# Strategies for addressing collinearity in multivariate linguistic data

Fabian Tomaschek, Peter Hendrix, and R. Harald Baayen
Department of General Linguistics, University of Tübingen, Germany

July 18, 2018

## Abstract

When multiple correlated predictors are considered jointly in regression modeling, estimated coefficients may assume counterintuitive and theoretically uninterpretable values. We survey several statistical methods that implement strategies for the analysis of collinear data: regression with regularization (the elastic net), supervised component generalized linear regression, and random forests. Methods are illustrated for a data set with a wide range of predictors for segment duration in a German speech corpus. Results broadly converge, but each method has its own strengths and weaknesses. Jointly, they provide the analyst with somewhat different but complementary perspectives on the structure of collinear data.

**keywords**: elastic net, supervised component generalized linear regression, random forests, collinearity, concurvity, segment duration.

## 1. Introduction

Response measures in linguistics and phonetics are often a function not of a single predictor but of many predictors jointly, reflecting a move away from mono-causal to multifactorial explanations. For instance, reductions and deletions in speech have been shown to correlate with a range of measures which include frequencies of occurrence and conditional probabilities at word and segment level (among others Jurafsky et al., 2000; Aylett and Turk, 2004; Gahl, 2008; Bell et al., 2009; Tremblay and Tucker, 2011; Priva, 2015). For example, Tremblay and Tucker (2011) used no less than 18 such measures to predict the durations of four-word sequences. Typically, many of the covariates included in these analyses serve as controls for potential confounds with predictors of central theoretical interest.

When predictors are completely uncorrelated and fully orthogonal, the results of a multivariable regression model and separate regressions with one predictor each will be virtually identical. Multiple regression comes into its own for data with non-orthogonal predictors. For such data, it serves as a mathematically principled arbiter for teasing apart relevant from irrelevant predictors. However, when predictors are strongly correlated, i.e., for collinear data, this arbitrage tends to result in counterintuitive and uninterpretable coefficients (Farrar and Glauber, 1967; Belsley et al., 1980). In this study, we review statistical methods that work around this problem.

When a data set is characterized by substantial collinearity, several problems arise. First, as already mentioned, parameter estimates may assume unexpected and theoretically uninterpretable values. Second, the model fit to the data will be unstable, in the sense that removal of just a few data points may have substantial consequences for the estimates of regression parameters. This holds both for linear regression and for the linear mixed model. Third, it can happen that no predictor on its own is significant, whereas all predictors jointly are successful in explaining a significant part of the variance in the response (Chatterjee et al., 2000).

In what follows, we begin with an introduction to the problem of collinearity[1] and

[1] In the context of nonlinear regression, collinearity also rears its ugly head in the form of concurvity. Concurvity can render models such as generalized additive (mixed) models unstable. We therefore briefly discuss how concurvity can be assessed, and what measures the analyst might consider when concurvity is high, in the appendix.

its adverse consequences for the magnitude and sign of estimated coefficients. We then describe a data set with substantial collinearity that will serve as the test case for our analyses. Subsequently, we introduce and illustrate three methods for analyzing collinear data. The first of these is a non-parametric technique from machine learning, random forests. Random forests enable the analyst to assess the relative importance of predictors. The second method is supervised component generalized linear regression (SCGLR). SCGLR performs dimensionality reduction on the predictor space, resulting in a smaller set of orthogonal predictors (the supervised components). SCGLR comes with visualization methods for inspecting how the original predictors load on the supervised components, and it provides regression coefficients for the original predictors that are properly shrunk. The third method that we discuss is the elastic net, a regularized regression technique that not only shrinks coefficients, but shrinks some of these completely to zero. This method therefore can be used to perform variable selection. For each method, we introduce the general concepts, and then illustrate its use for our example data set.

There is no fixed set of guidelines that guarantee the “correct” analysis of collinear data. George Box’s famous aphorism that all models are wrong but some are useful (Box, 1976) is especially relevant with respect to models for highly collinear data. The methods we review in the present study therefore provide the analyst with a toolkit that we find useful for exploring and understanding in complementary ways to what extent, and how a response might be shaped by a set of collinear predictors.

All analyses discussed in this study are documented step by step in the supplementary materials, to be downloaded from https://osf.io/5merb/. For these analyses, we made use of the statistical programming environment R (R Core Team, 2018) and specialist packages available for R (introduced below).

## 2. Suppression and enhancement

Suppression and enhancement occur in the linear regression model when two (or more) predictors for a given response $Y$ are strongly correlated. Take, for example, an analysis in which response times (dependent variable $Y$) in auditory lexical decision have to be predicted by word frequency counts in American English (predictor $A$) and British English (predictor $B$). Given that such frequency counts will tend to be strongly correlated, suppression and enhancement are likely to make the coefficients of the regression model uninterpretable. To understand why this happens, first consider the case in which we fit two one-predictor regression models to $Y$,

$$\begin{aligned} Y_i &= \beta_0 + \beta_A A_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma), &(1)\\ Y_i &= \beta_0 + \beta_B B_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma). &(2)\end{aligned}$$

where the $\beta_0$ represent the intercepts, $\beta_A$ and $\beta_B$ denote the coefficients for predictors $A$ and $B$, and $\epsilon$ is a Gaussian error term. When $A$ and $B$ are uncorrelated and completely orthogonal, the results of these two one-predictor models will almost completely identical to a multivariable regression model in $Y$ in predicted from $A$ and $B$ jointly:

$$Y_i = \beta_0 + \beta_A A_i + \beta_B B_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma). \tag{3}$$

In this case, the multivariable regression model has nothing to add about the effects of $A$ and $B$ that we did not already know from the two one-predictor analysis. However, when $A$ and $B$ are correlated, and not strictly orthogonal, then multiple regression comes into its own as the arbiter deciding which predictors should be given more (or less) weight. When predictors are only mildly correlated, there is little collinearity and the weights estimated by the multiple regression model (3) will make sense, but when strong collinearity is present, the resulting model will become theoretically uninterpretable.

Following Friedman and Wall (2005), we illustrate this phenomenon by varying the correlation between predictors $A$ and $B$, while keeping constant the correlations between $A$ and $Y$ as well as the correlations between $B$ and $Y$. We simulated multiple data sets with 1000 observations each, using the `mvrnorm` function from the **MASS** package (Venables and Ripley, 2002). $Y$, $A$ and $B$ are all standard normal random variables. We manipulated the correlation between $A$ and $B$ ($r_{AB}$) to range from close $-1$ to close to $+1$ in steps of 0.01. We fixed the correlation between $B$ and $Y$ at $r_{BY} = 0.3$, but considered three different correlations between $A$ and $Y$: $r_{AY} = -0.3$, $r_{AY} = 0.0$ and $r_{AY} = 0.6$. When $r_{AB} = 0$, $\beta_A$ is equal to $r_{AY}$ and $\beta_B = r_{BY}$.

Figure 1 illustrates the consequences of varying the correlation between $A$ and $B$ for the estimates of slopes $\beta_A$ and $\beta_B$ (top panels) and the corresponding $t$-values (bottom panels). Across all panels of Figure 1, dashed lines represent $\beta_A$ and solid lines $\beta_B$. The three values of $r_{AY}$ are listed above their respective panels.

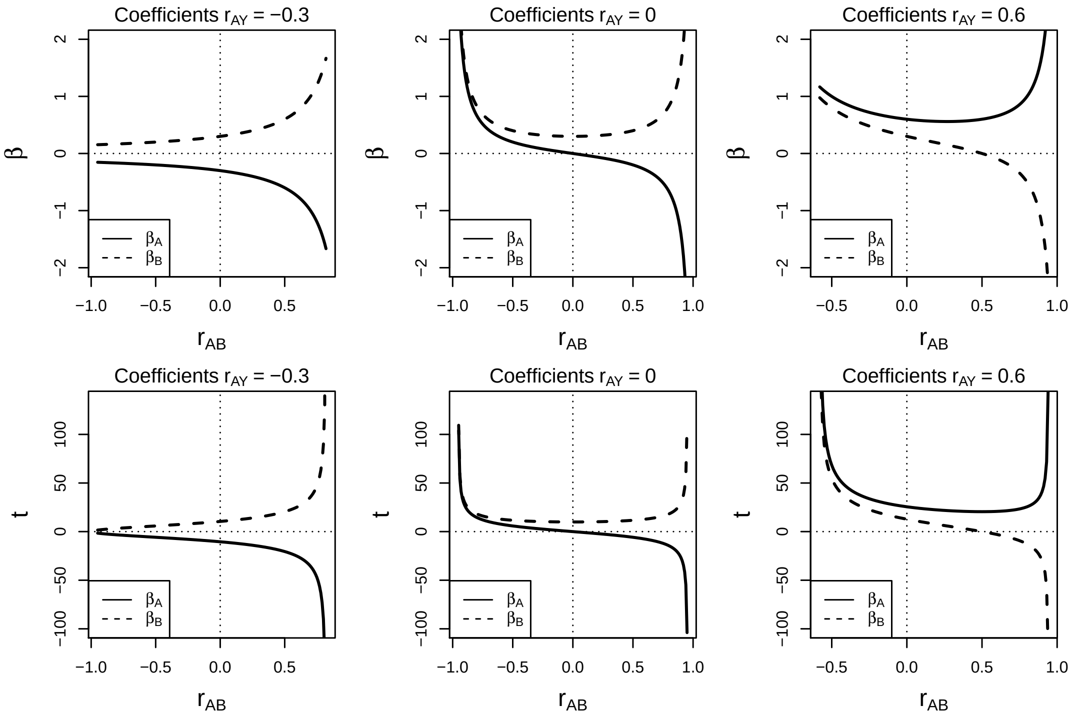

Figure 1: *$\beta$ coefficients (top row), and t-values (bottom row) as a function of $r_{AB}$, for $r_{BY} = 0.3$ and varying correlations $r_{AY}$ (left column: -0.3, middle column: 0, right column: 0.6).*

First consider the panels graphing coefficients against $r_{AB}$. When $r_{AB}$ is zero, $\beta_A$ is -0.3 when $r_{AY} = -0.3$, it is 0 when $r_{AY} = 0$, and it is 0.6 when $r_{AY} = 0.6$. As $r_{BY}$ is fixed at 0.3, $\beta_B$ is always 0.3 when $r_{AB} = 0$. When $r_{AB}$ moves away from zero, the coefficients change, and the more extreme $r_{AB}$ becomes, the more extreme the changes in the coefficients are. When $r_{AB}$ approximates 1, we find large positive and negative values for both $\beta_A$ and $\beta_B$. Which predictor receives a positive coefficient and which a negative depends on $r_{AB}$. When $r_{AB}$ is shifted towards $-1$, coefficients are not enhanced, but suppressed: both $\beta_A$ and $\beta_B$ assume smaller values than they have when $r_{AB} = 0$. It is noteworthy that $\beta_A$ is strongly enhanced even when $r_{AY} = 0$.

Estimates of the $t$-values associated with the coefficients also vary with $r_{AB}$ and can be very large for extreme positive values of $r_{AB}$. This leads to false positives for $\beta_A$ when $r_{AY} = 0$ and $r_{AB}$ is large. In other words, the model supports a significant effect of $A$ although there is in fact none. False negatives arise when $r_{AY} = -0.3$, $r_{BY} = 0.3$, and $r_{AB}$ is close to $-1$. In other words, the model does not support a significant effect of $A$ and $B$ although they are in fact significantly correlated with $Y$. In fact, strong collinearity can give rise to a model that succeeds in explaining variance of the predictor, without a single regressor being significant (see, e.g., Hadi, 1988; Chatterjee and Hadi, 2012b; Friedman and Wall, 2005, for examples).

Large coefficients with opposite sign for strongly correlated predictors are the hallmark of collinearity. In this case, the coefficients become difficult to interpret. For the above example of American and British frequency of occurrence, one frequency measure will reveal a coefficient with the expected negative sign, but the other frequency measure will emerge with a coefficient with an uninterpretable positive sign.

When strong collinearity is present, it is important to take a step back, and to address the question of how the artifacts of strong collinearity are best avoided. Before introducing possible strategies for addressing the adverse effects of collinearity, we first introduce the data set that we use to illustrate these strategies, the KIEL corpus.

## 3. Data set: word and segment durations in the KIEL corpus

The KIEL corpus (Kohler, 1996; Peters, 2003) comprises quasi-spontaneous speech as well as speech elicited by dictation. The corpus is annotated at the word level, the segment level, and the prosodic level. Annotations at the segmental level were manually corrected and contain indicators about missing canonical segments. Prosodic annotation provides information about primary and secondary stress in words. The entire corpus contains 32,460 word tokens (2,216 types), recorded from a total of 107 speakers.

From the KIEL corpus, we extracted durations for those vowels that occur in monosyllabic words and that were recorded in quasi-spontaneous speech. Of this set of vowels, we selected the first $10,000$ (from a total of 314 unique word types) for further analysis. The response variable of interest is vowel duration.

For each vowel, we registered `speaker`, carrier `word`, and `segment` identity, three random-effect factors. We recorded `stress` (levels `none, primary, secondary`), an indicator variable for whether the segment is located in a word at the end of a sentence, (`EndOfSentence`, with levels `true, false`), and phonological length of the vowel (`Vowellength`, with levels `long`, `short`).

In addition, we included `SpeakingRate` (number of syllables per second) and word duration (`wDur`). Following previous research (Jurafsky et al., 2000; Aylett and Turk, 2004; Bell et al., 2009; Tremblay and Tucker, 2011; Priva, 2015), we added 16 probabilities for segments and words from the frequencies of words and segments in the KIEL corpus. In what follows, we use `W` to denote words, `S` to denote segments, `target` for the current unit (`W` or `S`), and `prev` and `next` to denote preceding and following units. The probabilities we considered are: the probability (relative frequency) of the preceding, current, and following unit: `P(Wtarget)`, `P(Wnext)`, `P(Wprev)`, `P(Starget)`, `P(Sprev)`, `P(Snext)`; the joint probability with the preceding, or following unit: `P(Wprev, Wtarget)`, `P(Wtarget, Wnext)`, `P(Sprev, Starget)`, `P(Starget, Snext)`;the joint probability with both the preceding and following unit: `P(Wprev, Wtarget, Wnext)`,`P(Wprev, Wtarget, Wnext)`; the conditional probability given the preceding unit: `P(Wtarget | Wprev)`, `P(Starget | Sprev)`; and the conditional probabilities given the following unit: `P(Wtarget | Wnext)`, `P(Starget | Snext)`.

To this set of continuous predictors we added a final set of covariates: phonological neighborhood density (`NHD`), the count of words identical to the target word except for one segment; the count of segments in a word (`nSegperWord`); the number of speakers using a word (`Dispersion`) (see Adelman et al., 2006; Keuleers et al., 2015, for lexical dispersion across texts and speakers). In recent years, more and more researchers use measures derived from cognitive and neural networks to predict human behavior in cognitive tasks. These measures, such as activation estimated with naive discriminative learning (Baayen et al., 2011, 2016; Milin et al., 2017), are correlated with frequency measures to various degrees. To increase the number of potentially correlated predictors, we added the activation of the word given a word's diphones (`WordActivation SmallWindow`), and the activation of the word as provided by all diphones that occur in a five-word window around the target word (see Tomaschek et al., 2018, for further discussion). Larger activations are expected to be associated with shorter durations. The total number of numeric predictors thus amounts to 24. There are potentially other collinear predictors due to the nature of how they were created. For example, conditional probabilities are derived from frequencies of occurrence, which is why they are collinear by design.

Before analysis, we transformed numeric variables where necessary. As indicated by a Box-Cox test, the response variable was transformed by taking its square root. Several predictors were subjected to either a logarithmic transform, or to the root transform, depending on which transformation succeeded in rendering the distribution of values more symmetrical and with fewer outliers. For discussion of why transformations of response and predictor variables are necessary in the context of linear regression, see Zuur et al.

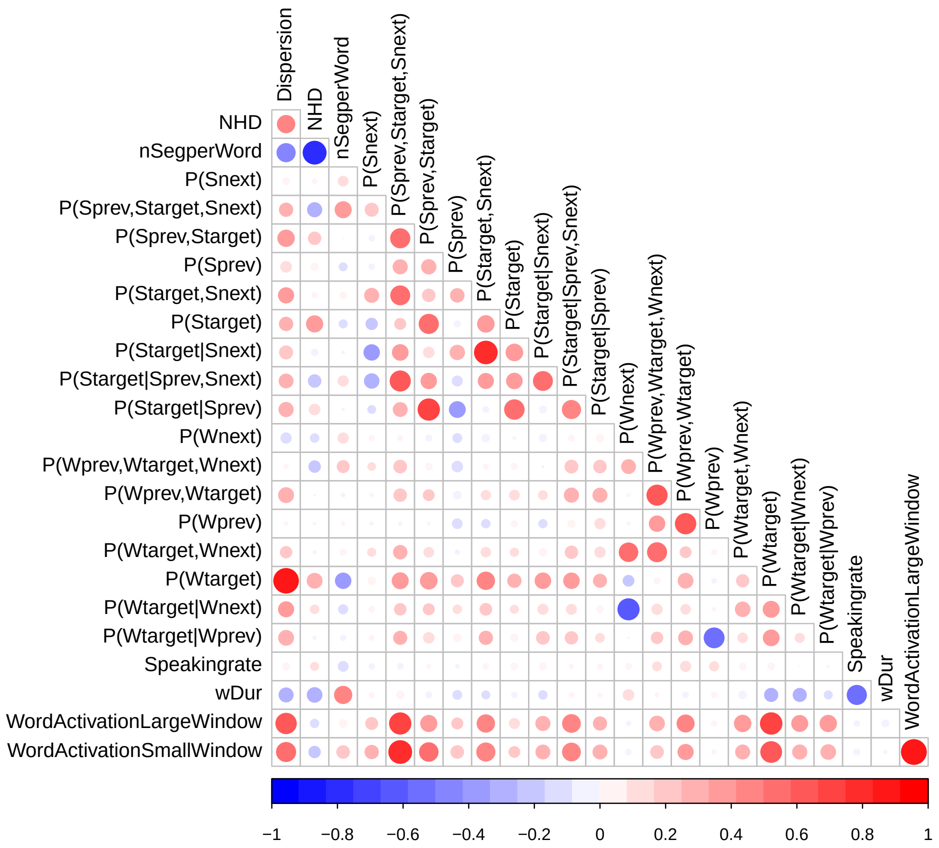

Figure 2: *Correlation map for numeric predictors in the KIEL corpus.*

(2010); Chatterjee and Hadi (2012a); Sheather (2009).

## 4. Diagnostics for collinearity

### *4.1. Correlation plot*

When a linear model is fit to the segment durations, a first indication of trouble is that there are predictors for which the coefficients are not estimated. Furthermore, which predictors are inestimable depends on the order of the predictors in the model formula.

As a first step towards a diagnosis of what is wrong, we inspect the correlations between the predictors, using a correlation map (obtained with the **corrplot** package (Wei et al., 2017)). In Figure 2, red dots represent positive correlations, whereas blue dots represent negative correlations. The size of the dots is proportional to the magnitude of the correlation. It is clear that many predictors are correlated to some extent. There are especially large correlations for `WordActivation SmallWindow` and `P(Sprev, Starget, Snext)`: $r = 0.77$, `Dispersion` and `P(Wtarget)`: $r = 0.90$, `nSegperWord` and `NHD`: $r =$

-0.78, `WordActivation SmallWindow` and `WordActivation LargeWindow`: $r = 0.91$, and `P(Starget | Snext)` and `P(Starget, Snext)`: $r = 0.76$. The problem with the correlation matrix as a diagnostic for collinearity is that although high correlations indeed point to a potential collinearity problem, the absence of high correlations does not guarantee that there is no problem (see Belsley et al., 1980, p. 92–93 for further discussion).

### *4.2. Variance inflation factors*

A better diagnostic for assessing whether coefficients are poorly estimated due to collinearity are the variance inflation factors (VIF) for the coefficients. The variance $\mathrm{VAR}[\hat{\beta}_j]$ of an estimated coefficient $\hat{\beta}_j$ for predictor $j$ is

$$\mathrm{VAR}[\hat{\beta}_j] = \frac{1}{1 - R_j^2} \cdot \frac{\sigma^2}{(n-1)S_j^2}, \tag{4}$$

where $S_j$ denotes the standard deviation of predictor $j$, $n$ is the number of data points, $\sigma^2$ the common variance of the errors, and $R_j^2$ the value of $R^2$ obtained from regressing the $j$-th predictor on all other remaining predictors. When predictor $j$ is highly dependent on one or more other predictors, $R_j^2$ will be large, and as a consequence $1/(1 - R_j^2)$ will be large as well. If predictor $j$ is orthogonal to the other predictors, $R_j^2$ is close to zero, and $1/(1-R_j^2)$ close to 1. The ratio $1/(1-R_j^2)$ is called the $j$-th variance inflation factor. One rule of thumb is that coefficients with a variance inflation factor exceeding five are poorly estimated and untrustworthy (Sheather, 2009). In R, variance inflation factors can be obtained with, e.g., the `vif()` function of the **car** package (Fox and Weisberg, 2011). When we try to apply `vif()` to the above-mentioned linear model, it reports that it cannot do so: not all coefficients in this model are estimable. When we refit the model with two troublesome predictors removed (e.g., `P(Wnext)`, `P(Wprev)`), we find that there are 13 predictors with variance inflation factors exceeding 5. For five of these, the variance inflation factor exceeds 10.

A problem with variance inflation factors is that it is not clear what a meaningful boundary is for a low versus a high value. For instance, Chatterjee and Hadi (2012b) state that values exceeding 10 are diagnostic of collinearity problems (p. 250), whereas (Sheather, 2009) puts the boundary at 5 (p. 203). For the present data, however, it is clear that there is a serious collinearity problem.

### *4.3. Condition number*

Whereas variance inflation factors are useful for finding individual predictors that clearly suffer from collinearity, the collinearity of the full set of predictors jointly is still not well assessed. This led Belsley et al. (1980) to propose a 'systemic' measure for collinearity, called the condition number $\kappa$. To understand what $\kappa$ actually assesses, we write out the estimates of the coefficients as a function of the model matrix $\boldsymbol{X}$ (the matrix with the predictors and a column of ones for the intercept) and the observed values of the response $\boldsymbol{y}$:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

The Achilles heel of the linear model is calculating the inverse of the (square) matrix $\boldsymbol{X}^T\boldsymbol{X}$. (When all predictors are centered and scaled, $\boldsymbol{X}^T\boldsymbol{X}$ is the correlation matrix.) The inverse of a square matrix need not exist. It does not exist when there are columns (or rows) that are (weighted) combinations of each other. To ascertain whether a matrix is singular, it can be decomposed into a product of three matrices, the middle matrix of which is zero except possibly for the elements on its main diagonal. These elements are known as eigenvalues. When a matrix is singular, at least one of these eigenvalues is zero. For empirical data, it is unlikely that predictors will be exactly (weighted) combinations of each other. Typically, predictors are themselves not exact but noisy.

Nevertheless, the more similar one or more empirical predictors are, the more $\boldsymbol{X}^T\boldsymbol{X}$ starts to resemble a singular matrix. This resemblance becomes stronger when one or more eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ are very close to zero. As we need the reciprocals of the eigenvalues to calculate the inverse matrix, it is clear that eigenvalues close to zero are going to give rise to huge reciprocals. Such huge reciprocals make the inverse matrix, and hence the estimates of the coefficients, unstable.

It turns out that the eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ are the squares of the so-called singular values of the design matrix $\boldsymbol{X}$ (the diagonal elements of the center matrix when $\boldsymbol{X}$ itself is decomposed into a product of three matrices). Therefore, very small singular values for $\boldsymbol{X}$ are also indicative of a collinearity problem. Belsley et al. (1980) show that the ratio of the largest and smallest singular values, the condition number $\kappa$, is the pivotal scaling factor for an upper bound for the effect of small changes in the response variable on the magnitude of the coefficients. Likewise, it provides such a scaling factor for small changes in the predictors. In other words, if $\kappa$ is large, very small differences in the response or predictor variables have huge consequences for the estimated regression coefficients.

For calculating $\kappa$, we start off with the matrix with predictors, we add a column of ones for the intercept, and then scale each column so that it has unit length. Without this scaling, the value of $\kappa$ would depend heavily on the measurement units of the variables, and as a consequence, it would become useless as a general diagnostic of collinearity. Belsley et al. (1980) point out that predictors should not be centered (see also Belsley, 1984, for detailed discussion): transformation of variables to Z-scores does not remove collinearity but makes it invisible. The singular values of the resulting matrix can be calculated, from which we obtain $\kappa$. All this is implemented in `collin.fnc()` from the **languageR** package (Baayen, 2008), which follows Belsley et al. (1980). (The `kappa` function of R does not include the intercept, and hence, even when its directive `exact` is set to `TRUE`, will give rise to different results.) Values of $\kappa$ exceeding 15 typically indicate that harmful effects of collinearity will be present. Values exceeding 30 point to strong collinearity for which corrective action is essential. These cutoff values are based on experience that has accumulated over the years in data analysis (Belsley et al., 1980; Chatterjee and Hadi, 2012b). For the predictors in the KIEL data set, $\kappa$ is no less than $1,809,457,843,187,094$.

### *4.4. Inspecting the sign*

When in doubt about the severity of collinearity and potential adverse effects of enhancement, it may be useful to check whether the sign of a coefficient is in accordance with the sign of a simple correlation of the same predictor with the response. If there is indeed a change of sign, it is worth investigating whether corrective measures are required.

## 5. Strategies for addressing collinearity

### *5.1. Common sense strategies*

When the set of predictors includes a set of variables that are theoretically strongly related, it makes sense to include only one in the regression analysis. By way of example, frequency counts based on a range of corpora will show strong correlations. When the nature of these corpora and the corresponding consequences for word use are not of primary interest, selecting one frequency measure from the set will help bring down collinearity.

Instead of selecting one measure by hand, one could alternatively orthogonalize the available measures using, for instance, principal component analysis, and select the first principal component, or the first couple of principal components, as representative for the full set of measures. Principal component analysis is explained in more detail in Section 5.3.1. Baayen et al. (2006) used this approach for 10 strongly correlated measures of orthographic and phonological consistency. Below, we discuss a method, supervised component generalized linear regression, that carries out orthogonalization in a more principled way.

Sometimes it is possible to de-correlate two related predictors by selecting one predictor and including the ratio of the first and second predictor as a new predictor. For instance, Baayen et al. (2006) were interested in frequency of occurrence in spoken and written English, and included written English as one predictor, and the ratio of written to spoken English as second predictor. The new predictor, which gauges the extent to which a word is used more often in writing than in speech, is by far not as strongly correlated with written frequency as the original spoken frequency measure.

These common-sense strategies all share one disadvantage: a strong dependence on manual intervention. Although hand-crafting the set of predictors may be justified by domain knowledge, methods that minimize manual intervention are worth considering. We discuss three such methods below.

One strategy that is not recommended is to reduce collinearity through residualization (see, e.g. Tremblay and Tucker, 2011; Priva, 2015, for applications of this strategy). A predictor $A$ that is correlated with another predictor $B$ is not entered into the analysis directly. Instead, $A$ is regressed against predictor $B$, and the residuals of this regression ($A_{\text{residuals}}$) are then entered into the analysis as a predictor instead of $A$. Since $A_{\text{residuals}}$ is orthogonal to predictor $B$, this reduces collinearity.

York (2012) and Wurm and Fisicaro (2014), however, demonstrated that the statistical characteristics of $\beta_A$ and $\beta_{A_{\text{residuals}}}$ are identical. By contrast, unfortunately, residualization can lead to an exaggeration of the statistical importance of the non-residualized predictor $B$ or an overestimation of the importance of data in regions of enhancement, depending on magnitude and sign of the correlation between $A$ and $B$. As a consequence, residualization may strongly affect the results and the interpretation of a regression analysis. In what follows, we consider strategies for analysing collinear data that do not require removing or orthogonalizing predictors by hand.

### *5.2. Random forests*

Collinearity is a problem of the linear model and the way in which it estimates regression coefficients. One way in which one can avoid the problems that arise in the context of the linear model due to collinearity is to step away from the regression framework, and to use instead a non-parametric method from machine learning. In what follows, we discuss random forests, which make use of decision trees and recursive partitioning.

Conditional variable importance measures calculated in random forests take into account the correlations between predictors. One issue with conditional variable importances, however, is that they are heavy on resources. Furthermore, these measures tend to inflate variable importance scores for uncorrelated data (Nicodemus et al., 2010). For this reason, we decided to use the unconditional variable importances provided by the **ranger** package (Wright and Ziegler, 2017) for R[2].

Before discussing further details, we clarify the contexts in which this method is of use. When the aim of the analysis is a model with outstanding prediction accuracy, random forests are an excellent choice. Random forests, however, do not provide detailed insight in the effects of individual predictors and their interactions. What they do provide is an assessment of predictor importance. When interest resides primarily in the effects of individual predictors and their significances, random forests remain useful as a tool for exploratory data analysis, just like visualization.

#### *5.2.1. Recursive partitioning*

Random forests are based on decision trees, which use a set of binary rules to predict a response variable. The response variable in a decision tree can be categorical or numerical in nature. Recursive partitioning trees for categorical responses are known as classification trees, trees for numerical responses are referred to as regression trees. The dependent

[2] We are thankful to Bodo Winter to pointing us to the **ranger** package, which outperforms alternative R packages such as **party** (Hothorn et al., 2018a), **partykit** (Hothorn et al., 2018b), and **randomForest** (Breiman et al., 2018) in terms of computational efficiency.

variable in the current study is segment duration, which is numerical, and consequently the decision trees introduced here are regression trees. In the analyses that follow, we use the variables as transformed for regression modeling, but such transformations are not required for analyses based on random forests and decision trees.

Decision trees are built through a process that is commonly referred to as recursive partitioning. Recursive partitioning algorithms start with the full data set, which includes all observations. The algorithm starts off with finding the predictor and the predictor value that split the data into two groups in an optimal manner. A commonly used splitting criterion, also used in the random forest analyses below, is the reduction in uncertainty (i.e., the reduction in entropy, which is also referred to as the information gain) about the value of the response variable (e.g., Therneau et al., 2017). Splits are implemented for a predictor value that reduce the uncertainty about the response variable the most. For each of the two subsets of the data that result from the split this process is repeated. The process of implementing binary splits for a branch of the tree continues until a stopping criterion is reached that is based on the extent to which additional splits improve the quality of the model fit. The model fitting procedure is concluded when the stopping criterion has been reached for all branches of the decision tree.

An example of a recursive partitioning tree is shown in Figure 3, top. For ease of illustration, we limited this tree to a maximum depth (i.e., number of splits) of 2. The initial split is made on word duration (`wDur`), at a value of $-0.51$. Observations for which the (normalized) word duration is smaller than $-0.51$ are assigned to the left branch of the trees, whereas observations for which word duration is equal to or greater than $-0.51$ are assigned to the right branch of the tree. The second split in the left branch of the tree is based on the value of phonological neighborhood density (`NHD`), whereas the second split in the right branch of the tree is based on the number of segments in a word (`nSegPerWord`).

The colored boxes provide more information about the observations in a node. The top value in a colored box is the mean segment durations for the observations in the corresponding node, whereas the bottom value in a box provides the percentage of observations in the data set that fall under the corresponding node. Mean segment durations for the observations in the four terminal nodes, i.e. the nodes at the last layer of the tree, differ substantially, which demonstrates that the implemented splits were successful at

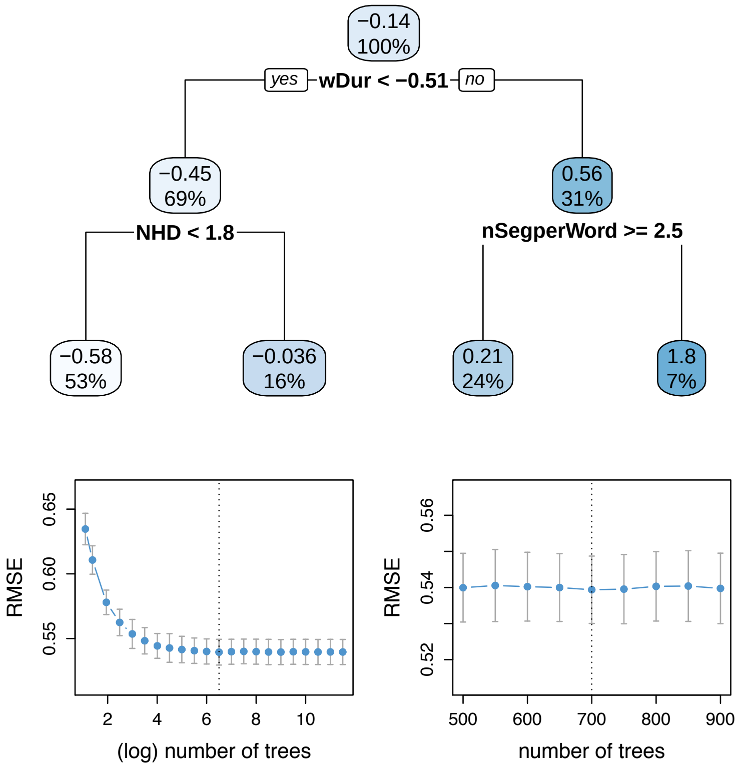

Figure 3: ***top:*** *Recursive partitioning tree fit to the segment durations in the KIEL corpus data. Colored boxes indicate mean predictor values and percentage of observations for the observations in each node.* ***bottom:*** *Results of the random forest models fit to the segment durations in the KIEL data. Optimal cross-validation performance for different numbers of trees on a coarse grid (left panel) and on a fine grid (right panel). The dashed lines indicates the number of trees for which the* MSE *is minimal.*

dividing the data into subsets with different segment durations.

Random forests (Breiman, 2001) fit not one, but multiple decision trees to the data. The idea behind random forests is to prevent overfitting by averaging over the predictions of a large number of trees. To make this idea work, it is crucial to ensure that the individual trees are not too similar. Simply fitting multiple decision trees to the complete data set would result in a series of identical trees. To overcome this problem, random forests combine two statistical concepts: bootstrap aggregating (bagging) and random predictor subset selection. Both of these techniques reduce the correlation between individual trees.

Bootstrap aggregating (bagging) is a method to artificially obtain more samples than the data can provide. The original data set acts as a pseudo-population. From this population, we take pseudo-samples that have the same size as the population and that are drawn from the population with replacement. As a result, a sample contains approximately two thirds of the observations in the population, whereas one third of the observations is left out. The observations that are in the sample are referred to as the in-bag observations, whereas the observations that are not in the sample are referred to as the out-of-bag observations. Each tree in a random forest is fit to a different bootstrapped sample.

The trees in random forest are not only fit to a subset of the observations. Also, each tree in a random forest is fit for a different subset of the predictor variables. For numerical dependent variables, a typical size of the subset of predictors that is considered for each tree is the number of predictors divided by 3 (Hastie et al., 2001; Strobl et al., 2009). The relatively small size of the subset of considered predictors ensures that the trees in a random forest are not too similar.

#### *5.2.2. Prediction and performance*

The prediction of a random forest model is defined as the average prediction of the individual trees for the out-of-bag observations (i.e. out-of-bag predictions). The performance of a random forest is evaluated by comparing the average of the out-of-bag predictions with the observed data. The average out-of-bag prediction has less variance and thus suffers less from overfitting when the predictions of individual trees are less correlated. Both bagging and random predictor subset selection ensure that the predictions of the individual trees in a random forest are not too similar. Unlike individual decision trees, random forests therefore tend not to overfit the data and have excellent generalization performance.

The interpretation of the results from a random forest are based on a measure of variable importance. Different measures of variable importance exist. The measure we use here is based on permutation tests (Breiman, 2001). To establish the importance of a predictor, the values for that predictor are randomly permuted. The accuracy of the out-of-bag predictions for the permuted predictor is then compared with the accuracy of the out-of-bag predictions for the original predictor. A predictor is regarded to be more important, the greater the difference in prediction accuracy between the original predictor and the permuted predictor.

### *5.2.3. Predictors and parameters for random forests*

The KIEL data set contains a number of categorical predictors. The **ranger** package is able to handle categorical variables, while the **glmnet** package (Friedman et al., 2018) that we will use below to illustrate regularized regression models is not. To be able to compare the variable importances of the random forest with the coefficients in regularized regression on a fair basis, we converted the categorical predictors in the data to numerical variables using *one-hot encoding* that converts the categorical predictors in the KIEL corpus to numerical variables.

To understand how one-hot encoding works, consider the categorical predictor `Stress`. `Stress` has three levels: `Primary`, `Secondary` and `None`. We encoded the information in the categorical variable `Stress` in two numerical predictors: `StressPrimary` and `StressSecondary`. `StressPrimary` was set to 1 for words with primary stress and to 0 otherwise. Similarly, `StressSecondary` was set to 1 for words with secondary stress and to 0 otherwise. The information for the third level of the categorical variable `Stress`, `None`, is implicitly encoded in `StressPrimary` and `StressSecondary`. Whenever both `StressPrimary` and `StressSecondary` are zero, the word has no stress. We applied one-hot encoding to all categorical predictors in the KIEL data set. (In linear regression modeling, R's default for categorical predictors, treatment coding, automatically adds such one-hot encoded predictors for factorial predictors to the model matrix.)

A crucial parameter in the `ranger()` function is `num.trees`, which determines the number of decision trees that should be fit. The **caret** package (Kuhn, 2018) for R provides grid search functionality for a large number of predictive models, which helps the user tune model parameters. To determine an appropriate value of `num.trees`, we fit a series of random forests with an increasing number of trees to the KIEL corpus data using the `train()` function of the **caret** package. We evaluated the prediction accuracy under (10-fold) cross-validation[3]

[3] Cross-validation is a technique to assess the accuracy of a model. The data is partitioned into a training set on the basis of which the model is fit and a test set on the basis of which the accuracy of the model is assessed. In 10-fold cross-validation, the model is trained on 90% of the data and tested on the remaining 10% of the data.

for each model with the root mean squared error (RMSE, which is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)}{n}} \tag{5}$$

where $\hat{y}_i$ and $y_i$ are the predicted and observed segment duration for observation $i$, respectively.

The RMSE for different numbers of trees is presented in Figure 3, bottom. The bottom left panel of Figure 3 shows the results of a coarse grid search, with the number of trees ranging from 1 to 11.5 on a log scale (i.e., from 3 to $98,716$ on a non-logged scale). The minimal RMSE in the coarse grid search was observed for a value of 6.5 on the log scale (665 trees, RMSE: 0.5395). We then carried out a second grid search, using numbers of trees near the optimal number of trees in the coarse grid search. The results of this fine grid search are presented in the bottom right panel of Figure 3. The minimal RMSE in the fine grid search was observed for 700 trees (RMSE: 0.5394).

It is worth noting that highly similar RMSEs were observed across a wide range of values of `num.trees`. A post-hoc analysis revealed that the RMSE for models with 23 or more trees were not significantly different from the optimal RMSE. Given the fact that random forests tend to not overfit the data, this is a typical pattern of result in a random forest analysis.

#### *5.2.4. Variable importance*

Following the results of the grid searches, we ran the final random forest with the `num.trees` parameter set to 700. The parameter for the number of predictors that are considered in each tree, `mtry`, was set to 10. Unscaled permutation-based variables importances were calculated by setting the value of the parameter `importance` to "permutation" (see Nicodemus et al., 2010, for a discussion of the benefits of unscaled variable importances). Default values were used for all other parameters. The RMSE for the out-of-bag predictions of the final model (0.5394) was nearly identical to the RMSE of the same model under cross-validation.

The variable importances for the random forest are presented in Figure 4. The variable with the highest variable importance is the duration of the word (*wDur*), unsurprisingly. The random forest model furthermore indicates that phonological neighborhood density (`NDH`) and the number of segments of the word (`nSegperWord`) are highly predictive of segment duration as well. To gain further insight into predictor effects, one can plot the recursive partitioning tree produced by the `rpart()` function (cf. Figure 3, top, but allowing for greater tree depth).

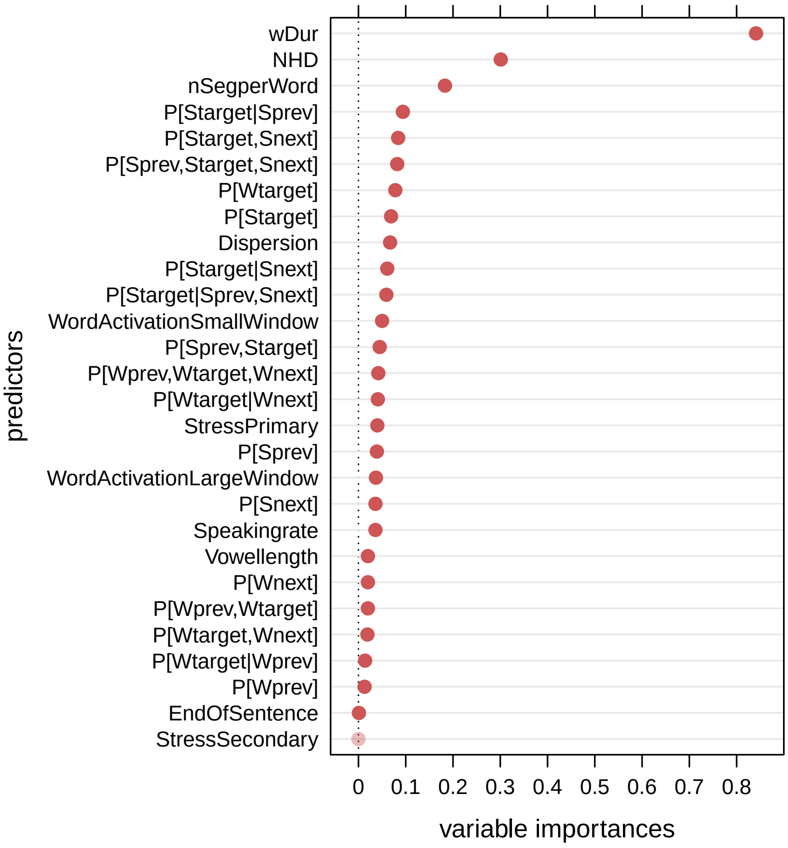

Figure 4: *Variable importances for the random forest model. Opaque red dots indicate non-zero variable importances, transparent red dots represent variable importances that are zero.*

Random forest variable importances provide an excellent assessment of the relative value of highly correlated predictors. The reason that a predictor gets a chance to show what it is worth, even though it is highly correlated with an even more powerful predictor, is that there are trees in the forest in which this more powerful predictor is not included among the set of predictors for that tree. In a standard recursive partitioning tree that considers all predictors, each split is based on the most powerful predictor available. In the forest of trees there are trees where the most powerful predictor is withheld, and hence the importance of less powerful predictors can be assessed, without the dangers of suppression or enhancement (see Strobl et al., 2009, for detailed discussion).

Random forests provide impressive prediction accuracy. Under cross-validation, a linear model fit to the segment durations with the 24 numerical covariates explains 50.37% of the variance in the durations. By contrast, a random forest based on the same set of predictors explains no less than 70.13% of the variance. As will become apparent below, none of the other methods for analyzing collinear data comes anywhere close to the prediction accuracy of the random forest. In the general discussion, we return to this finding, and discuss its possible theoretical implications.

### 5.3. Supervised component generalized linear regression

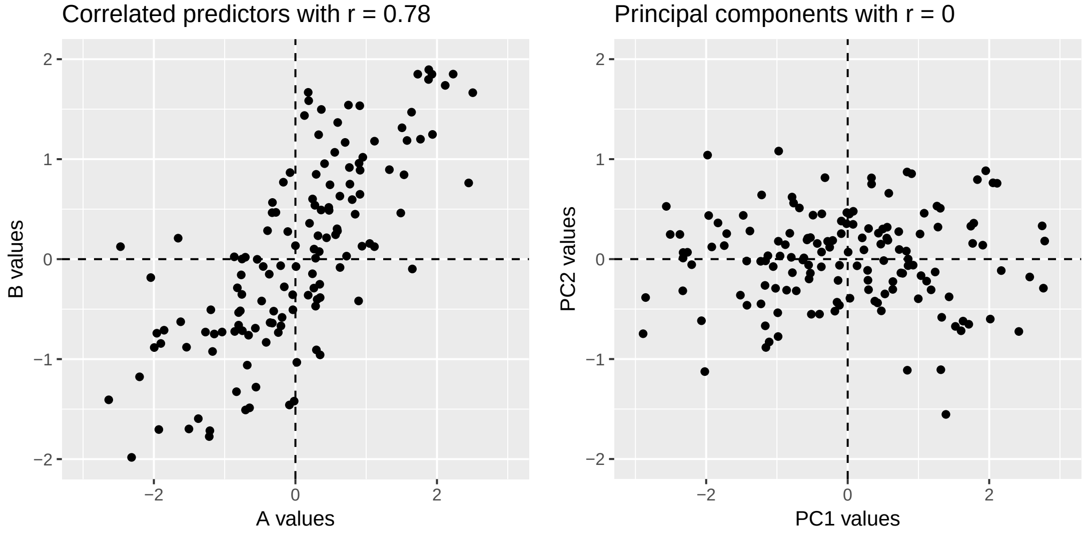

Figure 5: *S*imulated data with correlated predictors (left) and the corresponding orthogonalized predictors (right).

#### 5.3.1. Principal components regression and SCGLR

In multivariable regression with $k$ observations and $n$ predictors, an observation $i$ is a point in a $n$-dimensional space, whose $n$ axes are set up by the $n$ predictors. When all predictors are orthogonal, all axes are necessary to define the position of observation $i$ in this space. When predictors are correlated, there are empty regions in the $n$-dimensional space, and a smaller number of axes would suffice to properly locate each datapoint in a lower-dimensional space. The observations of collinear data sets are points in a space that, for all practical purposes, has a lower dimensionality than its number of predictors $n$.

Principal component analysis (Pearson, 1901) is a dimension reduction technique that finds new, orthogonal, axes for the data points, such that the first axis explains the highest proportion of the variance in the space of observations, the second axis explains the next highest proportion of the variance, and so on. For the case of $n = 2$, observations are points on a plane. For the case of $n = 3$, observations are points in a cube. If all points actually lie close to a line in the cube, the first principal component will be a new axis that will be close to all data points. Of the three principal components, the first will explain almost all of the variance. The second and third principal components are superfluous, explaining hardly any variance. Thus, a problem that at first sight appears to be a problem in a three-dimensional space has been reduced to a much simpler problem in a one-dimensional space. This is called dimensionality reduction.

Principal components regression is multiple regression that uses principal components derived from the original predictors as regressors. Crucially, not all principal components should be used, otherwise collinearity is back again on the doorstep (Belsley et al., 1980).

To make this more concrete, consider Figure 5. The scatter of points in the left panel indicates that predictors $A$ and $B$ are strongly correlated ($r = 0.78$). A principal component analysis rotates the data points anti-clockwise by approximately 130 degrees, resulting in the scatter in the right panel of Figure 5. Most of the variance in the data is now expressed along the horizontal axis, which represents the first principal component (PC1). The remaining variance is found on the vertical axis, which represents the second principal component (PC2). Both principal components are linear combinations of the original $A$ and $B$ axes. The extent to which the old axes are correlated with the new axes is proportional to the so-called loadings of the original variables on the principal components. Principal components are usually entered into a regression analysis simultaneously. As we have explained above, because they are orthogonal their coefficients will not differ from coefficients obtained in uni-variate models. Principal components regression can be performed using the **pls** package for R.

The goal of a principal components analysis is to reduce the dimensionality of the space in which the observations are points. A commonly used rule of thumb is that the first $m$ components that jointly capture 95% of the variance in the data are retained as new axes (predictors). In a principal components regression, therefore, the $c$ components that explain very small proportions of the variance are discarded, whereas $k - c$

orthogonal predictors are retained as predictors for the response, where $k$ is the original number of predictors. Once a linear model has been estimated for the $k - c$ principal components, the coefficients of the original predictors, given the dimension reduction, can be obtained. The magnitude of these coefficients will be substantially reduced compared to the estimates of a straightforward linear model, whenever the original predictors are substantially collinear.

Supervised component generalized linear regression (SCGLR, implemented in the **SCGLR** package, Bry et al. (2013)) builds on the concepts underlying principal component regression, but the mathematical implementation is substantially different. For the analyst, the important differences are the following.

First, SCGLR is designed such that multiple response variables (which can be any of Gaussian, binomial, and Poisson) can be modeled simultaneously. For the KIEL corpus, for instance, we could have included as further predictors the number of segment deletions or syllable durations, the idea being that the predictors for segment duration should also be relevant for understanding segment deletion and syllable duration. In the present survey, space restrictions limit demonstration of this aspect of SCGLR modeling to the supplementary materials.

Second, unlike standard principal components regression, SCGLR orthogonalizes not just the predictors, but the predictors and response variables jointly. Whereas principal components regression finds high variance directions in the covariate space, SCGLR sets out to find those directions in the space of the covariates that are optimal for predicting the response variables. Just as in principal components analysis, the components, now called supervised components, are estimated step by step. The first supervised component optimizes a trade-off between the variance it captures in the full variable space (predictors and responses) and the goodness of fit of that component as sole predictor of the response. The second component is selected in the same manner, with the restriction that it has to be orthogonal to the first component. This procedure is repeated until $K$ complementary and mutually independent components are obtained.

Third, whereas in principal components regression the number of principal components to retain is typically based on a rule of thumb, SCGLR implements a cross-validation procedure to determine the optimal number of supervised components.

Fourth, SCGLR allows for the possibility that there are predictors that do not need

Table 1: *Coefficients of supervised components and factorial predictors in the SCGLR model.*

| predictor | Estimate | Std. Error | t-value | p-value |
|---|---|---|---|---|
| Intercept | -0.321 | 0.019 | -16.657 | 0.000 |
| SC1 | 0.165 | 0.004 | 42.804 | 0.000 |
| SC2 | -0.221 | 0.005 | -44.963 | 0.000 |
| SC3 | -0.162 | 0.005 | -30.622 | 0.000 |
| SC4 | -0.096 | 0.005 | -19.055 | 0.000 |
| SC5 | -0.098 | 0.006 | -16.461 | 0.000 |
| SC6 | -0.011 | 0.006 | -1.784 | 0.074 |
| EndOfSentence | 0.122 | 0.045 | 2.741 | 0.006 |
| StressPrimary | 0.421 | 0.022 | 18.913 | 0.000 |
| StressSecondary | 0.226 | 0.428 | 0.528 | 0.598 |
| Vowellength | 0.099 | 0.021 | 4.796 | 0.000 |

to be orthogonalized. For the present data set, such predictors could be the sex and age of the speaker. Both sex and age are not expected to enter into strong correlations with the word and segment-bound predictors.

### *5.3.2. Working with SCGLR*

The steps in an SCGLR analysis are the following. First, the response variables are selected, and for each response variable, it is determined whether it is Gaussian, binomial, or Poisson. The single response variable of our working example, `sDur`, is a Gaussian response.

Second, the predictors are grouped into two sets. One set contains the collinear predictors that require orthogonalization, and the other predictors that are not orthogonalized. For the KIEL data set, the 24 variables laid out in section 3 are assigned to the first set. The second set comprises the factorial predictors `Stress` (`none`, `primary`, `secondary`), `EndOfSentence` (`true`, `false`) and `Vowellength` (`long`, `short`).

Next, the optimal number $K$ of supervised components needs to be determined. For this, the **SCGLR** package makes available the function `scglrCrossVal`, which requires the user to specify the maximum number of components to take into account. We set this value to 15. As the results of cross-validation may vary from run to run, we carried out the cross-validation procedure 8 times, and selected the best-supported value, which turned out to be 6.

Finally, the model itself is fit with the `scglr` function, with the parameter $K$ set to 6. The model object produced is a list with several components. Of these, the `gamma` component, provides a table of coefficients, together with their standard errors and associated statistics (see Table 1). The first five supervised components are all well supported as predictors for segment duration, and the same holds for the factorial predictors.

The summary of an `scglr` object generates two tables that are essential for the interpretation of the supervised components. The `rho` table lists the squared correlations ($r^2$) of the predictors with the supervised components. The `rho.pred` table provides the same information for the response variables. The information provided by these two tables is merged in Table 2. The first row of this table concerns the response. The greatest $r^2$ is observed for the second supervised component (SC2). The next largest $r^2$ is listed for SC1. Thus, in the 6-dimensional space spanned by the 6 SCs, the plane defined by SC2 and SC1 is the plane in which the response variable is most strongly represented. This plane is therefore listed in Table 2 as the 'best plane'. The 'best value' is the sum of the $r^2$ values for the axes of the best plane, and represents the variance in the response captured by the best plane. The remaining rows of Table 2 pertain to the predictors. Like the response, the `Dispersion` measure is most strongly expressed on the plane defined by SC1 and SC2, but for word duration (`wDur`) and speaking rate (`Speakingrate`) , the best plane is given by SC3 and SC4.

Interpretation of tables such as Table 2 is facilitated by visualization. The plot method implemented for `scglr` objects produces correlation plots, examples of which are presented in Figure 6. A correlation plot locates, by means of arrows, variables in the space defined by two (user-selected) supervised components. To avoid visual cluttering, a threshold (represented by a dashed circle) is set such that variables with a best value less than the threshold are not shown. The coordinates of a variable in the plane are the correlations $r$ (the square roots of the values listed in Table 2) of the variable with the pertinent supervised components. The length of a variable's arrow is, by Pythagoras' theorem, the square root of its best value. Its sign is taken from the correlation between a SC and the original predictor. In Figure 6, the arrows of predictors are presented in black, and that of the response in blue. The threshold was set at 0.5. Measures with best values (arrow lengths) less than 0.5, therefore, are not included in the plots.

The left panel of Figure 6 shows that on the SC1 by SC2 plane, neighborhood density

Table 2: *Squared correlations between predictors and supervised components.*

| predictor | SC1 | SC2 | SC3 | SC4 | SC5 | SC6 | best plane | best value |
|---|---|---|---|---|---|---|---|---|
| sDur | 0.330 | 0.361 | 0.199 | 0.059 | 0.051 | 0.001 | 1/2 | 0.690 |
| Dispersion | 0.35 | 0.40 | 0.01 | 0.00 | 0.06 | 0.00 | 1/2 | 0.751 |
| wDur | 0.12 | 0.01 | 0.43 | 0.22 | 0.02 | 0.01 | 3/4 | 0.656 |
| Speakingrate | 0.06 | 0.02 | 0.14 | 0.32 | 0.10 | 0.02 | 3/4 | 0.454 |
| WordActivation LargeWindow | 0.68 | 0.00 | 0.05 | 0.01 | 0.11 | 0.00 | 1/5 | 0.791 |
| P(Wnext) | 0.01 | 0.04 | 0.20 | 0.02 | 0.01 | 0.57 | 3/6 | 0.777 |
| NHD | 0.01 | 0.69 | 0.08 | 0.04 | 0.00 | 0.03 | 2/3 | 0.764 |
| P(Wtarget) | 0.49 | 0.26 | 0.01 | 0.00 | 0.05 | 0.00 | 1/2 | 0.751 |
| WordActivation SmallWindow | 0.66 | 0.00 | 0.08 | 0.04 | 0.08 | 0.01 | 1/5 | 0.743 |
| nSegperWord | 0.01 | 0.51 | 0.21 | 0.07 | 0.00 | 0.03 | 2/3 | 0.720 |
| P(Sprev, Starget, Snext) | 0.56 | 0.03 | 0.11 | 0.09 | 0.00 | 0.00 | 1/3 | 0.670 |
| P(Wprev) | 0.00 | 0.00 | 0.13 | 0.33 | 0.00 | 0.28 | 4/6 | 0.608 |
| P(Starget\|Snext) | 0.31 | 0.00 | 0.01 | 0.11 | 0.28 | 0.00 | 1/5 | 0.586 |
| P(Starget\|Sprev, Snext) | 0.36 | 0.00 | 0.16 | 0.00 | 0.20 | 0.01 | 1/5 | 0.561 |
| P(Starget, Snext) | 0.43 | 0.00 | 0.01 | 0.12 | 0.01 | 0.00 | 1/4 | 0.557 |
| P(Starget) | 0.09 | 0.27 | 0.06 | 0.00 | 0.25 | 0.00 | 2/5 | 0.529 |
| P(Starget\|Sprev) | 0.07 | 0.18 | 0.32 | 0.05 | 0.06 | 0.00 | 2/3 | 0.499 |
| P(Snext) | 0.01 | 0.03 | 0.00 | 0.00 | 0.47 | 0.01 | 2/5 | 0.497 |
| P(Wprev,Wtarget,Wnext) | 0.11 | 0.06 | 0.21 | 0.27 | 0.01 | 0.01 | 3/4 | 0.483 |
| P(Wtarget\|Wnext) | 0.24 | 0.01 | 0.08 | 0.01 | 0.04 | 0.20 | 1/6 | 0.440 |
| P(Sprev, Starget) | 0.24 | 0.19 | 0.09 | 0.02 | 0.02 | 0.00 | 1/2 | 0.430 |
| P(Wprev, Wtarget) | 0.17 | 0.01 | 0.15 | 0.24 | 0.02 | 0.04 | 1/4 | 0.406 |
| P(Wtarget\|Wprev) | 0.21 | 0.00 | 0.00 | 0.04 | 0.01 | 0.19 | 1/6 | 0.399 |
| P(Wtarget, Wnext) | 0.18 | 0.02 | 0.06 | 0.10 | 0.02 | 0.20 | 1/6 | 0.374 |
| P(Sprev) | 0.07 | 0.00 | 0.11 | 0.16 | 0.01 | 0.00 | 3/4 | 0.270 |

(`NHD`) and word length (`nSegperWord`) align, with opposite sign, with SC2. Several predictors (`P(Sprev, Starget, Snext)`, `WordActivation SmallWindow`, `WordActivation LargeWindow`, `P(Starget | Sprev, Snext)`, `P(Starget, Snext)`) align with PC1. In the (SC1, SC2) plane, these predictors are orthogonal to word length and neighborhood density. A third group of predictors, including `P(Wtarget)` and `P(Starget)`, are positioned between the axes, with medium correlations on both axes, instead of large correlations with either SC1 or SC2.

In the (SC1, SC2) plane, the response, represented by the blue arrow, emerges as

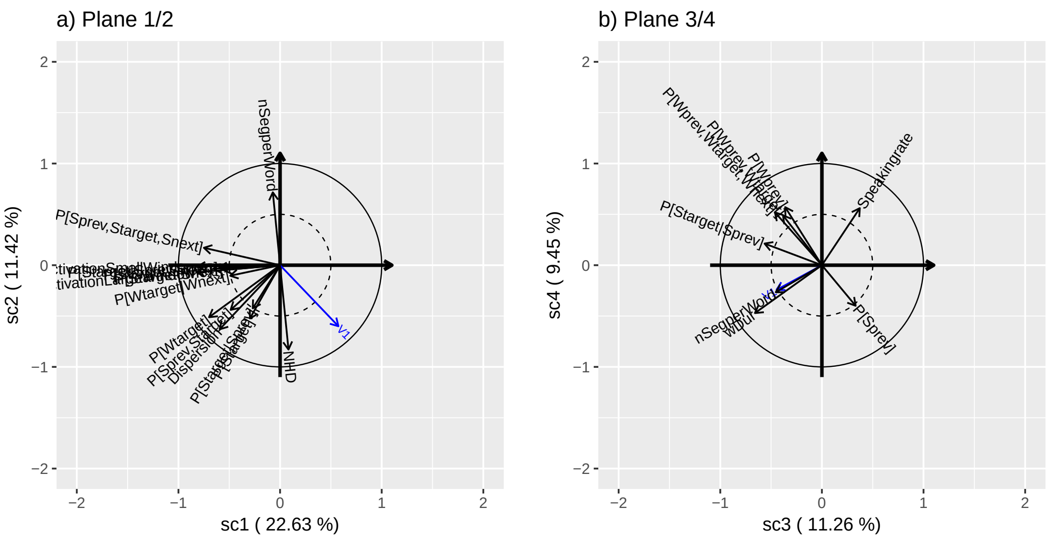

Figure 6: *Correlation plots for predictors and response in the planes defined by supervised components SC1 and SC2 (left) and SC3 and SC4 (right). Measures with square root best values less than 0.5 (which fall within the dashed circle) are not shown. The response variable is shown in blue.*

positively correlated with neighborhood density (`NHD`) and negatively correlated with word length. It is also negatively correlated with the predictors aligning with SC1, but more weakly. In this plane, the response is roughly orthogonal to the third group of predictors.

The right panel of Figure 6 presents the plane spanned by the third and fourth supervised components. In this plane, the response shows strong positive correlations with word length and word duration, and a strong negative correlation with speaking rate. Apart from `P(Starget|Sprev)`, other predictors that are well expressed in this plane are almost completely orthogonal to the response.

Considered jointly, the left and right panels of Figure 6 show that the orthogonalized space constructed by `scglr` succeeds to a considerable degree in allocating different kinds of variables to different subspaces. Durational measures (speaking rate, word duration) are dominant in the (SC3, SC4) plane, whereas a host of probability measures are dominant in the (SC1, SC2) plane. Furthermore, predictors that are well aligned with the response, either positively or negatively, such as neighborhood density and word length in (SC1, SC2), and word length, word duration and speaking rate in (SC3, SC4) may be expected to be strong predictors.

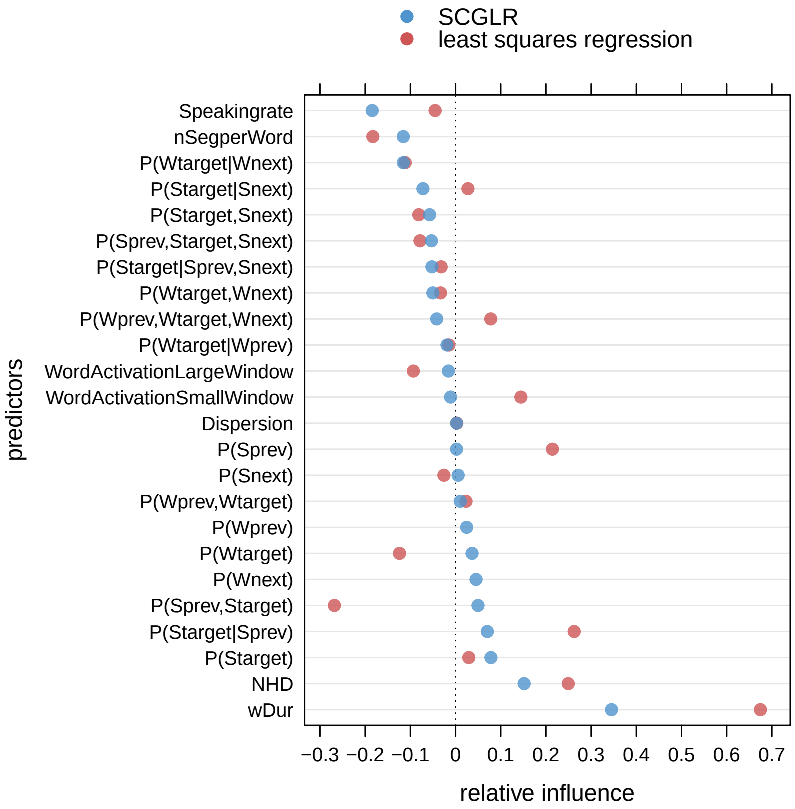

Figure 7: *Coefficients for the predictors estimates by the SCGLR (blue) and by a least squares regression model (red). The SCGLR substantially shrinks many of the large coefficients of the regression model towards zero.*

For assessing the strength of predictors, **SCGLR** makes available a table with the coefficients of the original predictors, which it derives from reduced space of supervised components. These coefficients reflect the cumulative support from all the dimensions of the (reduced) space of orthogonal supervised components. Figure 7 presents these coefficients in blue, together with the corresponding coefficients estimated by a standard linear regression model. Many of the large coefficients of the ordinary least squares regression have been shrunk towards zero in the SCGLR. For instance, `WordActivation LargeWindow` has a negative coefficient, whereas `WordActivation SmallWindow` has a positive coefficient in ordinary least squares regression. By contrast, the coefficients for both predictors are shrunk towards zero in the SCGLR. Likewise `P(Sprev, Starget)` and `P(Starget | Sprev)` have large coefficients with opposite signs in least squares regression, but substantially reduced positive coefficients in the SCGLR model.

In other words, the coefficients of the linear model have undergone ‘regularization’: the adverse effects of enhancement have been removed. This will be explained in more detail below in Section 5.4.1. Estimates of uncertainty about the coefficients of the predictors are not available after regularization, however. It is only for the supervised components that standard errors and p-values can be derived.

### *5.3.3. Advantages and disadvantages of SCGLR*

The squared correlation of the model predictions with the response are 0.428 for SCGLR and 0.504 for the standard regression model. When prediction accuracy is of primary importance, SCGLR is therefore a suboptimal choice compared to random forests.

What SCGLR does provide is insight into the magnitude and sign of the shrunk predictors. Here, it offers an important advantage over principal component regression. Recall that in contrast to principal components regression, which is designed to find high variance directions in the space of the predictors, SCGLR aims to find dimensions that are optimal for predicting the response. These different design principles enable SCGLR to better distinguish which of a set of correlated predictors are actually predictive for the response. We illustrate this for two highly correlated standard normal predictors, $A$ and $B$ and a dependent variable $Y$, for which the correlation between $A$ and $Y$, $r_{AY}$, is 0.5, and further $r_{BY} = 0$, and $r_{AB} = 0.8$. Analogous to the cases illustrated in Figure 1, a standard regression model will result enhancement, estimating a slope of -1.1 for $B$ even though $B$ is uncorrelated with $Y$. Orthogonalization with principal components analysis results in one predictor, the first principal component, that has loadings of 0.71 with both $A$ and $B$. Back-transformed coefficients using the *pcr* function from the **pls** package (Mevik et al., 2018) are 0.14 for both $A$ and $B$. In other words, the PCA regression does not detect that $B$ is not predictive for $Y$. However, SCGLR performs much better, with back-transformed coefficients for $A$ and $B$ of 0.42 and 0.05 respectively, a much improved approximation of the actual correlations 0.5 and 0.

## *5.4. Regression with the elastic net*

The elastic net (Zou and Hastie, 2005) is a regression technique that addresses collinearity by penalizing overly large $\beta$ estimates. In this way and unlike in SCGLR, highly collinear predictors may be pruned completely from the data. The elastic net combines the ideas behind two other regularization techniques: the lasso (Tibshirani, 1996) and ridge regression (also known as Tikhonov regularization; Hoerl, 1962; Hoerl and Kennard, 1970a,b).

Both ridge regression and the lasso penalize non-zero $\beta$ coefficients in an attempt to improve generalization performance. Ridge regression shrinks non-zero $\beta$ coefficients towards zero, but never to exactly zero. By contrast, the lasso shrinks $\beta$ coefficients of variables with limited predictor power to exactly zero. The lasso, therefore, allows for the selection of a set of the most predictive variables. The selection of a set of highly predictive variables is referred to in the machine learning and data mining literature as variable selection, predictor selection, or feature selection.

### *5.4.1. Regularization*

For a proper understanding of regularized regression it is important to understand how $\beta$ coefficients are estimated in standard linear regression. Standard linear regression models are least squares regression models, which minimize the sum of the squares of the residuals, commonly referred to as the residual sum of squares (henceforth RSS). The RSS is defined as:

$$\text{RSS} = \sum_{i=1}^{n} \Bigg( y_i - \big(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}\big) \Bigg)^2, \tag{6}$$

where $n$ is the number of observations, $p$ is the number of predictors, $y$ is the response variable, and $x_{ij}$ is the value of predictor $j$ for observation $i$. The term $y_i - \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$ represents the difference between the predicted and the observed values (equivalent to $\epsilon$ in Equation 3).

The RSS is small when the squared differences between the observed values ($y$) and the predicted values ($\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$) are small. Minimization of the RSS results in a high-quality fit to the data the model was fit to, but at the cost of suppression and enhancement for collinear data. Regularized regression, instead of minimizing the RSS, minimizes the RSS plus a penalty term that makes it costly to have large or many non-zero $\beta$ coefficients. The term that is minimized in the elastic net is:

$$\text{RSS} + \lambda \sum_{j=1}^{p} \Big( (1-\alpha)\beta_j^2 + \alpha|\beta_j| \Big). \tag{7}$$

The parameter $\lambda$ determines the strength of the penalty imposed on non-zero $\beta$ coefficients. As can be seen in Equation 7, both the absolute values of the coefficients ($|\beta|$) and the squared values of the coefficients ($\beta^2$) are penalized. The relative weight of the penalties on the absolute values of the coefficients and the squared values of the coefficients is set by the parameter $\alpha$. The setting of $\alpha$ determines the number of non-zero coefficients in the model, with higher values of $\alpha$ leading to fewer non-zero coefficients. The parameter $\alpha$ thus modulates the extent to which variable selection is performed. When $\alpha = 1$, the model imposes the lasso penalty, and when $\alpha = 0$, the ridge penalty is used.

### *5.4.2. Data preparation*

An implementation of the elastic net for R is available in the **glmnet** package (Friedman et al., 2010, 2018). Before we can run an elastic net model on the KIEL corpus data, we need to prepare the data for analysis with this package. The **glmnet** package does not support categorical predictors. We therefore converted categorical predictors in the KIEL corpus to numerical variables using one-hot encoding, as we did for the random forest analysis in section 5.2.

Estimates of the $\beta$ coefficients are sensitive to the scale of predictors. A change in the scale of a predictor leads to an equivalent change in the scale of the $\beta$ estimate, but does not influence the RSS of a regression model. By contrast, since the penalty term in regularized regression models takes into account $\beta$ coefficients, it is sensitive to the scale of predictors. The sensitivity of the penalty term to the scale of predictors has serious consequences for the estimation of the $\beta$ coefficients in regularized regression models because coefficients for predictors with larger scales are penalized more heavily than coefficients for predictors with smaller scales. As a result, regularized regression models are biased towards predictors with smaller scales. To prevent regularized regression models from being biased towards predictors with smaller scales, the predictors should be on the same scale. One way to ensure that predictors are on the same scale is standardization, which is enabled by default in the `glmnet` function.

### *5.4.3. Estimation of parameters*

Optimal values of $\alpha$ and $\lambda$ can be obtained with a grid search. Given a value for $\alpha$, the `glmnet` function will select an optimal value for $\lambda$. By letting $\alpha$ range over a sequence of values between 0 and 1, the optimal values of $\alpha$ and $\lambda$ can be found. To avoid overfitting, we made use of 10-fold cross-validation, using the `cv.glmnet()` function with the number of folds $n$ set to 10, and using the mean squared error (MSE), i.e. the average of the squared differences between the model predictions and the observed data, as an index of generalization performance. The MSEs reported below are average values of the mean squared error across the 10 folds.

Figure 8, top left, shows the cross-validation performance of the elastic net model for different values of $\alpha$. For each $\alpha$, the cross-validation score of the best model across 100 values of $\lambda$ is presented. The MSE of the best model for $\alpha = 0$ (which amounts to the ridge penalty), for instance, is 0.484. Error bars represent one standard error confidence intervals. As performance of the elastic net is highly similar for different values of the $\alpha$, apparently, for the KIEL data set, the influence of the balance between the squared and absolute values of the coefficients on the performance of the model for unseen data is minimal. Nonetheless, since we have to select a value of $\alpha$, we chose $\alpha = 0.7$, as this value yielded the lowest MSE of 0.480.

The center left panel of Figure 8 demonstrates how the MSE for $\alpha = 0.7$ varies with $\lambda$ under 10-fold cross-validation. To increase readability, $\lambda$ values are plotted on the log scale, which increases the relative distance between small values of $\lambda$ selected by the `cv.glmnet()` function. The cross-validation performance of the elastic net model is optimal for the smallest value of $\lambda$ that we inspected: $\lambda = 0.000472$ (MSE = 0.480). As $\lambda$ approaches zero, the contribution of the penalty term to the estimate of the coefficients approaches zero as well. As a consequence, the estimated coefficients for small values of $\lambda$ approach the least squares estimates of the coefficients. The fact that cross-validation performance of the elastic net is optimal for a very small value of $\lambda$ indicates that a least squares solution may generalize well for the current data.

For reasons of interpretability, we increase $\lambda$ beyond its optimal value to enforce regularization, as larger values of $\lambda$ result in a smaller number of non-zero coefficients. The key question is how much predictive accuracy we are willing to sacrifice for a more interpretable model. A common strategy is to choose the largest value of $\lambda$ for which the MSE is within one standard error of the minimum MSE (Breiman et al., 1984; Hastie et al., 2001). For the current model, this approach would lead to fixing $\lambda$ at 0.0162 ($\log \lambda$ = -4.124). For this value of $\lambda$, however, no less than 19 predictors still have non-zero coefficients.

Instead of using the one-standard error rule, we therefore placed a threshold on the percentage by which we allow the MSE of a model to be higher than the minimum MSE.

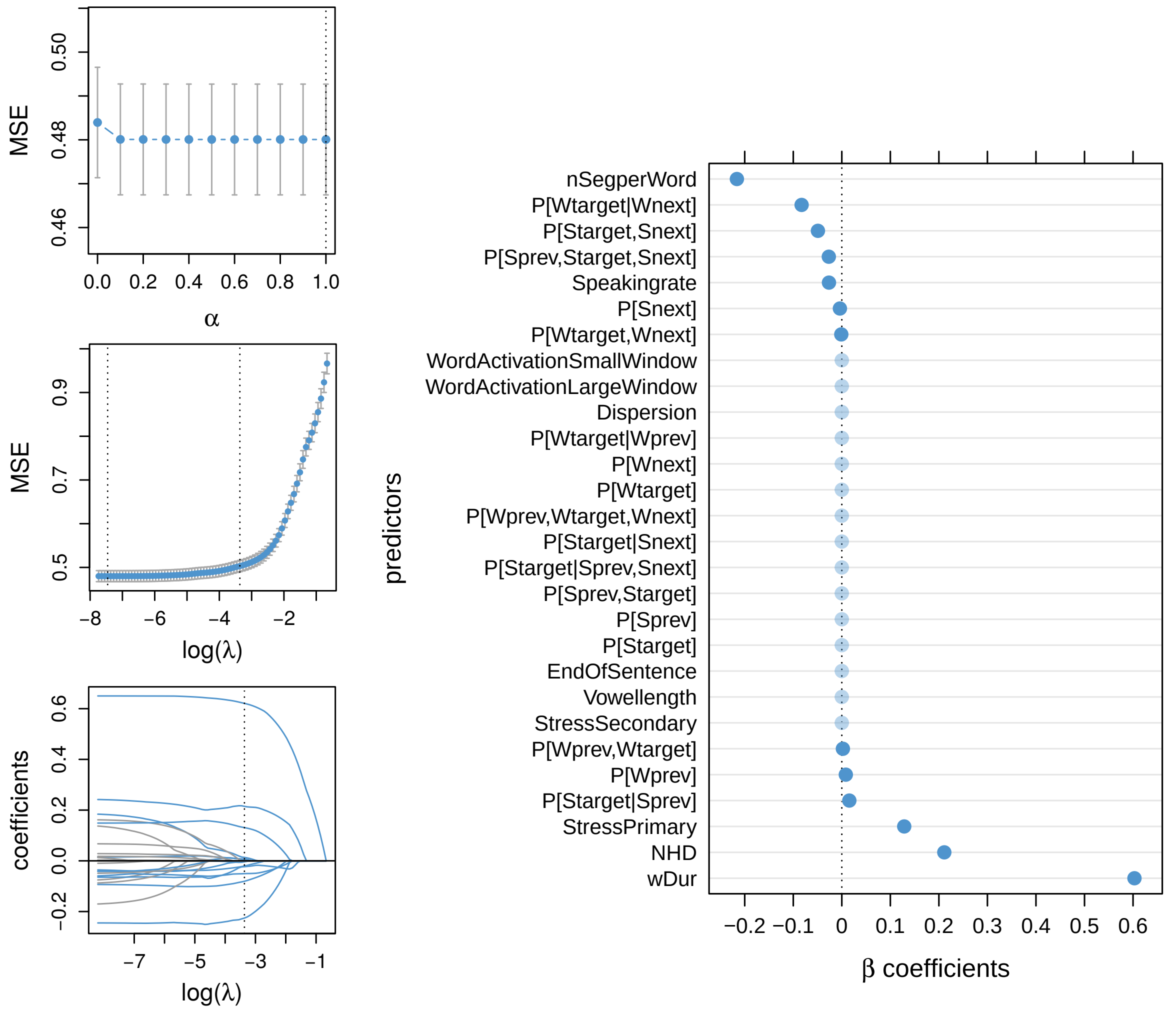

Figure 8: ***left:*** *Results of the elastic net models fit to the segment durations in the KIEL data. Top panel: optimal cross-validation performance for different values of the tuning parameter $\alpha$. The dashed line indicates the value of $\alpha$ for which the* MSE *is minimal ($\alpha = 0.7$) across all* MSE *values. Middle panel: cross-validation performance of the elastic net model with $\alpha = 0.7$ for different (logged) values of the penalty parameter $\lambda$. The dashed lines indicate the value of $\lambda$ for which the* MSE *is minimal ($\lambda = 0.00047$, $\log \lambda = -7.659$,* MSE *$= 0.480$) and the largest value of $\lambda$ for which the increase in* MSE *as compared to the* MSE *for the optimal value of $\lambda$ is no greater than 5% ($\lambda = 0.0494$, $\log \lambda = -3.007$,* MSE *$= 0.504$). Bottom panel: coefficient estimates for the elastic net model with $\alpha = 0.7$ as a function of $\lambda$. The dashed line indicates the largest value of $\lambda$ for which the increase in* MSE *as compared to the* MSE *for the optimal value of $\lambda$ is no greater than 5% ($\lambda = 0.0494$, $\log \lambda = -3.007$).* ***right:*** *Coefficient estimates for the elastic net model ($\alpha = 0.7$, $\lambda = 0.0494$). Opaque blue dots indicate non-zero coefficients, transparent blue dots represent coefficients shrunk to zero.*

The optimal value for the threshold depends on the relative importance we would like to place on predictive accuracy versus variable selection. Higher threshold values will lead to more variable selection, but less predictive power. We set the threshold to a relatively conservative value of 5%. The greatest value of $\lambda$ for which the increase in MSE is smaller than or equal to 5% is 0.0494 ($\log \lambda$ = -3.007, MSE = 0.504, increase in MSE = 4.94%). This allows us to update the function that regularizes the regression model to:

$$\mathrm{RSS} + 0.0494 \sum_{j=1}^{p} \Big(0.3 * \beta_j^2 + 0.7 * |\beta_j|\Big), \tag{8}$$

where $j$ is the number of predictors.

Figure 8, bottom left, shows how the magnitude of the 24 coefficients is shrunk towards zero as $\lambda$ is increased. Coefficients that for $\lambda = 0.0494$ ($\log \lambda$ = -3.007) and $\alpha = 0.7$ are not completely shrunk to zero are shown in blue, and the coefficients that are penalized to zero are shown in gray. For extremely small values of $\lambda$, the estimates of the coefficients approximate the least squares estimates of the predictors: no predictor selection is performed. For very large values of $\lambda$, the penalty term is very large and all coefficients are shrunk to zero. The coefficients for the selected value of $\lambda$ are located on the dashed line in Figure 8, bottom left. The right panel of Figure 8 presents the same shrunk coefficients in a dotplot, non-zero coefficients are represented by opaque blue dots and coefficients that are zero are represented by transparent blue dots. A total of 15 out of 28 coefficients were shrunk to zero.

Several predictors show the expected pattern of results: Segment durations (`nSegperWord`), for instance, are substantially shorter for words with more segments (Altmann, 1980) and greater conditional probability `P(Wtarget | Wnext)` of the word (Bell et al., 2009). By contrast, longer word durations (`wDur`), primary word stress (`StressPrimary`, Moon and Lindblom (1994)), and greater phonological neighborhood density (`NHD`) lead to longer segment durations. The direction of `NHD` is in line with findings by Scarborough (2003) and Baese-Berk and Goldrick (2009) who report enhancement of a segment's acoustic signal in words with greater `NHD`, but is at odds with recent findings by Gahl and Strand (2016), who reported shorter word durations for greater `NHD`.

Accurate standard errors for regularized regression models are not available (see Goeman, 2010). It is therefore advisable to refrain from reporting $p$-values for regularized regression models. Since cross-validated regularized regression models separate the pre-

Table 3: *Estimates of coefficient provided by the elastic net and by a least squares regression model fit to the reduced data set that contains only predictors with non-zero coefficients in the elastic net. Standard errors (S.E.), t-values and p-values are reported for the coefficients estimates of the least squares regression model.*

| term | elastic net $\beta$ | $\beta$ | S.E. | $t$-value | $p$-value |
|---|---|---|---|---|---|
| nSegperWord | -0.216 | -0.269 | 0.016 | -16.972 | < 0.001 |
| P(Wtarget \| Wnext) | -0.083 | -0.108 | 0.010 | -11.324 | < 0.001 |
| P(Starget, Snext) | -0.049 | -0.057 | 0.008 | -7.292 | < 0.001 |
| P(Sprev, Starget, Snext) | -0.027 | -0.032 | 0.010 | -3.336 | 0.001 |
| Speakingrate | -0.026 | -0.046 | 0.007 | -6.514 | < 0.001 |
| P(Snext) | -0.004 | -0.023 | 0.007 | -3.183 | 0.001 |
| P(Wtarget, Wnext) | -0.001 | -0.022 | 0.007 | -3.449 | 0.001 |
| StressSecondary | 0.000 | 0.276 | 0.406 | 0.679 | 0.497 |
| P(Wprev, Wtarget) | 0.002 | 0.033 | 0.008 | 4.176 | < 0.001 |
| P(Wprev) | 0.008 | 0.014 | 0.009 | 1.467 | 0.142 |
| P(Starget \| Sprev) | 0.015 | 0.032 | 0.007 | 4.304 | < 0.001 |
| StressPrimary | 0.129 | 0.368 | 0.020 | 17.980 | < 0.001 |
| NHD | 0.211 | 0.237 | 0.012 | 19.143 | < 0.001 |
| wDur | 0.603 | 0.650 | 0.011 | 58.616 | < 0.001 |

dictors into effective predictors (with non-zero coefficients) on the one hand, and ineffective predictors (with zero-coefficients) on the other hand, the selection of effective predictors replaces variable selection based on $p$-values and some (relatively arbitrary) $\alpha$-level.

It is of course possible to fit an unpenalized regression model with only those predictors that have non-zero coefficients in the regularized regression model. The coefficients of such a least squares regression model on the segment durations in the KIEL corpus are presented in Table 3, which also lists the corresponding values given by the elastic net. The two sets of predictors are similar, with the same signs, and a Pearson correlation of $r = 0.913$. There is only one coefficient, that for `P(WPrev)`, that is retained by the elastic net without being significant according to the unpenalized regression. Although for the unpenalized model all variance inflation factors are well below 5, the condition number is still high: 20.22. In this light, it is not surprising that the (absolute) magnitudes of the coefficients of the elastic net are smaller than those of the unpenalized regression, which

Table 4: *Lower triangle of the correlation matrix for the relative influences according to the elastic net, supervised component generalized linear regression (SCGLR), the random forest, and least squares regression.*

| | elastic net | SCGLR | least squares |
|---|---|---|---|
| SCGLR | 0.565 | | |
| least squares | 0.934 | 0.717 | |
| random forest | 0.965 | 0.413 | 0.861 |

are, on average, 0.049 and 0.155 respectively. The penalization implemented in the elastic net protects the estimates for the coefficients against collinearity-induced enhancement.

## 6. Discussion

Random forests, supervised component generalized linear regression, and the elastic net assess collinear data in very different ways. This raises the question of how results obtained with these statistical techniques compare.

To address this question, we need appropriate measures of the relative influence of a predictor. For the random forest analysis, we defined the relative influence of a predictor as its variable importance divided by the sum of the variable importances for all predictors. For the regression models, the relative influence of a predictor was defined as the absolute value of its coefficient divided by the sum of the absolute values of the coefficients of all predictors. For each of the three models, the relative influence of the predictors sums up to 1.

Figure 9 presents the relative influence of the predictors according to the elastic net (blue dots), according to the SCGLR (green dots), according to the random forest (red dots), and according to the least squares regression (yellow dots). The vertical axis shows the predictors in the KIEL data, in descending order of mean relative influence in the four models. As can be seen in Figure 9, the most important predictors have substantial relative influences according to all four modeling techniques. Similarly, the least important predictors have negligible relative influences across models.

Further information about the similarity of the relative influence of the predictors in the different models is presented in Table 4, which lists the correlations between the relative influences of the predictors across the four models. Relative influences of predictors

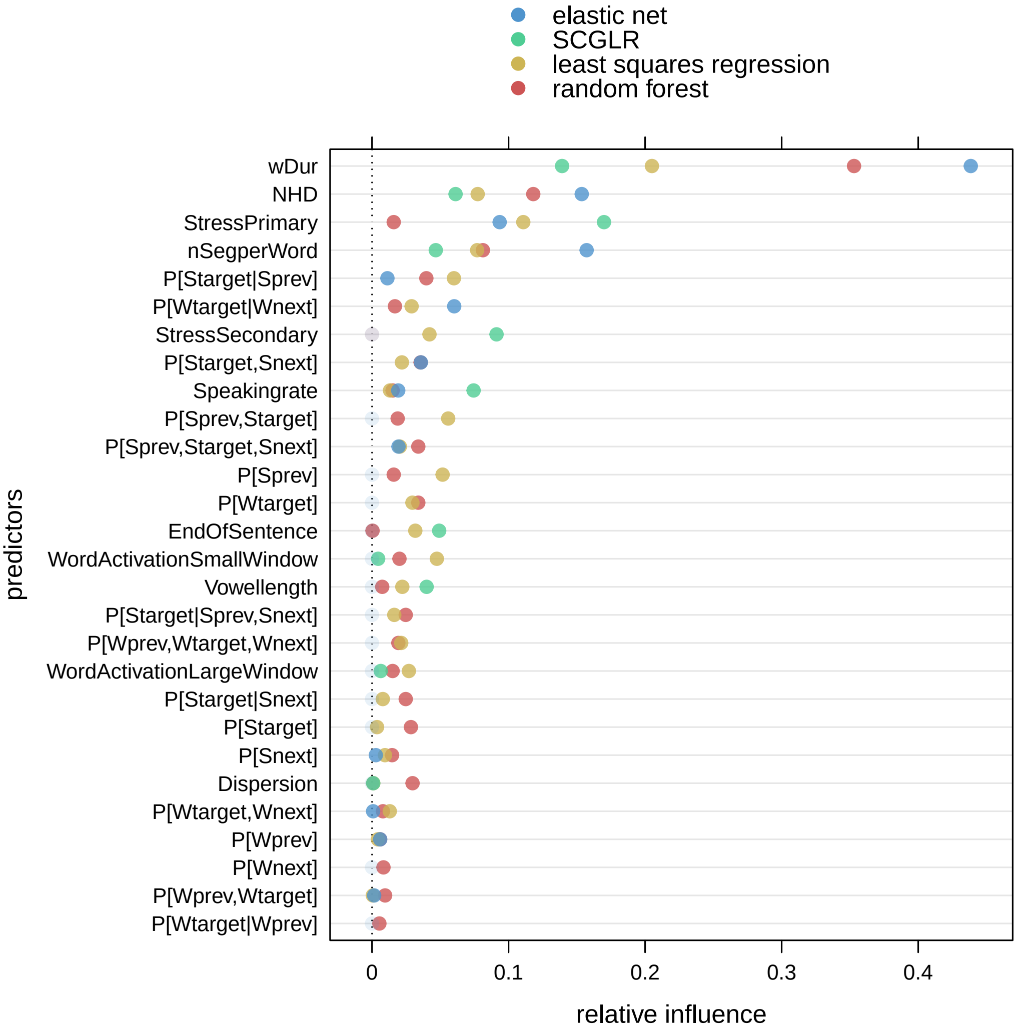

Figure 9: *Relative influence of the predictors according to the elastic net (blue dots), the SCGLR (green dots), least squares regression (yellow dots), and the random forest (red dots).*

are highly similar for the elastic net, least squares regression, and the random forest (all $r > 0.86$). The results of the random forest thus strongly converge with the results of two of the three regression techniques. The relative influences in the SCGLR are less similar to the relative influences in the other models (all $r < 0.72$, minimum $r = 0.413$). In part, this is due to our decision not to shrink factorial predictors when we fit the SCGLR model.

Prediction accuracy also differs substantially across models: the squared correlation of predicted and observed segment durations are 0.428 for SCGLR, 0.481 for the elastic net, 0.504 for the linear model, and 0.701 for the random forest. The low value of $R^2$ for SCGLR is unsurprising, as this model works with only 10 parameters, whereas the elastic retains 14 parameters, and the unpenalized linear model has no less than 28 parameters at its disposal (excluding the intercept).

The remarkable accuracy of the random forest is due to several factors. First, it is not assumed a-priori that effects of predictors are linear. Second, for SCGLR and the elastic net, we considered main-effect models only, as interactions between numeric predictors are best addressed with the generalized additive model (see Wieling, this issue), and not with the (highly constraining) multiplicative interaction available to the linear model. Conditional inference trees and random forests, however, are able to capture complex interactions involving many predictors. Third, random forests exploit the strengths of all predictors.

Thus, the choice of method will depend on the goal of the analysis. If this goal is prediction accuracy, the random forest is by far the best choice. If the goal is to understand the effects of predictors through the sign and magnitude of their slopes in a linear model, the elastic net conveniently weeds out insignificant predictors and provides estimates for the remaining coefficients that are properly shrunk.

SCGLR is an informative option when the goal is to better understand the high-dimensional space in which response and predictors are defined, and the joint effect of clusters of predictors on the response is of theoretical interest. Especially for studies in which the predictors are themselves not free of error and are best understood as contributing imperfect probes of the locations of data points in a high-dimensional space, SCGLR comes into its own.

To illustrate this point, consider the relative influence of word duration (`wDur`), number of segments (`nSegperWord`), and speaking rate (`Speakingrate`) in Figure 9. The elastic net assigns word duration the greatest relative influence, with number of segments as runner up. Speaking rate, by contrast, has a small relative influence that is much reduced compared to that of number of segments. Theoretically, this pattern is puzzling, as one would expect speaking rate to be the causal factor driving word duration. Furthermore, since the number of segments in a word is a poor man's substitute for word duration, it is also worrisome that the elastic net values number of segments so much over speaking rate. The relative influences of these predictors according to SCGLR, by contrast, are more intuitive. The relative influence of word duration is muted compared to unpenalized regression, instead of enhanced, as in the elastic net. Furthermore, speaking rate is accorded a much higher relative influence that exceeds that of number of segments. Because SCGLR has discovered that these three predictors are strongly represented in the (SC3, SC4) plane (see Figure 6), where they align with the response, it treats them similarly — the coefficients for SC3 and SC4 will give the three predictors the same boost (modulo their individual loadings on the SCs). As a result, their relative influences are more similar to each other. The reason that the elastic net generates very high relative influences for word duration and number of segments is that the penalty

$$\lambda \sum_{j=1}^{p} \Big( (1-\alpha)\beta_j^2 + \alpha|\beta_j| \Big).$$

in equation (7) can be kept low by substantially penalizing many intermediate coefficients and only mildly penalizing a few extreme coefficients. Importantly, the way the penalty is set up has no intrinsic value for linguistic theory, it is just a way to let fewer predictors do more work in such a way that prediction accuracy is optimized. The result is — indeed — a model with optimized prediction accuracy, but such a model may not be optimal from a theoretical perspective.

The data set with which we illustrated strategies for the analysis of collinear data includes information on the speaker, a predictor that within the general framework of mixed models would be included as a random-effect factor. This raises the question of how to adapt the three strategies discussed above when random-effect factors need to be taken into account.

Our experience with random forests is that, when participants are included into the term, partitions are made almost if not totally exclusively on subsets of participants, typically the largest source of variance. For random-effect factors with many factor levels, the combinatorics of working through possible partitions typically are too demanding for conditional inference trees and random forests to be estimable.

To our knowledge, there is no version of the elastic net that allows for the inclusion of random effects as in the linear mixed model (LMM Bates et al., 2014) and the generalized additive mixed model (GAMM Wood, 2006). It is possible to one-hot encode individual participants; the mechanism of penalization will ensure that the random effects for participants will be shrunk.

Principal components regression is easy to extend to the LMM and GAMM frameworks. For instance, a set of collinear predictors bound to items can be orthogonalized using principal components analysis, and pertinent principal components can then be used as predictors for the LMM or GAMM. For fully crossed mixed designs, SCGLR offers the possibility of bringing together subject responses into a multivariate response matrix, to be predicted from the (collinear) set of item-bound predictors. Unlike principal components regression, which orthogonalizes just the space of predictors, SCGLR will search for those directions in the space of the covariates that are optimal for predicting the responses of all of the subjects jointly. The resulting supervised components can, if required, be extracted from the model and used as predictors within a LMM or a GAMM.

From the preceding discussion, it will be clear that there are no hard and fast rules for the analysis of multivariate data with substantial collinearity.

Each of the statistical methods that we have reviewed has its advantages and disadvantages, and the choice of a method will depend, to a large extent, on the goals of the analysis. Regression models tend to be well-interpretable, but can be much less accurate than random forests. By contrast, random forests tend to provide surprisingly good predictions, but are more like a black box that does not allow inspection of how predictors work together to produce these good predictions. Even when individual trees are inspected, the number of interactions discovered by the tree can be overwhelming.

In addition, Important limitations of the regression-based methods is that effects are supposed to be linear, and that interactions of numeric predictors cannot be incorporated in a principled way. The generalized additive model (see Wieling, this volume) does not have these limitations. Unfortunately, the regression methods that we have surveyed are limited to linear (or linearizable) relations between response and predictors.

Furthermore, in the nonlinear world, the problem of collinearity resurfaces in the more general form of concurvity. Concurvity can lead to similar problems of interpretation, and it can render model estimates unstable. Concurvity occurs when one smooth term in the model can be approximated by other smooths in the model. This can happen, for instance, if a smooth of time is included together with further smooths for other time-varying covariates. Appendix A provides further information on how concurvity can be assessed, and how one might proceed if substantial concurvity in the model is detected.

We conclude with a reflection on the application of statistical analyses. In the context of confirmatory inference for collinear data, with as goal establishing whether a particular covariate is significant, the elastic net seems a good choice. If the covariate is not shrunk to zero, it can be accepted as supported, possibly in combination with further support from a least squares regression that discards all predictors that have been shrunk to zero by the net. For exploratory data analysis, all methods surveyed above are useful. The multiple testing method of Goeman and Solari (2011); Meijer and Goeman (2015), which is designed specifically for exploratory data analysis of collinear data, is an excellent companion to SCGLR.

## 7. Acknowledgements

This research was partially funded by two grants from the Deutsche Forschungsgemeinschaft (German Research Foundation) (BA 3080/3-1 and 387774888). Our thanks go to Theresa Schmitt for her help with preprocessing the KIEL corpus.

## References

Adelman, J., Brown, G., and Quesada, J. (2006). Contextual diversity, not word frequency, determines word-naming and lexical decision times. Psychological Science, 17(9):814.

Altmann, G. (1980). Prolegomena to menzerath's law. Glottometrika, 2:1–10.

Aylett, M. and Turk, A. (2004). The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. Language and Speech, 47(1):31–56.

Baayen, R. H. (2008). Analysing linguistic data: A practical introduction to statistics. languageR package Version 1.4.1. Cambridge University Press, Cambridge, MA.

Baayen, R. H., Feldman, L., and Schreuder, R. (2006). Morphological influences on the recognition of monosyllabic monomorphemic words. Journal of Memory and Language, 53:496–512.

Baayen, R. H., Milin, P., and Ramscar, M. (2016). Frequency in lexical processing. Aphasiology, 30(11):1174–1220.

Baayen, R. H., Milin, P., Đurđević, D. F., Hendrix, P., and Marelli, M. (2011). An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. Psychological review, 118(3):438–481.

Baayen, R. H., Vasishth, S., Kliegl, R., and Bates, D. (2017). The cave of shadows. addressing the human factor with generalized additive mixed models. Journal of Memory and Language, pages 206–234.

Baese-Berk, M. and Goldrick, M. (2009). Mechanisms of interaction in speech production. Language and Cognitive Processes, 24:527–554.

Bates, D., Maechler, M., Bolker, B., and Walker, S. (2014). lme4: Linear mixed-effects models using Eigen and S4.

Bell, A., Brenier, J. M., Gregory, M., Girand, C., and Jurafsky, D. (2009). Predictability effects on durations of content and function words in conversational English. Journal of Memory and Language, 60(1):92 – 111.

Belsley, D. (1984). Demeaning conditioning diagnostics through centering. The American Statistician, 38:73–77.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). Regression Diagnostics. Identifying

Influential Data and sources of Collinearity. Wiley Series in Probability and Mathematical Statistics. Wiley, New York.

Box, G. E. P. (1976). Science and statistics. Journal of the American Statistical Association, 71:791–799.

Breiman, L. (2001). Random forests. Machine Learning, 45(1):5–32.

Breiman, L., Cutler, A., Liaw, A., and Wiener, A. (2018). Package ´´randomForest".

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). Classification and decision trees. Wadsworth and Brooks, Monterey, CA.

Bry, X., Trottier, C., Verron, T., and Mortier, F. (2013). Supervised component generalized linear regression using a pls-extension of the fisher scoring algorithm. package version 2.0.3. Journal of Multivariate Analysis, 119:47 – 60.

Chatterjee, S. and Hadi, A. (2012a). Regression analysis by example. John Wiley & Sons, New York.

Chatterjee, S., Hadi, A., and Price, B. (2000). Regression analysis by example. John Wiley & Sons, New York.

Chatterjee, S. and Hadi, A. S. (2012b). Regression Analysis by Example. Fifth Edition. Wiley.

Farrar, D. E. and Glauber, R. R. (1967). Multicollinearity in regression analysis: The problem revisited. The Review of Economics and Statistics, 49(1):92–107.

Fox, J. and Weisberg, S. (2011). An R Companion to Applied Regression. Car Package Version 2.1-6. Sage, Thousand Oaks CA, second edition.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22.

Friedman, J., Hastie, T., Tibshirani, R., Simon, N., Narasimhan, B., and Qian, J. (2018). Package ´´glmnet", Version 2.0-13.

Friedman, L. and Wall, M. (2005). Graphical views of suppression and multicollinearity in multiple linear regression. The American Statistician, 59(2):127–136.

Gahl, S. (2008). "thyme" and"time" are not homophones. word durations in spontaneous speech. Language, 84(3):474–496.

Gahl, S. and Strand, J. (2016). Many neighborhoods: Phonological and perceptual neighborhood density in lexical production and perception. Journal of Memory and Language, 89:162 – 178.

Goeman, J. J. (2010). L1 penalized estimation in the Cox proportional hazards model. Biometrical Journal, 52(1):70–84.

Goeman, J. J. and Solari, A. (2011). Multiple testing for exploratory research. Statistical Science, 26(4):584–597.

Hadi, A. S. (1988). Diagnosing collinearity-influential observations. Computational Statistics and Data Analysis, 7(2):143 – 159.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). The Elements of Statistical Learning. Springer, New York.

Hoerl, A. E. (1962). Application of ridge analysis to regression problems. Chemical Engineering Progress, 58:54—-59.

Hoerl, A. E. and Kennard, R. W. (1970a). Ridge regression: Applications to nonorthogonal problems. Technometrics, 12(1):69–82.

Hoerl, A. E. and Kennard, R. W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67.

Hothorn, T., Hornik, K., Strobl, C., and Zeileis, A. (2018a). Package ´´party”.

Hothorn, T., Seibold, H., and Zeileis, A. (2018b). Package ´´partykit”.

Jurafsky, D., Bell, A., Gregory, M., and Raymond, W. D. (2000). Probabilistic relations between words: Evidence from reduction in lexical production. In Bybee, J. and Hopper, P., editors, Frequency and the emergence of linguistic structure. John Benjamins, Amsterdam.

Keuleers, E., Stevens, M., Mandera, P., and Brysbaert, M. (2015). Word knowledge in the crowd: Measuring vocabulary size and word prevalence in a massive online experiment. The Quarterly Journal of Experimental Psychology, (8):1665–1692.

Kohler, K. J. (1996). Labelled data bank of spoken standard German – The Kiel Corpus of read/spontaneous speech.

Kuhn, M. (2018). Package ´´caret”, Version 3.3.

Meijer, R. J. and Goeman, J. J. (2015). A multiple testing method for hypotheses structured in a directed acyclic graph. Biometrical Journal, 57(1):123–143.

Mevik, B.-H., Wehrens, R., Liland, K. H., and Hiemstra, P. (2018). Package ´´pls”, Version 2.6-0.

Milin, P., Feldman, L. B., Ramscar, M., Hendrix, P., and Baayen, R. H. (2017). Discrimination in lexical decision. PLOS-one, 12(2):e0171935.

Moon, S.-J. and Lindblom, B. (1994). Interaction between duration, context, and speaking style in english stressed vowels. he Journal of the Acoustical Society of America, 96:40–55.

Nicodemus, K. K., Malley, J. D., Strobl, C., and Ziegler, A. (2010). The behaviour of random forest permutation-based variable importance measures under predictor correlation. BMC Bioinformatics, 11(1):110.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. Philosophical Magazine, 2(6):559–572.

Peters, B. (2003). Die Datenbasis The Kiel Corpus.

Priva, U. C. (2015). Informativity affects consonant duration and deletion rates. Laboratory Phonology, 6(2):243–278.

R Core Team (2018). R: A Language and Environment for Statistical Computing, Version 3.3.3. R Foundation for Statistical Computing, Vienna, Austria.

Scarborough, R. (2003). Lexical confusability and degree of coarticulation. Annual Meeting of the Berkeley Linguistics Society, 29(1):367–378.

Sheather, S. (2009). A modern approach to regression with R. Springer Science & Business Media.

Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application and characteristics of classification and regression trees, bagging and random forests. Psychological Methods, 14(4):323–348.

Therneau, T., Atkinson, B., and Ripley, B. (2017). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-11.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, 58(1):267–288.

Tomaschek, F., Plag, I., Ernestus, M., and Baayen, R. H. (2018). Modeling the duration of word-final s in English with naive discriminative learning. Manuscript, University of Siegen/Tübingen/Nijmegen.

Tremblay, A. and Tucker, B. V. (2011). The effects of n-gram probabilistic measures on the recognition and production of four-word sequences. The Mental Lexicon, 6(2):302–324.

Venables, W. N. and Ripley, B. D. (2002). Modern Applied Statistics with S, Version 7.3-45. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Wei, T., Simo, V., Levy, M., Yihui, X., Jin, Y., and Zemla, J. (2017). Package ´´corrplot”, Version 0.84.

Wood, S. N. (2006). Generalized additive models: an introduction with R. Chapman and Hall/CRC, Boca Raton, Florida, U. S. A.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. version 0.10.1. Journal of Statistical Software, 77(1):1–17.

Wurm, L. H. and Fisicaro, S. A. (2014). What residualizing predictors in regression analyses does (and what it does not do). Journal of Memory and Language, 72:37–48.

York, R. (2012). Residualization is not the answer: Rethinking how to address multicollinearity. Social Science Research, 41(6):1379 – 1386.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, 67(2):301–320.

Zuur, A. F., Ieno, E. N., and Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. Methods in Ecology and Evolution, 1(1):3–14.

## Appendix A: Concurvity

For the analysis techniques in the main text we assumed that the effects of covariates are linear. To relax the linearity assumption, we exchange (regularized) regression modeling by regression with the generalized additive model (see Wieling, this volume, for an introduction). In the nonlinear world, the problem of collinearity resurfaces in the more general form of concurvity. Concurvity can lead to similar problems of interpretation, and can make model estimates to some extent unstable. Concurvity occurs when one smooth term in the model can be approximated closely by other smooths in the model.

The **mgcv** package provides a function `concurvity`, that calculates several related indices that all range between 0 and 1. The closer the concurvity index for a smooth is to 1, the greater the risk of a lack of identifiability of a clear estimate. The indices are all based on a decomposition of a given smooth $f$ into two parts, a part $u$ that is unique to that smooth's space, and a part $g$ that lies completely in the space of one or more other smooths. The indices evaluate how $g$ compares to $f$. In what follows, we consider the index that is the ratio of the squared Euclidean lengths of the vectors of $f$

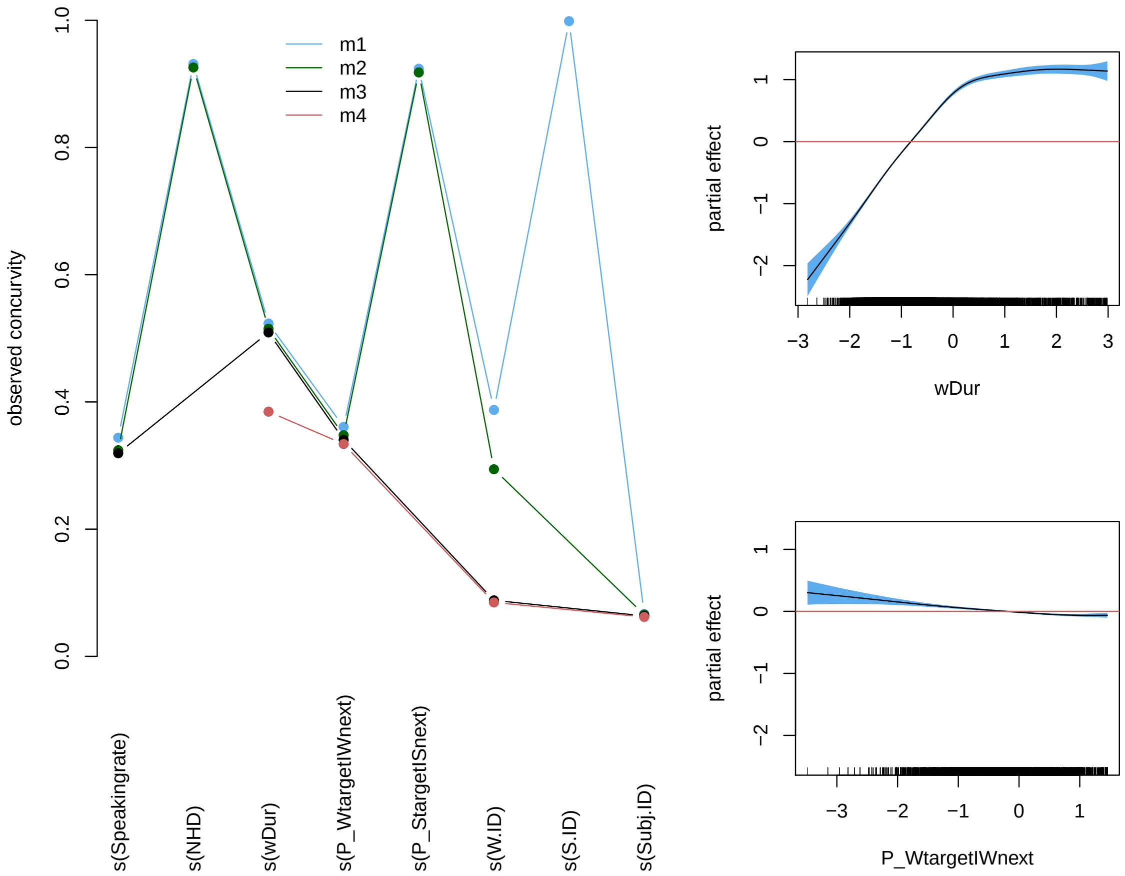

Figure 10: *Observed concurvity for models* `m1` *(blue),* `m2` *(dark green),* `m3` *(black) and* `m4` *(red), left panel, and the nonlinear effects of word duration and* *`P(Wtarget | Wnext)`* *(right panel).*

and $g$ when evaluated at the observed values of the covariates. This measure is possibly somewhat over-optimistic, for more pessimistic measures, the reader is referred to the documentation of the `concurvity` function.

We illustrate how concurvity can be diagnosed and addressed by fitting a generalized additive mixed model to the segment durations in the KIEL corpus. We include random intercepts for `speaker`, `word`, and `segment`, and the top seven best predictors that emerged from the analyses in the main text: `Speakingrate`, `nSegperWord`, `NHD`, `wDur`, `Stress`, `P(Wtarget | Wnext)`, and `P(Starget | Snext)` (see Figure 9). With the exception of `nSegperWord`, all numerical variables were modeled with thin plate regression spline smooths. The left panel of Figure 10 presents four GAM models with different sets of predictors. Model `m1` (blue) includes all predictors, whereas model `m4` (red) includes only two random effect factors, speaker and word, and only two smooths terms (`wDur` and (P(Wtarget | Wnext)). Models m2 and m3 are intermediate between m1 and m4 with respect to the predictors included. The left panel of Figure 10 presents the observed concurvity for each model. For the full model (m1), the random intercepts for segment emerge as completely unidentifiable. This model is clearly overspecified. But the neighborhood density measure (NHD) and the probability P(Starget | Snext) also are not well identifiable — they contribute little that is not already contributed by other predictors. Model m2 removes the by-segment random intercepts, but this does little to alleviate the problems with NHD and P(Starget | Snext). Model m3 removes these two predictors from the model specification, and model m4 removes Speakingrate, which was not well supported, thereby reducing the concurvity for wDur (which is strongly correlated with speaking rate). The right panel presents the nonlinear effects of wDur and (P(Wtarget | Wnext) in model m4; both predictors show muted effects for higher values, especially so for word duration.

In summary, when effects are nonlinear, concurvity may make it impossible to identify the unique contributions of predictors, even when model summaries suggest predictors are well supported. The problem is not that the requested model cannot be fit, or that the requested model does not improve on simpler models. Rather, the problem is that especially in the nonlinear world, the unique contribution of strongly correlated predictors will often not be separable. In this case, to further understanding without overfitting the data, while at the same time complying with Occam's razor, it is best to keep the model simple by removing predictors with high concurvity indices.

In the context of confirmatory data analysis where model m1 was the planned model, the removal of unidentifiable predictors would be part of model criticism, with as aim to obtain more reliable estimates of the effects (see Baayen et al. (2017), for discussion of the importance of model criticism in the context of confirmatory data analysis).