Variable Selection under Measurement Error: Comparing the Performance of Subset Selection and Shrinkage Methods

Ellen Sasahara

Bachelor's Thesis

Supervisor: Prof. Dr. Thomas Augustin
Department of Statistics
Ludwig–Maximilians–University
Munich

July 27, 2016

Abstract

In order to compare the performances of subset selection methods best subset and stepwise selection with shrinkage methods ridge and lasso under measurement error, I studied their median mean-squared errors of predictions and average number of zero-covariates of resulting models. I simulated data for a small number of large effects, a small to moderate number of medium-sized effects and a large number of small effects for three different kinds of classical measurement error. Cases were additional measurement error with homoscedastic, additional with heteroscedastic, and multiplicative with homoscedastic error structure. Underlying relationship between response and latent variables was linear.

For the first case predictions based on both shrinkage methods show the tendency to have lower median MSE's than subset selection methods. On a confidence level of 90%, ridge performs significantly better than best subset for all three kinds of effects. It was evident that with both non-present and present error subset selection methods tend to return more sparse models in average than lasso. Still there was no overall tendency for the models to in- or exclude more variables.

The second case where the error structure was heteroscedastic and dependant on the latent variable shows similar results. Generally with shrinkage methods there tend to be more accurate predictions than with subset selection. Ridge returns significantly lower median MSE's than best subset on a 90% confidence level for all analysed effects; with medium-sized and small effects, lasso does so as well. Again, both subset selection methods return more parsimonious models in average than lasso even under measurement error.

With a multiplicative measurement error and constant variance there was no significant difference (confidence level 95%) between the median MSE's of the methods, for any kind of effect used. Unlike in the additive cases, compared to non-present measurement error there wasn't as much change in which methods performed best or worst. The average number of zero-coefficients didn't change as drastically as well; subset selection methods keep to return more parsimonious models.

Concluding: when it is known or assumed that classical measurement error is present, best subset should probably avoided as it was shown that this method sometimes results in significantly higher MSE's than other methods. For more accurate predictions shrinkage methods tend to suit best; for sparse models stepwise selection does.

Contents

Chapter 1

Introduction and Overview

Why model selection? There are two reasons why the ordinary least squares does often not satisfy. First, there is *prediction accuracy*; although the least squares estimate generally has low bias, it also comes with large variance. To improve their accuracy, several methods shrink or set some coefficients to zero. This causes higher bias, but lower variance. The second reason is *interpretation*. Rather than having a large group of covariates, it is easier to interpret a smaller subset that exhibits the strongest effects. [3, p.57]
Measurement error happens when there is a variable that can't be fully observed, and one has to find some sort of replacement variable for it. This so-called surrogate contains the information of the real variable, but also comes with an error. Measurement Error in covariates cause what is referred to as the *triple whammy of measurement error*. First, Measurement Error results in *bias in parameter estimation*. Second, it causes *loss of power* which makes it harder to detect relationships among variables. Thirdly, measurement error *masks features of the data*, making graphical model analysis less efficient.[1, p.1]
Following graphs in 1.1 illustrate the impact measurement errors can have; this example is based on the one found in Carrol et.al's book. Say we have 200 observations of a variable $\mathbf{X}$ equally distributed between -2 and 2, which is somehow related to the response Y. The residuals ϵ of Y follow normal distribution with mean $sin(2X)$ and variance 0.09. Upper frame in figure 1.1 displays the real relationship between Y and $\mathbf{X}$, which is a simple linear equation $Y = X + \epsilon$. The dots seem to follow a sinusoid curve; the connection between the variables is clearly visible. But let's assume we don't know the real $\mathbf{X}$. Instead, we observe a surrogate $\mathbf{W} = \mathbf{X} + \mathbf{U}$ measured with an additive measurement error $\mathbf{U}$, standard normal. Now looking at the bottom frame, where Y is plotted against $\mathbf{W}$, it is harder to see how the two are related: the *features of the data is masked* and hidden; regression coefficients are *biased*, because the regression line would most likely not be a sinusoid; and last, the higher variability to the real regression curve means *loss of power*. [1, p.1]
This thesis aims to provide an overview on some model selection methods and compare their performance under measurement error in various situations. Basically, to answer the question: *Which method does best under given circumstances?*
In **Chapter 2** selected model/variable selection methods are presented. Starting with subset selection methods best subset and stepwise methods, it continues with shrinkage methods ridge and lasso. **Chapter 3** briefly deals with the theory behind classical measurement error, both additive and multiplicative. Simulations are done in **Chapter 4**. Here the performances of previously explained methods will be compared via MSE-Median. This chapter is divided in additive measurement error with homoscedastic error structure, and further structures. The

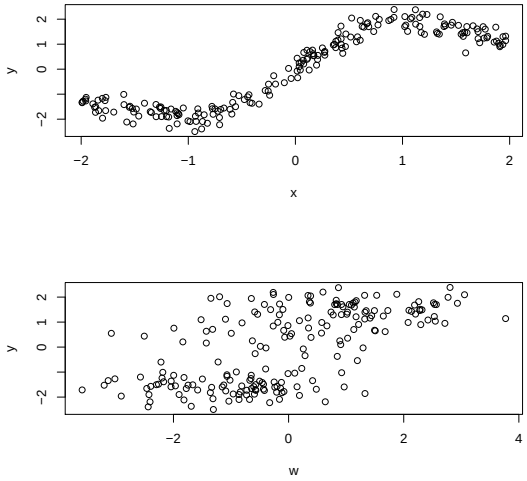

Figure 1.1: as in Carroll et.al [1, p.2]

varying component in the simulations is the type of real coefficient vector β. There's three scenarios: small number of large effects, small to moderate number of medium-sized effects and large number of small effects. **Chapter 5** summarizes the results of this thesis and tries to give a general conclusion on which methods do best under given circumstances. Moreover it gives suggestions to further possible research on this topic.

Throughout this thesis following notation will be used:
n sample size
p number of covariates (without intercept)
k number of datasets
$\mathbf{X}$ latent covariates
$\mathbf{W}$ corresponding surrogates
Y response variables
β regression parameters
boldface letters: collection of vectors indexed by $i = 1, ..., n$

Chapter 2

Variable Selection Methods

2.1 Subset Selection

Subset selection especially helps with the interpretation of larger models. Based on a chosen criterion, a predictor can either be in- or excluded. This means it is a discrete process; using the retained subset of variables, coefficients are estimated by least squares regression. [3, p.57]

2.1.1 Best Subset Selection

Best subset selection, also referred to as all-subset selection [2, p.164], tries to find a subset for each possible number of covariates $j \in \{1, ..., p\}$ with the smallest residual sum of squares. The desired model size j is chosen based on bias-variance tradeoff, parsimony and objective criteria like estimation of the expected prediction error via cross validation. Typically the smallest model that minimizes an estimate of it is chosen. An often used alternative is the AIC criterion. Downside to this method is its complexity; with today's computers it is excessive for large numbers of covariates $p \gg 40$. [3, p.57] This method can be performed using the leaps and bounds algorithm by Furnival & Wilson. [2, p.164]

2.1.2 Stepwise Methods

Having said that best subset selection might be too complex in some situations, stepwise methods are a different approach to perform subset selection. There are two main advantages over best subset. First one is *computational*, because stepwise methods work for larger p as well. Second is *statistical*: by selecting the best subset for each possible size, there would be more variance. Stepwise methods do a more constrained search, resulting in lower variance but maybe more bias. There's a few different ways do stepwise variable selection, but their idea is always the same. Forward-stepwise selection starts with an intercept model and subsequently adds the predictor that most improves the fit according to chosen criterion. Backward-stepwise starts, as the name suggests, with the full model and deletes the predictor with least impact on the fit. These stepwise methods are greedy methods, meaning that they don't 'go back'- they either always add or always exclude a predictor. [3, p.59] A hybrid of these two ways is the bothwards approach. Here, both directions are considered in each step. All three ways to do stepwise selection generally don't return the best model concerning the chosen criterion, but still a pretty good one. Possible criteria to base the methods on are BIC, AIC, CV (cross-validation to estimate expected prediction error) and Mallow's C_p (complexity parameter). In the context of

maximum-likelihood inference the most commonly used one is the AIC. When sparse models are desired, one might prefer the BIC over the AIC because this criterion generally returns more parsimonious ones. [2, p.162]

2.2 Shrinkage Methods

Subset selection is a discrete process, where a variable is either retained or discarded. Therefore, if all variables should be included in the model, prediction error of the full model would not be reduced. Another disadvantage is that models selected by this kind of method tend to have high variance. Shrinkage methods offer more improved qualities. They are more continuous and don't suffer as much from high variance. [3, p.61]

2.2.1 Ridge

The idea behind Ridge regression is to shrink regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares, [3, p.61]

$$\hat{\beta}^{ridge} = argmin\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\}.$$

Penalization happens with $\lambda \geq 0$, a complexity parameter which controls the amount of shrinkage. The bigger λ, the bigger the amount of shrinkage, meaning: coefficients are shrunk faster toward zero. [3, p.63] If $\lambda = 0$, so no penalty, we get ordinary least squares coefficients. Another way to write the equation is

$$\hat{\beta}^{ridge} = argmin \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{2.1}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t. \tag{2.2}$$

This visibly separates the residual sum of squares from the size constraint. In this form of equation t does what λ does in the other; only this time a smaller t means more shrinkage. In linear regression models, highly correlated variables can often become a problem as they can cause bias and loss of power. For example, a positive coefficient could be cancelled out by a negative coefficient on a variable that its correlated with; this issue can be eased by imposing a size constraint on them. One thing to remember is that ridge solutions vary depending on the scaling of the inputs, so they should be standardized beforehand. [3, p.63]

2.2.2 Lasso

Lasso is an abbreviation for *least absolute shrinkage and selection operator*. As the name suggests, it shrinks coefficients while also selecting variables. So basically this method offers *continuous subset selection* by shrinking some coefficients to exactly zero. [3, p.68]

$$\hat{\beta}^{lasso} = argmin \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{2.3}$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t \tag{2.4}$$

t is again the tuning parameter which controls the amount of shrinkage applied to the estimators. If t is large enough (larger than the sum of absolute β's), lasso estimator will be the same as ordinary least squares. Say, t is half the sum of absolute least squares *beta*'s, the new coefficients will be shrunk about 50% in average. If t is sufficiently small, some coefficients will be shrunk to zero; this provides continuous subset selection. To find the best value for t, one generally uses cross validation. It is chosen adaptively to minimize the estimation of the expected prediction error. [3, p.69] An equivalent from to write the problem (*Lagrangian form*) is

$$\hat{\beta}^{lasso} = argmin\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}.$$

The equations for ridge and lasso look quite similar; only difference is that the L_2 ridge penalty $\sum_{j=1}^{p}\beta_j^2$ (as in 2.2) now becomes the L_1 lasso penalty $\sum_{j=1}^{p}|\beta_j|$ (2.4). The lasso constraint makes the solutions nonlinear in y_i, and computing it is a quadratic programming problem. [3, p.69]

Chapter 3

Classical Measurement Error

Measurement error is present whenever the variable of interest can't be observed directly or measured correctly. The relationship between the available surrogate $\mathbf{W}$ and the latent variable $\mathbf{X}$ can be described with an error model;[6] in classical measurement error models, $var(\mathbf{W}) > var(\mathbf{X})$ holds. [1, p. 28]

3.1 Additive Measurement Error

Most common error model is *classical additive measurement error*. The underlying structure is taken to be

$$\mathbf{W} = \mathbf{X} + \mathbf{U}.$$

$\mathbf{X}$ is the true predictor that cannot be observed exactly for all subjects. This is sometimes referred to as *gold standard*. It is observed through its surrogate $\mathbf{W}$. The model assumes that $\mathbf{W}$ is an unbiased measure of $\mathbf{X}$ and does not contain any information that is not in the true predictor. [6] $\mathbf{U}$ denotes the additive error. Its expectation conditioned on $\mathbf{X}$ is assumed to be zero.[1, p. 3] Furthermore, the measurement errors are taken to be independent of each other. Their distribution must be known or estimated; [6] the error structure can be either homoscedastic or heteroscedastic, so a constant or inconstant variance.[1, p. 3] This means that they don't necessarily have to be identically distributed. [6]

3.2 Multiplicative Measurement Error

Another form of classical measurement error is *multiplicative measurement error*, where the relationship between latent variable and surrogate is thought to have this structure:

$$\mathbf{W} = \mathbf{X} * \mathbf{U}.$$

Here it is assumed that $\mathbf{U}$ has mean one.[1, p. 13] All other assumptions for the classical measurement error model apply, as explained in the previous section.

Chapter 4

Simulations: Comparing their Performance under Measurement Error

Methods analysed are best subset, stepwise, ridge and lasso. For reference, I included least squares as well. They will be compared via mean squared error median and average number of zero coefficients. Similar to Tibshirani's original lasso paper, the performance in three scenarios are examined:

1. small number of large effects
2. small to moderate number of medium-sized effects
3. large number of small effects

Tibshirani already discussed the relative merits of various methods in no-measurement-error case. There, lasso and ridge perform worse than best subset selection when exposed to a small number of large effects. With medium-sized effects, lasso does best followed by ridge and subset selection. For models with a large number of small effects, ridge by far surpasses lasso and subset selection. [5]

Data

As said above, the varying component in the simulations is the real coefficient vector β. For each distribution I simulated $k = 50$ datasets containing $n = 200$ observations. One set contains response Y, $p = 20$ latent variables $\mathbf{X}$, and their surrogates $\mathbf{W} = \mathbf{X} + \mathbf{U}$ in additive case or $\mathbf{W} = \mathbf{X} * \mathbf{U}$ in multiplicative case. $\mathbf{X}$ is always multivariate normal distributed with mean zero and $\sigma = 1$; the covariates are correlated with covariances $0.5^{|i-j|}$. Tibshirani used this covariance structure in one of his examples as well. It results in covariates with a slight correlation (which is probably more realistic than having none at all), but not too much (which is known to cause trouble in linear models). In additive measurement error cases, $\mathbf{U}$ follows normal distribution with mean zero and either constant (homoscedastic) or inconstant (heteroscedastic) variance. In the multiplicative measurement error case, $\mathbf{U}$ follows log-normal distribution with mean one and constant variance. The chosen constant variance is two in both cases; this way the ratio of variances is $1 : 2$, making sure the additional measurement error has an impact. In the heteroscedastic case the variances are $(2\mathbf{X})^2$, so dependent on the latent variable. The residuals ϵ of

the response Y are standard normal, and the relationship between response and covariates is of linear structure $Y = \mathbf{X}\beta + \epsilon$. In each step of the simulation, the current dataset is split to 66% training data and 33% test data. Resulting subsets are then divided in sets containing response and latent variables, and response and surrogates. Based on the training data the models are fitted, which then are used to predict response Y using remaining test data. Dividing the sets in training and test data avoids giving an advantage to methods that tend to overfit. So in the end we get results for a prediction of Y based on both $\mathbf{X}$ and $\mathbf{W}$, allowing to compare the changes in performance.

Used Methods

I used R for programming and simulation. For best subset selection I used the `bestglm` function with five-fold cross validation. Stepwise model selection is performed using the bothwards approach in `step`. For Ridge and Lasso I used `cv.glmnet`.

Output

The simulation computes the median mean-squared error for the predictions of each method over the 50 subsets. The MSE is the mean of the squared differences between observed and fitted numbers, so it tells how good the fit is. Here, the fitted numbers are calculated using the model (from the training data) on the remaining test data; together with their according responses, these are then used to calculate the MSE. I decided to choose median over mean for the MSE's to avoid the values being biased by possible outliers. The standard error of the vector of medians for each method is returned too, to detect tendencies that might just have happened by chance and determine whether there's significant differences. The average number of eliminated covariates is displayed as well, which can be used to compare parsimony of returned models.

4.1 Additive Measurement Error with homoscedastic error structure

4.1.1 Small Number of large Effects

In this scenario we have the real coefficient vector $\beta = (0, 6, 0, 0, 0, 0, 0, 0, 6, 0, 0, 6, 0, 0, 0, 0, 0, 0, 6, 0)^T$.
The table returned by the simulation contains the names of used methods, the median MSE

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	99.554 (2.199)	0
best subset	1.062 (0.025)	14.56	105.799 (2.282)	13.9
stepwise	1.103 (2.047)	12.64	100.498 (2.016)	10
ridge	2.367 (0.085)	0	97.546 (2.071)	0
lasso	1.115 (0.027)	10.38	98.483 (1.994)	5.3

Table 4.1: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

with standard errors in parentheses, plus the average number of zero coefficients all for both $\mathbf{X}$

and **W** as predictors.
Here in table 4.1 we can see that without measurement error, if the methods are exposed to a small number of large effects - 16 of the 20 entries in the real coefficient vector being zero - best subset selection does best with a median MSE of 1.062. It is followed closely by stepwise (1.103), lasso (1.115) and least squares (1.171). Ridge returns the least accurate predictions with a median MSE of 2.367. For most methods standard errors are very small, so it is unlikely that the numbers happened 'by chance'. Only exception is stepwise, where the standard error is 2.047. Looking at their 95% confidence intervals $[medianMSE - 1.96 * SE; medianMSE + 1.96 * SE]$ [4] the median MSE's of least squares, best subset, stepwise and lasso aren't significantly different because the intervals cross (see appendix for their lower and upper limits). Ridge results in a significantly higher median MSE than least squares, best subset and lasso. So what happens when there's the measurement error? First of all, the median MSE goes through the roof (as expected); but more interesting, the ratio between the methods changes. If we were only able to observe the surrogate **W**, ridge does best (median MSE 97.546), followed closely by lasso (98.483). Least squares lands on the third place with a median MSE of 99.554, and stepwise on forth with 100.498; best subset, formerly the winner, performs worst of them all by quite a margin (105.799). The standard errors don't differ much between the methods. Using a 95% confidence interval, the median MSE's aren't significantly different because the intervals cross. If we tighten the interval to a 90% one with multiplier 1.64, both shrinkage methods ridge and lasso result in a significantly lower median MSE than best subset. It seems that if exposed to this additive measurement error, predictions made with shrinking methods tend to have a lower mean squared error than with subset selection methods, especially best subset.
As for parsimony, least squares and ridge don't perform variable selection and will be left out of this part of th analysis. For both latent variable and surrogate models, best subset sets most coefficients to zero and stays comparably close to the real number 16 in both cases. Stepwise shows a similar tendency; it sets about twelve coefficients to zero in average for no-error case and ten with present error. With lasso, there's more change: although it returns a sparing model before, when the measurement error is added only about five covariates in average are left out of it.

4.1.2 Small to moderate Number of medium-sized Effects

The second scenario uses the real coefficient vector $\beta = (3, 2, 0, 0, 1.5, 0, 0, 0, 2.5, 4, 3, 2, 0, 0, 1.5, 0, 0, 0, 2.5, 4)^T$.
When there is a small to moderate number of medium-sized effects and no measurement error

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	73.565 (2.157)	0
best subset	1.112 (0.027)	9	87.157 (2.466)	12.62
stepwise	1.112 (0.029)	7.96	79.115 (2.276)	8.38
ridge	1.447 (0.041)	0	76.443 (2.164)	0
lasso	1.144 (0.028)	4.78	76.365 (2.128)	3.34

Table 4.2: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

(table 4.2), the results for best subset (1.112), stepwise (1.112) and lasso (1.144) only differ by

second to forth decimal place. And ridge with a median MSE of 1.447 does worst - by first decimal place. Unlike in the previous scenario, standard error is very small for all methods. Using the 95% confidence interval, ridge results in a significantly higher median MSE than the rest (which don't differ significantly). Again, things change with the added measurement error. In that case least squares performs best with a median MSE of 73.565, followed by lasso (76.365) and ridge (76.443). Both subset selection methods result in less accurate predictions. Stepwise returns a median MSE of 79.115, and best subset does - again - worst, with 87.157. Compared to the size of the median MSE's, standard errors are bigger than when there's few large effects, but constant over the methods. It turns out that least squares, ridge and lasso perform significantly better than best subset ona confidence level of 95%. Same as for the first scenario, models computed with shrinkage methods tend to result in a lower median MSE than with subset selection when exposed to additive measurement error.
So far the results of the first and second scenario are quite similar. The average number of excluded covariates on the other hand suggests some different tendencies. Best subset and stepwise return more sparse models than lasso in both no-error and present error cases; what differs from before are the numbers within the methods. First, they don't all drop when there's measurement error. Both subset selection methods return more parsimonious models when there is one. Second, the count of average zeroes for lasso doesn't change as much as in the first scenario - but was quite far away from accurate to begin with; about five in average without and three with measurement error.

4.1.3 Large Number of small Effects

In this third scenario seen in table 4.3 we have a large number of small effects:
$\beta = (0.1, 0.1, 0.1, 0.1, 1, 1, 1, 1, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 1, 1, 1, 1, 0.2, 0.2)^T$. When predictions are

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	11.158 (0.272)	0
best subset	1.256 (0.036)	8.66	12.778 (0.291)	12.44
stepwise	1.206 (0.030)	6.02	11.298 (0.255)	7.98
ridge	1.118 (0.028)	0	10.769 (0.263)	0
lasso	1.122 (0.029)	2.3	10.888 (0.265)	3.3

Table 4.3: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

based on the real variables, the median MSE doesn't differ much for the five methods; ridge performs best with a median MSE of 1.118. When lasso is used, the median MSE is 1.122. The shrinkage methods are followed by least squares (1.171), then stepwise (1.206) and best subset (1.256). The standard errors are quite similar for all methods; both shrinkage methods result in a significantly lower median MSE than best subset. When predictions are based on the surrogate with added measurement error, the order doesn't change. Ridge returns the most accurate predictions (median MSE of 10.769) and is closely followed by lasso (10.888). Predictions made using least squares (11.158) and stepwise (11.298) result in higher median MSE's. Best subset again performs worst with a median MSE of 12.778. Standard errors are relatively small, so the median of the MSE most likely represents their typical behaviour over all datasets. Interestingly now all methods perform significantly better than best subset to a confidence level of

95%. Through all three types of β's we observed a clear tendency: under additive measurement error with constant variance, models computed using shrinkage methods result in predictions with significantly lower median MSE than best subset. It is noticeable how in general shrinkage methods tend to return models with lower median MSE than subset selction methods. As for the average number of zero coefficients, both subset selection methods keep to return more parsimonious models with or without measurement error. Still there was no evident tendency for more or less sparse models when the measurement error was added.

4.2 Further Measurement Error Structures

4.2.1 Heteroscedastic Error Structure

In this section I changed another component of the simulation: the distribution of the measurement error **U**. Its mean remains zero, but instead of a constant variance, it has a heteroscedastic error structure with variances $(2\mathbf{X})^2$. This results in a measurement error that is dependent on **X**; this is not necessarily a common case. All other components remain the same as for the previous set-up.

Small Number of large Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	127.619 (3.193)	0
best subset	1.062 (0.025)	14.56	129.976 (3.071)	15.04
stepwise	1.103 (2.047)	12.64	126.287 (3.074)	10.52
ridge	2.367 (0.085)	0	119.091 (2.988)	0
lasso	1.115 (0.027)	10.38	121.517 (2.969)	6.28

Table 4.4: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

As the same **X** and Y are used as before, the first two columns of the results table are always equal; the parts concerning the surrogate **W** should differ. In the presence of an additive measurement error with inconstant variance dependent on the latent variable (table 4.4) and few large effects, ridge performs best with a median MSE of 119.091, followed by lasso (121.517). The subset selection methods stepwise (126.287) and least squares (127.619) result in a higher number than both shrinkage methods. Best subset with a median MSE of 129.976 returns (again) the least accurate predictions. Here, standard errors are quite large with about three for all methods, so this exact constellation might not represent the ones for all datasets. On a confidence level of 95%, the median MSE's aren't significantly different; as for a confidence level of 90%, ridge performs significantly better than best subset.
The average number of zero coefficients remains almost the same for best subset (from 14.56 to 15.04 in average), with or without error. As for stepwise, it falls from about 13 to about eleven; for lasso it falls from about ten to six in average.

Small to moderate Number of medium-sized Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	100.509 (3.736)	0
best subset	1.112 (0.027)	9	112.761 (3.624)	14.96
stepwise	1.112 (0.029)	7.96	103.724 (3.579)	10.06
ridge	1.447 (0.041)	0	96.988 (3.204)	0
lasso	1.144 (0.028)	4.78	97.299 (3.463)	5.28

Table 4.5: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

In case there is a small to moderate number of medium-sized effects with this measurement error type (table 4.5), ridge does best (median MSE 96.988), closely followed by lasso (97.299). Least squares (100.509) and both subset selection methods result in a higher median MSE. Stepwise returns a median MSE of 103.724 and best subset performs worst by a good margin with a median of 112.761. Again the standard errors are quite constant over all methods. On a confidence level of 95%, both shrinkage methods ridge and lasso return models with a significantly lower median MSE than best subset.
The additive heteroscedastic measurement error results in more sparse models for all best subset, stepwise and lasso. With error present, best subset sets about 15 coefficients to zero in average, which is further away than before (nine) from the real number ten. On the other hand, stepwise sets about eight coefficients to zero without error but about ten in average when there is one. Lasso shows the least absolute amount of change in covariates excluded; with and without measurement error it sets about five coefficients to zero. With or without present error, the ratio between the methods remains similar: both subset selection methods tend to set more coefficients to zero than lasso.

Large Number of small Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171	0	15.952 (0.460)	0
best subset	1.256	8.66	17.196 (0.512)	14.56
stepwise	1.206	6.02	15.444 (0.508)	9.6
ridge	1.118	0	15.118 (0.415)	0
lasso	1.122	2.3	15.486 (0.442)	5.16

Table 4.6: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

When given this large number of small effects and error is present, we can see in table 4.6 that ridge results in the lowest median MSE 15.118. It is closely followed by stepwise (15.444), lasso (15.486) and least squares (15.952). Models computed using best subset return the highest median MSE (17.196) with some margin. On a confidence level of 95%, ridge performs significantly better than best subset; on a level of 90%, both shrinkage methods and stepwise selection

do. The average number of zero coefficients rises in general, getting further away from the real number zero. Best subset remains the method that results in the most sparse models with about 15 coefficients set to zero; stepwise excludes in average about ten covariates and lasso five from their models.
Overall the methods behave similar with additive error having either homoscedastic or heteroscedastic error structure. Most of the time shrinkage methods tend to result in more accurate predictions than subset selection when measurement error is added. This is the case even when subset selection methods performed better when error wasn't present. Especially when based on best subset, predictions often turn out to be significantly less accurate than based on ridge and sometimes even more methods. It is also consistent that both subset selection methods keep to return more parsimonious models than lasso with or without added error. For both types of additive measurement error there is no evident tendency to in- or exclude more variables in any method.

4.2.2 Multiplicative

So far the chosen error model has been an additive model. Another option is to assume multiplicative measurement error of the form $\mathbf{W} = \mathbf{X} * \mathbf{U}$. In this set-up $\mathbf{U}$ follows log-normal distribution with constant variance two (homoscedastic error structure) and mean one. This avoids the surrogates to become too different from the latent variables.

Small Number of large Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	17.698 (0.620)	0
best subset	1.062 (0.025)	14.56	17.067 (0.656)	13.52
stepwise	1.103 (2.047)	12.64	17.400 (1.834)	11.18
ridge	2.367 (0.085)	0	18.416 (0.569)	0
lasso	1.115 (0.027)	10.38	17.016 (0.548)	7.6

Table 4.7: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

In the presence of multiplicative measurement error and a few large effects (as seen in table 4.7), lasso performs best with a median MSE of 17.016. It is closely followed by best subset (17.067), stepwise (17.400) and least squares (17.698). Least accurate predictions are returned by ridge (median MSE 18.416). Unlike with the additive measurement error, the median MSE doesn't differ as much between the methods. Without present error, prediction accuracy of all methods except ridge was almost the same and ridge performed little worse. We can see the same tendency for models computed based on the surrogate with multiplicative error. Similar thing goes for the standard errors: for every method except stepwise they are small and pretty close; and stepwise has a comparably large one, like when no measurement error present. On a confidence level of 95%, there is no significant difference between all the median MSE's.
The average number of zero coefficients falls for all variable selection methods. Best subset leaves out about 14 and stepwise eleven covariates in average. For lasso it changes the most, falling from about ten to about eight; thus meaning they all get further away from the real number of zero coefficients 16.

Small to moderate Number of medium-sized Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	10.889 (0.462)	0
best subset	1.112 (0.027)	9	10.519 (0.437)	8.98
stepwise	1.112 (0.029)	7.96	10.889 (0.452)	7.48
ridge	1.447 (0.041)	0	11.013 (0.428)	0
lasso	1.144 (0.028)	4.78	10.924 (0.443)	4.34

Table 4.8: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

With present error and a small to moderate amount of medium-sized effects (table 4.8), best subset returns predictions with lowest median MSE (10.519). Second comes stepwise and least squares both with a median of 10.889. Both shrinkage methods lasso (10.924) and ridge (11.013) perform a bit worse. There is no significant difference between the median MSE's of the methods; without measurement error, only ridge showed a significantly higher result. The average number of zero coefficients falls only slightly with multiplicative measurement error. For best subset it remains about nine, and stepwise excludes about seven variables in average (without error eight). Lasso also returns little less parsimonious models, without error it excludes five and with multiplicative error about four in average.

Large Number of small Effects

Method	Median MSE latent X	average no. of 0 coefficients latent X	Median MSE surrogate W	average no. of 0 coefficients surrogate W
least squares	1.171 (0.029)	0	2.306 (0.071)	0
best subset	1.256 (0.036)	8.66	2.502 (0.072)	9.28
stepwise	1.206 (0.030)	6.02	2.363 (0.097)	6.64
ridge	1.118 (0.028)	0	2.316 (0.065)	0
lasso	1.122 (0.029)	2.3	2.276 (0.068)	2.64

Table 4.9: Median mean-squared error (standard errors in parentheses) and average number of zero coefficients for non-present and present measurement error

In this last scenario (table 4.9), lasso performs best with present error and small effects, followed by least squares and ridge. Lasso returns predictions with a median MSE of 2.276, least squares 2.306 and ridge 2.316. The subset selection methods stepwise (2.363) and best subset (2.502) result in a slightly higher median MSE. Standard error doesn't differ much between the methods, and their resulting median MSE's are not significantly different. Presence of multiplicative measurement error results in models of about the same sparsity than without. For stepwise the number of average zero coefficients rises from about six to seven, and for lasso from two to three. Best subset remains to exclude about nine variables from its models.
Summarizing the results for multiplicative error, there is no evident tendency towards one method that seems to perform best or worst. There never is a significant difference of the performances

of any methods. Generally with the error simulated here we can see that the ratio between the performances of the methods doesn't change as much as with the additive measurement error. When a method returns higher or lower median MSE with models based on the latent variables, it tends to do so with models based on the surrogates as well. Again both subset selection methods set more coefficients to zero than lasso even with present error; yet again there is no overall tendency to more or less sparse models when multiplicative error is present.

Chapter 5

Concluding Remarks

Summary

In the first part of this thesis I gave an overview on subset selection and shrinkage methods, and classical measurement error. Presented methods are subset selection methods best subset and stepwise, and shrinkage methods ridge and lasso. The measurement error model discussed is classical, both additive and multiplicative structures. In the second part I compare the performances of presented methods under different kinds of measurement error. To do so, I calculated the median MSE's of the methods (plus least squares for reference) over 50 simulated datasets. Each dataset contains twenty multivariate normal distributed latent variables $\mathbf{X}$, and a response variable Y that is calculated using a linear model; for its real coefficient vector I examined three scenarios: a small amount of large effects, small to moderate number of medium-sized effects and a large number of small effects. The dataset also contains the corresponding surrogates $\mathbf{W}$ of the latent variables. These are computed using the according error model: additive with a homoscedastic error structure for the measurement error $\mathbf{U}$, additive with a heteroscedastic error structure, or multiplicative with a homoscedcastic error structure. In addition to the median MSE's of the predictions using models based on both latent variables and surrogates, average number of zero-covariates is calculated.

So for each of the three error structures, three scenarios (types of effects) have been analysed. When the measurement error is additive and has chosen homoscedastic error structure (variance two), shrinkage methods tend to result in predictions with lower median MSE than subset selection. On a confidence level of 90%, ridge and sometimes other methods perform significantly better than best subset in all scenarios. Overall best subest results in the highest medium MSE under this kind of measurement error. The average number of zero coefficients didn't show a general tendency; it was evident that the subset selection methods return more parsimonious models than lasso in both non-present and present measurement error cases. It seems that when one can assume this additive measurement error with constant variance, best subset should be avoided. There is also a tendency that shrinkage methods result in more accurate predictions than subset selection. If there's a strong desire for parsimony, stepwise is probably he best choice; this method returns more sparse models than lasso and its median MSE of the predictions is not significantly different.
In second case where the measurement error is additive but has heteroscedastic error structure (variance $(2\mathbf{X})^2$) dependent on the latent variable, the results are similar. Shrinkage methods often tend to perform better than subset selection, and best subset often results in the highest median MSE. For a small number of large effects, ridge returns a significantly smaller median MSE than best subset on a confidence level of 90%. With a moderate number of medium-sized

effects, both shrinkage methods ridge and lasso perform significantly better than best subset on a confidence level of 95%. When there's a large number of small effects, ridge performs significantly better on a 95% confidence level and lasso and stepwise as well on a 90% level. As for the average number of zero coefficients, there was no clear tendency towards more or less sparse models. Again both subset selection methods give more parsimony than lasso. We come to the same conclusions like with homoscedastic structure: generally best subset performs worst, and shrinkage methods a little better than subset selection. Again when sparse models are desired, stepwise does this better than lasso and its accuracy of predictions don't significantly differ. The third case covers multiplicative measurement error with constant variance (two). Here, the results of the simulation are not the same as for additive measurement error. The ratio between the methods (which one does best or worst) doesn't change as much. On a confidence level of 95%, there is no significant difference between the median MSE's with any of the three kinds of effects. There is no general tendency which methods perform best or worst. Only thing that's similar as with the additive measurement error is that the average number of zero coefficients doesn't show a tendency as well. Yet unlike the previous two cases the number doesn't change as much when the error as present versus non-present; subset selection methods return more sparse models in average than lasso. Concluding, when one can assume this kind of multiplicative measurement error with constant variance, there is no method to be preferred or avoided. It can be chosen based on own desires, like parsimony; then best subset would return the most sparse models in average.
Overall, when it is known or assumed that classical measurement error is present, best subset should probably avoided. We have seen that this method results in significantly higher MSE's than other methods on various occasions. When accurate predictions are desired, shrinkage methods tend to suit better; when parsimony is desired, stepwise selection does.

Outlook

Obviously this is thesis is not a comprehensive guide on what model selection method should be chosen when exposed to measurement error. For example the range of methods analysed could be expanded to more than subset selection and shrinkage methods. It would also be possible to use the same methods, but based on different criteria (e.g. BIC for stepwise). Another thing could be to include correction methods for measurement error before applying the methods. For the simulation, there's a couple of compounds that could be varied. As it is already known that shrinkage methods tend to cope with correlated variables better than some other methods, simulating highly correlated ones could result in interesting insights. Also using much bigger datasets with more variables might show some new tendencies. Another thing possible is to look at different distributions of the response; maybe having a link function between predictor and response makes methods behave diffently under measurement error. The structure of the measurement errors can be altered as well. This thesis focused on classical measurement error, and another option would be the berkson error. Last but not least, this whole topic could be analysed with a more formal approach (and not a simulation study): the influence of different components in the formula and what disturbance measurement error would cause.

Bibliography

[1] Raymond J. Carroll. *Measurement error in nonlinear models: A modern perspective*. 2nd ed. Vol. 105. Monographs on statistics and applied probability. Boca Raton, FL: Chapman & Hall/CRC, 2006. ISBN: 9781584886334. URL: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10144647.

[2] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression: Modelle, Methoden und Anwendungen*. 2., durchges. Aufl. Statistik und ihre Anwendungen. Berlin u.a.: Springer, 2009. ISBN: 978-3-642-01837-4.

[3] Trevor J. Hastie, Robert J. Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. 2. ed., corr. at 7. printing. Springer series in statistics. New York, NY: Springer, 2013. ISBN: 9780387848587.

[4] Lionel Hertzog. *Standard deviation vs Standard error*. 2015. URL: http://www.r-bloggers.com/standard-deviation-vs-standard-error/.

[5] Robert Tibshirani. *Regression Shrinkage via the Lasso*. Journal of the Royal Statistical Society. Series B (Methodological), 1996. URL: http://statweb.stanford.edu/~tibs/lasso/lasso.pdf.

[6] Thomas Augustin. "Survival analysis under measurement error". Habilitationsschrift. München: Ludwig-Maximilians Universität, 2002.

Appendix A

Confidence Intervals of Median MSE's

Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
least squares	1.11	1.23	95.24	103.86
best subset	1.01	1.11	101.33	110.27
stepwise	-2.91	5.12	96.55	104.45
ridge	2.20	2.53	93.49	101.61
lasso	1.06	1.17	94.58	102.39

Table A.1: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with homoscedastic error structure, small number of large effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.12	1.22	95.95	103.16
2	best subset	1.02	1.10	102.06	109.54
3	stepwise	-2.25	4.46	97.19	103.80
4	ridge	2.23	2.50	94.15	100.94
5	lasso	1.07	1.16	95.21	101.75

Table A.2: upper and lower limits of the 90% confidence intervals around the median MSE, case: additive measurement error with homoscedastic error structure, small number of large effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	69.34	77.79
2	best subset	1.06	1.16	82.32	91.99
3	stepwise	1.05	1.17	74.65	83.58
4	ridge	1.37	1.53	72.20	80.69
5	lasso	1.09	1.20	72.19	80.54

Table A.3: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with homoscedastic error structure, small to moderate number of medium-sized effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	10.62	11.69
2	best subset	1.19	1.33	12.21	13.35
3	stepwise	1.15	1.26	10.80	11.80
4	ridge	1.06	1.17	10.25	11.29
5	lasso	1.07	1.18	10.37	11.41

Table A.4: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with homoscedastic error structure, large number of small effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	121.36	133.88
2	best subset	1.01	1.11	123.96	136.00
3	stepwise	-2.91	5.12	120.26	132.31
4	ridge	2.20	2.53	113.23	124.95
5	lasso	1.06	1.17	115.70	127.34

Table A.5: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with heteroscedastic error structure, small number of large effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.12	1.22	122.38	132.86
2	best subset	1.02	1.10	124.94	135.01
3	stepwise	-2.25	4.46	121.25	131.33
4	ridge	2.23	2.50	114.19	123.99
5	lasso	1.07	1.16	116.65	126.39

Table A.6: upper and lower limits of the 90% confidence intervals around the median MSE, case: additive measurement error with heteroscedastic error structure, small number of large effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	93.19	107.83
2	best subset	1.06	1.16	105.66	119.86
3	stepwise	1.05	1.17	96.71	110.74
4	ridge	1.37	1.53	90.71	103.27
5	lasso	1.09	1.20	90.51	104.09

Table A.7: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with heteroscedastic error structure, small to moderate number of medium-sized effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	15.05	16.85
2	best subset	1.19	1.33	16.19	18.20
3	stepwise	1.15	1.26	14.45	16.44
4	ridge	1.06	1.17	14.30	15.93
5	lasso	1.07	1.18	14.62	16.35

Table A.8: upper and lower limits of the 95% confidence intervals around the median MSE, case: additive measurement error with heteroscedastic error structure, large number of small effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.12	1.22	15.20	16.71
2	best subset	1.20	1.31	16.36	18.04
3	stepwise	1.16	1.26	14.61	16.28
4	ridge	1.07	1.16	14.44	15.80
5	lasso	1.07	1.17	14.76	16.21

Table A.9: upper and lower limits of the 90% confidence intervals around the median MSE, case: additive measurement error with heteroscedastic error structure, large number of small effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	16.48	18.91
2	best subset	1.01	1.11	15.78	18.35
3	stepwise	-2.91	5.12	13.80	21.00
4	ridge	2.20	2.53	17.30	19.53
5	lasso	1.06	1.17	15.94	18.09

Table A.10: upper and lower limits of the 95% confidence intervals around the median MSE, case: multiplicative measurement error with homoscedastic error structure, small number of large effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	9.98	11.80
2	best subset	1.06	1.16	9.66	11.38
3	stepwise	1.05	1.17	10.00	11.78
4	ridge	1.37	1.53	10.17	11.85
5	lasso	1.09	1.20	10.06	11.79

Table A.11: upper and lower limits of the 95% confidence intervals around the median MSE, case: multiplicative measurement error with homoscedastic error structure, small to moderate number of medium-sized effects

	Method	lower limit latent X	upper limit latent X	lower limit surrogate W	upper limit surrogate W
1	least squares	1.11	1.23	2.17	2.44
2	best subset	1.19	1.33	2.36	2.64
3	stepwise	1.15	1.26	2.17	2.55
4	ridge	1.06	1.17	2.19	2.44
5	lasso	1.07	1.18	2.14	2.41

Table A.12: upper and lower limits of the 95% confidence intervals around the median MSE, case: multiplicative measurement error with homoscedastic error structure, large number of small effects

Appendix B

R-Code

Disclaimer: Some comment lines are altered to fit the page, and line breaks were added to code when necessary. See digital appendix for exact code and data used.

B.1 Illustrations.R

```
#-------------------------------------------------------------------------------
# Variable Selection under Measurement Error
# Bachelor's Thesis
# Illustrations
# Author: Ellen Sasahara
# Date: 27.07.2016
#-------------------------------------------------------------------------------

setwd("C:/Users/E. Sasahara/Google Drive/Studium/2016 SS/Bachelorarbeit/
Illustrations") # set working directory

# ME illustration ---------------------------------------------------------------
# illustration for introduction:
# the triple whammy of measurement error

set.seed(123) # set seed
x <- runif(200, -2,2) # equally distributed latent x
epsilon <- rnorm(length(x), mean=sin(2*x), sd= 0.3) # residuals of y
y <- x + epsilon # response y
# surrogate w, with additional measurement error u standard normal:
w <- x + rnorm(length(x))

# plotting response vs. latent variable and response vs. surrogate
pdf("illustration.pdf")
par(mfrow=c(2,1))
plot(x,y, ylab="y")
plot(w, y, ylab="y")
dev.off()
```

B.2 Functions.R

```
#-------------------------------------------------------------------------------
# Variable Selection under Measurement Error
# Bachelor's Thesis
# Functions
# Author: Ellen Sasahara
# Date: 21.07.2016
#-------------------------------------------------------------------------------

setwd("C:/Users/E. Sasahara/Google Drive/Studium/2016 SS/Bachelorarbeit/
Simulations")

library(hydroGOF)
library(boot)
library(glmnet)
library(MASS)
library(bestglm)
library(Matrix)

#-------------------------------------------------------------------------------
# Data Generation Function
# Input: number of dataframes k, type of real coefficient vector beta
# Output: list of length k of data frame with 200 observations of 21 variables;
# standard normal response Y (option "beta" to determine coefficients)
# multivariate normal 'latent' covariables X, mean 0 and symmetric
# covariance matrix
# surrogates W=X+U or W=X*U with U normally distributed,
# mean 0 and variance 2 or 1 and (2X)^2 (options "type" and "error")

mygen <- function(k=50,beta=c("few large", "some medium", "lot small"),
error=c("homoscedastic","heteroscedastic"), type=c("additive","multiplicative")){

dataframes <- vector(mode="list", length=k) # empty result vector
n <- 200 # 200 observations
p <- 20 # 20 variables
# real coefficient vector beta
if(beta=="few large") {beta <- rep(c(0,6,0,0,0,0,0,0,6,0), times=2)
} else if (beta=="some medium") {beta <- rep(c(3,2,0,0,1.5,0,0,0,2.5,4), times=2)
} else if (beta=="lot small") {
beta <- rep(c(0.1,0.1,0.1,0.1,1,1,1,1,0.2,0.2), times=2)}
set.seed(123) # seed
# Covariance matrix Sigma for latent variable X
Sigma <- matrix(c(NA),p,p)
for(i in 1:p){
if(i==1){
for(j in 1:p){
if(j==1){
vector <- 1
```

```
} else {
vector <- c(vector, 0.5^(j-i))}
}
Sigma[i,] <- vector
} else {
Sigma[i,] <- c(rep(0, times=(i-1)),vector[1:(p+1-i)])
}
}
Sigma <- as.matrix(forceSymmetric(Sigma))
Sigma <- round(Sigma, digits=4)

for (i in 1:k){
# latent variable X: multivariate normal distribution
X <- mvrnorm(n, mu=rep(0,times=p), Sigma = Sigma)
# error term U
if(type=="additive" && error=="homoscedastic"){
U <- matrix(rnorm(n*p, mean=0, sd=sqrt(2)), ncol=p)
} else if(type=="additive" && error=="heteroscedastic"){
U <- matrix(rnorm(n*p, mean=0, sd=abs(2*X)),ncol=p)
}

# surrogate W
if(type=="additive"){
W <- X+U
} else if (type=="multiplicative"){
U <- matrix(rlnorm(n*p, meanlog=0, sdlog=log(sqrt(2))), ncol=p) # U is lognomal
W <- X*U
}

# response Y
epsilon <- rnorm(n, mean=0, sd=1)
y <- X%*%beta + epsilon

# make dataframe
dataframes[[i]] <- data.frame(y,X=X,W=W)
}
return(dataframes)
}

#-----------------------------------------------------------------------------
# Simulation Function (Linear model)
# Input: list of dataframes dataframes, methods to be compared methods
# dataframe has to have response as first column, then latent variables X
# followed by surrogates W;
# data generation with function mygen
# Output: dataframe with median MSE and average number of zero-covariates of
# models with X and W

mysim <- function(dataframes, methods=c("least squares","best subset",
```

```
"stepwise","ridge","lasso") ){
k <- length(dataframes) #number of simulations
n <- dim(dataframes[[1]])[1] #number of observations per dataframe
p <- (dim(dataframes[[1]])[2] - 1) /2 # number of covariates (without intercept)
# set up empty result vectors
results <- data.frame(Method=vector(),Median.MSE.latent.X=numeric(),
SE.latent.X=numeric(), Avg.Zero.C.latent=numeric(),
Median.MSE.Surrogate.W=numeric(), SE.Surrogate.W=numeric(),
Avg.Zero.C.Surrogate=numeric())
mseX <- matrix(nrow=k,ncol=length(methods))
mseW <- matrix(nrow=k,ncol=length(methods))
zerosX <- matrix(nrow=k,ncol=length(methods))
zerosW <- matrix(nrow=k,ncol=length(methods))

# Factor for splitting data into test and traing sets
splitfactor <- 0.66

for(i in 1:k){
# sample train and test data
sample <- dataframes[[i]] # this loop's data
# indices for training data:
Train_index <- sample(1:nrow(sample),round(n*splitfactor),replace=FALSE)
Train_sample <- sample[Train_index,] # training set, 66% of the data
Test_sample <- sample[-Train_index,] # test set
# dividing sets into covariates X and W
Train_X <- Train_sample[,1:(1+p)]
Test_X <- Test_sample[,1:(1+p)]
Train_W <- cbind(Train_sample[,1], Train_sample[,(2+p):(1+2*p)])
Test_W <- cbind(Test_sample[,1], Test_sample[,(2+p):(1+2*p)])
colnames(Train_W)[1] <- "y"
colnames(Test_W)[1] <- "y"
# extracting X, W, Y
Xtr <- as.matrix(Train_X[,-1])
Wtr <- as.matrix(Train_W[,-1])
Ytr <- Train_X[,1]
Xte <- as.matrix(Test_X[,-1])
Wte <- as.matrix(Test_W[,-1])
Yte <- Test_X[,1]
# full least squares model on traing data
fullModelX <- lm(y~., data=Train_X)
fullModelW <- lm(y~., data=Train_W)
# Test Matrices for predictions
Test_MatrixX <- cbind(rep(1, n - length(Train_index)), Xte)
Test_MatrixW <- cbind(rep(1, n - length(Train_index)), Wte)

for(j in 1:length(methods)){
# for every method,
# 1. a model is fitted based on training data
# 2. y predicted by using the fitted model on test data
```

```
if(j==1){
# least squares
PredX <-  Test_MatrixX %*% coef(fullModelX)
PredW <- Test_MatrixW %*% coef(fullModelW)
# count zero-coefficients
thisZerosX <- sum(coef(fullModelX)==0)
thisZerosW <- sum(coef(fullModelW)==0)
} else if (j==2){
# best subset
fitX <- bestglm(Train_X[,c(2:(1+p),1)], family = gaussian, IC="CV", t=5)
setX <- Test_MatrixX[ , which(colnames(Test_MatrixX) %in%
as.vector(names(fitX$BestModel$coefficients)))]
setX <- cbind(rep(1, n - length(Train_index)), setX)
PredX <- setX %*% fitX$BestModel$coefficients
fitW <- bestglm(Train_W[,c(2:(1+p),1)], family = gaussian, IC="CV", t=5)
setW <- Test_MatrixW[ , which(colnames(Test_MatrixW) %in%
as.vector(names(fitW$BestModel$coefficients)))]
setW <- cbind(rep(1, n - length(Train_index)), setW)
PredW <- setW %*% fitW$BestModel$coefficients
# count zero-coefficients
thisZerosX <- p - length(fitX$BestModel$coefficients)
+ any(names(fitX$BestModel$coefficients)=="(Intercept)")
thisZerosW <- p - length(fitW$BestModel$coefficients)
+ any(names(fitW$BestModel$coefficients)=="(Intercept)")
} else if (j==3){
# stepwise
fitX <- stepAIC(fullModelX, trace="FALSE",direction="both")
setX <- Test_MatrixX[ , which(colnames(Test_MatrixX) %in%
as.vector(names(fitX$coefficients)))]
setX <- cbind(rep(1, n - length(Train_index)), setX)
PredX <- setX %*% fitX$coefficients
fitW <- stepAIC(fullModelW,trace="FALSE",direction="both")
setW <- Test_MatrixW[ , which(colnames(Test_MatrixW) %in%
as.vector(names(fitW$coefficients)))]
setW <- cbind(rep(1, n - length(Train_index)), setW)
PredW <- setW %*% fitW$coefficients
# count zero-coefficients
thisZerosX <- p - length(fitX$coefficients) +
any(names(fitX$coefficients)=="(Intercept)")
thisZerosW <- p - length(fitW$coefficients) +
any(names(fitW$coefficients)=="(Intercept)")
} else if(j==4){
# ridge
fitX <- cv.glmnet(Xtr,Ytr, family="gaussian", alpha=0)
PredX <- predict(fitX, Xte, s="lambda.min")
fitW <- cv.glmnet(Wtr,Ytr, family="gaussian", alpha=0)
PredW <- predict(fitW, Wte, s="lambda.min")
# count zero-coefficients
thisZerosX <- p - fitX$nzero[which(fitX$lambda.min==fitX$lambda)]
```

```
thisZerosW <- p - fitW$nzero[which(fitW$lambda.min==fitW$lambda)]
} else if(j==5){
# lasso
fitX <- cv.glmnet(Xtr,Ytr, family="gaussian", alpha=1)
PredX <- predict(fitX, Xte, s="lambda.min")
fitW <- cv.glmnet(Wtr,Ytr, family="gaussian", alpha=1)
PredW <- predict(fitW, Wte, s="lambda.min")
# count zero-coefficients
thisZerosX <- p - fitX$nzero[which(fitX$lambda.min==fitX$lambda)]
thisZerosW <- p - fitW$nzero[which(fitW$lambda.min==fitW$lambda)]
}
# calculate MSE
thisMseX <- mse(as.numeric(PredX),Yte)
thisMseW <- mse(as.numeric(PredW),Yte)
mseX[i,j] <- thisMseX
mseW[i,j] <- thisMseW
# save number of zero-coefficents in matrix
zerosX[i,j] <- as.numeric(thisZerosX)
zerosW[i,j] <- as.numeric(thisZerosW)
}
}
# calculate MSE-Median, MSE-vector's standard error
# and average number of zero-coefficients
for(j in 1:length(methods)){
results[j,] <- c(methods[j], round(median(mseX[,j]), digits=5),
round(sd(mseX[,j])/sqrt(length(mseX[,j])), digits=5),
round(mean(zerosX[,j], na.rm=T), digits=5),
round(median(mseW[,j]), digits=5),
round(sd(mseW[,j])/sqrt(length(mseW[,j])), digits=5),
round(mean(zerosW[,j], na.rm=T), digits=5))
}
# return results
return(results)
}

#-----------------------------------------------------------------------------
# Computing confidence intervals for MSE's
# Input: resulting dataframe from mysim, type of confidence interval (90 or 95%)
# Output: dataframe with lower and upper confidence interval limits for
# median MSE's

myconf <- function(data, level=c(90,95)){
multiplier <- (level==90)*1.64 + (level==95)*1.96
lower.X <- data[,2]-data[,3]*multiplier
upper.X <- data[,2]+data[,3]*multiplier
lower.W <- data[,5]-data[,6]*multiplier
upper.W <- data[,5]+data[,6]*multiplier
results <- data.frame(Method=data[,1], lower.X=round(lower.X, digits=3),
upper.X=round(upper.X, digits=3),
```

```
lower.W=round(lower.W, digits=3), upper.W=round(upper.W, digits=3))
return(results)
}
```

B.3 Simulations.R

```
#-------------------------------------------------------------------------------
# Variable Selection under Measurement Error
# Bachelor's Thesis
# Simulations
# Author: Ellen Sasahara
# Date: 07.07.2016
#-------------------------------------------------------------------------------

# set working directory:
setwd("C:/Users/E. Sasahara/Google Drive/Studium/2016 SS/Bachelorarbeit/
Simulations")
source("Functions.R") # loading functions written in Functions.R

# library(hydroGOF)
# library(boot)
# library(glmnet)
# library(MASS)
# library(bestglm)
library(xtable)

set.seed(123) # set seed

#-------------------------------------------------------------------------------
# Linear Model: Homoscedastic U ------------------------------------------------
# Simulation 1: few large effects ----------------------------------------------
# Additive error W=X+U, homoscedastic error structure for U, var(U)=2
# see file Functions.R for detailed description of mygen, mysim and myconf
# Normal distributed respose y
# real beta (0,6,0,0,0,0,0,0,6,0,0,6,0,0,0,0,0,0,6,0)^T

# Simulate over k subsets with sample size n=200
# generating k dataframes:
N_dataframes1 <- mygen(k=50, beta="few large", error="homoscedastic",
type="additive")
for(i in 1:50){
# save dataframes:
write.csv(N_dataframes1[[i]], paste0("Data/Normal/N_dataframes1_",i))
}
# compute MSE median of model selection methods:
N_sim1_k50n200 <- mysim(dataframes=N_dataframes1)
write.csv(N_sim1_k50n200, "Results/N_sim1_k50n200") # save results
# xtable(N_sim1_k50n200) # return table for latex file
```

```
# computing confidence interval limits for median MSE's
conf95_N_sim1_k50n200 <- myconf(data=N_sim1_k50n200, level=95)
write.csv(conf95_N_sim1_k50n200, "Results/conf95_N_sim1_k50n200")
# xtable(conf95_N_sim1_k50n200)
conf90_N_sim1_k50n200 <- myconf(data=N_sim1_k50n200, level=90)
write.csv(conf90_N_sim1_k50n200, "Results/conf90_N_sim1_k50n200")
# xtable(conf90_N_sim1_k50n200)

# Simulation 2: some medium effects ----------------------------------------
# real beta (3,2,0,0,1.5,0,0,0,2.5,4,3,2,0,0,1.5,0,0,0,2.5,4)^T

N_dataframes2 <- mygen(k=50, beta="some_medium", error="homoscedastic",
type="additive")
for(i in 1:50){
write.csv(N_dataframes2[[i]], paste0("Data/Normal/N_dataframes2_",i))
}
N_sim2_k50n200 <- mysim(dataframes=N_dataframes2)
write.csv(N_sim2_k50n200, "Results/N_sim2_k50n200")
# xtable(N_sim2_k50n200)

conf95_N_sim2_k50n200 <- myconf(data=N_sim2_k50n200, level=95)
write.csv(conf95_N_sim2_k50n200, "Results/conf95_N_sim2_k50n200")
# xtable(conf95_N_sim2_k50n200)

# Simulation 3: many small effects -----------------------------------------
# real beta (0.1,0.1,0.1,0.1,1,1,1,1,0.2,0.2,0.1,0.1,0.1,0.1,1,1,1,1,0.2,0.2)^T

N_dataframes3 <- mygen(k=50, beta="lot_small", error="homoscedastic",
type="additive")
for(i in 1:50){
write.csv(N_dataframes3[[i]], paste0("Data/Normal/N_dataframes3_",i))
}
N_sim3_k50n200 <- mysim(dataframes=N_dataframes3)
write.csv(N_sim3_k50n200, "Results/N_sim3_k50n200")
# xtable(N_sim3_k50n200)

conf95_N_sim3_k50n200 <- myconf(data=N_sim3_k50n200, level=95)
write.csv(conf95_N_sim3_k50n200, "Results/conf95_N_sim3_k50n200")
# xtable(conf95_N_sim3_k50n200)

#-------------------------------------------------------------------------
# Further Options: heteroscedastic U ---------------------------------------
# Simulation 1: few large effects ------------------------------------------
# heteroscedastic error structure for U: var(U)=(2X)^2, same beta's

H_dataframes1 <- mygen(k=50, beta="few_large", error="heteroscedastic",
type="additive")
for(i in 1:50){
write.csv(H_dataframes1[[i]], paste0("Data/Heteroscedastic/H_dataframes1_",i))
```

```
}
H_sim1_k50n200 <- mysim(dataframes=H_dataframes1)
write.csv(H_sim1_k50n200, "Results/H_sim1_k50n200")
# xtable(H_sim1_k50n200)

conf95_H_sim1_k50n200 <- myconf(data=H_sim1_k50n200, level=95)
write.csv(conf95_H_sim1_k50n200, "Results/conf95_H_sim1_k50n200")
# xtable(conf95_H_sim1_k50n200)

conf90_H_sim1_k50n200 <- myconf(data=H_sim1_k50n200, level=90)
write.csv(conf90_H_sim1_k50n200, "Results/conf90_H_sim1_k50n200")
# xtable(conf90_H_sim1_k50n200)

# Simulation 2: some medium effects ----------------------------------------

H_dataframes2 <- mygen(k=50, beta="some_medium", error="heteroscedastic",
type="additive")
for(i in 1:50){
write.csv(H_dataframes2[[i]], paste0("Data/Heteroscedastic/H_dataframes2_",i))
}
H_sim2_k50n200 <- mysim(dataframes=H_dataframes2)
write.csv(H_sim2_k50n200, "Results/H_sim2_k50n200")
# xtable(H_sim2_k50n200)

conf95_H_sim2_k50n200 <- myconf(data=H_sim2_k50n200, level=95)
write.csv(conf95_H_sim2_k50n200, "Results/conf95_H_sim2_k50n200")
# xtable(conf95_H_sim2_k50n200)

# Simulation 3: many small effects ----------------------------------------

H_dataframes3 <- mygen(k=50, beta="lot_small", error="heteroscedastic",
type="additive")
for(i in 1:50){
write.csv(H_dataframes3[[i]], paste0("Data/Heteroscedastic/H_dataframes3_",i))
}
H_sim3_k50n200 <- mysim(dataframes=H_dataframes3)
write.csv(H_sim3_k50n200, "Results/H_sim3_k50n200")
# xtable(H_sim3_k50n200)

conf95_H_sim3_k50n200 <- myconf(data=H_sim3_k50n200, level=95)
write.csv(conf95_H_sim3_k50n200, "Results/conf95_H_sim3_k50n200")
# xtable(conf95_H_sim3_k50n200)

conf90_H_sim3_k50n200 <- myconf(data=H_sim3_k50n200, level=90)
write.csv(conf90_H_sim3_k50n200, "Results/conf90_H_sim3_k50n200")
# xtable(conf90_H_sim3_k50n200)

#-----------------------------------------------------------------------------
# Further Options: multiplicative U ---------------------------------------
```

```
# Simulation 1: few large effects ----------------------------------------------
# W = X*U multiplicative error; homoscedastic error structure for U,
# with mean 1, non-negative
# same beta's

M_dataframes1 <- mygen(k=50, beta="few_large", error="homoscedastic",
type="multiplicative")
for(i in 1:50){
write.csv(M_dataframes1[[i]], paste0("Data/Multiplicative/M_dataframes1_",i))
}
M_sim1_k50n200 <- mysim(dataframes=M_dataframes1)
write.csv(M_sim1_k50n200, "Results/M_sim1_k50n200")
# xtable(M_sim1_k50n200)

conf95_M_sim1_k50n200 <- myconf(data=M_sim1_k50n200, level=95)
write.csv(conf95_M_sim1_k50n200, "Results/conf95_M_sim1_k50n200")
# xtable(conf95_M_sim1_k50n200)

# Simulation 2: some medium effects --------------------------------------------

M_dataframes2 <- mygen(k=50, beta="some_medium", error="homoscedastic",
type="multiplicative")
for(i in 1:50){
write.csv(M_dataframes2[[i]], paste0("Data/Multiplicative/M_dataframes2_",i))
}
M_sim2_k50n200 <- mysim(dataframes=M_dataframes2)
write.csv(M_sim2_k50n200, "Results/M_sim2_k50n200")
# xtable(M_sim2_k50n200)

conf95_M_sim2_k50n200 <- myconf(data=M_sim2_k50n200, level=95)
write.csv(conf95_M_sim2_k50n200, "Results/conf95_M_sim2_k50n200")
# xtable(conf95_M_sim2_k50n200)

# Simulation 3: many small effects ---------------------------------------------

M_dataframes3 <- mygen(k=50, beta="lot_small", error="homoscedastic",
type="multiplicative")
for(i in 1:50){
write.csv(M_dataframes3[[i]], paste0("Data/Multiplicative/M_dataframes3_",i))
}
M_sim3_k50n200 <- mysim(dataframes=M_dataframes3)
write.csv(M_sim3_k50n200, "Results/M_sim3_k50n200")
# xtable(M_sim3_k50n200)

conf95_M_sim3_k50n200 <- myconf(data=M_sim3_k50n200, level=95)
write.csv(conf95_M_sim3_k50n200, "Results/conf95_M_sim3_k50n200")
# xtable(conf95_M_sim3_k50n200)
```

Appendix C

Urheberrechtserklärung

Hiermit bestätige ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

München, 28. Juli 2016

(Ellen Sasahara)