INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Styrian dialect classification: comparing and fusing classifiers based on a feature selection using a genetic algorithm

Thomas Kisler, Raphael Winkelmann, Florian Schiel

Bavarian Archive for Speech Signals,
Institute of Phonetics and Speech Processing,
Ludwig Maximilian University Munich

{kisler, raphael, schiel}@phonetik.uni-muenchen.de

Abstract

Many classifiers struggle when confronted with a high dimensional feature space like in the data sets provided for the Interspeech ComParE challenge. This is because most features do not significantly contribute to the prediction. To alleviate this problem, we propose a feature selection based on a Genetic Algorithm (GA) that uses an SVM as the fitness function. We show that this yields a reduced subset (1) which results in an Unweighted Average Recall (UAR) that beats the challenge baseline on the development set for the 3-class classification problem. Further, we extract an additional per-phoneme feature set, where the features are inspired by the ComParE features. On this set the same GA-based feature selection is performed and the resulting set is used for training in isolation (2) and in combination with the aforementioned reduced challenge features (3). Five classifiers were tested on the three subsets, namely SVMs, DNNs, GBMs, RFs, and regularized regression. All classifiers achieved a UAR above the baseline on all three sets. The best performance on set (1) was achieved by an SVM using an RBF kernel and on sets (2) and (3) by a fusion of classifiers.

Index Terms: computational paralinguistics, ComParE challenge, genetic algorithm, SVM, DNN, lasso, GBM, RF

1. Introduction

The underlying assumption of dialect classification is that speech patterns systematically vary from region to region. This is usually attributed to early childhood influences such as a speakers' parents, peers and social environment [1].

Dialect classification based on acoustic features aims to identify these varying characteristics in the speech signal and to assign the correct dialect class to the speaker. Especially regarding English dialects, this area of research has received a lot of attention in the past (e.g. [2, 3, 4, 5]). For German dialects only a few studies on automatic dialect classification exist (e.g. [6, 7, 8]). [6] compared the performance of two systems, one that only applied acoustic features and a second that also incorporated word n-grams. They found that using n-grams improved the accuracy by a factor of 1.73. [7] used only acoustic information, but the method relied on a manually obtained orthographic transcription and a subsequent automatic phonetic alignment. [8] applied a similar method as [7], but aimed to estimate the speaker's geographical position instead of her/his dialect class. The ComParE 2019 dialect classification challenge is comparable to the above studies as it consists of a three-class dialect classification problem. In it, the participants are tasked with classifying three phonetically close East Bavarian dialects ([9]) recorded in the area of Styria, Austria.

In dialect classification a variety of features have been applied in the past. Examples are Mel-frequency cepstral coefficients (MFCCs; e.g. [3, 4, 5, 10]), signal energy (e.g. [4, 10]), Perceptual Linear Prediction coefficients (e.g. [4, 10, 11]), voicing probability (e.g. [12, 13]), and fundamental frequency (e.g. [10]). This is also true for the used classifiers, examples being Support Vector Machines (SVMs; e.g. [12, 14, 5, 8]), Deep Neural Networks (DNNs; e.g. [15]), Random Forests (RFs; e.g. [7]), Gradient Boosting Machines (GBMs; e.g. [16]), and decision trees (e.g. [12]).

For the challenge a vast number of features ($> 10k$) is provided by the organizers (including all, but is not limited to the aforementioned features used in previous studies). The key assumption for this (and previous) challenges is: a high number of features makes it likely that at least a few carry information regarding the task at hand. While this is a reasonable hypothesis, it is known that the performance of many classifiers decreases on data sets containing many features that do not contribute to the prediction (e.g. [17, 18, 19, 20]). It can safely be assumed that in the vast *Challenge Feature Set*, a number of features will not contribute to the dialectal classification and can, therefore, be considered as noise for the classification. To reduce the feature set and therefore alleviate this "noise" issue, a variety of feature selection algorithms have been proposed (see [17, 18] for good overviews). An algorithm that has been shown to provide a good selection of successful features (e.g. [20]) by mimicking an evolutionary process is *GA* (cf. Sec. 3.2).

2. Data & Features

2.1. Speech material

The speech signals provided for the challenge are snippets of recordings from the STYRIALECTS corpus [21]. The speech in this corpus was elicited by a questionnaire, a picture naming task, and free speech [21]. The majority of signals in the challenge data sets are single words or short sequences of a few words; signals have a duration between $300\,\mathrm{ms}$ and $1500\,\mathrm{ms}$, averaging at $855.35\,\mathrm{ms}$, and are labelled with their respective dialect groups: *EasternS*, *NorthernS*, or *UrbanS*.

2.2. Challenge Features

ComParE Features: The first official feature set ComParE is the standard set used for all challenges since Interspeech 2013 [22] and consists of 6373 features, which are the long-term functionals that cover the entire signal and are based on 141 low-level descriptors (LLDs) per frame including Δ, and $\Delta\Delta$ features [22, 23].

http://dx.doi.org/10.21437/Interspeech.2019-2540

BoAW Features: The second official feature set is an unsupervised representation, known as *bag-of-audio-words (BoAW)*. As the name suggest, these features are extracted in a similar way to bag-of-words features for text. Here they are calculated based on the same 65 LLDs as the ComParE feature set, and the histogram is based on a code book (for more information cf. [24, 23]). For feature extraction the open source toolkit openXBOW is used [25].

AUDEEP Features: The third official feature set is based on recurrent sequence to sequence autoencoders and is extracted using the AUDEEP toolkit [26]. These features are extracted unsupervised with recurrent neural networks based on a set of Mel-scaled spectrograms extracted from the raw speech signal (for more information cf. [26, 23]).

2.3. Additional Features in this Study

We extracted a set of additional features that potentially contain additional information for the classification. While the challenge features are extracted for the complete signal, our additional feature set relies on a phonetic segmentation and labeling of the signal. Extracting features on a per-phoneme-basis could be beneficial to the classification, as it is possible to model pronunciation differences in certain categories between regions.

The same feature set has been successfully applied to classify speakers of German-speaking areas (i.e., mostly Germany, Austria, and Switzerland) into broad categories (North/South; East/West) [7], and to continuously estimate a speaker's geographical position [8]. In both cases, however, an orthographic transcription existed and, therefore, a forced-alignment using WebMAUS [27] was possible.

As our extraction process relies on phoneme labels and segment boundaries and the challenge data sets does not contain orthographic transcript, we used the publicly available phoneme recognizer WebMINNI[1]. WebMINNI is a variant of the forced-alignment tool MAUS developed by the third author of this paper (for more information cf. [28, 27]).

The performance of WebMINNI varies considerably depending on the language, the speaker and the quality of the speech signal. However, [8] showed that even noisy phoneme classes can successfully contribute to distinguishing different dialect areas. Hence, we assumed that for the current scenario the performance of WebMINNI is sufficient to get a rough categorization of the speech material to unveil regional variation.

The features are extracted using openSMILE [29] with a window size of 25 ms and a step size of 10 ms. All per-frame feature vectors that span 20% of the phoneme midpoint ($\pm 10\%$) are averaged to form the final vector of the current phoneme. The same method was applied in [7, 8]. The features extracted are based on the ComParE feature set and include several LLDs (e.g., energy, pitch, MFCCs, chroma features, harmonics-to-noise ratio, jitter, shimmer, spectral band energies, formants, linear predictive coding, line spectral pairs, intensity, and various spectral features such as slope, skewness, sharpness, etc.) and their Δs and $\Delta\Delta$s. Additionally, the current, the previous, and the following phoneme label plus the current phoneme's duration are also used as features. [8] gives a more detailed description of the features used.

Since multiple predictions (per phoneme) exist for each signal, the final prediction is made using a majority vote based on the prediction for each phoneme in a signal file. In case of ties the class with more training samples is chosen.

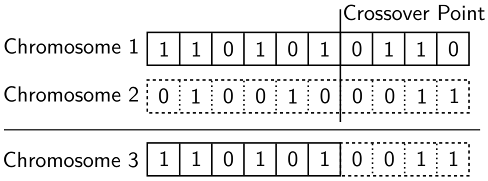

Figure 1: *Visualization of the crossover process in a GA in which two parent chromosomes (1 and 2) are combined to bread a new one (3).*

2.4. Applied Feature Sets

From the feature sets described above, three test sets were constructed. All feature sets were standardized based on the mean and standard deviation of the respective training set.

Challenge Features: The features provided, namely the ComParE, the BoAW (1000), and the AUDEEP (fused) features, were combined to form a set of 12,470 features with 9732 observations (5227 training, 2570 development, and 1935 test). This will be referred to as the *Challenge Feature Set*.

Additional Features: The second set as described in Sec. 2.3 consisted of 742 features (LLDs and short-time functionals Δ and $\Delta\Delta$s) and will be referred to as the *Additional Feature Set*. This feature set has 45,546 observations (25,705 training, 10,640 development, and 9201 test), this is more than the *Challenge Feature Set* as each signal file contains multiple phonemes and, therefore, multiple feature vectors.

It is worth noting that three files could not be processed, as the phoneme recognition detected only pauses/noise. These files got assigned the majority class during classification, making sure to not influence the prediction positively, as this leads to a UAR[2] of $0.3\overline{3}$.

Combined Feature Set: On both feature sets, *Challenge* and *Additional Feature Set*, an independent feature selection was performed as is explained in the Sec. 3.2, and the resulting subsets were combined to form a third feature set, consisting of presumably the most useful features from both individual sets. This combined feature set consisted of 429 features (cf. Sec. 4.1) and the same amount of observations as the additional set. This will be referred to as the *Combined Feature Set*.

3. Method

3.1. Overview

We propose improving classification performance by reducing the initially large feature sets by using a Genetic Algorithm (GA) for feature selection. On the resulting feature subsets five different classifiers, namely SVMs, DNNs, lasso, GBMs, and RFs were applied in this study. The performance of each classifier is reported individually and in a fusion step, where the 3-best majority vote (a vote based on the three best classifiers) predicted the final class label.

3.2. Genetic Algorithms

3.2.1. Overview

Genetic Algorithms (for a good overview cf. [20]) mimic an evolutionary process in order to 'select' the optimal set of features. To do so, each feature subset is described by a *chromosome* consisting of 0s and 1s (a 1 denotes a feature being part

[1] Link to the interface: `clarin.phonetik.uni-muenchen.de/BASWebServices/interface/WebMINNI`

[2] The UAR is the mean recall for all three classes (cf. [30]).

of the subset, a 0 that it is not). In each generation the fitness of all chromosomes is evaluated and the best ones are combined to find a presumably better subset. A common method for the combination is depicted in Fig. 1, where a randomly chosen position is used for combining the chromosomes. To increase the process' search space, random mutations can be specified.

The process of creating new sets and estimating the fitness is repeated for a number of iterations, until the search converges or a maximal number is reached. An option called *elitism* ensures that always the best feature subsets are used for the generation of new ones, by taking over a certain percentage of the best chromosomes of each generation into the next one.

A parameter called *0-to-1* ratio controls the number of features selected by the chromosomes. The higher the 0-to-1 ratio, the fewer features are added to the final set.

3.2.2. Settings

We used an adapted version of the R [31] package *genalg* [32], where the adaption enabled us to evaluate the subset performance in each generation in parallel.

Due to different amounts of features in the feature sets, parameters were different, except for the chance of mutation (0.01) and elitism (20%) for both sets. We set the population size to 150 for the *Challenge Feature Set*, 40 for the *Additional Feature Set*, the 0-to-1 ratio was set to 30 for the *Challenge Feature Set* – aiming at 300 to 500 features – and 5 for the *Additional Feature Set* – aiming at 100 to 200 features.

We used an SVM with a Radial Basis Function (RBF) kernel to evaluate the fitness of the candidate chromosomes (with standard setting for $C = 1$ and $\gamma = \frac{1}{p}$). We chose the SVM with a RBF kernel as it is able to learn arbitrary decision boundaries, has the benefit of not requiring random steps, and can be calculated in reasonable time on commodity hardware.

3.3. Classifiers

Support Vector Machine: To train the SVM we used the R packages e1071 [33]. SVMs are known to be sensitive to their hyperparameters, which when using an RBF kernel are C and γ. Therefore, we performed a hyperparameter tuning using a standard grid search for values for C and γ of $1 * 10^x$, where $x \in -3, -2, -1, 0, 1, 2, 3$, as well as $\gamma \in \frac{1}{p}$ (e1071 default).

lasso: To train the regularized regression with the least absolute shrinkage and selection operator (lasso) we used the package glmnet [34]. We performed a grid search for the regularization parameter for λ for 1^x, where x was varied between the values 1 and -4 and decremented in steps of size 0.1.

Random Forest: We used the R package ranger [35] for training the RF. As RFs are also known to be insensitive to their hyperparameters [36, 7], the standard setting of $\sqrt{p}$ for the number of features randomly considered at each split was kept. We decided to use 500 trees in the RF.

Gradient Boosting Machine: We used the R package gbm [37] to train the GBM. We trained 500 trees for the *Challenge* and the *Additional Feature Set*, and 700 trees for the *Combined Feature Set*. We kept the other parameters at their standard values. These are *shrinkage* (0.1), which controls the learning rate, and *bag.fraction* (0.5), which controls the random subsampling.

Deep Neural Network: We used the R package keras [38] to train the DNN. We tested different topologies, where the first layer always had the number of input features and the last 3 output neurons. The amount of hidden layers was varied between 2 and 5 and the number of neurons per hidden layer between 50, 100, 200, and 500. The dropout rate was 0.2, weights were randomly initialized and training stopped if the validation loss (categorical cross-entropy) did not decrease for 10 epochs.

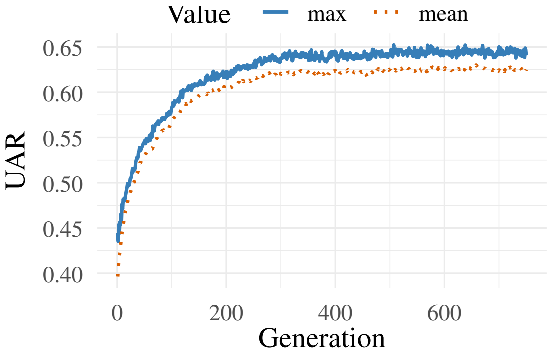

Figure 2: *Visualization of the development of the maximal UAR during the application of the GA on the Challenge Feature Set.*

Note: All classifiers were trained with class weights to account for the class imbalance in the data set. The weights were calculated as the inverse amount of training samples and then normalized so the majority class "UrbanS" had the weight 1.

4. Results

4.1. Feature Selection

4.1.1. Challenge Feature Set

The genetic algorithm was stopped after 750 iterations with 150 candidate feature sets in each iteration (which translates to roughly $110k$ trained models). The best result obtained was a UAR of 0.6523 selecting 323 features (cf. results in Tbl. 1). The development of the UAR over the generations can be seen in Fig. 2, showing the mean (orange, dotted) and the maximal UAR (blue, solid) in each generation. These 323 features are composed of the three subsets in the following way: 171 ComParE features, 65 BoAW, and 87 AUDEEP features. We will call the 323 features containing subset the *Challenge Feature Set*.

4.1.2. Additional Feature Set

Due to the time complexity of $O(m^3)$ of SVMs, where m is the number of training patterns (e.g. [39]), we decided to train the SVM only on 50% of the available 25,705 training samples during feature selection with the GA for this data set. These 50% are randomly sampled from the training patterns, which means they differ for all models during the evaluation process.

This random subsampling might influence the performance of the prediction on the Development set, as the complete data set is not available to the model. Therefore, according to the achieved UAR during feature selection, a SVM model was trained on the full subset and subsequently evaluated against the development set, for the 50 best subsets. The final result reported in Tbl. 1 is the best UAR achieved on those reduced feature subsets using all training samples. The best subset contained 149 features (110 Δ and $\Delta\Delta$) and achieved a UAR of 0.5609. Hence this subset is called *Additional Feature Set*.

4.2. Classification Results

4.2.1. Challenge Feature Set

The best performance on the *Challenge Feature Set* could be achieved by the SVM that was also used to select the features

($C = 1, \gamma = \frac{1}{p}$). This is not surprising, as the selected subset during the evolution of subsets is highly adapted to the algorithm assigning the fitness score and the used validation set. However, a moderately complex DNN (configuration neurons per layer: 323, 50, 50, 50, 3) also achieved a good UAR of 0.6200, which also holds true for the regularized linear regression with lasso with a UAR of 0.6149 ($\lambda = 0.00631$; for results cf. Tbl. 1). Especially the result of lasso on the reduced set – since it is a linear combination of the input features – can be taken as an indicator of a valid subset selection.

4.2.2. Additional Feature Set

The performance on the *Additional Feature Set* is worse than for that of the *Challenge Feature Set*. Here the best performance was achieved by a fusion of the 3-best individual classifiers yielding a UAR of 0.5713 (cf. Tbl. 1).

The three best classifiers were lasso ($\lambda = 0.0005102$, UAR $= 0.5482$), followed by the SVM ($C = 1$, $\gamma = 0.001$, UAR $= 0.5462$), and the DNN (topology 156, 50, 50, 50, 3, UAR $=$ 0.5123).

4.2.3. Combined Feature Set

The results of the individual classifiers on the *Combined Feature Set* fall behind the UAR of the SVM on the *Challenge Feature Set*. lasso ($\lambda = 0.003981$) achieved a UAR of 0.6489, the SVM ($C = 1$, $\gamma = \frac{1}{p}$) a UAR of 0.6404, and the DNN (topology: 440, 50, 50, 50, 3) a UAR of 0.6257.

However, the 3-best fusion of these classifiers achieved a UAR of 0.7055, which is better than the UAR of all other classifiers and fusions (cf. Tbl. 1).

Table 1: *UAR and accuracy resulting from various classifiers applied to the validation set based on a feature selection using GAs. Classifiers are ordered according to performance on the* Challenge Feature Set. *The best result in each subset based on UAR is highlighted in bold font.*

Set	Method	UAR Dev	Acc Dev	UAR Test
Challenge	**SVM**	**0.6523**	0.6689	0.3983
	DNN	0.6200	0.6584	-
	lasso	0.6149	0.6603	-
	GBM	0.5429	0.5673	-
	RF	0.4743	0.4716	-
	Fusion	0.6493	0.6778	-
Additional	SVM	0.5462	0.5609	-
	DNN	0.5123	0.4406	-
	lasso	0.5482	0.6442	-
	GBM	0.4898	0.5298	-
	RF	0.4766	0.5457	-
	Fusion	**0.5713**	0.5727	0.4200
Combined	SVM	0.6404	0.6839	-
	DNN	0.6257	0.6540	-
	lasso	0.6489	0.6765	-
	GBM	0.5303	0.5555	-
	RF	0.4947	0.5403	-
	Fusion	**0.7055**	0.7443	0.4129

5. Results on Test Set

To prove the generalization of the trained models, they were applied to an independent test set. The prediction of the model that resulted in the best UAR using the *Challenge Feature Set* (cf. 4.1) yields a UAR of 0.3983, which is much worse than on the development set. This could be due to the GA highly optimizing towards/overfitting the used classifier and validation set. A similar drop in performance can be seen using the *Additional Feature Set* (cf. 4.1), where the 3-best majority vote achieves a UAR of 0.42 and for the *Combined Feature Set*, on which the 3-best majority vote achieves a UAR of 0.4129. Hence, all results fall behind the model reported in the baseline paper trained on the $-50\,\mathrm{dB}$ AUDEEP feature set [23]. It is worth noting that the UAR of 0.47 is rather surprising and not in line with the results on similar sets. Except for this one result, the UAR achieved on all feature sets reported in the current study are better than in the baseline paper [23].

6. Conclusions

We have shown that the feature selection using a GA approach generally benefits various classifiers, as suggested in the literature (e.g. [20]). Using this approach we were able to achieve UARs on the development set that outperformed the UAR reported in the baseline paper.

One major drawback of any supervised feature selection, i.e. including the GA, is that the feature set is highly adapted to a) the validation set and b) the classifier used to evaluate the fitness/performance. This implies that the development set needs to closely approximate the data during the test and/or application phase. If this is not the case, the performance will suffer. This is likely to be the reason why the performance of our classifiers on the test set was significantly worse than for the development set.

To further improve the accuracy on unseen speakers, i.e. the generalization of the classifier, a n-fold Cross Validation (CV) could have been employed during the evaluation of the fitness that ensures different speakers in the test and validation set during the folds. However, other than the dialect labels, no speaker information was included in the challenge data set. Additionally, using a n-fold CV would have increased training time with the amount of folds.

A further observation is that the tree-based classifiers fall behind the other methods. This was to be expected, as decision boundaries in the feature space are less complex. However, even though the performance is not as good, they all also achieve a UAR that is better on the development set, than the ones in the challenge paper and for all but one on the test set [23].

The performance of the lasso classifier is also quite interesting. Being a regularized linear regression, it *linearly* combines the input features leading to a less complex decision boundary when compared with, e.g., DNNs or SVMs using an RBF kernel. Despite the linear combination of the input features lasso outperforms the SVM (used during feature selection) on the *Additional* and *Combined Feature Set*. This good performance might be attributed to the similarity of lasso and SVMs with linear kernels [40]. Albeit, being different to SVMs with RBF kernel, this indicates that the feature selection did generally produce good feature sets and could potentially be improved, e.g. with a more carefully designed CV during the fitness evaluation.

The fact that it was possible to enhance the *Additional Feature Set*, which is extracted per-phoneme, with features that are extracted over the whole signal file is worth noting here. By doing this, the UAR improves significantly in the 3-best classifier fusion and should be further investigated, for example in combination of the method in [7, 8] together with a proper n-fold CV.

7. References

[1] J. K. Chambers, "Dialect acquisition," *Language*, vol. 68, no. 4, pp. 673–705, 1992.

[2] M. Huckvale, "ACCDIST: a metric for comparing speakers' accents," in *Proc. Interspeech*, 2004, pp. 29–32.

[3] ——, "ACCDIST: an accent similarity metric for accent recognition and diagnosis," in *Speaker Classification II*. Springer, 2007, pp. 258–275.

[4] A. Hanani, M. Russell, and M. Carey, "Human and computer recognition of regional accents and ethnic groups from british english speech," *Computer Speech & Language*, vol. 27, no. 1, pp. 59–74, 2013.

[5] G. Brown, "Automatic recognition of geographically-proximate accents using content-controlled and content-mismatched data," in *Proc. ICPhS*, T. S. C. for ICPhS 2015, Ed., Glasgow, UK, 2015.

[6] M. Stadtschnitzer, C. Schmidt, and D. Stein, "Towards a localised German automatic speech recognition," in *Proc. of Speech Communication; 11. ITG Symposium*. VDE, 2014, pp. 1–3.

[7] T. Kisler and F. Schiel, "Towards a speaker localization from spontaneous speech: North-south classification for speakers of contemporary German," in *Elektronische Sprachsignalverarbeitung (ESSV) 2018 - Tagungsband der 29. Konferenz*, vol. 29. Ulm: TUDpress, 2018, pp. 200–207.

[8] T. Kisler, "Methods for large-scale data analyses of regional language variation based on speech acoustics," Ph.D. dissertation, Ludwig Maximilian University Munich, accepted.

[9] P. Wiesinger, "The central and southern Bavarian dialects in Bavaria and Austria," *The dialects of modern German: A linguistic survey*, pp. 438–519, 1990.

[10] S. Sinha, A. Jain, and S. Agrawal, "Acoustic-phonetic feature based dialect identification in Hindi speech." *International Journal on Smart Sensing & Intelligent Systems*, vol. 8, no. 1, 2015.

[11] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," Ph.D. dissertation, Columbia University, 2011.

[12] C. Woehrling, P. B. de Mareüil, and M. Adda-Decker, "Linguistically-motivated automatic classification of regional French varieties." in *Proc. Interspeech*, 2009, pp. 2183–2186.

[13] S. Finkelstein, A. Ogan, C. Vaughn, and J. Cassell, "Alex: A virtual peer that identifies student dialect," in *Proc. Culturally-aware Technology Enhanced Learning in conjuction with EC-TEL*, 2013.

[14] F. Biadsy, J. Hirschberg, and M. Collins, "Dialect Recognition using Phone-GMM-Supervector-based SVM Kernel," in *Proc. Interspeech*, 2010.

[15] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic Language Identification using Deep Neural Networks," in *Proc. ICASSP*. IEEE, 2014, pp. 5337–5341.

[16] N. B. Chittaragi and S. G. Koolagudi, "Acoustic features based word level dialect classification using svm and ensemble methods," in *Proc. IC3*, Aug 2017, pp. 1–6.

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[18] J. Tang, S. Alelyani, and H. Liu, *Feature selection for classification: A review*. CRC Press, 1 2014, pp. 37–64.

[19] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Advances in neural information processing systems*, Jan. 2000, pp. 668–674.

[20] J. Yang and V. Honavar, *Feature Subset Selection Using a Genetic Algorithm*. Boston, MA: Springer US, 1998, pp. 117–136.

[21] R. Vollmann. (2015) STYRIALECTS Projekt zur dialektgeografischen Untersuchung der Steiermark. Last accessed: 2019-03-08. [Online]. Available: https://homepage.uni-graz.at/de/ralf.vollmann/styrialects-styrian-dialect-research/

[22] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "Paralinguistics in speech and languagestate-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.

[23] B. W. Schuller, A. Batliner, C. Bergler, F. B. Pokorny, J. Krajewski, M. Cychosz, R. Vollmann, S.-D. Roelen, S. Schnieder, E. Bergelson, A. Cristia, A. Seidl, A. Warlaumont, L. Yankowitz, E. Nöth, S. Amiripian, S. Hantke, and M. Schmitt, "The INTERSPEECH 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds & orca activity." in *Proc. Interspeech*, 2019, pp. –.

[24] M. Schmitt, C. Janott, V. Pandit, K. Qian, C. Heiser, W. Hemmert, and B. Schuller, "A bag-of-audio-words approach for snore sounds' excitation localisation," in *Speech Communication; 12. ITG Symposium*, Oct. 2016, pp. 1–5.

[25] M. Schmitt and B. Schuller, "OpenXBOW: Introducing the Passau open-source crossmodal bag-of-words toolkit," *The Journal of Machine Learning Research*, vol. 18, no. 96, pp. 1–5, Oct. 2017.

[26] S. Amiriparian, M. Freitag, N. Cummins, and B. Schuller, "Sequence to sequence autoencoders for unsupervised representation learning from audio," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 17–21.

[27] T. Kisler, U. Reichel, and F. Schiel, "Multilingual processing of speech via web services," *Computer Speech & Language*, 2017.

[28] F. Schiel, "Automatic Phonetic Transcription of Non-Prompted Speech," in *Proc. ICPhS*, San Francisco, Aug. 1999, pp. 607–610.

[29] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE: the Munich versatile and fast open-source audio feature extractor," in *Proc. ACMMM*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1459–1462.

[30] B. Schuller, A. Batliner, S. Steidl, and D. Seppi, "Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge," *Speech Communication*, vol. 53, no. 9-10, pp. 1062–1087, 2011.

[31] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2018.

[32] E. Willighagen and M. Ballings, *genalg: R Based Genetic Algorithm*, 2015, r package version 0.2.0.

[33] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2018, r package version 1.7-0.

[34] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[35] M. N. Wright and A. Ziegler, "ranger: A fast implementation of random forests for high dimensional data in C++ and R," *arXiv preprint arXiv:1508.04409*, 2015.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] B. Greenwell, B. Boehmke, J. Cunningham, and G. Developers, *gbm: Generalized Boosted Regression Models*, 2019, r package version 2.1.5.

[38] J. Allaire and F. Chollet, *keras: R Interface to 'Keras'*, 2018, r package version 2.2.4.

[39] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 363–392, 2005.

[40] M. Jaggi, "An equivalence between the lasso and support vector machines," *Regularization, optimization, kernels, and support vector machines*, pp. 1–26, 2013.