# Feature Selection Methods and Their Combinations in High-Dimensional Classification of Speaker Likability, Intelligibility and Personality Traits

Jouni Pohjalainen[a,*], Okko Räsänen[a], Serdar Kadioglu[b]

[a] *Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland*
[b] *Oracle America Inc., Burlington, MA 01803, USA*

## Abstract

This study focuses on feature selection in paralinguistic analysis and presents recently developed supervised and unsupervised methods for feature subset selection and feature ranking. Using the standard k-nearest-neighbors (kNN) rule as the classification algorithm, the feature selection methods are evaluated individually and in different combinations in seven paralinguistic speaker trait classification tasks. In each analyzed data set, the overall number of features highly exceeds the number of data points available for training and evaluation, making a well-generalizing feature selection process extremely difficult. The performance of feature sets on the feature selection data is observed to be a poor indicator of their performance on unseen data. The studied feature selection methods clearly outperform a standard greedy hill-climbing selection algorithm by being more robust against overfitting. When the selection methods are suitably combined with each other, the performance in the classification task can be further improved. In general, it is shown that the use of automatic feature selection in paralinguistic analysis can be used to reduce the overall number of features to a fraction of the original feature set size while still achieving a comparable or even better performance than baseline support vector machine or random forest classifiers using the full feature set. The most typically selected features for recognition of speaker likability, intelligibility and five personality traits are also reported.

*Keywords:* feature selection, pattern recognition, machine learning, computational paralinguistics

## 1. Introduction

Automatic paralinguistic analysis of speech signals aims to uncover aspects of speech that are not related to the linguistic content of the signal (Schuller et al., 2013). Typical problems include the recognition of a speaker's age, gender, emotional state and possible altered states such as sleepiness or intoxication. The two main approaches to these classification and regression problems are 1) to design the system specifically for the paralinguistic analysis task using expert knowledge in the domain or 2) to generate a large number of suitably high-level features, typically depicting some aspect of the speech signal over several seconds or one utterance, and then apply generic machine learning methods to the high-dimensional feature data. This study is concerned with the latter approach, focusing on automatic selection of useful signal features with the goal of improving classification performance from a large, non-selective baseline feature set and in order to gain better understanding of the given paralinguistic analysis tasks.

In machine learning, high-dimensional feature spaces are sparsely populated by limited training data since the number of data points inside a volume unit decreases with an increasing feature space dimensionality. This effect, commonly referred to as the *curse of dimensionality*, weakens the reliability of trained analysis systems (Duda et al., 2001; Theodoridis and Koutroumbas, 2003) as *overfitting* them to the data becomes easier. However, if the complete feature set $F$ is comprehensive and contains informative features, then certain feature subsets out of the vast amount of possible subsets ($2^{|F|} - 1$) should define lower-dimensional feature spaces in which learning is more reliable with the limited training data. Provided that such feature spaces can be found, even a basic analysis system could perform better than a complex system operating in the complete feature space due to the lessened effect of the curse of dimensionality. When the high-dimensional analysis problem is viewed in this way, it becomes one of finding these feature subsets by means of *feature selection.* Therefore, this study investigates a feature-selection approach to tackle paralinguistic classification of speech when a large and varied set of high-level features is available (Schuller et al., 2013). More specifically, the focus is on a problem of finding a robust subset of features when the overall number of potential features highly exceeds the number of data samples available for training and evaluation of the system, making the process of feature selection highly susceptible to overfitting. The remainder of the introduction discusses issues related to feature selection in

[*] Corresponding author. Tel.: +358 504066900.
*Email addresses:* jouni.pohjalainen@aalto.fi (Jouni Pohjalainen), okko.rasanen@aalto.fi (Okko Räsänen), serdark@cs.brown.edu (Serdar Kadioglu)

pattern recognition (Section 1.1) and outlines the specific aims of the study (Section 1.2).

### *1.1. Feature Selection in Pattern Recognition*

Feature selection algorithms are often used to reduce feature space dimensionality in pattern classification and regression. This study focuses on feature selection for supervised classification, i.e., classification of data patterns – consisting of the selected features – into predefined categories that have meaning in the real world. Automatic feature selection can be formulated as the problem of finding the best possible subset $S$ of features from an initial, and possibly a very large, set of features $F$ (i.e., $S \subset F$). The learning of a more compact set of features can induce some or all of the following benefits (Blum and Langley, 1997; Reunanen, 2003):

1. Enhanced classification performance due to the removal of noisy or unreliable features.
2. Lower computational costs in the final system due to reduced dimensionality in feature extraction, model training and classification.
3. Simpler classifiers with less input variables, which often leads to better generalization ability towards new samples.
4. Better hands-on understanding of the classification problem through discovery of relevant and irrelevant features.

Since the ultimate goal is to perform classification of data samples, one could define the optimal subset of features as the one that provides the best classification ability in the given task as measured by a criterion function $G(S, D, M) = c$, where $D$ denotes the data set used and $M$ denotes the classification model (with its parameters) applied in the task. Value $c$ of the criterion function can correspond to the overall classification performance on $D$ in the given task (in the so-called *wrapper* methods) or be heuristically defined otherwise (as in the *filter* methods; Blum and Langley 1997; Guyon and Elisseeff 2003; Kohavi and John 1997). However, the above definition already contains at least two potential problems. First, the number of possible feature subsets grows exponentially as a function of the initial feature pool size, making exhaustive search of the best subset impossible for all but the simplest selection problems (the *search* problem). Second, the value of the criterion function has to be computed from a finite number of data points $D$ available for the selection process and there are no guarantees that the locations of local or global maxima of the criterion function with respect to $S$ will remain the same for previously unseen data, i.e., when $D$ is replaced by other data (the *generalization* or *overfitting* problem; Reunanen 2003). In general, the more there are features to be considered in $F$ and the less there are labeled data $D$ available for the feature selection process, the higher the risk that the chosen feature set (out of a very large population of candidate feature sets) performs well on a given small training data set, "by chance", but its predictive power generalizes poorly to new data sets.

In order to tackle the search problem, all practical algorithms apply some heuristics to guide the search process, either explicitly as with the wrapper methods or indirectly as with the filter methods (Blum and Langley, 1997). Sequential backward elimination (SBE), originally described by Marill and Green (1963), starts from the complete set of features and sequentially eliminates the one whose elimination results in the best score $G(S, D, M)$. A feature set of size $d$ is thus given by

$$S_d = S_{d+1} \setminus \underset{f}{\operatorname{argmax}}\, G(S_{d+1} \setminus f, D, M). \tag{1}$$

Sequential forward selection (SFS), proposed by Whitney (1971), works in the opposite direction: starting from an empty set, the feature set is iteratively updated by including, in each step, the feature $f$ which results in maximal score $G(S, D, M)$. Thus, the feature set of size $d$ is given by

$$S_d = S_{d-1} \cup \underset{f}{\operatorname{argmax}}\, G(S_{d-1} \cup f, D, M). \tag{2}$$

Typically, these methods are used as wrapper feature selection methods such that the criterion function $G(S, D, M)$ is evaluated using an actual classifier $M$ which is trained and evaluated on different parts of the data set $D$. They are greedy search algorithms, as they always exclude or include the most promising feature. Thus, the contribution of including or excluding a new feature is measured with respect to the set of previously chosen features using a hill-climbing scheme in order to optimize the criterion function $G(S, D, M)$, making these approaches susceptible to its local maxima with respect to $S$ (Blum and Langley, 1997). The feature sets found by SFS and SBE are *nested*, i.e., each feature set found by them is a subset of each larger feature set found earlier or later in the search.

Pudil et al. (1994) proposed the improved "floating" versions of SBE and SFS, which after exclusion or inclusion of a new feature, respectively, include or exclude as many previously excluded/included features as possible without decreasing the previous scores $G(S_d, D, M)$ associated with each feature set size. The sequential floating search algorithms do not have the nesting property and are not strictly greedy, potentially improving their performance (Pudil et al., 1994).

In the family of so-called embedded algorithms (see, e.g., Blum and Langley 1997 for a review), feature selection occurs by the internal mechanisms of the classification algorithm. For example, decision trees (e.g., CART by Breiman et al. 1984) and their variants such as random forests (Breiman, 2001) carry out recursive partitioning of the data based on features that are the most useful in distinguishing between different data classes.

Finally, the filter methods attempt to perform feature selection by replacing the role of the classifier $M$ in the criterion function $G(S, D, M)$ by a heuristic method to assess the relevance of features and their combinations, i.e., independently of the classifier used. This can be accomplished by using, e.g., measures of class separability in the feature distributions such as divergence or Bhattacharyya or Mahalanobis distance (Marill and Green, 1963; Theodoridis and Koutroumbas, 2003).

In the simplest type of filter algorithms, features are individually assigned scores which are assumed to reflect their usefulness in the intended classification task. The features can then be ranked by their importance according to the score. The ranking can be used to sequentially select a given number of best features, or the filter algorithm itself can provide a means to determine the size of the feature set, e.g., a threshold value for the score above which the features are considered useful in the task. Other filter algorithms are *feature subset selection* algorithms (Kohavi and John, 1997; Theodoridis and Koutroumbas, 2003), in that they consider feature subsets jointly using some criterion, but in contrast to the wrapper methods which also perform feature subset selection, the filter methods do not evaluate the subsets directly using the target classifier. Examples include correlation-based feature selection (CFS; Hall 1999) and the minimal-redundancy-maximal-relevance (MRMR) approach (Peng et al., 2005). CFS and MRMR analyze correlation and mutual information, respectively, and attempt to maximize it between the features and the class information while simultaneously minimizing it between features in the selected feature set. Filter methods are typically faster to compute than wrapper methods, but do not have direct access to the performance on the actual classification task and do not take into account the interaction between the chosen features and the classifier used. On the other hand, this reduces the risk that the feature selection overfits to the training data when used in conjunction with classifiers that themselves are prone to overfitting. Also, in order to alleviate the overfitting problem, combinations of different feature selection algorithms have recently been used in some studies (Pohjalainen et al., 2012; Saeys et al., 2008).

As can be seen from the above discussion, the generalization problem is inherently tied to the search problem, especially in the case of wrapper algorithms. The more the search relies on sequential steps based on the local maximum gain in the criterion function, the more important it becomes that the criterion function of the available data truly follows the topology of all of the data that the classifier will be applied to. The practical way to alleviate this problem is to have more data for training and development sets and to ensure that the division into training and development data is performed carefully so that there are no unrealistic similarities between them (e.g., no same talkers or identical linguistic content when performing classification of spontaneous speech). However, data collection and labeling is often expensive and ideally the feature selection algorithms should provide useful results with as little data as possible.

The practical measure of overfitting is to test the system on a set of held-out test data (Reunanen, 2003) after the feature selection and classifier training has been performed using the training and development sets. However, the use of an independent test set does not provide help in the selection of good features as such, because any observations on the test set performance that are propagated back to the system design will basically endanger the validity of the test set itself, again leading to the potential problem of overfitting. Instead, a proper use of held-out validation set simply shows whether the proposed methods generalize well or not.

### *1.2. Aims of the Study*

In this study, new feature selection methods based on various criteria, as well as methods for their combination, are proposed and applied with a basic nearest-neighbor classifier in a set of challenging paralinguistic speaker trait recognition tasks (Schuller et al., 2012). The data is characterized by a very large number of features and a small number of instances, making the tasks especially critical for robustness and generalization capabilities of feature selection algorithms. Therefore, the avoidance of overlearning in the selection phase is a central concern.

The results obtained are compared against various baselines in both feature selection and pattern classification. In feature selection, the goal is to validate the proposed selection methods as well as to investigate the benefits of combining different types of selection criteria. In classification, the goal is to compare the feature selection approach, whose basic idea was outlined in the beginning of the introduction, against high-dimensional pattern classification systems. Moreover, analyses of the types of acoustic features selected for binary classification of speaker likability, intelligibility and the Big Five personality traits are aimed at uncovering information related to these speech analysis tasks. The study is a continuation of the authors' previous work in the Interspeech 2012 Speaker Trait Challenge (Pohjalainen et al., 2012; Schuller et al., 2012).

## 2. Material

As the source of evaluation material, this study uses three databases, which are briefly described.

The Speaker Likability Database (SLD; Burkhardt et al. 2011) is a subset of the German Agender database originally recorded to study automatic age and gender recognition from telephone speech (Burkhardt et al., 2010). The speech has been recorded over fixed and mobile telephone lines using a sampling rate of 8 kHz. A set of 800 speakers, each speaking one utterance, comprises the SLD subset. The speaker population has been balanced for age and gender and 18 utterance types are included. In the generation of the labelings, listeners judged the likability of each speaker on a seven-point scale and the likability of each utterance was established using evaluator-weighted estimator (EWE; Grimm and Kroschel 2005) which weights the reliability of each listener based on the cross-correlation of his or her ratings with ratings averaged over listeners. The EWE rating was discretized into two categories, herein referred to as "likable" and "not-likable", based on the overall median EWE rating.

The "NKI CCRT Speech Corpus" (NCSC; van der Molen et al. 2012) was used as material in recognizing speaker intelligibility. The corpus contains recordings of utterances spoken in Dutch by 55 speakers (45 male and 10 female) who underwent concomitant chemo-radiation treatment (CCRT) for inoperable tumors of the head and neck. Recordings were made both before and after CCRT. Thirteen expert listeners rated the intelligibility of each recording on a seven-point scale and, similarly to the likability data, EWE ratings were computed and discretized to "intelligible" and "not-intelligible" categories based on the global median.

The "Big Five" or "OCEAN" personality characteristics – openness to experience, conscientiousness, extraversion, agreeableness and neuroticism – were recognized using the "Speaker Personality Corpus" (SPC; Mohammadi and Vinciarelli 2012) as material. The corpus consists of 640 speech audio clips, with a single speaker in each clip of approximately 10 seconds, randomly extracted from news in French broadcast by Radio Suisse Romande, the Swiss national broadcast service, during February 2005. The total number of speakers is 322 with the most frequent speaker appearing in 16 clips. The personality assessment was performed by 11 judges, each of whom listened to all the clips, by filling the BFI-10, a 10-item personality assessment questionnaire (Rammstedt and John, 2007). In order to generate the data labelings for each of the five personality traits, each clip was labeled as representing a personality trait if at least six judges (the majority) gave it a score that was higher than their personal average score for the same trait. Otherwise the clip was labeled as not representing the trait.

For each utterance in each of the three databases, 6125 long-term, utterance-level features have been extracted by the organizers of the Interspeech 2012 Speaker Trait Challenge (Schuller et al., 2012) using the openSMILE feature extractor (Eyben et al., 2010). For the present work, only these data sets consisting of 6125 features per utterance and the associated class labelings of the data instances (utterances) were used.

Each of the three databases was partitioned in the Speaker Trait Challenge into three data sets: a training set (denoted by Train), a development set (denoted by Development) and a test set (denoted by Test). The grounds for partitioning the utterances of each of the three databases into the Train, Development and Test sets are given by Schuller et al. (2012). Table 1 shows the number of instances for each such subset in the likability, intelligibility and personality data.

Table 1: *Number of instances in Train, Development (Devel) and Test sets in Likability, Intelligibility and Personality data.*

| | Data set | | | |
|---|---|---|---|---|
| Database | Train | Devel | Test | Total |
| Likability | 394 | 178 | 228 | 800 |
| Intelligibility | 901 | 746 | 739 | 2386 |
| Personality | 256 | 183 | 201 | 640 |

The problem in each task is to automatically classify data points, representing speech audio clips, with respect to traits $X \in$ {likable, intelligible, open, conscientious, extraverted, agreeable, neurotic} as either "$X$" or "not-$X$", i.e., whether the trait is present or not. Evaluation of the system is performed by comparing the generated hypotheses against the ground truth labelings associated with the data sets.

## 3. Methods

In this study, the nearest-neighbor classification rule (Section 3.1) is used in combination with different feature selection algorithms and their combinations. The feature selection algorithms are divided into two main categories: *subset selection algorithms* (Section 3.2) that provide a feature *set* as an output, and *scoring algorithms* (Section 3.3) that provide a scalar value of feature usefulness for each candidate feature (easily converted to a ranking of features). The scoring methods require a way to determine the size of the final feature set and solutions to this problem are presented in Section 3.4. Different approaches for combining multiple feature selection methods are described in Section 3.5. Finally, computational costs of individual feature selection methods are analyzed in Section 3.6.

### *3.1. Classification Algorithm*

$K$ nearest neighbors (kNN) is applied as the classification rule in the current study (Duda et al., 2001): the hypothesized class label for each test instance is determined as the one that is seen most frequently among the $k$ labeled training instances that are closest to the sample in terms of the Euclidean distance. Despite being conceptually simple and easy to implement (barring computational efficiency issues), it is nevertheless a powerful pattern classification method that, given enough training data, can model complex nonlinear decision boundaries in the feature space (Theodoridis and Koutroumbas, 2003). However, kNN is known to be susceptible to the effects of the curse of dimensionality (Duda et al., 2001; Theodoridis and Koutroumbas, 2003). From another viewpoint, kNN in its basic form does not have any internal mechanism to deal with feature relevance. This is in contrast to classifiers such as support vector machines and random forests, which are better able to handle high dimensionalities and irrelevant features. These things justify the choice of kNN for the purpose of the present study: being a capable, nonlinear pattern classification method whose performance, however, is relatively highly dependent on the quality of the feature set, it is particularly suitable for comparing the performance and robustness of different feature selection approaches.

Prior to kNN classification, each feature in both the training and evaluation dataset is normalized to have zero mean and unit variance within the corresponding data set (in the evaluation phase, this corresponds to the system having been in operation long enough to have these normalization statistics of the observed features available). When making a decision on an input vector based on its $k$ nearest neighbors according to the Euclidean distance, the counts of different classes within the $k$-neighborhood are scaled by dividing them by the frequencies of occurrence of the same classes in the training data in order to compensate for potentially biased class distributions.

The number of neighbors $k$ is chosen in the present work by first selecting the best-performing value $k_0$, from a given range of values, in the classification of a development set $D_{\mathrm{DEVEL}}$ using training data $D_{\mathrm{TRAIN}}$. In classifying the test data $D_{\mathrm{TEST}}$, the development set is included in the training material. In order to maintain the size of the $k_0$-neighborhood in terms of the Euclidean distance despite the increased sample density of the extended training set $D_{\mathrm{TRAIN}} \cup D_{\mathrm{DEVEL}}$, the final value of the parameter is determined as

$$k = \left\lfloor k_0 \frac{|D_{\mathrm{TRAIN}} \cup D_{\mathrm{DEVEL}}|}{|D_{\mathrm{TRAIN}}|} \right\rceil, \tag{3}$$

where $\lfloor \ldots \rceil$ denotes rounding to an integer.

### 3.2. Feature Subset Selection Algorithms

These methods return a subset of features based on an intrinsic determination of the feature set size. They include a well-known wrapper method, sequential forward selection, a more recently proposed filter method, minimal-redundancy-maximal-relevance, and two new approaches, namely feature set selection as Set Covering Problem (SCP) and Random Subset Feature Selection (RSFS).

#### 3.2.1. Sequential Forward Selection (SFS)

Sequential forward selection (SFS; Whitney 1971) was chosen as the baseline method for feature selection, as it is well known and widely used in practice. In addition, Reunanen (2003) found this simple algorithm to often perform competitively to sequential floating forward selection (SFFS; Pudil et al. 1994), which is widely regarded as one of the state-of-the-art feature selection algorithms. While its good performance level was not disputed in (Reunanen, 2003), it was argued that the intensive search strategy of SFFS causes it to more effectively overfit the features to the feature selection data set and may have led researchers to overestimate its performance if additional validation data have not been used. In addition to these considerations, the computational cost of SFFS (the original version proposed by Pudil et al. 1994) was found to be too high for the large initial feature pool used in the current study. In most cases, SFFS failed to converge to a stable feature set of a specific size nor was it able to reach the predefined maximum feature set size of 500 features in a reasonable computation time, in the order of weeks, on machines with four Xeon E3-1230 3.2 GHz CPUs and 8 GB of memory. In several previous studies that have evaluated SFFS, the algorithm has not been run until feature sets of such size leading to very large search spaces (Reunanen, 2003), not even in recent paralinguistic analysis studies (Batliner et al., 2010). In the few cases where SFFS did manage to reach feature set sizes stipulated for the algorithms in this study, it was found to perform worse than SFS. The computational cost of SFS, on the other hand, was feasible, although still noticeably high in comparison to all the new feature selection algorithms evaluated. Therefore, SFS was an obvious choice for the feature selection baseline in the present study.

SFS is implemented according to Eq. 2 with the function $G(S_d, D, M)$ evaluated by means of kNN classification of a subset of the data $D$ while using the remainder of $D$ as training observations. The value of $G(S_d, D, M)$ is the maximum unweighted average recall (UAR) over a range of $k$ values $k \in \{5, 10, \ldots, 150\}$. UAR is the class-specific correct classification rate averaged over the actual classes. SFS is run until 500 features, the predefined maximum, is reached and the feature set finally chosen as

$$S = \operatorname*{argmax}_{S_d} G(S_d, D, M). \tag{4}$$

#### 3.2.2. Minimal-Redundancy-Maximal-Relevance (MRMR) Feature Subset Selection

Minimal-redundancy-maximal-relevance (MRMR) is a filter-based feature selection approach proposed by Peng et al. (2005). It analyzes the mutual information between discretized features and class labels to maximize the feature relevance while simultaneously considering the mutual information among the discretized features in the selected feature set to minimize redundancy. To use this method, each of the continuous-valued features is quantized to three levels by placing the quantization boundaries at $\mu \pm \sigma$ where $\mu$ and $\sigma$ denote the feature's estimated mean and standard deviation. This yields a set of discretized features $y(f)$, $f \in F$. Features are then selected, one at a time, using the rule

$$S_d = S_{d-1} \cup \operatorname*{argmax}_{f} \left[ I(y(f), z) - \frac{1}{d-1} \sum_{g \in S_{d-1}} I(y(f), y(g)) \right], \tag{5}$$

where $I$ denotes mutual information (see Eq. 11) and $z$ is a categorical variable containing the class labeling. In choosing the feature set size, $G(S_d, D, M)$ is evaluated with a kNN classifier ($k = 40$) for $1 \leq d \leq 500$ and the final feature set chosen, as with SFS, according to Eq. 4.

### *3.2.3. Feature Set Selection as Set Covering Problem (SCP) of Correct Classifications*

The SCP *(Set Covering Problem)* approach to feature selection is based on first classifying an evaluation set using each single feature separately and then selecting enough features to "cover" the set with correct classifications. The single-feature classifications are based on either supervised or unsupervised discrimination between two classes using Gaussian mixture models (GMMs) to model the unidimensional probability distributions.

With the goal of classifying utterances as either "$X$" or "not-$X$" (where, in this study, $X$ is one of the seven speaker traits), *unidimensional* GMMs $\lambda_{\mathrm{X}}$ and $\lambda_{\text{not-}X}$ are trained *separately to represent each feature*. In the first *supervised* training step, for each feature, five iterations of EM (expectation-maximization; Dempster et al. 1977) re-estimation for GMMs (Xu and Jordan, 1996) are carried out using class-specific data to train each of the two GMMs. Before training, the mixture weights of the GMMs are initialized by uniform distributions and the variance parameters of each component by 0.1 times the global variance of the feature. The mean parameters of each component are initialized by feature values selected by the heuristic approach described in (Katsavounidis et al., 1994). In classification, the class decision for each observation is based on the logarithmic likelihood ratio $L = L_{\mathrm{X}} - L_{\text{not-}X}$, where $L_{\mathrm{X}}$ and $L_{\text{not-}X}$ are the logarithmic likelihoods of the observation having been produced by each GMM. Class $X$ is decided if $L \geq T$, where $T$ is the decision threshold adjusted on the training data set according to the equal error rate (EER) criterion typically used in detection applications. The EER threshold corresponds to equal misclassification rate for both classes.

After the initial supervised training, mixture-based classification using individual features is also performed with *unsupervised* learning. The parameters of two $J$-component GMMs are joined to form one composite GMM with $2J$ components and the component weight parameters are multiplied by 0.5. The composite GMM is trained with further five iterations of *modified* EM training where the sums of the $J$ weight parameters belonging to the $X$ and not-$X$ classes are both normalized back to 0.5 before each "expectation" step (E step; Dempster et al. 1977). This modification to the EM algorithm ensures that the prior probabilities of the two classes remain equal in the E step. Otherwise, the GMM parameters are allowed to freely adapt to the complete data set consisting of observations from both classes. After this unsupervised training, the sub-GMMs belonging to the two classes are again separated and the EER decision threshold $T$ is determined based on the training data. EM, which is used to find maximum likelihood parameter estimates for models with latent variables, is guaranteed to converge towards at least a local, if not a global, maximum of the likelihood function (Dempster et al., 1977). Therefore, the unsupervised method should lead to good classifications with features for which solutions that discriminate between the two classes are close in likelihood to a local maximum or a saddle point of a natural clustering solution of the training data.

Classifications of another held-out data set are obtained using both of the above methods, i.e., by supervised and unsupervised learning in the training phase. For both cases, matrices are constructed where rows correspond to audio clips, columns correspond to features, and the value is 1 if the clip in question has been correctly classified by the feature in question and 0 otherwise.

We make the following observation. Our goal to select a subset of features can be formulated as an Integer Linear Programming (ILP) problem. In particular, we map our problem to the well-known Set Covering Problem (SCP; Cormen et al. 2001). The decision version of the set covering problem is one of Karp's 21 NP-complete problems as was shown in 1972 (Karp, 1972). In the SCP problem, we are given a finite set $U := \{1, ..., m\}$ of items, a family $F = \{U_1, ..., U_n \subseteq U\}$ of subsets of $U$, and a cost function $c : F \rightarrow R^+$. The objective is to find a subset $S \subseteq F$ such that $\sum_{U_i \in S} c(U_i)$ is minimized. The SCP has numerous practical applications such as crew scheduling for airlines or railway companies (Caprara et al., 1997; Hoffmann and Padberg, 1993; Housos and Elmoth, 1997), location of emergency facilities (Toregas et al., 1971), and production planning in various industries (Vasko and Wolf, 1987). In our formulation, the set of items consists of data points (audio clips) and the family of sets corresponds to individual features, each associated with a binary vector indicating the correct and incorrect classifications of the data points. For each feature we attribute a cost of one, that is, we are dealing with the so-called unicost SCP. Finally, our objective is to *cover* all the data points using the minimum number of features.

As explained above, before solving this ILP, each feature has been evaluated on each data point, or speech audio clip, and it has been noted whether the data point can be classified correctly using that feature. This yields knowledge about which items (data points) are covered in which sets (features). Then, our formulation can be written as:

$$\begin{array}{lll} \text{Minimize} & \sum_{s \in F} x_s & \\ & \sum_{s \in F : e \in s} x_s \geq 1 & \forall\, e \in U \\ & x_s \in \{0, 1\} & \forall\, s \in F \end{array} \tag{6}$$

The decision variables $x_s$ denote whether the feature $s$ is selected while the first constraint ensures that we consider every data point. Our goal is to select a subset of features (this is the feature reduction aspect in our case) such that all the items (audio clips) are covered (can be identified) with at least one feature. This observation has an immediate bearing on our problem. We can leverage general techniques for solving ILP's; namely using a branch-and-bound algorithm based on linear relaxation of the original problem where the integer decision variables $x_s \in \{0, 1\}$

are replaced with $x_s \in [0, 1]$. While this algorithm solves the problem to optimality, in general, finding the minimum set cover is NP-hard (Nemhauser and Wolsey, 1988).

In order to solve the resulting SCP models, we employed IBM Ilog CPLEX solver (IBM, 2009), the state-of-the-art mathematical programming solver. Unfortunately, the SCP instances corresponding to our feature selection problem could not be optimally solved due to the huge number of binary decision variables in the ILP formulation. We found the memory requirement of solving the SCP to optimality to be prohibitive in practice. In most of the cases, we hit the memory limit when using a Dell PowerEdge M610s with 16 Xeon 2.4 GHz CPUs and 16GB of memory.

When optimal solutions cannot be computed efficiently, it is possible to trade optimality with efficiency. One alternative is to use a simple and fast greedy algorithm to obtain an approximate solution. The other alternative is to go beyond greedy algorithms and use sophisticated local search methods and meta-heuristics that are specifically designed for the SCP problem. We tried both of these options and found the following. The greedy algorithm selects a set that covers the most items using the least cost at each step until all items are covered (Chvátal, 1979). In essence, it resembles sequential forward selection. This algorithm achieves an $H_n$ factor approximation algorithm for the minimum set cover problem, where $H_n = 1 + 1/2 + ... + 1/n$ (Vazirani, 2001). However, we noticed that very dense sets exist in our SCP representation, i.e., there are some features which are able to correctly classify most of the audio clips. As a result, the greedy algorithm is able to find a covering by selecting as few as three to six sets. This results in an undesirable, overlearning-prone feature selection approach which favors very few features. We next employed a powerful local search SCP solver from (Kadioglu and Sellmann, 2009) which is based on the dialectic search paradigm. Dialectic search was shown to perform well on the challenging SCP instances from the Operations Research Library (Beasley, 1990) and outperformed the previously best local search approaches based on Tabu Search and Iterated Local Search that are specifically tuned for solving the unicost SCP. We found out that local search approaches are also subject to the previous problem. They are trapped in a local optimum obtained by using very few sets. As was the case for the greedy algorithm, the neighborhood selection operators also favor the dense sets.

We noticed that unlike finding the integer optimal solution, solving the linear relaxation at the root node of the branch-and-bound tree can, in practice, be performed efficiently without being subject to the aforementioned memory issues. Moreover, linear relaxation is not subject to the problem of favoring very few sets, as it yields fractional values for a considerable subset of the decision variables. Hence, in this work, we used the well-known approximation technique for solving the SCP problem which first solves the linear relaxation to optimality and then uses the rounding-up method to obtain an integral solution (Vazirani, 2001).

Depending on whether the SCP was based on classifications using GMMs trained in a completely supervised or a partially unsupervised manner, the feature selection method is termed supervised- or unsupervised-training SCP (SSCP or USCP, respectively). In the experiments of this study, the SSCP method uses $J = 8$ components in each GMM and the joint GMM in USCP has $2J = 16$ components. This choice is based on the considerations of having as many components as possible in the GMMs to accurately represent the probability density functions and possibly complex-shaped cluster structures while at the same time also having enough training data to reliably estimate each component.

### *3.2.4. Random Subset Feature Selection (RSFS)*

*Random Subset Feature Selection* (RSFS) is a feature selection algorithm that aims to discover a set of features that perform better than an average feature of the available feature set. The set of "good" features is obtained by repetitively choosing a random subset of features from the set of all possible features and then classifying the data with a kNN classifier using these features. During each iteration, the relevance of each feature is adjusted according to the classification performance of the subset that the feature participates in. As more iterations are performed, the quality of the feature set gradually improves as random components in the selection process become averaged out. In this manner, each feature becomes evaluated in terms of its average usefulness in the context of many other feature combinations. Also, since the relevance values are not dependent on the previous choices in the selection process but are a result of many independent trials, RSFS should not be susceptible to a locally optimal solution like the greedy hill-climbing-based feature selection methods (see also Räsänen and Pohjalainen 2013).

The RSFS is based on the idea of Random Forests (Breiman, 2001) and Random kNN (RKNN; Li et al. 2011) where the classification task is split into a set of classifiers that use random subsets of features, and where the quality of each individual feature can be evaluated according to its participation in correct classifications. The main difference of RSFS to RKNN is that the final feature selection process in RSFS is based on a statistical comparison against random walk statistics. In contrast, RKNN performs two subsequent stages with the first stage computing the relevance of each feature using a fixed number of random subset classifiers and the second stage performing backward elimination of the least relevant features in order to find the feature set with the best classification performance (Li et al., 2011). In RSFS, the random subset classification is performed as many times as is necessary in order to distinguish good features from features that simply appear useful due to the random components of the process. Thus, no greedy backward elimination steps are required.

In RSFS, each *true* feature $f_j$ from a full set of features $F$ has a relevance value $r_j \in (-\infty, \infty)$ associated with it. In addition, a set of *dummy* features $z_j \in Z$ with related relevances $q_j$ is also defined.

During each iteration $i$, the RSFS algorithm performs the following steps:

1. Randomly select a subset $S_i$ of $n$ features ($|S_i| = n$) from the full set $F$ by sampling from a uniform distribution.
2. Perform kNN classification on the given data set using $S_i$ and compute the value of a desired criterion function $c_i$ which measures classification performance.
3. Update relevances $r_j$ of all used features $f_j$ by replacing them with

$$r'_j = r_j + c_i - E\{c\}, \tag{7}$$

   where $c_i$ is the value of the criterion function for the current iteration $i$ and $E\{c\}$ is the expected value of the criterion function (in the current work, this corresponds to the average of $c_i$ across all previous iterations).
4. Repeat the process from 1) with a new random subset.

In parallel to updating feature relevances, a similar process is performed for the dummy features by always selecting a random subset of $m$ dummy features and then updating the relevance values of these features according to Eq. 7 but using the criterion function value of the true features from the same iteration. The dummy features are never used in the actual classification process (i.e., they have no values for any data sample) but their relevances are still accumulated across trials similarly to the true features. Thus, the relevance $q_j$ of any dummy feature $z_j$ essentially becomes a random walk process with no correspondence to any actual classification performance. In this manner, the relevance of the dummy features provides a baseline level $r_{\text{rand}}$ that should be exceeded by a true feature in order to be considered as useful in the classification task.

Finally, in order to find the set of features $S \subset F$ that truly exceeds the dummy features' relevance ratings, a statistical test is performed. More specifically, it is required that the relevance $r_j$ of a true feature $f_j$ satisfies

$$p(r_j > r_{\text{rand}}) \geq \delta, \qquad \forall f_j \in B, F, \tag{8}$$

where $r_{\text{rand}}$ is the relevance of a non-useful feature and $\delta$ is a user-set threshold for probability. The random baseline level $r_{\text{rand}}$ is modeled as a normal distribution of the dummy relevances $q_j$ and thereby the probability that a feature is more relevant than a dummy feature is obtained from the cumulative normal distribution

$$p(r_j > r_{\text{rand}}) = \frac{1}{\sigma_{\text{g}}\sqrt{2\pi}} \int_{-\infty}^{r_j} \exp(\frac{-(x-\mu_g)^2}{2\sigma_g^2} dx \tag{9}$$

where $\mu_g$ and $\sigma_g$ are the mean and standard deviation of the dummy feature relevances $q_j$ across all dummy features. In practice, the statistical testing can be performed between each iteration of the RSFS and the feature selection process can be stopped when the number of features exceeding the random baseline no longer increases, or the algorithm can simply be run for a fixed number of iterations that is preferably much higher than the total number of features in $F$.

In this study (see also Räsänen and Pohjalainen 2013), the unweighted average recall (UAR) was used as the criterion function $c$ in Eq. 7 and the probability threshold was set to $\delta = 0.99$. The number of features used in classification during each iteration was set to $n = 78 \approx \sqrt{|F|}$ according to Li et al. (2011). In a similar vein, a total of 50 dummy features were created and their relevances were updated on each iteration. The sampling process was repeated for 300 000 iterations before selecting the final set of features according to Eqs. 8 and 9. As for the kNN used as the criterion function, the number of neighbors $k$ in the voting was always fixed to $k = 2$, as a small value of $k$ was suggested to be used in the context of the RKNN algorithm (see Li et al. 2011).

### *3.3. Feature Scoring Algorithms*

In contrast to the subset selection methods, feature scoring algorithms provide only a score value for each feature to reflect its usefulness. In order to use these feature scoring methods for subset determination, additional considerations are needed to determine the size of the subset (see Section 3.4).

#### *3.3.1. Statistical Dependency (SD) Between Features and Labels*

The goal of the *Statistical Dependency* (SD) method is simply to measure whether the values of a feature are dependent on the associated class labels, or whether the two simply co-occur by chance. Each feature value is first quantized into one of $Q_{\text{S}}$ levels, where the feature-specific quantization scale is adaptively determined such that each bin will contain roughly an equal amount of samples over the entire data set. The bins are chosen in this way, instead of a conventional uniform quantization scale, in order to lend some statistical validity to the occurrence of different quantization levels. The statistical dependence between the discretized feature values $y$ and the class labels $z$ is evaluated according to the formula

$$SD = \sum_{y \in Y} \sum_{z \in Z} p(y,z) \frac{p(y,z)}{p(y)p(z)}. \tag{10}$$

The larger the SD, the higher is the dependency between the feature values and the class labels. In the case that the feature is fully independent of the class labels, the SD will obtain the minimal value of 1. Note the similarity of this measure with mutual information (MI), which is given by

$$MI = \sum_{y \in Y} \sum_{z \in Z} p(y,z) \log \left( \frac{p(y,z)}{p(y)p(z)} \right). \qquad (11)$$

The formula in Eq. 10, which omits the logarithm, has been found preferable to the conventional MI measure (Eq. 11) in assessing statistical dependence in problems like the present one (Pohjalainen et al., 2012). This may be due to the fact that the SD is more sensitive to individual highly informative quantization levels due to the absence of logarithmic compression of MI. Nevertheless, the MI measure is also included in the experiments of this study for comparison purposes. The SD and MI methods both result in a scoring and ranking of features, according to which a chosen number of features having the highest values can be selected.

Denoting the feature selection data set by $D_{\mathrm{FS}}$, the number of quantization levels for both methods was experimentally chosen as $L_{\mathrm{S}} = \lfloor |D_{\mathrm{FS}}|/10 \rceil$, i.e., each bin will contain approximately 10 data samples on the average.

### *3.3.2. Distribution Alignment and Matching (DAM) Between Training and Test Data*

So far, we have considered methods that base the feature selection on labeled data. In contrast to these conventional "offline" approaches, we would also like to investigate whether it is possible to select features "online" by comparing the observed sample distributions of each feature between the training data and a test data set, on which the predictor is being used.

Assuming that various random effects can cause local or global stretching, compacting or shifting of one sample distribution of a feature relative to another, we can attempt to compensate for these effects by *warping* the random variable's value axis in one of the sample distributions to better correspond to the other. Furthermore, we can attempt to compensate for the effect of different *class membership distributions* between the two data sets by normalizing according to the mean of the warped distributions. The motivation is that if these two distributions of the feature value become similar after trying to eliminate out local differences in feature value distributions and possibly different class distributions in the two data sets, the feature is more likely to be helpful in classification. On the other hand, if the distributions of some feature can not be easily matched between training and test data, even though both should consist of the same classes, it suggests that the feature does not behave systematically across different data sets with respect to the classes. The *Distribution Alignment and Matching* (DAM) feature scoring method is fully *unsupervised* as it does not make use of the class labels at all. Therefore, while it is reasonable to anticipate that its performance by itself may not be the best of the methods evaluated, it may have the potential to effectively complement conventional, supervised feature selection algorithms.

This approach is implemented as follows.

1. Given the training dataset $D_{\mathrm{TRAIN}}$ consisting of $N_1$ observations of the features in $F$, i.e., $f_i(n)$, $1 \leq i \leq |F|$, $1 \leq n \leq N_1$, and the testing dataset $D_{\mathrm{TEST}}$ consisting of $N_2$ observations of the same features, i.e., $g_i(n)$, $1 \leq i \leq |F|$, $1 \leq n \leq N_2$, histograms of feature values with $Q_{\mathrm{D}}$ bins are constructed separately for each feature in both datasets. The bins of the histogram for a feature $f_i \in D_{\mathrm{TRAIN}}$ cover the range of values of $f_i$ over the training data set while the bins of the histogram for the corresponding feature $g_i \in D_{\mathrm{TEST}}$ cover the range of values of $g_i$ over the test data set. After this, each histogram is individually normalized to have a peak value of 1. Let us denote the normalized training and test set histograms as $H(f_i, j)$ and $H(g_i, j)$, respectively, with $1 \leq i \leq |F|$ and $1 \leq j \leq Q_{\mathrm{D}}$.
2. Next, in order to compensate for small deviations in the feature distributions between the two data sets, the training data histograms $H(f_i, j)$ are aligned with the test data histograms $H(g_i, j)$ using dynamic programming. The alignment of the distributions is accomplished using an implementation of the *dynamic time warping* (DTW) method which is usually applied to time alignment of feature vector sequences (O'Shaughnessy, 2000). This algorithm finds the minimum-cost path through a grid of $Q_{\mathrm{D}} \times Q_{\mathrm{D}}$ nodes, where $Q_{\mathrm{D}}$ is the number of bins in both histograms and each node $(m, n)$ corresponds to a pair of training and test histogram bins $(H(f_i, m), H(g_i, n))$ and has the associated cost $d(m, n) = (H(f_i, m) - H(g_i, n))^2$. The constrained-endpoints version of DTW is used, i.e., the path is required to start at node $(1, 1)$ and end at node $(Q_{\mathrm{D}}, Q_{\mathrm{D}})$, and the local continuity constraints on the permitted paths dictate that any grid node $(m, n)$ can be reached by one move only from one of the nodes $(m-1, n)$, $(m, n-1)$ or $(m-1, n-1)$, except at grid boundaries where $m = 1$ or $n = 1$. In addition, at most two consequtive moves from $(m, n-1)$ to $(m, n)$ are permitted, except at the grid boundary where $m = Q_{\mathrm{D}}$. For each training data histogram $H(f_i, j)$, the alignment procedure gives a new version $H'(f_i, j)$ which has been aligned with the corresponding test data histogram $H(g_i, j)$.
3. After step 2, the training histograms $H'(f_i, j)$ have been optimally aligned (according to the chosen DTW constraints) with the test histograms $H(g_i, j)$. Thus, the remaining disagreement between the histograms should be primarily due to the class distributions that may be different between $D_{\mathrm{TRAIN}}$ and $D_{\mathrm{TEST}}$. The exact effect of changes in the class distribution on each feature value distribution is generally not known. However, the aggregate effect of the class distribution is manifested in the mean of the feature value distributions. Thus, as the following step, an attempt is made to compensate for the effects of different class distributions by subtracting $H'_{\mathrm{MEAN}}(j) = (1/|F|) \sum_i H'(f_i, j)$, i.e., the mean of the aligned his-

tograms across each feature, from each test histogram $H(g_i, j)$ to yield $H''(g_i, j) = H(g_i, j) - H'_{\mathrm{MEAN}}(j)$.

4. Step 2 is repeated to align the *original* training histograms $H(f_i, j)$ with the corrected test histograms $H''(g_i, j)$ to yield the alignment cost $C_i$ for each feature.
5. A score for the similarity of the matched training and test distributions of each feature is obtained as $1/C_i$.

For the experiments, the number of histogram bins was chosen as $Q_{\mathrm{D}} = 8$. It is worth pointing out that DAM requires a certain amount of analysis data to already be available in order to estimate the feature value distributions as histograms with sufficient reliability. In practice, this is not likely to present a problem except in a scenario where the statistical properties of the analyzed data have very recently changed.

### *3.4. Determination of the Size of the Feature Set for the Feature Scoring Methods*

The feature subset selection algorithms (RSFS, SSCP and USCP) discussed in this study each use an intrinsic criterion to determine the subset size: for RSFS, this is determined by statistical comparison of the relevance values of features against those of dummy features, and for the SCP-based methods, by the approximation level allowed in obtaining a suboptimal solution which covers the data set with more than the minimal number of features.

The feature scoring methods proposed in this study, SD and DAM, only provide a scoring and associated ranking of features, using different criteria. The size of the feature set selected by using these methods has to be estimated using some additional algorithm. Two methods for accomplishing this are considered. For both, the following steps are first performed:

1. Rank the features according to their scores. The feature scores are sorted in best-first order to obtain a ranking order for the features $o_i$, $1 \leq i \leq |F|$, where $o_1$ is the index of the feature with the best score, $o_2$ is the index of the second best score, etc.
2. Evaluate each sorted feature set $\{o_1, \ldots, o_q\}$ up to a maximum allowed size (500 features in this study) in classification using a range of classifier parameter values (for kNN, the value of $k \in \{5, 6, \ldots, 150\}$) and record the best classification score for each number of features $q$. Denote this score as $u_i$, $1 \leq i \leq |F|$.
3. Obtain $R$ *random* permutations of features and for each permutation, evaluate each allowed size of feature set $q$ in classification by taking the $q$ first features from the permutation. For each $q$, average the obtained classification scores over the $R$ permutations and denote this score by $v_i$ ,$1 \leq i \leq |F|$. In the experiments, $R = 10$ random orderings are considered.
4. Apply a smoothing filter over the number of features $i$ to both $u_i$ and $v_i$ in order to make the subset size selection less exact and less susceptible to local maxima (with respect to the features chosen) of the classification score, which may be specific to the evaluated data set. The motivation is thus to reduce the risk of overfitting. This step, which was included in an earlier study (Pohjalainen et al., 2012), is performed using a three-tap moving average filter.

The first method of dimensionality determination directly chooses the number of features giving the highest (smoothed) score on the optimization data:

$$d = \operatorname*{argmax}_i (u_i). \tag{12}$$

The second method was motivated by the concern that choosing one particular subset of features based on just one data set may be prone to overlearning. However, the amount of data is limited in problems like the present ones. Thus, we work on the hypothesis that perhaps the data set contains more class-specific structure in subspaces of particular dimensionalities than in subspaces of other dimensionalities. However, we also want to utilize the information about a suitable dimensionality provided by the actual feature ordering that we are working with, like in the first approach. Thus, we combine the first criterion, i.e., choosing the dimensionality based on the classification performance of subsets based on the ordering provided by the score function, with another criterion, where the suitable dimensionality is assessed based on the averaged classification performance of randomly selected subsets. This approach was successfully used by Pohjalainen et al. (2012, the present authors) in the Interspeech 2012 Speaker Trait Challenge (Schuller et al., 2012). The feature space dimensionality is thus determined as

$$d = \operatorname*{argmax}_i (a u_i + b v_i), \tag{13}$$

where $a$ and $b$ are weighting constants. In this study, $a = b$ such that the score-based ranking and the random permutations are given equal weight in choosing the feature set size.

### *3.5. Combining the Selection Methods*

Some recent studies have found the combination of feature selection algorithms to help resist overlearning (Pohjalainen et al., 2012; Saeys et al., 2008). In this section, some methods of combining different types of feature selection algorithms are briefly described. The algorithms are combined differently depending on whether they result in a feature subset or feature-specific scores.

#### *3.5.1. Combination of subset and scores*

In this combination approach, we take as starting point a complete feature set, such as one provided by SSCP, USCP or RSFS, and use a feature scoring method, such as SD or DAM, to further select features within the subset using one of the dimensionality determination methods discussed in Section 3.4. More formally:

1. Given an initial subset $S$ consisting of features $f_i$, $S = \{f_i\}$, apply a feature scoring method to get scores $w_i$ for each feature.
2. Sort the scores in the best-first order in order to obtain a ranking order of the features.
3. Choose the $d$ best features according to the ranking, with $d$ given by one of the algorithms described in Section 3.4.

### *3.5.2. Combination of subsets*

Given subsets of features determined by a complete subset selection algorithm, such as SSCP, USCP or RSFS, we investigate two methods of combining their results: union and intersection. Union straightforwardly expands the feature set to take into account each feature selected by the different algorithms. While intersection may result in small feature sets that are sub-optimal for classification, especially when subsets obtained using different feature selection criteria are combined, it can nevertheless provide us insight on two issues: firstly, which features seem so useful that they are selected using different criteria, and secondly, what would the classification performance level be like using a minimal feature set which can however be considered to be of high quality?

When combining subsets and scorings according to Section 3.5.1, a combination of subsets can straightforwardly be used as the initial feature set from which features are further selected using feature scores.

### *3.5.3. Combination of scores*

To combine feature scores, either as the primary feature selection method or as a refinement to a subset according to Section 3.5.1, two methods are investigated: the scores are either multiplied or added together. Before this, each individual score is normalized to the range of $(0, 1)$.

### *3.6. Computational Costs*

Since the applicability of feature selection algorithms does not only depend on the overall quality of the chosen feature sets, but also on the computational resources required to perform the selection process, the time complexities of the individual feature selection algorithms studied in this work were also analyzed.

Sequential forward selection (SFS) performs up to $d-1$ iterations, if $d$ here denotes the total number of candidate features. During the $k$th iteration, $d-k+1$ feature sets are evaluated. Therefore, it is straightforward to show that the number of total feature set evaluations is $O(d^2)$. The time complexity of evaluating *one* feature set using kNN classification is $O(N^2 d)$, where $N$ is the number of data points approximately equally divided between the training and testing sets. However, by storing in memory the squared Euclidean distance components of previously selected features it becomes $O(N^2)$. Therefore, with efficient implementation, SFS with kNN is $O(N^2 d^2)$.

Minimum-redundancy-maximum-relevance (MRMR) feature selection begins with the quantization operation which for $Q$ quantization levels is $O(NQd)$. Being a nested-subset method like SFS, MRMR requires up to $d-1$ iterations. Computation of the mutual information between quantized features and $C$ class labels is $O(NCQ)$ and computation of the mutual information between the quantized features is $O(NQ^2)$. Therefore, this method is $O(NQ^2d^2)$ for $Q \geq C$ and $O(NCQd^2)$ for $Q < C$. In this study, the choice of the feature set size is performed by evaluating the feature set of each size in kNN classification, a procedure which is $O(N^2 d)$.

The time complexity of both the statistical dependency (SD) and mutual information (MI) methods is $O(NCQd)$ (quantization $O(NQd)$ and SD/MI computation $O(NCQd)$). However, the time complexity of the subsequent, independent dimensionality determination step applied in this study consists of sorting the feature scores using an efficient sorting algorithm ($O(d \log d)$), evaluating approximately $G$ different variants of each feature set size in kNN classification ($O(GN^2 d)$), smoothing the obtained scores by a moving average filter ($O(d)$) and finding the optimal size ($O(d)$). With typical values of $d$ such that $\log d < GN^2$, dimensionality determination is thus $O(GN^2 d)$ and overshadows the actual cost of the feature scoring algorithm.

Distribution alignment/matching (DAM) first computes the histogram for each feature, which can be obtained as a by-product of quantization to $Q$ histogram bins ($O(NQd)$). This is followed, for each feature, by dynamic time warping (DTW) to find the minimum-cost path through a $Q \times Q$ grid, an algorithm which is known to be $O(Q^2)$. Therefore, for $d$ features the DTW is $O(Q^2 d)$. As in practice $N > Q$ always, in practical implementations this method is therefore $O(NQd)$. In the present study, similarly to the SD and MI methods, this method is complemented with the dimensionality determination algorithm whose complexity was found above to be clearly larger than this.

As for the RSFS, the time complexity estimation is more complex as the algorithm does not terminate automatically but converges to relatively stable set of features after $I$ iterations. The cost of the kNN classification during each RSFS iteration is $O(N^2\sqrt{d})$ since RSFS uses a subset of features whose size is the square root of the full feature pool size. The algorithm is run for $I$ iterations, yielding a total cost of $O(N^2\sqrt{d}I)$. However, the number of iterations needed to evaluate the relevance of each feature increases with the size of the original feature pool $d$ as the number of possible feature subsets increases. In theory, the number of different subsets is bounded from above by the growth of binomial coefficient. In practice, the algorithm is never run this many iterations, as the idea is to approximate the relevance of each feature by random sampling and not to perform exhaustive evaluation over all possible feature combinations. The number of required iterations in RSFS was empirically tested by varying the feature set size $d$ and finding the minimum number of iterations required for the convergence in the feature set size. Convergence point was defined as the iteration number at which the size of the chosen feature set had had a maximum of ±1 % variation in size during the last 2000 iterations. As a result, the increase in the number of required iterations $I$ was found to be linearly increasing with $d$ ($\rho = 0.92$, Pearson correlation). Since $I = ad$ where $a$ is a constant, the overall complexity of RSFS is $O(N^2 d^{1.5})$ when used with the kNN classifier.

The training of the scalar GMM classifiers for SSCP and USCP is $O(JNd)$, the cost of the EM iteration, where $J$ is the number of GMM components. The same cost applies to using the GMM classifiers to generate labelings. Solving SCP to optimality is an NP-hard problem. As described in Section 3.2.3, we have tried both complete and incomplete algorithms. The runtime of the exact ILP formulation is exponential in the number of sets (features) and in the worst case it has to (implicitly) generate each one of the $2^{|F|}$ possible subsets. The greedy algorithm, on the other hand, runs in quadratic time in the number of sets. Nevertheless, it yields undesirable results from the point of view of feature selection, favoring sets of only a few features (less than 5 in all our datasets). This means that even if solving the exact problem was within practical limits, proving the optimality of covers with a few dense sets would not improve our accuracy. As we proposed, solving the linear relaxation at the root node of the ILP formulation can be done in polynomial time in the general case. In practice, the simplex algorithm and the advances in off-the-shelf mixed-integer programming solvers allow finding a relaxation very fast. In all of our experiments, solving a relaxation of the exact SCP formulation took under a few minutes.

In summary, all the proposed methods can be expected to be clearly faster than methods such as SFS (or the even more complex SFFS) when dealing with large numbers of original features. This was also empirically encountered in our simulations where the proposed methods generally finished within minutes while SFS required days to finish each task.

## 4. Experimental Results

### *4.1. Test Procedure*

The described methods were evaluated in binary classification of presence or absence of seven speaker traits: likability, intelligibility, openness, conscientiousness, extraversion, agreeableness and neuroticism. This was done using the following procedure:

1. *Feature selection:* The Train and Development sets are used together for feature selection. When kNN classification is performed in connection to SFS or MRMR for feature subset evaluation and/or subset size determination, the Train set is used for training and the classification results on the Development set are used to compute the score. The SCP methods apply the data sets in both directions in producing the single-feature classifications: train using the Train set to classify the Development set and vice versa. SCP is then performed over the combined Train and Development set. RSFS uses the Train set for training and the Development set for evaluating the criterion in Eq. 7. SD and MI assess feature relevance using the combined data and labels from the Train and Development sets. The unsupervised DAM method matches the aligned feature value distributions of the combined Train and Development set with the corresponding distributions from the Test set, making no use of the class labels.
2. *Parameter optimization:* Use the Train set for training and the Development set for evaluation in order to determine the best value $k_0$ for the number of neighbors in kNN (see Section 3.1). The number of features to select is also determined in this stage; see Sections 3.2.1-3.2.2 for the nested-subset methods SFS and MRMR and Section 3.4 for the scoring methods MI, SD and DAM.
3. *Classification:* The Train and Development sets are used together for training in order to classify the Test set. Unless otherwise noted, the parameter $k$ for kNN classification is determined according to Eq. 3 with $k_0 \in \{5, 6, \ldots, 150\}$. With subset methods, the best $k_0$ value for Development set classification is chosen. In the case of scoring methods, $k_0$ is chosen so as to provide the best Development set classification performance averaged over each evaluated feature set size.
4. *Evaluation:* Evaluate the classification performance by using unweighted average recall (UAR) as the measure.

### *4.2. Performance of Individual Methods*

Table 2 shows the classification performance (UAR) of a kNN classifier with different feature selection methods on the Development and Test sets. Each of the feature selection methods described in Section 3 was evaluated individually. The performance of three classifiers *without* feature selection, using the full set of 6125 features, is also shown: kNN, linear-kernel support vector machine (SVM) and random forests, the latter two of which were the baseline algorithms of the Interspeech 2012 Speaker Trait Challenge (Schuller et al., 2012). In this and each subsequent classification result table, the best feature selection score for both the Development and Test sets of each subtask is indicated by boldface. In addition, the best scores within a selected set of feature selection methods are shown near the bottom of each table.

The two methods of dimensionality determination, given by Eqs. 12 and 13, were evaluated. While direct optimization of $d$ based on the Development set classification score (Eq. 12) unsurprisingly gave better Development set scores, the randomized version (Eq. 13) produced better scores on the held-out Test data and was thus selected for further evaluations.

Table 2: *Classification performance (UAR %) on the Test sets and Development sets (in parentheses) using kNN with different individual feature selection methods and by itself without feature selection. The Train and Development sets have been used as feature selection data. The number of neighbors for kNN classification and the number of features for MI, SD and DAM have been optimized on the Development set. Two methods of choosing the feature set size have been evaluated. The baselines from the Interspeech 2012 Speaker Trait Challenge obtained using SVM and random forest classification are also shown (Schuller et al., 2012). The subtasks are Likability (L), Intelligibility (I), Openness (O), Conscientiousness (C), Extraversion (E), Agreeableness (A) and Neuroticism (N).*

| Classification method | Feature selection method | Method of dimensionality determination | L | I | O | C | E | A | N | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| kNN | none | | 57.3 | 67.9 | 54.2 | 76.2 | 71.4 | 57.5 | 59.8 | 63.5 |
| | | | (55.6 ) | (66.3 ) | (59.8 ) | (73.6 ) | (81.4 ) | (57.9 ) | (69.3 ) | (66.3 ) |
| | SFS | | 52.1 | 60.2 | 52.6 | 71.7 | **76.1** | 51.3 | 62.5 | 60.9 |
| | | | **(79.5)** | **(80.1)** | **(84.2)** | **(88.1)** | **(98.4)** | **(86.6)** | **(86.8)** | **(86.2)** |
| | MRMR | | 59.3 | 66.2 | 52.0 | 78.6 | 75.1 | 53.9 | **65.6** | 64.4 |
| | | | (57.7 ) | (63.1 ) | (62.3 ) | (76.7 ) | (84.7 ) | (69.2 ) | (71.1 ) | (69.3 ) |
| | SSCP | | 59.0 | 67.8 | 59.3 | 77.0 | 68.6 | 57.2 | 64.5 | 64.8 |
| | | | (61.6 ) | (66.3 ) | (59.5 ) | (74.8 ) | (81.4 ) | (61.6 ) | (72.1 ) | (68.2 ) |
| | USCP | | 54.2 | **68.2** | 54.6 | 77.6 | 71.7 | 52.4 | 63.1 | 63.1 |
| | | | (61.5 ) | (63.4 ) | (62.3 ) | (76.8 ) | (85.2 ) | (62.2 ) | (70.3 ) | (68.8 ) |
| | RSFS | | 58.5 | 64.9 | **59.5** | 76.1 | 75.6 | **59.2** | 60.3 | 64.9 |
| | | | (75.2 ) | (77.1 ) | (77.9 ) | (76.1 ) | (86.3 ) | (75.5 ) | (75.4 ) | (77.6 ) |
| | MI | ranking | 61.1 | 66.3 | 58.0 | 78.6 | 72.0 | 56.7 | 62.0 | 65.0 |
| | | | (65.4 ) | (71.0 ) | (66.9 ) | (77.6 ) | (85.8 ) | (71.8 ) | (73.1 ) | (73.1 ) |
| | SD | ranking | 59.7 | 67.6 | 58.8 | 77.1 | 75.3 | 56.4 | 63.1 | 65.4 |
| | | | (63.7 ) | (71.1 ) | (65.3 ) | (78.6 ) | (85.2 ) | (73.4 ) | (73.8 ) | (73.0 ) |
| | DAM | ranking | 53.9 | 62.8 | 57.2 | 75.1 | 70.9 | 57.1 | 58.6 | 62.2 |
| | | | (58.7 ) | (69.3 ) | (61.6 ) | (74.1 ) | (82.0 ) | (60.5 ) | (70.8 ) | (68.1 ) |
| | MI | randomized | 61.1 | 67.5 | 57.8 | 77.6 | 71.5 | 55.9 | 59.9 | 64.5 |
| | | | (65.4 ) | (70.5 ) | (66.6 ) | (76.1 ) | (86.3 ) | (71.9 ) | (73.5 ) | (72.9 ) |
| | SD | randomized | **62.5** | 67.6 | 59.0 | **79.6** | 75.1 | 58.6 | 59.4 | **65.9** |
| | | | (62.6 ) | (71.1 ) | (66.5 ) | (75.8 ) | (83.0 ) | (72.7 ) | (73.1 ) | (72.1 ) |
| | DAM | randomized | 54.6 | 62.8 | 57.2 | 75.1 | 71.3 | 57.0 | 59.4 | 62.5 |
| | | | (58.3 ) | (68.3 ) | (61.6 ) | (74.5 ) | (80.9 ) | (60.8 ) | (69.4 ) | (67.7 ) |
| Development set best method | | | 58.5 | 64.9 | 59.5 | 77.1 | 75.6 | 59.2 | 60.3 | 65.0 |
| (MRMR,SSCP,USCP,RSFS,SD,DAM) | | | (75.2 ) | (77.1 ) | (77.9 ) | (78.6 ) | (86.3 ) | (75.5 ) | (75.4 ) | (78.0 ) |
| Test set best method | | | 62.5 | 68.2 | 59.5 | 79.6 | 75.6 | 59.2 | 65.6 | 67.2 |
| (MRMR,SSCP,USCP,RSFS,SD,DAM) | | | (62.6 ) | (63.4 ) | (77.9 ) | (75.8 ) | (86.3 ) | (75.5 ) | (71.1 ) | (73.2 ) |
| SVM | none | | 55.9 | 68.4 | 57.8 | 80.1 | 76.2 | 60.2 | 65.9 | 66.4 |
| | | | (58.5) | (61.4) | (60.4) | (74.5) | (80.9) | (67.6) | (68.0) | (67.3) |
| Random forests | none | | 59.0 | 69.6 | 58.8 | 80.1 | 75.3 | 64.2 | 64.5 | 67.4 |
| | | | (57.6) | (65.1) | (57.7) | (74.9) | (82.8) | (67.2) | (68.9) | (67.7) |

In the individual evaluation of the proposed methods, the “fully supervised” methods SSCP, RSFS and SD show improvement in the averaged score upon standard kNN using the full feature set, while the partially unsupervised USCP and the fully unsupervised DAM come close to its performance level. These two unsupervised methods still outperform SFS which shows obvious effects of overlearning. SFS achieves the best Development set score in each task but, with the exception of the extraversion task, this performance does not carry over to the held-out Test sets where SFS is generally the worst of the methods evaluated. The previously published MRMR method (Peng et al., 2005) performs on approximately similar level as the methods proposed in the current study.

### *4.3. Performance of Combined Methods*

Table 3 shows the kNN classification performance (UAR) with combinations of subset selection methods as well as the highest classification baseline obtained with either SVM or random forests using the full feature set. MRMR was included as a representative of earlier methods, as it clearly outperformed SFS in the individual evaluations. The best scores obtained in each task according to Development data (used for feature selection itself as well as for the optimization of the $k$ parameter) and Test data (a held-out data set) are also shown. As MRMR is found to give quite compact feature sets (see Section 4.4), it is not combined with other methods by intersection. Also the intersections of feature sets given by SSCP, USCP and RSFS are relatively small, especially when RSFS is combined with the SCP methods. The set unions are correspondingly larger than the average feature set. In terms of classification performance, set union generally gives higher scores. MRMR generally does not offer advantage to the proposed methods by feature set fusion.

Task-specific features selected by at least two of SSCP, USCP and RSFS are listed in the Appendix.

Table 4 shows the classification performance with combinations of scoring and subset selection methods. Addition appears to be a more effective way of combining the SD and DAM scores than multiplication. Therefore, addition is used for combining these scores whenever they are used for feature ranking in order to refine some initial subset.

The performance of the baseline classification methods is exceeded in most of the tasks by at least one combined feature selection method when used together with the simpler kNN classifier. Again, the Development set classification scores of different methods are not reliable indicators of Test set performance; the method that appears to perform best on the Development set is generally not the actual best method on the Test set, as shown near the bottom of both tables.

### *4.4. Sizes of Feature Sets*

The sizes of feature sets given by different feature selection methods are shown in Table 5. The scoring methods had a maximum allowed feature set size of 500 features while the subset methods did not have a hard limit. SFS was likewise run until a maximum feature set size of 500, which is larger than that eventually selected by the majority of the evaluated approaches in any task (the only exceptions being RSFS in the extraversion task and the set-union combined methods). It can be noticed that, apart from combinations obtained solely by set union or intersection, the methods generally return feature sets of roughly the same size which is close to 5 % of the total number of features. Of the individual methods, MRMR produces the most compact sets, a result that could be expected on the basis of the fundamental principle of MRMR to avoid feature redundancy.

## 5. Discussion

All the proposed feature subset selection algorithms (RSFS, SSCP and the partially unsupervised USCP) and feature scoring algorithms (SD as well as DAM, the unsupervised method) outperformed conventional sequential forward selection (SFS) when kNN was used as the classification rule. In classification of the Development data sets used in the feature selection process itself, however, SFS achieved the best score of all the methods in each of the seven classification tasks but this did not carry over to the held-out Test sets. This result clearly demonstrates, once again, the pitfall of overlearning in feature selection optimized on a single data set (Reunanen, 2003, 2012; Smialowski et al., 2010), which is particularly dangerous when the number of features greatly exceeds the number of instances and the number of potential feature sets is huge. The proposed methods were thus shown to be more resistant against overlearning in such sparse high-dimensional problems than the conventional, greedy hill-climbing forward selection, which is widely used today. In comparison to SFS, MRMR generalizes better to new test data and leads to compact feature sets. However, despite its higher computational complexity, it does not surpass in performance the feature selection methods proposed in the current study.

The proposed methods, when used by themselves, managed to reduce the feature space dimensionality on the average to between 2.0 % and 6.2 % of the maximum dimensionality of 6125 features (see Table 5). In any practical paralinguistic analysis application, e.g., using a simple classifier such as the kNN used in the present study, this means a huge improvement in terms of both the computational load and memory requirements of the system. At the same time, classification performance was preserved or improved: the new methods achieved comparable or better performance on the Test sets than kNN using the full feature set. Moreover, despite being confined to basic kNN classification, the individual feature selection methods managed to exceed the performance of the baseline high-dimensional classifiers, namely SVM and random forests, in certain cases. The results show promise for building simple and efficient classifiers using various classification methods. Since simpler classifiers often lead to better generalization (Duda et al., 2001), various classification systems employing the proposed feature selection methods can be expected to have better generalization ability than the corresponding higher-dimensional ones would have. A logical next step would be to evaluate the feature selection methods in combination with other, potentially more robust classifiers, such as SVM, in paralinguistic analysis tasks. However, the classification baseline results of the current study already seem to point out that different tasks are most easily solved by different classifiers, as seen in the differences between kNN, SVM and random forests in many of the tasks. Indeed, a single best generic classification approach for these types of paralinguistic analysis tasks probably does not exist (cf. the No Free Lunch Theorem; Duda et al. 2001).

One general trend in the results is the variability of algorithm performance across different tasks. In addition to the differences between classification methods across the seven analysis tasks, none of the studied individual feature selection approaches perform clearly better than the others. Instead, the best results in the seven tasks are obtained by five different feature selection algorithms. Also, some algorithms that perform well in some tasks fall far below the classification baseline in others (for example, USCP is the best in the intelligibility task and the second worst in the agreeableness task). These observations may reflect the large differences in nature and complexity among the analysis tasks, as already evidenced by the

Table 3: *Classification performance (UAR %) on the Test sets and Development sets (in parentheses) using combinations of feature subset selection methods. The Train and Development sets have been used as feature selection data. Union of subsets is denoted by + and intersection by ×. Also shown are the official baselines of the Interspeech 2012 Speaker Trait Challenge obtained using SVM and random-forest classification (Schuller et al., 2012).*

| Subset selection | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MRMR | SSCP | USCP | RSFS | L | I | O | C | E | A | N | Mean |
| | × | × | | 58.4 | 66.2 | 50.2 | 64.9 | 73.4 | 54.4 | 62.0 | 61.3 |
| | | | | (59.5 ) | (62.4 ) | (54.1 ) | (70.5 ) | (84.7 ) | (64.5 ) | (73.6 ) | (67.0 ) |
| | × | | × | 50.7 | 63.2 | 54.9 | 67.6 | 71.8 | 52.4 | **65.4** | 60.9 |
| | | | | (68.1 ) | (70.6 ) | (61.4 ) | (77.4 ) | (86.3 ) | (74.0 ) | (76.8 ) | (73.5 ) |
| | | × | × | 51.5 | 62.8 | 57.1 | 72.1 | 73.3 | 56.3 | 63.9 | 62.4 |
| | | | | (69.7 ) | (67.8 ) | (62.4 ) | (77.4 ) | (84.7 ) | (73.0 ) | (75.8 ) | (73.0 ) |
| | × | × | × | 47.6 | 59.8 | 54.0 | 62.5 | 74.8 | 53.6 | 58.9 | 58.7 |
| | | | | (60.9 ) | (61.7 ) | (55.6 ) | (71.4 ) | (85.3 ) | (71.4 ) | (73.3 ) | (68.5 ) |
| | + | + | | 50.4 | 68.1 | 52.6 | 77.6 | 71.4 | 55.0 | 60.6 | 62.2 |
| | | | | (61.2 ) | (66.5 ) | (61.3 ) | (75.1 ) | (84.1 ) | (61.9 ) | (71.2 ) | (68.8 ) |
| | + | | + | 59.8 | 66.2 | 53.9 | **78.6** | 73.3 | 61.5 | 61.9 | 65.0 |
| | | | | **(77.4)** | (75.1 ) | (72.7 ) | (78.4 ) | (86.9 ) | (72.1 ) | (77.9 ) | **(77.2)** |
| | | + | + | 56.2 | 67.0 | 56.4 | 78.1 | 73.8 | **64.5** | 60.9 | **65.3** |
| | | | | (74.7 ) | **(76.0)** | (69.9 ) | (79.2 ) | (86.9 ) | **(74.8)** | (74.7 ) | (76.6 ) |
| | + | + | + | 56.8 | 67.4 | 56.8 | **78.6** | 72.2 | 62.1 | 61.4 | 65.1 |
| | | | | (74.0 ) | (75.2 ) | (67.9 ) | **(79.5)** | **(87.4)** | (69.3 ) | (76.2 ) | (75.6 ) |
| + | + | | | **61.4** | 67.0 | **58.0** | 77.2 | 72.6 | 55.8 | 61.8 | 64.8 |
| | | | | (59.9 ) | (66.9 ) | (61.8 ) | (75.8 ) | (85.2 ) | (64.5 ) | (72.2 ) | (69.5 ) |
| + | | + | | 54.3 | **69.1** | 53.0 | 75.2 | 71.3 | 57.5 | 62.5 | 63.3 |
| | | | | (60.7 ) | (65.1 ) | (61.6 ) | (74.8 ) | (85.2 ) | (63.2 ) | (71.2 ) | (68.8 ) |
| + | | | + | 57.7 | 66.4 | 55.5 | 76.1 | **76.8** | 59.1 | 60.3 | 64.6 |
| | | | | (75.1 ) | (74.5 ) | **(75.4)** | (78.7 ) | (85.8 ) | **(74.8)** | (74.8 ) | (77.0 ) |
| + | + | + | | 56.7 | 68.2 | 55.2 | 75.7 | 70.7 | 53.4 | 62.5 | 63.2 |
| | | | | (60.5 ) | (67.1 ) | (60.8 ) | (74.8 ) | (84.7 ) | (60.7 ) | (70.7 ) | (68.5 ) |
| + | + | | + | 56.4 | 66.8 | 51.8 | 78.1 | 73.4 | 62.2 | 61.5 | 64.3 |
| | | | | (76.2 ) | (73.2 ) | (72.5 ) | (77.5 ) | **(87.4)** | (72.2 ) | **(78.4)** | (76.8 ) |
| + | | + | + | 58.1 | 68.4 | 54.1 | 77.6 | 74.3 | 61.6 | 60.9 | 65.0 |
| | | | | (74.2 ) | (73.9 ) | (69.0 ) | (76.3 ) | (86.4 ) | (74.3 ) | (75.1 ) | (75.6 ) |
| + | + | + | + | 57.4 | 68.0 | 51.8 | 78.1 | 74.8 | 60.9 | 61.4 | 64.6 |
| | | | | (73.8 ) | (72.7 ) | (66.5 ) | (78.2 ) | (86.9 ) | (68.7 ) | (76.2 ) | (74.7 ) |
| Development set best method | | | | 59.8 | 67.0 | 55.5 | 78.6 | 73.4 | 64.5 | 61.5 | 65.8 |
| (MRMR,SSCP,USCP,RSFS combinations) | | | | (77.4 ) | (76.0 ) | (75.4 ) | (79.5 ) | (87.4 ) | (74.8 ) | (78.4 ) | (78.4 ) |
| Test set best method | | | | 61.4 | 69.1 | 58.0 | 78.6 | 76.8 | 64.5 | 65.4 | 67.7 |
| (MRMR,SSCP,USCP,RSFS combinations) | | | | (59.9 ) | (65.1 ) | (61.8 ) | (79.5 ) | (85.8 ) | (74.8 ) | (76.8 ) | (71.9 ) |
| Best of SVM, random forests | | | | 59.0 | 69.6 | 58.8 | 80.1 | 76.2 | 64.2 | 65.9 | 67.7 |
| | | | | (57.6) | (65.1) | (57.7) | (74.9) | (80.9) | (67.2) | (68.0) | (67.3) |

differences among their baseline or average classification scores.

The performance on the development set was generally a poor indicator of algorithm performance on the held-out test set. This not only shows the importance of measuring the ultimate generalization with an independent test set (cf. Reunanen 2003), but also indicates, on a related note, how strict optimization of development set performance may guide the system to a highly overfitted solution. While this very obviously happened with sequential forward selection, which makes no attempt to avoid it, it was also observed in varying degrees with some of the proposed methods, despite the various attempts to avoid the most distinct local maxima in the criterion function. These included stochastic sampling, adapted quantization scales, unsupervision, randomized dimensionality determination, computation of the criterion function across various values of k in kNN etc.

Even though it proved to be difficult to predict the performance of any selection method on the held-out test sets, the results show that the combination of multiple feature selection criteria has the potential to improve feature selection performance and even to exceed the performance level of state-of-the-art high-dimensional classifiers, i.e., SVM and random forests, using the simple nearest-neighbor classification rule. The performance of these clas-

Table 4: *Classification performance (UAR %) on the Test sets and Development sets (in parentheses) using combinations of feature subset selection and scoring methods. The Train and Development sets have been used as feature selection data. Union of subsets is denoted by + and intersection by ×. For combining scoring methods, the same symbols refer to addition and multiplication. When combining subset methods with scoring methods, the latter are used as a refinement to the former according to Section 3.5.1. Also shown are the official baselines of the Interspeech 2012 Speaker Trait Challenge obtained using SVM and random-forest classification (Schuller et al., 2012).*

| Subset selection | | | Scoring | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSCP | USCP | RSFS | SD | DAM | L | I | O | C | E | A | N | Mean |
| | | | × | × | 56.8<br>(63.3 ) | 65.2<br>(70.0 ) | 49.9<br>(63.7 ) | 78.5<br>(76.7 ) | 75.4<br>(84.7 ) | **62.8**<br>(66.0 ) | 59.7<br>(72.3 ) | 64.1<br>(71.0 ) |
| | | | + | + | 55.8<br>(64.0 ) | 65.5<br>(70.7 ) | 56.5<br>(62.5 ) | **79.5**<br>(76.4 ) | 76.4<br>(83.6 ) | 60.8<br>(67.6 ) | 61.8<br>(72.9 ) | **65.2**<br>(71.1 ) |
| × | | | × | | 61.2<br>(61.6 ) | 66.5<br>(67.4 ) | 57.2<br>(66.5 ) | 72.6<br>(77.6 ) | 74.5<br>(84.1 ) | 53.1<br>(65.4 ) | 62.8<br>(75.1 ) | 64.0<br>(71.1 ) |
| | × | | × | | 51.4<br>(68.0 ) | 68.0<br>(67.9 ) | 55.7<br>(62.3 ) | 74.6<br>(78.2 ) | 71.6<br>(85.8 ) | 55.4<br>(69.9 ) | **65.3**<br>(75.9 ) | 63.2<br>(72.6 ) |
| | | × | × | | 58.7<br>**(78.7)** | 64.3<br>**(78.1)** | 56.6<br>**(79.0)** | 74.6<br>(78.0 ) | 75.3<br>(87.4 ) | 58.3<br>(75.5 ) | 58.9<br>(75.9 ) | 63.8<br>(78.9 ) |
| + | + | + | × | | 58.4<br>(72.9 ) | 66.3<br>(74.5 ) | 56.8<br>(70.2 ) | 76.1<br>(77.7 ) | 75.4<br>(86.3 ) | 55.8<br>(71.9 ) | 59.8<br>(75.8 ) | 64.1<br>(75.6 ) |
| × | | | | × | 59.8<br>(61.6 ) | 66.6<br>(67.3 ) | 53.9<br>(64.8 ) | 76.5<br>(77.2 ) | 71.4<br>(85.2 ) | 55.3<br>(64.2 ) | 64.4<br>(76.1 ) | 64.0<br>(70.9 ) |
| | × | | | × | 51.1<br>(67.0 ) | **70.4**<br>(66.1 ) | 53.0<br>(64.3 ) | 79.1<br>(76.8 ) | 72.1<br>(86.9 ) | 52.4<br>(62.2 ) | 60.8<br>(74.4 ) | 62.7<br>(71.1 ) |
| | | × | | × | 57.5<br>(77.8 ) | 64.1<br>(77.8 ) | 58.9<br>(77.9 ) | 74.1<br>**(79.8)** | **76.8**<br>**(88.5)** | 59.4<br>(75.5 ) | 60.4<br>**(78.0)** | 64.4<br>**(79.3)** |
| + | + | + | | × | 59.5<br>(74.8 ) | 66.3<br>(76.1 ) | 55.7<br>(71.4 ) | 75.6<br>(79.2 ) | 73.3<br>(88.0 ) | 59.4<br>(70.6 ) | 62.4<br>(76.8 ) | 64.6<br>(76.7 ) |
| × | | | + | + | 57.5<br>(63.9 ) | 67.1<br>(67.3 ) | 51.2<br>(67.1 ) | 77.1<br>(77.4 ) | 72.0<br>(85.2 ) | 56.7<br>(65.0 ) | 62.0<br>(75.4 ) | 63.4<br>(71.6 ) |
| | × | | + | + | 51.6<br>(61.5 ) | 69.8<br>(69.4 ) | 52.0<br>(64.6 ) | 79.1<br>(76.8 ) | 71.5<br>(85.8 ) | 56.5<br>(69.2 ) | 62.2<br>(73.3 ) | 63.2<br>(71.5 ) |
| | | × | + | + | 55.9<br>(76.3 ) | 65.7<br>(78.0 ) | **60.5**<br>(78.4 ) | 75.1<br>(77.4 ) | 73.8<br>(86.9 ) | 57.4<br>**(76.1)** | 61.5<br>(75.9 ) | 64.3<br>(78.4 ) |
| + | + | + | + | + | **61.4**<br>(75.1 ) | 66.0<br>(75.6 ) | 54.7<br>(74.6 ) | 77.1<br>(78.1 ) | 74.8<br>(86.3 ) | 60.4<br>(70.9 ) | 59.1<br>(74.5 ) | 64.8<br>(76.5 ) |
| Development set best method<br>(all combined methods, Tables 3-4) | | | | | 58.7<br>(78.7 ) | 64.3<br>(78.1 ) | 56.6<br>(79.0 ) | 74.1<br>(79.8 ) | 76.8<br>(88.5 ) | 57.4<br>(76.1 ) | 61.5<br>(78.4 ) | 64.2<br>(79.8 ) |
| Test set best method<br>(all combined methods, Tables 3-4) | | | | | 61.4<br>(75.1 ) | 70.4<br>(66.1 ) | 60.5<br>(78.4 ) | 79.5<br>(76.4 ) | 76.8<br>(88.5 ) | 64.5<br>(74.8 ) | 65.4<br>(76.8 ) | 68.4<br>(76.6 ) |
| Best of SVM, random forests | | | | | 59.0<br>(57.6) | 69.6<br>(65.1) | 58.8<br>(57.7) | 80.1<br>(74.9) | 76.2<br>(80.9) | 64.2<br>(67.2) | 65.9<br>(68.0) | 67.7<br>(67.3) |

sifiers was exceeded in most cases (11 out of 14) using at least one (combined) feature selection method. This suggests another potential direction for future research: study the use of another held-out validation data set in order to select a suitable combined feature selection method for each analysis task. It would be interesting to investigate whether the benefit of an additional validation data set, used solely for selecting a feature selection method, would outweigh the cost of having a smaller portion of the data available for the feature selection process itself.

The most effective combination methods were the union of sets and refinement of subsets by score-based ranking, whereas intersection of multiple feature sets led to very compact but still reasonably well performing systems, also revealing the most relevant features for each analysis task. Combination of supervised and unsupervised feature selection criteria showed promise, as the best-performing combined method in five out of seven analysis tasks included the fully unsupervised DAM method based on matching feature value distributions between data sets.

## 6. Conclusions

Classification of high-dimensional paralinguistic speaker trait data was approached with a special focus on feature selection. Several new feature selection algorithms with different supervised, partially supervised and unsupervised selection criteria were presented, as well as methods for combining the algorithms. These were evaluated and compared against widely used baseline methods from

Table 5: *The size of feature set returned by different (individual and combined) feature selection methods; see captions of Tables 2-4 for details. For the scoring methods SD and DAM, the ranking-based and randomized methods of feature set size determination (see Section 3.4) are denoted by 1) and 2), respectively. The randomized method of size determination is used with the combined selection algorithms.*

| Subset selection | | | Scoring | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSCP | USCP | RSFS | SD | DAM | L | I | O | C | E | A | N | Mean |
| $\times$ | | | | | 406 | 318 | 297 | 287 | 273 | 322 | 312 | 316 |
| | $\times$ | | | | 423 | 347 | 314 | 293 | 291 | 337 | 320 | 332 |
| | | $\times$ | | | 263 | 454 | 205 | 378 | 649 | 315 | 402 | 381 |
| | | | $\times^{1)}$ | | 75 | 425 | 81 | 13 | 129 | 15 | 99 | 120 |
| | | | | $\times^{1)}$ | 185 | 396 | 447 | 350 | 297 | 367 | 495 | 362 |
| | | | $\times^{2)}$ | | 278 | 425 | 83 | 369 | 422 | 50 | 414 | 292 |
| | | | | $\times^{2)}$ | 97 | 463 | 447 | 349 | 497 | 314 | 375 | 363 |
| $\times$ | $\times$ | | | | 123 | 67 | 85 | 65 | 72 | 78 | 75 | 81 |
| $\times$ | | $\times$ | | | 22 | 27 | 19 | 26 | 34 | 25 | 39 | 27 |
| | $\times$ | $\times$ | | | 26 | 21 | 23 | 22 | 51 | 21 | 43 | 30 |
| $\times$ | $\times$ | $\times$ | | | 7 | 2 | 5 | 8 | 16 | 7 | 12 | 8 |
| + | + | | | | 706 | 598 | 526 | 515 | 492 | 581 | 557 | 568 |
| + | | + | | | 647 | 745 | 483 | 639 | 888 | 612 | 675 | 670 |
| | + | + | | | 660 | 780 | 496 | 649 | 889 | 631 | 679 | 683 |
| + | + | + | | | 928 | 1006 | 694 | 853 | 1072 | 857 | 889 | 900 |
| | | | $\times$ | $\times$ | 375 | 482 | 421 | 173 | 110 | 181 | 389 | 304 |
| | | | + | + | 150 | 490 | 231 | 375 | 442 | 68 | 344 | 300 |
| $\times$ | | | $\times$ | | 406 | 301 | 11 | 105 | 162 | 226 | 173 | 198 |
| | $\times$ | | $\times$ | | 71 | 86 | 314 | 183 | 290 | 91 | 64 | 157 |
| | | $\times$ | $\times$ | | 209 | 450 | 199 | 348 | 480 | 315 | 233 | 319 |
| + | + | + | $\times$ | | 458 | 497 | 399 | 349 | 427 | 418 | 447 | 428 |
| $\times$ | | | | $\times$ | 406 | 301 | 116 | 190 | 150 | 143 | 144 | 207 |
| | $\times$ | | | $\times$ | 376 | 280 | 151 | 293 | 284 | 337 | 159 | 269 |
| | | $\times$ | | $\times$ | 224 | 452 | 205 | 162 | 259 | 315 | 373 | 284 |
| + | + | + | | $\times$ | 432 | 421 | 315 | 326 | 314 | 412 | 323 | 363 |
| $\times$ | | | + | + | 123 | 306 | 46 | 132 | 148 | 274 | 249 | 183 |
| | $\times$ | | + | + | 423 | 143 | 193 | 293 | 290 | 72 | 223 | 234 |
| | | $\times$ | + | + | 260 | 449 | 202 | 280 | 408 | 314 | 169 | 297 |
| + | + | + | + | + | 494 | 488 | 454 | 287 | 370 | 215 | 487 | 399 |
| SFS | | | | | 165 | 162 | 456 | 453 | 304 | 368 | 119 | 290 |
| MRMR | | | | | 10 | 328 | 28 | 233 | 112 | 12 | 5 | 104 |

the perspective of both feature selection and pattern classification. In addition, combined selection methods were used to identify the most relevant features for the seven analysis tasks consisting of speaker likability, intelligibility and the Big Five personality traits.

The results demonstrate five things: 1) the proposed methods are more resistant against overlearning in feature selection than conventional, hill-climbing forward selection; 2) a huge reduction of feature space dimensionality is achieved without sacrificing performance on the held-out test data, indicating potential computational savings; 3) furthermore, the nearest-neighbor classification performance is improved by many individual and combined feature selection methods, suggesting a potentially improved ability of classifiers to generalize with limited training data when using the proposed methods for feature selection; 4) due to different amounts of overfitting generally shown by different feature selection algorithms, the performance of any given method is difficult to predict without independent evaluation data; and 5) by combining supervised and unsupervised feature selection methods and a basic classifier, the performance of state-of-the-art high-dimensional pattern classification methods can be reached. In future research, potential directions suggested by the results are automatic selection of a feature selection method using independent evaluation data and the application of the proposed methods to various practical analysis problems with different classifiers.

## 7. Acknowledgements

The work of J. Pohjalainen was supported by Academy of Finland (256961) and the EC FP7 project Simple4All (287678). The work of O. Räsänen was performed as a part of the Data to Intelligence (D2I) project funded by Tekes, Finland. S. Kadioglu was supported by Paris Kanellakis fellowship at Brown University when conducting the work contained in this document. This paper reflects his opinions and should not be interpreted, either expressed or implied, as those of his current employer.

The authors would like to thank the two anonymous reviewers for their valuable comments.

MATLAB implementations of some of the studied feature selection methods can be found online at `http://www.acoustics.hut.fi/~jpohjala/featureselection/`.

## References

Batliner, A., Steidl, S., Schuller, B., Seppi, D., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Aharonson, V., Kessous, L., Amir, N., 2010. Whodunnit – searching for the most important feature types signalling emotion-related user states in speech. Computer Speech and Language 25 (1), 4–28.

Beasley, J., 1990. Or-library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41, 1069–1072.

Blum, A. L., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271.

Breiman, L., 2001. Random forests. Machine Learning 3, 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984. Classification and Regression Trees. Wadsworth International, Belmont, CA.

Burkhardt, F., Eckert, M., Johannsen, W., Stegmann, J., May 19–21 2010. A database of age and gender annotated telephone speech. In: Proc. International Conference on Language Resources and Evaluation (LREC). Valletta, Malta, pp. 1562–1565.

Burkhardt, F., Schuller, B., Weiss, B., Weninger, F., August 27–31 2011. 'Would you buy a car from me?' – on the likability of telephone voices. In: Proc. 12th Annual Conference of the International Speech Communication Association (Interspeech). Florence, Italy, pp. 1557–1560.

Caprara, A., Fischetti, M., Toth, P., Vigo, D., Guida, P. L., 1997. Algorithms for railway crew management. Mathematical Programming 79, 125–141.

Chvátal, V., 1979. A greedy heuristic for the set-covering problem. Mathematics of Operations Research 4 (3), 233–235.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. Introduction to Algorithms, 2nd Edition. McGraw-Hill Higher Education, Cambridge, MA.

Dempster, A. P., Laird, N. M., B.Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B 39, 1–38.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern Classification. John Wiley and Sons Inc., New York, NY.

Eyben, F., Wöllmer, M., Schuller, B., October 25–29 2010. openSMILE – the Munich versatile and fast open-source audio feature extractor. In: Proc. ACM Multimedia. Florence, Italy, pp. 1459–1462.

Furui, S., February 1986. Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Transactions on Acoustics, Speech and Signal Processing 34 (1), 52–59.

Grimm, M., Kroschel, K., 2005. Evaluation of natural emotions using self assessment manikins. In: Proc. ASRU. pp. 381–385.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research (3), 1157–1182.

Hall, M., 1999. Correlation-based Feature Selection for Machine Learning. University of Waikato, Hamilton, New Zealand, Ph.D. Thesis.

Hess, W., 1983. Pitch Determination of Speech Signals. Springer-Verlag, Berlin.

Hoffmann, K. L., Padberg, M. W., 1993. Solving airline crew scheduling problems by branch-and-cut. Management Science 39 (6), 657–682.

Housos, E., Elmoth, T., 1997. Automatic optimization of subproblems in scheduling airline crews. Interfaces 27 (5), 68–77.

Huang, X., Acero, A., Hon, H.-W., 2001. Spoken Language Processing. Prentice Hall PTR.

IBM, 2009. IBM CPLEX reference manual and user manual. v12.1.

Kadioglu, S., Sellmann, M., 2009. Dialectic search. In: Gent, I. P. (Ed.), CP. Vol. 5732 of Lecture Notes in Computer Science. Springer, Berlin, pp. 486–500.

Karp, R. M., 1972. Reducibility among combinatorial problems. Complexity of Computer Computations, 85–103.

Katsavounidis, I., Kuo, C.-C. J., Zhang, Z., 1994. A new initialization technique for generalized Lloyd iteration. IEEE Signal Processing Letters 1 (10), 144–146.

Kohavi, R., John, G., 1997. Wrappers for feature subset selection. Artificial Intelligence 97, 273–324.

Li, S., Harner, J., Adjeroh, D., 2011. Random kNN feature selection – a fast and stable alternative to random forests. BMC Bioinformatics 12.

Marill, T., Green, D., 1963. On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory 9 (1), 11–17.

Mermelstein, P., 1975. Automatic segmentation of speech into syllabic units. Journal of the Acoustical Society of America 58 (4), 880–883.

Mohammadi, G., Vinciarelli, A., 2012. Automatic personality perception: Prediction of trait attribution based on prosodic features. IEEE Transactions on Affective Computing 3 (3), 273–284.

Nemhauser, G. L., Wolsey, L. A., 1988. Integer and Combinatorial Optimization. John Wiley and Sons Inc., New York, NY.

O'Shaughnessy, D., 2000. Speech Communications: Human and Machine, 2nd Edition. IEEE Press, New York, NY.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8), 1226–1238.

Pohjalainen, J., Kadioglu, S., Räsänen, O., September 9–13 2012. Feature selection for speaker traits. In: Proc. 13th Annual Conference of the International Speech Communication Association (Interspeech). Portland, Oregon, USA.

Pudil, P., Novovicová, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125.

Rabiner, L. R., Schafer, R. W., 1978. Digital Processing of Speech Signals. Prentice-Hall, Upper Saddle River, NJ.

Rammstedt, B., John, O., 2007. Measuring personality in one minute or less: A 10-item short version of the big five inventory in English and German. Journal of Research in Personality 41, 203–212.

Räsänen, O., Pohjalainen, J., August 25–29 2013. Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech. In: Proc. 14th Annual Conference of the International Speech Communication Association (Interspeech). Lyon, France.

Reunanen, J., 2003. Overfitting in making comparisons between variable selection methods. Journal of Machine Learning Research (3), 1371–1382.

Reunanen, J., 2012. Overfitting in Feature Selection: Pitfalls and Solutions. Aalto University, Espoo, Finland, Ph.D. Thesis.

Saeys, Y., Abeel, T., de Peer, Y. V., 2008. Robust feature selection using ensemble feature selection techniques. In: Proc. European conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '08). pp. 313–325.

Scheirer, E., Slaney, M., Apr. 1997. Construction and evaluation of a robust multifeature speech/music discriminator. In: Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97). Munich, Germany, pp. 1331–1334.

Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C., Narayanan, S., 2013. Paralinguistics in speech and language - state-of-the-art and the challenge. Computer Speech and Language 27, 4–39.

Schuller, B., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, E., Burkhardt, F., van Son, R., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G., Weiss, B., September 9–13 2012. The Interspeech 2012 speaker trait challenge. In: Proc. 13th Annual Conference of the International Speech Communication Association (Interspeech). Portland, Oregon, USA.

Smialowski, P., Frishman, D., Kramer, S., 2010. Pitfalls of supervised feature selection. Bioinformatics 26 (3), 440–443.

Theodoridis, S., Koutroumbas, K., 2003. Pattern Recognition, 2nd Edition. Academic Press, Amsterdam.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1971. The location of emergency service facilities. Operational Research 19 (6), 1363–1373.

van der Molen, L., van Rossum, M. A., Jacobi, I., van Son, R. J. J. H., Smeele, L. E., Rasch, C. R. N., Hilgers, F. J. M., 2012. Pre- and posttreatment voice and speech outcomes in patients with advanced head and neck cancer treated with chemoradiotherapy: Expert listeners' and patient's perception. Journal of Voice 26 (5), 664.e25–664.e33.

Vasko, F. J., Wolf, F. E., 1987. Optimal selection of ingot sizes via set covering. Operational Research 35, 115–121.

Vazirani, V. V., 2001. Approximation Algorithms. Springer-Verlag, Berlin.

Whitney, A. W., 1971. A direct method of nonparametric measurement selection. IEEE Transactions on Computers 20, 1100–1103.

Xu, L., Jordan, M. I., Jan. 1996. On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation 8 (1), 129–151.

Zwicker, E., Fastl, H., 1990. Psychoacoustics, Facts and Models. Springer-Verlag, Berlin.

## Appendix A. Discovered Features for Likability, Intelligibility and Personality Traits

Table A.6: *Features selected for Likability from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods SSCP and RSFS (Test set UAR 55.0%).*

| **Likability** |
|---|
| pcm_ RMSenergy_ sma_ kurtosis |
| audSpec_ Rfilt_ sma[3]_ downleveltime75 |
| mfcc_ sma[4]_ quartile3 |
| mfcc_ sma[5]_ lpc1 |
| mfcc_ sma[5]_ lpc2 |
| mfcc_ sma[7]_ lpgain |
| mfcc_ sma[8]_ falltime |
| mfcc_ sma[11]_ lpc0 |
| mfcc_ sma[13]_ iqr2-3 |
| audSpec_ Rfilt_ sma_ de[1]_ minPos |
| audSpec_ Rfilt_ sma_ de[2]_ iqr2-3 |
| audSpec_ Rfilt_ sma_ de[16]_ quartile2 |
| audSpec_ Rfilt_ sma_ de[22]_ downleveltime75 |
| pcm_ Mag_ psySharpness_ sma_ de_ lpc3 |
| mfcc_ sma_ de[10]_ lpc0 |
| F0final_ sma_ minPos |
| F0final_ sma_ percentile1.0 |
| voicingFinalUnclipped_ sma_ skewness |
| audSpec_ Rfilt_ sma[0]_ linregc1 |
| audSpec_ Rfilt_ sma[6]_ peakRangeRel |
| mfcc_ sma[9]_ peakDistStddev |
| mfcc_ sma_ de[5]_ posamean |

Table A.7: *Features selected for Intelligibility from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods USCP and RSFS (Test set UAR 63.5 %).*

| **Intelligibility** |
|---|
| pcm_ RMSenergy_ sma_ de_ lpc2 |
| audSpec_ Rfilt_ sma[0]_ percentile99.0 |
| pcm_ Mag_ spectralSkewness_ sma_ range |
| pcm_ Mag_ spectralSkewness_ sma_ pctlrange0-1 |
| pcm_ Mag_ spectralKurtosis_ sma_ stddev |
| mfcc_ sma[4]_ quartile3 |
| mfcc_ sma[9]_ iqr2-3 |
| mfcc_ sma[11]_ upleveltime50 |
| pcm_ Mag_ spectralSkewness_ sma_ de_ percentile1.0 |
| mfcc_ sma_ de[3]_ quartile1 |
| mfcc_ sma_ de[9]_ pctlrange0-1 |
| mfcc_ sma_ de[14]_ iqr1-3 |
| shimmerLocal_ sma_ de_ quartile3 |
| shimmerLocal_ sma_ de_ stddev |
| shimmerLocal_ sma_ de_ lpgain |
| audSpec_ Rfilt_ sma[0]_ flatness |
| pcm_ Mag_ fband250-650_ sma_ qregc3 |
| pcm_ Mag_ spectralFlux_ sma_ peakRangeAbs |
| mfcc_ sma[12]_ linregc2 |
| shimmerLocal_ sma_ amean |
| pcm_ Mag_ spectralRollOff50.0_ sma_ de_ flatness |

Our approach of selecting features based on different criteria permits us to single out certain features that tend to be favored across different feature selection criteria. Because of the generally good performance shown by the complete subset selection methods SSCP, USCP and RSFS in combination with each other as well as with other methods (see Tables 3-4), we chose to study the features that were selected by at least two of them in a given classification problem. To select the best combination for each task, intersections of subsets given by these methods were evaluated in kNN classification with $k$ optimized on the Development set according to Eq. 3 with $k_0 \in \{5, 10, 15, \ldots, 150\}$. Tables A.6-A.12 show the features that were selected by the best intersection-combination of these methods, as found by the described procedure (which differs from those of the previous sections in the allowed values of $k$), in the likability, intelligibility and personality classification tasks. The abbreviations have been given by the organizers of the Speaker Trait Challenge and explanation on their meaning can be found in the paper on the challenge (Schuller et al., 2012).

Overall, the popular features are based on various different low-level descriptors (LLDs), including energy and zero-crossing rate (ZCR; Rabiner and Schafer 1978), spectral skewness and kurtosis, spectral flux and roll-off points (Scheirer and Slaney, 1997), mel-frequency cepstral coefficients (MFCCs; Huang et al. 2001; Mermelstein 1975), pitch (F0) estimates (Hess, 1983) and psychoacoustic sharpness (Zwicker and Fastl, 1990). It can also be observed that the compact feature sets selected for different tasks are very distinct. In fact, no features were selected by the described combined method for more than two tasks and only nine features were selected for two tasks. Depending on the analysis task, different long-term functionals appear effective in modeling the time behavior of the

Table A.8: *Features selected for Openness from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods USCP and RSFS (Test set UAR 57.1 %).*

| **Opennness** |
|---|
| audspec_ lengthL1norm_ sma_ de_ risetime |
| audSpec_ Rfilt_ sma[8]_ lpc0 |
| audSpec_ Rfilt_ sma[23]_ skewness |
| pcm_ Mag_ spectralVariance_ sma_ lpc2 |
| pcm_ Mag_ psySharpness_ sma_ iqr2-3 |
| mfcc_ sma[3]_ lpc1 |
| mfcc_ sma[5]_ iqr1-3 |
| mfcc_ sma[5]_ upleveltime90 |
| mfcc_ sma[10]_ upleveltime90 |
| mfcc_ sma[12]_ maxPos |
| mfcc_ sma[13]_ upleveltime75 |
| audSpec_ Rfilt_ sma_ de[0]_ upleveltime25 |
| audSpec_ Rfilt_ sma_ de[25]_ upleveltime90 |
| pcm_ Mag_ fband1000-4000_ sma_ de_ upleveltime90 |
| pcm_ Mag_ spectralRollOff75.0_ sma_ de_ maxSegLen |
| mfcc_ sma_ de[2]_ lpc0 |
| mfcc_ sma_ de[3]_ lpc1 |
| mfcc_ sma_ de[6]_ downleveltime75 |
| mfcc_ sma_ de[9]_ lpgain |
| audSpec_ Rfilt_ sma[19]_ peakDistStddev |
| pcm_ Mag_ fband1000-4000_ sma_ qregc2 |
| mfcc_ sma[5]_ meanPeakDist |
| mfcc_ sma[5]_ linregerrQ |

LLDs and their delta features (Furui, 1986; Huang et al., 2001) to yield the utterance-level features.

The feature sets listed in Tables A.6-A.12 worked reasonably well in the classification tasks; their average UAR was 64.0 %, i.e., roughly the same as that of the complete feature set. In addition, the features contained in them were independently picked by at least two algorithms based on very different considerations (SCP and RSFS). Finally, as mentioned above, the features selected seem rather specific to the task – there are no generic features selected for most tasks. Therefore, these features can justifiably be assumed to have at least some relevance in the paralinguistic analysis tasks they were selected for.

Table A.9: *Features selected for Conscientiousness from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods USCP and RSFS (Test set UAR 74.6 %).*

| **Conscientiousness** |
|---|
| audSpec_ Rfilt_ sma[25]_ quartile1 |
| pcm_ Mag_ spectralFlux_ sma_ quartile3 |
| mfcc_ sma[1]_ downleveltime25 |
| mfcc_ sma[6]_ risetime |
| mfcc_ sma[11]_ falltime |
| audSpec_ Rfilt_ sma_ de[4]_ iqr2-3 |
| audSpec_ Rfilt_ sma_ de[18]_ downleveltime25 |
| audSpec_ Rfilt_ sma_ de[20]_ quartile3 |
| pcm_ Mag_ fband250-650_ sma_ de_ lpc4 |
| pcm_ Mag_ spectralEntropy_ sma_ de_ iqr1-3 |
| pcm_ Mag_ spectralSkewness_ sma_ de_ iqr2-3 |
| pcm_ Mag_ spectralKurtosis_ sma_ de_ iqr1-3 |
| mfcc_ sma_ de[3]_ iqr2-3 |
| pcm_ zcr_ sma_ meanPeakDist |
| pcm_ Mag_ spectralFlux_ sma_ meanPeakDist |
| pcm_ Mag_ spectralKurtosis_ sma_ peakMeanRel |
| mfcc_ sma[7]_ meanPeakDist |
| audspec_ lengthL1norm_ sma_ de_ flatness |
| audSpec_ Rfilt_ sma_ de[14]_ flatness |
| audSpec_ Rfilt_ sma_ de[19]_ flatness |
| audSpec_ Rfilt_ sma_ de[22]_ flatness |
| mfcc_ sma_ de[1]_ flatness |

Table A.10: *Features selected for Extraversion from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods SSCP, USCP and RSFS (Test set UAR 74.8 %).*

| **Extraversion** |
|---|
| pcm_ zcr_ sma_ segLenStddev |
| pcm_ RMSenergy_ sma_ de_ skewness |
| audSpec_ Rfilt_ sma[23]_ upleveltime75 |
| audSpec_ Rfilt_ sma[25]_ quartile3 |
| pcm_ Mag_ spectralFlux_ sma_ upleveltime25 |
| mfcc_ sma[1]_ quartile2 |
| mfcc_ sma[1]_ risetime |
| mfcc_ sma[7]_ iqr1-2 |
| mfcc_ sma[11]_ lpc1 |
| mfcc_ sma[12]_ lpc0 |
| pcm_ Mag_ spectralFlux_ sma_ de_ quartile3 |
| mfcc_ sma_ de[14]_ lpc2 |
| voicingFinalUnclipped_ sma_ de_ quartile2 |
| audspec_ lengthL1norm_ sma_ peakDistStddev |
| pcm_ Mag_ spectralVariance_ sma_ qregc3 |
| mfcc_ sma_ de[3]_ rqmean |

Table A.11: *Features selected for Agreeableness from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods USCP and RSFS (Test set UAR 58.4 %).*

| **Agreeableness** |
|---|
| audspec_ lengthL1norm_ sma_ de_ lpc4 |
| audSpec_ Rfilt_ sma[10]_ lpc4 |
| pcm_ Mag_ fband250-650_ sma_ iqr2-3 |
| pcm_ Mag_ spectralRollOff90.0_ sma_ iqr1-2 |
| mfcc_ sma[1]_ iqr1-3 |
| mfcc_ sma[1]_ lpgain |
| mfcc_ sma[3]_ range |
| mfcc_ sma[7]_ lpc0 |
| mfcc_ sma[8]_ lpc0 |
| mfcc_ sma[12]_ lpc3 |
| audSpec_ Rfilt_ sma_ de[3]_ meanSegLen |
| audSpec_ Rfilt_ sma_ de[3]_ segLenStddev |
| pcm_ Mag_ spectralKurtosis_ sma_ de_ lpc1 |
| mfcc_ sma_ de[2]_ maxSegLen |
| F0final_ sma_ quartile1 |
| F0final_ sma_ percentile1.0 |
| logHNR_ sma_ de_ range |
| audspec_ lengthL1norm_ sma_ meanFallingSlope |
| pcm_ zcr_ sma_ de_ flatness |
| pcm_ Mag_ spectralVariance_ sma_ de_ flatness |
| pcm_ Mag_ psySharpness_ sma_ de_ flatness |

Table A.12: *Features selected for Neuroticism from the Speaker Trait Challenge feature set (Schuller et al., 2012) by each of the subset selection methods SSCP and RSFS (Test set UAR 64.8 %).*

| **Neuroticism** |
|---|
| audspec_ lengthL1norm_ sma_ upleveltime90 |
| audspec_ lengthL1norm_ sma_ de_ iqr1-2 |
| pcm_ RMSenergy_ sma_ de_ quartile3 |
| pcm_ zcr_ sma_ de_ iqr1-3 |
| audSpec_ Rfilt_ sma[4]_ quartile3 |
| audSpec_ Rfilt_ sma[9]_ upleveltime25 |
| audSpec_ Rfilt_ sma[22]_ downleveltime25 |
| pcm_ Mag_ spectralRollOff25.0_ sma_ skewness |
| pcm_ Mag_ spectralFlux_ sma_ upleveltime50 |
| pcm_ Mag_ spectralSkewness_ sma_ quartile1 |
| pcm_ Mag_ spectralKurtosis_ sma_ quartile1 |
| mfcc_ sma[1]_ quartile3 |
| mfcc_ sma[6]_ pctlrange0-1 |
| mfcc_ sma[8]_ lpgain |
| mfcc_ sma[14]_ skewness |
| audSpec_ Rfilt_ sma_ de[11]_ upleveltime25 |
| pcm_ Mag_ fband1000-4000_ sma_ de_ percentile99.0 |
| mfcc_ sma_ de[1]_ percentile99.0 |
| mfcc_ sma_ de[12]_ quartile3 |
| mfcc_ sma_ de[12]_ iqr1-2 |
| mfcc_ sma_ de[13]_ quartile3 |
| mfcc_ sma_ de[14]_ lpgain |
| F0final_ sma_ quartile1 |
| F0final_ sma_ downleveltime25 |
| logHNR_ sma_ lpc0 |
| logHNR_ sma_ de_ risetime |
| audspec_ lengthL1norm_ sma_ meanFallingSlope |
| audspecRasta_ lengthL1norm_ sma_ peakRangeRel |
| pcm_ zcr_ sma_ meanPeakDist |
| pcm_ Mag_ spectralFlux_ sma_ amean |
| pcm_ Mag_ spectralFlux_ sma_ peakRangeAbs |
| pcm_ Mag_ spectralFlux_ sma_ linregc2 |
| pcm_ Mag_ spectralVariance_ sma_ peakMeanRel |
| pcm_ Mag_ psySharpness_ sma_ meanPeakDist |
| mfcc_ sma[12]_ qregerrQ |
| shimmerLocal_ sma_ amean |
| pcm_ zcr_ sma_ de_ flatness |
| pcm_ Mag_ psySharpness_ sma_ de_ flatness |
| mfcc_ sma_ de[12]_ rqmean |