] R. Hariharan and O. Viikki, "On combining vocal tract length normalisation and speaker adaptation for noise robust speech recognition," in *Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999*, 1999. [Online]. Available: http://www.isca-speech.org/archive/eurospeech_1999/e99_0215.html

[18] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[19] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.

[20] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[21] M. Grimaldi and F. Cummins, "Speaker identification using instantaneous frequencies," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 6, pp. 1097–1111, 2008.

[22] "Karaokeversion," www.karaoke-version.com, accessed: 11.03.2016.

[23] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "MedleyDB: A multitrack dataset for annotation-intensive mir research," in *15th International Society for Music Information Retrieval Conference*, 2014.

[24] J. Mora, F. Gomez, E. Gomez, F. Escobar-Borrego, and M. Diaz-Banez, "Melodic characterization and similarity in a cappella flamenco cantes," in *11th International Society for Music Information Retrieval Conference*, 2010.

[25] T. Giannakopoulo, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS ONE*, 2015.

PUBLICATION IV

An Active Learning Method Using Clustering and Committee-Based Sample Selection for Sound Event Classification

Zhao S.Y., T. Heittola and T. Virtanen

16th International Workshop on Acoustic Signal Enhancement (IWAENC), 2018, 116–120

AN ACTIVE LEARNING METHOD USING CLUSTERING AND COMMITTEE-BASED SAMPLE SELECTION FOR SOUND EVENT CLASSIFICATION

Zhao Shuyang, Toni Heittola, Tuomas Virtanen

Tampere University of Technology

ABSTRACT

This paper proposes an active learning method to control a labeling process for efficient annotation of acoustic training material, which is used for training sound event classifiers. The proposed method performs K-medoids clustering over an initially unlabeled dataset, and medoids as local representatives, are presented to an annotator for manual annotation. The annotated label on a medoid propagates to other samples in its cluster for label prediction. After annotating the medoids, the annotation continues to the unexamined sounds with mismatched prediction results from two classifiers, a nearest-neighbor classifier and a model-based classifier, both trained with annotated data. The annotation on the segments with mismatched predictions are ordered by the distance to the nearest annotated sample, farthest first. The evaluation is made on a public environmental sound dataset. The labels obtained through a labeling process controlled by the proposed method are used to train a classifier, using supervised learning. Only 20% of the data needs to be manually annotated with the proposed method, to achieve the accuracy with all the data annotated. In addition, the proposed method clearly outperforms other active learning algorithms proposed for sound event classification through all the experiments, simulating varying fraction of data that is manually labeled.

Index Terms: active learning, K-medoids clustering, committee-based sample selection, sound event classification

1. INTRODUCTION

Sound event classification [1, 2] has many applications such as environmental noise monitoring [3], road surveillance [4] and remote health care [5]. Nowadays, the majority of sound event classification systems [6, 7] are based on supervised learning, which depends on annotated recordings as training material. Preparing the training material is commonly the most time-consuming part in developing a sound event classifier and annotating audio typically costs much more time than recording it. Similar situation has been faced in other applications such as speech recognition [8] and recommendation systems [9], where unlabeled data is abundant but manual labels are expensive to obtain.

The maximum number of labels can be manually assigned is commonly called a *labeling budget*. In order to optimize the classification accuracy with a limited labeling budget, three techniques have been established, including transfer learning [10], active learning [11, 12] and semi-supervised learning [13]. Transfer learning utilizes an audio representation learned from other tasks, where more labeled data is available. Active learning controls which samples will be annotated in order to efficiently utilize the labeling budget. Semi-supervised learning predicts labels for unlabeled data and use them as training material. The three techniques are not mutual exclusive, and can be combined. There are two previous active learning studies on sound event classification, semi-supervised active learning (SSAL) [11] and medoid-based active learning (MAL)[12]. Both of them involves a sample selection mechanism to control the labeling process, and a label prediction mechanism for unlabeled data.

SSAL performs sample selection and label prediction based on a classifier trained with previously labeled data. Samples with low classification confidence are selected for annotation, whereas samples with high confidence, are assigned with the predicted labels. The classifier relies on a decent amount of annotated data to achieve reliable label prediction and confidence estimation. Thus it can hardly optimize a labeling process at the very early stage when little annotated data is available. A solution to this drawback is to utilize the similarities between data points, which rely on no annotation.

MAL completely relies on the similarities between unlabeled data points. It structures unlabeled data into small clusters using K-medoids clustering. Each medoid, as a local representative, is selected for annotation. The label of an annotated medoid is propagated to the whole cluster. After all the medoids are annotated, MAL repeats the whole process on the data that has not been annotated, clustering the data again and presenting the medoids for annotation. However, repeating the process does not utilize previously annotated data, which is important for optimizing the labeling process, after a decent amount of annotated labels are collected.

In this study, we propose an active learning method that targets on optimizing the whole labeling process, utilizing both the similarities between data points and data annotated previously in the labeling process. The proposed method performs clustering and presents medoids to an annotator similarly to MAL. After annotating all the medoids, the annotation continues to the samples with mismatched prediction results from two classifiers: a nearest-neighbor classifier and a model-based classifier, both trained with annotated data. A segment with mismatched predictions is ranked by the distance to its nearest annotated sample, farthest first. In each iteration, a batch of top ranked samples are selected for annotation, and the rest of the samples update their predicted labels to the labels of their nearest annotated samples.

The structure of the paper is as follows. The problem of optimizing a labeling process is described in Section 2. The proposed active leaning algorithm is introduced in Section 3. The evaluation of the proposed system is presented in Section 4. The conclusion is drawn in Section 5.

2. PROBLEM STATEMENT

We state the problem of optimizing the process of labeling acoustic training material. A set of N sound segments $\mathcal{S} = \{s_1, s_2, ...s_N\}$ is given, initially unlabeled. A set of M sound event classes $\mathcal{C} =$

Funded by European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND and 737472 SMARTSOUND.

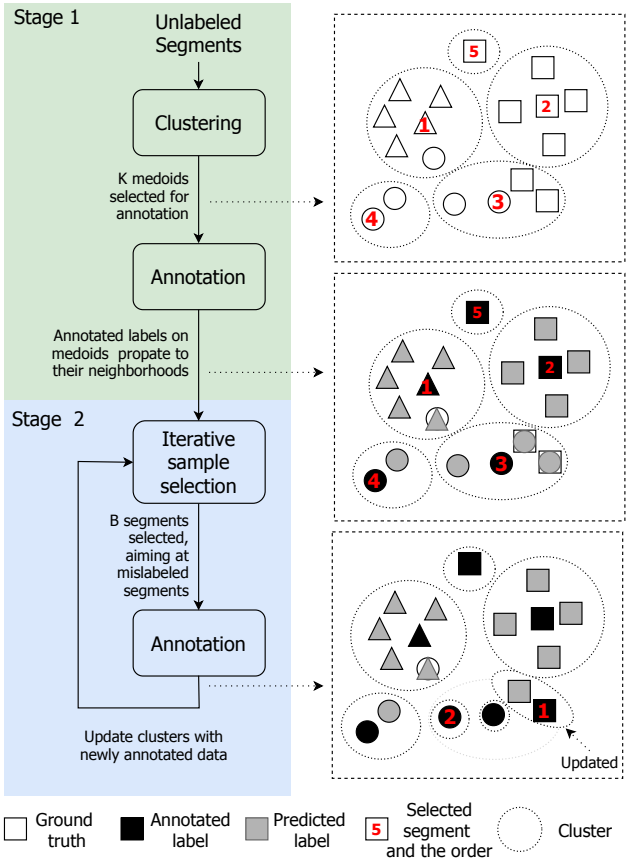

Fig. 1. Illustration of the labeling process, controlled by the proposed method. Each segment is represented with a geometric drawing and the shape represents the class.

$\{c_1, c_2, ...c_M\}$ is pre-defined. A label $l = (s, c) \in \mathcal{S} \times \mathcal{C}$ associates a segment s with a class c.

In a labeling process, an annotator examines sound segments and assigns labels. A label $l = (s, c)$ is added to a label set $\mathcal{L} \subset \mathcal{S} \times \mathcal{C}$, by associating a segment s to a class c. The segments that are manually examined and annotated are called *annotated segments*, denoted as $\mathcal{A}$. The segments that are not examined are called, *unexamined segments*, and denoted as $\mathcal{U} = \mathcal{S} \setminus \mathcal{A}$.

A labeling process produces a label set $\mathcal{L}$, including annotated labels ($\mathcal{L}_\mathcal{A}$) on $\mathcal{A}$ and possibly machine-generated predicted labels ($\mathcal{L}_\mathcal{U}$) on $\mathcal{U}$. The produced label set is used to train a supervised classifier. The problem is to optimize the labeling process that the obtained label set results in the most accurate classifier, under a labeling budget.

3. THE PROPOSED METHOD

The proposed method is illustrated in Figure 1. The input is a set of segments $\mathcal{S}$, initially unlabeled. Sound segments are typically sliced from audio recordings.A set of labels $\mathcal{L}$ is produced through a labeling process, controlled by the proposed method. The labeling process ends when all the segments are annotated or the labeling budget runs out. After the labeling process, $\mathcal{L}$ are used for supervised learning.

The proposed method has two stages. In the first stage, K-medoids clustering is performed and the medoids, as local representatives, are presented to an annotator for manual annotation. An annotated label propagates to the whole cluster as predicted labels. By the end of the first stage, each segment gets a label, either annotated or predicted. In the second stage, a batch of B samples are selected for annotation in each iteration. The selection is based on the prediction mismatch between two classifiers: nearest-neighbor prediction based on $\mathcal{A}$ and a model-based classifier trained with $\mathcal{A}$. The segments are further ranked by the distance to the nearest annotated segment. In the second stage, the clusters are updated, using $\mathcal{A}$ as cluster centroids and assigning each unexamined segment to its nearest annotated segment.

3.1. Distance matrix

The proposed method relies on a distance metric relevant to the target classification problem. The distances between segments under the same class should be generally smaller, compared to segments under different classes. We compute a distance matrix consisting of pairwise distances between all the sample.

Mel-frequency cepstral coefficients (MFCCs), its first-order and second-order derivatives are used as acoustic features. The MFCCs within a sound segment is represented by a multi-variate Gaussian distribution, based on the mean and the variance. Symmetric Kullback–Leibler (KL) divergence is used to measure the dissimilarity between a segment pair. The measured dissimilarity between two segments x and y is called distance for simplicity, and denoted as $d(x, y)$, though KL divergence is not distance. The distance from a segment to itself is zero and the distance matrix $D^{N\times N}$ is symmetric with diagonal values being zero.

The MFCCs-Gaussian-KL as a similarity measurement has been widely used in acoustic information retrieval [15, 16]. Besides MFCC-Gaussian-KL as a static programmed similarity measurement, there are studies on machine-learned metrics, which outperformed static programmed similarity metrics in problems such as content-based music recommendation [17] and sound event query by voice-imitated examples [18]. However, this does not suit the targeted situation, since a learned metric itself requires labeled data to train.

3.2. Stage one: Clusters with representatives

K-medoids clustering is performed based on the distance matrix. The clustering algorithm finds a set of K medoids $\mathcal{M} = \{m_1, m_2, ..m_k\}$, that minimizes the total distance from each segment to its nearest medoid, as $\sum_{x\in\mathcal{S}} min\{d(x, y)|y \in \mathcal{M}\}$. This can be interpreted that $\mathcal{M}$ is an optimized set of segments to make nearest-neighbor prediction to the rest. Thus, medoids are presented to the annotator for labeling. The annotated label assigned to a medoid propagates to the whole cluster as predicted labels. The label propagation is equivalent to nearest neighbor prediction based on $\mathcal{M}$.

3.2.1. K-medoids clustering

K-medoids [19, 20] is a partitioning-based clustering algorithm, similar to more widely-used K-means. The main difference is that K-medoids uses real data point as centroids, whereas in K-means, a cluster centers at an arbitrary data point.

The initialization of medoids is based on farthest-first traversal [21]: a traversed set starts as a singleton of a random segment and the farthest segment to the current traversed set (the distance from a

point x to a set $\mathcal{S}$ is defined as $d(x, \mathcal{S}) = min\{d(x, y)|y \in \mathcal{S}\})$ is iteratively added to the traversed set. Farthest-first traversal has been proved to give an efficient approximation of k-center problem [22].

3.2.2. Choosing the number of clusters

We analyse the number of clusters K inversely, using a factor $KI = \frac{N}{K}$, where KI can be interpreted as the average cluster size. KI controls the trade-off between quantity and accuracy of generated predicted labels. In the previous MAL study [12], KI has been fixed to four, based on a preliminary experiment on a small scale dataset. However, the best choice of KI varies along with each dataset.

We propose a median neighborhood test method to determine KI, estimating the largest cluster size that an annotated label can reliably propagate to. The test needs to manually annotate a small number of segments. Firstly, we choose a pivotal segment p, the segment that has the median distance to its nearest neighbor among $\mathcal{S}$, targeting on a segment with average neighborhood density. A counter i is initially set to one. The algorithm queries the label for the ith nearest neighbor of p. The counter increments if the label of the ith nearest neighbor matches with p. Otherwise, we settle with $KI = i$ and runs K-medoids clustering with it. In case KI ends up to be one, the method will be equivalent to random sampling. This happens when the distance metric is highly irrelevant to the target classification problem.

3.3. Stage two: Mismatch-first farthest-search

The sample selection in the second stage is iterative. In each iteration, a batch of B samples are selected for annotation, denoted as $\mathcal{B}$. The selection is based on mismatch-first farthest-search, targeting on segments with wrong predicted labels.

Our first sample selection criteria is based on committee-based sample selection [14]. The principle is to select samples with mismatched prediction results from different types of classifiers, trained with the same material, as a decision committee. It is based on two assumptions. The first one is that a classifier is more likely to be wrong when another type of classifier makes a mismatched prediction, compared to the case that the whole committee agrees on the prediction. The other assumption is that a classifier benefits more from a counter example, where the classifier makes mistakes, than an example where the classifier succeeds. Every selected sample is a counter example to at least one classifier in the committee, thus the committee as a whole efficiently improves with the selected samples.

The proposed method intrinsically involves two types of classifiers: the nearest-neighbor classifier for label prediction and the model-based classifier trained after the labeling budget runs out. The model-based classifier is trained with $\mathcal{L}_\mathcal{A}$ and the prediction results on $\mathcal{U}$ are compared with the $\mathcal{L}_\mathcal{U}$. The prediction mismatch between the two classifiers is the first criteria in the sample selection.

There are typically multiple unexamined segments with mismatched predictions. The second criteria is the distance of label propagation, assuming that the label propagating the largest distance is most likely to be wrong. Thus, the segments with mismatched predictions are further ranked by the distance to its nearest annotated segment. Practically, the segments with mismatched prediction are added to $\mathcal{B}$ based on farthest-first traversal, as is defined in Section 3.2.1, adding the sample that has the farthest distance to $\mathcal{A} \cap \mathcal{B}$ to $\mathcal{B}$ until $\mathcal{B}$ reaches the size of B. In case that less than B segments have mismatched predictions, farthest-first traversal continues to segments of matched predictions.

After annotating a batch of segments, the predicted label of each unexamined segment is updated based on its nearest annotated segment. This is equivalent to replacing $\mathcal{M}$ by $\mathcal{A}$ as medoids and updating the partition in K-medoids clustering. Since $\mathcal{M} \subseteq \mathcal{A}$ in the second stage, the sizes of updated clusters are equal or smaller compared to the first stage.

4. EVALUATION

In order to evaluate an active learning algorithm, we use the obtained labels to train a supervised classifier, with which the classification accuracy on a test dataset is used for evaluation. The labels obtained with different active learning algorithms vary in terms of quantity and accuracy, thus the resulted classifier is used for evaluation.

4.1. Dataset

Previous study on MAL used UrbanSound8K [24] for evaluation. We use the same dataset in this study for consistency. UrbanSound8K is a public environmental sound dataset, based on real field-recordings. The dataset includes 8 732 manually annotated sound segments with maximum duration of 4 seconds, totalling 8.75 hours. The dataset includes 10 sound event classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. The dataset provides a 10-folds division for cross validation.

4.2. Experimental setup

The experimental setup also follows the previous MAL study. An active learning algorithm output training material that requires a supervised classifiers to evaluate. Since the purpose of the evaluation is not to find the best model, we simply use a support vector machine (SVM) classifier, the baseline classifier of the UrbanSound8K dataset with radial basis function as kernel.

The acoustic feature extraction in the supervised learning also follows the baseline in UrbanSound8K, using the following summary statistics of MFCCs in each segment: minimum, maximum, median, mean, variance, skewness, kurtosis and the median and variance of the first and second derivatives. MFCCs used in the similarity measurement and supervised learning are the same. The audio signal is divided into frames with 24 ms length and 50% frame overlap. We compute 1st to 25th MFCCs from 40 Mel bands between 25 Hz and 22 050 Hz.

In each round of evaluation, nine folds are used for training and one fold is used for testing. The labels provided by the dataset are used as ground truth. In a training set, the ground truth labels are initially all hidden. Annotating a sound segment consumes the labeling budget by one. The annotated labels are always simulated with the ground truth.

Unweighted accuracy is used to evaluate the performance. It weighs different classes the same, regardless to the number of instances. The classification accuracy is reported averaging the accuracy across all 10 folds. Due to the random elements, medoid initialization and random sampling, in the experiments, all the experiments are repeated three times and the averaged results are reported.

4.3. Reference methods

Random sampling is commonly used as a baseline in active learning studies [11, 12]. It presents the data to the annotator in a random permutation.

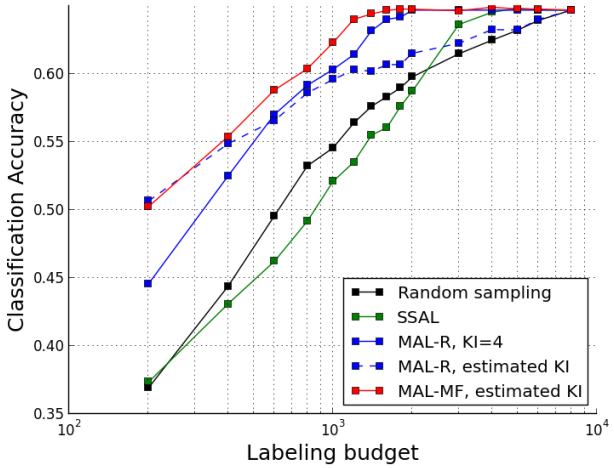

Fig. 2. Classification accuracy as a function of labeling budget. The proposed method, MAL-MF, is evaluated with SSAL [11], MAL-R [12] and random sampling as reference methods.

SSAL [11] is used as the second reference method. In the first stage, 200 samples are randomly selected. In the second stage, the sample selection is iterative. In each iteration, the annotated labels are used to train a classifier. In each iteration, the least confident 50 samples to the classifier are selected for annotation. When the labeling process ends, unexamined segments get predicted labels from the classifier, and all the obtained labels are used to train a final classifier. Originally in the SSAL study, it has a maximum confidence threshold for sample selection and samples are randomly selected under the threshold. In addition, it has a minimum confidence threshold for label prediction. Our reference method does not use these two thresholds, since there is not an established rule to set them.

Previous MAL [12], named here MAL-recursive (MAL-R), has the similar procedure as the first stage of the proposed method, with fixed $KI = 4$. It runs a recursive process, repeating the first stage process on unexamined segments, after all the medoids are annotated. We firstly evaluate MAL-R with $KI = 4$, as it has been originally proposed. Additionally, we evaluate MAL-R with the KI estimated using the proposed median neighborhood test.

The proposed method, medoid-based active learning with mismatch-first farthest-search (MAL-MF) uses median neighborhood test to determine KI. The batch size in the second stage is set to 50, the same as the experimental setup on SSAL.

4.4. Results

Figure 2 illustrates the performance of the proposed method (MAL-MF), compared to MAL-R, SSAL and random sampling. All segments in the training set get annotated labels when the labeling budget is 8 000. When all the segments are labeled as ground truth, the obtained classifier achieves an accuracy about 64.7%, which is the ceiling performance of all compared methods. Experimentally in some cases, a few errors in predicted labels result in a classifier with higher accuracy. As a result, some results in the illustration may be slightly higher than the ceiling performance. We call a result to approximate the ceiling performance when the difference in accuracy is lower than 0.5%.

The result shows that the proposed method outperforms all the reference methods through the experiments. The proposed method requires only 20% of the training data to be manually annotated to approximate the ceiling performance. In comparison, SSAL outperforms baseline only when the labeling budget is more than 25% of the training data. The main reason is that the labels predicted with SVM are much less accurate than the labels propagated from the local representatives, when the labeling budget is low.

The proposed method and MAL-R shares the same process in the first stage. The proposed method uses KI estimated separately for each fold. Based on the proposed median neighborhood test, the choice of KI ranges in $[4, 16]$ across the ten folds, with the median of 12. When MAL-R uses fixed $KI = 4$ as previously proposed, the cluster size is relatively small,thus the purity of the clusters is more than 97%. It approximates the ceiling performance by annotating all the medoids, using 25% of unlabeled data as labeling budget. The proposed method, considering the median case $KI = 12$, produces labels three times fast as $KI = 4$, with the purity of clusters dropping to 85%. The higher number of obtained labels allows better performance on small labeling budget. The second stage process allows the proposed method to effectively correct the errors in predicted labels. As a result, the proposed method approximates the ceiling performance using only 20% of unlabeled data as labeling budget. When MAL-R uses the same KI estimated with the proposed median neighborhood test, it has the same performance to MAL-MF with low labeling budget, however the accuracy of MAL-R increases slowly as labeling budget grows, due to its non-optimal second stage.

In order to analyse the sample selection performance in the second stage, we observed the label prediction error rate in unexamined segments, unexamined segments with mismatched predictions and selected segments. From the beginning of the second stage to where the performance approximates the ceiling, the prediction error rate of segments with mismatched predictions is typically 1.5 times to the error rate in all unexamined segments. The selected segments, the segments with mismatched prediction and ranking top 50 by the distance to the nearest annotated segment, has 3-10 times label prediction error rate, compared to error rate in all unexamined segments. Typically the ratio grows from three to ten along with the labeling process.

5. CONCLUSIONS

This study proposes an active learning algorithm to control the labeling process on sound event data, to save the annotation effort to prepare training material. The proposed method has two stages. In the first stage, K-medoids clustering is performed on an unlabeled dataset and the medoids are selected for annotation. The annotated label on a medoid propagates to its cluster. In the second stage, the selection is based on mismatch-first farthest-search, an extension and committee-based sample selection. The predicted labels are updated using nearest-neighbor prediction, based on the annotated data.

The evaluation is based on the classification accuracy on a test dataset, using a support vector machine classifier, trained based on labels obtained in the active learning process. The results show that only 20% of the data needs to be manually annotated with the proposed method, to achieve the performance with all the data annotated. Furthermore, it clearly outperforms all the reference method, SSAL and MAL-R, through all the experiments.

In the future, the proposed method can be tried to save labeling budget to classify other media type, if there is a exists a similarity metric that gives decent retrieval performance.

6. REFERENCES

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, Feb 2018.

[2] Karol J. Piczak, "ESC: dataset for environmental sound classification," in *23rd Annual ACM Conference on Multimedia Conference*, 2015, pp. 1015–1018.

[3] Panu Maijala, Zhao Shuyang, Toni Heittola, and Tuomas Virtanen, "Environmental noise monitoring using source classification in sensors," *Applied Acoustics*, vol. 129, no. 6, pp. 258267, January 2018.

[4] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE Transaction on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 279–288, 2016.

[5] Ha Manh Do, Weihua Sheng, and Meiqin Liu, "Human-assisted sound event recognition for home service robots," *Robotics and Biomimetics*, vol. 3, no. 1, pp. 7, Jun 2016.

[6] Emre Çakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[7] Juncheng Li, Wei Dai, Florian Metze, Shuhui Qu, and Samarjit Das, "Comparison of deep learning methods for environmental sound detection," in *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '17*, 2017, pp. 126–130.

[8] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.

[9] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan, "Active learning in recommender systems," pp. 809–846, 2015.

[10] Yusuf Aytar, Carl Vondrick, and Antonio Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016.

[11] Wenjing Han, Eduardo Coutinho, Huabin Ruan, Haifeng Li, Bjrn Schuller, Xiaojie Yu, and Xuan Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments," *PLOS ONE*, vol. 11, no. 9, pp. 1–23, 09 2016.

[12] Shuyang Zhao, Toni Heittola, and Tuomas Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 751–755.

[13] Zixing Zhang and Björn W. Schuller, "Semi-supervised learning helps in sound event classification," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, 2012, pp. 333–336.

[14] Hyunjune S. Seung, Mike Opper, and Haim Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, New York, NY, USA, 1992, pp. 287–294.

[15] Marko Leonard Helén and Tuomas Virtanen, "Audio query by example using similarity measures between probability density functions of features," *EURASIP J. Audio, Speech and Music Processing*, vol. 2010, 2010.

[16] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera, "Semantic audio content-based music recommendation and visualization based on user preference examples," *Inf. Process. Manage.*, vol. 49, no. 1, pp. 13–33, 2013.

[17] Rui Lu, Kailun Wu, Zhiyao Duan, and Changshui Zhang, "Deep ranking: Triplet matchnet for music metric learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 121–125.

[18] Yichi Zhang and Zhiyao Duan, "IMINET: convolutional semi-siamese networks for sound search by vocal imitation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2017, New Paltz, NY, USA, October 15-18, 2017*, 2017, pp. 304–308.

[19] Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley, 1990.

[20] Hae-Sang Park and Chi-Hyuck Jun, "A simple and fast algorithm for k-medoids clustering," *Expert System with Application*, vol. 36, no. 2, pp. 3336–3341, 2009.

[21] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *4th SIAM International Conference on Data Mining*, 2004, pp. 333–344.

[22] Dorit S. Hochbaum and David B. Shmoys, "A best possible heuristic for the *k*-center problem," *Mathematics of Operations Research*, vol. 10, no. 2, pp. 180–184, 1985.

[23] Giuseppe Riccardi and Dilek Hakkani-Tür, "Active learning: theory and applications to automatic speech recognition," *IEEE Transaction on Speech and Audio Processing*, vol. 13, no. 4, pp. 504–511, 2005.

[24] Justin Salamon, Christopher Jacoby, and Juan P. Bello, "A dataset and taxonomy for urban sound research," in *2014 ACM International Conference on Multimedia*, 2014, pp. 1041–1044.

PUBLICATION V

Active Learning for Sound Event Detection
Zhao S.Y., T. Heittola and T. Virtanen

IEEE Transactions on Audio, Speech and Language Processing 28.(2020), 2895–2905

Active Learning for Sound Event Detection

Zhao Shuyang, Toni Heittola, and Tuomas Virtanen

***Abstract*—This paper proposes an active learning system for sound event detection (SED). It aims at maximizing the accuracy of a learned SED model with limited annotation effort. The proposed system analyzes an initially unlabeled audio dataset, from which it selects sound segments for manual annotation. The candidate segments are generated based on a proposed change point detection approach, and the selection is based on the principle of mismatch-first farthest-traversal. During the training of SED models, recordings are used as training inputs, preserving the long-term context for annotated segments. The proposed system clearly outperforms reference methods in the two datasets used for evaluation (TUT Rare Sound 2017 and TAU Spatial Sound 2019). Training with recordings as context outperforms training with only annotated segments. Mismatch-first farthest-traversal outperforms reference sample selection methods based on random sampling and uncertainty sampling. Remarkably, the required annotation effort can be greatly reduced on the dataset where target sound events are rare: by annotating only 2% of the training data, the achieved SED performance is similar to annotating all the training data.**

***Index Terms*—Active learning, sound event detection, change point detection, mismatch-first farthest-traversal, weakly supervised learning**

I. Introduction

Sound event detection (SED) is a task of automatically identifying sound events such as gunshot, glass smash, and baby cry from an audio signal. It predicts the presence of each target sound event and its onset/offset. SED has been applied in various applications, including noise monitoring [1], healthcare monitoring [2], wildlife monitoring [3], urban analysis [4], and multimedia indexing and retrieval [5].

Due to the large number and variability of sound events in real-life acoustic environments, there does not exist a universal SED model. Most SED applications require their own models. The development of a SED model is commonly based on supervised learning, which typically requires a large amount of labeled data as training material. Compared to capturing audio, annotating them is much more time-consuming in most cases. Thus, a practical problem is to optimize the SED accuracy with a limited annotation effort.

Recently, weakly supervised learning has been studied to reduce the required annotation effort in the development of SED models [6], [7]. Weak labels indicate the presence of target event classes in an audio signal, without temporally locating them. In most cases, assigning weak labels is much simpler, compared to assigning strong labels, which requires the onset/offset of each individual sound event.

Despite the existence of weakly supervised learning, annotating a large amount of data is still time-consuming. Active learning has been used in various machine learning problems [8], [9], where labels are difficult, time-consuming, or expensive to obtain. An active learning algorithm controls a labeling process by selecting the data to be labeled, typically based on an estimate of the capability to improve an existing model. In most cases, active learning targets the situation where unlabeled data is abundant, but the amount of annotations that can be made is limited. The total duration of audio that can be manually labeled is called a labeling budget.

Active learning for SED has previously not been studied, though a few active learning studies have been made on sound classification [10], [11], [12], [13], [14]. All of these studies are limited to single-label classification on sound segment datasets [15], [16], where a sound segment contains an isolated event. However, the situation is different in SED, which typically deals with long signals containing many sound events, possibly overlapping in time. In this paper, we propose an active learning system for SED. The proposed system includes the following novelties: (i) Variable-length sound segments are generated as selection candidates using a change point detection approach. To the best of our knowledge, audio change point detection has previously not been used for active learning. Change point detection is used to avoid generating segments that contain only a part of an event, which is sometimes hard to recognize either manually or automatically. (ii) The selection of candidate segments is based on the mismatch-first farthest-traversal principle, which has been shown effective in sound classification [14]. In this study, the selection principle is generalized to the whole labeling process, without clustering in the first stage as is originally proposed. As a result, the process does not require the cluster number as a hyper-parameter, which is sometimes hard to estimate. Furthermore, the sample selection method is extended to multi-label classification. (iii) We propose to use a partial sequence loss during the training of SED models, to preserve the temporal context of annotated segments: each recording is used as training input and the training loss is computed based on only the outputs within annotated segments. Previously, segments generated from the same recordings are processed independently in the training, such as in UrbanSound8K [16] and AudioSet [17].

The structure of the rest of the paper is as follows. Related works are discussed in Section II. The proposed system is introduced in Section III. The evaluation of the proposed system is presented in Section IV. The conclusions are drawn in Section V.

II. Related works

A. Weakly supervised learning

Weakly supervised learning has recently attracted lots of research interests in the field of SED, especially after the

The research leading to these results has received funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND.

release of a large publicly available sound event dataset, AudioSet [17], which provides only weak labels. AudioSet has been used to learn high-level representations in [18]. The learned representation clearly outperforms hand-crafted features such as log-mel spectrogram in an environmental sound classification dataset [15] and an acoustic scene classification dataset [19]. Furthermore, weakly supervised learning can be also used to directly learn SED models, such as in Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 task 4 subtask B [20].

Previous weakly supervised learning studies [6], [7], [18], [21], [22] use pooling functions to aggregate frame-level class probabilities into segment-level. Among the studied pooling functions, attention pooling [6], [21] appears to be the most popular one [22]. Besides the class probability, an attention neural network [6], [21] predicts a pooling weight for each frame. The segment-level output is based on the weighted average of the frame-level class probabilities. Besides attention pooling, softmax pooling has also been shown effective in [7], [22]. An adaptive pooling method, introducing a learnable hyper-parameter to softmax pooling [7], achieved similar SED performances compared to using strong labels, in three SED datasets.

B. Sample selection

There are different problem setups defined in the field of active learning. Previous studies on sound classification follow the setup of pool-based sampling, where a large collection of unlabeled data is available from the very beginning of a labeling process. Uncertainty sampling method was studied in [10], [11], [12], where the uncertainty to classify a sample with an existing model was used for sample selection. One of the problems with uncertainty sampling is the unreliable certainty estimation unless a decent amount of data is labeled. In many cases, uncertainty sampling does not outperform random sampling when the labeling budget is low [10], [11]. Another problem with uncertainty sampling is the low diversity in a selection batch, since the samples uncertain to the same model are often similar [8], [23].

Cluster-based active learning was proposed in [13]. Segment-to-segment similarities were measured based on the distribution of MFCCs in each sound segment in the training dataset. K-medoids clustering was performed on the sound segments, and the centroids of clusters (medoids) were selected for annotation. The method is called medoid-based active learning (MAL). A label assigned to a medoid was propagated to all segments within the same cluster. When all the medoids were annotated, another round of clustering was performed. Both the annotated labels and the propagated labels were used in training acoustic models. MAL relies completely on the similarity measurement. The advantage is that it enables good performance with a low labeling budget, since it does not require a reliable model. However, the method is not optimal as the labeling budget grows, since the selection of samples does not take previously annotated samples into account. Another problem is that the choice of the number of clusters K requires a prior knowledge about a dataset.

As an extension of MAL, mismatch-first farthest-traversal was proposed in [14]. It performs only one round of K-medoids clustering as the first stage. After annotating the medoids, the sample selection is continued with mismatch-first farthest-traversal as the second stage. The samples with mismatched predictions were selected as the primary criterion. They were further selected by their distances to previously selected samples as the secondary criterion. The target is to maximize the diversity of selected samples. The first stage of the method is equivalent to MAL, and the second stage, which starts at the labeling budget of k, clearly outperforms the original MAL and other reference methods with all evaluated labeling budget. In addition, an approach was proposed to estimate the cluster number K. However, it assumed a relatively balanced number of instances from each sound class. This assumption can hardly be satisfied in SED problems.

In comparison to the previous active learning studies on sound classification [12], [13], [14], the problem setup in this study has the following differences. Firstly, generating segments for annotation is considered as a part of the active learning system in this study, whereas previous studies utilize sound segments that are already generated before the active learning process. Secondly, this study allows a set of classes assigned to a segment, whereas the previous studies require exactly only one class assigned to a segment. Thirdly, this study predicts not only the event class as the previous studies, but also the onset and offset of each individual event.

III. The proposed method

The proposed active learning system aims at optimizing the accuracy of a learned SED model, with a limited annotation effort. The general overview of the proposed system is illustrated in Figure 1. It takes a set of unlabeled audio recordings as input and outputs a SED model. A human annotator is required to assign labels to sound segments that the system selects from the recordings. The SED model is trained with annotated sound segments.

At the beginning of the active learning process, change point detection is performed, splitting each recording into segments. Each segment, later called a sample, is used as a candidate for being selected to be annotated. The definition of sample, sampling, and training example follows [13]. The active learning process is iterative, following batch mode active learning scheme [8]. In each iteration, a batch of samples is selected for annotation, and a SED model is trained with annotated samples. The sample selection is based on mismatch-first farthest-traversal. Mismatch-first as the primary criterion targets on the samples that are previously wrongly predicted. Farthest-traversal as the secondary criterion aims at maximizing the diversity of selected samples.

In order to save annotation effort, the system requires only weak labels that are assigned to individual segments. In each recording, the annotated segments are visualized in pink in Fig.1. During the training of SED models, original recordings are used as training inputs, regarded as partially labeled sequences. The training loss is derived from only the annotated parts of each recording, and the unlabeled parts are used to provide context information.

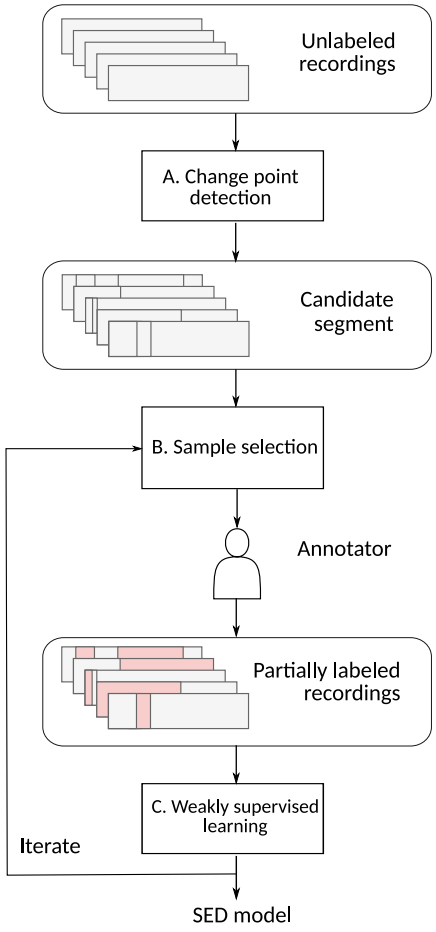

Fig. 1: The overview of the proposed active learning system. The three processing blocks correspond to the three subsections in Section II.

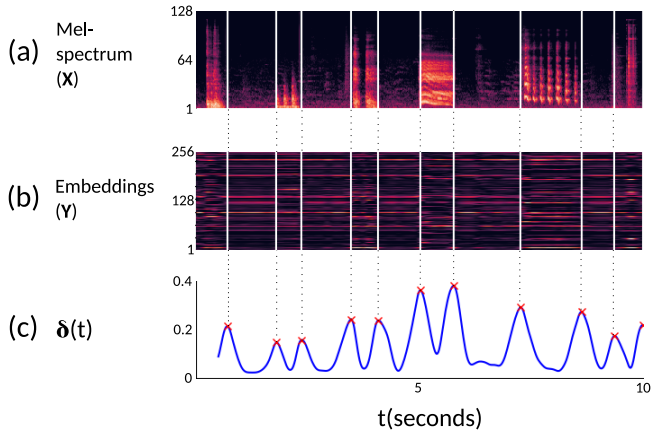

Fig. 2: Panel (a) is the log-mel spectrogram of an example audio signal, with the detected change points marked by white vertical lines. Panel (b) visualizes the embeddings extracted using a pre-trained model. Panel (c) illustrates the estimated likelihood of change on each time step. The peaks in the likelihood sequence are detected as change points, which are marked with red crosses.

A. *Change point detection*

In the proposed system, recordings are first split into short segments, as illustrated in Figure 2. Short segments have two advantages over full recordings as basic units for annotation. The temporal resolution of weak labels, indicating event presences in each recording, is sometimes insufficient to train SED models, especially when sound events are dense. In addition, the diversity of acoustic content in a recording is sometimes limited, since the sounds are typically produced from the same sources. In many cases, annotating only representative segments within each recording is sufficient.

The segments are generated based on a change point detection approach, in order to obtain segments containing complete sound events, since segments with only part of an event are sometimes difficult to annotate. Aiming at discriminative features for sound event activities, embeddings are extracted per frame using a pre-trained model. The architecture of the pre-trained model follows the network architecture used in [21]. The details of the architecture is described in Section III C. The training material and validation criterion used for training the pre-trained model generally follows the setup in [18]. Change point detection is performed on the embeddings $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_T]$, where each embedding vector $\mathbf{y}_t$ corresponds to the time frame $t = 1, 2, ...T$. A likelihood of a change $\delta(t)$ is measured for each time frame t by the cosine distance between the means of the past M frames and the future M frames. The M frames correspond to 0.5 seconds, thus one second is the length of the analysis window for the estimation of $\delta(t)$. Previous unsupervised audio segmentation approaches are mostly proposed for speaker diarization [24], [25]. These methods typically use a fixed or variable length analysis window around two seconds, based on the expected duration of speaker utterances [24]. This study uses an analysis window of one second based on the expected duration of short sound events such as gunshot or glass break.

The panel (c) in Figure 2 illustrates the likelihood of change estimated at each frame in an example audio signal. A peak in the likelihood is used as a change point. The change points divide an audio signal into segments, which are used as candidates for sample selection and annotation.

B. *Sample Selection*

Figure 3 illustrates the active learning process with the generated candidate segments as samples. The sample selection method follows the principle of mismatch-first farthest-traversal [14]. Detailed visualization of the sample selection method is given online[1].

When selecting the first batch of samples, no annotated samples are available. In order to maximize the diversity of selected samples, farthest-traversal is performed on the whole training set. Farthest-traversal is explained later in this section. An annotator assigns labels to the selected samples, with which a SED model is trained.

Two types of predicted labels are generated for each unlabeled sample. Based on a trained SED model, *model-predicted*

[1]https://github.com/zhao-shuyang/active_learning

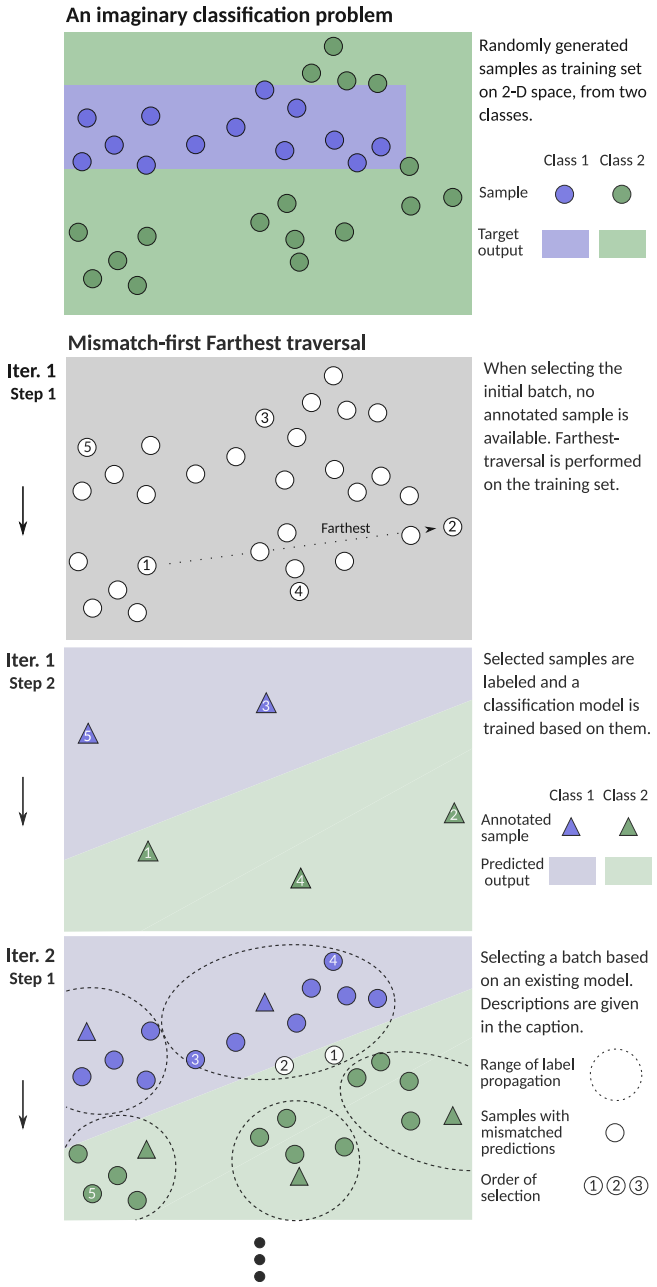

Fig. 3: A visualization of mismatch-first farthest-traversal on an imaginary binary classification problem. In the bottom panel, the range of label propagation is used to visualize the area where an annotated data point propagates its label. Farthest-traversal is first performed on samples where propagated labels mismatch with model predictions, and then on samples with matched predictions.

labels are generated. Based on the nearest neighbor prediction, *propagated labels* are generated, according to a distance metric. The similarity between the two types of predicted labels is measured for each unlabeled sample. The measurement of the prediction similarity is given in the subsection about the mismatch-first criterion. The samples are primarily ranked by the prediction similarities, lowest first. There are typically multiple samples with the same prediction similarities. They

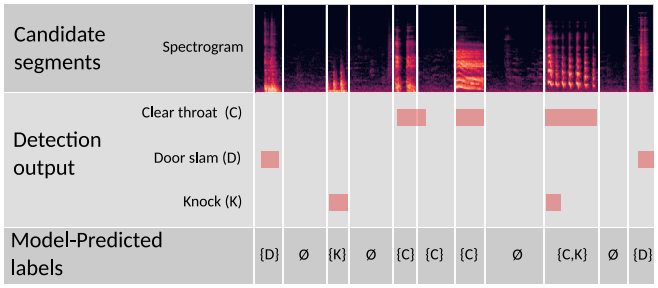

Fig. 4: An example of deriving model-predicted labels from sound event detection output.

are further ranked by the distance to the previously selected samples, farthest first. A batch of samples with the highest rank is presented to the annotator and the active learning process continues to the next iteration.

1) Mismatch-first criterion: At the beginning of each iteration, except the first one, model-predicted labels and propagated labels are generated for each unlabeled sample. Model-predicted labels are derived from the SED outputs of each recording as is illustrated in Figure 4. When a class of sound event is detected within a candidate segment, a model-predicted label is generated, associating the class of the sound event to the segment. The classes associated with a sample x according to the SED outputs are denoted as a set $\mathcal{A}_x$. Propagated labels are generated based on the nearest neighbor prediction. Each unlabeled sample x is assigned the labels of its nearest annotated sample. The distance between two samples is measured by the cosine distance between the means of embeddings within the two samples. These propagated labels are denoted as a set $\mathcal{B}_x$.

In a multi-label classification problem, the similarity between the propagated labels and the model-predicted labels on a sample x is measured based on the Jaccard index as,

$$J(x) = \begin{cases} \frac{|\mathcal{A}_x \cap \mathcal{B}_x|}{|\mathcal{A}_x \cup \mathcal{B}_x|} & , \text{if } \mathcal{A}_x \cup \mathcal{B}_x \neq \emptyset \\ 1 & , \text{if } \mathcal{A}_x \cup \mathcal{B}_x = \emptyset \end{cases}. \quad (1)$$

Samples are first selected within the set $\mathcal{M}$, which consists of the samples with the lowest prediction similarities among the set of unlabeled samples.

The mismatch-first criterion is based on an assumption that a model benefits more from a counterexample, where it makes an error, in comparison to an example where it makes a correct prediction. When the prediction results based on two mechanisms mismatch, the sample is a counter example for at least one of the mechanisms. Since the nearest neighbor prediction and neural network prediction are two fundamentally different mechanisms, their prediction results are usually supplementary information to each other. In addition, the two prediction mechanisms are based on different contexts. The nearest neighbor prediction is based only on annotated segments, whereas the SED model uses original recordings as a context for annotated segments.

2) Farthest-traversal: Farthest-traversal aims at optimizing the diversity of selected samples. It selects the sample farthest

to the previously selected samples. The distance between two samples is measured by the cosine distance between the means of embeddings within the two samples. The previously selected samples are denoted as a set $\mathcal{S}$, which is the union of annotated samples and the samples already selected in the current iteration. As a result, a selected sample is neither similar to annotated samples, nor to the ones to be annotated in the same batch. The distance from a sample x to the set of previously selected samples $\mathcal{S}$ is defined as $d(x, \mathcal{S}) = min_{y \in \mathcal{S}} d(x, y)$.

With mismatch-first as the primary criterion and farthest-traversal as the secondary criterion, a sample is selected as

$$s = \underset{x \in \mathcal{M}}{\operatorname{argmax}}\, d(x, \mathcal{S}), \tag{2}$$

where $\mathcal{M}$ is the set of samples with the lowest prediction similarities.

The selected samples are added one by one into a selection batch and removed from the set of unlabeled samples until the batch reaches a pre-defined batch size. After that, the batch of selected samples is presented to the annotator, querying for weak labels. Weak labels of a segment is a set of sound event classes, that are present in the segment.

Previous active learning studies on sound classification incorporate the idea of semi-supervised learning, where predicted labels on unlabeled data are also used in training [12], [13], [14]. Since semi-supervised learning techniques have been rapidly developed in recent years, this study considers semi-supervised learning as a separate problem and focuses only on active learning. The optimal combination with semi-supervised learning is considered as future work.

C. Weakly supervised learning

Previous active learning studies [12], [13], [14] use support vector machine to classify sound segments. This study uses a neural network to perform SED, since neural networks are commonly used for SED problems. The architecture of the network follows an attention-based weakly supervised learning system [21], which ranks the 1st in the audio tagging subtask and the 2nd in SED subtask in a weakly supervised learning challenge, DCASE 2017 task 4. In [21], each training input is an annotated segment sliced from a YouTube video. In comparison, this study uses each original recording as a training input, preserving the context for annotated segments.

The network architecture is illustrated in Figure 5. The input of the network is the log-mel spectrogram of a recording, denoted as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_T]$, where each vector $\mathbf{x}_t$ represents the log-mel band energies in a time frame $t = 1, 2, ...T$. The target output is a vector $\boldsymbol{\tau}$, corresponding to the event class activities. Each element in the target output vector $\boldsymbol{\tau} = [\tau_1, ..., \tau_C]$ represents the presence/absence of an event class, 0 for absence and 1 for presence, and C denotes the number of classes.

The network consists of six blocks of gated CNNs, each of which consists of a linear CNN layer and a sigmoid CNN layer. The element-wise product between the outputs of the two CNN layers is fed to the next layer. Compared to traditional CNNs that use rectified linear units as activation function, the gated CNNs reduce the gradient vanishing

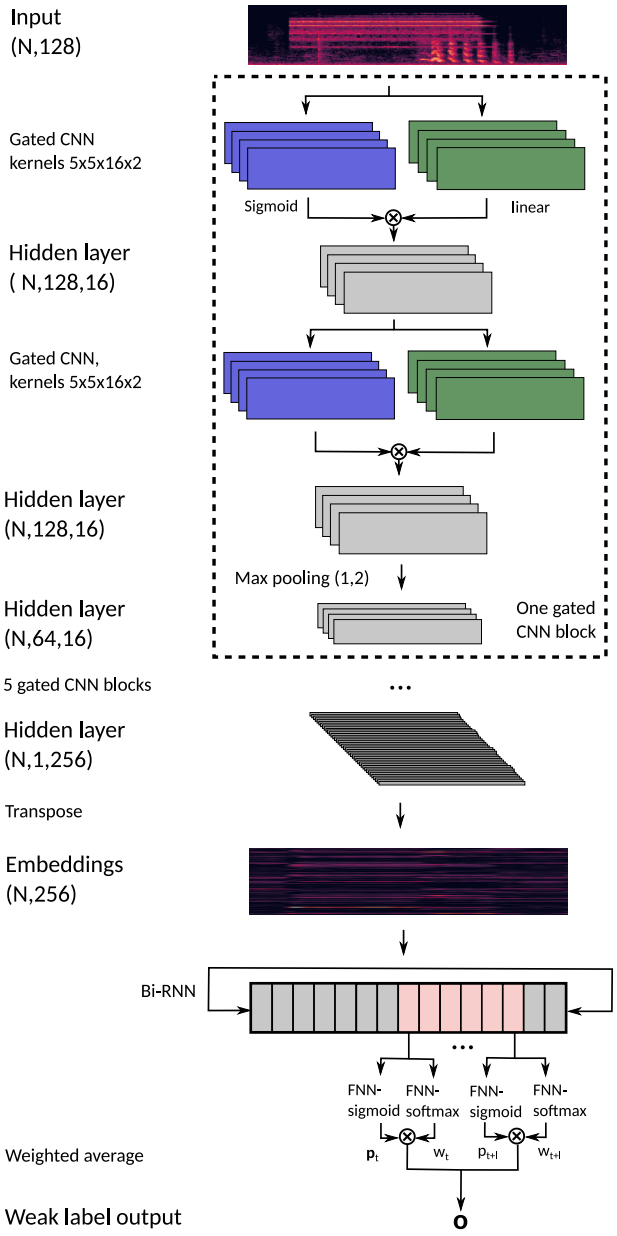

Fig. 5: The diagram of the network architecture used in weakly supervised learning. The frames marked red in the bidirectional RNN outputs correspond to an annotated segment.

problem in a deep structure [26]. The gated CNNs transfer the input log-mel spectrogram into a sequence of embeddings $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_T]$, where an embedding vector $\mathbf{y}_t$ corresponds to a time step t. In order to model a long-term temporal context, three bi-directional gated recurrent unit (GRU) layers are used. The GRUs process the embedding sequence, and output a vector $\mathbf{y}'_t$ in each time step. A fully-connected sigmoid layer is used to estimate the class probabilities in each time step as $\mathbf{p}_t = cla(\mathbf{y}'_t)$. In parallel, a fully-connected softmax layer estimates the pooling weights as $\mathbf{w}_t = att(\mathbf{y}'_t)$.

In order to derive the output for an annotated segment, the weighted average of the class probabilities is computed across all frames within the segment. Given the start time point of a

segment as t and the length of it as l, the weak label output of the segment is computed as

$$\mathbf{o} = \frac{\sum_{i=t}^{t+l} \mathbf{w}_i \cdot \mathbf{p}_i}{\sum_{j=t}^{t+l} \mathbf{w}_j}, \quad (3)$$

where $\cdot$ represents element-wise multiplication. Binary cross-entropy is used to measure the loss between the prediction output $\mathbf{o}$ and the target output $\boldsymbol{\tau}$ for each annotated segment, as

$$L(\boldsymbol{\tau}, \mathbf{o}) = \sum_{k=1}^{C} -(o_k \log(\tau_k) + (1 - o_k) \log(1 - \tau_k)), \quad (4)$$

where C is the number of classes. The training loss for a recording is the sum of the loss from each annotated segment within it.

In this study, the gated CNNs that extract embeddings are pre-trained with the balanced set of AudioSet [17]. The embedding extraction function is considered as a general knowledge, which can be transferred to different SED problems. During the pre-training, the GRU layers are not used, and embedding vectors are directly fed to the fully-connected layers. The output of the second last layer of a classification network is used as embeddings. This follows the common practice in previous transfer learning studies [18], [27] on sound classification.

In the active learning process, the pre-trained embedding extraction function e is fixed. The parameters of the GRU layers gru, the sigmoid layer cla, and the softmax layer att are trained with data annotated in the active learning process. With a limited labeling budget, usually a small number of segments are labeled in each recording. During the training, the log-mel spectrogram of full recordings are used as input, but the training loss is derived from only the frames corresponding to labeled segments. When performing SED on test data, the detection output is based on the class probabilities, the output of cla, without using the layer att.

Previous studies [16], [17], [13], [14] use each annotated segment as input, instead of the original recordings. As a result, they lose the contextual information in the original recordings. The contextual information may benefit the SED performance from different aspects. Firstly, given background sounds as contextual information, a model can learn the unique characteristics of an event out of the background. Secondly, the contextual information can be used to model the dependencies between acoustic events and scenes. For example, it is common to hear key rattling before door opening and it is common to hear a bird chirping in a forest.

IV. Evaluation

In order to evaluate the performance of the proposed system, two sets of experiments are made on two different datasets. The first one focuses on the training input and annotation unit. The second one focuses on the sample selection method.

A. Datasets and settings

In order to evaluate active learning performances with different SED scenarios, two SED datasets are used in the evaluation. The statistics comparing the two datasets are shown in Table I. The first dataset is TUT Rare Sound Events 2017 [20], which is used in the challenge of Detection and Classification of Acoustic Scenes and Events (DCASE) 2017, as task 2. The second dataset is TAU Spatial Sound Events 2019 - Ambisonic, which is used in the challenge of DCASE 2019 [28], as task 3.

Both datasets consist of synthetic mixtures created by mixing isolated sound event clips with background sounds. Previous sound event detection studies [29], [30] use synthetic datasets as primary evaluation datasets, since the timestamps of sound events in these datasets are precise and consistent. In contrast, real-life recordings use manual annotation, where the subjectivity may lead to inconsistency and possible errors in the labels. The two datasets in this study are chosen to represent scenarios with different sound event densities, which largely affects the active learning performance.

Dataset	TUT Rare Sound Events 2017	TAU Spatial Sound Events 2019
Total duration	25 h	6 h 40 m
Training set duration	12 h 30 m	5 h
Target event classes	3	11
EBR	[-6 db, 0 db, 6 db]	30 db
Recording length	30 s	1 m
Events per minute	1	55

TABLE I: A Summary of datasets used in the evaluation, explained in Section IV.A.

1) TUT Rare Sound Events 2017: TUT Rare Sound Events 2017 dataset, referred to as rare sound dataset later, is created by mixing isolated target sounds from Freesound with background audio in TUT Acoustic Scenes 2016 dataset [19]. There are three target event classes: baby cry, gunshot, and glass breaking. Most gunshot and glass breaking sounds are short, lasting around 200 milliseconds. In comparison, baby cry events are longer, typically ranging from one to four seconds. The background consists of sounds from 15 classes of real acoustic scenes, 78 instances each class. The acoustic scenes are bus, cafe/restaurant, car, city center, forest, grocery store, home, lakeside beach, library, metro station, office, residential area, train, tram, and urban park.

All the background audio tracks last 30 seconds. The sampling rate is 44100 Hz. An audio signal in the rare sound dataset might be either pure background or a target event mixed with a background. The event-to-background ratio (EBR) in dB is randomly chosen from $\{-6, 0, 6\}$, and the positioning of the target sound in a mixture is also random. The sound events are rare in this dataset, on average one event per minute.

The original rare sound dataset is split into a development training set, development test set, and evaluation set. Each split of the dataset contains mixtures created with a separate set of

background and target sounds. In this study, the development training set is used for training, and the development test set is used for evaluation. Both the training and test set contains approximately 1500 audio signals, with 250 target events of each class.

2) TAU Spatial Sound Events 2019: The dataset TAU Spatial Sound Events 2019 dataset, is originally a spatial audio dataset, which is used for sound event detection and spatial localization task in DCASE 2019 challenge. The dataset is synthetic, and the source of the mixtures are sound events from 11 classes, with 20 instances in each class. Each recording in the spatial sound dataset has around one-minute duration, which is mixed with target sound events. On average, each minute of the signal contains 55 events, randomly positioned, with possibly overlapping in time. The background is relatively quiet and the EBR of the mixtures is about 30 dB.

The original sampling rate of the dataset is 48 kHz. In the experiments, the recordings are resampled to 44.1 kHz, to match the sampling rate of the pre-trained embedding extraction model. The audio in this dataset has four channels, however, only the first channel is used in this study, since this study does not deal with multi-channel audio.

Similar to the usage of the rare sound dataset, this study uses only the development set, ignoring the evaluation set in the challenge. Four-fold cross-validation is used, following the original setup of the dataset.

B. Evaluation metric

In this study, a segment-based error rate (ER) is used to evaluate the performance of a SED model [31]. The segment length in the segment-based evaluation is one second, which is a common setup in sound event detection studies, such as DCASE 2017 task 3.

The aim of active learning is to optimize the accuracy of learned SED models with a limited labeling budget. Thus, the active learning performance is evaluated by ER as a function of the labeling budget, which is given in proportion to the whole training set.

C. Basic experimental setups

Experiments are made to evaluate each component in the proposed active learning system. This section describes common setups used through all the experiments in the evaluation.

When computing the spectrogram, the frame length is 40 ms and hop length is 20 ms. In each frame, the signal is windowed with the Hanning window and then log-mel energies in 128 bands are calculated. The gated CNN pre-trained with AudioSet maps a log-mel spectrogram into an embedding sequence with the same number of frames and 256 dimensions.

The likelihood of change is estimated for each frame based on the past 24 frames and the future 24 frames, aggregating to an analysis window of one second. Detected change points can be closer than one second, for example, the second and third change point in Fig 2. However, annotating very short segments can be difficult in practice. The actual annotation effort is underestimated, when the annotator needs to listen to the extra context of a candidate segment for annotation. In order to avoid very short segments, the change points detected within one second to the previous ones are skipped when generating the candidate segments. As a result, the minimum length of the generated segments is one second.

In the simulation of the labeling process, the ground truth labels are initially hidden to the system. Upon the label query on a segment, annotated labels are simulated according to the ground truth. When a ground-truth sound event overlaps a queried candidate segment with more than 0.1 seconds, a weak label is generated, associating the event class with the segment. It is presumed that an event shorter than 0.1 seconds cannot be perceived by an annotator.

A SED model is trained with simulated annotations and the performance is benchmarked when the number of simulated labels reaches an evaluated labeling budget. In this study, the following proportions of the training data as labeling budget are evaluated: $1\%, 2\%, 3\%, 4\%, 5\%, 6\%, 7\%, 8\%, 9\%, 10\%, 20\%, 100\%$. During the training of a SED model in each iteration, one-third of the labeled data is randomly chosen for validation.

The experiments on the TUT Rare Sound dataset are repeated five times, and the average performance is reported. The 4-fold validation experiments on the TAU Spatial sound dataset are repeated twice, and the average of the eight results is reported.

In all the experiments with reported results, the same network architecture is used. A preliminary study was made to investigate the effect of the model complexity with low labeling budget: we tested using a single GRU layer instead of three when only 1% of the training data was labeled. As a result, the performances are similar among the tested models with different number of layers.

D. Experiments

In order to evaluate each component in the proposed active learning system, four experiments have been made, as is summarized in Table II.

The proposed system uses variable-length segments as candidate segments for annotation. In order to preserve the context for the annotated segments, the original recordings are used as training inputs, regarded as partially labeled sequences. Experiment A evaluates the effect of preserving the context. Experiment A1 investigates the training input. System 1 uses full recordings as training inputs as is proposed, whereas System 2 uses only annotated segments as training inputs. Experiment A2 investigates the annotation unit. System 3 uses variable-length segments as an annotation unit as is proposed, whereas System 4 uses a full recording as an annotation unit. Strong labels are used in experiment A2 since weak labels are not informative for full recordings in the TAU Spatial Sound dataset, where most recordings include all the 11 sound event classes. During the model training with strong labels, the attention layer is not used and the training loss is directly computed as the binary cross-entropy between the target and the class probability output on a frame basis. Random sampling is used in all the systems in Experiment A.

	System	Annotation unit	Label type	Sample selection method	Training input
Experiment A1	1	variable-length segment	weak label	random sampling	**recordings**
	2	variable-length segment	weak label	random sampling	**segments**
Experiment A2	3	**variable-length segment**	strong label	random sampling	recordings
	4	**recording**	strong label	random sampling	recordings
Experiment B	1	variable-length segment	weak label	**random sampling**	recordings
	5	variable-length segment	weak label	**mismatch-first farthest-traversal**	recordings
	6	variable-length segment	weak label	**uncertainty sampling**	recordings
Experiment C	5	**variable-length segment**	weak label	mismatch-first farthest-traversal	recordings
	7	**fixed-length segment**	weak label	mismatch-first farthest-traversal	recordings

TABLE II: A summary of experiments. Bold font is used to highlight the investigated aspect in each experiment.

Experiment B focuses on the sample selection method. It compares mismatch-first farthest-traversal, with two reference methods based on random sampling and uncertainty sampling. Random sampling is used in System 1, which is also used in Experiment A1. System 5 uses mismatch-first farthest-traversal, and System 6 uses uncertainty sampling. In random sampling, each candidate segment has an equal probability of being selected. In uncertainty sampling, the certainty of predicting a class c is measured as $2 \times |o_c - 0.5|$, where o_c is the weak label output or segment-wise class probability. The overall prediction certainty on a sample is defined as the minimum prediction certainty over all the classes. Since uncertainty sampling and mismatch-first farthest-traversal are batch mode active learning, the performance depends on the size of a selection batch. Typically a smaller batch size leads to better accuracy, but it requires more training time. In this experiment, the selection batch size is set to 0.5% of the whole trained set, which is about 150 segments in the TUT Rare Sound dataset and 60 segments in the TAU Spatial Sound dataset. The batch size is chosen for convenience, since the performance of the learned SED model is reported after every two selection batches, according to the evaluated labeling budget.

Experiment C focuses on the proposed segmentation method based on change point detection. System 5 is a combination of all proposed components in this study. In comparison to System 5, System 7 uses segments with a fixed-length of two seconds. The total number of fixed-length segments is similar to the total number of variable-length segments generated using change point detection.

E. Experimental results

The results of experiment A1, illustrated in Figure 6, show that preserving original recordings as the context clearly outperforms training with only annotated segments. In some cases, more than 60% of the labeling budget can be saved to achieve the same accuracy. A sound event is sometimes detected not only based on the audio signal where the event happens but also the difference compared to the background sounds in the temporal context, preserved in the original recordings. The results of experiment A2, illustrated in Figure 7, show that annotating segments is more efficient compared to annotating full recordings. The segments randomly sampled from all the recordings have typically higher diversity, in comparison to a small amount of fully annotated recordings. In addition, by comparing the results of System 1 and System 3, close performance is achieved by using attention pooling with weak labels, compared to using strong labels.

The experimental results comparing the sampling methods are illustrated in Figure 8. The results show that the proposed method outperforms reference methods with all evaluated labeling budgets.

In the experiments on the TUT Rare Sound dataset, the proposed method outperforms reference methods to a large extent. Most of the training data have little relevance to the target problem since the target sound events are rare in this dataset. Therefore, the annotation effort can be greatly reduced by selective sampling, if irrelevant data can be ruled out in the sample selection. In addition, uncertainty sampling also outperforms random sampling to a large extent.

Remarkably, the proposed active learning method requires only 2% of the training data to be annotated to achieve similar performance, compared to annotating all the data. Surprisingly, the best performance is achieved by annotating only 5% of the training set. The sound events are rare in the dataset, and most of the segments containing target events are selected within the first 5% of the training set. By the time when 5% of the training data is labeled in a typical case, the segments containing a target event comprise 35% of the labeled data, whereas, only 1.25% of the unlabeled data contains a target event. Although more labeled data is available when labeling budget increases, the high label distribution bias has a negative effect on the accuracy of learned models. As a result, the accuracy does not improve with increasing labeling budget.

In the experiments on the TAU Spatial Sound dataset, The proposed method slightly outperforms the two reference methods. In the TAU Spatial Sound dataset, target sound events are dense. In principle, little improvement can be made with selective sampling, when majority of the dataset are relevant to the target SED problem. In this case, the proposed method cannot save much annotation effort.

Combining the effect of sample selection and training with original recordings as context, a clear improvement in performance can be made with the proposed system. This can be evaluated by comparing System 5 with System 2. To achieve ER of 0.55 in the TUT Rare Sound dataset, System 2 requires 20% of the training set as a labeling budget. In comparison, the proposed method, System 5 requires annotating only 1% of the training set. To achieve ER of 0.5 in the TAU Spatial Sound dataset, System 2 requires 6% of the training set as labeling budget. In comparison, System 5 requires annotating only 4% of the training set.

The experimental results comparing the two segmentation

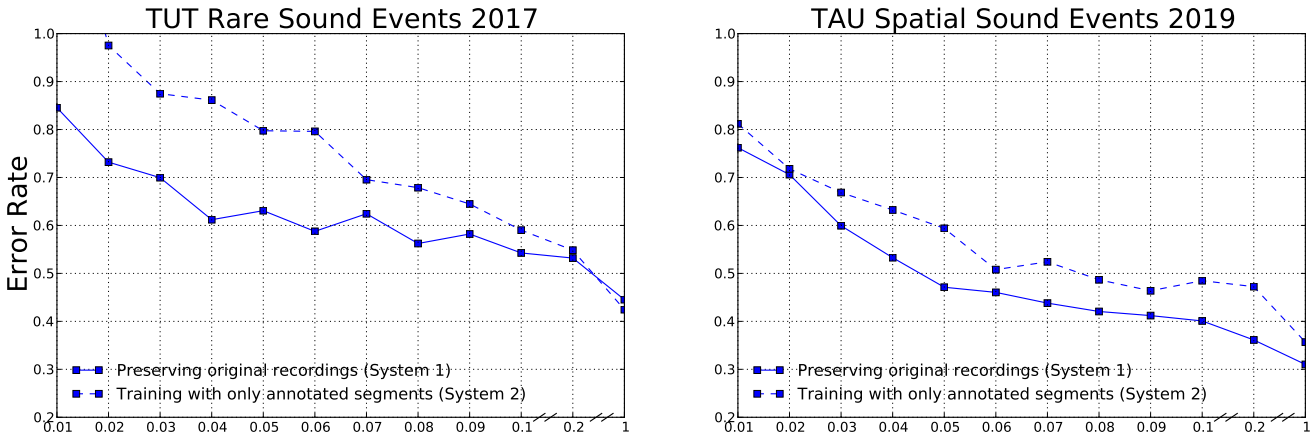

Fig. 6: Error rate of learned models as the function of labeling budget for methods that use different training inputs, corresponding to experiment A1.

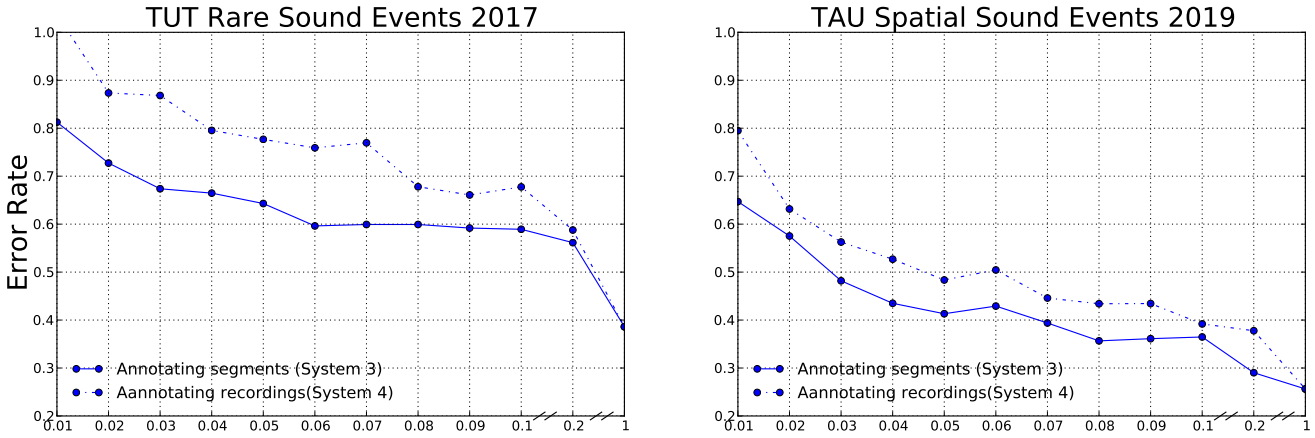

Fig. 7: Error rate of learned models as the function of labeling budget for methods that use different annotation units, corresponding to experiment A2.

methods are illustrated in Figure 9, when mismatch-first farthest-traversal is used. The experiments show that variable-length segments lead to better performance. Mismatch-first farthest-traversal largely depends on the similarity analysis. Since fixed-length segments often contain part of events, the similarities between fixed-length segments are less relevant to their labels, compared to the similarities between variable-length segments, which is targeted to contain complete events.

V. Conclusion

In this study, we propose an active learning system for sound event detection (SED), which targets on optimizing the accuracy of a learned SED model with limited annotation effort. The proposed system analyzes an initially unlabeled audio dataset, querying for weak labels on selected sound segments from the dataset. A change point detection method is used to generate variable-length audio segments. The segments are selected and presented to an annotator, based on the principle of mismatch-first farthest-traversal. During the training, full recordings are used as input to preserve the long-term context for annotated segments.

Experimental results show that training with original recordings as a context for annotated segments clearly outperforms training with only annotated segments. Mismatch-first farthest-traversal clearly outperforms reference sampling methods based on random sampling and uncertainty sampling. The performance of mismatch-first farthest-traversal depends on the segmentation method that generates the candidate segments. Variable-length segments generated by change point detection lead to clearly better performance than fixed-length segments.

Overall, the proposed method effectively saves annotation effort to achieve the same accuracy, with respect to reference methods. The amount of annotation effort can be saved depends on the distribution of target sound events in the training dataset: a larger amount of annotation effort can be saved when the target sound events are rare. On the dataset with rare events, more than 90% of labeling budget can be saved by using the proposed system, with respect to a system that

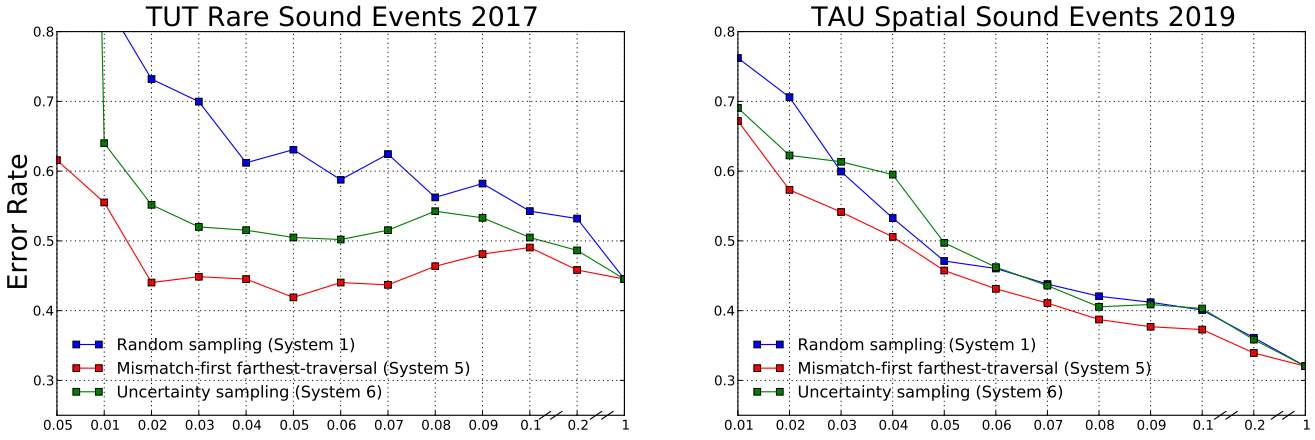

Fig. 8: Error rate of learned models as the function of labeling budget for different sampling methods, corresponding to experiment B.

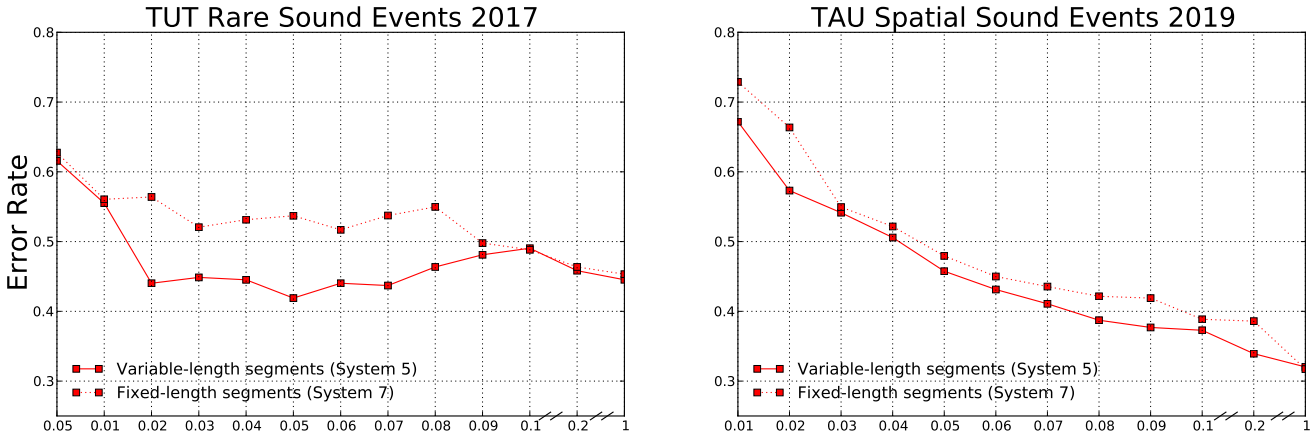

Fig. 9: Error rate of learned models as the function of labeling budget for different segmentation methods, corresponding to experiment C.

uses random sampling and annotated segments only for model learning. Notably, by annotating 2% of the training data, the proposed method achieves the same accuracy as training with all the data.

In future work, the optimal combination of active learning and semi-supervised learning methods can be studied for SED. Recent semi-supervised learning studies, particularly those based on the mean-teacher method [32], have been shown effective for SED problems in DCASE 2019 task 4 [33]. We expect that more annotation effort can be saved, by incorporating semi-supervised learning to further utilize the unlabeled part of the dataset.

REFERENCES

[1] P. Maijala, Zhao S.Y., T. Heittola, and T. Virtanen, "Environmental noise monitoring using source classification in sensors," *Applied Acoustics*, vol. 129, no. 6, p. 258–267, January 2018.

[2] S. Goetze, J. Schröder, S. Gerlach, D. Hollosi, J. Appell, and F. Wallhoff, "Acoustic monitoring and localization for social care," *JCSE*, vol. 6, no. 1, pp. 40–50, 2012. [Online]. Available: https://doi.org/10.5626/JCSE.2012.6.1.40

[3] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, "Robust sound event detection in bioacoustic sensor networks," *CoRR*, vol. abs/1905.08352, 2019. [Online]. Available: http://arxiv.org/abs/1905.08352

[4] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *23rd European Signal Processing Conference, EUSIPCO 2015, Nice, France, August 31 - September 4, 2015*, 2015, pp. 724–728. [Online]. Available: https://doi.org/10.1109/EUSIPCO.2015.7362478

[5] R. Cai, L. Lu, A. Hanjalic, H. Zhang, and L. Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 1026–1039, 2006. [Online]. Available: https://doi.org/10.1109/TSA.2005.857575

[6] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019. [Online]. Available: https://doi.org/10.1109/TASLP.2019.2930913

[7] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 11, pp. 2180–2193, 2018. [Online]. Available: https://doi.org/10.1109/TASLP.2018.2858559

[8] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th international*

conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006, 2006, pp. 633–642. [Online]. Available: https://doi.org/10.1145/1135777.1135870

[9] Y. Liu, "Active learning with support vector machine applied to gene expression data for cancer classification," *Journal of Chemical Information and Modeling*, vol. 44, no. 6, pp. 1936–1941, 2004. [Online]. Available: https://doi.org/10.1021/ci049810a

[10] G. Riccardi and D. Hakkani-Tür, "Active learning: theory and applications to automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 13, no. 4, pp. 504–511, 2005. [Online]. Available: https://doi.org/10.1109/TSA.2005.848882

[11] D. Hakkani-Tür, G. Riccardi, and A. L. Gorin, "Active learning for automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, May 13-17 2002, Orlando, Florida, USA*, 2002, pp. 3904–3907. [Online]. Available: https://doi.org/10.1109/ICASSP.2002.5745510

[12] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments," *PLOS ONE*, vol. 11, no. 9, pp. 1–23, 09 2016.

[13] Zhao S.Y., T. Heittola, and T. Virtanen, "Active learning for sound event classification by clustering unlabeled data," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 751–755.

[14] ——, "An active learning method using clustering and committee-based sample selection for sound event classification," in *16th International Workshop on Acoustic Signal Enhancement, IWAENC 2018, Tokyo, Japan, September 17-20, 2018*, 2018, pp. 116–120. [Online]. Available: https://doi.org/10.1109/IWAENC.2018.8521336

[15] K. J. Piczak, "ESC: dataset for environmental sound classification," in *23rd Annual ACM Conference on Multimedia Conference*, 2015, pp. 1015–1018.

[16] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *2014 ACM International Conference on Multimedia*, 2014, pp. 1041–1044. [Online]. Available: http://doi.acm.org/10.1145/2647868.2655045

[17] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 776–780. [Online]. Available: https://doi.org/10.1109/ICASSP.2017.7952261

[18] A. Kumar, M. Khadkevich, and C. Fügen, "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 2018, pp. 326–330. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8462200

[19] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference, EUSIPCO 2016, Budapest, Hungary, August 29 - September 2, 2016*, 2016, pp. 1128–1132. [Online]. Available: https://doi.org/10.1109/EUSIPCO.2016.7760424

[20] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 85–92.

[21] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 121–125. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8461975

[22] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 2019, pp. 31–35. [Online]. Available: https://doi.org/10.1109/ICASSP.2019.8682847

[23] M. Sassano, "An empirical study of active learning with support vector machines for japanese word segmentation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, 2002, pp. 505–512. [Online]. Available: https://www.aclweb.org/anthology/P02-1064/

[24] M. Kotti, E. Benetos, and C. Kotropoulos, "Computationally efficient and robust bic-based speaker segmentation," *IEEE Trans. Speech Audio Process.*, vol. 16, no. 5, pp. 920–933, 2008. [Online]. Available: https://doi.org/10.1109/TASL.2008.925152

[25] S. Cheng, H. Wang, and H. Fu, "Bic-based audio segmentation by divide-and-conquer," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, 2008, pp. 4841–4844.

[26] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 933–941. [Online]. Available: http://proceedings.mlr.press/v70/dauphin17a.html

[27] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 892–900. [Online]. Available: http://papers.nips.cc/paper/6146-soundnet-learning-sound-representations-from-unlabeled-video

[28] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *J. Sel. Topics Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019. [Online]. Available: https://doi.org/10.1109/JSTSP.2018.2885636

[29] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[30] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 2016, pp. 6440–6444. [Online]. Available: https://doi.org/10.1109/ICASSP.2016.7472917

[31] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[32] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1195–1204.

[33] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," June 2019, working paper or preprint. [Online]. Available: https://hal.inria.fr/hal-02160855

Zhao Shuyang received the M.Sc. degree in signal processing from Tampere University of Technology (TUT), 2014. Since 2013, he has been working in Audio Research Group in TUT, where he is currently working towards ph.D. degree. His main research interests include audio content analysis and machine learning.

Toni Heittola is a doctoral student at Tampere University, Finland. He received his M.Sc. degree in Information Technology from Tampere University of Technology (TUT), Finland, in 2004. He is currently pursuing the Ph.D. degree at Tampere University. His main research interests are sound event detection in real-life environments, sound scene classification and audio content analysis.

Tuomas Virtanen Tuomas Virtanen is Professor at Tampere University, Finland, where he is leading the Audio Research Group. He received the M.Sc. and Doctor of Science degrees in information technology from Tampere University of Technology in 2001 and 2006, respectively. He has also been working as a research associate at Cambridge University Engineering Department, UK. He is known for his pioneering work on single-channel sound source separation using non-negative matrix factorization based techniques, and their application to noise-robust speech recognition and music content analysis. Recently he has done significant contributions to sound event detection in everyday environments. In addition to the above topics, his research interests include content analysis of audio signals in general and machine learning. He has authored more than 190 scientific publications on the above topics, which have been cited more than 9000 times. He has received the IEEE Signal Processing Society 2012 best paper award for his article "Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria" as well as three other best paper awards. He is an IEEE Senior Member, member of the Audio and Acoustic Signal Processing Technical Committee of IEEE Signal Processing Society, and recipient of the ERC 2014 Starting Grant.