Real-time Recognition of Affective States from Non-verbal Features of Speech and its Application for Public Speaking Skill Analysis

Tomas Pfister and Peter Robinson

Abstract—This paper presents a new classification algorithm for real-time inference of affect from non-verbal features of speech and applies it to assessing public speaking skills. The classifier identifies simultaneously occurring affective states by recognising correlations between emotions and over 6000 functional-feature combinations. Pairwise classifiers are constructed for 9 classes from the Mind Reading emotion corpus, yielding an average cross-validation accuracy of 89% for the pairwise machines and 86% for the fused machine. The paper also shows a novel application of the classifier for assessing public speaking skills, achieving an average cross-validation accuracy of 81% and a leave-one-speaker-out classification accuracy of 61%. Optimising support vector machine coefficients using grid parameter search is shown to improve the accuracy by up to 25%. The emotion classifier outperforms previous research on the same emotion corpus and is successfully applied to analyse public speaking skills.

Index Terms—Affect analysis, speech analysis, public speaking, speech coaching, emotion in human-computer interaction.

1 Introduction

EMOTIONS are fundamental for humans, impacting perception and everyday activities such as communication, learning and decision-making. They are expressed through speech, facial expressions, gestures and other non-verbal clues.

Affective speech analysis refers to the analysis of spoken behaviour as a marker of emotion, with focus on the non-verbal aspects of speech. Its assumption is that the affective state of a person can be objectively measured by analysing features from speech. Supporting evidence for this assumption includes empirical evidence that some emotions are associated with physiological reactions which produce a change in how the voice is produced. For example, anger often produces changes in respiration and increases muscle tension, influencing the vibration of the vocal folds and vocal tract shape, thus affecting the acoustic characteristics of the speech [1].

A completely new application of emotion detection proposed in this paper is speech tutoring. Especially in persuasive communication, the non-verbal clues a speaker conveys require focused attention. Untrained speakers often come across as bland, lifeless and colourless. Precisely measuring and analysing the voice is a difficult task and has in the past been entirely subjective. By using a similar approach as for detecting emotions, this paper shows that such judgements can be made objective.

• *Tomas Pfister and Peter Robinson are with the University of Cambridge Computer Laboratory, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK.*
E-mail: tomas.pfister@cantab.net, pr10@cam.ac.uk

In the past, there was a lack of interest in emotions among computer scientists [2]. Pioneering work by Rosalind Picard [3] in the late 1990s enabled a larger audience to see the need for integrating emotions into computing. Although the field has recently received an increase in contributions, it remains a new area of study with a number of potential applications. These include emotional hearing aids for people with autism; detection of an angry caller at an automated call centre to transfer to a human; or presentation style adjustment of a computerised e-learning tutor if the student is bored.

Discovering which features are indicative of emotional states and consecutively capturing them can be a difficult task. Furthermore, features indicating different states may be overlapping, and there may be multiple sets of features expressing the same emotional state. One widely used strategy is to compute as many features as possible. Optimisation algorithms can then be applied to select the features contributing most to the discrimination while ignoring others. This atheoretical approach to emotions avoids making difficult *a priori* decisions about which features may be relevant.

Previous studies indicate that several emotions can occur simultaneously [4]. Examples of co-occurring emotions include being happy at the same time as being tired, or feeling touched, surprised and excited when hearing good news. Improving upon the inference solution for co-occurring emotions presented by Sobol Shikler [5], the new system proposed in this paper is able to achieve real-time performance and higher classification accuracy.

In this paper, we describe an approach for real-time classification of co-occurring emotions. [6] The classi-

fication output is a set of classes rather than a single one, allowing nuances and mixtures of emotions to be detected. Finally, we present a novel application of the classifier to virtual speech coaching for improving public speaking skills and show that objective computerised analysis of public speaking skills is feasible.

2 Application for Public Speaking Skill Assessment

In addition to presenting an emotion classifier achieving state-of-the-art accuracy, we present a novel application of the classifier for assessing the quality of public speaking skills using our new Speech Tutor corpus.

In persuasive communication, the non-verbal clues a speaker conveys require special attention. Untrained speakers often come across as bland and lifeless. Precisely analysing the voice is difficult for humans and is subjective. By using a similar approach as for detecting emotions, our system enables more objective assessment of public speaking skills.

Potential applications for the analysis include speech coaching and presentation skill practice. A speech coach could use the tool to achieve more objective and precise analysis. A presenter could use the immediate presentation feedback to practise public speaking at home and modify the speaking style according to the computer's feedback and the advice from a speech coach.

3 Previous Research

One of the first studies investigating the link between intonation and emotions was done by Uldall [7], who modified the intonation contour artificially. Huttar [8] later recorded natural speech in lectures and found significant correlations between prosodic features and emotions. Significant fundamental work on speech emotion detection was done by Dellaert et al. [9], who proposed the use of statistical pattern recognition techniques for emotion detection and set the basic system architecture still used today. Their initial work achieved accuracies of 60-65% using four classes of acted data.

Starting from then, the accuracy has continually been improving. Since people are not very good at faking emotions on request, the focus has shifted from acted data to induced and natural data. Batliner et al. [10] detected the existence of emotions with 95% accuracy using an induced corpus. More recently, Forbes-Riley et al. [11] and others have been using human-human dialogues, achieving 84% accuracy for valence. Steidl et al. [12] among others worked with human-machine databases, achieving 60% accuracy for four classes. Other recent works include work using both acoustic and language features to detect affect from call centre data [13] and medical dialogues [14]. Schuller et al. [15] analysed interest in human-human dialogues using a Multiple-Instance Learning approach on frames instead of doing segmental analysis.

More recently, the INTERSPEECH speech emotion recognition competition [16] was a collective effort for increasing the classification accuracy for a specific set of data. Lee et al. [17] achieved a high emotion recognition accuracy using hierarchical binary decision trees. Dumouchel et al. [18] achieved 70% recall using cepstral features. By fusing the output from all candidates' systems, the organisers managed to achieve performance that exceeded all of the individual results.

A wide range of corpora are commonly used in research, including among others the Belfast database [19], the EmoTV corpus [20], EMO-DB [21], eNTERFACE corpus [22], Audio Visual Interest Corpus (AVIC) [23] and the Mind Reading corpus [24]. The choice of emotion corpus is heavily influenced by the application and situation.

Popular classification models used include, among others, different decision trees [25], support vector machines [5], [26], [27], [17], [6], neural networks [12], [10] and Hidden Markov Models. Again, which is the best classifier often depends on the application and corpus. To combine the benefits of different classifiers, classifier fusion [28], [29] is starting to become common. Schuller et al. [29] combine support vector machines, decision trees and Bayesian classifiers to yield higher classification accuracy. Scherer et al. [28] combine three different KNN classifiers to improve the results.

Many studies in psychology have also examined vocal expressions of emotions. Russell et al. [30], Scherer [31] and Scherer et al. [32] provide reviews of these. Non-linguistic vocalisations such as laughter, cries, sighs and yawns were investigated but mounting evidence questions whether they are each linked to a specific discrete state [33]. For vocal expressions of speech, the strongest single association found for vocal acoustics has been with the arousal level [34].

The first issue of this journal had a number of theoretical overviews of emotion recognition. Calvo and d'Mello [35] examines various emotion theories and reveals several problematic assumptions common in affective computing. The authors suggest broadening of the mental states studied, investigating dimensions versus categories for labelling, and integrating context into recognition. Reisenzein [36] discusses the limitations of affect recognition purely from nonverbal expressions of emotion.

In this paper, we apply some of these ideas to achieve state-of-the-art classification performance on the Mind Reading corpus, and present a novel application of the classifier for successfully analysing public speaking skills.

4 Implementation Methodology

The design of the classifier considers three main factors: (i) the need for real-time performance, (ii) the ability to recognise co-occurring emotions, (iii) the choice of a training corpus.

Achieving real-time performance required a careful choice of feature extraction and classification algorithms. Recognising co-occurring emotions needed a method for ranking candidate emotions.

The overall system design is shown in Figure 1. Explanations of the choices made are given in the subsections below.

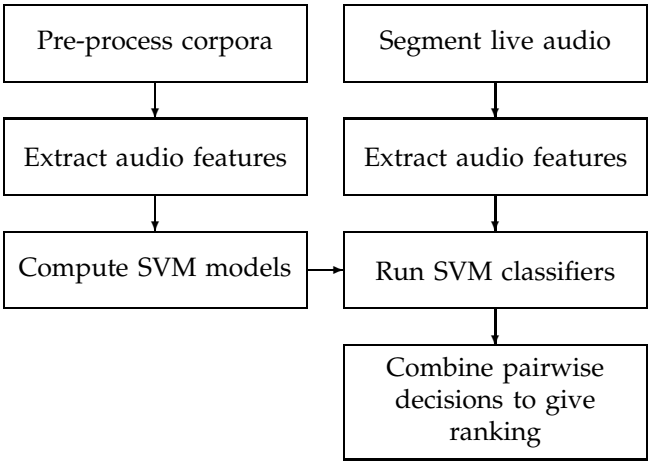


Fig. 1. Schematic flowchart of the functionality implemented for affect recognition from speech.

4.1 Emotion Classification

For emotion classification we choose the Mind Reading corpus [24] which provides a hierarchical structure between groups with a large number of emotion concepts. It was developed by psychologists at University of Cambridge Autism Research Centre, aiming to help autistic children and adults to recognise both basic and complex emotions. The corpus consists of 2927 acted everyday sentences, covering 442 different concepts of emotions, each with 5-7 sentences. The acting was induced and the labelling was done by ten people in different age groups [37]. The labelling of each sample in the corpus required the agreement of 8 members of a panel of 10 expert assessors. Although the samples are acted, the large number of samples makes the corpus suitable for training an emotion classifier.

The main emotion groups of Mind Reading are shown in Table 1. Each of these is further divided into concepts, giving a total of 422 concept subgroups. For the classifier, a subset of 9 categories representing a large variety of emotions is chosen. This subset consisted of 548 samples spoken by 10 different actors. Each category contains samples from the groups as shown in Table 1. The subcategories *absorbed* and *interested* were extracted from the concepts in the *interested* main category. The subcategory *joyful* was extracted from the concepts in the *happy* main category, *opposed* from *unfriendly* and *stressed* from *bothered*. Other subcategories were extracted from the main category with the same name. The subcategories are chosen to minimise the overlap between categories. The subcategories and samples are the same as those used by Sobol Shikler [5], [25], allowing direct comparison of results.

TABLE 1
The 24 emotion groups in the Mind Reading corpus [37]. The superscripts indicate the main groups from which a subset of affective states is selected to allow comparison of the results to previous research [5]. These subsets, with their respective number of samples, are: absorbed[1] (41), excited[2] (46), interested[3] (44), joyful[4] (94), opposed[5] (38), stressed[6] (87), sure[7] (53), thinking[8] (68) and unsure[9] (77).

afraid	angry	bored	bothered[6]	disbelieving
disgusted	excited[2]	fond	happy[4]	hurt
interested[1,3]	kind	liked	romantic	sad
sneaky	sorry	sure[7]	surprised	think[8]
touched	unfriendly[5]	unsure[9]	wanting	

4.2 Public Speaking Skill Assessment

For assessing public speaking skills, we retrain our classifier using six labels shown in Table 12. Following the requirements by Schuller et al. [16], we use non-acted, non-prompted, realistic data with many speakers, using all obtained data. An experienced speech coach was asked to label 124 one-minute-long samples of natural audio from 31 people (13 female and 18 male) attending speech coaching sessions. The chosen six labels are the ones that the professional is accustomed to using when assessing the public speaking skills of clients. The samples are labelled on a scale 4–10 for each class. We then divided the samples of classes into higher and lower halves according to the score. The upper half represents a positive detection of the class (e.g. *clear*), and the lower half represents a negative detection (e.g. *not clear*).

4.3 Support Vector Machines

Several potential classifiers were investigated. In previous work on emotion recognition from speech on the Mind Reading corpus [5], support vector machines (SVMs) and tree algorithms such as C4.5 were found to be effective. In our experiments SVMs gave the most promising results.

SVMs create a model by constructing an N-dimensional hyperplane that optimally separates data

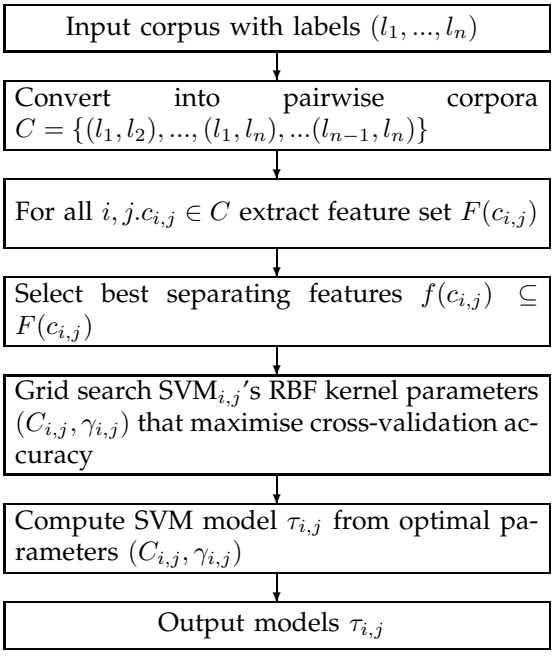


Fig. 2. The training system architecture. SVM$_{i,j}$ represents the support vector for comparing label l_i with l_j.

into two categories. We use a modified version of SVMs [38] that allows for mislabelled examples by choosing a hyperplane as cleanly as possible even if there is no hyperplane that can split the two classes.

We use a non-linear classifier, replacing the linear dot product by a kernel function that transforms the original input space into a higher-dimensional feature space, allowing the SVM to potentially better separate the two classes. After trialling several possible kernel function candidates, the Radial Basis Function (RBF) kernel was found to yield the most promising results. To generalise SVMs to more than two classes, pairwise classification is used. This reduces a single multiclass problem into multiple binary problems by building a classifier for each pair of classes, using only instances from two classes at a time.

4.4 Training

The training system architecture is shown in Figure 2. Its main components are discussed below.

4.4.1 Noise reduction

The Mind Reading corpus used for emotion detection was recorded with high-quality equipment and is largely noise-free. However, the Speech Tutor corpus that is used for public speaking skill assessment was recorded in different environments and contained background noise. To avoid the noise affecting the feature extraction and the construction of the hyperplanes, it was necessary to remove noise from the corpus.

A number of noise reduction algorithms, such as Power subtraction [39] and Time frequency block thresholding [40] were tried. The former models the speech signal as a random process to which uncorrelated random noise is added. The noise is measured during a silence period in the speech, and the estimated power spectrum of the noise is then subtracted from the noisy input signal. This method resulted in an artefact which sounds like random musical notes caused by narrowband tonal components that appeared in unvoiced sound and silence regions after the noise reduction.

Time frequency block thresholding dynamically adjusts spectrogram filter parameters using the Stein risk estimator, which gives an indication of the estimator's accuracy. It was found to eliminate the musical noise of the Power subtraction method, and was thus used as a pre-processing filter for the Speech Tutor corpus.

4.4.2 Feature Extraction

For this work, the openSMILE [26] feature extraction algorithms are used. OpenSMILE provides sound recording and playback via the open-source PortAudio library, echo cancellation, windowing functions, fast Fourier transforms and autocorrelation. Moreover, it is capable of extracting features such as pitch, loudness, energy, mel-spectra, voice quality, mel-spectrum frequency coefficients, and can calculate various functionals such as means, extremes, peaks, percentiles and deviations with a Real-Time Factor $\ll 1$. The ten most commonly used class-differentiating features are shown in Table 2. These consist of functionals calculated for two main categories.

4.4.3 Feature Selection

Since a large feature set will be extracted from the speech, it is expected that there are some irrelevant and redundant data that will not improve the SVM prediction performance. Classification algorithms are unable to attain high classification accuracy if there is a large number of weakly relevant and redundant features, a problem known as the *curse of dimensionality* [41]. Algorithms also suffer from computational load incurred by the high dimensional data.

Our approach is to use the predefined openSMILE set `emo_large` with 6552 features, and pick the most relevant ones using feature selection. For choosing relevant features, the Correlation-based Feature Selection (CFS) algorithm [42] is used. It uses a heuristic based on the assumption that good feature sets contain features highly correlated with the class and uncorrelated with each other. It defines the score for a feature subset S as

$$Merit_S = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}}$$

where k is the number of features in S, r_{ff} is the average feature-feature intercorrelation and r_{cf} is

TABLE 2
Ten most used features in emotion detection. The first column shows the number of pairwise machines in the total $\binom{9}{2} = 36$ that used the feature. All features are smoothed with a moving average filter. MFCC denotes Mel frequency cepstral coefficient, MFSM denotes Mel frequency spectrum magnitude and FFT denotes Fast Fourier Transform.

SVMs	Feature name	Description
12	mfcc_sma[12]_range	Range of MFCC 12
9	pcm_Mag_fband250-650_sma_de_centroid	Centre of gravity of 1st order delta coefficient for FFT magnitude in band 250–650 Hz
8	pcm_Mag_melspec_sma_de_de[4]_quartile3	75% percentile of 2nd order delta for 4th band (317–416 Hz) of MFSM
8	mfcc_sma_de_de[4]_qregerrQ	Quadratic error between 2nd order delta for MFCC 4 and its quadratic regression line
8	mfcc_sma[8]_range	Range of MFCC 8
7	pcm_Mag_melspec_sma_de[2]_quartile2	50% percentile of 1st order delta for 2nd band (143–226 Hz) of MFSM
7	mfcc_sma[4]_minameandist	Difference between arithmetic mean and minimum value of MFCC 4
7	mfcc_sma[4]_iqr1-2	Difference between 50% and 25% percentiles of MFCC 4
6	pcm_Mag_melspec_sma_de_de[4]_iqr2-3	Difference between 75% and 50% percentiles of 2nd order delta for 4th band (317–416 Hz) of MFSM
6	pcm_Mag_melspec_sma_de_de[19]_minPos	Absolute position (frame) of the minimum value of 2nd order delta for 19th band (3423–3827 Hz) of MFSM

the average feature-class correlation. After discretising the features, CFS calculates feature-class and feature-feature correlations using a symmetric form of information gain

$$H_{sym}(X;Y) = \frac{2I(X;Y)}{H(X)+H(Y)}$$

for random variables X, Y where

$$H(X) = -\sum_{x\in X} p(x)\log_2 p(x)$$

$$H(X|Y) = -\sum_{y\in Y} p(y)\sum_{x\in X} p(x|y)\log_2 p(x|y)$$

with $H(X)$ representing the entropy of X and $H(X|Y)$ the entropy of X after observing Y. The information gain is

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

$H_{sym}(X;Y)$ fixes the problem of $I(X;Y)$ assigning a higher value for features with a greater number of values although they may be less informative. It also normalises the values to the range $[0, 1]$.

After calculating the correlations, CFS starts with an empty set of features and applies forward best first search, terminating when it encounters five consecutive fully-expanded subsets that show no improvement.

4.4.4 Scaling

After the feature selection stage, the features $f(c_{i,j}) \subseteq F(c_{i,j})$ are scaled from $\mathbb{R}$ to $[-1, +1]$. This is done to

1) Prohibit attributes in a greater numeric range from dominating those in smaller ranges.
2) Avoid numerical difficulties during the calculation, such as division by large $f(c_{i,j})$.

4.4.5 Grid Search

When using the Radial Basis Function SVM kernel, it is important to choose a suitable penalty for mislabelled examples C and the exponentiation constant γ. Because the optimal values are model-specific, a search algorithm is needed for finding a near-optimal set of values. The optimisation is done on the training data with testing data kept unseen.

The goal is to identify good (C, γ) values so that the classifier can accurately predict unseen testing data, rather than choosing them to maximise prediction accuracy for the training data whose labelling is already known. In this work we use v-fold cross-validation. The training set is divided into v equal-sized subsets, with each subset sequentially tested used a classifier trained on the remaining $v - 1$ subsets.

We use a GRID SEARCH algorithm (Algorithm 1) that sequentially tries pairs of (C, γ) in a given range, and picks the one with the highest cross-validation accuracy. Exponentially growing sequences worked well in practice, confirming findings in previous research [43]. The algorithm is run recursively on a shrinking area.

Once optimal (C, γ) are determined, final SVM models are computed for all pairwise corpora using the LibSVM [44] library.

4.5 Classification

The real-time classification system architecture is shown in Figure 3. Its main components are discussed below.

4.5.1 Segmentation

Real-time analysis of speech requires segmenting the audio. One approach to the problem would be to just process every 1 second separately. However, this would create two problems.

1) The single second may only contain silence, but the SVMs will still need to make a binary decision between two classes by mapping the features on one side of the hyperplane.

Algorithm 1 Grid search algorithm. The algorithm is run by giving the mislabelling penalty C and exponentiation constant γ ranges initial values. Variable n_{steps} defines how fine-grained the grid search is.

GRID-SEARCH($[C_{low}, C_{high}, C_{step}]$,$[\gamma_{low}, \gamma_{high}, \gamma_{step}]$, n_{steps})

1) Initialise $P = 0$, $C_{opt} = 0$, $\gamma_{opt} = 0$.
2) For all $i \in \mathbb{Z}^+ \cup \{0\}$ such that $C_i = (C_{low} + iC_{step}) \leq C_{high}$
   a) For all $j \in \mathbb{Z}^+ \cup \{0\}$ such that $\gamma_j = (\gamma_{low} + j\gamma_{step}) \leq \gamma_{high}$
      i) Divide training model into v equal subsets (for predefined v).
      ii) Sequentially classify one subset using a classifier trained on the remaining $v-1$ subsets using parameters $C = 2^{C_i}$ and $\gamma = 2^{\gamma_i}$.
      iii) If $P < (n_{correct}/v)$, set $P = (n_{correct}/v)$, $C_{opt} = C_i$, $\gamma_{opt} = \gamma_j$.
3) If $n_{steps} \leq 0$, return (C_{opt}, γ_{opt}).
4) Else GRID-SEARCH($[C_{opt} - n_{steps}, C_{opt} + n_{steps}, C_{step}/2]$,$[\gamma_{opt} - n_{steps}, \gamma_{opt} + n_{steps}, \gamma_{step}/2]$, $n_{steps} - 1$).

2) Emotions may not be expressible within a short time interval. Furthermore, some features have a temporal characteristic which will not be extractable if a single segment length is used.

As a result it was necessary to choose the segment length dynamically, approximating to one segment per sentence. The signal energy could be used for differentiating between silence and speech. The choice was between a simple algorithm that uses static thresholds and a more complex algorithm that implements dynamic thresholding. In accordance with the iterative development model, a simple static algorithm was implemented first. By adding complexity in layers, we could at each step check that performance sufficient for real-time operation was achieved.

The SEGMENTATION algorithm (Algorithm 2) achieves this by defining three thresholds. First, the silence threshold η defines the threshold for the energy $E = \sum_i^n |s_i|^2 > \eta$, for signals s_i in frame of size n. Second, ρ_{start} sets the number of frames with energy above η that are required until a segment start is detected. Third, ρ_{end} is the number of frames below η until a segment end is detected. From the start to the end of a segment, the features are extracted. At the end the pairwise SVMs are run in parallel.

It turned out that the segmentation results of this algorithm were more than sufficient for detecting pauses in the speech. In practice, separate η could be used for a quiet room and for a noisier environment. In our experiments we set ρ_{start} and ρ_{end} to 10 and 40 frames respectively.

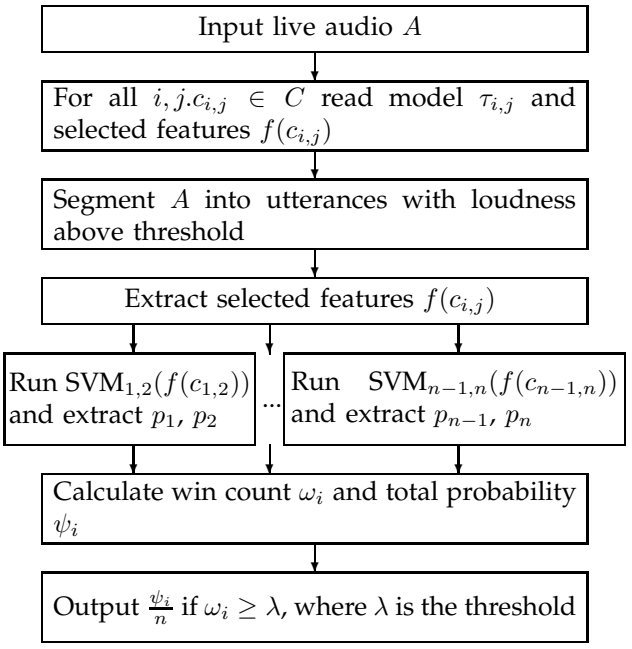


Fig. 3. The real-time classifier architecture. SVM$_{i,j}$ computes the probabilities p_i and p_j for labels i, j, using features $f(c_{i,j})$.

Algorithm 2 Audio segmentation algorithm. η is the silence threshold based on the energy of the signal. ρ_{start} and ρ_{end} specify the thresholds for the number of frames that need to be above and below η for a segment to start and end, respectively.

SEGMENTATION(η, ρ_{start}, ρ_{end})

1) Initialise $C_{start} = C_{end} = 0$
2) Repeat
   a) If $\sum_i^n |s_i|^2 > \eta$
      i) Set $C_{end} = 0$
      ii) Increment C_{start}
      iii) If $C_{start} > \rho_{start}$
         A) Send `start_inference` message to all pairwise SVMs.
   b) Else
      i) Set $C_{start} = 0$
      ii) Increment C_{end}
      iii) If $C_{end} > \rho_{end}$
         A) Send `end_inference` message to all pairwise SVMs.

4.5.2 Support Vector Machines

Once the audio is segmented and the features are extracted, the $n(n-1)/2$ pairwise machines can be run in parallel to predict the class for a segment. The hyperplanes separate two emotion classes in an $|f|$-dimensional space, where f is the set of features being considered. Implementation-wise, this required

preparing the features for interfacing with the LibSVM library and then using the results to do further processing. The LibSVM library provides a highly optimised algorithm for solving the Lagrange multiplier optimisation problem for SVMs.

The RUN-SVM algorithm (Algorithm 3) describes this process. First, all but the features selected by the correlation-based feature selection algorithm needed to be filtered away from the model files. The algorithm then waits for extracted features from openSMILE. Once it receives these, it scales them by the same ratio as used in the training phase. It then predicts probabilities for both labels by calling the LibSVM C++ library, and sends the results paired with the labels to the PAIRWISE-COMBINATOR module.

Algorithm 3 Run-SVM algorithm. τ is the model file, (l_i, l_j) are the pairwise class labels, S is the scaling file and ζ is the feature selection file. These are computed in the training phase.

RUN-SVM($\tau, (l_i, l_j), S, \zeta$)

1) Load τ, (l_i, l_j), S and ζ from files
2) Filter away all features except ζ
3) Repeat
   a) Receive features $f_i \in \zeta$ from openSMILE components
   b) Apply scaling S used in training
   c) Predict probabilities (p_i, p_j) by applying LibSVM on (τ, f)
   d) Send $[(l_i, p_i), (l_j, p_j)]$ to PAIRWISE-COMBINATOR

4.5.3 Pairwise Classification

When the pairwise SVMs have been run, their results need to be combined so that the label of the segment can be predicted. This glues together the SVMs running in parallel and the voting algorithm.

The PAIRWISE-COMBINATOR algorithm (Algorithm 4) achieves this by receiving the results from the $\frac{1}{2}n(n-1)$ pairwise SVMs running in parallel. It keeps count of the labels of the winning classes. Once it has received all pairwise classification results for a segment and computed the number of wins ω_i for each label i, it resets the wins for the next frame and runs a pairwise voting algorithm that determines the winning class. The voting is described in the next section.

4.5.4 Pairwise Fusion Mechanism

In order to determine the most probable class, the probabilities of the multiple binary classifiers are fused.

We propose a fusion method for determining co-occurring emotions. Whereas in traditional single-label classification a sample is associated with a single

Algorithm 4 Pairwise combinator algorithm. n is the number of labels. As there are $\frac{1}{2}n(n-1)$ pairwise machines, each label should receive $n-1$ probabilities.

PAIRWISE-COMBINATOR(n)

1) Initialise $P = \{S_1, ..., S_n\}$ with $\forall i.S_i = \{\}$, $W = \{\omega_1, ..., \omega_n\}$ with $\forall i.\omega_i = 0$
2) Repeat
   a) If $\forall i.|S_i| < n-1$
      i) Receive $[(l_i, p_i), (l_j, p_j)]$ from RUN-SVM
      ii) Insert p_i into S_i and p_j into l_j
   b) Else
      i) For all i set $\omega_i = \sum_{p \in S_i} g(p)$ where $g(p) = \begin{cases} 1 & \text{for } p \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$
      ii) Run PAIRWISE-VOTING(P, W) and set $\forall i.S_i = \{\} \wedge \omega_i = 0$

label l_i from a set of disjoint labels L, multi-label classification associates each sample with a set of labels $L' \subseteq L$. A previous study concluded that the use of complex non-linear fusion methods yielded only marginal benefits (0.3%) over linear methods when used with SVMs [45]. Therefore, three linear fusion methods are implemented:

1) Majority voting using wins from binary classifiers.
2) Maximum combined probability from binary classifiers.
3) Binary classification wins above a threshold.

In the first method we consider all $n-1$ SVM outputs per class as votes and select the class with most votes. Assuming that the classes are mutually exclusive, the *a posteriori* probability for feature vector $\mathbf{f}$ is $p_i = P(\mathbf{f} \in class_i)$. The classifier $\text{SVM}_{i,j}$ computes an estimate $\hat{p}_{i,j}$ of the binary decision probability

$$p_{i,j} = P(\mathbf{f} \in class_i | \mathbf{f} \in class_i \cup class_j) \quad (1)$$

between classes i and j. The final classification decision $\hat{D}_{voting}$ is the class i for which

$$\hat{D}_{voting} = \arg \max_{1 \leq i \leq n} \sum_{j \neq i} g(\hat{p}_{i,j}) \quad (2)$$

where

$$g(p) = \begin{cases} 1 & \text{for } p \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

Ties are solved by declaring the class with higher probability to be the winner.

In the second method, the maximum probability $\psi_i = \sum_{p \in S_i} p$ of the binary SVMs is determined. The winner of decision $\hat{D}_{probability}$ is i such that

$$\hat{D}_{probability} = \arg\max_{1\leq i\leq n} \sum_{j\neq i} \hat{p}_{i,j}. \quad (4)$$

Finally, for detecting co-occurring emotions, the classes are ranked according to the number of wins. The classes with wins above a threshold λ are returned, with the classification decision $\hat{D}_{threshold}$ being the set of classes

$$\hat{D}_{threshold} = \{i \mid \sum_{j\neq i} g(\hat{p}_{i,j}) \geq \lambda\}. \quad (5)$$

The PAIRWISE-VOTING algorithm (Algorithm 5) combines these three approaches. An experiment comparing the results using the three methods is detailed in the evaluation section. The total probabilities ψ_i of probabilities p_i are calculated for each label l_i and normalised to $norm(\psi_i)$ according to the number of labels. The resulting label and its normalised probability is sent to the console output when the wins for the label are above the threshold λ.

Algorithm 5 Pairwise voting algorithm. μ is the mean win count and σ is the standard deviation. The threshold definition follows previous research [5].

PAIRWISE-VOTING(P, W)

1) Set win threshold $\lambda = \lfloor(\mu + \sigma)(n-1)\rfloor$
2) For each l_i with $i \leq n$ using $p_i \in S_i$, $S_i \in P$ and $\omega_i \in W$
   a) Set the total probability $\psi_i = \sum_{p\in S_i} p$ with $norm(\psi_i) = \frac{\psi_i}{n-1}$
   b) If $\omega_i \geq \lambda$
      i) Send $(l_i, norm(\psi_i))$ to console output

We set $\lambda = \lfloor(\mu + \sigma)n\rfloor$ where μ is the mean win count, σ is the standard deviation and n is the class cardinality to allow comparison with Sobol Shikler [5]. By the central limit theorem, the distribution of a sum of many independent, identically distributed random variables (RVs) tends towards the normal distribution. By assuming that the SVMs exhibit such RVs, and since for the normal distribution $\mu + \sigma \approx 0.841$, $\lambda = \lfloor 0.841(n-1)\rfloor$. In particular, for the 9 classes chosen for evaluation, $\lambda = 6$.

4.6 Method for Assessing Public Speaking Skills

In this subsection we describe the changes in the emotion classifier that enabled the classifier to successfully analyse public speaking skills.

One binary SVM per class is used to derive a class-wise probability. If a pairwise approach similar to that in emotion classification had been used, the same samples would have existed in several classes, making separating the classes intractable. As a result, unlike in emotion detection where the most prominent labels describing the speech are selected, for speech quality assessment all classes are detected, each labelled with a probability. This allows users to attempt to maximise all class probabilities, a goal which is more useful for speech coaching. As for emotion recognition, Radial Basis Function kernel parameter in the SVMs were optimised using Algorithm 1.

The data were recorded in different environments with varying background noise level. Therefore an additional challenge was to normalise the data to avoid learning to only recognise speakers. As explained in Section 4.4.1, we successfully used time frequency block thresholding to reduce the noise level without introducing artefacts common in other methods.

As for emotion recognition, the Correlation-based Feature Selection algorithm was used to select the best discriminating features. The ten most commonly used class-differentiating features are given in Table 3. As in emotion recognition features shown in Table 2, features based on computations of Mel frequency cepstral coefficients and Mel frequency spectrum magnitudes dominate the list. In particular, Mel frequency cepstral coefficients were very commonly used. In addition, a functional calculated on the zero-crossing rate of the time signal was also commonly used for discrimination.

5 EVALUATION

In this section we evaluate the overall classification results.

5.1 Grid Search for SVM Parameter Optimisation

In previous studies optimisation of the machine learning algorithm has not received much attention. This study employs a method based on maximising cross-validation accuracy in training data to obtain a considerable improvement in recognition accuracy. The experiment below demonstrates how this method improves the accuracy of the emotion classifier.

The greedy grid search algorithm chooses optimal (C, γ) parameters for each pairwise SVM. It first does a rough search over the values and then recursively narrows down the search space by searching around the values that produced the highest cross-validation accuracy. This turned out to be a major contributor to the high accuracy of the SVM approach. Previous work has ignored this subtle but clearly important classifier optimisation.

The effect for using grid search with the three pairwise fusion mechanisms is shown in Table 4. A significant improvement, between 10% and 25%, is observed. This is as high an improvement as that gained from choosing SVM over C4.5. As the optimisation maximises the cross-validation accuracy instead of the training data classification accuracy, the optimisation did not result in overfitting of the model. The optimisation is done on the training data, with the testing data kept unseen.

TABLE 3
Ten most used features in public speaking skill assessment. The first column shows the rank of the feature. The rank is derived from the number of classes that used the feature in a leave one speaker out experiment. All features are smoothed with a moving average filter. MFCC denotes Mel frequency cepstral coefficient and MFSM denotes Mel frequency spectrum magnitude.

Rank	Feature name	Description
1	mfcc_sma[6]_percentile98.0	98% percentile of MFCC 6
2	mfcc_sma[3]_zcr	Zero-crossing rate of MFCC 3
3	mfcc_sma_de_de[8]_skewness	3rd order moment of 2nd order delta for MFCC 8
4	pcm_zcr_sma_iqr1-3	Difference between 75% and 25% percentiles of zero-crossing rate of the time signal
5	mfcc_sma_de_de[12]_meanPeakDist	Mean distance between peaks of 2nd order delta for MFCC 12
6	mfcc_sma[4]_percentile95.0	95% percentile of MFCC 4
7	mfcc_sma[3]_nzgmean	Geometric mean for non-zero absolute values of MFSM 3
8	pcm_Mag_melspec_sma[4]_meanPeakDist	Mean distance between peaks of 4th band (317–416 Hz) of MFSM
9	mfcc_sma_de_de[1]_meanPeakDist	Mean distance between peaks of 2nd order delta for MFCC 1
10	pcm_Mag_melspec_sma_de[0]_skewness	3rd order moment of 1st order delta for 0th band (0–68 Hz) of MFSM

TABLE 4
Detection accuracies in percentages with a 70–30% training/testing split for the three fusion methods, with and without grid search.

Type of data	Threshold	Max probability	Max wins
Grid search	86	72	70
No grid search	76	47	48

5.2 Real-time Performance

The experiment below demonstrates that the classifier achieves real-time performance for common sentence lengths.

The average latency in milliseconds of the classification stage is shown in Figure 4. It was measured as the time between the detection of the end of a segment and the output of the result. As shown in the figure, normal sentences (1–15 s) are classified in 0.046–0.110 s, making the delay barely noticeable. Improving upon Sobol Shikler's inference solution [5], this allows real-time classification.

The increase of latency with longer sentences is caused by the feature extractors that need to process more frames with longer audio segments. However, since speakers need to pause to breath, and thus limit the segment length, this increase in latency did not cause problems in practice.

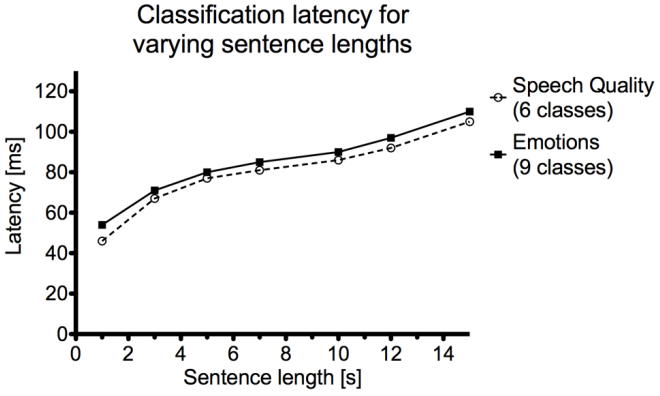


Fig. 4. Average live classification latency in milliseconds of 50 runs on a dual-core 2.66 GHz PC with 4 GB RAM.

5.3 Different Classifiers

This experiment aims to compare the accuracy of the three machine learning algorithms that were found to achieve highest accuracies. A wide range of different classifiers were tried, with SVMs and decision tree-based C4.5 and Random Forest performing best.

C4.5 constructs a decision tree from a set of data by dividing up the data according to the information gain $I(X;Y)$. It recursively splits the tree by the attribute with the highest $I(X;Y)$ in the training, yielding a decision tree that can be reused for classification.

Random Forest builds a set of classification trees. Each tree is created by taking a bootstrap sample from the training data. The best attribute for the split is selected from an arbitrarily chosen subset of attributes. The classification result is derived through majority voting by the tree classifiers.

Unlike in previous research [5], the experiment found that SVMs could provide significantly higher performance than from C4.5 for every pairwise machine. This is achieved by using grid search to optimise the SVM parameters. The results are illustrated in Table 5, where the SVM results are compared to the results with C4.5 and Random Forest. In three cases, Random Forest outperforms the optimised SVM.

Another factor that affected the results is feature selection. The square-bracketed values in Table 5 show the number of features in each SVM. These are only a small fraction of the original 6669 feature combinations extracted by openSMILE. When training on all 6669 features, accuracies above 60% were rarely obtained.

TABLE 5
Ten-fold cross-validation percentages using grid search-optimised SVMs, C4.5 and Random Forest. The numbers without brackets are the results using SVMs. The results using C4.5 are shown in parentheses. The results using Random Forest are in curly brackets. The number of features used are in square brackets. The numbers in bold represent the highest accuracy for the pairwise machines.

	absorbed	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed		**93** (79) {89} [53]	87 (74) {**93**} [55]	**96** (80) {90} [79]	**96** (85) {85} [46]	**89** (80) {**89**} [58]	**85** (74) {**85**} [39]	82 (76) {**86**} [52]	**84** (79) {83} [36]
excited			**90** (72) {82} [44]	**84** (70) {76} [38]	81 (77) {**84**} [48]	**80** (69) {77} [31]	**94** (70) {76} [47]	**90** (78) {89} [97]	**87** (69) {80} [57]
interested				**92** (81) {86} [65]	**92** (72) {85} [54]	**91** (68) {78} [46]	**90** (77) {86} [68]	**90** (79) {84} [85]	**85** (69) {73} [34]
joyful					**86** (77) {82} [53]	**85** (70) {77} [54]	**99** (75) {84} [75]	**95** (84) {91} [103]	**92** (73) {80} [62]
opposed						**93** (75) {81} [43]	**91** (73) {82} [28]	**94** (75) {85} [86]	**92** (76) {84} [28]
stressed							**86** (68) {81} [61]	**88** (74) {83} [100]	**86** (68) {78} [34]
sure								**94** (80) {89} [78]	**88** (70) {82} [54]
thinking									**90** (74) {81} [72]
unsure									

5.4 Different Feature Selection Algorithm

This experiment compares the accuracy of two commonly used feature selection algorithms, the filter-based Correlation-based Feature Selection (CFS) and wrapper-based Dynamic Oscillating Search (DOS) [46] methods.

Wrapper feature selection uses a search algorithm to search through the space of possible features and evaluates each feature subset by running the classifier on the subset. In contrast, filter feature selection evaluates feature subsets by classifier-independent criteria. DOS is a relatively new search method for wrapper-based feature selection that has been shown to outperform many other methods [46]. It extends the principle of Oscillating Search [47] which repeatedly modifies the current feature subset of a pre-determined size. Dynamic Oscillating Search determines the feature subset size automatically. Importantly, DOS performs efficiently on large feature sets. Exhaustive wrapper search of subsets for the 6552 extracted features would be very time-consuming.

The results are illustrated in Table 6, where the CFS results are compared to the results with DOS feature selection. As shown in the table, CFS outperforms DOS in all but one case. CFS selects on average over ten times more features than DOS. As the feature sets chosen by CFS still allowed real-time performance, CFS was selected for the final implementation.

TABLE 6
Ten-fold cross-validation percentages using grid search-optimised SVMs with CFS and DOS feature selection algorithms. The numbers without brackets are the results using CFS feature selection. The results using DOS are shown in parentheses. The number of features used by DOS are in square brackets. The numbers in bold represent the highest accuracy for the pairwise machines.

	absorbed	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed		**93** (82) [4]	**87** (85) [4]	**96** (83) [4]	**96** (86) [4]	**89** (85) [4]	**85** (70) [5]	82 (**89**) [4]	**84** (77) [2]
excited			**90** (79) [6]	**84** (68) [4]	**81** (67) [4]	**80** (65) [4]	**94** (82) [5]	**90** (76) [5]	**87** (60) [5]
interested				**92** (83) [5]	**92** (61) [6]	**91** (67) [5]	**90** (78) [5]	**90** (87) [7]	**85** (67) [5]
joyful					**86** (76) [5]	**85** (81) [7]	**99** (80) [6]	**95** (82) [6]	**92** (72) [7]
opposed						**93** (68) [5]	**91** (59) [3]	**94** (72) [6]	**92** (68) [6]
stressed							**86** (70) [6]	**88** (83) [8]	**86** (65) [5]
sure								**94** (90) [8]	**88** (64) [5]
thinking									**90** (73) [8]
unsure									

5.5 Ten-fold Cross Validation Results

The experiment below demonstrates the ten-fold cross-validation results of the emotion classifier and compares them to previous research using the same training data [5].

The ten-fold cross-validation results for the pairwise SVMs are shown in Table 7. All accuracies are greater than the values obtained in previous research using the same classes and corpus. The results are constantly above 80%, in contrast to the lower bound 60% obtained previously.

TABLE 7
The ten-fold cross-validation accuracy for pairwise SVMs in percentages. The average accuracy is 89%. Sobol Shikler's results [5] are in parentheses.

	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed	**93** (81)	**87** (82)	**96** (82)	**96** (78)	**89** (87)	**85** (84)	**82** (73)	**84** (64)
excited		**90** (71)	**84** (60)	**81** (71)	**80** (61)	**94** (83)	**90** (72)	**87** (75)
interested			**92** (77)	**92** (75)	**91** (66)	**90** (78)	**90** (84)	**85** (72)
joyful				**86** (71)	**85** (61)	**99** (83)	**95** (72)	**92** (75)
opposed					**93** (84)	**91** (72)	**94** (81)	**92** (79)
stressed						**86** (84)	**88** (75)	**86** (78)
sure							**94** (75)	**88** (78)
thinking								**90** (89)

5.6 Different Fusion Methods

The experiment below demonstrates the accuracies for the three fusion methods proposed in Section 4.5.4.

A summary of the accuracies for the three different fusion methods is shown in Table 8. The average accuracies are higher than or equal to the results achieved previously on the same corpus [5]. The average accuracy of the maximum probability fusion technique is higher than that achieved by majority voting (72% vs 70%). However, for some classes the majority voting accuracy is higher (e.g. *stressed* and *interested*). A higher average accuracy could be achieved by combining these methods. In future work, more advanced fusion methods such as the ensemble classification presented by Schuller et al. [27] and the tree-based approach by Lee et al. [17] will be investigated.

Confusion matrices for fusion using thresholding, maximum probability and majority voting are shown in Tables 9, 10 and 11 respectively. Inspection of the confusion matrices reveals that some classes are better detected than others. The classes *opposed* and *sure* present the lowest values using any method. This is reflected by the lower number of training samples (38 and 53 samples, compared to the average of 61) resulting from the categorisation choice to allow comparison to Sobol Shikler [5]. Similarly, the class with most samples (*joyful*, 94 samples) is most frequently mistaken to be the correct class. In future work classes with equal numbers of training samples could be used.

TABLE 8
Accuracies in percentages for the three fusion methods. Sobol Shikler's results [5] are shown in parentheses. 2.5 classes were inferred on average with a threshold $\lambda = 6$. The reported results are averages over 10 random 70–30% partitions.

Type of data	Threshold	Max probability	Voting
70–30% train/test split	**86** (79)	72	70
Cross-validation on 70% training split	**99** (81)	86	88

As expected, the thresholding fusion method for co-occurring emotion classification yields highest detection accuracies since several classes can be selected at a time. This, however, also leads to much higher confusion values because of the assumption that more than one emotion can be occurring simultaneously. For example, as shown in Table 9, samples labelled *excited* are detected as *joyful* in 35% of cases, compared to a correct detection rate of 85%. Since more than one class is selected at a time, the rows do not add to 100 like in conventional confusion matrices. In Tables 10 and 11 the rows do not add up to 100 due to rounding. It is likely that some high confusion rates are caused by the overrepresentation of certain classes.

5.7 Evaluation of Public Speaking Skill Assessor

This experiment evaluates the accuracy of the public speaking skill assessor.

The results of public speaking skill assessment are shown in Table 12. All classes can be accurately detected in 10-fold cross-validation. The classes *competent* and *dynamic* present slightly lower detection accuracies. Overall, however, the speech quality assessment cross-validation accuracies are high (average 81%).

We also provide evaluation data using a speaker-independent evaluation on unseen test data. The system was separately trained for 31 partitions, each in which one out of the 31 speakers was left out. Each partition's classification performance was then tested

TABLE 9
Matrix with confusion values for thresholded pairwise fusion. Row headings show ground truth and columns show inferences. Average accuracy is 86%. A random choice would result in 11% accuracy. Leave one speaker out average accuracy is 50%.

	absorbed	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed	93	15	22	15	0	15	11	48	48
excited	4	85	2	35	14	60	19	15	8
interested	15	10	83	21	6	31	6	42	52
joyful	0	29	14	91	22	56	4	19	24
opposed	0	27	3	41	73	51	16	24	22
stressed	4	46	10	39	11	92	9	19	31
sure	12	24	11	22	17	31	74	28	26
thinking	23	6	17	23	7	24	11	93	56
unsure	24	14	14	22	8	29	9	56	91

TABLE 10
Confusion matrix using max probability for pairwise fusion. Row headings show ground truth and columns show inferences. Average accuracy is 72%. A random choice would result in 11% accuracy. Leave one speaker out average accuracy is 31%.

	absorbed	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed	74	0	4	4	0	4	0	7	7
excited	0	75	0	10	2	8	0	0	4
interested	2	2	69	6	0	6	2	8	6
joyful	0	6	0	79	2	3	2	3	4
opposed	0	0	0	16	62	8	5	0	8
stressed	1	2	2	11	1	67	2	4	8
sure	2	6	2	4	2	9	63	11	2
thinking	1	0	3	3	0	1	0	86	6
unsure	1	1	1	4	0	8	0	17	68

on the speaker that was left out. The results shown in Table 12 present the average classification accuracy for the speakers that were left out.

As can be seen from the data, the classes *clear*, *credible*, *dynamic* and *persuasive* can all be successfully detected with greater than 60% accuracy in the leave-one-speaker-out test. However, *competent*, which gave high 10-fold cross-validation results, obtains low leave-one-speaker-out results. This shows the large extent to which speaker-dependency may skew the cross-validation results. In particular, the lower accuracy could indicate that the class is a more subjective quality and hence is more difficult to classify. The class *clear*, for example, which achieved the highest leave-one-speaker-out accuracy, could intuitively be easier to assess than whether a speaker sounds *competent*.

TABLE 11
Confusion matrix using majority voting for pairwise fusion. Row headings show ground truth and columns show inferences. Average accuracy is 70%. A random choice would result in 11% accuracy. Leave one speaker out average accuracy is 29%.

	absorbed	excited	interested	joyful	opposed	stressed	sure	thinking	unsure
absorbed	74	0	4	4	0	4	0	7	7
excited	0	65	0	13	2	10	4	2	4
interested	0	3	73	4	0	4	2	8	6
joyful	0	6	2	76	5	3	2	3	3
opposed	0	3	0	16	59	8	5	1	8
stressed	1	2	2	10	1	71	2	4	7
sure	2	6	4	4	2	7	63	10	2
thinking	2	0	3	1	1	3	0	83	7
unsure	3	1	0	5	3	8	0	16	64

The number of training samples did not correlate with the accuracy. Even though the class *clear* achieved the highest accuracy, with more samples than average, class *pleasant* achieved lower accuracy with even more training samples. The weighted average accuracy is the same as the unweighted average accuracy 61%.

Overall, most of the classes are well detected even in a speaker-independent evaluation. This is a promising result, and indicates that objective computerised analysis of public speaking skills is feasible.

6 Conclusion

We have presented a framework for real-time classification of co-occurring emotions in speech whose accuracy outperforms previous work using the same corpus [5]. We have also shown that the novel application of the system for assessing public speaking skills achieves high classification accuracy.

The emotion classification framework consists of $n(n-1)/2$ pairwise SVMs for n labels, each with a differing set of features selected by the Correlation-based Feature Selection algorithm. The classifier was trained using the Mind Reading corpus of acted speech.

We demonstrated a considerable improvement in classification accuracy from optimising the misclassification and exponentiation coefficients (C, γ) using

TABLE 12
Detection accuracies in percentages for assessing public speaking skills. A random choice would result in 50% accuracy. The number of features is given in square brackets.

Class	10-fold cross-validation	Leave one speaker out	Training samples
clear	80 [53]	72 [50]	66
competent	74 [32]	47 [36]	49
credible	80 [23]	64 [22]	42
dynamic	77 [41]	64 [38]	45
persuasive	82 [17]	62 [15]	79
pleasant	93 [43]	57 [43]	73
Mean	**81 [35]**	**61 [34]**	**59**

a grid search algorithm. Improvements between 10% and 25% were observed. We further illustrated that the parameter-optimised SVMs outperform tree-based algorithms for most classes in our corpus. This underlines the importance of parameter optimisation, an issue several recent studies in the field have overlooked.

In an experiment on feature selection, we showed that the filter-based Correlation-based Feature Selection (CFS) algorithm outperforms a wrapper-based feature selection with the Dynamic Oscillating Search (DOS) algorithm for our emotion corpus. However, CFS chose ten times more features than DOS. DOS could therefore be a good choice for resource-limited settings where a large set of features presents an unnecessarily high overhead.

We further applied our emotion classifier to assessing public speaking skills. For this we used our own corpus of non-acted, non-prompted realistic data with 124 one-minute-long samples from 31 people attending speech coaching sessions. We illustrated that most of the classes are well detected even in a speaker-independent evaluation. This is a promising result, and indicates that it is feasible to perform objective computerised analysis of public speaking skills.

Overall, this paper presented a high-accuracy training and classification framework for emotion detection from speech. It also shows a novel application of the classifier that successfully performs real-time assessment of public speaking skills. It can be used for training one's public speaking skills, and for assisting or even replacing a human speech coach.

7 Future Work

At present, our system is the most accurate classifier trained on the Mind Reading corpus known to the authors. It is also the only system known to us which is able to provide automatic feedback on public speaking skills. In the near future, we plan to apply it to a number of tasks. First, we intend to investigate how the system could be used as an assistant for a professional speech coach. The system could help the learner both in-session and at home by providing instant feedback on the speaking skills, and could provide more objective and consistent analysis. Second, we will investigate the use of emotion classification for speech coaching. To be persuasive, it is critical to non-verbally show the emotion that is consistent with the verbal content. The system trained on the Mind Reading corpus could provide useful feedback on this front. Third, we acknowledge that an acted database may not be suitable for all applications. We therefore intend to focus on natural and induced emotional databases in the future. Finally, we plan to investigate the performance of the system together with facial expression analysis. A multi-modal system could potentially achieve more accurate classification for both affect and public speaking skills.

Acknowledgments

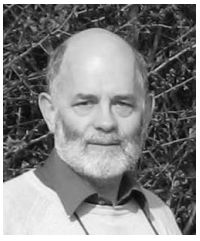
The authors would like to thank a local speech coach who kindly agreed to label her speech tutor recordings and allowed them to be used in this work. We also wish to thank Tal Sobol Shikler, who helpfully located some recordings used for evaluating the system. Finally, we are grateful to Florian Eyben for checking our feature name descriptions, and to Sophia Zhang and the anonymous reviewers for helping to improve the paper.

References

[1] K. R. Scherer, "Vocal affect expression: A review and a model for future research." *Psychological bulletin*, vol. 99, pp. 143–165, 1986.
[2] R. Picard, "Affective Computing: From Laughter to IEEE," *IEEE Transactions on Affective Computing*, vol. 1, pp. 11–17, 2010.
[3] ——, "Affective computing." The MIT Press, 2000.
[4] J. D. Haynes and G. Rees, "Decoding mental states from brain activity in humans," *Nature Reviews Neuroscience*, vol. 7, pp. 523–534, 2006.
[5] T. Sobol Shikler, "Analysis of affective expression in speech," Ph.D. dissertation, Cambridge University, 2007.
[6] T. Pfister and P. Robinson, "Speech emotion classification and public speaking skill assessment," in *Human Behavior Understanding, International Conference on Pattern Recognition*, ser. Lecture Notes in Computer Science, 2010, vol. 6219, pp. 151–162.
[7] E. Uldall, "Attitudinal Meanings Conveyed by Intonation Contours," *Language and Speech*, vol. 3, no. 4, pp. 223–234, October/December 1960.
[8] G. L. Huttar, "Relations between prosodic variables and emotions in normal american english utterances," *The Journal of the Acoustical Society of America*, vol. 41, pp. 1581–1581, 1967.
[9] F. Dellaert, T. Polzin, and A. Waibel, "Recognizing emotion in speech," *Proc. of fourth international conference on spoken language processing*, vol. 3, pp. 1970–1973, 1996.
[10] A. Batliner, K. Fisher, R. Huber, J. Spilker, and E. Noth, "Desperately seeking emotions or: Actors, wizards, and human beings," in *Proc. of the International Speech Communication Association Workshop on Speech and Emotion*, 2000, pp. 195–200.
[11] K. Forbes-Riley and D. Litman, "Predicting emotion from spoken dialogue from multiple knowledge sources," in *Proc. of human language technology conference of the north american chapter of the association for computational linguistics*, 2004.

[12] S. Steidl, M. Levit, A. Batliner, E. Noeth, and E. Niemann, "Of all things the measure is man – Automatic classication of emotions and interlabeler consistency," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal processing*, 2005.
[13] C. Lee, S. Narayanan, and R. Pieraccini, "Combining acoustic and language information for emotion recognition," *Proc. of International Conference on Spoken Language Processing*, pp. 873–876, 2002.
[14] R. Craggs and M. Wood, "A two-dimensional annotation scheme for emotion in dialogue," *Proc. of AAAI spring symposium on exploring attitude and affect in text: Theories and applications*, 2004.
[15] B. Schuller and G. Rigoll, "Recognising interest in conversational speech – comparing bag of frames and supra-segmental features," in *Interspeech*, Brighton, UK, 2009.
[16] B. Schuller, S. Steidl, and A. Batliner, "The Interspeech 2009 emotion challenge," in *Interspeech*, Brighton, UK, 2009.
[17] C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, "Emotion recognition using a hierarchical binary decision tree approach," in *Interspeech*, Brighton, UK, 2009.
[18] P. Dumouchel, N. Dehak, Y. Attabi, R. Dehak, and N. Boufaden, "Cepstral and long-term features for emotion recognition," in *Interspeech*, Brighton, UK, 2009.
[19] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach, "Emotional speech: Towards a new generation of databases," *Speech Communication*, vol. 40, no. 1-2, pp. 33–60, 2003.
[20] S. Abrilian, L. Devillers, S. Buisine, and J. Martin, "EmoTV1: Annotation of real-life emotions for the specification of multimodal affective interfaces," *Proc. of Human-Computer Interaction International*, 2005.
[21] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Proc. of Interspeech, Lissabon*, 2005, pp. 1517–1520.
[22] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The eNTERFACE05 Audio-Visual Emotion Database," *22nd International Conference on Data Engineering Workshops*, vol. 0, p. 8, 2006.
[23] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Hörthker, and H. Konosu, "Being bored? Recognising natural interest by extensive audiovisual integration for real-life application," *Image and Vision Computing*, vol. 27, no. 12, pp. 1760–1774, 2009.
[24] S. Baron-Cohen, O. Golan, S. Wheelwright, and J. J. Hill, *Mind Reading: The Interactive Guide to Emotions*. University of Cambridge: Jessica Kingsley Publishers, 2004, iSBN 1 84310 214 5.
[25] T. Sobol Shikler and P. Robinson, "Classification of complex information: Inference of co-occurring affective states from their expressions in speech," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2009.
[26] F. Eyben, M. Wöllmer, and B. Schuller, "openEAR – Introducing the Munich open-source emotion and affect recognition toolkit," in *Proc. 4th International HUMAINE Association Conference on Affective Computing and Intelligent Interaction 2009 (ACII 2009), IEEE*, Amsterdam, The Netherlands, September 2009.
[27] B. Schuller, S. Reiter, R. Müller, M. Al-Hames, M. Lang, and G. Rigoll, "Speaker independent speech emotion recognition by ensemble classification," in *IEEE International Conference on Multimedia and Expo*, 2005.
[28] S. Scherer, F. Schwenker, and G. Palm, "Classifier fusion for emotion recognition from speech," in *Advanced Intelligent Environments*. Springer US, 2009, pp. 95–117.
[29] B. Schuller, R. Müller, M. Lang, and G. Rigoll, "Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles," in *Eurospeech*, 2005.
[30] J. Russell, J. Bachorowski, and J. Fernandez-Dols, "Facial and vocal expressions of emotion." *Annual Review of Psychology*, pp. 329–350, 2003.
[31] K. Scherer, "Vocal communication of emotion: A review of research paradigms," *Speech communication*, vol. 40, no. 1-2, pp. 227–256, 2003.
[32] K. Scherer, T. Johnstone, and G. Klasmeyer, "Vocal expression of emotion," *Handbook of affective sciences*, pp. 433–456, 2003.
[33] R. Barr, B. Hopkins, and J. Green, *Crying as a sign, a symptom, and a signal*. Mac Keith Press, 2000.
[34] R. Cowie and R. Cornelius, "Describing the emotional states that are expressed in speech," *Speech Communication*, vol. 40, no. 1-2, pp. 5–32, 2003.
[35] R. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *Affective Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 18–37, 2010.
[36] R. Reisenzein, "Broadening the scope of affect detection research," *Affective Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 42–45, 2010.
[37] O. Golan, S. Baron-Cohen, S. Wheelwright, and J. J. Hill, "Systemizing empathy: Teaching adults with asperger syndrome and high functioning autism to recognize complex emotions using interactive multimedia," *Development and Psychopathology*, vol. 18, pp. 589–615, 2006.
[38] V. N. Vapnik, "The nature of statistical learning theory," *Springer*, 1998.
[39] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 208–211, 1979.
[40] G. Yu, S. Mallat, and E. Bacry, "Audio denoising by time-frequency block thresholding," *IEEE Transactions on signal processing*, vol. 56, pp. 1830–1839, 2008.
[41] H. Altun and G. Polat, "New frameworks to boost feature selection algorithms in emotion detection for improved human-computer interaction," in *Advances in Brain, Vision, and Artificial Intelligence*, ser. Lecture Notes in Computer Science, vol. 4729. Springer Berlin / Heidelberg, 2007, pp. 533–541.
[42] M. A. Hall and L. A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper," *Florida Artificial Intelligence Symposium*, 1999.
[43] L. Qing-kun and Q. Pei-wen, "Model selection for SVM using mutative scale chaos optimization algorithm," *Journal of Shanghai University*, vol. 10, pp. 531–534, 2006.
[44] R. Fan, P. Chen, and C. Lin, "Working set selection using second order information for training SVM," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.
[45] S. Pöyhönen, A. Arkkio, P. Jover, and H. Hyötyniemi, "Coupling pairwise support vector machines for fault classification," *Control Engineering Practice*, vol. 13, pp. 759–769, 2005.
[46] P. Somol, J. Novovicová, J. Grim, and P. Pudil, "Dynamic oscillating search algorithm for feature selection," in *International Conference on Pattern Recognition 2008*. IEEE Computer Society, 2009, pp. 1–4.
[47] P. Somol and P. Pudil, "Oscillating search algorithms for feature selection," in *International Conference on Pattern Recognition 2000*. IEEE Computer Society, 2000, p. 2406.

Tomas Pfister received the BA degree in computer science at University of Cambridge, UK. His research interests include affective computing, human-computer and human-robot interaction, in particular affect recognition from speech and facial features.



Peter Robinson is Professor of Computer Technology and Deputy Head of the Computer Laboratory at the University of Cambridge in England, where he leads the Rainbow Group working on computer graphics and interaction. His research concerns new technologies to enhance communication between computers and their users, and new applications to exploit these technologies. Recent work has included desk-size projected displays and inference of users mental states from facial expressions, speech, posture and gestures. He is a Chartered Engineer and a Fellow of the British Computer Society.