S. Krajči (ed.): ITAT 2018 Proceedings, pp. 92–99
CEUR Workshop Proceedings Vol. 2203, ISSN 1613-0073, © 2018 Jiří Kožusznik, Petr Pulc, and Martin Holeňa

Sentiment Analysis from Utterances

Jiří Kožusznik[1], Petr Pulc[1,2], Martin Holeňa[2]

[1] Faculty of Information Technology, Czech Technical University, Thákurova 7, Prague,Czech Republic
[2] Institute of Computer Science, Czech Academy of Sciences, Pod vodárenskou věží 2, Prague, Czech Republic

Abstract: The recognition of emotional states in speech is starting to play an increasingly important role. However, it is a complicated process, which heavily relies on the extraction and selection of utterance features related to the emotional state of the speaker. In the reported research, MPEG-7 low level audio descriptors[10] serve as features for the recognition of emotional categories. To this end, a methodology combining MPEG-7 with several important kinds of classifiers is elaborated.

1 Introduction

The recognition of emotional states in speech is expected to play an increasingly important role in applications such as media retrieval systems, car management systems, call center applications, personal assistants and the like. In many languages it is common that the meaning of spoken words changes depending on speakers emotions, and consequently the emotional information is important in order to understand the intended meaning. Emotional Speech recognition is a complicated process. Its performance heavily relies on the extraction and selection of features related to the emotional state of the speaker in the audio signal of an utterance. For most of them, the methodology has already been implemented, and they have been experimentally tested and compared Berlin database of emotional speech.

In the reported work in progress, we use MPEG-7 low level audio descriptors[10] as features for the recognition of emotional categories. To this end, we elaborate a methodology combining MPEG-7 with several important kinds of classifiers. For most of them, the methodology has already been implemented and tested with the publicly available Berlin Database of Emotional Speech [1].

In the next section, the task of sentiment analysis from utterances is briefly sketched. Section 3 recalls the necessary background concerning MPEG-7 audio descriptors and the considered classification methods. In Section 4, the principles of the proposed approach are explained. Finally, Section 5 presents results of experimental testing and comparison of the already implemented classifiers on the publicly available Berlin database of emotional speech.

2 Sentiment Analysis from Utterances

Due to the importance of recognizing emotional states in speech, research into sentiment analysis from utterances has been emerging during recent years. We are aware of 3 publications reporting research with the same database of emotional utterances as we used – the Berlin Database of Emotional Speech, used in our research. Let us recall each of them.

The research most similar to ours has been reported in [12], where the authors also used MPEG-7 descriptors for sentiment analysis from utterance. However, they used only scalar MPEG-7 descriptors or scalars derived with time-series descriptors using the software tools Sound Description Toolbox [13] and MPEG-7 Audio Reference Software Toolkit[2], whereas we are implementing also a long-short-term memory network that will use directly the time series. They also used only one classifer in their experiments, a combination of a radial basis function network and a support vector machine.

In [11], emotions are recognized using pitch and prosody features, which are mostly in time domain. Also in that paper, the experiments were performed, and the authors used only one classifer, this time a support vector machine (SVM).

The authors of [16] proposed a set of new 68 features, such as some new based on harmonic frequencies or on the Zipf distribution, for better speech emotion recognition. This set of features is used in a multi-stage classification. When performing the sentiment analysis of the Berlin Database, the utterance classification to the considered emotional categories was preceded with a gender classification of the speakers, and the gender of the speaker was subsequently used as an additional feature for the classification of the utterances.

3 MPEG-7 Audio Descriptors

MPEG-7 is a standard for low-level description of audio signals, describing a signal by means of the following groups of descriptors[10]:

1. Basic: Audio Power (AP), Audio Waveform(AWF). Temporally sampled scalar values for general use, applicable to all kinds of signals. The AP describes the temporally-smoothed instantaneous power of samples in the frame,in other words it is a temporally measure of signal content as a function of time and offers a quick summary of a signal in conjunction with other basic spectral descriptors. The AWF describes audio waveform envelope (minimum and maximum), typically for display purposes.

2. Basic Spectral: Audio Spectrum Envelop (ASE), Audio Spectrum Centroid (ASC), Audio Spectrum Spread (ASS), Audio Spectrum Flatness (ASF).
   All share a common basis, all deriving from the short term audio signal spectrum (analysis of frequency over time). They are all based on the ASE Descriptor, which is a logarithmic-frequency spectrum. This descriptor provides a compact description of the signal spectral content and represents the similar approximation of logarithmic response of the human ear. The ASE descriptor is an indicator as to whether the spectral content of a signal is dominated by high or low frequencies. The ASC Descriptor could be considered as an approximation of perceptual sharpness of the signal. The ASS descriptor indicates whether the signal content, as it is represented by the power spectrum, is concentrated around its centroid or spread out over a wider range of the spectrum. This gives a measure which allows the distinction of noise-like sounds from tonal sounds. The ASF describes the flatness properties of the spectrum of an audio signal for each of a number of frequency bands.

3. Basic Signal Parameters: Audio Fundamental Frequency (AFF) and Audio Harmonicity (AH).
   The signal parameters constitute a simple parametric description of the audio signal. This group includes the computation of an estimate for the fundamental frequency (F0) of the audio signal. The AFF descriptor provides estimates of the fundamental frequency in segments in which the audio signal is assumed to be periodic. The AH represents the harmonicity of a signal, allowing distinction between sounds with a harmonic spectrum (e.g., musical tones or voiced speech e.g., vowels), sounds with an inharmonic spectrum (e.g., bell-like sounds) and sounds with a non-harmonic spectrum (e.g., noise, unvoiced speech).

4. Temporal Timbral: Log Attack Time (LAT), Temporal Centroid (TC).
   Timbre refers to features that allow one to distinguish two sounds that are equal in pitch, loudness and subjective duration. These descriptors are taking into account several perceptual dimensions at the same time in a complex way. Temporal Timbral descriptors describe the signal power function over time. The power function is estimated as a local mean square value of the signal amplitude value within a running window. The LAT descriptor characterizes the "attack" of a sound, the time it takes for the signal to rise from silence to its maximum amplitude. This feature signifies the difference between a sudden and a smooth sound. The TC descriptor computes a time-based centroid as the time average over the energy envelope of the signal.

5. Timbral Spectral descriptors: Harmonic Spectral Centroid (HSC), Harmonic Spectral Deviation (HSD), Harmonic Spectral Spread (HSS), Harmonic Spectral Variation (HSV) and Spectral Centroid.
   These are spectral features extracted in a linear-frequency space. The HSC descriptor is defined as the average, over the signal duration, of the amplitude-weighted mean of the frequency of the bins (the harmonic peaks of the spectrum) in the linear power spectrum. It is has a high correlation with the perceptual feature of "sharpness" of a sound. The HSD descriptor measures the spectral deviation of the harmonic peaks from the global envelope. The HSS descriptor measures the amplitude-weighted standard deviation (Root Mean Square) of the harmonic peaks of the spectrum, normalized by the HSC. The HSV descriptor is the normalized correlation between the amplitude of the harmonic peaks between two subsequent time-slices of the signal.

6. Spectral Basis, which consists of Audio Spectrum Basis (ASB) and Audio Spectrum Projection (ASP).

3.1 Tools for Working with MPEG-7 Descriptors

We utilized the Sound Description Toolbox [13] and MPEG-7 Audio Analyzer - Low Level Descriptors Extractor [15] for our experiments. Both of them extract a number of MPEG-7 standard descriptors, both scalar ones and time series. In addition, the SDT also calculates perceptual features such as Mel Frequency Cepstral Coefficients, Specific Loudness and Sensation Coefficients. From this descriptors calculate means, covariances, means of first-order differences and covariances of first order differences. The Total number of features provided by this toolbox is 187.

4 Employed Classification Methods

We have elaborated our approach to sentiment analysis from utterances for six classification methods: k nearest neighbors, support vector machines, multilayer perceptrons, classification trees, random forests [7] and long short-term memory (LSTM) network [5, 6, 8]. The first five of them have already been implemented and tested (cf. Section 5), the last and most advanced one is still being implemented.

4.1 k Nearest Neighbours (kNN)

A very traditional way of classifying a new feature vector $x \in \mathscr{X}$ if a sequence of training data $(x_1,c_1),\ldots,(x_p,c_p)$ is available is the nearest neighbour method: take the x_j that is the closest to x among $x_1,\ldots,x_p$, and assign to x the class assigned to x_j, i.e., c_j.

A straightforward generalization of the nearest neighbour method is to take among $x_1,\ldots,x_p$ not one, but k feature vectors $x_{j_j},\ldots,x_{j_k}$ closest to x. Then x is assigned the

class $c \in C$ fulfilling

$$|\{i, 1 \leq i \leq k | c_{j_i} = c\}| = max_{c' \in C} |\{i, 1 \leq i \leq k | c_{j_i} = c'\}|. \quad (1)$$

This method is called, expectedly, k nearest neighbours, or k-NN for short.

4.2 Support Vector Machines (SVM)

Support vector machines are classifiers into two classes. This method attempts to derive from the training data $(x_1, c_1), \ldots, (x_p, c_p)$ the best possible generalization to unseen feature vectors.

If both classes, more precisely their intersections with the set $\{x_1, \ldots, x_p\}$ of training inputs, are in the space of feature vectors linearly separable, the method constructs two parallel hyperplanes $H_+ = \{x \in \mathbb{R}^n | x^\top w + b_+ = 0\}, H_- = \{x \in \mathbb{R}^n | x^\top w + b_- = 0\}$ such that the training data fulfil

$$c_k = \begin{cases} 1 & \text{if } x^\top w + b_+ \geq 0, \\ \text{-1} & \text{if } x^\top w + b_- \leq 0, \end{cases} k = 1, \ldots, p, \quad (2)$$

$$H_+ \cap \{x_1, \ldots, x_p\} \neq \emptyset, H_- \cap \{x_1, \ldots, x_p\} \neq \emptyset. \quad (3)$$

The hyperplanes H_+ and H_- alle called support hyperplanes. Their common normal vector w and intercepts b_+, b_- are obtained through solving the following constrained optimization task:

Maximize with respect to w, b_+, b_- the distance

$$d(H_+, H_-) = \frac{b_+ - b_-}{\|w\|} \quad (4)$$

on condition that the p inequalities (2) hold.

The distance (4) is commonly called margin. The solution to this optimization task coincides with the $(w^*, b_+^*, b_-^*, \alpha_1^*, \ldots, \alpha_p^*)$ of the Lagrange function

$$L(w, b_+, b_-, \alpha_1, \ldots, \alpha_p) = \|w\|^2 + \sum_{k=1}^{p} \alpha_k (\frac{, b_+ - b_-}{2} - c_k x_k^\top w) \quad (5)$$

where $\alpha_1, \ldots, \alpha_p \geq 0$ are Lagrange coefficients of the optimization task. Once the saddle point $(w^*, b_+^*, b_-^*, \alpha_1^*, \ldots, \alpha_p^*)$ is found, the classifier is defined by

$$\phi(x) = \begin{cases} 1 & \text{if } \sum_{x_k \in \mathscr{S}} \alpha_k^* c_k x^\top x_k + b^* \geq 0, \\ -1 & \text{if } \sum_{x_k \in \mathscr{S}} \alpha_k^* c_k x^\top x_k + b^* < 0, \end{cases} \quad (6)$$

where $b^* = \frac{1}{2}(b_+^* + b_-^*)$ and

$$\mathscr{S} = \{x_k | \alpha_k^* > 0\}. \quad (7)$$

Due to the Karush-Kuhn-Tucker (KKT) conditions,

$$\alpha_k^* (\frac{b_+^* - b_-^*}{2} - c_k x_k^\top w^*) = 0, k = 1, \ldots, p, \quad (8)$$

all feature vectors from the set $\mathscr{S}$ lie on some of the suport hyperplanes (3). Therefore, they are called support vectors. This name together with the observation that they completely determine the classifier defined in (6) explains why such a classifier is called support vector machine.

If the intersections of both classes with the training inputs are not linearly separable, a SVM is constructed similarly, but instead of the set of possible fature vectors, now the set of functions

$$\kappa(\cdot, x) \text{ for all possible feature vectors } x \quad (9)$$

is considered, where κ is a kernel, i.e., a mapping on pairs of feature vectors that is symmetric and such that for any $k \in \mathbb{N}$ and any sequence of different feature vectors $x_1, \ldots, x_k$, the matrix

$$G_\kappa(x_1, \ldots, x_k) = \begin{pmatrix} \kappa(x_1, x_1) & \ldots & \kappa(x_1, x_k) \\ \ldots\ldots\ldots\ldots\ldots\ldots & & \\ \kappa(x_k, x_1) & \ldots & \kappa(x_k, x_k) \end{pmatrix}, \quad (10)$$

which is called the Gramm matrix of $x_1, \ldots, x_k$, is positive semidefinite, i.e.,

$$(\forall y \in \mathbb{R}^k)\ y^\top G_\kappa(x_1, \ldots, x_k) y \geq 0. \quad (11)$$

The most commonly used kinds of kernels are the Gaussian kernel with a parameter $\varsigma > 0$,

$$(\forall x, x' \in \mathbb{R}^{n'})\ \kappa(x, x') = \exp\left(-\frac{1}{\varsigma}\|x - x'\|^2\right), \quad (12)$$

and polynomial kernel with parameters $d \in \mathbb{N}$ and $c \geq 0$,

$$(\forall x, x' \in \mathbb{R}^{n'})\ \kappa(x, x') = (x^\top x' + c)^d. \quad (13)$$

It is known [14] that, due to the properties of kernels, if the joint distribution of a sequence of different feature vectors $x_1, \ldots, x_k$ is continuous, then almost surely any proper subset of the set of functions $\{\kappa(\cdot, x_1), \ldots, \kappa(\cdot, x_k)\}$ is in the space of all functions (9) linearly separable from its complement.

However, the featre vectors x and x_k can't be simply replaced by the corresponding functions $\kappa(\cdot, x)$ and $\kappa(\cdot, x_k)$ in the definition (6) of a SVM classifier because a transpose $x^\top$ exists for a finite-dimensional vector, but not a for an infinite-dimensional function. Fortunately, the transpose occurs in (6) only as a part of the scalar product $x^\top x_k$. And a scalar product can be defined also on the space of all functions (9). Namely, the properties of a scalar product has the function that to the pair of functions $(\kappa(\cdot, x), \kappa(\cdot, x')$ assigns the value $\kappa(x, x')$. Using this scalar product in (6), we obtain the following definition of a SVM classifier for linearly non-separable classes:

$$\phi(x) = \begin{cases} 1 & \text{if } \sum_{x_k \in \mathscr{S}} \alpha_k^* c_k \kappa(x, x_k) + b \geq 0, \\ -1 & \text{if } \sum_{x_k \in \mathscr{S}} \alpha_k^* c_k \kappa(x, x_k) + b \geq 0. \end{cases} \quad (14)$$

4.3 Multilayer Perceptrons (MLP)

A multilayer percptron is a mapping ϕ of feature vectors to classes with which a directed graph $G_\phi = (\mathscr{V}, \mathscr{E})$ is associated. Due to the inspiration from biological neural networks, the vertices of G_ϕ are called *neurons* and its edges are called *connections*. In addition, G_ϕ is required to have a layered structure, which means that the set $\mathscr{V}$ of neurons can be decomposed into $L+1$ mutually disjoint layers, $\mathscr{V} = \mathscr{V}_0 \cup \mathscr{V}_1 \cup \cdots \cup \mathscr{V}_L, L \geq 2$, such that

$$(\forall (u,v) \in \mathscr{E})\ u \in \mathscr{V}_i, i = 0, \ldots, L-1\ \&\ v \notin \mathscr{V}_i \Rightarrow v \in \mathscr{V}_{i+1}. \tag{15}$$

The layer $\mathscr{I} = \mathscr{V}_0$ is called input layer of the MLP, the layer $\mathscr{O} = \mathscr{V}_L$ its output layer and the layers $\mathscr{H}_1 = \mathscr{V}_1, \ldots, \mathscr{H}_{L-1} = \mathscr{V}_{L-1}$ its hidden layers.

The purpose of the graph G_ϕ associated with the mapping ϕ is to define a decomposition of ϕ into simple mappings assigned to hidden and output neurons and to connections between neurons (input neurons normally only accept the components of the input, and no mappings are assigned to them). Inspired by biological terminology, mappings assigned to neurons are called *somatic*, those assigned to connections are called *synaptic*.

To each connection $(u,v) \in \mathscr{E}$, the multiplication by a weight $w_{(u,v)}$ is assigne as a synaptic mapping:

$$(\forall \xi \in \mathbb{R})\ f_{(u,v)}(\xi) = w_{(u,v)}\xi. \tag{16}$$

To each hidden neuron $v \in \mathscr{H}_i$, the following somatic mapping is assigned:

$$(\forall \xi \in \mathbb{R}^{|\mathrm{in}(v)|})\ f_v(\xi) = \varphi\Big(\sum_{u \in \mathrm{in}(v)} [\xi]_u + b_v\Big), \tag{17}$$

where $[\xi]_u$ for $u \in \mathrm{in}(v)$ denotes the component of ξ that is the output of the synaptic mapping $f_{u,v}$ assigned to the connection (u,v), $\mathrm{in}(v) = \{u \in \mathscr{V} | (u,v) \in \mathscr{E}\}$ is the input set of v, and $\varphi : \mathbb{R} \to \mathbb{R}$ is called activation function. Though the activation functions, in applications typically sigmoidal functions are used to this end, i.e., functions that are non-decreasing, piecewise continuous, and such that

$$-\infty < \lim_{t \to -\infty} \varphi(t) < \lim_{t \to \infty} \varphi(t) < \infty. \tag{18}$$

The activation functions most frequently encountered in MLPs are:

- the logistic function,

$$(\forall t \in \mathbb{R})\ \varphi(t) = \frac{1}{1+\mathrm{e}^{-t}}; \tag{19}$$

- the hyperbolic tangent,

$$\varphi(t) = \tanh t = \frac{\mathrm{e}^t - \mathrm{e}^{-t}}{\mathrm{e}^t + \mathrm{e}^{-t}}. \tag{20}$$

To an output neuron $v \in \mathscr{O}$, also a somatic mapping of the kind (17) with the activation functions (19) or (20) can be assigned. If it is the case, then the class c predicted for a feature vector x is obtained as $c = \arg\max_i(\phi(x))_i$, where $(\phi(x))_i$ denotes the i-the component of $\phi(x)$. Alternatively the activation function assigned to an output neuron can be the step function, aka Heaviside function

$$\varphi(t) = \begin{cases} 0 & \text{if } t < 0, \\ 1 & \text{if } t \geq 0. \end{cases} \tag{21}$$

In that case, the value $(\phi(x))_c$ already directly indicates whether x belongs to the class c.

4.4 Classification Trees (CT)

A classifier $\phi : \mathscr{X} \to C = \{c_1, \ldots, c_m\}$ is called binary classification tree, if there is a binary tree $T_\phi = (V_\phi, E_\phi)$ with vertices V_ϕ and edges E_ϕ such that:

(i) $V_\phi = \{v_1, \ldots, v_L, \ldots, v_{2L-1}\}$, where $L \geq 2$, v_0 is the root of T_ϕ, $v_1, \ldots, v_{L-1}$ are its forks and $v_L, \ldots, v_{2L-1}$ are its leaves.

(ii) If the children of a fork $v \in \{v_1, \ldots, v_{L-1}\}$ are $v^L \in V_\phi$ (left child) and $v^R \in V_\phi$ (right child) and if $v = v_i, v^L = v_j, v^R = v_k$, then $i < j < k$.

(iii) To each fork $v \in \{v_1, \ldots, v_{L-1}\}$, a predicate φ_v of some formal logic is assigned, evaluated on features of the input vectors $x \in \mathscr{X}$.

(iv) To each leaf $v \in \{v_L, \ldots, v_{2L-1}\}$, a class $c_v \in C$ is assigned.

(v) For each input $x \in \mathscr{X}$, the predicate φ_{v_1} assigned to the root is evaluated.

(vi) If for a fork $v \in \{v_1, \ldots, v_{L-1}\}$, the predicate φ_v evaluates true, then $\phi(x) = c_{v^L}$ in case v^L is already a leaf, and the predicate φ_{v^L} is evaluated in case v^L is still a fork.

(vii) If for a fork $v \in \{v_1, \ldots, v_{L-1}\}$, the predicate φ_v evaluates false, then $\phi(x) = c_{v^R}$ in case v^R is already a leaf, and the predicate φ_{v^R} is evaluated in case v^R is still a fork.

4.5 Random Forests (RF)

Random Forests are ensembles of classifiers in which the individual members are classification trees. They are constructed by bagging, i.e., bootstrap aggregation of individual trees, which consists in training each member of the ensemble with another set of training data, sampled randomly with replacement from the original training pairs $(x_1, c_1), \ldots, (x_p, c_p)$. Typical sizes of random forests encountered in applications are dozens to thousands trees. Subsequently, when new subjects are input to the forest, each tree classifies them separately, according to the leaves at which they end, and the final classification by the forest is obtained by means of an aggregation function. The usual aggregation function of random forests is majority voting, or some of its fuzzy generalizations.

According to which kind of randomness is involved in the costruction of the ensemble, two broad groups of random forests can be differentiated:

1. *Random forests grown in the full input space.* Each tree is trained using all considered input features. Consequently, any feature has to be taken into account when looking for the split condition assigned to an inner node of the tree. However, features actually occurring in the split conditions can be different from tree to tree, as a consequence of the fact that each tree is trained with another set of training data. For the same reason, even if a particular feature occurs in split conditions of two different trees, those conditions can be assigned to nodes at different levels of the tree.

   A great advantage of this kind of random forests is that each tree is trained using all the information available in its set of training data. Its main disadvantage is high computational complexity. In addition, if several or even only one variable are very noisy, that noise gets nonetheless incorporated into all trees in the forest. Because of those disadvantages, random forests are grown in the complete input space primarily if its dimension is not high and no input feature is substantially noisier than the remaining ones.

2. *Random forests grown in subspaces of the input space.* Each tree is trained using only a randomly chosen fraction of features, typically a small one. This means that a tree t is actually trained with projections of the training data into a low-dimensional space spanned by some randomly selected dimensions $i_{t,1} \leq \cdots \leq i_{t,d_t} \in \{1,\ldots,d\}$, where d is the dimension of the input space, and d_t is typically much smaller than d. Using only a subset of features not only makes forest training much faster, but also allows to eliminate noise originating from only several features. The price paid for both these advantages is that training makes use of only a part of the information available in the training data.

4.6 Long Short-Term Memory (LSTM)

An LSTM network is used for classification of sequences of feature vectors, or equivalently, multidimensional time series with discrete time. Alternatively, it can be also employed to obtain sequences of such classifications, i.e., in situations when the neural network input is a sequence of feature vectors and its output is a a sequence of classes. Differently to most of other commonly encountered kinds of artificial neural networks, an LSTM layer connects not simple neurons, but units with their own inner structure. Several variants of an LSTM have been proposed (e.g., [5, 6]), all of them include at least the following four kinds of units described below. Each of them has certain properties of usual ANN neurons, in particular, the values assigned to them depend, apart from a bias, on values assigned to the unit input at the same time step and on values assigned to the unit output at the previous time step. Hence, an LSTM network layers is a recurrent network.

(i) *Memory cells* can store values, aka cell states, for an arbitray time. They have no activation function, thus their output is actually a biased linear combination of unit inputs and of the values from the previous time step coming through recurrent connections.

(ii) *Input gate* controls the extent to which values from the previous unit or from the preceding layer influence the value stored in the memory cell. It has a sigmoidal activation function, which is applied to a biased linear combination of unit inputs and of values from the previous time step, though the bias and synaptic weights of the input and recurrent connections are specific and in general different from the bias and synaptic weights of the memory cell.

(iii) *Forget gate* controls the extent to which the memory cell state is supressed. It again has a sigmoidal activation function, which is applied to a specific biased linear combination of unit inputs and of values from the previous time step.

(iv) *Output gate* controls the extent to which the memory cell state influences the unit output. Also this gate has a sigmoidal activation function, which is applied to a specific biased linear combination of unit inputs and of values from the previous time step, and subsequently composed either directly with the cell state or with its sigmoidal transformation, using another sigmoid than is used by the gates.

5 Experimental Testing

5.1 Berlin Database of Emotional Speech

For the evaluation of already implemented classifiers, we used the publicly available dataset "EmoDB", aka Berlin database of emotional speech. It consists of 535 emotional utterances in 7 emotional categories namely anger, boredom, disgust, fear, happiness, sadness and neutral. These utterances are sentences read by 10 professional actors, 5 males and 5 females [1], which were recorded in an anechoic chamber under supervision by linguists and psychologists) . The actors were advised to read these predefined sentences in the targeted emotional categories, but the sentences do not contain any emotional bias. A human perception test was conducted with 20 persons, different from the speakers, in order to evaluate the quality of the recorded data with respect to recognisability and naturalness of presented emotion. This evaluation yielded a mean accuracy 86% over all emotional categories.

5.2 Experimental Settings

As input features, the outputs from the Sound Description Toolbox were used. Consequently, the input dimension

was 187. The already implemented classifiers were compared by means of a 10-fold cross-validation, using the following settings for each of them:

- For the k nearest neighbors classification, the value $k = 9$ was chosen by a grid method from $\langle 1, 80 \rangle$. This classifer was applied to data normalized to zero mean and unit variance.
- Support vector machines are constructed for each of the 7 considered emotions, to classify between that emotion and all the remaining ones. They employ auto-scaled Gaussian kernels and do not use slack variables.
- The MLP has 1 hidden layer with 70 neurons. Hence, taking into account the input dimension and the number of classes, the overall architecture of the MLP is 187-70-7.
- Classification trees are restricted to have at most 23 leaves. This upper limit was chosen by a grid method from $\langle 1, 50 \rangle$, taking into account the way how classification trees are grown in their Matlab implementation.
- Random forests consist of 50 classification trees, each of them taking over the above restriction. The number of trees was selected by a grid method from 10, 20,. . . ,100.

5.3 Comparison of Already Implemented Classifiers

First, we compared the already implemented classifiers on the whole Berlin database of emotional speech, with respect to accuracy and area under the ROC curve (area under curve, AUC). Since a ROC curve makes sense only for a binary classifier, we computed areas under 7 separate curves corresponding to classifiers classifying always 1 emotion against the rest. The results are presented in Table 1 and in Figure 1. They clearly show SVM as the most promising classifier. It has the highest accuracy, and also the AUC for binary classifiers corresponding to 5 of the 7 classifiers

Then we compared the classifiers separately on the utterances of each of the 10 speakers who created the database. The results are summarized in Table 2 for accuracy and Table 3 for AUC averaged over all 7 emotions. They indicate a great difference between most of the compared classifiers. This is confirmed by the Friedman test of the hypotheses that all classifiers have equal accuracy and equal average AUC. The Friedman test rejected both hypotheses with a high significance: With the Holm correction for simultaneously tested hypotheses [9], the achieved significance level (aka p-value) was $4 \cdot 10^{-6}$. For both hypotheses, posthoc tests according to [3, 4] were performed, testing equal accuracy and equal average AUC between individual pairs of classifiers. For

Table 1: Accuracy and area under curve (AUC) of the implemented classifiers on the whole Berlin database of emotional speech. AUC is measured for binary classification of each of the considered 7 emotions against the rest

Classifier	Accuracy	AUC emotion against the rest		
		Anger	Boredom	Disgust
kNN	0.73	0.956	0.933	0.901
SVM	**0.93**	**0.979**	**0.973**	**0.966**
MLP	0.78	0.977	0.969	0.964
DT	0.59	0.871	0.836	0.772
RF	0.71	0.962	0.949	0.920

Classifier	AUC emotion against the rest			
	Fear	Happiness	Neutral	Sadness
kNN	0.902	0.856	0.962	0.995
SVM	**0.983**	0.904	0.974	**0.997**
MLP	0.969	**0.933**	**0.983**	0.996
DT	0.782	0.683	0.855	0.865
RF	0.921	0.882	0.972	0.992

Table 2: Comparison between pairs of implemented classifiers with respect to accuracy, based on 10 independent parts of the Berlin database of emotional speech corresponding to 10 different speakers. The result in a cell of the table indicates on how many parts the accuracy of the row classifier was higher : on how many parts the accuracy of the column classifier was higher. A result in bold indicates that after the Friedman test rejected the hypothesis of equal accuracy of all classifiers, the post-hoc test according to [3, 4] rejects the hypothesis of equal accuracy of the particular row and column classifiers. All simultaneously tested hypotheses were corrected in accordance with Holm [9]

classifier	kNN	SVM	MLP	DT	RF
kNN		**0:10**	3.5:6.5	9:1	5:5
SVM	**10:0**		10:0	**10:0**	**10:0**
MLP	6.5:3.5	0:10		**10:0**	7:3
DT	1:9	**0:10**	**0:10**		0:10
RF	5:5	**0:10**	3:7	10:0	

the family-wise significance level 5%, they reveal the following Holm-corrected significant differences between individual pairs of classifiers: both for accuracy and averaged AUC: (SVM,DT), (MLP,DT), and in addition between (kNN,SVM), (SVM,RF) for accuracy.

6 Conclusion

The presented work in progress investigated the possibilities to analyse emotions in utterances based on MPEG7 features. So far, we implemented only five classification methods not using time series features, but only 187 scalar features, namely the k nearest neighbours classifier, support vector machines, mutilayer perceptrons, decision trees and random forests. The obtained results in-

Table 3: Comparison between pairs of implemented classifiers with respect to the AUC averaged over all 7 emotions, based on 10 independent parts of the Berlin database of emotional speech corresponding to 10 different speakers. The result in a cell of the table indicates on how many parts the AUC of the row classifier was higher : on how many parts the AUC of the column classifier was higher. A result in bold indicates that after the Friedman test rejected the hypothesis of equal AUC of all classifiers, the post-hoc test according to [3, 4] rejects the hypothesis of equal AUC of the particular row and column classifiers. All simultaneously tested hypotheses were corrected in accordance with Holm [9]

classifier	kNN	SVM	MLP	DT	RF
kNN		2:8	0:10	10:0	4:6
SVM	8:2		5:5	**10:0**	9:1
MLP	10:0	5:5		**10:0**	9:1
DT	0:10	**0:10**	**0:10**		0:10
RF	6:4	1:9	1:9	10:0	

dicate that especially support vector machines and multilayer perceptrons are quite successfull for this task. Statistical testing confirms significant differences between these two kinds of classifiers on the one hand, and decision trees an random forests on the other hand.

The next step in this ongoing research is to implement the long short-term memory neural network, recalled in Subsection 4.6, because they can work not only with scalar features but also with features represented with time series.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GAČR) grant 18-18080S.

References

[1] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss. A database of german emotional speech. In *Interspeech*, pages 1517–1520, 2005.

[2] M. Casey, A. De Cheveigne, P. Gardner, M. Jackson, and G. Peeters. MPEG-7 multimedia software resources. http://mpeg7.doc.gold.ac.uk/, 2001.

[3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[4] S. Garcia and F. Herrera. An extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[5] F.A. Gers, J. Schmidhuber, and J. Cummis. Learning to forget: Continual prediction with LSTM. In *9th International Conference on Artificial Neural Networks: ICANN '99*, pages 850–855, 1999.

[6] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, TU München, 2008.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, 2nd Edition*. Springer, 2008.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[9] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[10] H.G. Kim, N. Moreau, and T. Sikora. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. John Wiley and Sons, New York, 2005.

[11] S. Lalitha, A. Madhavan, B. Bhusan, and S. Saketh. Speech emotion recognition. In *International Conference on Advances in Electronics*, pages 92–95, 2014.

[12] A.S. Lampropoulos and G.A. Tsihrintzis. Evaluation of MPEG-7 descriptors for speech emotional recognition. In *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 98–101, 2012.

[13] A. Rauber, T. Lidy, J. Frank, E. Benetos, V. Zenz, G. Bertini, T. Virtanen, A.T. Cemgil, S. Godsill, D. Clark, P. Peeling, E. Peisyer, Y. Laprie, A. Sloin, A. Alfandary, and D. Burshtein. MUSCLE network of excellence: Multimedia understanding through semantics, computation and learning. Technical report, TU Vienna, Information and Software Engineering Group, 2004.

[14] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[15] T. Sikora, H.G. Kim, N. Moreau, and S. Amjad. MPEG-7-based audio annotation for the archival of digital video. http://mpeg7lld.nue.tu-berlin.de/, 2003.

[16] Z. Xiao, E. Dellandrea, W. Dou, and L. Chen. Multi-stage classification of emotional speech motivated by a dimensional emotion model. *Multimedia Tools and Applicaions*, 46:119–145, 2010.

Figure 1: ROC curve for all emotions on the whole Berlin database