Ensemble of Machine Learning and Acoustic Segment Model Techniques for Speech Emotion and Autism Spectrum Disorders Recognition

Hung-yi Lee\textsuperscript{1}, Ting-yao Hu\textsuperscript{2}, How Jing\textsuperscript{3}, Yun-Fan Chang\textsuperscript{1}, Yu Tsao\textsuperscript{1}, Yu-Cheng Kao\textsuperscript{3} and Tsang-Long Pao\textsuperscript{3}

\textsuperscript{1}Research Center for Information Technology Innovation, Academia Sinica
\textsuperscript{2}Graduate Institute of Communication Engineering, National Taiwan University
\textsuperscript{3}Department of Computer Science and Engineering, Tatung University
tlkagkb93901106@gmail.com, yu.tsao@citi.sinica.edu.tw, tlpao@gm.ttu.edu.tw

Abstract

This study investigates the classification performances of emotion and autism spectrum disorders from speech utterances using ensemble classification techniques. We first explore the performances of three well-known machine learning techniques, namely, support vector machines (SVM), deep neural networks (DNN) and k-nearest neighbours (KNN), with acoustic features extracted by the openSMILE feature extractor. In addition, we propose an acoustic segment model (ASM) technique, which incorporates the temporal information of speech signals to perform classification. A set of ASMs is automatically learned for each category of emotion and autism spectrum disorders, and then the ASM sets decode an input utterance into series of acoustic patterns, with which the system determines the category for that utterance. Our ensemble system is a combination of the machine learning and ASM techniques. The evaluations are conducted using the data sets provided by the organizer of the INTERSPEECH 2013 Computational Paralinguistics Challenge.

Index Terms: Autism, Emotion

1. Introduction

Paralinguistic analysis such as the recognition of speech emotion and pathology is increasingly turning into a mainstream topic in speech and language processing [1]. Work on emotion recognition has spanned a large range of approaches [2, 3, 4, 5, 6, 7, 8], and there are some recent studies focusing on the acoustic characteristics of autism spectrum disorders [9, 10, 11, 12, 13, 14, 15]. This paper reports the results of Autism and Emotion Sub-Challenges introduced by INTERSPEECH 2013 Computational Paralinguistics Challenge [16]. In Emotion Sub-Challenge, the Arousal and Valence tasks require a system to determine the dimensions (positive/negative) of arousal or valence of an utterance, and the system should determine the emotion of an utterance from 12 categories in 12-way Emotion task. In Autism Sub-Challenge, the type of pathology of a child has to be determined. Two evaluation tasks have been defined: a binary Typicality task (typically vs. atypically developing children), and a four-way Diagnosis task (further classifying the atypically developing children into different named categories).

Fig. 1 is the architecture of our system used in the challenge. In the system, three well-known machine learning techniques, namely, support vector machines (SVM), deep neural networks (DNN) and k-nearest neighbours (KNN), are used to classify utterances based on their acoustic characteristics, which are represented as fixed-length feature vectors. Here we further investigate the performance of DNN with “dropout” and multi-class SVM using a direct training approach. In addition, to incorporate the temporal information of speech signals in classification, we propose an acoustic segment model (ASM) approach, which classifies utterances by their acoustic feature sequences. A set of ASMs, each represents an acoustic pattern, is automatically learned for each category of emotion and autism spectrum disorders. Then based on the ASM sets, an input utterance is decoded into series of ASM units, with which the system determines the category for that utterance. The final output of the system is an ensemble of the machine learning and ASM techniques.

Below the machine learning and ASM techniques will be described respectively in Sections 2 and 3. The experimental results are in Section 4, and in Section 5 the conclusions and future work will be given.


2.1. Support Vector Machine (SVM)

For Arousal, Valence and Typicality tasks, ordinary binary SVMs are used. On the other hand, for Diagnosis and 12-way Emotion, which are multi-class classification tasks, we adopt a direct approach for training multi-class SVM [17]. In this approach, with Y categories provided, the system would have a set of weight vectors \( \{ w_1, \ldots, w_y \} \), each corresponds to one category. An input utterance \( x \), represented as a fixed-
length feature vector \( f(x) \), will be classified as class \( \hat{y} \) whose corresponding weight vector \( w_y \), maximizing the inner product of \( w_y \) and \( f(x) \):

\[
\hat{y} = \text{arg} \max_y w_y \cdot f(x).
\]  

(1)

Given a set of training examples \( \{x_n, y_n\}_{n=1}^{N} \), in which \( x_n \) is an utterance, \( y_n \) the category of \( x_n \), and \( N \) the number of examples, all the weight vectors \( \{w_1, \ldots, w_Y, \ldots, w_Y\} \) are jointly learned by solving the following equation:

\[
\min_{\{w_1, \ldots, w_Y\}} \frac{1}{2} \sum_{n=1}^{N} ||w_{y_n}||^2 + C \sum_{n=1}^{N} \epsilon_n,
\]

s.t. \( \forall n = 1, \ldots, N, w_y \neq y_n : \)

\[
w_{y_n} \cdot f(x_n) - w_y \cdot f(x_n) \geq \delta(y_n, y) - \epsilon_n, \quad \epsilon_n \geq 0.
\]

(2)

The constraints in (2) require that for each training utterance \( x_n \), the distances between \( w_{y_n} \cdot f(x_n) \) and \( w_y \cdot f(x_n) \) are larger than a margin \( \delta(y_n, y) \). The margin \( \delta(y_n, y) \) of two categories \( y_n \) and \( y \) is defined by prior knowledge. Here small margins are given to the categories that have some common properties because the feature vectors of the utterances in these categories would inherently be close. For example, in I2-way Emotion task, the two emotion categories with the same dimensions of valence and arousal are given smaller margin than those having different valence and arousal. Each constraint is padded with a per-example slack variable \( \epsilon_n \) whose sum over the training set is minimized. The norm of the parameters to be learned and the scale of the slack variables are traded off with a parameter \( C \) just like ordinary SVM.

2.2. Deep Neural Network (DNN)

DNN [18] is a feed-forward artificial neural network model, which consists of multiple layers of neurons. The neurons in each layer are fully connected to the neurons in the next layer. DNN maps the input feature vector of an utterance into the posterior probability of each category, and the utterance is considered as belonging to the category with the highest posterior probability.

With a set of training data, the parameters in DNN are first initialized by restricted Boltzmann machines (RBM), and then the back-propagation algorithm fine-tunes those parameters. When DNN is trained on a small training set, it typically performs poorly on test data. This “over-fitting” is greatly reduced by randomly “dropout” some hidden units at the feed-forward phase of back-propagation because each neuron is forced to learn effectively due to the “unreliability” of other neurons. With the “dropout” technique, even though the testing data is mismatched to the training data or corrupted by noises, DNN can still provide very effective performance. The “dropout” technique has been verified to give big improvements on many benchmark tasks including speech and object recognition [19].

2.3. Weighted Discrete K-nearest Neighbours (WD-KNN)

In WD-KNN [20], the system first computes the Euclidean distance between the feature of input utterance \( x \) and all the utterances in the training data. Then for each category \( y \), the system finds the \( K \) utterances belonging to \( y \), which are nearest to the input utterance \( x \). The distance of an input utterance \( x \) and a category \( y \) is thus defined:

\[
D(x, y) = \sum_{k=1}^{K} a_k d(x, x_k^y),
\]

(3)

where \( x_k^y \) is the \( k \)-th nearest utterance in category \( y \), \( d(x, x_k^y) \) is the Euclidean distance between the feature vectors of \( x \) and \( x_k^y \), and \( \{a_1, \ldots, a_K\} \) is a set of weights rescaling the distances, which satisfies \( a_1 \geq a_2 \geq \ldots \geq a_K \). An input utterance \( x \) is classified into the class \( \hat{y} \), which has minimum \( D(x, y) \).

3. Acoustic Segment Model (ASM) Approach

ASM [21, 22, 23] and other unsupervised acoustic pattern discovery approaches have been successfully utilized for enhancing speaker recognition [24, 25], spoken document classification [26, 27, 28, 29], spoken document retrieval [30], spoken term detection [31, 32], and music retrieval [33]. In this paper, we propose to use the ASM approach to incorporate temporal information in acoustic feature sequences to determine the category of speech. Here each ASM is characterized by an HMM consisting of a sequence of states representing a phone-like acoustic pattern. It is assumed that each category of speech has its own specific characteristics, which can be described by a set of ASMs. Therefore, \( Y \) sets of ASMs are learned from the speech of \( Y \) different categories. During classification, these \( Y \) sets of ASMs are adopted to determine the category of the input utterance.

The training procedure of ASM includes two stages: initialization and model training. First in the initialization stage, the system performs an even segmentation on all of the training utterances. Other segmentation approaches are also feasible, such as maximum likelihood segmentation [34], finding the spectral discontinuities [27], or using watershed transform over the blurred self similarity dotplot [35]. The acoustic features in each segment are averaged to represent the segment. Then, K-means clustering method is used to cluster all the segments based on their averaged acoustic features. After the clustering, the segments in the same cluster are regarded as belonging to the same ASM, and thus the training utterances have initial ASM unit transcriptions \( W_0 \). Then in the model training stage, the system learns the ASMs by iteratively refining the ASMs’ parameters and the ASM unit transcriptions. At the \( i \)-th iteration, the following two steps are conducted:

1. Learn a set of ASMs \( \theta_i \) maximizing the likelihoods of the ASM unit transcriptions \( W_{i-1} \) obtained in the last iterations. This is done by using Baum-Welch algorithm.

2. Use the ASM set \( \theta_i \) obtained in the last step to find the most probable ASM unit transcriptions \( W_i \) via Viterbi algorithm.

Following the above procedures, the system learns a set of ASMs for each category. During classification, the system uses the ASM sets of all the categories to decode the input utterance by Viterbi algorithm. The utterance is classified into the category whose ASM set has the maximum likelihood among all the categories.

Because some particular categories may have only a small amount of training data, a direct training procedure for ASM can cause over-fitting. To avoid the issue, we further propose a two-phase training procedure for ASM: (1) We follow the initialization and model training stages described in the last paragraph to build ASM universal background models (ASM-UBM) using the entire set of training utterances. Because the utterances in all categories are used for training, the ASM-UBM describes the overall acoustic patterns of human speech. (2)
Table 1: Results of Emotion and Autism Sub-Challenges in terms of unweighed average recall (UAR) on the development set and the testing set.

<table>
<thead>
<tr>
<th>Development</th>
<th>Test</th>
<th>(1) Emotion Sub-Challenge</th>
<th>(2) Autism Sub-Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UAR</td>
<td>12-way Emotion</td>
<td>Typicality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Arousal</td>
<td>Valence</td>
</tr>
<tr>
<td>SVM</td>
<td>82.4%</td>
<td>77.9%</td>
<td>40.1%</td>
</tr>
<tr>
<td>DNN</td>
<td>83.2%</td>
<td>76.9%</td>
<td>45.0%</td>
</tr>
<tr>
<td>WD-KNN</td>
<td>82.4%</td>
<td>52.4%</td>
<td>40.1%</td>
</tr>
<tr>
<td>ASM</td>
<td>82.4%</td>
<td>52.4%</td>
<td>40.1%</td>
</tr>
<tr>
<td></td>
<td>(D-1)</td>
<td>ASM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(D-2)</td>
<td>UBM-ASM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.0%</td>
<td>61.6%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>73.0%</td>
<td>63.5%</td>
</tr>
</tbody>
</table>

Table 2: Multi-class SVM based on the one-against-one approach and the direct approach with uniform and non-uniform margins for 12-way Emotion and Diagnosis tasks on the development set.

<table>
<thead>
<tr>
<th>UAR</th>
<th>One-against-one</th>
<th>Direct Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Uniform</td>
<td>Non-uniform</td>
</tr>
<tr>
<td>12-way Emotion</td>
<td>41.1%</td>
<td>43.9%</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>55.7%</td>
<td>55.6%</td>
</tr>
<tr>
<td></td>
<td>36.2%</td>
<td></td>
</tr>
</tbody>
</table>

4. Experiments

Table 1 shows the results of Emotion and Autism Sub-Challenges in terms of unweighed average recall (UAR) on the development set (labelled Development) and the testing set (labelled Test). Part (1) is the results for Emotion Sub-Challenge, in which the three columns report the results of Arousal, Valence and 12-way Emotion tasks. Part (2) is for Autism Sub-Challenge, and the two columns in part (2) are respectively for Typicality and Diagnosis tasks. To cope with imbalanced class distribution in Autism Sub-Challenge, the under-presented categories were up-sampled in the experiments.

The challenge baselines provided by the organizer are also included in Table 1 [16]. In the challenge baselines, the openSMILE toolkit extracted fixed-length features consisting of acoustic characteristics for utterances, and WEKA data mining toolkit classified the features by SVM. Exactly the same openSMILE features were used by the machine learning techniques described in Section 2.

4.1. Support Vector Machine (SVM)

This subsection reports the experimental results based on SVM. In all of the experiments in this subsection, we chose the trade-off parameter $C \in \{10^0, 10^1, \ldots, 10^{-6}\}$ for SVM that achieved the best UAR on the development set. For Diagnosis task, our multi-class SVM did not directly classify the input utterance into four categories. Instead, a binary SVM was first used to classify an utterance into typical or atypical, and then a three-class SVM further classified the utterances of atypical speakers into the different named categories.

In Table 2, we compare the multi-class SVM based on mainstream one-against-one approach (in the column labelled One-against-one) and our direct approach described in Section 2.1 (in the column labelled Direct). The two rows labelled 12-way Emotion and Diagnosis are respectively for 12-way Emotion task in Emotion Sub-Challenge and Diagnosis task in Autism Sub-Challenge. In the one-against-one approach, suppose there are $Y$ classes, $Y$ ($Y - 1$) independent binary SVM classifiers are trained for all pairs of categories. The category of a testing object is thus determined by the majority vote of the $Y$ ($Y - 1$) binary SVMs. Here libSVM [37] was used to implement the one-against-one approach, while the direct approach in our system was implemented by SVMstruct [38, 39]. In Table 2, we further investigate the performance of the direct approach with uniform margins (in the column labelled Uniform) and non-uniform margins (in the column labelled Non-uniform). For uniform margins, $\delta(y_n, y)$ in (2) were always set to be 1. On the other hand, for non-uniform margins, $\delta(y_n, y)$ was decided by the properties of the two categories $y_n$ and $y$ based on our prior knowledge. In 12-way Emotion, the two classes with the same dimensions of arousal and valence were given margin 1 (e.g. amusement and elation had margin 1). If two classes only had the same dimensions of arousal or valence, they would have margin 1.25 (e.g. cold anger vs hot anger), and the classes with totally different dimensions of valence and arousal would be given margin 1.5 (e.g. amusement vs sadness). For non-uniform margins used in Diagnosis task, all the category pairs had margin 1, except that the category related to dysphasia (language impairment) was assigned margin 1.5 with other types of autism disorders. This is because we consider that dysphasia has its own special manifestations in speech, which can be easily identified from other kinds of Autism spectrum disorders.

For 12-way Emotion, we found that direct approach outperformed the one-against-one approach even with uniform margins (Uniform vs One-against-one in the row labelled 12-way Emotion). This is because the direct approach learned all parameters for multi-class SVM jointly, but the one-against-one approach considered them independently. Moreover, the non-uniform margins which incorporated the prior knowledge into classification provided further improvement over the uniform margins (Uniform vs Non-uniform in the row labelled 12-way Emotion). For Diagnosis task, the one-against-one approach and the direct approach with uniform margins were just comparable (Uniform vs one-against-one in the row labelled Diagnosis). Because there are only three categories in Diagnosis task, learning the parameters independently did not have too much difference from learning them jointly. Nevertheless, with non-uniform margins, the direct approach remarkably outperformed the one-against-one approach (Uniform vs Non-uniform in the
row labelled Diagnosis).
All results on the development set based on SVM are reported in the row (A) of Table 1. The results of 12-way Emotion and Diagnosis tasks are exactly the ones reported in the column labelled Non-uniform in Table 2. In these tasks, due to the direct approach and non-uniform margins, our system remarkably outperformed the challenge baselines on the development set (row (A) vs Challenge Baseline in the part labelled Development and the columns labelled 12-way Emotion and Diagnosis). The binary SVMs used in Typicality, Arousal and Valence were all implemented by LibSVM. Compared with the baseline results on the development set, our system obtained outperformed results on Typicality and Arousal but underperformed results on Valence. This may be because different toolkits were used for implementing SVMs, and the parameters C had different values.

4.2. Deep Neutral Network (DNN)
The DNNs used here had two hidden layers. The first hidden layer had 4,000 neurons, and the second ones had 2,000. RBMs trained by 100 iterations were used to initialize the DNNs, and there were 50 iterations for back-propagation. We modified DeepLearnToolbox [40] to implement these training algorithms. Part (B) in Table 1 shows the results of DNNs on the development set. Row (B-1) reports the results of standard DNNs without “dropout” during training. We found that DNNs without dropout only slightly outperformed the baselines in Emotion Sub-Challenge and did not observe any improvements over the baselines in Autism Sub-Challenge on the development set (row (B-1) vs Challenge Baselines in the part labelled Development). Due to the small amount of training data available in the tasks, standard DNNs over-fitted to the training data. Thus, their power was degraded. Row (B-2) shows the results with 50% dropout. On the development set, the performances of DNNs were dramatically improved by the dropout technique (rows (B-2) vs (B-1)), and DNNs with dropout remarkably outperformed the baselines in all of the tasks (row (B-2) vs Challenge Baselines in the part labelled Development).

4.3. Weighted Discrete K-nearest Neighbour (WD-KNN)
For WD-KNN, the weights \( w_{K} \) to \( w_{1} \) were Fibonacci sequence, which empirically yielded better performance [20]. To obtain better performance, not all of the feature components were used in computing the Euclidean distances. The feature components involved were selected by the development set. The results of WD-KNN on the development set are reported in the row (C) of Table 1. We found that the results of WD-KNN were not comparable with the challenge baselines on the development set in Valence and Typicality tasks.

4.4. Acoustic Segment Model (ASM) Approach
Instead of transforming an utterance into a fixed-length feature vector, the ASM approach directly considered the utterances’ acoustic feature sequence. The acoustic feature sequences used here were MFCC sequences. The results of the ASM approach on the development set are shown in the part (D) of Table 1. Row (D-1) is the results of direct training, where the ASMs for each category were learned independently. Row (D-2) is the results of two-phase training, where a set of ASM-UBM is first trained on the training data of all categories and then adapted to category-specific ASM sets. We found that on the development set the two-phase training outperformed the direct training in most tasks (rows (D-2) vs. (D-1)). This verifies that the two-phase training can effectively handle the over-fitting issue. Notably the two-phase training resembles the speaker and acoustic environment adaptation techniques that have been widely used in speech recognition. A seed (speaker- or environment-independent) model was first established to cover the entire acoustic space, and then model adaptation approaches were applied on the seed model to obtain speaker- or environment-specific models [41, 42]. Although the performances of ASMs were not comparable with the challenge baselines and other machine learning techniques, it is still appealing because it takes the advantages of temporal information in speech and provides complementary knowledge for other subsystems in the integrated system.

4.5. Ensemble System
Finally, the results from the four subsystems were integrated. Here each subsystem generated its estimated posterior probabilities to all of the categories. For the SVM-based subsystem, in binary classification tasks, the posterior probabilities were estimated by the algorithm built in libSVM [37]; whereas in multi-class classification tasks, the class \( \hat{y} \) with the highest \( w_{y} \cdot f(x) \) had posterior probability 1, and others 0. The output of a DNN is intrinsically the posterior probabilities of the categories. For the subsystems based on KNN and ASM, the class \( \hat{y} \) with the smallest \( D(x, y) \) in (3) or the class whose ASMs having the highest likelihood was assigned posterior probability 1, and others 0. The final decision score for each category was the weighted sum of the posterior probabilities from the four subsystems. The weights for the subsystems were determined by the development set. The category with the highest decision score is the output of the ensemble system. The results of the ensemble system on the development set are reported in the row labelled “Ensemble” and the part labelled Development in Table 1. Our final ensemble system remarkably outperformed the challenge baselines on the development set in all tasks (Ensemble vs Challenge Baselines in the part labelled Development in Table 1). However, the final ensemble system did not outperform the challenge baselines on the testing set in the Valence task of Emotion Sub-Challenge and the Diagnosis task of Autism Sub-Challenge (Ensemble vs Challenge Baselines in the part labelled Test in Table 1). Because in these tasks the UAR of the baselines on the development and testing set were very different, it seems that the development set and the testing set have remarkably different properties in these tasks. Therefore, the final ensemble system which may over-fitted for the development set does not guarantee to yield good results in these tasks.

5. Conclusions and Future Work
In this paper, we report the results of an ensemble system based on the integration of multiple well-known machine learning techniques and ASM approach. The improvement of the proposed ASM technique is on progress. We plan to use language models to enhance the ASM approach. We are also investigating the performance of transforming the UBM to category-specific ASMs using better adaptation techniques, which can more efficiently learn new models with limited amount of data (especially for the 12-way Emotion classification task).
6. References


