Computer Intensive Methods in Statistics

Topics in EM Algorithm, SIMEX, Variable Selection[1]

Silvelyn Zwanzig, Behrang Mahjani

[1]This is the fourth chapter of the in-progress textbook Computer Intensive Methods in Statistics.

Chapter 4

Simulation based Methods

In this chapter, we introduce methods that are based on embedding principle. Assume we have a data set that belongs to a complicated model, such as missing data, hidden unobserved variable, or additional constrains. If we would have more observations, unconstraint without missing data or without hidden variables, we could find a big and "easier" model and we could apply "easier" methods. The trick of embedding based methods is to find such a big model and to simulate additional observations. Then we combine the original data set with the simulated observations and apply methods of the big model. It is just a change of the perspective, like in the Figure 4.1 and 4.2.

4.1 EM - Algorithm

The EM Algorithm is the oldest and maybe one of the most used method basing on the embedding principle. It was introduced in 1977 by Dempster et al.

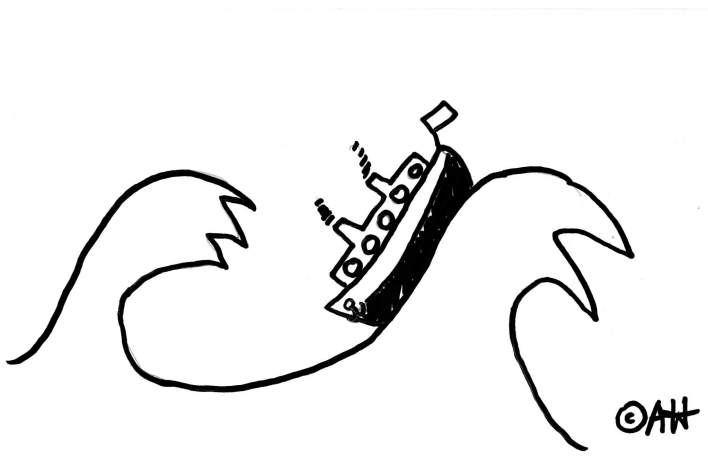


Figure 4.1: The observed environment.

Figure 4.2: Change the perspective!

in the Journal of Royal Statistical Society, B. The EM stands for expectation and maximization. As additional literature Chapter 10 and 12 of the Springer book (1999) on "Numerical Analysis for Statisticians" written by Kenneth Lange is recommended.
In the original formulation of the EM algorithm, there is no simulation step, instead a projection method is proposed. The set up is formulated as follows. Assume a data set includes the following observations:

$$\mathbf{Y} = (Y_1, \ldots, Y_m)$$

from the **observed model** with density $g(\mathbf{y} \mid \theta)$. We wanted to calculate the maximum likelihood estimator of θ. The observations are interpreted as a transformation

$$\mathbf{Y} = T(\mathbf{X}), \tag{4.1}$$

where

$$\mathbf{X} = (X_1, \ldots, X_n)$$

belongs to a **complete model** with density $f(\mathbf{x} \mid \theta)$. The transformation in (4.1) means that the observed data $\mathbf{y}$ are augmented to the data set $\mathbf{x}$. The trick is now to use the projection (conditional expectation) of the log likelihood function $\ln f(\mathbf{x} \mid \theta)$ as estimation criterion. The conditional mean is a projection of the likelihood function from complete model on to the observed model. In the original paper it was called surrogate function and is defined by

$$Q(\theta \mid \theta') = E(\ln f(\mathbf{X} \mid \theta) \mid T(\mathbf{X}), \theta').$$

where we have two parameters, θ is a free parameter in the likelihood function, and θ' is the parameter of the underlying distribution. Here it is assumed that the transformation is insufficient. The case of a sufficient statistic T is uninteresting, because in that case minimizing of the surrogate function is the same as minimizing the likelihood function $l(\theta) = \ln g(\mathbf{y} \mid \theta)$. For insufficient transformations the conditional expectation of $\mathbf{X}$ given $T(\mathbf{X})$ depends on the true underlying parameter θ_{true}. The EM algorithm is an iterative procedure for solving

$$\arg\max_{\theta} Q(\theta \mid \theta_{true}).$$

Algorithm 4.1 EM Algorithm

Given the current state $\theta^{(k)}$.

1. E-step (Expectation): Calculate $Q(\theta \mid \theta^{(k)})$.
2. M-step (Maximization):

$$\theta^{(k+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(k)}).$$

Why does EM work?
First we prove the following theorem, the information inequality. This inequality is the main argument that the maximum likelihood principle works.

Theorem 4.1 *Let $f > 0$ a.s, and $g > 0$ a.s. be two densities, then*

$$E_f \ln f \geq E_f \ln g.$$

Proof:
According to the Jensen inequality, let W be a random variable and let $h(w)$ be a convex function then

$$Eh(W) \geq h(E(W))$$

provided both expectations exists. Note, the function $-\ln(w)$ is strictly con-

vex. Thus

$$E_f \ln f - E_f \ln g = -E_f \ln\left(\frac{g}{f}\right) \geq -\ln E_f\left(\frac{g}{f}\right).$$

Furthermore

$$E_f\left(\frac{g}{f}\right) = \int g dx = 1.$$

We get

$$E_f \ln f - E_f \ln g \geq -\ln(1) = 0.$$

□

Now we compare the surrogate function with the log likelihood of the observed model

$$l(\theta) = \ln g(\mathbf{y} \mid \theta).$$

Theorem 4.2 *It holds*

$$Q(\theta' \mid \theta') - l(\theta') \geq Q(\theta \mid \theta') - l(\theta). \tag{4.2}$$

Proof: We have

$$\begin{aligned} Q(\theta \mid \theta') - l(\theta) &= E \ln(f(\mathbf{X} \mid \theta) \mid \mathbf{Y} = \mathbf{y}, \theta') - \ln g(\mathbf{y} \mid \theta) \\ &= E \ln(\frac{f(\mathbf{X} \mid \theta)}{g(\mathbf{y} \mid \theta)} \mid \mathbf{Y} = \mathbf{y}, \theta'). \end{aligned}$$

Note

$$\frac{f(\mathbf{X} \mid \theta)}{g(\mathbf{y} \mid \theta)}$$

is the density of the conditional distribution of $\mathbf{X} \mid (\mathbf{Y} = \mathbf{y}, \theta)$. We apply the information inequality and obtain

$$Q(\theta \mid \theta') - l(\theta) \leq E \ln(\frac{f(\mathbf{X} \mid \theta')}{g(\mathbf{y} \mid \theta')} \mid \mathbf{Y} = \mathbf{y}, \theta') = Q(\theta' \mid \theta') - l(\theta'). \tag{4.3}$$

□

The inequality (4.2) says that a maximizing of the surrogate function implies an increasing of the likelihood function $l(\theta)$ because for

$$\theta^{(k+1)} = \arg\max_{\theta} Q\left(\theta \mid \theta^{(k)}\right)$$

it holds

$$\begin{aligned} l(\theta^{(k+1)}) &\geq Q\left(\theta^{(k+1)} \mid \theta^{(k)}\right) + l(\theta^{(k)}) - Q\left(\theta^{(k)} \mid \theta^{(k)}\right) \\ &\geq l(\theta^{(k)}). \end{aligned}$$

In the original paper of Dempster et al. the following example was discussed.

Example 4.1 (**EM- Algorithm: Genetic Linkage Model**) Given data

$$\mathbf{y} = (y_1, y_2, y_3, y_4) = (125, 18, 20, 34)$$

from a four-category multinomial distribution

$$\mathbf{Y} \sim M(n, p_1, p_2, p_3, p_4),$$

with

p_1	p_2	p_3	p_4
$\frac{1}{2} + \frac{1}{4}p$	$\frac{1}{4}(1-p)$	$\frac{1}{4}(1-p)$	$\frac{1}{4}p$

We represent the data $\mathbf{y}$ as incomplete data from a five-category multinomial population:

$$y_1 = x_1 + x_2,\ y_2 = x_3,\ y_3 = x_4,\ y_4 = x_5 \tag{4.4}$$

with

$$\mathbf{X} \sim M(n, p_1, p_2, p_3, p_4, p_5) \tag{4.5}$$

where

p_1	p_2	p_3	p_4	p_5
$\frac{1}{2}$	$\frac{1}{4}p$	$\frac{1}{4}(1-p)$	$\frac{1}{4}(1-p)$	$\frac{1}{4}p$

(4.6)

The conditional distribution of $(X_1, X_2) \mid Y_1 = y_1$ is

$$(X_1, X_2) \mid Y_1 = y_1 \sim M(y_1, \frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2}).$$

Especially

$$X_1 \mid Y_1 = y_1 \sim Bin(y_1, \frac{p_1}{p_1 + p_2}), \;\; X_2 \mid Y_1 = y_1 \sim Bin(y_1, \frac{p_2}{p_1 + p_2}).$$

We are interested in a maximum likelihood estimation of p. Then under this set up the EM algorithm is as follows.

EM Algorithm: Genetic Linkage Example

Given the current state $p^{(k)}$.

1. (E) Estimate the "missing" observation x_2 by the conditional expectations.

$$x_2^{(k)} = E(X_2 \mid Y_1 = y_1, p^{(k)}) = y_1 \frac{\frac{1}{4}p^{(k)}}{\frac{1}{2} + \frac{1}{4}p^{(k)}}.$$

2. (M) Calculate the maximum likelihood estimation $p^{(k+1)}$ in model (4.5) with the data estimated observations $(x_1^{(k)}, x_2^{(k)}, y_2, y_3, y_4)$:

$$p^{(k+1)} = \frac{x_2^{(k)} + y_4}{x_2^{(k)} + y_2 + y_3 + y_4}.$$

Let us compare the algorithm above with the general EM algorithm. First we consider the more general set up of a multinomial model with additional parameterized cell probabilities

$$\mathbf{X} \sim M(n, p_1(\theta), \ldots, p_K(\theta)).$$

In this model the log likelihood function up to an additive constant is

$$\sum_{j=1}^{K} x_j \ln(p_j(\theta)) = \sum_{j=1}^{K-1} x_j \ln(p_j(\theta)) + (n - \sum_{j=1}^{K-1} x_j) \ln(1 - \sum_{j=1}^{K-1} p_j(\theta))$$

and the surrogate function up to an additive constant is

$$\begin{aligned} Q\left(\theta \mid \theta^{(k)}\right) &= E\left(\sum_{j=1}^{K} x_j \ln(p_j(\theta)) \mid \mathbf{Y} = \mathbf{y}, \theta^{(k)}\right) \\ &= \sum_{j=1}^{K} \ln(p_j(\theta)) E(x_j \mid \mathbf{Y} = \mathbf{y}, \theta^{(k)}), \end{aligned}$$

which is just the maximum likelihood function with the estimated data

$$\hat{x}_j^{(k)} = E(x_j \mid \mathbf{Y} = \mathbf{y}, \theta^{(k)}).$$

Consider now (4.6). $l'(\theta) = 0$ respects to

$$x_2 \frac{1}{p} - x_3 \frac{1}{1-p} - x_4 \frac{1}{1-p} + x_5 \frac{1}{p} = 0.$$

Note, x_1 is not included, because the the probability $p_1 = \frac{1}{2}$ does not depend on the parameter of interest p. We get

$$\widehat{p}_{ML} = \frac{x_2 + x_5}{x_2 + x_3 + x_4 + x_5}.$$

□

In the observed model, the likekilhood equation is more complicated to solve. We have

$$y_1 \frac{1}{\frac{1}{2} + \frac{1}{4}p} - (y_2 + y_3) \frac{1}{1-p} + y_4 \frac{1}{p} = 0.$$

Thus, using the MLE in the complete model with estimated observations $\widehat{x}_2$ give us an easier expression.

In this case of multinomial distributions with incomplete data the calculations in the E step and in the M step are carried out explicitly. Another important example is a mixture distributions, which belongs to models with a latent not observed variable. Also in this case some of the calculations in the E step and in the M step can be done explicitly.

Example 4.2 (**EM Algorithm: Mixture Distribution**)
Assume an i.i.d. sample $\mathbf{Y} = (Y_1, \ldots, Y_n)$ from $Y \sim F$, a mixture distribution with density $g(y \mid \theta)$, $\theta = (\pi, \vartheta_1, \vartheta_2)$

$$g(y \mid \theta) = (1 - \pi) f_{1,\vartheta_1}(y) + \pi f_{1,\vartheta_1}(y), \ \pi \in [0, 1]. \tag{4.7}$$

Mixture distributed random variables have a stochastic decomposition:

$$\begin{array}{rcl} Y & = & \Delta\, Z_1 + (1 - \Delta)\, Z_2, \\ Z_1 & \sim & F_1 \text{ with } f_1 = f_{1,\vartheta_1}, \quad Z_2 \sim F_2, \text{ with } f_2 = f_{2,\vartheta_2}, \text{ independent} \\ \Delta & \sim & Ber(\pi) \text{ independent of } Z_1 \text{ and } Z_2. \end{array}$$

The Bernoulli distributed random variable $\Delta \sim Ber(\pi)$ is latent and unobservable. The complete model is the following. Let be

$$\mathbf{X} = (X_1, \ldots, X_n)$$

an i.i.d. sample from with

$$X = (Y, \Delta) \ \ \Delta \sim Ber(\pi), \ X \mid \Delta = 0 \sim F_1, \ X \mid \Delta = 1 \sim F_2.$$

Then

$$f(x \mid \theta) = f(y \mid \Delta, \theta)\, f(\Delta \mid \theta) = \left(f_{1,\vartheta_1}(y)(1-\pi)\right)^{(1-\Delta)} \left(f_{2,\vartheta_2}(y)\pi\right)^{\Delta}.$$

Thus the log likelihood of the complete model is

$$\ln f(\mathbf{x} \mid \theta) = \sum_{i=1}^{n} (1-\Delta_i) \ln\left(f_{1,\vartheta_1}(y_i)(1-\pi)\right) + \Delta_i \ln\left(f_{2,\vartheta_2}(y_i)\pi\right).$$

The observed model is $T(\mathbf{X}) = \mathbf{Y}$ with density (4.7). The surrogate function is

$$Q\left(\theta \mid \theta^{(k)}\right) = \sum_{i=1}^{n} (1-\gamma_{i,k}) \ln\left(f_{1,\vartheta_1}(y_i)(1-\pi)\right) + \gamma_{i,k} \ln\left(f_{2,\vartheta_2}(y_i)\pi\right)$$

with $\gamma_{i,k} = \gamma\left(y_i, \theta^{(k)}\right)$, where

$$\begin{aligned} \gamma(y,\theta) &= E(\Delta \mid \theta) \\ &= P(\Delta = 1 \mid T(X) = y, \theta) \\ &= \frac{P(y \mid \Delta = 1, \theta)\, P(\Delta = 1)}{P(y \mid \theta)} \\ &= \frac{f_{2,\vartheta_2}(y)\pi}{f_{1,\vartheta_1}(y)(1-\pi) + f_{2,\vartheta_2}(y)\pi}. \end{aligned}$$

The $\gamma_{i,k}$ are called responsibilities and are interpreted as k'th iteration of the estimates of the latent variable Δ_i.
For the calculation of

$$\theta^{(k+1)} = \arg\max_{\theta} Q\left(\theta \mid \theta^{(k)}\right)$$

we solve the normal equations

$$\frac{d}{d\vartheta_1} Q\left(\theta \mid \theta^{(k)}\right) = \sum_{i=1}^{n} (1-\gamma_{i,k}) \frac{d}{d\vartheta_1} \ln\left(f_{1,\vartheta_1}(y_i)\right) = 0$$

$$\frac{d}{d\vartheta_2} Q\left(\theta \mid \theta^{(k)}\right) = \sum_{i=1}^{n} \gamma_{i,k} \frac{d}{d\vartheta_2} \ln\left(f_{2,\vartheta_2}(y_i)\right) = 0$$

and

$$\frac{d}{d\pi} Q\left(\theta \mid \theta^{(k)}\right) = \sum_{i=1}^{n} (1-\gamma_{i,k}) \frac{-1}{1-\pi} + \gamma_{i,k} \frac{1}{\pi} = 0.$$

The last equation gives $\pi^{(k+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}$.
Let us summarize the calculations and formulate the algorithm.

Algorithm 4.2 EM Algorithm: Mixture Distributions

Given the current state $\theta^{(k)} = \left(\pi^{(k)}, \vartheta_1^{(k)}, \vartheta_2^{(k)}\right)$.

1. Calculate the responsibilities for $i = 1, \ldots, n$

$$\gamma_{i,k} = \frac{f_{2,\vartheta_2^{(k)}}(y_i)\pi^{(k)}}{f_{1,\vartheta_1^{(k)}}(y_i)(1-\pi^{(k)}) + f_{2,\vartheta_2^{(k)}}(y_i)\pi^{(k)}}.$$

2. Find ϑ_1 such that

$$\sum_{i=1}^{n}(1-\gamma_{i,k})\frac{d}{d\vartheta_1}\ln\left(f_{1,\vartheta_1}(y_i)\right) = 0$$

and find ϑ_2 such that

$$\sum_{i=1}^{n}\gamma_{i,k}\frac{d}{d\vartheta_2}\ln\left(f_{2,\vartheta_2}(y_i)\right) = 0.$$

Update $\theta^{(k+1)} = \left(\pi^{(k+1)}, \vartheta_1^{(k+1)}, \vartheta_2^{(k+1)}\right)$ with

$$\pi^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n}\gamma_{i,k} \quad \vartheta_1^{(k+1)} = \vartheta_1, \quad \vartheta_2^{(k+1)} = \vartheta_2.$$

Consider the example of the mixture of two normals, $F_1 = N(\mu_1, \sigma_1^2)$, $F_2 = N(\mu_2, \sigma_2^2)$. Then

$$\mu_1^{(k+1)} = \frac{\sum_{i=1}^{n}(1-\gamma_{i,k})y_i}{\sum_{i=1}^{n}(1-\gamma_{i,k})}, \quad \mu_2^{(k+1)} = \frac{\sum_{i=1}^{n}\gamma_{i,k}y_i}{\sum_{i=1}^{n}\gamma_{i,k}} \tag{4.8}$$

and

$$\sigma_{1,k+1}^2 = \frac{\sum_{i=1}^{n}(1-\gamma_{i,k})(y_i-\mu_1^{(k+1)})^2}{\sum_{i=1}^{n}(1-\gamma_{i,k})}, \quad \sigma_{2,k+1}^2 = \frac{\sum_{i=1}^{n}\gamma_{i,k}(y_i-\mu_2^{(k+1)})^2}{\sum_{i=1}^{n}\gamma_{i,k}} \tag{4.9}$$

□

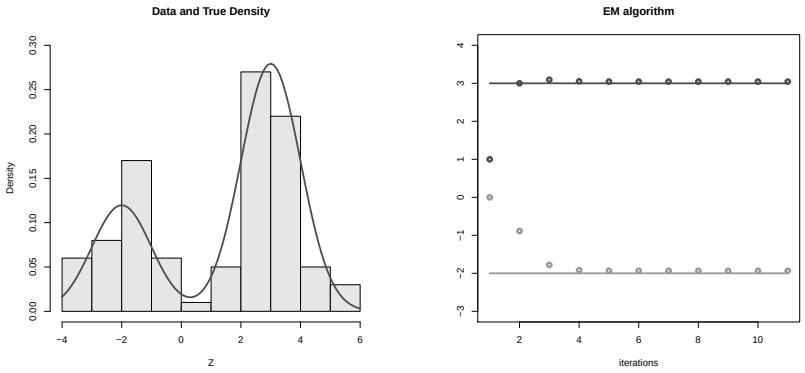


Figure 4.3: Example: EM algorithm for normal mixtures.

R Code 4.1.1. R-code:EM Algorithm for normal mixture

In this example, for simplicity we that π is known.

```
data<-Z # data simulated of two normal mixtures, sample size 100
pi<-0.3
r<-rep(NA,100) # responsibilities
Mu1<-rep(0,11)
# series of estimate of the first expectation
Mu2<-rep(0,11)
# series of estimate of the second expectation
Mu1[1]<-0; Mu2[1]<-1 # start values
for( j in 1:10)
{
    for (i in 1:100)
    {
        r[i]<-pi*dnorm(Z[i],Mu2[j],1)/((1-pi)*dnorm(Z[i],Mu1[j],1)
        +pi*dnorm(Z[i],Mu2[j],1));
    }
    Mu1[j+1]<-sum((1-r)*Z)/sum(1-r)
    Mu2[j+1]<-sum(r*Z)/sum(r)
 }
```

Note, mixture models deliver good approximation, but the interpretation of the latent will be lost. The EM algorithm becomes a simulation method, when the E step is carried out by a Monte Carlo method. This was proposed in 1990 by Wei and Tanner in the JASA. They assume that observed model of

$\mathbf{Y}$ has latent variables $\mathbf{Z}$. The complete model is related to $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$. Their algorithm is called MCEM.

Algorithm 4.3 MCEM Algorithm

Given the current state $\theta^{(k)}$.

1. E-step (Expectation): Calculate $Q(\theta \mid \theta^{(k)})$ by a Monte Carlo simulation
   (a) Generate independently $\mathbf{z^{(1)}}, \ldots, \mathbf{z^{(M)}}$ from $f(\mathbf{y}, \mathbf{z} \mid \theta)$, under the current step.
   (a) Approximate $Q(\theta \mid \theta^{(k)})$ by

$$Q_{k+1}(\theta \mid \theta^{(k)}) = \frac{1}{M} \sum_{j=1}^{M} \ln(f(\mathbf{y}, \mathbf{z^{(j)}} \mid \theta^{(k)}))$$

2. M-step (Maximization):

$$\theta^{(k+1)} = \arg\max_{\theta} Q_{k+1}(\theta \mid \theta^{(k)}).$$

The simulated latent variables $\mathbf{z^{(1)}}, \ldots, \mathbf{z^{(M)}}$ are also called **multiple imputations**, for more details see Rubin (2004).

4.2 SIMEX

The SIMEX method was introduced in 1994 by Cook and Stefanski in the JASA. SIMEX stands for Simulation-Extrapolation estimation. SIMEX is a measurement error model, and a special case of latent variable models. In difference to the EM algorithm, we observe the latent variable with an additional error (measurement error). SIMEX is a simulation method to get rid of the measurement error in the estimator. The main idea is to increase the measurement error by adding pseudo errors with stepwise increasing variances. The change of the estimator with respect to the increasing perturbation is modelised. The SIMEX estimator is the backwards extrapolated value to the theoretical point of no measurement error.

"Just make the bad thing worse, study how it behaves and guess the good situation."

Let us introduce the measurement error model and the general principle of SIMEX. After that we will study the simple linear model with measurement errors in more detail. First suppose a model without measurement errors:

$$(\mathbf{Y}, \mathbf{V}, \mathbf{U}) = (Y_i, V_i, U_i)_{i=1,\dots,n} \sim F_\theta$$

with

$$E(Y_i \mid U_i, V_i) = G(\theta, U_i, V_i).$$

Example 4.3 (**Blood Pressure**) Consider a study on the influence of blood pressure on the risk for a stroke. The data set is given by

$$(Y_i, V_i, X_i)_{i=1,\dots,n},$$

where Y_i is Bernoulli distributed. The "success" probability is the probability that the i'th patient get a stroke. The variable V_i consist of control variables like the sex, the weight, age of patient i. The variable X_i is the measurement of the blood pressure U_i of patient i. We are interested in estimating the parameter $\theta = (\beta_0, \beta_1^T, \beta_2)$ of

$$E(Y_i \mid U_i, V_i) = \frac{\exp(\beta_0 + \beta_1^T V_i + \beta_2 U_i)}{1 + \exp(\beta_0 + \beta_1^T V_i + \beta_2 U_i)}.$$

□

Another example for a measurement error model is the problem of calibration of two measurement methods.

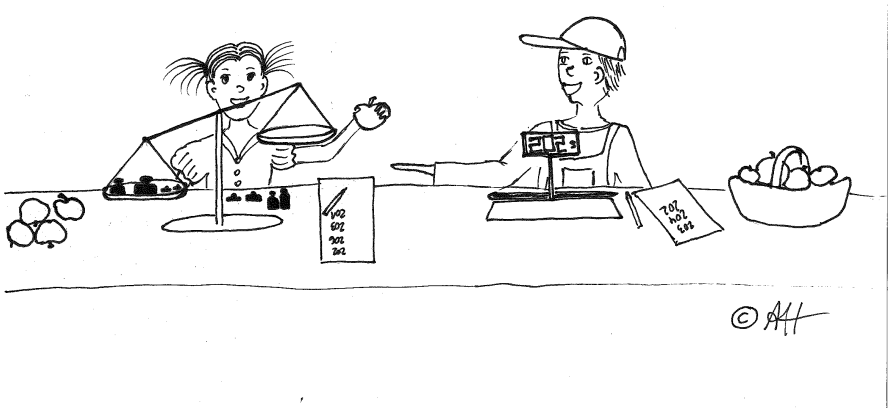


Figure 4.4: Calibration of a balance with a digital instrument.

Example 4.4 (**Calibration**) Suppose we are interested in comparing two instruments. Each instrument produces another type of measurement error. Our aim is to find a relation for transforming data sets from an old instrument in order to compare them with new data obtained from a new advanced instrument. For that we have to apply both instruments on the same objects as the children are doing in the picture 4.4. We observe for each apple i two weights x_i and y_i and the relation of interest is between the expected values.

$$EY_i = \alpha + \beta EX_i$$

□

We are interested in estimating θ and have already a favorite estimator

$$\widehat{\theta} = T(\mathbf{Y}, \mathbf{V}, \mathbf{U}).$$

In the measurement error model $\mathbf{U}$ is a latent variable, which cannot directly observed. We observe $\mathbf{X} = (X_1, \ldots, X_n)$ with

$$X_i = U_i + \sigma Z_i \;\; i = 1, \ldots, n,$$

where

$$Z_1, \ldots, Z_n \ \ i.i.d. \text{ are the measurement errors, } Var(Z_i) = \sigma^2.$$

In general, the naive use of the "old" estimation rule

$$\widehat{\theta}_{naive} = T(\mathbf{Y}, \mathbf{V}, \mathbf{X})$$

delivers an inconsistent procedure. SIMEX is a general method for correcting the naive estimator. We assume that σ^2 is known.

Algorithm 4.4 SIMEX

1. Simulation: For every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$ generate new errors Z_i^* independent on the data with expectation zero and calculate new samples

$$X_i(\lambda) = X_i + \sqrt{\lambda}\sigma Z_i^*;\ i = 1, \ldots, n, \quad \mathbf{X}(\lambda) = (X_1(\lambda), \ldots, X_n(\lambda)).$$

2. Calculate for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$\widehat{\theta}(\lambda) = T(\mathbf{Y}, \mathbf{V}, \mathbf{X}(\lambda)).$$

3. Extrapolation:
   Fit a curve $\widehat{f}(\lambda)$ such that $\sum_{k=1}^{K} \left(\widehat{f}(\lambda_k) - \widehat{\theta}(\lambda_k)\right)^2$ is minimal.
4. Define

$$\widehat{\theta}_{simex} = \widehat{f}(-1).$$

Note

$$Var(X_i) = \sigma^2 \ \text{ and } \ Var(X_i(\lambda)) = (1+\lambda)\sigma^2$$

such that $\lambda = -1$ respects the case of no measurement error. Obviously we can only generate new data for positive λ, that is why we need the backwards extrapolation step. The knowledge of the measurement error variance is necessary in the algorithm above.

Let us now study the simple linear model with and without measurement errors. Consider the linear simple relationship:

$$y_i = \alpha + \beta\xi_i + \epsilon_i, \quad x_i = \xi_i + \delta_i. \tag{4.10}$$

The $\xi_1, \ldots, \xi_n$ are unknown design points (variables). The first equation is the

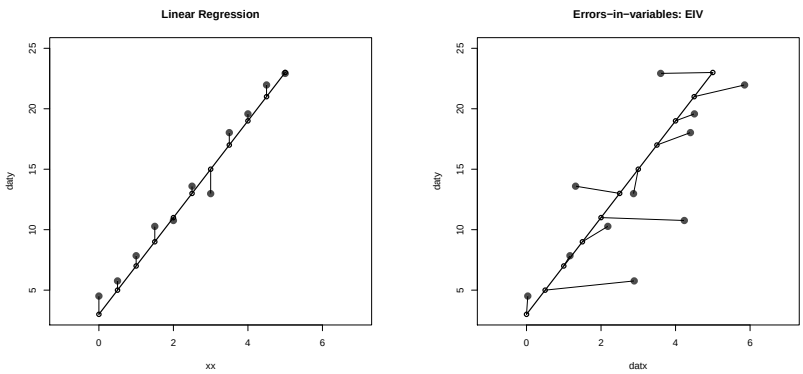


Figure 4.5: The simple linear regression model without errors in variables and with errors in variables.

usual simple linear regression. The second equation is the errors-in-variables equation. Regression models with observed independent variables are also called errors-in-variables models (EIV).
In errors-in-variables models the situation is much more complicated. The observation (x_i, y_i) can deviate in all directions from the point on the regression line $(\xi_i, \alpha + \beta\xi_i)$, compare Figure 4.5.
The least squares estimators in the model with known design points is defined by

$$(\widehat{\alpha}, \widehat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta\xi_i)^2.$$

Introduce the usual denotations $m_{xx},\ m_{xy},\ m_{\xi\xi},\ m_{\xi y}$ by

$$m_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2, \quad m_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

and so on. Then

$$\widehat{\alpha} = \bar{y} - \widehat{\beta}\bar{\xi}, \quad \widehat{\beta} = \frac{m_{y\xi}}{m_{\xi\xi}}.$$

The naive use of this estimator gives

$$\widehat{\alpha}_{naive} = \bar{y} - \widehat{\beta}_{naive}\bar{x}, \quad \widehat{\beta}_{naive} = \frac{m_{xy}}{m_{xx}}.$$

Remind the central limit theorem, we know

$$m_{xy} = \beta m_{\xi\xi} + o_P(1) \text{ and } m_{xx} = m_{\xi\xi} + Var(\delta) + o_P(1).$$

We obtain for the naive estimator

$$\widehat{\beta}_{naive} = \frac{m_{xy}}{m_{xx}} = \frac{\beta m_{\xi\xi}}{m_{\xi\xi} + Var(\delta)} + o_P(1).$$

In case that the measurement error variance $Var(\delta) = \sigma^2$ is known we can correct the estimator

$$\widehat{\beta}_{corr} = \frac{m_{xy}}{m_{xx} - \sigma^2}.$$

Assume that the errors (ϵ_i, δ_i) are i.i.d. from a two dimensional normal distribution with expected values zero, uncorrelated and $Var(\epsilon) = \sigma^2$ and $Var(\delta) = \kappa\sigma^2$, where κ is known. Then the maximum likelihood estimator is the total least squares estimator defined by

$$(\widehat{\alpha}_{tls}, \widehat{\beta}_{tls}) = \arg \min_{\alpha,\beta,\xi_1,\ldots,\xi_n} \left(\sum_{i=1}^{n} (y_i - \alpha - \beta\xi_i)^2 + \kappa \sum_{i=1}^{n} (x_i - \xi_i)^2 \right).$$

In this simple linear EIV model the minimization problem can be solved explicitly. The nuisance parameter can be eliminated by orthogonal projection onto the regression line

$$\widehat{\xi}_i = \arg \min_{\xi_i} \left((y_i - \alpha - \beta\xi_i)^2 + \kappa(x_i - \xi_i)^2 \right),$$

that is

$$\widehat{\xi}_i = \frac{\beta y_i + \kappa x_i}{\kappa + \beta^2}.$$

Thus it remains to solve the minimization problem

$$\min_{\alpha,\beta} \frac{1}{\kappa + \beta^2} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2.$$

We get

$$\widehat{\alpha}_{tls} = \bar{y} - \widehat{\beta}_{tls}\, \bar{x}$$

and

$$\widehat{\beta}_{tls} = \frac{m_{yy} - \kappa m_{xx} + \sqrt{(m_{yy} - \kappa m_{xx})^2 + 4\kappa m_{xy}^2}}{2m_{xy}}. \tag{4.11}$$

for $m_{xy} \neq 0$, otherwise $\widehat{\beta}_{tls} = 0$.

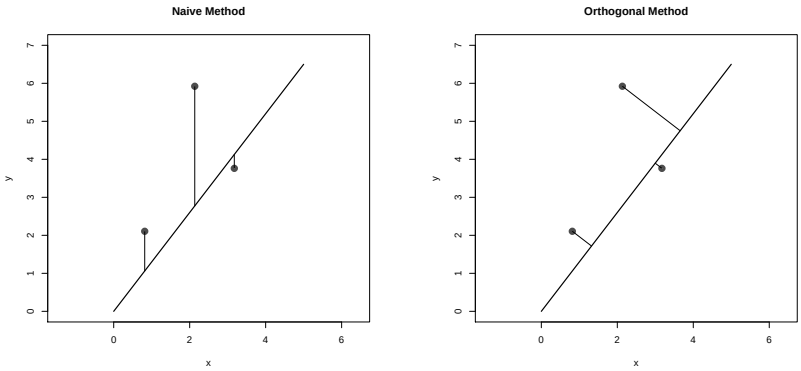


Figure 4.6: The unknown design points are estimated by the observation or estimated by the projection on to the line.

Algorithm 4.5 SIMEX, σ^2 known

1. Simulation: For every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$ generate new errors Z_i^* independently on the data with expectation zero and calculate new samples

$$X_i(\lambda) = X_i + \sqrt{\lambda}\sigma Z_i^*;\ i = 1, \ldots, n, \quad \mathbf{X}(\lambda) = (X_1(\lambda), \ldots, X_n(\lambda)).$$

2. Calculate for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$\widehat{\beta}_{naive}(\lambda) = \frac{m_{x(\lambda)y}}{m_{x(\lambda)x(\lambda)}}.$$

3. Fit a curve

$$(\widehat{a}, \widehat{b}) = \arg\min_{a,b} \sum_{k=1}^{K} \left(\frac{a}{b + \lambda_k} - \widehat{\beta}_{naive}(\lambda_k) \right)^2.$$

4. Define

$$\widehat{\beta}_{simex} = \frac{\widehat{a}}{\widehat{b} - 1}.$$

It can be shown that

$$\widehat{\beta}_{simex} = \widehat{\beta}_{corr} + o_{P^*}(1).$$

The remainder term $o_{P^*}(1)$ converges to zero in probability for K tending to infinity with respect to the pseudo error generating probability P^*.

The model (4.10) is symmetric in both variables, set $EY_i = \eta_i$ then

$$\eta_i = \alpha + \beta\xi_i \quad \xi_i = -\frac{\alpha}{\beta} + \frac{1}{\beta}\eta_i.$$

The first equation is related to a regression of Y on X, the second to an inverse regression of X on Y. In case of errors in variables both regressions are the same. Thus, we have the choice between two different naive estimators

$$\widehat{\beta}_{1,naive} = \frac{m_{xy}}{m_{xx}}, \quad \widehat{\beta}_{2,naive} = \frac{m_{yy}}{m_{xy}}.$$

It holds $\widehat{\beta}_{1,naive} \leq \widehat{\beta}_{tls} \leq \widehat{\beta}_{2,naive}$.

The following SIMEX algorithm explores the symmetry and is called SYMEX. It has the advantages that the knowledge of the measure error variance is not needed, but the quotient κ is supposed to be known. The idea is to apply the first SIMEX steps to each of the naive estimators. The symex estimator is defined by the crossing point of the two extrapolation curves.

Algorithm 4.6 SYMEX

1. Regression Y on X
   (a) Simulation: Generate new samples for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$X_i(\lambda) = X_i + \sqrt{\lambda}\kappa Z^*_{1,i};\ i = 1, \ldots, n \quad \mathbf{X}(\lambda) = (X_1(\lambda), \ldots, X_n(\lambda)).$$

   (b) Calculate for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$\widehat{\beta}_{1,naive}(\lambda) = \frac{m_{x(\lambda)y}}{m_{x(\lambda)x(\lambda)}}.$$

   (c) Fit a curve $b_1(\lambda) = \frac{\widehat{a}}{\widehat{b}+\lambda}$ with

$$(\widehat{a}, \widehat{b}) = \arg\min_{a,b} \sum_{k=1}^{K} \left(\frac{a}{b+\lambda_k} - \widehat{\beta}_{naive}(\lambda_k) \right)^2.$$

2. Regression X on Y
   (a) Simulation: Generate new samples for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$Y_i(\lambda) = Y_i + \sqrt{\lambda}Z^*_{2,i};\ i = 1, \ldots, n \quad \mathbf{Y}(\lambda) = (Y_1(\lambda), \ldots, Y_n(\lambda)).$$

   (b) Calculate for every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$

$$\widehat{\beta}_{2,naive}(\lambda) = \frac{m_{y(\lambda)y(\lambda)}}{m_{y(\lambda)x}}.$$

   (c) Fit a line $b_2(\lambda) = \widehat{a} + \widehat{b}\lambda$ with

$$(\widehat{a}, \widehat{b}) = \arg\min_{a,b} \sum_{k=1}^{K} \left(b + a\lambda_k - \widehat{\beta}_{naive}(\lambda_k) \right)^2.$$

3. Find λ^* such that

$$b_1(\lambda^*) = b_2(\lambda^*).$$

4. Define

$$\widehat{\beta}_{symex} = b_2(\lambda^*).$$

This estimator can also generalized to a multivariate EIV model, for more details see Polzehl and Zwanzig (2013). It holds

$$\widehat{\beta}_{symex} = \widehat{\beta}_{tls} + o_{P^*}(1),$$

where the remainder term $o_{P^*}(1)$ converges to zero in probability for K tending to infinity with respect to the pseudo error generating probability P^*.

The following R code is for a SIMEX algorithm with linear exploration model.

R Code 4.2.2. R-code: SIMEX

```
# data=(daty,datx)
# add pseudo errors with one fixed lamda
#  and calculate the naive estimator
simest<-function(n,lamda,B)
{
    b1<-rep(NA,B)
    for (j in 1:B)
    {
        x<-datx+sqrt(lamda)*rnorm(n,0,1)
        M<-lm(daty~x)
        b1[j]<-coef(M)[2]
    }
    return(b1)
}
simest(11,0.5,10)
mean(simest(11,0.5,10)) # stabilizing
#### SIM - step
l<-c(0.1,0.2,0.3,0.4,0.5,0.6)
K<-length(l)
b<-rep(NA,K)
B<-1
n<-length(daty)
for(j in 1:K)
{
    b[j]<-mean(simest(n,l[j],B))
}
##### EX - step ######
M1<-lm(b~l)
### backwards extrapolation, Var=0.5 #####
simb<-coef(M1)[1]-0.5*coef(M1)[2]
```

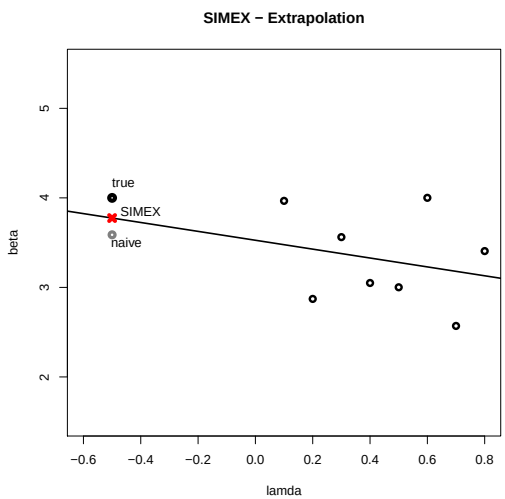


Figure 4.7: Illustration of the SIMEX method with a linear extrapolation model.

4.3 Variable Selection

In this section we consider the problem of variable selection based on simulation. The data set consists of $p+1$ columns of dimension n. Each column includes the observations related to one variable. The first variable we call Y is the response variable (dependent variable) the other columns contain the observations of the independent variables (features, covariates) $X_{(1)}, \ldots, X_{(p)}$.

$$\left(\mathbf{Y}, \mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(p)}\right) = \begin{pmatrix} Y_1 & X_{(1),1} & \cdots & X_{(p),1} \\ \vdots & \vdots & \vdots & \vdots \\ Y_n & X_{(1),n} & \cdots & X_{(p),n} \end{pmatrix} \tag{4.12}$$

The problem is to select the minimal set of covariates $X_{(j_1)}, \ldots, X_{(j_m)}$, which are needed for predicting the values of Y

$$E\left(Y \mid X_{(1)}, \ldots, X_{(p)}\right) = E\left(Y \mid X_{(j_1)}, \ldots, X_{(j_m)}\right). \tag{4.13}$$

We call the variables $X_{(j_1)}, \ldots, X_{(j_m)}$ appearing on the right hand in (4.13) important and the other variables are named unimportant. The set of indices of the important variables $\mathcal{A} = \{j_1, \ldots, j_m\}$ get the name active set.

Example 4.5 (**Diabetes**) This example is quoted from Efron et. al. (2003). 442 diabetes patients were measured on 10 different variables (age, sex, bmi, map, tc, ldl, hdl, tch, ltg, glu) and a response Y a measurement of disease progression. The problem is to find out which of the covariates are important factors for the decease progression. □

R Code 4.3.3. R-code: Example Diabetes

```
library(lars)
data(diabetes)
```

4.3.1 F-Backward and F-Forward procedures

As introduction we briefly repeat classical selection methods in linear regression. Let us suppose that the data follow a linear regression model

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon,\ \mathbf{X} = \left(\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(p)}\right),\ \ rank(\mathbf{X}) = p,\ \ \varepsilon \sim N_n(0, \sigma^2 \mathbf{I}_n). \tag{4.14}$$

Then the model (4.13) corresponds to the hypothesis

$$H_0 : \beta_j = 0, \text{ for all } j \notin \mathcal{A}. \tag{4.15}$$

In linear models (4.14) the main tool for testing is the F-test. The F-statistic

measures the difference in the model fit of a "small" model with the variables $X_{(j)}, j \in J_m$ and a "big" model with the variables $X_{(j)}, j \in J_q$, where $J_m \subset J_q \subseteq \{1, \ldots, p\}$ and $\#(J_m) = m$, $\#(J_q) = q$, $m < q$

$$F\left(J_m, J_q\right) = \frac{q-m}{n-q} \frac{RSS_{J_m} - RSS_{J_q}}{RSS_{J_q}}.$$

Here the model fit is measured by the residual sum of squares (RSS). Let be RSS_J the residual sum of squares in the linear model including all variables $X_{(j)}$ $j \in J \subseteq \{1, \ldots, p\}$

$$RSS_J = \min_{\beta, \beta_j = 0,\ j \notin J} \left\|\mathbf{Y} - \mathbf{X}\beta\right\|^2.$$

Note, the RSS_J criterion is also reasonable, when the linear model assumptions are not fulfilled. Then for $J = \{j_1, \ldots, j_q\}$ the criterion RSS_J is a measure for the best linear approximation of $E(Y \mid X_{(1)}, \ldots, X_{(p)})$ by a linear function of $X_{(j_1)}, \ldots, X_{(j_q)}$ and

$$\begin{aligned} RSS_J \quad = \quad & \min_{\beta, \beta_j = 0, j \notin J} \left\|E(\mathbf{Y} \mid \mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(p)}) - \mathbf{X}\beta\right\|^2 \\ & + tr\left(Cov(Y \mid X_{(1)}, \ldots, X_{(p)})\right) + \ n\, o_p(1) \end{aligned}$$

In (4.14) under the null hypothesis, the small model is true and the F-statistic is F-distributed with $df_1 = n - q$ and $df_2 = q - m$ degrees of freedom. Let be $q(\alpha, df_1, df_2)$ the respective α quantile. Selection algorithms perform combinations of F-tests. As examples the backwards and forward algorithm are presented. In practice methods which combine backward and forward steps are in use.

The backward algorithm starts with the model including all variables of the data set. The first step is a simultaneous test on the p null hypotheses that a component of β in (4.13) is zero. All possible partial F-statistics are compared. One performs the F-test based on the F-statistic with minimal value. No rejection of the null hypothesis means that the respective variable is deleted from the model, otherwise the whole model is accepted. In the next step the model with the deleted variable is the big model and all small models with one additional deleted variable are simultaneously tested.

Algorithm 4.7 F-Backward
Choose α.

1. Start with the model (4.14) which includes all variables $X_{(1)}, \ldots, X_{(p)}$.
   (a) For all $j \in J = \{1, \ldots, p\}$ consider $H_{0,j}: \ \beta_j = 0$ with the partial F-statistic $F_j = F(J \setminus \{j\}, J)$.
   (b) Select l_1 with $F_{l_1} = \min_{j \in J} F_j$.
   (c) Perform the F-test for the hypothesis H_{0,l_1}:
   If $F_{l_1} < q(\alpha, 1, n-p)$ then delete the variable $X_{(l_1)}$ and go to the next step.
   Otherwise stop, select the model with the variables $X_{(1)}, \ldots, X_{(p)}$.
2. Assume the model with the variables $X_{(j)}, \ j \in J_1 = \{1, \ldots, p\} \setminus \{l_1\}$.
   (a) For all $j \in J_1$ calculate the F-statistics $F_j = F(J_1 \setminus \{j\}, J_1)$.
   (b) Select l_2 with $F_{l_2} = \min_{j \in J_1} F_j$.
   (c) If $F_{l_2} < q(\alpha, 1, n-p+1)$ then delete the variable $X_{(l_2)}$ and go to the next step.
   Otherwise stop, select the model with the variables $X_{(j)}, j \in J_1 = \{1, \ldots, p\} \setminus \{l_1\}$.
3. ...
4. Last step: Assume the model with the variable $X_{(l_p)}, \ \{l_p\} = \{1, \ldots, p\} \setminus \{l_1, \ldots, l_{p-1}\}$
   (a) Consider $H_{0,l_p}: \ \beta_{l_p} = 0$ and $F_{l_p} = F(\{\}, \{l_p\})$
   (b) If $F_{l_p} < q(\alpha, 1, n)$ then delete the variable $X_{(l_p)}$ and no variable is selected.
   Otherwise stop, select the model with the variable $X_{(l_p)}$.

The F-forward selection algorithm is a stepwise regression starting with all possible simple linear regressions. In every step new models with one added variable are compared by the partial F-statistic. For the model with the maximal F-statistic the partial F-test at level α is carried out. In case of rejecting the null hypothesis this variable with the maximal F-statistic is added to the model. Otherwise, all null hypotheses are not rejected and no more variable is added and the procedure stops.

Note, forward algorithms have the advantage that they can be applied for big data sets, where $p > n$.

Algorithm 4.8 F-Forward
Choose α.

1. Suppose p different models with variable $X_{(j)}$, $j \in J = \{1, \ldots, p\}$.
   (a) In each model calculate the F-statistics $F_j = F(\{\}, \{j\})$ for testing $H_{0,j} : \beta_j = 0$.
   (b) Select j_1 with $F_{j_1} = \max_j F_j$.
   (c) For $F_{j_1} > q(\alpha, 1, n-1)$ select the variable $X_{(j_1)}$ and go to Step 2. Otherwise stop, no variable is selected.
2. Suppose $p-1$ different models with the variables $X_{(j_1)}, X_{(j)}$ $j \in J_1 = \{1, \ldots, p\} \setminus \{j_1\}$
   (a) Calculate the F-statistics $F_{(j_1,j)} = F(\{j_1\}, \{j, j_1\})$ for testing $H_{0,j} : \beta_j = 0$
   (b) Select j_2 with $F_{(j_1,j_2)} = \max_j F_{(j_1,j)}$.
   (c) For $F_{j_1,j_2} > q(\alpha, 1, n-2)$ select variable $X_{(j_2)}$, and go to Step 3. Otherwise stop, the final model contains $X_{(j_1)}$.
3. Suppose $p-2$ different models with the variables $X_{(j_1)}, X_{(j_2)}, X_j$, $j \in J_2 = J_1 \setminus \{j_2\}$ ...
4. ...

The tuning parameter α is the size of every partial F-test. But the final model bases on several tests, and α is not any more the size of the whole procedure. The size of the whole procedure is called family size and defined by

$$\alpha_{family} = P(\text{the true model is not selected}).$$

In case of the backward algorithm, a bound of the real family size can be calculated. Suppose the chosen model contains q variables, then the decisions $D_1, \ldots, D_{p-q+1}$ are made, where only the last decision is a rejection of the partial null hypothesis. Using

$$\begin{aligned} & P_0(D_1 \ldots D_{p-q+1}) \\ = \; & P_0(D_{p-q+1} \mid D_1 \ldots D_{p-q}) P_0(D_{p-q} \mid D_1 \ldots D_{p-q-1}) \cdots P_0(D_1) \end{aligned}$$

we have

$$\begin{aligned} & P(\text{model is selected} \mid \text{model is true}) \\ = \; & P_0\left(\beta_j = 0, j \in J \; \mathcal{A} \text{ is not rejected, } \beta_{l_{q+1}} = 0 \text{ is rejected}\right) \\ = \; & (1-\alpha)^{p-q-1}\alpha. \end{aligned}$$

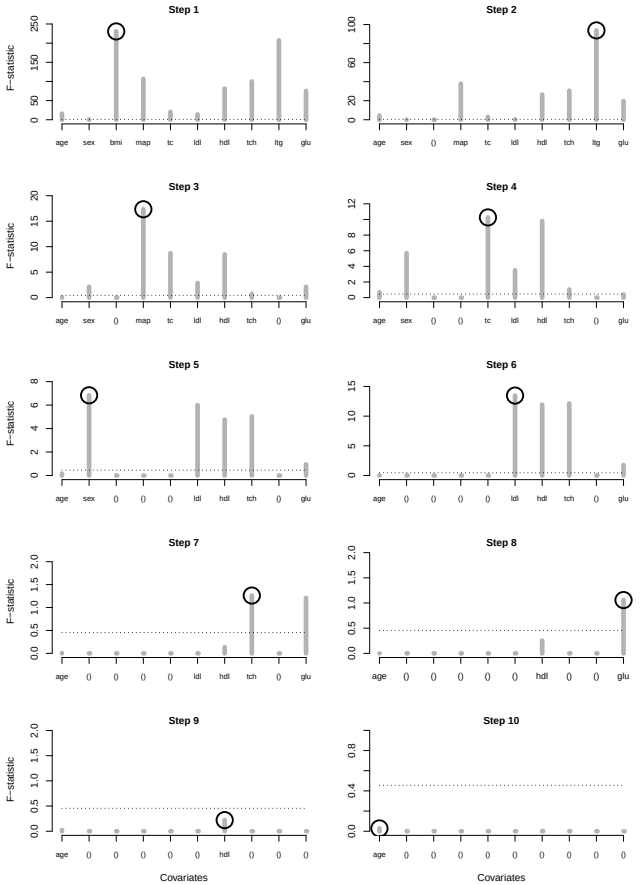
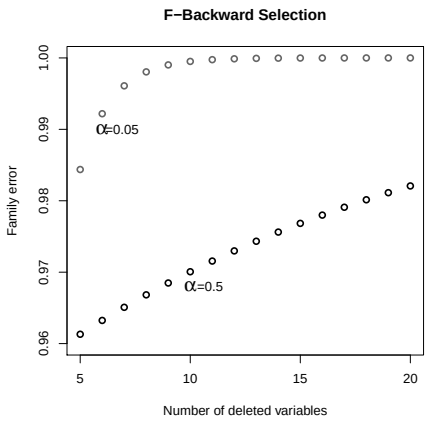


Figure 4.8: All steps of the F-forward algorithm in Example 4.5. The broken line is the $\alpha = 0.5$ quantile. The algorithm stops after Step 8. The selected variables are bmi, ltg, map, tc, sex, ldl, tch, glu.

The main problem is now to determine the tuning parameter α which control the threshold of the model fit criterion. The F-statistic as measure of the influence of covariates is interesting also in case of nonlinear models. Then it compares the best linear approximations. If the condition on the error distribution is violated, the F-statistics are not F-distributed. The comparison of the F-statistic with a bound $q(\alpha, df_1, df_2)$ controlled by the tuning parameter α is still reasonable. The bound is now simply a threshold and not any more the quantile of the F distribution F_{df_1,df_2} and the tuning parameter α is not the size of the partial test.

In R, variable selection procedures are implemented, where the models are compared by Akaike information criterion AIC or the Bayesian information



criterion BIC. For a linear model with normal errors, unknown error variance and included variables $X_{(j)}, j \in J = \{j_1, \ldots, j_q\}$ the AIC criterion is defined as

$$AIC_J = n \ln(RSS_J) + 2q$$

and the BIC

$$BIC_J = n \ln(RSS_J) + \ln n \, q.$$

Note, some times the criteria differ by an additive const. Another alternative is Mallows C_p

$$C_{p,J} = \frac{RSS_J}{RSS_{\{1,\ldots,p\}}} \frac{n-p}{1} + 2q.$$

R Code 4.3.4. R-code: Variable selection

```
library(lars)
data(diabetes)
library(leaps) # subset selection
all=regsubsets(y~x,data=diabetes)
## stepwise adding variables can be done as follows ##
# step1
lm0<-lm(y~1)
A1<-add1(lm0,~1+x[,1]+x[,2]+x[,3]+x[,4]+x[,5]+
              x[,6]+x[,7]+x[,8]+x[,9]+x[,10],test="F")
F1<-A1$F[2:11] #partial F statistic for every new added variable
AIC.1<-A1$AIC[2:11] # AIC for every new added variable
A1$AIC[1] # minimal AIC of the step before
max(F1)
which.max(F1) # here at bmi= x[,3]
# step2
```

```
lm1<-lm(y~1+x[,3],data=diabetes)
A2<-add1(lm1,~x[,1]+x[,2]+x[,3]+x[,4]+x[,5]+
              x[,6]+x[,7]+x[,8]+x[,9]+x[,10],test="F")
```

The AIC-forward algorithm is a stepwise minimization algorithm of the AIC criterion. The model with the smallest AIC is taken for the next step. The procedure stops, when no smaller AIC can be reached.

Algorithm 4.9 AIC-Forward

1. Let be $J_q = \{j_1, \ldots, j_q\}$ the current set of selected variables and $AIC_{(q)} = AIC_{J_q}$
2. For all $j \in \{1, \ldots, p\} \setminus J_q$ calculate the AIC-statistics
$$AIC_{J_q \cup j} = n \ln(RSS_{J_q \cup j}) + 2n(q+1).$$
3. Select j_{q+1} with
$$AIC_{J_q \cup j_{q+1}} = \min_j AIC_{J_q \cup j}.$$
and set $AIC_{(q+1)} = AIC_{J_q \cup j_{q+1}}$
4. For $AIC_{(q+1)} < AIC_{(q)}$ select the variable $X_{(j_{q+1})}$ and update $J_{q+1} = J_q \cup \{j_{q+1}\}$, otherwise stop.

Both forward algorithms are greedy algorithms, because in every step the next step is optimized. In every step, we compare models with the same number of variables and

$$\begin{aligned} RSS_1 < RSS_2 &\Leftrightarrow AIC_1 < AIC_2 \\ &\Leftrightarrow BIC_1 < BIC_2 \\ &\Leftrightarrow F_2 < F_1. \end{aligned}$$

Thus, there is no difference in selecting the next variable between F-forward, AIC-forward or BIC-forward algorithm. The procedures differ in their stopping rules. In case that the number q_{true} of variables in the true model is known, and it is only unknown which of the p variables in the data set should be included, all methods stop after selecting q_{true}variables and deliver the same final model. These assumption that the active set $\mathcal{A}$ contains not more than q_{true} elements is called sparsity assumption.
Note the equivalences work for all criteria, which is a monotone function of RSS, especially for criteria which penalize the RSS by an additive constant depending on the number of model parameters. The equivalence does not hold for a criterion of following type

$$\min_{\beta, \beta_j = 0, j \notin J} (\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda pen(\beta, q)),$$

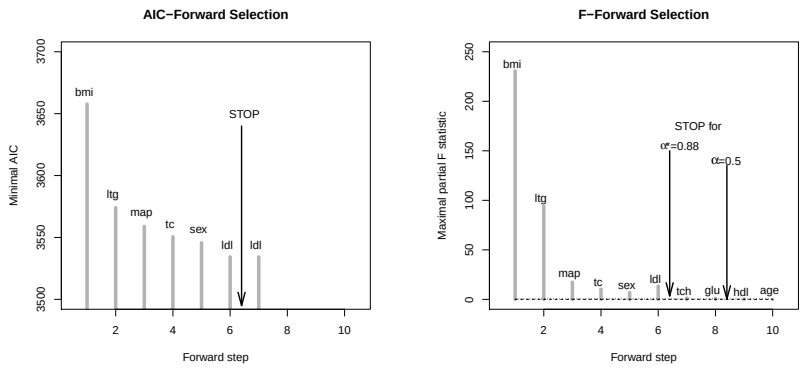


Figure 4.9: Example 4.5. The AIC-forward algorithm stops after the 6th step. The selected variables are bmi, ltg, map, tc, sex, ldl. Using the FSR method with α^*, the selected variables are bmi, ltg, map, tc, sex, ldl. In this example both methods deliver the same final model.

where the penalty depends on the unknown parameter β.

4.3.2 FSR-Forward procedure

The next procedure applies data augmenting for determining the stopping condition. In Miller's textbook, he proposed the generation of pseudo variables for calibrating forward selection procedures Miller (2002). Instead of the original data (4.12), extended data sets are introduced. Let

$$\left(\mathbf{Y}, \mathbf{X}_{(1)}, ..., \mathbf{X}_{(p)}, \mathbf{Z}_{(1)}, ..., \mathbf{Z}_{(K)}\right) =$$

$$\begin{pmatrix} Y_1 & X_{(1),1} & \cdots & X_{(p),1} & Z_{(1),1} & \cdots & Z_{(K),1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_n & X_{(1),n} & \cdots & X_{(p),1} & Z_{(1),n} & \cdots & Z_{(K),n} \end{pmatrix},$$

where the observations $Z_{(j),i}, j = 1, \ldots, K \quad , i = 1, \ldots, n$ are generated such that the generated variables $Z_{(1)}, \ldots, Z_{(K)}$ are unimportant by construction. For example, generate $Z_{(j),i}$ i.i. standard normal distributed. In the literature, they are called phony variables, control variables, or pseudo variables. The arguments are as follows.

A procedure which selects many of phony variables will select many unimportant variables. Otherwise, a procedure which never accepts a phony variable, has the big risk to miss an important variable in (4.12).

Wu developed an algorithm, in his dissertation, and call it FSR-algorithm [REF]. FSR stands for false selecting rate. Running a selection procedure on the data $(\mathbf{Y}, \mathbf{X})$, let $U(\mathbf{Y}, \mathbf{X})$ be the number of unimportant selected variables and $S(\mathbf{Y}, \mathbf{X})$ the number of all selected variables, then

$$FSR = \frac{U(\mathbf{Y}, \mathbf{X})}{S(\mathbf{Y}, \mathbf{X}) + 1}.$$

It is not possible to count $U(\mathbf{Y}, \mathbf{X})$, but the number of selected phony variables is known. Wu's main idea is to determine the size α of the partial F-tests in the F-forward procedure by checking how many of the phony variables are selected.

Algorithm 4.10 FSR-Forward
Given a bound γ_0 on the rate of selected phone variables.

1. For every $\alpha \in \{\alpha_1, \ldots, \alpha_M\}$
   (a) For every b, $b = 1, \ldots, B$
      i. Generate $\mathbf{Z}_b = \left(\mathbf{Z}_{(\mathbf{1})}, \ldots, \mathbf{Z}_{(\mathbf{K})}\right)$ independent of $(\mathbf{Y}, \mathbf{X})$.
      ii. Run a F-forward selection with α on the data $(\mathbf{Y}, \mathbf{X}, \mathbf{Z}_b)$ and count $U_b^*(\alpha) = \#\{k : \mathbf{Z}_{(\mathbf{k})} \text{ selected}\}$ and count $S_b(\alpha)$ the number of all selected variables.
   (b) Calculate the averages

$$\overline{U}^*(\alpha) = \frac{1}{B}\sum_{j=1}^{B} U_b^*(\alpha), \quad \overline{S}(\alpha) = \frac{1}{B}\sum_{j=1}^{B} S_b(\alpha)$$

   and as an indicator for the FSR

$$\gamma(\alpha) = \frac{\overline{U}^*(\alpha)}{\overline{S}(\alpha) + 1}.$$

2. Determine the adjusted α level

$$\alpha^* = \sup_{\alpha \in \{\alpha_1, \ldots, \alpha_M\}} \{\alpha : \gamma(\alpha) \leq \gamma_0\}.$$

3. Run an F-forward selection on the original data $(\mathbf{Y}, \mathbf{X})$ with α^*.

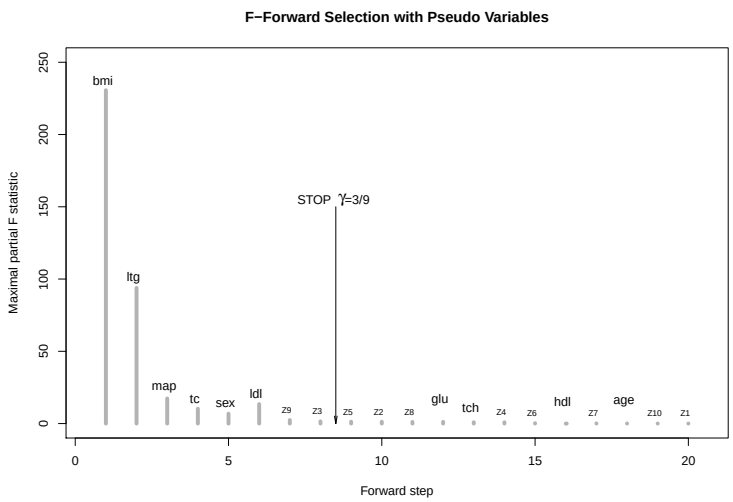


Figure 4.10: All possible steps of an F-Forward algorithm with 10 pseudo variables.

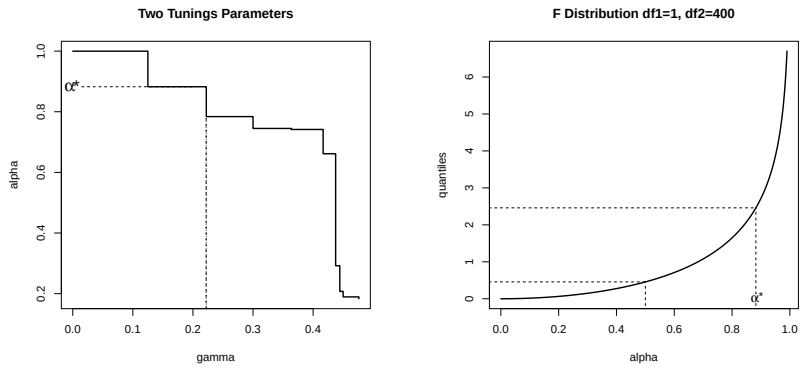


Figure 4.11: Determining the α^* of the F-forward algorithm by the FSR estimate from the extended data set.

4.3.3 SimSel

SimSel applies both the data augmentation of Wu's FSR algorithm and the successive data pertubation of SIMEX. The main idea behind SimSel is to determine the relevance of a variable $\mathbf{X}_{(j)}$ by successively disturbing it and study the effect on the residual sum of squares (RSS). In case that the RSS remains unchanged, we conclude that the variable $\mathbf{X}_{(j)}$ is not important.
The SimSel algorithm borrows the simulation step, where pseudo errors are

added to the independent variables from the SIMEX method. However, the extrapolation step in SIMEX is not performed. Thus, it is called SimSel for *simulation* and *selection*. The variables $\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(p)}$ are checked each after the other. Let us assume that we are interested in $\mathbf{X}_{(1)}$. The original data set

$$\left(\mathbf{Y}, \mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(p)}\right)$$

is embedded into

$$\left(\mathbf{Y}, \mathbf{X}_{(1)} + \sqrt{\lambda}\varepsilon^*, \ldots, \mathbf{X}_{(\mathbf{p})}, \mathbf{Z}\right)$$

where $\mathbf{Z} = (Z_1, ..., Z_n)^T$ is an independent **pseudo variable**, independently generated from $\mathbf{Y}, \mathbf{X}_1, ..., \mathbf{X}_p$, the pseudo errors are generated such that $\varepsilon^* = (\varepsilon_1^*, ..., \varepsilon_n^*)^T$, ε_i^* are i.i.d P^*, with $E(\varepsilon_i^*) = 0$, $Var\left(\varepsilon_i^*\right) = 1$, $E\left(\varepsilon_i^*\right)^4 = \mu$
The phony variable $\mathbf{Z}$ serves as an untreated control group in a biological experiment. The influence of the pseudo errors is controlled by stepwise increasing λ. The main idea is, if λ "does not matter", then $\mathbf{X}_{(\mathbf{1})}$ is unimportant. For explaining this see Figure 4.12. There, we consider the simple linear case $\mathbf{Y} = \beta_1 \mathbf{X}_1 + \varepsilon$ and we fit

$$RSS_1\left(\lambda_k\right) = \min_{\beta_1, \beta_2} \left\| \mathbf{Y} - \beta_1 \left(\mathbf{X}_1 + \sqrt{\lambda_k}\varepsilon^*\right) - \beta_2 \mathbf{Z} \right\|^2 .$$

and

$$RSS_2\left(\lambda_k\right) = \min_{\beta_1, \beta_2} \left\| \mathbf{Y} - \beta_1 \mathbf{X}_1 - \beta_2 \left(\mathbf{Z} + \sqrt{\lambda_k}\varepsilon^*\right) \right\|^2$$

by a linear regression. Intuitively "does not matter" respects to a constant trend of $RSS\left(.\right)$.

The theoretical background to the SimSel algorithm is given by the following result:

Theorem 4.3 *Under the assumption that* $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ *exists, it holds*

$$\frac{1}{n} RSS\left(\lambda\right) = \frac{1}{n} RSS + \frac{\lambda}{1 + h_{11}\lambda} \left(\widehat{\beta}_1\right)^2 + o_{P^*}(1)$$

where h_{11} *is the* $(1,1)-$*element of* $\left(\frac{1}{n}\mathbf{X}^T\mathbf{X}\right)^{-1}$ *and* $\widehat{\beta}_1$ *is the first component of the OLSE estimator* $\widehat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$.

Line of the proof: It holds

$$\frac{1}{n} RSS\left(\lambda\right) = \frac{1}{n}\mathbf{Y}^T\mathbf{Y} - \frac{1}{n}\mathbf{y}^T P(\lambda)\mathbf{Y} \tag{4.16}$$

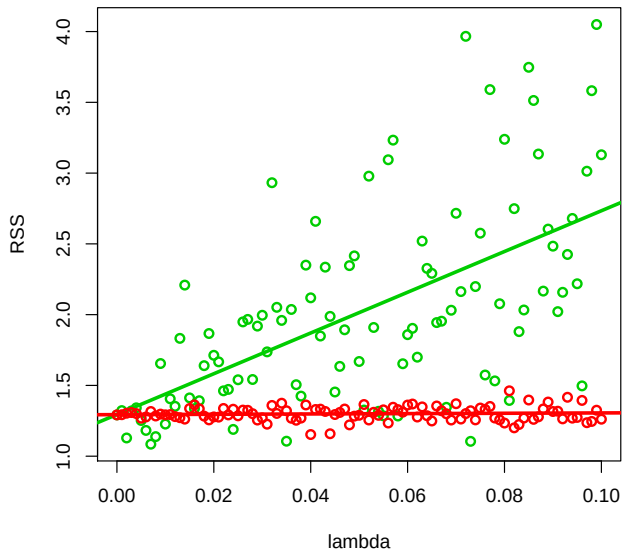


Figure 4.12: The constant regression respects to the unimportance of the pseudo variable. The heteroscedastic regression is related to the worse model fit under the disturbed important variable.

with

$$P(\lambda) = \mathbf{X}(\lambda)\left(\mathbf{X}(\lambda)^T\mathbf{X}(\lambda)\right)^{-1}\mathbf{X}(\lambda)^T. \tag{4.17}$$

Further

$$\frac{1}{n}\mathbf{X}(\lambda)^T\mathbf{Y} = \left(\frac{1}{n}\mathbf{X} + \frac{1}{n}\sqrt{\lambda}\Delta\right)^T\mathbf{y},$$

where Δ is the $(n \times p)-$ matrix

$$\Delta = \begin{pmatrix} \varepsilon_1^* & 0 & \cdots & 0 \\ \varepsilon_2^* & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \vdots \\ \varepsilon_{n-1}^* & 0 & \cdots & \vdots \\ \varepsilon_n^* & 0 & \cdots & 0 \end{pmatrix}.$$

The law of large number applied to the pseudo errors delivers

$$\frac{1}{n}\mathbf{X}(\lambda)^T\mathbf{Y} = \frac{1}{n}\mathbf{X}^T\mathbf{Y} + o_{P^*}(1). \tag{4.18}$$

Consider now $\mathbf{X}(\lambda)^T\mathbf{X}(\lambda)$:

$$\frac{1}{n}\left(\mathbf{X}+\sqrt{\lambda}\Delta\right)^T\left(\mathbf{X}+\sqrt{\lambda}\Delta\right) \tag{4.19}$$

$$= \frac{1}{n}\mathbf{X}^T\mathbf{X}+\frac{1}{n}\sqrt{\lambda}\mathbf{X}^T\Delta + \frac{1}{n}\sqrt{\lambda}\Delta^T\mathbf{X}+\frac{1}{n}\lambda\Delta^T\Delta \tag{4.20}$$

Hence

$$\left(\frac{1}{n}\mathbf{X}(\lambda)^T\mathbf{X}(\lambda)\right)^{-1} = \left(\frac{1}{n}\mathbf{X}^T\mathbf{X} + \lambda\mathbf{e}_1\mathbf{e}_1^T\right)^{-1} + o_{P^*}(1).$$

□

Note, in the procedure we approximate

$$\frac{\lambda}{1+h_{11}\lambda} \approx \lambda.$$

In linear errors-in-variable models the naive LSE is inconsistent. But if β_1 is zero, then naive LSE also converges to zero. This gives the motivation for successful application of SimSel to errors-in-variables models.

The significance of the pertubation on the model fit is controlled by a Monte Carlo F-test for the regression of $RSS(\lambda)$ on λ. That is the testing step in the SimSel procedure. The comparison of the model fit is repeated M times and a paired sample of F- statistics are obtained, where sample $F_{i,1}, ..., F_{i,M}$ related to the variable under control $\mathbf{X}_{(i)}$, the other sample $F_{p+1,1}, ..., F_{p+1,M}$ related to the pseudo variable $\mathbf{Z} = \mathbf{X}_{(p+1)}$. Kernel estimates $\widehat{f}_i$, $\widehat{f}_{p+1}$ for each sample are calculated and the overlapping of the densities are compared with given tuning parameters α_1, α_2, see Figure 4.13.

Example 4.6 (**Prostate**) The prostate cancer dataset contains 97 observations of one dependent variable (the log of the level of prostate-specific antigen, lpsa) and eights independent variables; the logarithm of the cancer volume (lcavol), the logarithm of the prostate's weight (lweight), age, the logarithm of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), the logarithm of the level of capsular penetration (lcp), Gleason score (gleason), and percentage Gleason scores 4 and 5 (pgg45). For more details is the R package ElemStatLearn, data(prostate). □

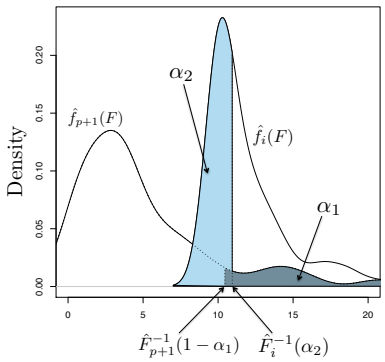


Figure 4.13: Significance for chosen levels $\alpha_1,\ \alpha_2$.

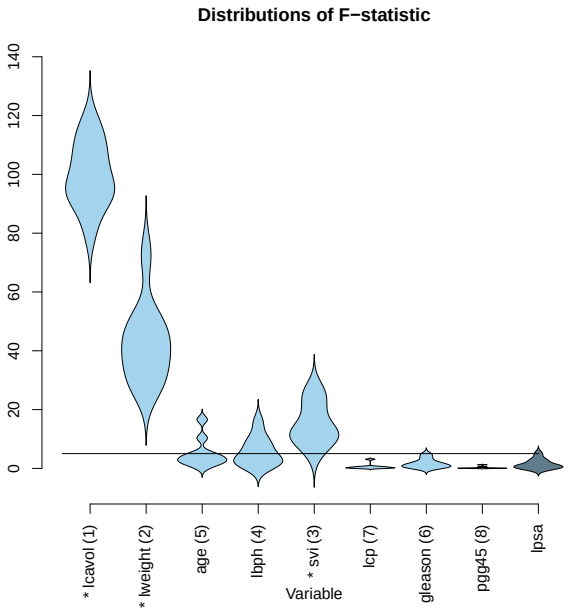


Figure 4.14: Graphical output of SimSel Example 4.6.

Algorithm 4.11 The SimSel - Algorithm
Given $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_K, M, \alpha_1, \alpha_2$ For m from 1 to M
Generate unimportant pseudo variables $\mathbf{Z} = \mathbf{X}_{p+1}$.
For i from 1 to $p+1$
For k from 1 to K
Generate pseudo errors for each $\mathbf{Z} = \mathbf{X}_{p+1}$ to $\mathbf{X}_i$ and add them.
Compute $RSS_i(\lambda_k)$.
Regression step. Calculate the F-statistics $F_{i,m}$.
Plotting step, violin plot of the F-statistics.
Ranking step, according to the sample median of $(F_{i,m})_{m=1...M}$.
Testing step.

For more details regarding this chapter, see Dempster (1977), Cook and L.A. (1994), Polzehl and Zwanzig (2003), Eklund and Zwanzig (2011), Wei and Tanner (1990), Fuller (1987), and Wu et al. (2007) .

Bibliography

J.R. Cook and Stefanski L.A. Simulation-Extrapolation Estimation in parametric measurement error models. *JASA*, 89:1314–1327, 1994.

Laird N.M. Rubin D.B. Dempster, A.P. Maximum likelihood from incomplete data via EM algorithm. (with discussion). *J. Roy. Stat. Soc. B*, 39:1–38, 1977.

Martin Eklund and Silvelyn Zwanzig. Simsel: a new simulation method for variablen selection. *J Comp and Sim*, 85(515-527), 2011.

Wayne A. Fuller. *Measurement Errors Models*. Wiley, 1987.

Alan Miller. *Subset Selection in Regression*. Chapman and Hall CRC, 2002.

Jörg Polzehl and Silvelyn Zwanzig. On a symmetrized simulation extrapolation estimator in linear errors-in-variables models. *ComputationaL Statistics and Data Analysis*, 47(675-688), 2003.

Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley-Interscience, 2004.

Gerg c G Wei and Martin A. Tanner. A monte carlo implementation of the em algorithm and the poor manÔÇÖs data augmentation algorithms. *Journal of the American Statistical Association*, 85(411), 1990.

Yujun Wu, Dennis D Boos, and Leonard A Stefanski. Controlling variable selection by the addition of pseudovariables. *Journal of the American Statistical Association.*, 102, 2007.