Long-term Statistical Feature Extraction from Speech Signal and its Application in Emotion Recognition

Erfan Loweimi, Mortaza Doulaty, Jon Barker, and Thomas Hain

Speech and Hearing Research Group (SPandH), University of Sheffield, Sheffield, UK
{e.loweimi, m.doulaty, j.barker, t.hain}@dcs.shef.ac.uk

Abstract. In this paper we propose a statistical-based parametrization framework for representing the speech through a fixed-length supervector which paves the way for capturing the long-term properties of this signal. Having a fixed-length representation for a variable-length pattern like speech which preserved the task-relevant information allows for using a wide range of powerful discriminative models which could not effectively handle the variability in the pattern length. In the proposed approach, a GMM is trained for each class and the posterior probabilities of the components of all the GMMs are computed for each data instance (frame), averaged over all utterance frames and finally stacked into a supervector. The main benefits of the proposed method are making the feature extraction task-specific, performing a remarkable dimensionality reduction and yet preserving the discriminative capability of the extracted features.This method leads to an 7.6% absolute performance improvement in comparison with the baseline system which is a GMM-based classifier and results in 87.6% accuracy in emotion recognition task. Human performance on the employed database (Berlin) is reportedly 84.3%.

Keywords: Discriminative model. Emotion recognition. Feature extraction. Generative model. Speech signal

1 Introduction

Speech is the most natural way of human communication. It reflects many aspects of us and this turns it into a complicated signal which as well as its lingual content, encodes a wide variety of information including environmental and speaker-dependent information like identity, emotional state, accent, dialect, age and health condition. These components of the speech are combined through a complicated process and disentangling such a complex signal into the aforementioned underlying dimensions is a challenging yet interesting task from both signal processing and machine learning points of view.

In this regard, the main problem is that such attributes are subjective in essence and developing an objective model or method for capturing them is difficult. In fact, other than very general clues about the aforementioned properties in the time and frequency domains, we do not have any particular extra

information to steer the hand-crafted deterministic parametrization algorithms to the right direction. As a result, in a wide range of applications in speech processing, MFCC serves as the swiss-knife army of this field and is used as the main feature representation despite the fact that it is basically proposed for speech recognition [5]. That is why the general tendency is to put the back-end at the center of attention for building a system in different applications.

Typically, pattern recognition systems consist of two main blocks, namely the front-end and back-end [6]. The front-end is tasked with extracting a representation of the data in which the task-pertinent attributes are preserved/enhanced and the irrelevant/misleading aspects of the data are filtered/weakened. This process, among other steps, requires data filtering in a very high-level domain where each attribute occupies a particular subspace. So, the front-end ideally should do *information filtering* in the *information space* and it turns out to be very challenging. The reason backs to the fact that such information space is categorically abstract and subjective. Therefore, mathematical underpinning of a mapping which takes the data from the low-level quantitative domain to such a high-level qualitative/subjective space is extremely complicated.

In this paper, we aim at enhancing the conventional feature extraction process with an interface which to some extent contributes toward conducting information filtering. This interface is a generative model which targets learning a task-dependent representation. As well, it affords further dimensionality reduction and renders a fixed-length representation for speech. This paves the way for the discriminative model employed at the back-end to return more accurate results because most of these models cannot effectively deal with the variable length patterns like speech. In the emotion recognition task, such coupling of the generative and discriminative models results in up to 7.6% performance elevation in comparison with the GMM-based classifier and leads to 87.6% accuracy which is higher than the reported human performance (84.3%) on the this task and database [3].

The rest of this paper is organized as follows. In Section 2 the main difficulties and issues in extracting hand-crafted features are reviewed and discussed. The proposed parametrization method is introduced and explained in Section 3. Experimental results are presented and analyzed in Section 4 and Section 5 concludes the paper.

2 Feature Extraction

Feature extraction (also known as parametrization or front-end) bears the task of converting the sensory data into a sequence of numbers which should preserve the relevant information in a compact way and discards the irrelevant and misleading aspects of the data. As well, the front-end should present the data in an appropriate way. Since its output serves as the input of the back-end, parametrization process output should be in harmony with the assumptions that the back-end makes about its input. For instance, a back-end with a probabilistic basis makes some statistical assumptions about the distribution of its input.

Coherency of the fed features with such expectations would substantially affect the overall performance and efficacy of the system.

Having a fixed back-end, different parametrization algorithms, could be assessed through three main criteria, namely discriminability, robustness and complexity. Discriminability is about the capability of the front-end in extracting features with both high intra-class similarity and inter-class dissimilarity. It could be evaluated by using the train data as the test data. Robustness relates to the ability of the feature in handling a reasonable amount of noise and/or mismatch between the test and train conditions. It is a challenging issue and could be assessed by using unseen/noisy data. Complexity is connected to the computational load of the feature extraction process and the lower the better.

Another important issue in feature engineering is that the output of the designed algorithm should be task-dependent because the target attribute and consequently the focus of the front-end for each application is different. That is, by setting the aim of a system to capture one of the speech properties, say speaker's emotional state, all the other elements like lingual content, speaker ID, etc. are turned into noise and should be suppressed. In practice, the ideal emotion recognition system should be able to recognize the emotion regardless of the speaker identity, lingual content and background noises.

Presence of irrelevant/misleading factors poses two main problems. First, learning the structure of the target attribute(s) under the existence of the other irrelevant attributes will be more difficult due to the clutter which they arise in the information space and it leads to hindering the learning process. Second, even if the system performs relatively well across the seen data during training, its accuracy over the unseen data would be strictly questionable. The reason backs to the fact that the misleading components would highly restrict the situation where the system performs well and the performance becomes oversensitive to any mismatch even in irrelevant traits. As a result, the generalization will be poor and any mismatch with the training condition, even in the irrelevant aspects, would noticeably degrade the accuracy and reliability of the system. To overcome these issues, the front-end should be able to filter out the misleading/irrelevant characteristics and only passes through the pertinent properties.

However, such filtering is not straightforward. As a matter of fact, it takes place in a conceptual space where each attribute presumably occupies a distinct subspace. This high-level information domain does exist based on what we subjectively perceive from the speech signal. However, expressing it in an efficient objective/mathematical way which allows for filtering nuisance characteristics and only permitting the relevant dimensions is highly challenging, if not impossible. That is why most of the researches in the pattern recognition field are focused on the back-end and less attention is paid to the front-end.

Dealing with these issues, researchers tended to develop techniques for *learning* the proper representation instead of using hand-crafted features. Currently, one of the very active branches of Machine Learning is Deep Learning via deep neural networks (DNN) which essentially solve the problem of data representation [1]. In other words, they learn a transform (or a set of transforms) that represents

the input data in the most suitable way based on the task requirement. However, for efficient training of such models which have enormous parameters, a huge amount of data and computational power are required. Although the later is no longer an impeding factor, the former is still troublesome at some fields. For example, in the task of emotion detection from speech, most of the available databases are not sufficiently big and do not allow for employing models with too many variables.

3 Proposed Method

As mentioned, feature learning under having limited data is problematic. However, due to lack in practical clues for engineering the feature extraction process for capturing the most pertinent aspects of the signal, we need to carry out a feature learning to steer the parametrization process toward a right task-dependent direction and avoid passing through irrelevant dimensions. In this section, we introduce our proposed method which serves as an interface between the conventional front-end and the classifier.

3.1 Workflow

Figure 1 shows the main parts of the proposed approach. First, each speech waveform is converted into a feature matrix, X, a $D - by - N$ matrix where D is the dimension of the feature vector and N is the number of frames of the utterance. We have used MFCC, although any feature may be utilized at this phase. Then, all the available class data is pooled for training a GMM with M *Gaussians* in order to estimate the corresponding distribution. After training a GMM for each class, the posterior probability of each component, $p(m_k|X, \theta_c)$, is computed, where m_k and θ_c denote the k^{th} Gaussian (component) and GMM's parameter set of class c, respectively.

In the next step, the posterior probabilities of each GMM are averaged over all frames of the utterance as follows

$$p(m_k|X, \theta_c) = \frac{1}{N} \sum_{n=1}^{N} log[p(m_k|x_n, \theta_c)] \tag{1}$$

where x_n represents the feature vector of the n^{th} frame of the utterance. Posterior probability can be computed based on the Bayes' rule as follows

$$p(m_k|x_n, \theta_c) = \frac{p(x_n|m_k, \theta_c)\, p(m_k|\theta_c)}{p(x_n|\theta_c)} = \frac{p(x_n|m_k, \theta_c)p(m_k|\theta_c)}{\sum_{k=1}^{M} p(x_n|m_k, \theta_c)p(m_k|\theta_c)} \tag{2}$$

where $p(x_n|m_k, \theta_c)$ is the likelihood of the x_n (n^{th} frame) given the m^{th} component of the GMM of the class c. The likelihood is computed based on the multivariate Gaussian distribution as follows

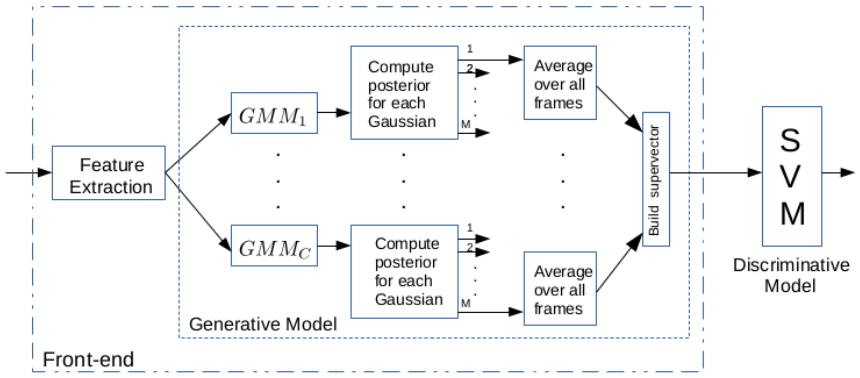

Fig. 1. Workflow of the proposed method. C and M denote number of classes and number of mixture components, respectively.

$$p(x_n|m_k,\theta_c) = \frac{\mathcal{H}_k^c}{\sqrt{(2\pi)^d}} exp(-\frac{1}{2}(x_n-\mu_k^c)^T\mathcal{H}_k^c(x_n-\mu_k^c)) \tag{3}$$

where μ_k^c and $\mathcal{H}_k^c$ denote the mean vector and precision (also known as concentration) matrix of the k^{th} component of class c, respectively.

After working out $p(m_k|x_n,\theta_c)$ for all components of all the GMMs and averaging over all utterance frames, the final feature vector is built by concatenating all the posterior probabilities in a supervector as follows

$$super\_vector = [\, p(m|X,\theta_{c_1})^T,\ p(m|X,\theta_{c_2})^T,\ ...,\ p(m|X,\theta_{c_C})^T\,] \tag{4}$$

where C denotes number of classes and

$$p(m|X,\theta_{c_i})^T = [\, p(m_1|X,\theta_{c_i}),\ p(m_2|X,\theta_{c_i}),\ ...,\ p(m_M|X,\theta_{c_i})\,]. \tag{5}$$

As a result, the speech signal will be represented by a fixed-length supervector which its length is C times M.

3.2 Advantages

As well as, representing the speech signal through a fixed-length pattern, the proposed theme has four main advantages

– First, it is no longer a general non-flexible feature extraction algorithm like MFCC. In comparison with the hand-engineered deterministic structure of the conventional front-ends, it has the spirit of the statistical feature learning paradigm. Such approach allows for learning some aspects of the data which are important for classification but we do not have any clear clue of them in order to somehow embed them in the parametrization workflow.

- Second, this approach provides an effective framework for capturing the long-term properties of the speech like emotion. From statistical standpoint, unlike the lingual content which changes on a short-term basis, the speaker-dependent attributes are fairly stationary during the utterance. As a result, it is more sensible to steer the front-end toward extracting features which reflect the long-term properties of the speech in tasks like emotion recognition. However, due to the non-stationarity of the speech and the Fourier transform limitation, we have to stick to the short-term processing. The proposed method paves the way for extracting the long-term properties of the speech from the short-term frame-based processing. This is due to the fact that the GMMs are trained based on all the frames of all the utterances of each class, without taking the timing issue into account. The underlying premise for validity of this argument is that the process to be modelled does not change in statistical term across the training data which holds with a reasonable approximation. In the second place, the supervector is the outcome of averaging over all the utterance frames. These two factors make the supervector highly correlated with the long-term properties of the signal which the GMMs are trained to capture them.
- Third benefit is further dimensionality reduction. In fact, instead of representing the speech via a matrix with D (typically 39) times N elements, the signal is represented with M times C elements which is by far more compact. As seen, the length of the supervector is no longer a function of neither the feature dimension nor the number of frames. As a result, a very lengthy and comprehensive feature set may be employed without increasing the computational load at the back-end.
- Since the supervector is built by stacking the posteriors of each class, it can be imagined that each class occupies a particular subspace in the feature space. This potentially enhances the discriminative capability of the extracted features and provides a better ground for the discriminative model to adjust the decision borders between the classes.

3.3 Comparison with UBM-GMM

Using universal background model (UBM) forms the status quo in the GMM-based feature extraction, in particular for speaker recognition [17]. In the UBM-based approach, at first, all the available training data is pooled and a shared universal model is trained for all the classes as the background. Then, for each utterance and through MAP adaptation [12], the parameters of the background model are modified and the supervector is built by stacking the mean vectors of all the Gaussians of the adapted model (Figure 2). As a result, the length of the supervector would be M times D which in comparison with the proposed method (M times C) is noticeably higher for applications where number of classes are lower than the feature vector length. Emotion and environment detection are examples of such scenarios although in the task of speaker recognition it may not be the case. One solution for this issue is to do the classification on a hierarchical basis. As such at each level, number of the categories to be classified would be

much less and this smooths the way for effective employment of the proposed method.

On the other hand, since the background model is trained with all the available data of all classes, the required number of gaussians for having a reasonable estimate of the corresponding distribution is expected to be higher than the components of a GMM which is trained for a single class. Another issue is that the adaptation process is done for each utterance, and the amount of the data provided by each signal is not enough for efficient model adaptation given the cardinality of the parameter set of the UBM. As well, the adaptation process has some hyperparameters such as *relevance factor* [17] which should be adjusted and there is no guarantee that the optimum value remains the same over all the classes and utterances. However, the proposed approach does not involve any adaptation and/or particular (hyper)parameter setting other than the number of mixture components.

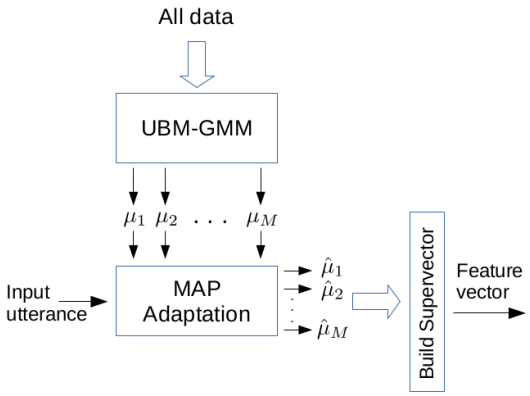

Fig. 2. Framework of the UBM-GMM method.

3.4 Dimension of the Supervector and Curse of Dimensionality

Having a long feature vector along with a probabilistic model runs the risk of facing with the curse of dimensionality. However, the issue is more manageable as far as the classification is done with a discriminative model. This backs to the fact that in such models only subspaces around the decision borders are taken into account which has a rather low volume and lack of data for covering the space is less problematic. In fact, this issue is more serious in case of working with generative models which consider the region inside each class borders because it could have a massive volume in the high-dimensional space and is really difficult to be covered by the limited available training data. As a matter of fact, for the SVM classifier (which we have used as the back-end) only the borders and particularly support vectors are concerned. As a result, the volume that should be covered would be much smaller and lack of training data and working in the high-dimensional space due to using a supervector is less an issue.

4 Experimental Results

4.1 System Setup

A wide variety of features and classifiers have been used in emotion classification. On feature side, pitch and energy (mean, max, median, variance, etc.) [21], LPCC [20], wavelet [11], sub-band filtering [9], RASTA-PLP [19] and modulation spectrum [22] have been utilized. On the back-end also a wide range of methods like GMM [2, 10], HMM [15, 18], Neural Networks (NN) [8, 14], K-nearest Neighbours (kNN) [7] and support vector machines (SVM) [4, 20] have been employed. We have used MFCC as the feature extraction algorithm along with the log-energy, delta and delta-delta coefficients. Frame length, frame shift and number of filters was set to 25 ms, 10 ms and 25, respectively, and Hamming window was applied. Window length in computing both delta and delta-delta is set to 1. It was observed that this setting for computing the dynamic coefficients returns better results in comparison with 2 for delta and delta-delta which is typically used in speech recognition setup. For classification at the back-end, SVM with RBF (radial basis function) kernel is employed. Slack variable and gamma coefficient of the kernel were set to 12 and 2, respectively.

It should be noted that the SVM could not handle variable length patterns efficiently and removing the interface block noticeably degrades its performance. In fact, without a fixed-length representation, the SVMs should be trained by each individual frame. One problem is that the amount of emotion-correlated information within a frame is not enough for effective emotion discrimination. On the the other hand, for classification, the decision should be made on a frame-wise basis and the class with maximum bincount would be the output of the classifier. This strategy is called *Max-Wins voting* and as well as the problem of making decision based on short-term observations, it suffers from the issue that all the frames either discriminative or non-discriminative would have the same weight in the voting and consequently decision making. This could negatively affect the performance of the system, however, the proposed approach does not suffer form such issues.

The GMM and SVM were trained using Scikit-learn package [16]. A GMM classifier is used as the baseline system. It includes 25 Gaussians, trained with 5 iterations based on EM algorithm and the covariance matrix is full. Publicly available Berlin emotional database (Emo-DB) [3] has been used which includes 7 acted emotions, namely Anger (A), Boredom (B), Disgust (D), Happiness (H), Fear (F), Sadness(S) and Neutral (N). It consists of 535 signals and 10 speakers (5 male and 5 female) who are professional actors read the predefined sentences in an anechoic chamber, under supervised conditions. Sampling rate of the signals is 16 kHz with a 16-bit resolution. A human perception test to recognize various emotions with 20 participants resulted in a mean accuracy of 84.3%.

4.2 Performance Evaluation

For evaluation, 5-fold cross-validation has been used. The confusion matrices of all folds were added together and from the resultant confusion matrix four

performance metrics were computed which are *Accuracy*, *Recall Rate*, *Precision* and *F-measure* (also known as $F-score$ or $F_1 score$). Assuming the rows of the confusion matrix determine the actual class and its columns show the predicted class, these measures were computed as follows

$$conf\_mat = \sum_{fold=1}^{5} conf\_mat_{fold} \tag{6}$$

$$Accuracy = \frac{\sum_{c=1}^{C} conf\_mat(c,c)}{\sum_{i=1}^{C}\sum_{j=1}^{C} conf\_mat(i,j)} \tag{7}$$

$$\begin{aligned} Recall(c) &= \frac{relevant\ class\ patterns\ \cap\ retrieved\ class\ patterns}{relevant\ class\ patterns} \\ &= \frac{TruePositive}{TruePositive\ +\ FalseNegative} = \frac{conf\_mat(c,c)}{\sum_{j=1}^{C} conf\_mat(c,j)} \end{aligned} \tag{8}$$

$$\begin{aligned} Precision(c) &= \frac{retrieved\ class\ patterns\ \cap\ relevant\ class\ patterns}{retrieved\ class\ patterns} \\ &= \frac{TruePositive}{TruePositive\ +\ FalsePositive} = \frac{conf\_mat(c,c)}{\sum_{i=1}^{C} conf\_mat(i,c)} \end{aligned} \tag{9}$$

$$F-measure(c) = 2\frac{Recall(c).Precision(c)}{Recall(c) + Precision(c)} \tag{10}$$

Accuracy is an overall performance measure and Recall Rate, Precision and consequently F-measure are defined for each class. In fact, they are originally designed for binary classification. For more details about these measures readers are referred to [13].

4.3 Results and Discussion

Table 1 and 2 show the confusion matrices of the baseline and the proposed systems, respectively. As seen, the errors which the systems make are not similar which imply that by combining these two approaches via an appropriate framework better results may be achieved. In case of the proposed method, confusion is less and the main misclassification occurs between the Happiness and Anger. This error could be alleviated by doing a hierarchical classification.

Table 3 and 4 show the performance metrics of the baseline and proposed method. As seen, the accuracy of the suggested system is noticeably higher than

the baseline. However, accuracy on its own is not enough for precise comparison of two systems and other factors such as precision and recall should be taken into account as they further clarify the type of errors which the system make. One important advantage of the proposed method is that the recall rate and precision are very closed to each other whereas in the baseline system (GMM-based classifier) the difference is noticeably higher. It should be noted that the recall and precision are inversely proportional and improving one would degrade the other. In an optimal setup they should be high and as close as possible which leads to having the maximum area under curve (AUC). This also results in having a higher F-score as it is the harmonic mean of the precision and recall. Harmonic mean is smaller than both geometric and arithmetic means and is close to the minimum of its inputs. So, as far as one of the metrics is too small, regardless of the goodness of the other one, the F-score would be poor. Therefore, the optimum performance in terms of the F-measure would be achieved if both recall and precision are high and almost equal. This is another point which shows the optimality of the proposed approach.

An important issue from practical standpoint is that how such approach can be extended to applications where there is a data stream. In such cases, the basic premise of the proposed method which was the stationarity of the speaker-dependent attribute across the signal is violated. In order to deal with this issue, one could decompose the stream into (overlapping) segments with an appropriate length. As such, instead of representing the whole utterance with a supervector, each chunk is buffered and represented by a supervector and the decision is taken locally for each sub-utterance. Some segmentation algorithms may be used depending on the task which allow for having an adaptive variable-length buffers. As well, if the task allows a finite-state machine may be employed for handling the transitions between the outputs of the system (labels of segments) using the previous local decisions (history) and some prior knowledge. This could contribute toward alleviating the errors that potentially occur in chunking the data stream (having more than one class within a segment) and improving the accuracy of the system.

Table 1. Confusion matrix of the baseline system.

	A	B	D	F	H	S	N
Anger	123	0	1	0	3	0	0
Boredom	0	61	1	0	0	5	14
Disgust	4	3	37	0	0	0	2
Fear	5	0	2	45	11	4	2
Happiness	29	0	0	2	39	0	1
Sad	0	1	0	0	0	60	1
Neutral	0	16	0	0	0	0	63

Table 2. Confusion matrix of the proposed method.

	A	B	D	F	H	S	N
Anger	113	0	1	2	11	0	0
Boredom	0	76	0	0	1	1	3
Disgust	2	2	36	1	1	1	3
Fear	2	0	0	63	0	1	3
Happiness	16	0	2	4	47	0	2
Sad	0	1	0	0	0	60	1
Neutral	0	4	0	0	1	0	74

Table 3. Performance of the baseline system.

	Recall	Precision	F-measure
Anger	96.9	76.4	85.4
Boredom	75.3	75.3	75.3
Disgust	80.4	90.2	85.1
Fear	65.2	95.7	77.6
Happiness	54.9	73.6	62.9
Sadness	96.7	87.0	91.6
Neutral	79.8	75.9	77.8
Average	78.5	82.0	79.4
Accuracy	80.0		

Table 4. Performance of the proposed method.

	Recall	Precision	F-measure
Anger	89.0	85.0	86.9
Boredom	93.8	91.6	92.7
Disgust	78.3	92.3	84.7
Fear	91.3	90.0	90.7
Happiness	66.2	77.1	71.2
Sadness	96.8	95.2	96.0
Neutral	93.7	86.1	89.7
Average	87.0	88.2	87.4
Accuracy	87.6		

5 Conclusion

In this paper, we proposed an interface block between the front-end and the back-end which is based on a generative (GMM) model and leads to a fixed-length representation for speech, filtering out the unwanted attributes, extracting a long-term feature from the signal, further dimensionality reduction and yet preserving the discriminative information. It paves the way for efficient classification through discriminative models like SVMs and results in an 7.6% absolute performance improvement. The main error which the proposed system makes during the classification is confusing the Anger and Happiness emotional states. Devising a hierarchical classification strategy could help in dealing with this issue. Using modulation spectrum, extending the feature vector with prosodic features and training the GMMs discriminatively could contribute toward improving the accuracy of the proposed method.

References

1. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1), 1–127 (2009), also published as a book. Now Publishers, 2009.
2. Bozkurt, E., Erzin, E., Erdem, A.T.: Improving automatic emotion recognition from speech signals. In: In Proc. INTERSPEECH. pp. 324–327 (2009)
3. Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., Weiss, B.: A database of german emotional speech. In: in Proceedings of Interspeech, Lissabon. pp. 1517–1520 (2005)
4. Chavhan, Y., Dhore, M.L., Yesaware, P.: Speech emotion recognition using support vector machine. International Journal of Computer Applications 1(20), 6–9 (February 2010), published By Foundation of Computer Science
5. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. Acoustics, Speech and Signal Processing, IEEE Transactions on 28(4), 357–366 (Aug 1980)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2Nd Edition). Wiley-Interscience (2000)

7. Feraru, M., Zbancioc, M.: Speech emotion recognition for srol database using weighted knn algorithm. In: Electronics, Computers and Artificial Intelligence (ECAI), 2013 International Conference on. pp. 1–4 (June 2013)
8. Han, K., Yu, D., Tashev, I.: Speech emotion recognition using deep neural network and extreme learning machine. In: Interspeech 2014 (September 2014)
9. Hosseini, Z., Ahadi, S.: A front-end for emotional speech classification based on new sub-band filters. In: Electrical Engineering (ICEE), 2015 23rd Iranian Conference on. pp. 421–425 (May 2015)
10. Hosseini, Z., Ahadi, S., Faraji, N.: Speech emotion classification via a modified gaussian mixture model approach. In: Telecommunications (IST), 2014 7th International Symposium on. pp. 487–491 (Sept 2014)
11. Krishna Kishore, K., Krishna Satish, P.: Emotion recognition in speech using mfcc and wavelet features. In: Advance Computing Conference (IACC), 2013 IEEE 3rd International. pp. 842–847 (Feb 2013)
12. Lee, C.H., Gauvain, J.L.: Speaker adaptation based on map estimation of hmm parameters. In: Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing: Speech Processing - Volume II. pp. 558–561. ICASSP'93, IEEE Computer Society, Washington, DC, USA (1993)
13. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. The MIT Press (2012)
14. Nicholson, J., Takahashi, K., Nakatsu, R.: Emotion recognition in speech using neural networks. In: Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on. vol. 2, pp. 495–501 vol.2 (1999)
15. Nwe, T.L., Foo, S.W., De Silva, L.C.: Speech emotion recognition using hidden Markov models. Speech Communication 41(4), 603–623 (Nov 2003)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
17. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. In: Digital Signal Processing. p. 2000 (2000)
18. Schuller, B., Rigoll, G., Lang, M.: Hidden markov model-based speech emotion recognition. In: Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2. pp. 401–404. ICME '03, IEEE Computer Society, Washington, DC, USA (2003)
19. Schwenker, F., Scherer, S., Magdi, Y.M., Palm, G.: The gmm-svm supervector approach for the recognition of the emotional status from speech. In: Alippi, C., Polycarpou, M.M., Panayiotou, C.G., Ellinas, G. (eds.) ICANN (1). Lecture Notes in Computer Science, vol. 5768, pp. 894–903. Springer (2009)
20. Shen, P., Changjun, Z., Chen, X.: Automatic speech emotion recognition using support vector machine. In: Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on. vol. 2, pp. 621–625 (Aug 2011)
21. Ververidis, D., Kotropoulos, C., Pitas, I.: Automatic emotional speech classification. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on. vol. 1, pp. I–593–6 vol.1 (May 2004)
22. Wu, S., Falk, T.H., Chan, W.Y.: Automatic speech emotion recognition using modulation spectral features. Speech Communication 53(5), 768–785 (2011)