Fusion Model for Speech Emotion Recognition with Low Level Descriptor Features

Cheng Chang[1,2,3,4], Huifeng Zhang[1,2,3,4], Zhangxuan Gu[4] and Yanmin Qian[1,2,3,4]
1. Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
2. SpeechLab, Department of Computer Science and Engineering
3. Brain Science and Technology Research Center
4. Shanghai Jiao Tong University, Shanghai, China

Abstract：Speech emotion recognition is one of the most challenging speech processing tasks, with many applications in the field of Human-Machine Interaction (HMI). Traditional works have been using Gaussian Mixture Models (GMM) for classification. In recent researches, Deep Neural Networks (DNN) have shown strong ability in feature learning and modeling in many tasks. In this paper, we present to utilize the DNN to learn extra features from Low Level Descriptor (LLD) features for other classifiers, and propose a decision-level fusion model for speech emotion recognition. We carry out experiments and evaluations on a public German corpus called FAU-AIBO with five emotions. And our experimental results using LLD features demonstrate that our proposed approach improves the recognition performance on unweighted results and outperforms the baseline using GMM significantly.

Keywords：speech emotion recognition; deep neural network; low level descriptor; fusion
中图分类号： TN912.33；TN912.16

In speech enabled Human-Machine Interfaces (HMI), the context knowledge plays an important role in improving the system. The emotion recognition technology can detect the emotion in speakers' voice, an important kind of context knowledge, to make it possible for HMI to respond naturally with proper emotion during interaction[1]. Its potential applications in daily scenarios and healthcare have attracted great research interests.

Generally, there are two main issues for developing a speech emotion recognition system. One is to find a set of efficient acoustic features which are able to well represent the emotional state. The other is to build an accurate and effective classifier that can utilize the extracted features to distinguish between different emotions.

For the first issue, some previous works have described the use of pitch, intensity, speaking rate, and voice quality features[1], some have studied the application of spectral features including Linear Prediction Cepstrum Coefficients (LPCC), Mel Frequency Cepstrum Coefficients (MFCC), and Perceptual Linear Prediction (PLP)[2]. In this paper, we will make use of Low Level Descriptor (LLD) features, which is a kind of utterance-level fusion feature composed of several different acoustic characteristics [3].

As for the second issue, one kind of approach is to generate the distribution of each emotional state by using features directly, such as Gaussian Mixture Model (GMM) [4] or Hidden Markov Model (HMM)[5], another is to apply statistical functions to the low-level features to obtain the global characteristic of each utterance and then construct discriminative classifiers, such as Support Vector Machine (SVM) [6], Linear Discriminative Analysis (LDA) and so on. And in recent researches, Deep Neural Networks (DNN) have shown strong ability in feature learning and modeling and can be applied to emotion classification[2][7]. However, the ability of a single model is limited, especially when the emotions in the corpus are biased. How to make use of different models to improve the whole performance is quite a challenge for the emotion recognition task.

In this paper, we mainly concentrate on the second issue and study how to classify the speech emotions with LLD features. Our experiments are carried out on the FAU Aibo Emotion Corpus

(FAU-AIBO)[8][9], which contains five emotions with quite biased distribution. We utilize the MLP to learn extra features from LLD features for SVM classifiers, and propose a decision-level fusion model for speech emotion recognition to take the advantage of different models to achieve better performance.

This paper is organized as follows. The next section introduces the related works on this task. Section 2 talks about the proposed fusion models in details. Section 3 describes experiments, including the corpus, feature extraction and experimental results of different models, followed by our conclusions and future works in section 4.

1 Related Works

Based on the LLD features, there are lots of methods that can be applied in this task. Since our proposed model is a fusion model, our work is based on a set of algorithms.

1.1 KNN

Let (X, Y) be the features-labels pair, then the squared error loss is $L\big(Y, f(X)\big) = (Y\text{-}f(X))^2$. This leads us to a criterion for choosing f,

$$EPE(f) = E(Y\text{-}f(X))^2 = \int (y\text{-}f(x))^2 \Pr(dx, dy)$$

By conditioning on X, we can rewrite EPE as

$$EPE(f) = E_X E_{Y|X}([Y\text{-}f(X)]^2|X)$$

and we can find that it suffices to minimize EPE pointwise:

$$f(x) = argmin_x E_{Y|X}([Y\text{-}c]^2|X = x)$$

The solution is

$$f(x) = E(Y|X = x)$$

The algorithm of K-Nearest-Neighbor (KNN) attempts to directly implement this recipe using the training data. Suppose that the data are fitted as $Y = f(X) + \varepsilon$, with $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$. Then the expected prediction error at x_0 is:

$$EPE_k(x_0) = \sigma^2 + \left[Bias^2(\bar{f})_k(x_0) + Var(\bar{f})_k(x_0)\right]$$

$$= \sigma^2 + [f(x_0)\text{-}\frac{1}{k}\sum_{l=1}^{k} f(x_l)]^2 + \frac{\sigma^2}{k}$$

So as k varies, there is a bias-variance tradeoff. To optimize the expected prediction error, we use cross validation to find a well-performed value of k.

1.2 SVM

Though the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. It was proposed that the original finite-dimensional space can be mapped into a much higher-dimensional space, where the data can be separated more easily. So the Support Vector Machine (SVM) method constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space[11]. Since in general, the larger the margin the lower the generalization error of the classifier, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin).

In this paper, we use SVMlib[12] as the tool for SVM-based models. And we found that the default settings of it ruin the result. After cross validation experiments, we finally choose RBF kernel and set γ for a RBF-SVM classifier to $\frac{1}{k}$, where k is the class number, that means, $\gamma = 0.2$.

1.3 LDA

The between-class variance of $Z = a^T X$ is $a^T B a$ which means the within-class variance is $a^T W a$ and B is the covariance matrix of the class centroid matrix M. Linear Discriminant Analysis (LDA) aims to maximize the Rayleigh quotient,

$$\max_a \frac{a^T B a}{a^T W a}$$

This is a generalized eigenvalue problem with a_1 given by the largest eigenvalue of $W^{-1}B$. Since it is a multi-class problem, we can find a_2 as the second largest eigenvalue , orthogonal to a_1.

As LDA gets the best result for single models (with default parameters), we naturally think that the training data probably be similar to Gaussian distribution.

1.4 AdaBoost

Adaptive boosing (AdaBoost) algorithm is a kind of “boosting” method, which is trained as following steps:

1. First train a weak classifier, like decision tree, with all the samples sharing same weight D. Then compare the predictions with labels and get an error named ε.
2. Change the weight of weak classifiers with ε.

$$\alpha = \frac{1}{2}\ln(\frac{1-\varepsilon}{\varepsilon})$$

3. Change the weight of training samples D with α if the sample is predicting the right

answer：

$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{-\alpha}}{Sum(D)}$$

Else if the prediction is wrong:

$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{\alpha}}{Sum(D)}$$

4. Return to 1. and train the classifiers with different samples' weight D.

When it comes to testing step, evaluate the test samples with all the weak classifiers and then use the sum of $output*\alpha$ as the whole model's prediction.

1.5 MLP

A MultiLayer Perceptron (MLP) is a feedforward artificial neural network model which maps the set of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation as the training method, meanwhile, it is a modification of the standard linear perceptron and can distinguish data that are not linearly separable. In conclusion, MLP is a simple neural network with often two layers of neurons.

1.6 MLP-SVM

For many tasks, MLP can be a kind of feature extractor and can be combined with other models to improve the ability of the whole classifiers.

For example, we can use the output layer of MLP as extra features and concat them with original features to construct more powerful features. Then the resulted features can be fed into SVM classifies and improve the performance of SVM models.

1.7 SMOTE

A dataset is unbalanced if the distribution of sample categories is not approximate uniform distribution. In this case, it is often helpful to augment dataset. Synthetic Minority Over-sampling Technique (SMOTE) is a powerful method that has shown a great deal of success in various applications[10].

It is an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples. It generates synthetic examples in a less application-specific manner, by operating in "feature space" rather than "data space". The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending on the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.

2 Decision-Level Fusion Model

Generally, a speech emotion recognition system is composed of four parts: data preprocessing, feature extraction, feature transforming and emotion classification. The model we propose in this paper is a fusion model which classify the data by voting of multiple well-performed sub-classifiers. We extract 384-dimensional LLD features in our experiments which are normalized before all other data processing. Since the data distribution is rather biased, in some of our classifiers, we do the re-sampling on the generalized LLD features before training. In this way, we hope to get benefits from all these classifiers to achieve better performance. The whole system structure is shown in Figure 1.

Firstly, we train many simple single classifier models (k-nearest-neighbors, SVM, Decision Tree, Random Forest, MLP, AdaBoost, Naive Bayes, QDA, LDA) to find the potentially effective ones. And then we choose the better ones among them as sub-classifiers in our fusion model. As described above, an important issue of this task is unbalance of data, so we did upsampling using SMOTE and do the experiments again, then get the corresponding results to evaluate the performance.

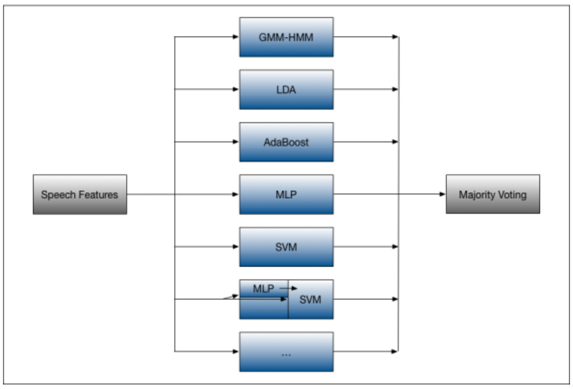

Figure 1 general structure of fusion model

Meanwhile, according to our prior knowledge, SVM and MLP are good at such low-dimension task.

So we may get good results if we fine tune them.

The MLPs are trained with adam algorithm to change learning rate automatically. The activation layers are all relu layers. And the codes are written using MXNET[7]. And from the experiments in next section we find out that for this task, deeper structures resulted in worse performances, SMOTE up-sampling is helpful in most settings. So the chosen parameters of the MLP sub-classifier in our final model is two hidden layer with the size of 512 and 5 individually with SMOTE up-sampling as preprocessing.

In the earlier stage of tuning SVM, different kernels are tried with other parameters fixed, and results show that RBF kernel performs best and cost the lowest time in this task. Both decision function of shape (one-vs-one and one-vs-rest) are tried and the results show that one-vs-rest is better. So the latter experiments on SVM are all based on RBF kernel and one-vs-rest decision function of shape.

Later, we tried the MLP-SVM model and find it practical to be a sub-classifier in our fusion model.

As a result, The chosen sub-classifiers include GMM-HMM, LDA, AdaBoost, MLP, SVM and MLP-SVM, and their parameters are all fine tuned. After generating the sub-classifiers, to combine them, for each input, we do the “voting” for each classifier and find out the most “voted” label as the output of the fusion model which is shown in Figure 1.

By using this method, we find it helpful to eliminate the errors of single models. The best results achieved by the fusion model is shown in table 9 in the next section. We can see that the result is way better than single classifiers.

3 Experiment

3.1 Experimental Setup

The corpus used in this paper is FAU-AIBO, which contains 5 different emotions: **Anger(A)**, **Empathic(E)**, **Neutral(N)**, **Positive(P)** and **Rest(R)**. The data was collected at two different schools, MONT and OHM, from 51 children (age 10-13, 21 male, 30 female; about 9.2 hours of speech without pauses). And to guarantee speaker independence, the data of one school (OHM, 13 male, 13 female) is used just for training and the data of the other school (MONT, 8 male, 17 female) just for testing. Table 1 shows the distribution of each label in the corpus. As the data is rather biased, for example, the percentage of emotion **N** is 56.1% while emotion **P** is 6.8%, how to deal with the unbalance is the key in this task.

Table 1 FAU-AIBO corpus data distribution

Label	**Train**	**Test**	**Sum**
A	881	611	1492
E	2093	1508	8257
N	5590	5377	10967
P	674	215	889
R	721	546	1267
Sum	9959	8257	18216

The LLD feature extracted in our experiments follow the instruction by Schuller et al.[9]. It’s basically a fusion of many ordinary features, such as Mel frequency cepstrum coefficients (MFCC), zero crossing rate(ZCR), pitch, Harmonics to Noise (HNR), Root mean square(RMS). An utterance level feature is then calculated over all samples by computing 12 different reduction methods, the details are depicted in table 2. Thus, the total feature vector for one utterance contains $16 * 2 * 12 = 384$ attributes.

Table 2 FAU-AIBO corpus data description

LLD(16*2)	**Functionals(12)**
(Δ)ZCR (Δ)RMS Energy (Δ)F0 (Δ)HNR (Δ)MFCC1-12	mean standard deviation kurtosis, skewness extremes: value, rel.position, range linear regression: offset, slope, MSE

The baseline of the task is constructed using LLD features as input to train HMMs as the classifier (classification by linear left-right HMM, one model per emotion, diverse number of states, 2 Gaussian mixtures, 6+4 Baum-Welch re-estimation iterations, Viterbi)[9]. To deal with the problem of unbalance, the baseline used SMOTE[10] to do up-sampling. The baseline results are shown in Table 3. The results are unweighted average (UA) of five emotions.

Table 3 baseline result on FAU-AIBO

Precision	**Recall**
0.296	0.355

3.2 Experiments of single models

For the first step, we train single classifier models to compare the results and aim to find the well-performed ones. Basically, we applied 9 models, and the unweighted results we got with different classifiers are listed in Table 4.

Table 4 results of different classifiers

Classifier	Precision	Recall	F1-score
KNN	0.223	0.225	0.214
SVM	0.130	0.200	0.158
Decision Tree	0.321	0.244	0.241
Random Forest	0.250	0.200	0.158
MLP	0.318	0.273	0.226
AdaBoost	0.284	0.308	0.282
Na ïve Bayes	0.289	0.310	0.171
LDA	0.360	0.338	0.340

Since there is an unbalance problem in the dataset, we do the up-sampling using SMOTE and get the corresponding results in Table 5.

Table 5 results of different classifiers after SMOTE

Classifier	Precision	Recall	F1-score
KNN	0.213	0.224	0.150
SVM	0.130	0.200	0.158
Decision Tree	0.291	0.327	0.221
Random Forest	0.272	0.357	0.224
MLP	0.326	0.317	0.184
AdaBoost	0.299	0.375	0.292
Na ïve Bayes	0.266	0.320	0.190
LDA	0.319	0.406	0.306

The results show that SMOTE up-sampling is helpful for some classifiers but not for all. From Table 4 we choose better performed LDA, AdaBoost as “voters”, besides we also include the baseline model GMMHMM into our final fusion model.

Then we train the MLP and SVM separately to evaluate the result of each classifier. The results we got while fine tuning MLP is listed in Table 6 and Table 7. Table 6 shows that too large (more than two) hidden layers are not helpful for this task.

Table 6 results of MLP

Hidden Layer Setting	Precision	Recall	F1-score
64 - 5	0.302	0.284	0.286
128 - 5	0.332	0.313	0.319
256 - 5	0.341	0.320	0.324
512 - 5	0.346	0.330	0.332
1024 - 5	0.357	0.321	0.330
128 - 128 - 5	0.305	0.318	0.306
256 - 256 - 5	0.306	0.313	0.305
512 - 512 - 5	0.316	0.308	0.308
1024 - 1024 - 5	0.320	0.311	0.311

Based on the results in Table 6, we implemented MLP with SMOTE, shown in Table 7. It can be seen that SMOTE only works when hidden layer size is small. So considering the results in Table 6 and Table 7, we choose “512 - 5” MLP without SMOTE as one of the sub-classifies of the fusion model.

Table 7 results of MLP with SMOTE

Hidden Layer Setting	Precision	Recall	F1-score
64 - 5	0.320	0.337	0.306
128 - 5	0.321	0.340	0.323
256 - 5	0.331	0.330	0.323
512 - 5	0.340	0.332	0.329
1024 – 5	0.337	0.324	0.323

Since the original SVM performs badly, we use MLP to extract extra features as described in Section 1.6. In experiments, we use a “256 - 128” MLP to extract 128-dimensional extra features, resulted in a 512-dimensional feature for SVM. And to make the SVM model more suitable to classify the unbalanced data, we apply different weight to data of different categories.

Table 8 shows some of the tuning results of SVM. The value “pow2.0” of “class weight” means that the weights of different classes are set to the element-wise 2.0 power of { A: 6.35, E: 2.67, N: 1, P: 8.3, R: 7.75} (this indicates the ratio of different calss samples), or more specifically, {A: $6.35^{2.0}$, E: $2.67^{2.0}$, N: $1^{2.0}$, P: $8.3^{2.0}$, R: $7.75^{2.0}$}. The meanings of “pow1.0”, “pow0.7” etc. are similar.

Table 8 results of SVM

Class weight	Precision	Recall	F1-score
(default)	0.496	0.284	0.298
pow0.33	0.434	0.330	0.340
pow0.4	0.411	0.342	0.349
pow0.5	0.404	0.358	0.360
pow0.6	0.381	0.368	0.364
pow0.7	0.364	0.370	0.361
pow1.0	0.348	0.386	0.360
pow2.0	0.339	0.387	0.350

The results of SVM after SMOTE are worse and not shown in the table. The result in table 5 also shows that SMOTE is not helpful for SVM. The reason may lie in that the generated data by SMOTE confuses the margin of support vectors and causes the original data to be classified wrongly by SVM.

Generally, by applying different weights in SVM, the results are quite promising. And we choose the best “pow0.6” model as one of the sub-classifies of the fusion model.

3.3 Experiments of fusion model

As a result, we choose the better performed and fine tuned five classifiers (GMM-HMM, LDA, AdaBoost, MLP, and SVM) to be the “voter” in our fusion model. After testing the fusion model, the result is listed in Table 9. The best unweighted result given by the proposed model achieved 0.384, 0.383, 0.377 of precision, recall, F1-score individually, which improved by 29.7% on accuracy and 7.9% on recall compared to the GMM-HMM based system used as a baseline.

Table 9 results of fusion model

Precision	**Recall**	**F1-score**
0.384	0.383	0.377

And the confusion matrix of this result is shown in Figure 2. From the matrix, we can see that the precision and recall of minority class 3 and class 4 are very low, which is a result of the confusion with majority class 2.

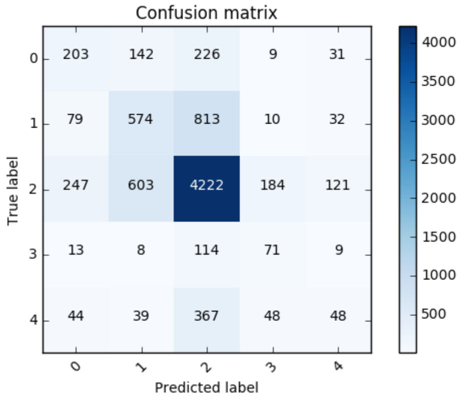

Figure 2 confusion matrix of fusion model

4 Conclusions and Future Works

In this paper, we propose a fusion model to handle the emotion detection challenge task which lets multiple classifiers to “vote” for the result after training them separately using only LLD features. And our experimental results demonstrate that our proposed approach improves the performance in different emotions steadily.

Though our system achieved a better result than the baseline, there are still lots of things can be done to improve the results. For example, the frame level MFCC features can be used individually or combined with LLD features. And more complex models like RNN, LSTM, CNN can be used to enhance the ability of the models.

5 Acknowledgement

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References

[1] Vogt T, André E, Wagner J. Automatic recognition of emotions from speech: a review of the literature and recommendations for practical realisation[M]//Affect and emotion in human-computer interaction. Springer Berlin Heidelberg, 2008: 75-91.

[2] Koolagudi S G, Rao K S. Emotion recognition from speech: a review[J]. International journal of speech technology, 2012, 15(2): 99-117.

[3] Banse R, Scherer K R. Acoustic profiles in vocal emotion expression[J]. Journal of personality and social psychology, 1996, 70(3): 614.

[4] Neiberg D, Elenius K, Karlsson I, et al. Emotion recognition in spontaneous speech[C]//Proceedings of fonetik. 2006: 101-104.

[5] Nwe T L, Foo S W, De Silva L C. Speech emotion recognition using hidden Markov models[J]. Speech communication, 2003, 41(4): 603-623.

[6] Mower E, Mataric M J, Narayanan S. A framework for automatic human emotion classification using emotion profiles[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2011, 19(5): 1057-1070.

[7] Stuhlsatz A, Meyer C, Eyben F, et al. Deep neural networks for acoustic emotion recognition: raising the benchmarks[C]//Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011: 5688-5691.

[8] Steidl S. Automatic classification of emotion related user states in spontaneous children's speech[M]. Erlangen, Germany: University of Erlangen-Nuremberg, 2009.

[9] Schuller B W, Steidl S, Batliner A. The INTERSPEECH 2009 emotion challenge[C]//Interspeech. 2009, 2009: 312-315.

[10] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[11] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.

[12] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.