Comparing Multiple Classifiers for Speech-Based Detection of Self-Confidence – A Pilot Study

Jarek Krajewski [1], Anton Batliner [2], Silke Kessel [1]
[1] *Experimental Business Psychology, Univ. of Wuppertal, Germany*
[2] *Lehrstuhl fuer Mustererkennung, Univ. of Erlangen-Nuremberg, Germany*
{krajewski, kessel}@wiwi.uni-wuppertal.de, batliner@informatik.uni-erlangen.de

Abstract

The aim of this study is to compare several classifiers commonly used within the field of speech emotion recognition (SER) on the speech based detection of self-confidence. A standard acoustic feature set was computed, resulting in 170 features per one-minute speech sample (e.g. fundamental frequency, intensity, formants, MFCCs). In order to identify speech correlates of self-confidence, the lectures of 14 female participants were recorded, resulting in 306 one-minute segments of speech. Five expert raters independently assessed the self-confidence impression. Several classification models (e.g. Random Forest, Support Vector Machine, Naïve Bayes, Multi-Layer Perceptron) and ensemble classifiers (AdaBoost, Bagging, Stacking) were trained. AdaBoost procedures turned out to achieve best performance, both for single models (AdaBoost LR: 75.2% class-wise averaged recognition rate) and for average boosting (59.3%) within speaker-independent settings.

1. Introduction

So far, there has been little empirical research on acoustic voice characteristics of self-confidence. Nevertheless, the following tendencies could be observed: confident speakers appeared to have a less monotonous way of presenting, less and shorter breaks, a fast rate of speech, a lower voice level, a higher intensity of speech, a hard voice quality, a short latency of response, and only a few corrections of own mistakes [1,2]. Most studies have analyzed single features or small feature sets, only containing perceptual acoustic features, whereas signal processing based speech and speaker recognition features (e.g. MFCCs) and pattern recognition algorithms have received little attention. Thus, the aim of this study is to apply a state-of-the-art speech emotion recognition (SER) engine [3, 4] on the detection of self-confidence with the focus on the comparison of several commonly applied classifiers.

The rest of this paper is organized as follows: Section 2 describes the speech database, section 3 feature extraction, and section 4 classifiers employed. After providing the results of the self-confidence detection in Section 5, results and future work are discussed in Section 6.

2. Database

Fourteen female participants took part in this study. Due to data sparseness, only female participants were chosen for enlarging the homogeneity and reducing additional confounding. The mean age of our subjects was 25.59, *SD* = 3.59 years. The recordings took place in lecture-rooms and consisted of regular lectures given by students or research assistants (sampling rate 44.1 kHz, quantisation 16 bit, microphone-to-mouth distance 3 meter). The speech data was split into fixed intervals of one minute, and rated in randomized order for self-confidence by five expert raters – i.e. assessors who had been formally trained to apply a standardized set of judging criteria. In order to measure self-confidence, the ratings were made on scales ranging from 1 (very uncertain) to 10 (very self-confident). Due to high inter-rater reliability ($r = .91$), the resulting values of our five raters were averaged.

For training and classification purposes, the records (interquartile range = 1.77) were further divided into two classes based on a median split: low self-confident (LC) and high self-confident (HC) samples with the boundary value ≥ 7.0 (15-25 samples per subject; total number of speech samples: 306 samples; 127 samples LC, 179 samples HC).

3. Feature Extraction

All acoustic measurements were done over one-minute segments using the Praat speech analysis software [5]. Formant processing (F1- F5) used a pre-emphasis filter with frequency response of 25 ms hamming window and 10 ms step size. For our study we extracted and computed the following types of features, typically used within the state-of-the-art SER [6]:

• prosodic features (26): fundamental frequency, intensity, and other types of supra-segmental information such as jitter and shimmer; in particular, we computed the functionals mean, 2^{nd} to 4^{th} quartile, standard deviation, maximum, minimum, range, positions and values of maxima and minima. Finally, we considered jitter and shimmer, and short-term fluctuations in energy and fundamental frequency.

•spectral features I (108): frequencies, bandwidths, and amplitudes of the F1-F5 formants, and the frequencies and amplitudes of the first 2 harmonics. Moreover, we calculated 4 Hammarberg indices and the average LTAS spectrum on 6 frequency bands (125-200 Hz, 200-300 Hz, 500-600 Hz, 1000-1600 Hz, 5000-8000 Hz), the proportion of low frequency energy under 500Hz/1000Hz, the slope of spectral energy above 1000 Hz, the Harmonic-to-Noise ratio (HNR), and spectral tilt features ("open quotient", "glottal opening", "skewness of glottal pulse", and "rate of glottal closure").

•spectral features II (36): the usual 36 MFCC features (12 MFCC, 12 ΔMFCC, 12 ΔΔMFCC). To calculate these coefficients, we averaged the frame-wise computed mel-cepstral coefficients and 12 time differences over the entire signal. We expect these coefficients to account for specific properties of the confident speech such as changed voice quality.

In sum, we computed a total amount of 170 features per speech sample, which were z-normalized in order to scale the data.

4. Classification

Various approaches have been suggested to build ensembles of classifiers including the application of different (a) subsets of training data with a single learning method, (b) training parameters with a single training method, and (c) learning methods. Experiments on several benchmark data sets and real world data sets showed improved classification results when using these techniques [7]. In this paper we particularly focus on the two ensembles techniques Bagging and Boosting [8, 9].

Generally speaking, Bagging and Boosting do not try to design learning algorithms which are accurate over the entire feature space, but work best for weak learning algorithms fitting in subsamples. They show highest gain for weak classifiers, but have also shown beneficial for strong ones such as SVM or C4.5 (Random Forests). The key principle of the bootstrapping and aggregating technique Bagging is to use bootstrap re-sampling to generate multiple versions of a classifier. Bootstrapping is based on random sampling with replacement. Thus, taking a random selection with replacement of the training data can get less misleading training objects ('outlier'). Therefore, the resulting classifiers may be sharper than those obtained on the training sets with outliers. The second ensemble technique Boosting works by repeatedly running a learning algorithm on various distributions over the training data, and then combining the classifiers. In contrast to Bagging, where training sets and classifiers are obtained randomly and independently from the previous step, training data is obtained sequentially and deterministic in the Boosting algorithm, reweighting incorrectly classified objects in a new modified training set. Boosting algorithms have also been applied in various research fields, as e.g., natural language processing. In order to determine the added-value of bagging and boosting in this application field, we applied these techniques on several commonly applied base-classifiers.

Classifiers typically used within SER include a broad variety of dynamic algorithms (Hidden Markov Models) and static classifiers [10]. When choosing a classifier within this highly correlated and noisy feature space, several aspects might be of importance such as low memory, low computation time, quick converging, and no suffering from overfitting. With respect to these requirements, we applied the following static classifiers from the popular 4.5 RapidMiner [11] software using standard parameter settings: Support Vector Machines ('LibSVM', rbf kernel function; 'FastLargeMargin' [12], linear kernel; 'W-SMO', Sequential Minimal Optimization), Logistic Regressions ('KernelLogisticRegression'; 'MyKLR' 'LogisticRegression'), Multilayer Perceptrons ('NeuralNetImproved', 1 hidden sigmoid layer, (number of attributes + number of classes) / 2 + 1 nodes; 'W-MultilayerPerceptron', 2 hidden sigmoid layer, 5 nodes each; 'Perceptron'), k-Nearest Neighbors ('NearestNeighbors'; k = 1, 5), Decision Trees ('DecisionTree', C4.5; 'RandomForest', 100 trees), Naive Bayes ('NaiveBayes'; 'W-DMNB', Discriminative Multinominal Naive Bayes), Rule

Learner ('RuleLearner'), and Logistic Base ('LogisticBase').

Table 1: *Class-wise averaged classification rate (CL) in % of several classifiers on the test set using speaker-dependent (SD) and speaker-independent (SI) validation schemes.*

Classifier	SI	SD	Classifier (contd.)	SI	SD
LibSVM	39.4	79.2	1-NearestNeighbor (1-NN)	48.6	70.3
Bagging LibSVM	50.0	50.0	Bagging 1-NN	51.3	81.9
AdaBoost LibSVM	53.3	78.8	AdaBoost 1-NN	47.3	70.2
FastLargeMargin	71.9	80.1	5-NN	47.1	85.6
Bagging FLM	73.4	81.9	Bagging 5-NN	47.3	84.8
AdaBoost FLM	72.2	80.1	AdaBoost 5-NN	47.1	87.7
SMO	65.9	82.5	RandomForest (RF)	40.5	84.9
Bagging SMO	65.6	83.2	Bagging RF	38.5	87.4
AdaBoost SMO	65.4	81.3	AdaBoost RF	38.1	86.6
MyKLR	65.8	79.2	DecisionTree (DT)	55.5	72.8
Bagging MyKLR	47.3	67.1	Bagging DT	46.4	82.0
AdaBoost MyKLR	65.8	64.9	AdaBoost DT	46.6	74.1
LogisticRegression (LR)	67.5	74.3	W-MDNB (MDNB)	55.4	70.1
Bagging LR	70.9	82.5	Bagging MDNB	55.4	74.4
AdaBoost LR	75.2	75.4	AdaBoost MDNB	58.0	80.3
Perceptron (PCT)	62.6	78.6	NaïveBayes (NB)	33.5	57.9
Bagging PCT	62.4	77.1	Bagging NB	38.8	57.6
AdaBoost PCT	63.2	77.1	AdaBoost NB	40.0	64.7
W-Multilayer Perceptron (MLP)	70.4	85.2	RuleLearner (RL)	45.3	79.4
Bagging MLP	71.7	80.5	Bagging RL	53.7	80.4
AdaBoost MLP	70.4	80.2	AdaBoost RL	55.6	81.9
NeuralNetImprov. (NNI)	61.4	82.8	LogisticBase (LB)	57.4	82.1
Bagging NNI	70.6	81.8	Bagging LB	62.0	84.7
AdaBoost NNI	72.5	80.5	AdaBoost LB	55.4	76.7

In a speaker-dependent validation protocol, we applied a stratified 2-fold crossvalidation. The final classification errors were calculated averaging over both classifications. In addition, a speaker-independent approach, i.e. a 2-fold cross-validation on unseen speaker, has been carried out using half of the speakers as test set and all other as train. Thus, utmost independence of the speaker and room acoustics is ensured.

Table 2: *Class-wise averaged classification rate (CL) in % of average single classifier, average bagging, and average boosting algorithms (sets are disjunct) on the test set using speaker-dependent (SD) and speaker-independent (SI) validation schemes.*

Classifier	SI	SD	ΔSI	ΔSD
Average single classifier	55.5	77.8	//	//
Average Bagging	56.6	77.3	+1.1	-0.5
Average AdaBoost	59.3	78.4	+3.8	+0.6

5. Results

In order to determine the detection performance, different classifiers were applied on the full set of 170 features. The unweighted average, i.e. the 'class-wise' determined recognition rate (CL) of the different classifiers for the two class prediction problems is computed (see Table 1). The ensemble results are depicted in Table 2. Within the speaker-independent approach, the AdaBoost Logistic Regression and the Bagging FastLargeMargin classifier (SVM, linear kernel) reached the highest CL of 75.2% and 73.4%. Applying just base classifiers would result in a maximum CL of 71.9% (AdaBoost, +3.3%; Bagging, +1.5%).

For the speaker-dependent approach, the AdaBoost 5-Nearest Neighbor and Bagging Random Forest classifier achieved the highest class-wise averaged classification rate of 87.7% and 87.4%. Again, applying just base classifier would result in an maximum CL of 85.6% (AdaBoost, +2.1%; Bagging, +1.8%). Within the ensemble classification schemes, the AdaBoost algorithm achieved the highest average CL resulting in a slight average improvement over the average single classifier (cf. Table 2)

6. Discussion

The main findings of this pilot study may be summarized as follows. First, using all acoustic features and all samples (without pre-selecting prototypical classes out of the whole database, cf. [13]) we achieved on the two-class detection problem (low self-confident vs. high self-confident speech) a CL of 87.7% on speaker-dependent data with the best performing classifier RF. Second, in our experiments on a two-class detection problem on unseen speaker, we achieved a best CL of 75.2% (AdaBoost 'LogisticRegression'). This corresponds to the average

loss of ca. 20% CL for choosing the speaker-dependent detection. Third, both ensemble techniques improved the maximum classifier performance (SI: AdaBoost, +3.3%, Bagging, +1.5%; SD: AdaBoost, +2.1%, Bagging, +1.8%). Despite of the noisy data, the AdaBoost reached higher performance gains than Bagging.

Our results are limited by several facts: We do need more speakers, and male and female speakers alike. Linguistic features such as bag-of-words/n-grams or part-of-speech should be added to the feature vector; they have been proven to be beneficial even if the SER task is based on automatic speech recognition results [14]. This performance gain might probably be higher than adding other fancy classifiers. Furthermore, segmenting the speech samples into phonetic or linguistic meaningful units such as phonemes, syllables, words, and later combine them into syntactically meaningful chunks or ememe sequences [3] might be a promising approach. Additionally, a performance gain could be reached by aggregating several 1-minute based classification decisions to a global confidence value of the speaker. Due to data scarcity, we could not split the data into an additional validation set. Thus, we abandoned the option of an explicit feature selection (e.g. cross-classifier feature selection applying sequential floating forward selection) or a tuning of the classifiers (e.g. optimizing regularization parameter C in SVM). This is of course the largest drawback of the present study: we only used standard parameter settings. As could be expected, large performance differences between single classifiers could be observed which may not be interpreted. However, the advantage of Average AdaBoost over Average Bagging and Average single classifier displayed in Table 2 might be large enough to warrant future experiments.

References

[1] K.R. Scherer, H. London, & J.J. Wolf. The voice of confidence: Paralinguistic cues and audience evaluation, Journal of Research in Personality, 7, 31-44, 1973.

[2] M. Boltz. Temporal dimensions of conversational interaction: The role of response latencies and pauses in social impression formation, Journal of Language and Social Psychology, 24, 103-138, 2005.

[3] A. Batliner, D. Seppi, S. Steidl, & B. Schuller. Segmenting into adequate units for automatic recognition of emotion-related episodes: a speech-based approach, Advances in Human-Computer Interaction. (in press).

[4] S. Steidl. Automatic classification of emotion-related user states in spontaneous children's speech. Logos Verlag, Berlin, 2009. PhD thesis.

[5] P. Boersma. Praat, a system for doing phonetics by computer, Glot International, 5, 9/10, 341–345, 2001.

[6] T. Athanaselis, S. Bakamidis, I. Dologlu, R. Cowie, E. Douglas-Cowie, & C. Cox. ASR for emotional speech: Clarifying the issues and enhancing performance. Neural Networks, 18:437–444, 2005.

[7] Breiman, L.: Bagging Predictors: Machine Learning, 24, 123-140, 1996.

[8] S.B. Kotsianti & D. Kanellopoulos Combining bagging, boosting and dagging for classification problems. Knowledge based intelligent information and engineering systems, Lecture Notes in Computer Science, 4693, 493-500, 2009.

[9] P. Boinee, A. De Angelis, & G. L. Foresti. Ensembling Classifiers - An application to image data classification from Cherenkov telescope experiment. IEC 2005, 394-398, 2005.

[10] Z. Zeng, M. Pantic, G. I. Roisman, & T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31:39–58, 2009.

[11] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, & T. Euler. YALE: Rapid prototyping for complex data mining tasks. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.

[12] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, & C.J. Lin. Liblinear: A library for large linear classification, Journal of Machine Learning Research 9, 2008.

[13] B. Schuller, D. Seppi, A. Batliner, A. Meier, & S. Steidl. Towards more reality in the recognition of emotional speech. In Proc. ICASSP, pp 941–944, Honolulu, 2007.

[14] B. Schuller, A. Batliner, D. Seppi, & S. Steidl. Emotion recognition from speech: Putting ASR in the loop. Proc. ICASSP, 4585-4588, Taipei, 2009.