Emotional Content Comparison in Speech Signal Using Feature Embedding

Stefano Rovetta, Zied Mnasri, and Francesco Masulli

Abstract Expressive speech processing has been improved in the recent years. However, it is still hard to detect emotion change in the same speech signal or to compare emotional content of a pair of speech signals, especially using unlabeled data. Therefore, feature embedding has been used in this work to enhance emotional content comparison for pairs of speech signals, cast as a classification task. Actually, feature embedding was proved to reduce the dimensionality and the intra-feature variance in the input space. Besides, deep autoencoders have recently been used as a feature embedding tool in several applications, such as image, gene and chemical data classification. In this work, a deep autoencoder is used for feature embedding before performing classification by vector quantization of the emotional content of pairs of speech signals. Autoencoding was performed following two schemes, for all features and for each group of features. The results show that the autoencoder succeeds (a) to reveal a more compact and a clearly separated structure of the mapped features, and (b) to improve the classification rates for the similarity/dissimilarity of all emotional content aspects that were compared, i.e neutrality, arousal and valence; in order to calculate the emotion identity metric.

S. Rovetta · Z. Mnasri (✉) · F. Masulli
DIBRIS, Università Degli Studi di Genova, Genoa, Italy
e-mail: zied.mnasri@enit.utm.tn

S. Rovetta
e-mail: stefano.rovetta@unige.it

Z. Mnasri
Electrical Engineering Department, ENIT, University Tunis El Manar, Tunis, Tunisia

F. Masulli
Sbarro Inst. for Cancer Research and Molecular Medecine, Temple University, Philadelphia, PA, USA
e-mail: francesco.masulli@unige.it

A. Esposito et al. (eds.), *Progresses in Artificial Intelligence and Neural Systems*, Smart Innovation, Systems and Technologies 184,
https://doi.org/10.1007/978-981-15-5093-5_5

1 Introduction

Speech technology is widely used in interactive applications. However, expressive speech still poses significant challenges. Emotional content analysis would be very useful for sophisticated man-machine interaction, with possible applications even beyond efficient vocal interfaces, for instance in collaborative robotics. However, detection of emotional content and its characteristics from speech signals is still inaccurate.

Since emotion recognition is a pattern recognition problem, data-driven models have been usually used for that. Indeed, supervised learning techniques like neural networks [5, 10], SVM [15], or generative models like HMM-GMM [11, 14] have been classically utilized. More recently, deep learning models have also been developed for that purpose using feedforward, recurrent or convolutional neural networks [7].

However, to analyse the emotional content in a huge database of speech signals, supervised learning would require tedious labeling, with the associated cost and the underlying risk of mistakes. Therefore, an unsupervised approach would be a suitable alternative. In this scope, there have been some successful works like in [19] where SOM were used to detect emotions from audiobooks, and in [3] where hierarchical k-means were applied to detect emotions from a corpus to build a model for expressive speech synthesis. However, to the best of our knowledge, unsupervised learning hasn't been used for emotional content comparison in speech signal so far.

Being able to compare two data items is a fundamental ability for machine learning methods ranging from clustering to kernel-based classification. Change point detection, one-class classification, novelty detection, outlier analysis, concept drift tracking are also made possible by the availability of similarity indexes. However, in the case of emotional content in speech, this task is challenging, since emotions are at an intermediate level between structural properties and semantic content.

Therefore, this work aims to find a better feature embedding which allows enhancing either clustering or classification results for emotional content comparison of speech signals. To achieve this goal, a deep autoencoder has been applied as a tool for feature embedding. More particularly, this work addresses the problem of emotional content analysis from speech independently from speaker or text. The similarity of the following expressive speech characteristics is modeled for each pair of speech signals: (a) neutrality of speech, (b) arousal and (c) valence. The input features undertake two types of preprocessing: normalization and/or embedding using the autoencoder. Finally, the results of vector quantization are aggregated to calculate a metric for emotion identity similarity.

The paper is organized as follows: Sect. 2 reviews the related work, including the standard feature sets for expressive speech analysis and the main feature extraction techniques used in expressive speech processing, Sect. 3 presents the feature embedding technique used in this work, i.e. deep autoencoder, Sect. 4 describes the speech material used in this work, whereas Sect. 5 details the experiments and discusses the obtained results.

2 Related Work

Since emotion recognition is a pattern recognition task, data-driven models have been looking for the feature set presenting the closest correlation to emotion classes. However, the usefulness of features used for emotion recognition has not been proved for emotional content comparison. Nevertheless, some combinations of speech parameters have been used for this purpose with a relative success.

Classical signal-extracted features have been proved to be extremely efficient for supervised emotion recognition, such as acoustic parameters like Mel-frequency cepstral coefficients (MFCC), prosodic parameters like fundamental frequency (F_0) and energy, or signal-related parameters like harmonic-to-noise ratio (HNR) and zero-crossing rate (ZCR). Such features, and others, have been grouped into standard feature sets for expressive speech analysis and/or recognition, such as Interspeech emotion and paralinguistic challenges [16, 17], and the Geneva minimalistic acoustic parameter set (GeMAPS) [4].

Though the aforementioned features have reached outstanding performance in emotion recognition using supervised learning, they haven't been quite efficient while using clustering techniques [13]. Besides, such an important quantity of features induces a high dimensionality of the input space. Therefore, feature extraction should be studied for such data sets in order to improve the performance.

The aim of feature analysis is to optimize the feature space so that only the most relevant features are selected or extracted. Several techniques based on ANOVA (Analysis of variance) and mutual information or cross-validation have been used for input selection, to keep only the most contributory features. An alternative way to reduce the feature space dimensionality consists in applying feature transformation methods such as PCA (Principal component analysis) and LDA (Linear discriminant analysis).

Also, to deal with feature sparseness, autoencoding neural networks were used [8]. An autoencoder is a neural network, which outputs are the same than its inputs. It is generally used to discover latent data structures in the inputs. In [2], an autoencoder was used to resolve the problem of feature transfer learning in emotion recognition. Actually, an emotion classifier trained on some kind of data, e.g. adult speech, wouldn't be efficient when tested on another kind of data, e.g. children voices. The technique consisted in applying a single autoencoder for each class of targets. The reconstructed data was then used to build the emotion recognition system.

3 Autoencoders for Feature Embedding

The autoencoder is an unsupervised learning algorithm, which is basically used for automatic extraction of features from unlabeled data. Therefore, the autoencoder can be used for feature extraction either for classification or for clustering.

3.1 Deep Autoencoder

An autoencoder is a neural network which approximates the identity function, i.e. the output is the same as the input. The autoencoder optimizes the weights (W) and the biases (b) of the neural network, such that $y_i = h_{W,b}(x_i) = x_i \ \forall x_i \in X = (x_1, x_2, ..., x_n) \subset R^n$, where x_i, y_i and $h_{W,B}$ are respectively the inputs, the outputs and the hidden layer code [9].

For real-valued data, the objective function is the mean square error $E = \sum_{i=1}^{N} ||x_i - h_{W,b}(x_i)||^2$ where $||.||$ denotes the Euclidean norm; whereas for binary data, the objective function is the cross-entropy $E = -\sum_{i=1}^{N}(x_i \log h_{W,b}(x_i) + (1 - x_i) \log(1 - h_{W,b}(x_i))$. The weights W_i and biases b_i are updated using a gradient descent algorithm, such as SGD (Stochastic gradient descent). To calculate the gradient of the objective function $J_{W,b} = (\frac{\partial E(W,b)}{\partial W}, \frac{\partial E(W,b)}{\partial b})$, the backpropagation algorithm is used [6].

The deep autoencoder is composed of two parts, namely the encoder and the decoder. Both parts consist of hidden layers, usually stacked in a mirror symmetry, with a bottleneck layer in the middle, i.e. the code layer. Then the encoded data are the output of the code layer. The usefulness of such an architecture consists in the structure of the encoded data. Actually it has been shown that the code layer can (a) reveal a hidden structure of the input features, discovered through the encoding process, (b) reduce the dimensionality of the input space. Then the encoded data will be used as an input for classifiers or clustering algorithms, in order to improve their accuracy.

3.2 Feature Embedding Using Autoencoders

Very often, original input features have a large variance between each other, which yields a complex distribution. To cope with this issue, non-linear mapping can be used to reduce the intra-feature variance in the input space. Such a mapping can be performed either by kernel methods, such as kernel k-means which applies a non-linear transformation using fixed kernel function [18], or by autoencoders, as it has been proved recently in several works [9, 18, 20, 21].

Since the autoencoder aims to learn a new representation of the input features, supplied by the code layer, then the use of a smaller number of nodes in this layer helps obtaining a new feature space with a smaller dimension. Furthermore, the new mapping often reveals a new structure where the input features (or their coded images) are grouped into compact regions, which would be more helpful for clustering tasks. However, autoencoding hasn't been widely used for clustering so far.

In [18], an autoencoder used for clustering was trained with a new objective function where the centroids are updated at the same time as the networks parameters, i.e. weights and biases. In [21], autoencoders were used for deep embedded clustering. The approach consists in a two-step algorithm by (a) initializing the parameters

with an autoencoder, (b) optimizing both the centroids and the autoencoder parameters. Optimization is performed with KL divergence as the objective function (cf. Sect. 3.1), to maximize the similarity between the distribution of the embedded features and the centroids.

4 Speech Material

This work was performed using a standard emotional speech database, i.e. EMO-DB [1], which has been widely used for emotion recognition and analysis. The feature set was selected among the standard ones (cf. Sect. 2). In particular, the Interspeech 2009 emotion challenge feature set has been proved to be highly efficient in emotion recognition.

4.1 Speech Database

EMO-DB is an acted speech database specifically designed for emotional speech processing. It has been known for providing the best emotion recognition rates using supervised classifiers such as SVM, and generative models like HMM-GMM [16]. It includes 5 short and 5 long sentences in German, uttered by 5 male and 5 female speakers. Each sentence is uttered in 7 emotional states (neutrality, anger, boredom, fear, disgust, happiness and sadness). The signals were registered at 16-KHz sampling rate in an anechoic chamber.

4.2 Feature Set

The Interpeech'2009 emotion challenge feature set, proposed by Schuller et al. [16] contains prosodic, acoustic and signal-related LLD's (low-level descriptor) required for emotion recognition. Each LLD is presented as a vector of 12 coefficients or functionals, including its most relevant statistics, calculated on the whole signal. Besides, each LLD vector is duplicated using its Δ value, i.e. temporal difference. Finally, each signal is represented by (16 LLD+ 16 Δ-LLD) $\times$ 12 functionals, thus by 384 coefficients (cf. Table 1). Then each pair of signals is represented by 768 coefficients.

Table 1 Interspeech 2009 emotion challenge LLD's and functionals [16]

Groups	LLD's	Functionals (for all LLD's)
Prosodic	(Δ) RMS energy	Min, max, range
	(Δ) ln F_0	Min rel. position, max rel. position
Signal-related	(Δ) ZCR	Kurtosis, skewness
	(Δ) HNR	Standard deviation, arithmetic mean
Spectral	(Δ) MFCC 1–12	Linear regression (offset, slope, MSE)

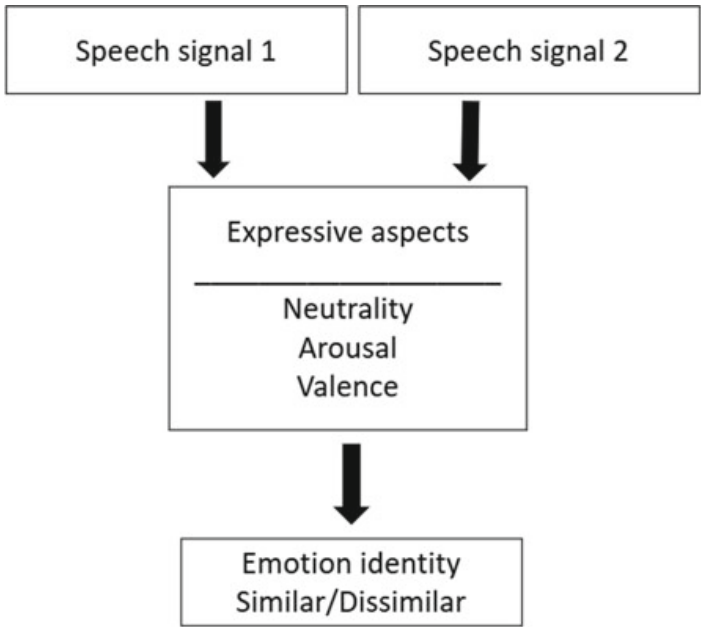

Fig. 1 Emotional content aspects and labels

4.3 Classes Related to Emotional Content

Since this work is interested in emotional content comparison, the signals of the database were grouped into pairs, of which 89386 were selected. Each pair has been assigned four labels regarding the similarity/dissimilarity of the following aspects: identity of emotions, neutrality, arousal and valence (cf. Fig. 1). It should be noted that except for valence, which is similar only for 40% of the pairs, 50% of them have similar emotions identity, neutrality and arousal.

5 Experiments

The experiments led in this work aim to evaluate (a) the effect of feature embedding on kmeans-based vector quantization of the emotional content of speech, and (b) the different ways of feature embedding using autoencoders.

5.1 Feature Embedding and Representation

Before applying vector quantization, the set of 768 features of each pair of signal (cf. Table 1) is preprocessed. Three types of preprocessing are achieved: (i) Normalization, to get zero-mean and unit-variance features, (ii) Normalization and application of the autoencoder on the whole feature set, so that the dimension of the feature vector, i.e. 768, is reduced to a lower value, (iii) Normalization and application of the autoencoder to the joint subsets of 12 coefficients, i.e. LLD, for each pair of signals (cf. Table 1). In this way, the 24-dimension of each pair of subsets is reduced to 1-dimension.

To apply the autoencoder, a deep neural network was implemented, where the output is the same as the input. The autoencoder architectures used in (ii) and (iii) are described in Table 2. In all the experiments, the training options were set as follows: ADAM optimizer, 50 epochs at maximum, a minibatch size of 32, a gradient threshold of 1, and a sigmoid transfer function.

The weights and biases of the code layer are utilized to calculate its output, using the sigmoid function. In the case of (LLD+ Δ-LLD) input features, the final embedded vector consists of the concatenated outputs of each autoencoder, i.e. a 32-coefficient vector.

The embedded data are finally represented by applying a vector quantization step [12], practically done with the kmeans algorithm using kmeans++ initialization. Classes are attributed to codevectors by majority voting. The codebook size was set to 100. The result is a smoothed-out representation of class distributions.

5.2 Feature Visualization

The autoencoder reveals the intrinsic structure of the input data, which helps in codevectors optimization. Figures 2, 3, 4, 5 show a comparison between the original input features and the autoencoded features, either all together or by LLD-group. It looks obvious that autoencoding allows visualizing (a) a compact structure, where features are more tightly distributed in the input space, (b) a clearer separation between the original classes. Therefore, the autoencoder, especially when applied to each LLD-group, seems to provide a good representation of the embedded features.

Table 2 Autoencoder architectures (layers and number of nodes)

Input features	Input layer	Hidden layer 1	Code layer	Hidden layer 3	Output layer
All features	768	500	32	500	768
LLD features	24	100	1	100	24

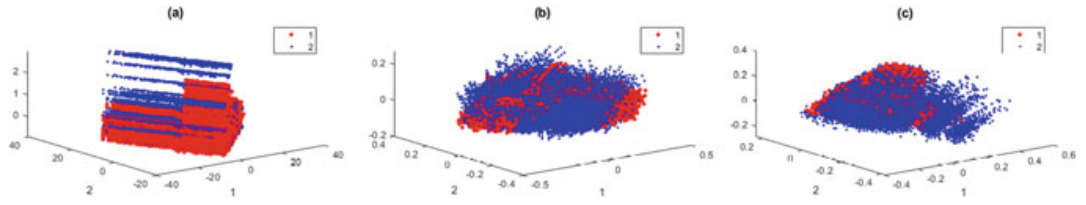

Fig. 2 Neutrality classes distribution: **a** original features, **b** autoencoded features, **c** autoencoded features by LLD-group

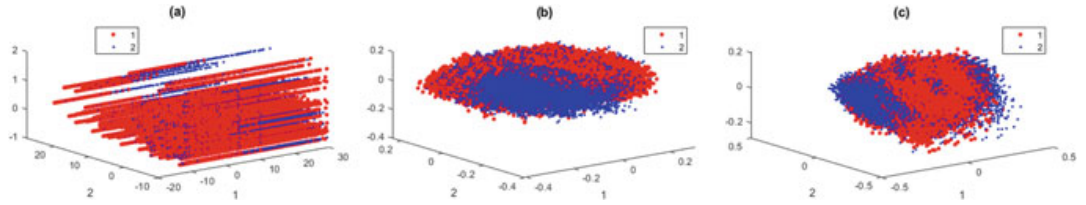

Fig. 3 Arousal classes distribution

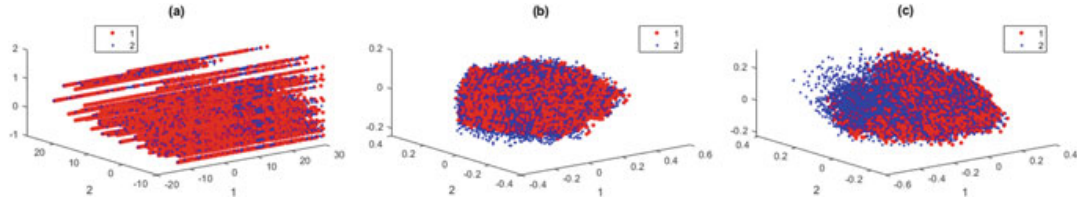

Fig. 4 Valence classes distribution

5.3 *Metric for Emotion Identity*

The three emotional aspects binary labels collected from vector quantization, i.e. neutrality, valence and arousal are aggregated to yield a metric for similarity measure of the emotion identity (μ_{id}). To calculate this metric, the mean value was used, i.e. $\mu_{id} = \frac{N+V+A}{3}$ where N, V and A are respectively neutrality, valence and arousal similarity/dissimilarity labels for each pair of signals. Then the identity metric is located into [0,1] interval.

As an application example, we cluster pairs of signals with similar emotions by using agglomerative hierarchical clustering to represent a dendrogram, where the leaves (x-axis) represents signals grouped using the distance between clusters calculated using the metric μ_{id} (cf. Fig. 5).

5.4 *Results and Discussion*

Table 3 shows the results of vector quantization using the aforementioned feature transformations (cf. Sect. 5.1). The following notes and interpretations can be drawn: (i) The effect of feature embedding on the classification results is clear, which may

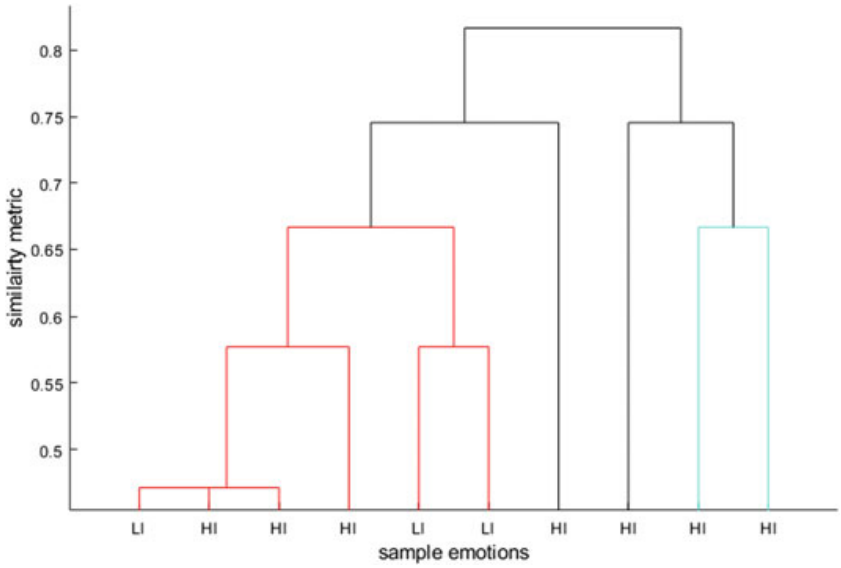

Fig. 5 Dendrogram of the hierarchical clustering of emotion identity similarity label based on the aggregated metric (HI: High-intensity emotions, LI: Low intensity emotions)

Table 3 Vector quantization accuracy using a codebook of 100 codevectors (NF: Normalized features, AF: Autoencoded features, AFL: Autoencoded features by LLD)

Expressive aspect	Total accuracy (%)			Similarity accuracy (%)			Dissimilarity accuracy (%)		
	NF	AF	AFL	NF	AF	AFL	NF	AF	AFL
Neutrality	67.9	75.2	77.5	75.3	81.0	83.9	60.5	69.3	71.1
Arousal	58.9	67.4	60.3	57.1	65.4	64.7	60.6	69.4	56.1
Valence	60.6	60.9	62.4	9.3	12.8	23.5	94.9	93.3	88.4

be explained by the improvement of features distribution, thanks to the autoencoder (cf. Figures 2, 3, 4, 5). (ii) The autoencoder applied on each LLD-group seems to improve, though slightly, the classification results. This may be due to the fact that this strategy allows selecting only one feature per LLD-group, thus avoiding redundant LLD information. (iii) The classification results using the autoencoder are more balanced between both classes, than those using only normalized data. This could be explained by the effect of compacting data, which allows detecting the codevectors more easily. (iv) Increasing the size of the codebook improves the classification results. However, it should be reasonably adapted to the number of samples, therefore we opted for a maximum of 100 codevectors for ca. 90,000 samples. (v) Using an aggregation metric for agglomerative hierarchical clustering allows grouping samples with similar emotions. However, in this case the result depends on the accuracy of vector quantization applied on the emotional aspects used to calculate the metric.

6 Conclusion

In this paper, emotional content comparison for pairs of speech signals by vector quantization using feature embedding was described. Embedding is a feature extraction technique which has been proved to enhance the learning performance. Actually, feature embedding allows reducing the input space dimensionality and the intra-feature variance. The autoencoder was used to achieve feature embedding, through the use of deep neural networks, with a bottleneck middle layer, which provides the encoded features. Hence, two types of autoencoding were applied, for all features, and for each group of features. First, the application of the autoencoder shows that the mapped features, using both schemes, have a more compact and distinguishable structure. The vector quantization results confirm this improvement, since the obtained classification rates are always better when using the autoencoder, especially when applied for each feature group. The predicted labels were aggregated to calculate a metric to compare emotion identity in speech. As an outlook, such a metric would form the basis for higher-level tasks, such as clustering utterances by emotional content, or applying kernel methods for expressive speech analysis.

Acknowledgments This work was supported by the research grant funded by "Fondi di Ricerca di Ateneo 2016" of the University of Genova.

References

1. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., & Weiss, B.: A database of German emotional speech. In: Ninth European Conference on Speech Communication and Technology (2005)
2. Deng, J., Zhang, Z., Marchi, E., Schuller, B.: Sparse autoencoder-based feature transfer learning for speech emotion recognition. In: 2013 IEEE Humaine Association Conference on Affective Computing and Intelligent Interaction, pp. 511–516 (2013)
3. Eyben, F., Buchholz, S., Braunschweiler, N., Latorre, J., Wan, V., Gales, M. J., & Knill, K.: Unsupervised clustering of emotion and voice styles for expressive TTS. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4009–4012 (2012)
4. Eyben, F., Scherer, K.R., Schuller, B.W., Sundberg, J., Andre, E., Busso, C., Truong, K.P.: The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. IEEE Trans. Affect. Comput. **7**(2), 190–202 (2016)
5. Hozjan, V., Kacic, Z.: Context-independent multilingual emotion recognition from speech signals. Int. J. Speech Technol. **6**(3), 311–320 (2003)
6. Huang, P., Huang, Y., Wang, W., & Wang, L.: Deep embedding network for clustering. In: IEEE 2014 22nd International Conference on Pattern Recognition, pp. 1532–1537 (2014)
7. Kim, J., Saurous, R.A.: Emotion recognition from human speech using temporal information and deep learning. Proc. Interspeech **2018**, 937–940 (2018)
8. Moneta, C., Parodi, G., Rovetta, S., Zunino, R.: Automated diagnosis and disease characterization using neural network analysis. In: Proceedings of 1992 IEEE International Conference on Systems, Man, and Cybernetics, pp. 123–128 (1992)
9. Ng, A.: Sparse autoencoder. CS294A Lecture notes. http://web.stanford.edu/class/cs294a/sparseAutoencoder2011.pdf

10. Nicholson, J., Takahashi, K., Nakatsu, R.: Emotion recognition in speech using neural networks. Neural Comput. Appl. **9**(4), 290–296 (2000)
11. Nwe, T.L., Foo, S.W., De Silva, L.C.: Speech emotion recognition using hidden Markov models. Speech Commun. **41**(4), 603–623 (2003)
12. Ridella, S., Rovetta, S., Zunino, R.: K-winner machines for pattern classification. IEEE Trans. Neural Networks **12**(2), 371–385 (2001)
13. Rovetta, S., Mnasri, Z., Masulli, F., & Cabri, A.: Emotion recognition from speech signal using fuzzy clustering. In:: EUSFLAT Conference (2019) (to appear)
14. Schuller, B., Rigoll, G., Lang, M.: Hidden Markov model-based speech emotion recognition. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing—ICASSP 2003, vol. 2, pp. II-1 (2003)
15. Schuller, B., Rigoll, G., Lang, M.: Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. I-577 (2004)
16. Schuller, B., Steidl, S., Batliner, A.: The interspeech 2009 emotion challenge. In: Tenth Annual Conference of the International Speech Communication Association (2009)
17. Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Mortillaro, M.: The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In: Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France (2013)
18. Song, C., Liu, F., Huang, Y., Wang, L., Tan, T.:Auto-encoder based data clustering. In: Iberoamerican Congress on Pattern Recognition, pp. 117–124. Springer, Berlin, Heidelberg (2013)
19. Szekely, E., Cabral, J. P., Cahill, P., Carson-Berndsen, J.: Clustering expressive speech styles in audiobooks using glottal source parameters. In: Twelfth Annual Conference of the International Speech Communication Association (2011)
20. Tian, F., Gao, B., Cui, Q., Chen, E., Liu, T.Y.: Learning deep representations for graph clustering. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
21. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016)