Agglomerative Hierarchical Clustering of Emotions in Speech Based on Subjective Relative Similarity

Ryoichi Takashima1, Tohru Nagano2, Ryuki Tachibana2, Masafumi Nishimura2

1Kobe University, 2IBM Research - Tokyo
takashima@me.cs.scitec.kobe-u.ac.jp, {tohru3,ryuki,nisimura}@jp.ibm.com

Abstract

When we humans are asked whether or not the emotions in two speech samples are in the same category, the judgment depends on the size of the target category. Hierarchical clustering is a suitable technique for simulating such perceptions by humans of relative similarities of the emotions in speech. For better reflection of subjective similarities in clustering results, we have devised a method of hierarchical clustering that uses a new type of relative similarity data based on tagging the most similar pair in sets of three samples. This type of data allows us to create a closed-loop algorithm for feature weight learning that uses the clustering performance as the objective function. When classifying the utterances of a specific sentence in Japanese recorded at a real call center, the method reduced the errors by 15.2%.

Index Terms: Semi-supervised clustering, relative similarity data, emotion recognition, feature selection.

1. Introduction

Analysis of the emotions in calls is a desirable capability for speech analytics systems for call centers. With utterances automatically classified into categories, a user of the system will be able to quickly focus on the transitions of the emotions of the speakers. Since obvious discharges of emotions such as angry shouts or weeping are rarely heard at real call centers, recognizing subtle emotional categories, including attitudes and intentions, is necessary for emotion classification techniques to be useful for call centers. But how many categories should the utterances be classified into? For example, two utterances classified into the same “calm and negative” category can be then further classified by more careful listening into two different categories, a “thinking” category and a “mildly expressing dissatisfaction” category. To simulate such multi-level perceptions by humans of similar emotions, we used hierarchical clustering to represent the emotional categories of speeches.

To reflect subjective similarities in clustering results, the clustering parameters or feature weights are usually learned with manually labeled data. Constraints expressed by Must-Link (ML) and Cannot-Link (CL) data pairs [1], or relative similarity data in the form of “X is closer to A than X is to B” is sometimes used for the learning [2]. In our work, we found many problems with these types of subjective similarity data for hierarchical clustering and therefore devised a new type of relative similarity data based on flagging the most similar pair in sets of three data samples. This type of relative similarity data allows (1) data collection without determining the number of output clusters, and (2) the consistent use of the same type of data for both feature weight learning and hierarchical clustering. The consistency advantage also supports a closed-loop algorithm for feature weight learning that incorporates evaluations of the clustering performance. The algorithm also allows the use of different weights at different layers of the hierarchical clustering. We show the effectiveness of the algorithm by the results of experiments classifying the utterances of a specific sentence of a female Japanese speaker at a real call center.

2. Related Work

Our ultimate goal is automatic recognition of the emotions of a speaker from the person’s speech. Though it is known that linguistic features such as the spoken words and their context, as well as the acoustic features, are useful clues for recognizing emotions, we focus our current analysis on the utterances of a specific and frequently used Japanese sentence, “Sou desu ne” (meaning “That’s right”). We concentrate on the acoustic features of that sentence in this research.

Recently, acoustic features for the recognition of emotions have been intensively studied. One project [3] reported on the prosodic features for disambiguating the functions of “yeah” in conversations. A study [4] showed the efficiency of spectral features such as MFCC and Mel band energies. While we consider these findings and use prosodic and spectral features, finding an optimal set of features is not our main interest. It would be possible to improve the classification accuracy by using a richer feature set, including linguistic features, as in these projects.

The most studied task in the research areas related to speech emotions is probably the classification of utterances into several predefined categories such as happy, angry, or sad. In contrast, our objective is semi-supervised classification of utterances into anonymous clusters. That is related to our intended application area being speech analytics systems for call center monitoring. We believe the anonymous clusters are sufficiently powerful so that the users can tag the points in the calls when the emotions of the speaker changed. In addition, with the hierarchical clusters, it is possible to provide the users with an interactive user interface allowing the users to interactively select the granularity of the emotional categories. The reason we do not classify into predefined categories is that it is a difficult and controversial challenge to define a set of separate emotional categories actually useful in real-world speech analytics applications.

One project [5] used unsupervised clustering to recognize the emotions. The method first classifies utterances into several clusters in an unsupervised manner and then trains Support Vector Machines (SVMs) to separate each of the clusters from the others. Based on the distances from the trained hyperplanes of the SVMs, each utterance is represented by a vector called a Cluster-Profile (CP). The emotion of an utterance is determined by the classification of the corresponding CP. Though the final output of the proposed method is large categories such as angry and happy, unsupervised clustering into anonymous clusters is
used as a tool to represent the subtle emotional characteristics of the utterances.

In general, to bias the clustering results to simulate the subjective similarity as perceived by humans, we must train the clustering parameters or feature weights based on manually labeled data. In the field of speech emotion recognition, classification algorithms such as SVMs or preprocessing for feature selection often do this training [6]. In the broader research field of clustering, there are two common approaches for reflecting subjective similarity into the clustering results, constraint-based supervised clustering and metric-based supervised clustering.

In constraint-based supervised clustering [1], the constraints within the training data are expressed by Must-Link (ML) and Cannot-Link (CL) data pairs. The ML constraints specify that two data samples have to be classified in the same cluster, while CL constraints specify that two data samples cannot be in the same cluster. The approach typically adjusts clustering parameters to satisfy the constraints. A problem when applying this approach to hierarchical clustering is that we need to determine the number of clusters to output when manually preparing the constraints. A pair which is ML when using four clusters can become CL when the number of output clusters is eight. Since the number of clusters cannot be determined when the training data is prepared, this approach is not suitable for hierarchical clustering.

The second approach is metric-based supervised clustering [2], which first trains the feature weights based on subjective relative similarity data and then does unsupervised clustering with the weighted features. The relative similarity data, which we call XAB data, gives information in the form of “X is closer to A than X is to B” for each three-tuple, which can be written as an inequality $\text{Sim}(X, A) > \text{Sim}(X, B)$, where $\text{Sim}(X, A)$ is the similarity of X and A. This XAB data can be collected without determining the number of output clusters. However, XAB data is inconvenient when used for evaluating the accuracy of hierarchical clustering, because the hierarchical clustering results for data samples including X, A, and B may resemble Figure 1(a), which means we cannot tell whether the $(X, A)$ pair or the $(X, B)$ pair is the more similar pair from the clustering results. This problem not only makes a large percentage of the data unusable for the evaluations but also may bias the training in the wrong way. Thus, both types of relative similarity data used in the two approaches have limitations when applied to hierarchical clustering, which blocks consistent optimization throughout the process flow of hierarchical clustering from the feature weight learning to the clustering result evaluation.

3. Proposed Method
3.1. Main Idea

Our main idea is to use a new type of relative similarity data for both the training and evaluation of a semi-supervised clustering algorithm. This enables incorporation of the clustering in the closed loop of feature weight learning so that the feature weights are optimized directly using the clustering results as the objective function. In addition, we can learn different sets of weights for different layers of the clustering hierarchy.

The key to the proposed method is our new type of relative similarity data, which we call ABC data. A set of this data is collected by having human labelers listen to sets of three data samples, A, B, and C, and asking the labelers to mark the most similar pair, such as $(A, B)$, among the three possible pairings. The task is equivalent to asking the labelers to find the most dissimilar data, C in this example, among the three. The workload for the task is comparable to the workload for marking the XAB relative similarity data. This procedure provides two relative similarity inequalities for each triplet, such as $\text{Sim}(A, B) > \text{Sim}(B, C)$ and $\text{Sim}(A, B) > \text{Sim}(C, A)$. Since the information is a superset of the information provided by the XAB data, we can use any metric-based clustering algorithms that require XAB data with ABC data. We can also use ABC data for evaluation of an obtained clustering hierarchy by comparing it to human-labeled ABC data. When evaluating a hierarchy, we can use ABC data without producing any unusable data, in contrast to the XAB data approach, as long as the hierarchy is produced by a hierarchical clustering algorithm that merges or splits one node at a time. In the example shown in Figure 1(b), the clustering hierarchy is incorrect for the example of A, B, and C because B and C meet at a node at a lower level than the node for A and B. Since the problem is selection of one pair from three pairs, the expected error value for random selection is 66.6%.

3.2. Algorithms

The proposed method consists of a learning phase, in which feature weights are learned based on labeled ABC training data, and a runtime phase, in which unsupervised agglomerative hierarchical clustering of the test data is performed by using the learned feature weights (Figure 2). The learning phase is a closed loop in each iteration in which the clustering of the labeled training data is done based on the latest feature weights, the clustering results are evaluated, and the weights are updated.

3.2.1. Unsupervised Clustering

The same agglomerative unsupervised clustering algorithm should be used in the learning phase and the runtime phase. We use Ward’s clustering method [7]. This bottom-up method starts with N (the number of training data samples) clusters of size 1 and continues until all the data are included in one cluster.
At each step of clustering, the two clusters whose fusion results in the smallest loss of information are fused. The information loss, the error of sum of squares (ESS), is defined as

\[
\text{ESS} = \sum_{j} \sum_{i \in C_j} \sum_{k} w_k |x_{i,j,k} - \mu_{j,k}|^2, \tag{1}
\]

where \( x_{i,j,k} \) is the \( k \)-th dimension of the \( i \)-th data in the \( j \)-th cluster \( C_j \), \( w_k \) is the weight for the \( k \)-th dimension, and \( \mu_{j,k} \) is the mean of the \( k \)-th dimension in the \( j \)-th cluster.

### 3.2.2. Feature Weight Learning

In the closed loop of feature weight learning, we use the weight update equations of the AdaBoost [8] algorithm with each feature as a weak classifier. Each iteration of AdaBoost selects a best feature (to be the weak classifier) from the pool of the feature indices, \( S_F = \{ f | 1 \leq f \leq F \} \). The weights of the training data, \( D_i \), are also updated depending on whether or not the latest weak classifier is correct about the \( i \)-th data. Here is the algorithm:

1. Initialize the training data weights as \( D_i = 1/N \).
2. For each iteration \( t = 1, \ldots, T \) do
   (a) For each feature \( f \in S_F \) do
      (1) Perform unsupervised clustering of the training data solely using the one-dimensional feature, \( f \).
      (2) Evaluate the clustering result by calculating the weighted error rate by using the training data weights
      \[
      \epsilon_f = \sum_i D_i I_i(fail), \tag{2}
      \]
      where \( I_i(fail) \) is an indicator function that has the value 1 if the clustering result is incorrect for the \( i \)-th data, and 0 otherwise.
   (b) Adopt the feature with the minimum error rate, \( f_t \), as the next feature.
   \[
   f_t = \arg\min_{f \in S_F} \epsilon_f, \tag{3}
   \]
   (c) Update the training data weights using
   \[
   \alpha_t = \frac{1}{2} \log \left( \frac{1 - \epsilon_{f_t}}{\epsilon_{f_t}} \right), \tag{4}
   \]
   \[
   d_i = \begin{cases} D_i \exp( +\alpha_t) & \text{(if } I_i(fail) > 0) \\ D_i \exp(-\alpha_t) & \text{(otherwise)} \end{cases}, \tag{5}
   \]
   \[
   D_i \leftarrow d_i / \sum_i d_i. \tag{6}
   \]
   Weights are increased for the data for which the clustering results are incorrect.
3. Calculate the weights for the features using
   \[
   w_f = \sum_t \alpha_t \delta_{f_t,f}, \tag{7}
   \]
   where \( \delta_{f_t,f} \) is the Kronecker’s delta, which has the value 1 if \( f_t \) is \( f \), and 0 otherwise.

### 3.3. Different Weights for Different Layers

When we asked human labelers to distinguish among the emotions in speech samples, we noticed that they focused on different features of the samples depending on the similarity of the samples. When the samples were quite different, they tended to focus solely on pitch to roughly separate them into active samples and inactive samples. In contrast, when the samples were similar, they tended to carefully examine the samples, taking into consideration various features including power and duration as well as pitch. To simulate such characteristics of the human perception of emotional speech, we use different sets of feature weights for different layers of the clustering hierarchy. Before starting the learning phase, we heuristically determined the number of layers, \( N_L \). The \( l \)-th layer (\( 1 \leq l \leq N_L \)) is defined by the lower boundary \( N_{l-1} \) and the upper boundary \( N_l (\leq N_L) \), which is the number of the clusters at the upper boundary of the \( l \)-th layer. The lower boundary of the first layer, \( N_0 \), equals the number of samples in the training data to be clustered, \( N \). A set of feature weights, \( W_l = \{ w_{k,f} | 1 \leq k \leq F \} \), is learned for each layer using this algorithm:

For each layer \( l = 1, \ldots, N_L \) do
1. Using the \( N_{l-1} \) clusters as the training data, learn a new set of weights, \( W_l \), using the procedure described in Section 3.2.2.
2. Perform clustering of the \( N_{l-1} \) clusters until the clusters are further merged into \( N_l \) clusters using the obtained \( W_l \).

### 4. Experiments

#### 4.1. Corpus

We evaluated the effectiveness of the proposed method with an emotion-clustering task for the utterances of a female speaker using a particular sentence. The utterances were excerpts from a recording of a call at a real-life Japanese call center. The recording was 8 kHz stereo, with the utterances of the agent recorded in the left channel and the customer in the right channel. The sentence we focused upon was “Sou desu ne?”, meaning “That’s right” in Japanese. This sentence can transmit various attitudes and intentions depending on how the speaker uses the sentence. We extracted from the 40-minute call 67 utterances of “Sou desu ne?” by the female agent, each of which was separated by silences from surrounding utterances. We used 33 utterances for learning and 34 utterances for testing. We prepared the ABC relative similarity data by asking 3 labelers to mark the most similar pair for each tuple consisting of three randomly selected samples. Each of the 3 labelers marked 100 tuples for learning and 250 tuples for testing.
4.2. Features

We manually separated each utterance of “Sou desu ne” into the three parts of “Sou”, “desu”, and “ne” and extracted 16 features: duration for the three parts and the entire utterance, and mean pitch, delta pitch, mean power, delta power, mean Harmonics-to-Noise Ratio (HNR), and delta HNR for the “Sou” and “ne” parts. The reason the features other than the duration was not extracted for the “desu” parts was that they include unvoiced regions where the six other features cannot be reliably measured. HNR is a degree of periodicity calculated based on the autocorrelation of the speech [9].

4.3. Compared Methods

We compared the following methods of feature selection and weighting. The same set of 16 feature candidates and the same unsupervised clustering algorithm were used for all of these methods.

Baseline Unsupervised clustering with neither feature selection nor weighting. The features were only normalized to make the mean 0 and the variance 1.

PCA Principal Component Analysis. The transformation matrix that converts the feature vectors of the training data to the principal components was learned in the learning phase. At runtime, the feature vectors of the test data were first transformed by the matrix and then were classified by the unsupervised clustering algorithm. The ABC relative similarity labels were not used.

FS Forward Selection method. This is a simplest feature selection method that uses the ABC data in a closed loop. The algorithm starts with zero-dimensional feature vectors. It increases the dimension of the feature vector by one dimension in each iteration by selecting an additional feature from the feature candidates. The feature that most improves the clustering performance is selected. The iterations end when none of the remaining features improves the performance.

FS(L) An FS method with two layers ($N_2 = 2$). The layer boundary was at $N_1 = 5$.

Proposed The proposed method. A common set of feature weights was used for the entire hierarchy.

Proposed(L) The proposed method with three sets of feature weights ($N_2 = 3$). The layer boundaries were at $N_1 = 7$ and $N_2 = 3$.

4.4. Results

The experimental results are shown in Table 1 and Table 2. Each number is the ratio of the number of the correct tuples over 750 tuples. A tuple was counted as a correct tuple when the manually labeled pair was judged by the clustering result as the most similar one among the three. In Table 1, we can see the Proposed method that learned feature weights based on the ABC data reduced the error rate by 6.2% relative to PCA that did not use the ABC data. Though FS also used the ABC data, it actually increased the error rate.

Table 2 shows the results of the methods with multiple sets of feature weights. By comparing Table 1 and Table 2, we can see the use of different weights for different layers significantly improved the performances both for FS and Proposed, by efficiently simulating the human perception of emotional similarity. Proposed(L) reduced the error rate by 9.6% relative to Proposed and by 15.2% relative to PCA. The improvements were significant ($P < 0.03$, Chi-square test).

Table 1: Clustering accuracies.

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>PCA</th>
<th>FS</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error rate</td>
<td>40.3 %</td>
<td>38.8 %</td>
<td>41.5 %</td>
<td>36.4 %</td>
</tr>
</tbody>
</table>

Table 2: Clustering accuracies for the methods with multiple layers.

<table>
<thead>
<tr>
<th>Method</th>
<th>FS(L)</th>
<th>Proposed(L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error rate</td>
<td>35.3 %</td>
<td>32.9 %</td>
</tr>
</tbody>
</table>

5. Conclusions

In this paper, we proposed ABC relative similarity data, which is well suited for hierarchical clustering algorithms because it can be collected without determining the number of output clusters and because it can be used for both of learning and training in a consistent manner. Our algorithm for semi-supervised clustering of emotional speech is designed to take advantage of this new type of data to learn the feature weights by a closed-loop algorithm that incorporates unsupervised clustering and its evaluation. We showed the effectiveness of the data and the algorithm by the experimental results in clustering the utterances of a female speaker using a particular sentence. Our future work includes speaker-independent clustering of arbitrary utterances.

6. References


