Emotion Recognition Using Support Vector Machine and Deep Neural Network

Chen Ruinian, Zhou Ying, Qian Yanmin
Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China

Abstract: We propose a classification system which is a combination of three subsystems to solve the speech emotion recognition problem. The first subsystem utilizes support vector machines (SVMs) to classify the features directly. The second subsystem utilizes deep neural networks (DNNs) to classify the features directly. In the third subsystem, we utilize DNNs to extract high level features from raw data and show that they are effective for speech emotion recognition. We first produce an emotion state probability distribution for each speech segment using DNNs, then we construct utterance-level features from segment-level probability distributions. These utterance level features are fed into an SVM to identify utterance-level emotions. The experimental results show that all the subsystems are outperform the HMM baseline system, and the combined system get the best result on F-score.

Key words: Emotion recognition; Deep neural networks; Support vector machine

1 Introduction

Despite the remarkable progress made in artificial intelligence recently, the human-machine interaction remains a challenging field. A speech message in which people express ideas or communicate has a lot of information that is interpreted implicitly. Much of the implied information can be acquired by recognizing the emotion of speech. Thus, speech emotion recognition, which plays an important role in human-machine interaction, has been widely studied.

Speech emotion recognition can be treated as a classification problem on sequences since it aims to firstly extract the effective features from speech and then determine the emotion status given global statistical features or sequential local features. Lots of work has been done on speech emotion recognition. Some used Gauss Mixture Models (GMMs) and Hidden Markov Models (HMMs) to learn the distribution of low-level acoustic features \(^1\), such as pitch-related features, energy-related features, Mel frequency cepstrum coefficients (MFCCs), etc. Some studies take low-level features as input and used Support Vector Machines (SVMs) \(^2\), deep neural networks (DNNs) or other machine learning methods \(^3\) for classification. Some other work using CNNs and RNNs to perform end-to-end speech emotion recognition \(^4,5\).

SVMs classify data through determination of a set of support vectors by minimizing the structural risk which reduces the average error of the inputs and their target vectors. These support vectors are members of the set of training inputs, and outline a hyperplane in feature space which defines the boundary between the different classes. This technique has successfully been applied to standard classification tasks, such as text classification and medical diagnosis.

A deep neural network (DNN) is a feed-forward neural network which has more than one hidden layers between its inputs and outputs. It is capable of learning high-level representation from the low-level features and can classify data effectively. With sufficient training data and appropriate training strategies, DNNs...
perform very well in many machine learning tasks (e.g., speech recognition [7]), it can be seen as a high-level feature extraction method [8,9] and also a great classifier.

In this paper, we introduce three subsystems with features of different levels and different classification methods, and combine them into a fusion system by voting mechanism. The first subsystem, which would be further referred as LLD-SVM, takes low-level descriptors (LLD) as features and uses Support Vector Machine for classification. The second subsystem, LLD-DNN, uses LLD features as well, but utilizes deep neural network as emotion recognizer. The performance of those two are complementary, thus we want to build a system to take the advantage from both LLD-SVM and LLD-DNN subsystem. To this end, we build the third subsystem, also referred as DNN-SVM[10]. We utilize the idea of [10], extract segment-level features to train a DNN, which then predicts the segment-level probabilities of each emotion state. We generate utterance-level features from the statistics of segment-level probabilities, and feed those features into a SVM classifier for emotion recognition of the utterance. The difference between [10] and our third subsystem is that we use SVMs as the utterance-level emotion recognition instead of the ELM(extreme learning machine) used in [10]. The combined system makes use of both low-level and high-level features to get the best performance. Our model is training and testing on the INTERSPEECH 2009 Emotion Challenge Dataset[11].

In the next section, we describe our methods for emotion recognition, and show the experimental results in section 3, finally, we conclude the paper in section 4.

2 Methods

In this emotion recognition task, we get our best performance by combining three subsystems: LLD-SVM subsystem, LLD-DNN subsystem and DNN-SVM subsystem.

2.1 LLD-SVM Subsystem

The LLD-SVM subsystem utilizes Support Vector Machines(SVMs) as the classifier for emotion recognition at utterance level. The lld features(utterance-level) are fed into this classifier.

Support Vector Machines (SVMs) view the classification problem as a quadratic optimization problem. SVMs plot the training vectors in high-dimensional feature space, and label each vector with its class. A hyperplane is drawn between the training vectors that maximizes the distance between the different classes. Those used training vectors are called support vectors. The hyperplane is determined through a kernel function, which is given as input to the classification software. The kernel function may be linear, polynomial, radial basis, or sigmoid. The shape of the hyperplane is generated by the kernel function, though many experiments select the polynomial kernel as optimal.

SVMs can avoid the “curse of dimensionality” by placing an upper bound on the margin between the different classes, making it a practical tool for large, dynamic datasets. The feature space may even be reduced further by selecting the most distinguishing features through minimization of the feature set size.

2.2 LLD-DNN Subsystem

The LLD-DNN subsystem utilizes Deep Neural Networks(DNNs) as the classifier for emotion recognition at utterance level. The input features are the utterance level lld feature that is the same as LLD-SVM subsystem.

DNN is a method to approximate a parametric function via neural networks with many hidden layers, and it is the basis of deep learning models. A neural network can represent the function \( f(x; \theta) \) where \( x \) is the input vector and \( \theta \) is a set of parameters. For each neuron which is the smallest unit of a DNN, it maps the weighted sum of input values to an activation value via the activation function \( f_{act}(x^T w + b) \), where \( x \) is the vector of inputs for the neuron, \( w \) and \( b \) are the parameters denoted as weights and bias, respectively. The neurons in the same layer usually use the same activation function. The output of a layer is the input of the next layer. This can be considered as forward propagation of the input through the network. And we can use Back-Propagation(BP) algorithm to train it.

2.3 DNN-SVM Subsystem

![Figure 1 DNN-SVM subsystem overview.](image-url)
A deep neural network has more than one hidden layers. The features feed to the first layer can be seen as low-level features. Each higher layer could extract slightly higher-level features. To this end, in the third subsystem, instead of the lld feature, we firstly utilizes a DNN as a high-level feature extractor to obtain utterance-level features which are later fed into further classifier.

The DNN-SVM subsystem combines Support Vector Machine and Deep Neural Network these two classifiers for emotion recognition. Fig.1 shows the overview of this subsystem. We first divide the signal into segments, and then extract the segment-level features to train a DNN. The trained DNN computes the emotion state distribution for each segment. From these segment-level emotion state distributions, we constructed high-level and utterance-level features and fed them into an SVM to determine the emotional state of the whole utterance.

1) Segment-level Feature

Since DNN training needs sufficient training data, the input signals is firstly converted into frames with overlapping windows, instead of utterance-level lld features. The frame-level feature vector $z(m)$ for each frame $m$ consists of MFCC, pitch period $\tau_n(m)$, the harmonics-to-noise ratio(HNR)s, and their delta feature across time frames. The HNR is computed as:

$$
\text{HNR}(m) = 10 \log \frac{ACF(\tau_n(m))}{ACF(0) - ACF(\tau_n(m))}
$$

where $ACF(\tau)$ denotes the auto-correlation function at time $\tau$. We form the segment-level feature vector by stacking features in the neighboring frames.

$$
x(m) = [z(m-w), \ldots, z(m), \ldots, z(m+w)]
$$

where $w$ is the window size on each side.

It is reasonable to assume that the segments with highest energy contain most important emotional information since not all segments in an utterance contain emotional information, we only choose segments with high energy in an utterance as the training samples which depend on a threshold parameter. In the test phase, we also use those segments with high energy with the same threshold to be consistent with the training phase.

2) DNN Training

For the segment-level emotion recognition, we train a DNN to predict the probabilities of each emotion state. The DNN can be treated as a segment-level emotion recognizer. The input to the recognizer is the segment-level feature and the training target is the label of the utterance, which means we assign the same label to all the segments in one utterance.

The number of input units of the DNN is consistent with the segment-level feature vector size. It uses a softmax output layer whose size is set to the number of possible emotions $K$. The trained DNN aims to produce a probability distribution $t$ over all the emotion states for each segment:

$$
t = [P(E_1), \ldots, P(E_k)]^T
$$

3) Utterance-level Features

Since it is not necessary true that the emotion states in all segments is identical to that of the whole utterance, we need a higher level classifier to guarantee our classification result. From DNN, we have obtained high-level abstraction of the segment information. We can form the emotion recognition problem as a sequence classification problem, given the segment information, we need to make decision for the whole utterance. Thus we utilizes the statistics of segment information to form whole utterance features.

The features in the utterance-level classification are computed from statistics of the segment-level probabilities. Specifically, let $P_s(E_k)$ denote the probability of the $k$th emotion for the segment $s$. We compute the feature for the utterance $i$ for all $k = 1, \ldots, K$.

$$
\begin{align*}
    f^1_k &= \max_{s \in U} P_s(E_k), \\
    f^2_k &= \min_{s \in U} P_s(E_k), \\
    f^3_k &= \frac{1}{|U|} \sum_{s \in U} P_s(E_k), \\
    f^4_k &= \frac{|P_s(E_k) > \theta|}{|U|}
\end{align*}
$$

where $U$ denotes the set of all segments used in the segment-level classification. The features $f^1_k, f^2_k, f^3_k$ correspond to the maximal, minimal and mean of segment-level probability of the $k$th emotion over the utterance, respectively. The feature $f^4_k$ is the percentage of segments which have high probability of emotion $k$. This feature is not sensitive to the threshold $\theta$, which can be empirically chosen from a development set.

4) SVM for Utterance-level Classification
Since the SVM is the most commonly used classifier for global features, we use a SVM as the utterance-level classifier, which is almost the same as the first subsystem except the input features. The utterance-level statistical features computed from statistics of the segment-level probabilities are fed into a classifier for emotion recognition of the utterance.

3 Experiments

3.1 Experiment Settings

1) Dataset

We use FAU Aibo Emotion Corpus as our evaluation dataset, which has 5 hours utterance for training and 4 hours utterance for test. The utterances are categorized into 5 classes: Anger, Emphatic, Neutral, Positive and Rest. The frequencies for the five-class are given in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>E</th>
<th>N</th>
<th>P</th>
<th>R</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>881</td>
<td>2093</td>
<td>5590</td>
<td>674</td>
<td>741</td>
<td>9959</td>
</tr>
<tr>
<td>test</td>
<td>661</td>
<td>1508</td>
<td>5377</td>
<td>215</td>
<td>546</td>
<td>8257</td>
</tr>
<tr>
<td>sum</td>
<td>1492</td>
<td>3601</td>
<td>10967</td>
<td>889</td>
<td>1267</td>
<td>18216</td>
</tr>
</tbody>
</table>

2) Feature Extraction

The FAU Aibo Emotion Corpus offer the lld feature for classification. In detail, the 16 low-level descriptors chosen are:

- **ZCR** zero-crossing-rate, 1 dimension.
- **RMS** root mean square, 1 dimension.
- **HNR** harmonics-to-noise ratio, 1 dimension.
- **F0** 1 dimension.
- **MFCC** 12 dimension.

To each of these, the delta coefficients are additionally computed with 12 functionals. Thus, the total feature vector per utterance contains $16 \times 2 \times 12 = 384$ attributes.

In the DNN-SVM subsystem we use the same settings as [10]. Firstly, we extract the frame-level MFCC feature, using a 25-ms window sliding at 10-ms each time. The size of the segment level feature is set to 25 frames, including 12 frames each side. In addition, 10% segments with the highest energy in an utterance are used in the training and test phase. The threshold in equation 7 is set to 0.2. We use the rbf kernel for SVM, the kernel coefficient gamma is set to $1 / n$ features.

3) Feature Preprocessing

Although the FAU Aibo Emotion Corpus offer the lld feature for classification, for better performance, we need to do some preprocessing on it:

- **Rescale**: We find that the scale of each dimension is totally different, it is important to scale all the dimension to the same range. So we rescale all dimension to range $[0, 1]$.

- **Clip**: After rescale, there is another issue that some dimensions have outliers. For example, in the 182th dimension, the mean value of this dimension is 0.0008, but the max value of this dimension is 1, which means most of the value in this dimension is really small but there are some points take big value. Hence we do data clip on all dimension to remove those outliers.

**Normalization**: Finally, we do normalization on all dimension.

4) Model Parameters

The dnn in LLD-DNN subsystem have 3 hidden layers, the size of each hidden layer is 512. The dnn in DNN-SVM subsystem have 5 hidden layers, the size of each hidden layer is 1024. The training algorithm for dnn is mini-batch sgd, the size of mini-batch is set to 128, learning rate is 0.1, momentum is 0.9, clip gradient is 3, weight decay is 0.0001. The activation function is ReLU, we also add batch normalization layer and dropout layer to those dnn. The dropout ratio is set to 0.5. All those parameters are chosen from develop set.

3.2 Results

Since the dataset have 5 classes, the final result is based on the unweighted average recall and average precision. The association also provide a HMM baseline for this task. The final results is in table 2. Raw means using raw lld feature, without any preprocessing. The experimental results shows that, there are significant improvement on those performance indexes for both LLD-SVM and LLD-DNN subsystems by applying feature preprocessing.
Comparing to the LLD-DNN subsystem, LLD-SVM subsystem have a better performance in recall but poorer precision rate; On the contrary, the LLD-DNN subsystem have a higher precision rate but lower recall rate.

<table>
<thead>
<tr>
<th></th>
<th>ave-precision</th>
<th>ave-recall</th>
<th>ave-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM(baseline)</td>
<td>0.296</td>
<td>0.355</td>
<td>-</td>
</tr>
<tr>
<td>lld-svm (raw)</td>
<td>0.264</td>
<td>0.201</td>
<td>0.161</td>
</tr>
<tr>
<td>lld-svm (raw)</td>
<td>0.004</td>
<td>0.200</td>
<td>0.062</td>
</tr>
<tr>
<td>lld-svm</td>
<td>0.340</td>
<td>0.427</td>
<td>0.354</td>
</tr>
<tr>
<td>lld-svm</td>
<td>0.420</td>
<td>0.356</td>
<td>0.356</td>
</tr>
<tr>
<td>dnn-svm</td>
<td>0.314</td>
<td>0.363</td>
<td>0.316</td>
</tr>
<tr>
<td>combine</td>
<td>0.366</td>
<td>0.386</td>
<td>0.370</td>
</tr>
</tbody>
</table>

LLD-SVM subsystem is higher than the LLD-DNN subsystem on all labels except the majority label 'N'. In the figure 2, we can observe the symmetric phenomenon that, the LLD-DNN subsystem has the higher precision rate on minority labels but LLD-SVM subsystem has the higher precision rate on majority label. That is to say the two subsystems are complementary, hence we want to build a system to take the advantage from both LLD-DNN and LLD-SVM system. To this end, we build the third DNN-SVM subsystem. Although the performance of DNN-SVM subsystem is poorer than those two, we combine all the 3 system by voting mechanism. The combined system have a promising performance on both precision and recall, and take the best performance in F-score.

4 Conclusion

Emotion recognition is becoming more and more popular in many of the research fields recently. It is important for a human-machine interaction system to track the emotion state of users.

In this work, we propose a combined system for emotion recognition. Firstly we trained 3 subsystems, all of those three subsystems are outperform the HMM baseline. The experimental results indicate that the performance of LLD-SVM subsystem and LLD-DNN subsystem are complementary. We utilize the advantages of those subsystems by combining them using voting mechanism. The combined system have a promising performance on both precision and recall, and take the best performance in F-score.

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.

Reference


