Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech

Okko Räsänen[1], *Jouni Pohjalainen*[1]

[1] Department of Signal Processing and Acoustics, Aalto University, Finland

okko.rasanen@aalto.fi, jouni.pohjalainen@aalto.fi

Abstract

This work studies automatic recognition of paralinguistic properties of speech. The focus is on selection of the most useful acoustic features for three classification tasks: 1) recognition of autism spectrum developmental disorders from child speech, 2) classification of speech into different affective categories, and 3) recognizing the level of social conflict from speech. The feature selection is performed using a new variant of random subset sampling methods with *k*-nearest neighbors (kNN) as a classifier. The experiments show that the proposed system is able to learn a set of important features for each recognition task, clearly exceeding the performance of the same classifier using the original full feature set. However, some effects of overfitting the feature sets to finite data are also observed and discussed.

Index Terms: paralinguistics, feature selection, pattern recognition, speaker traits, random sampling, classification

1. Introduction

Non-linguistic (aka. *paralinguistic*) aspects of speech are an important component in everyday communication. They provide complementary information to the literal verbal message, indicating, e.g., identity and emotional state of the talker. They also serve turn-taking behavior in dialogue and allow the placement of emphasis on different aspects of the spoken message. In addition, paralinguistic characteristics may serve clinical diagnostic purposes since they are often affected in pathologies and disorders related to the physiology of the mouth, throat, and lung areas, neural motor control, and speech-specific or generic cognitive development (e.g., [1-3]).

Given the above, it is evident that the automatic analysis of paralinguistic characteristics of speech can be beneficial in various applications. In order to build such automated systems, it would be useful to find a set of descriptive acoustic features for each paralinguistic property of interest. The features can be obtained by either using domain expertise in the task at hand, e.g., by knowing how the affective state of the speaker is reflected in the parameters describing the speech production system (e.g., [4]), or by computing a huge number of different acoustic descriptors from the data and then using feature selection techniques to find those features that are actually relevant to the current problem. Note that the conventional ASR features such as short-term MFCCs may not necessarily be optimal for paralinguistic analyses since they are mainly tuned to the time-frequency constraints imposed by the linguistic units of speech. In contrast, the aim of the paralinguistic analysis is to characterize aspects of speech that are independent of the spoken linguistic content and may occur at different time-scales.

In order to follow the feature selection strategy, this paper presents a novel variant of so-called random subsampling feature selection algorithms and studies its use on the selection of the most useful features from a huge pool of potential features. Unlike greedy heuristic algorithms, the proposed algorithm cannot converge to local minima in the objective function and provides a set of features that are statistically significantly better than any feature that is not beneficial to the classification performance. However, the algorithm is still susceptible to overfitting similarly to the majority of other approaches. Despite this, we show that the algorithm learns a good subset of features for a number of different paralinguistic classification tasks, demonstrating a reasonable performance on independent test sets held out from the selection process.

This study is carried out in the context of *Interspeech'2013 Computational Paralinguistics Challenge* and all speech data and the respective baseline results are provided by the challenge organizers [1]. Before describing the approach and the experiments in detail, a short introduction to the problem of feature selection is first given.

1.1. Automatic feature selection for data classification

Selection of the most distinctive signal features for data classification is a central problem in many pattern recognition applications. Ideally, the best set of features contains only those features that provide complementary information regarding the data classes. Given such a set, addition of any new features should not improve the classification performance whereas removing any of the chosen features should cause the classification performance to degrade. In addition to improving classification accuracy, smaller sets of features are also faster to process, lead to simpler classifiers (with better generalization capability), and may allow better insight to the classification problem at hand when analyzed more closely [5,6].

As simple as it may sound, there are two fundamental problems in finding the optimal subset of features: 1) the total number of possible subsets grows exponentially with the total number of features, making exhaustive search for the best subset almost always computationally impossible, and 2) objective measurement of the performance of a feature set is not easy due to the risk of *overfitting*, a process where a chosen set of features may provide the best performance on training data but still generalize poorly to novel data unseen during the optimization process. This problem is emphasized on datasets where the total number of features is large in comparison to the total number of available training data points (see, e.g., [7]).

In order to solve the first problem, typical feature selection methods attempt to find a good set of features using different types of heuristic approaches such as incrementally including the next best feature to the current feature set as in forward selection [8], removing the worst feature as in backward selection, or recursively doing both in order to avoid nesting of features (e.g., [9]; see also [5]). Another possibility is to find a correlate measure of feature usefulness that can be used

to rank and filter away features according to their usefulness ("filter approaches") without iteratively testing different subsets on the actual classification problem [5].

As for the overfitting problem, a typical solution is to simply estimate the amount of overfitting by optimizing the set of features on a separate training (and development) set and then testing the quality of the chosen set of features on an independent test set (see, e.g., [6]). However, this only provides an estimate for the amount of overfitting in the feature selection, but it does not help in the actual selection of the best set of features. Further optimization on the test set naturally makes the approach again prone to overfitting, and a new independent test set is required. In practice, the risk of overfitting in the feature selection can be most easily reduced by using a more comprehensive dataset during the feature selection, or by using algorithms that can somehow avoid convergence to a "false" optimum that can be highly specific to the training data.

In the current work, we utilize a feature selection method that avoids the selection of a single best subset of features with best classification performance on the training data. Instead, the method evaluates the relevance of each individual feature in the context of many other feature combinations and finally selects all features whose usefulness exceeds that of an average feature.

2. Material and features

The material consists of three major classification problems: recognition of developmental disorders, speaker emotions, and level of conflict from speech. Each problem is further divided into sub-tasks, yielding a total of 7 sub-tasks.

The recognition of developmental disorders was performed on Child Pathological Speech Database (CPSD; [3]), that contains two sub-tasks: binary classification of speech into typical and atypical (disorder-related) classes (from now on, referred to as the TY task), and four-way classification into categories of control group, autism disorder, pervasive development disorder – not otherwise specified, and specific language impairment (diagnosis, or from now on: DI).

Emotion recognition was performed on Geneva Multimodal Emotion Portrayals corpus (GEMEP; [10]) and contains two binary tasks: level of valence (VA) and arousal (AR), both graded as low or high, and a 12-way classification task with 12 emotion categories (EM).

Finally, the level of conflict was measured using the SSPNET Conflict Corpus [11] that contains two sub-tasks: a binary classification into low vs. high level of conflict (CL), and a regression task with a continuous target function: the level of verbal conflict in the range of [-10,10] (SC).

Each corpus was divided into training, development, and test sets as reported in [1]. Only the training and development sections were used for feature selection and classifier training. Instead of computing our own set of signal features, the corpora were accompanied with a readily available long-term features computed over each signal using the openSMILE feature extractor [12]. These features include a number of low-level descriptors (LLDs) such as energy, spectral and cepstral features, harmonicity and sharpness measures. In addition, a number of functionals have been computed from the LLDs, yielding a total set of 6373 features. In the current experiments, the full feature set was always used as a starting point for feature selection with the aim of finding a useful subset of the features for each task. A more detailed description of the feature set can be found in [1].

3. Methods

3.1. Pre-processing

Before further processing, all features were standardized to have a zero mean and unit standard deviation:

$$\hat{x}_i = (x_i - \mu_i)/\sigma_i \quad (1)$$

where x_i is the original value of feature i and μ_i and σ_i are the mean and standard deviation of the feature i values measured across the respective set (training, development, and test sets were normalized separately as this was found to improve performance).

3.2. Random subset feature selection

The basic goal of Random Subset Feature Selection (RSFS) is to find a subset of features that are beneficial in the given classification problem. These features are obtained by repetitively classifying the data with a kNN classifier while using randomly chosen subsets of all possible features and adjusting the *relevance* of each feature according to the classification performance of the subset that the feature participates in. Unlike greedy methods such as sequential forward selection [8] or their improved variants like sequential floating forward selection [9] where the gain of a feature is directly computed by including or excluding it from an existing set of features, each feature is evaluated in RSFS in terms of its average usefulness in the context of many other feature combinations. Therefore RSFS is also less prone to converge to a locally optimal solution dictated by the temporal order of selections.

In RSFS, the quality of the feature set gradually improves as more estimation iterations are performed. The algorithm is similar to the idea of random forests [13] and random kNN (RKNN; [7]) 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 the RKNN is that the final feature selection process in RSFS is based on a statistical comparison against random walk statistics. 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 [7]. In RSFS, the random subset classification is performed as many times as it is necessary in order to distinguish good features from features that simply appear useful due to the random components of the process.

In order to describe RSFS in more detail, we use the following notation: each *true* feature f_x from a full set of features F has a relevance value $r_x \in [-\infty,\infty]$ associated with it. In addition, a set of *dummy* features $d_y \in D$ with related relevancies g_y is also defined.

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

1) Randomly pick a subset S_i of n features f_x ($|S_i| = n$, $x \in [1,|F|]$) from the full set F by sampling from a uniform distribution.

2) Perform kNN classification on the given data set using S_i and measure the value of a desired criterion function c_i.

3) Update relevancies r_x of all used features f_x according to

$$r_X \leftarrow r_X + c_i - E\{c\} \quad (2)$$

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 arithmetic mean of c across all previous trials).

4) Repeat the process from 1) with a new random subset.

In parallel, a similar process is repeated 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. (2) but using the criterion function value of the true features. The dummy features are never used in the actual classification process but their relevance is still accumulated across trials. In this manner, the relevance g_y of any dummy feature d_y essentially becomes a random walk process with no correspondence to the actual classification performance. Therefore, the relevance of the dummy features provides a baseline level r_{rand} that should be exceeded by a true feature in order to be considered as useful in the task. Then the final goal is to find a "best" subset of features $B \subseteq F$ that satisfies

$$p(r_j > r_{\text{rand}}) \geq \delta, \quad \forall f_j \in B,F \tag{3}$$

where r_{rand} is the relevance of a non-useful feature and δ is a user set threshold for probability. The random baseline level r_{rand} is modeled as a normal distribution of the dummy relevancies g_y:

$$p(r_j > r_{\text{rand}}) = \frac{1}{\sigma_g \sqrt{2\pi}} \int_{-\infty}^{r_j} \exp(-(x-\mu_g)^2/(2\sigma_g^2))dx \tag{4}$$

where μ_g and σ_g are the mean and standard deviation of the dummy feature relevancies g_y across all dummy features.

In this study, the unweighted average recall was used as the criterion function c in Eq. (2) and the probability threshold was set to δ = 0.99. The number of features used in classification was set to $n = \text{round}(\sqrt{|F|}) = 80$ according to [7, 14]. In a similar vein, a total of 50 dummy features were created and the relevance of $m = \text{round}(\sqrt{|D|}) = 7$ random features were updated at each iteration. The sampling process was repeated for 300 000 iterations before selecting the final set of features according to Eq. (4). As for the kNN used as the criterion function, the number of neighbors k in the voting was always fixed to k = 2 (see [7]).

A variant of the RSFS was also tested in which the feature subsets were not sampled from a uniform distribution but from a distribution defined by the current relevance values. This approach leads to a faster convergence of the feature set, but due to the enormous size of the used feature space, this runs to a risk of neglecting some of important features that do not achieve sufficiently high relevance in the early phase of the learning. In practice, this approach achieved relatively good classification accuracies on the training data but the obtained feature sets had only 30-40% overlap between different runs of the feature selection algorithm. We also experimented on weighting each feature according to its relevance value during the kNN classification. However, together with the probabilistic sampling, the additional degrees of freedom associated with the weighting scheme led to significant problems with overfitting and to poor results on held-out data.

3.3. Classifiers

A standard kNN classifier was used for all tests except the task of predicting the continuous valued grade of conflict level (task SC). In SC, the grade of conflict was estimated as the mean of the conflict levels associated with k nearest data points. In standard tasks, the counts of different class labels in the neighborhood of k samples were also normalized according to the inverse of the relative frequency of the class in the training set in order to account for uneven distributions.

4. Experiments

4.1. Procedure and evaluation

Feature selection with the RSFS was always performed by using the samples in the training set as the basis for the kNN classifier and testing the accuracy on the development set. Subsequently, the optimal value of k (number of neighbors) in the kNN for the chosen feature set was searched using two different approaches: 1) Optimizing k for the best possible development set performance using the training set for training, referred to as *standard (S) condition*. 2) Optimizing k using combined training and development sets with 50-fold evaluation, referred to as *n-fold (N) condition*. In both cases, a search for the best performance was performed across all possible values of k. The motivation for the use of the two different approaches was to study the effects of over-fitting the value of k to the given data in these two conditions, ultimately estimated in terms of performance on the held-out test set.

The final values of k were scaled for test set evaluation according to the relative sizes of the optimization training set and the combined training and development set, i.e.,

$$k_{\text{test}} = k_{\text{opt}} * (N_{\text{train}} + N_{\text{dev}})/N_{\text{opt}} \tag{5}$$

where k_{opt} is the optimal k in condition (S) or (N), N_{train} and N_{dev} are the number of samples in the training and development sets, respectively, and N_{opt} is the number of samples in the training set of the kNN classifier used in optimization (i.e., $N_{\text{opt}} = N_{\text{train}}$ for (S)). The idea of the scaling was to maintain the estimated sizes of the neighborhoods (in terms of Euclidean distance) despite the increased sample density in the combined training and development sets. Results for the development set and test set are reported using both approaches for the k optimization.

Performance was measured according to unweighted average recall (UAR) and weighted average recall (WAR), where UAR is simply the mean of classification accuracies of individual classes whereas WAR is the proportion of correctly classified samples divided by the total number of samples in the test set. In the continuous regression task SC, the Pearson correlation coefficient (CC) was computed between the estimated and true conflict levels. The reference baseline results are provided in [1] and were computed using linear kernel Support Vector Machines (SVMs).

4.2. Feature selection results

After running the RSFS for 300 000 iterations, the sizes of obtained feature sets were 430 (DI), 304 (TY), 757 (EM), 361 (VA), 162 (AR), and 349 (CL & SC) features, corresponding to an average of 6.2% of the original feature set size. Fig. 1 shows an example of the evolution of the relevance values for a number of features as a function of the iteration number (task AR). Also, the number of features does not fully stabilize at 300 000 iterations in all of the sub-tasks. However, the experiments on the development set revealed that the use of additional iterations beyond 300 000 did not have notable effect on the development set classification performance, suggesting that the most important features were already included at 300 000 iterations. Note that the time complexity of the RSFS is only a fraction of that of the forward selection where the number of computations increases exponentially with increasing size of the feature pool.

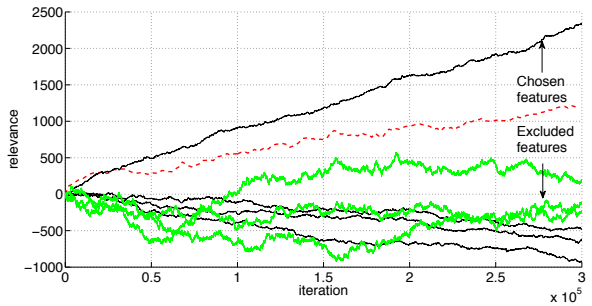

Figure 1: *An example of the development of the relevance values r_j as a function of iteration number for four true features (black lines) and three dummy features (noisy green lines). The relevance threshold for feature selection, computed from the dummy features, is denoted with red dashed line. Only the topmost true feature is chosen to the final feature set after all 300000 iterations.*

If the selection process was repeated several times, the proportion of the same features occurring in different runs was around 70-92% for different sub-tasks. The selected sets of features and their relative relevance values for each sub-challenge are available for download at http://www.acoustics.hut.fi/~orasanen/ComParE2013/.

4.3. Classification results

Table 1: *Classification accuracies for all sub-challenges. (B) denotes the baseline results reported in [1], (S) is the result obtained using a value of k optimized in a standard training/development set split, and (N) is the result using a value of k obtained in cross-validation using combined training and development set with 50 folds. UAR denotes unweighted average recall and WAR weighted average recall.*

		Devel (B)	Devel (S)	Devel (N)	Test (B)	Test (S)	Test (N)
DI	*UAR*	52.4	**63.7**	61.4	**67.1**	57.3	61.9
	WAR	N/A	**72.0**	70.1	N/A	73.1	**73.7**
TY	*UAR*	92.8	**93.2**	91.3	90.7	92	**92.2**
	WAR	N/A	**93.0**	90.4	N/A	91.3	**92.0**
EM	*UAR*	40.1	**42.8**	42.4	**40.9**	31.7	25.9
	WAR	N/A	40.3	**41.2**	N/A	**31.5**	25.6
VA	*UAR*	77.9	**85.6**	80.8	61.6	**64.4**	58.3
	WAR	N/A	**85.6**	80.8	N/A	**64.4**	58.4
AR	*UAR*	82.4	**88.3**	87.8	**75.0**	70.2	70.5
	WAR	82.2	**88.5**	88.0	N/A	70	70.2
CL	*UAR*	79.1	**85.6**	82.8	80.8	81.6	**83.9**
	WAR	N/A	**85.8**	83.3	N/A	82.1	**84.6**
SC	*CC*	0.8	**0.842**	N/A	**0.826**	**0.826**	**0.826**

Table 2: *The number of nearest neighbors k used on the test when estimated in (S) and (N) conditions and scaled according to Eq. (5).*

	DI	TY	EM	VA	AR	CL	SC
(S)	8	46	64	14	10	8	214
(N)	24	9	1	2	7	46	N/A

Table 1 shows the classification results for all sub-tasks and Table 2 shows the respective values of *k* used in the test set experiments. As can be seen from the table, the accuracy on the development set is above the baseline level for all sub-tasks. In comparison, the kNN accuracy without any feature selection (all 6373 features; (S) optimization) always led to UAR below the SVM baseline on the development set: 47.3% (DI), 88.0% (TY), 25.6% (EM), 70.2% (VA), 79.9% (AR), 77.7% (CL), and CC = 0.77 (SC).

As for the test set, improvement from the SVM baseline [1] is not obtained for all sub-tasks. Performance in some sub-tasks such as the speech typicality (TY), level of conflict (CL), and level of valence (VA) improves notably from the baseline, whereas the 12-class emotion recognition (EM) accuracy is notably below the baseline. The general trend seems to be that the current approach outperforms the baselines in binary classification tasks (except for the arousal level AR) whereas SVM with full feature set performs better on multiclass problems. The fact that the multi-class tasks have less training samples per each class suggests that overfitting of the feature sets or the values of *k* might have occurred.

Interestingly, the kNN performance in the social conflict (SC) task with a continuous target function is also comparable to the baseline performance despite the discrete nature of the kNN classifier. This suggests that the exemplar based discrete estimator for continuously valued target functions may be useful also in other applications where topology and sparsity of the sample space are problematic for continuous valued mapping functions and regression models.

As for the two different strategies for optimizing the value of *k*, it seems that neither standard training/development split (S) or cross-validation (N) performs unanimously better across all sub-tasks (see Table 2). Instead, it seems that the two tasks where (N) is notably worse are the tasks where the optimal *k* in (N) is very small in comparison to (S) condition. This again suggests that the emotion task has especially pronounced risk of overfitting to the training data, and this is emphasized if speech with identical linguistic content ends up in the training and testing data during a cross-validation procedure.

5. Conclusions

The results from the experiments in the recognition of paralinguistic characteristics of speech show that the RSFS algorithm is capable of selecting a useful subset of features from a large array of possible features, despite the fact that the number of training samples is smaller than the overall number of features. However, further work is needed in order to understand RSFS behavior in different types of feature selection tasks and in order to compare it against other standard feature selection approaches.

This study also shows that despite the relative simplicity of the kNN classifier, it can still achieve relatively good classification performance in paralinguistic recognition tasks when paired with a carefully chosen set of features, and achieves performance comparable to, or better than, the SVM-based baseline results reported in [1] (see [15] for similar findings).

6. Acknowledgements

This work was partially supported by Tivit program From Data to Intelligence (D2I) funded by Tekes.

7. References

[1] Schuller, B. Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Salamin, E., Polychroniou, A., Valente, F., and Kim, S., "The Interspeech 2013 Computational Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism", Proc. Interspeech 2013, ISCA, Lyon, France, 2013.

[2] Schuller, B., Steidl, S., Batliner, A., Nöth, E., Vinciarelli, A., Burkhardt, F., van Son, R., Weninger, F., Eyben, F., Bocklet, T., Mohammadi, G., and Weiss, B., "The Interspeech 2012 Speaker Trait Challenge", Proc. Interspeech'2012, ISCA, Portland, OR, 2012.

[3] Ringeval, F., Demouy, J., Szaszák, G., Chetouani, M., Robel, L., Xavier, J., Cohen, D., and Plaza, M., "Automatic Intonation Recognition for the Prosodic Assessment of Language-Impaired Children", IEEE Trans. Audio, Speech, and Language Processing, 19, 1328–1342, 2011.

[4] Airas, M., and Alku, P., "Emotion in Vowel Segments of Continuous Speech: Analysis of the Glottal Flow Using the Normalised Amplitude Quotient", Phonetica, 63, 26–46, 2006.

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

[6] Reunanen, J., "Overfitting in Making Comparisons Between Variable Selection Methods", Journal of Machine Learning Research, 3, 1371–1382, 2003.

[7] Li, S., Harner, J., and Adjeroh, D., "Random KNN feature selection – a fast and stable alternative to Random Forests", BMC Bioinformatics, 12:450, 2011.

[8] Whitney, A. W., "A direct method of nonparametric measurement selection", IEEE Trans. Computers, 20, 1100–1103, 1971.

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

[10] Bänziger, T., Mortillaro, M., and Scherer, K. R., "Introducing the Geneva Multimodal expression corpus for experimental research on emotion perception", Emotion, 12, 1161–1179, 2012.

[11] Kim, S., Filippone, M., Valente, F., and Vinciarelli, A., "Predicting the Conflict Level in Television Political Debates: an Approach Based on Crowdsourcing, Nonverbal Communication and Gaussian Processes", Proc. ACM International Conference on Multimedia, Nara, Japan, pp. 793–796, 2012.

[12] Eyben, F., Wöllmer, M., and Schuller, B., "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor", Proc. ACM Multimedia (MM'10), October 25–29, Firenze, Italy, pp. 1459–1462, 2010.

[13] Breiman, L., "Random Forests", Machine Learning, 45, 5–32, 2001.

[14] Li, S., "Random KNN Modeling and Variable Selection for High Dimensional Data", PhD thesis, West Virginia University, 2009.

[15] Pohjalainen, J., Kadioglu, S. & Räsänen, O., "Feature Selection for Speaker Traits", Proc. Interspeech'2012, Portland, Oregon, 2012.