TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Diplomarbeit

Outlier Detection in Predictive Time Series Models

Ausgeführt am Institut für

Stochastik und Wirtschaftsmathematik

der Technischen Universität Wien

unter Anleitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

durch

Nermina Mumic
Josefstraße 100/42
3100 St. Pölten

Datum

Unterschrift (Student)

Danksagung

An dieser Stelle möchte ich mich bei all jenen bedanken, die mich im Zuge meiner Masterarbeit unterstützt haben. Besonderer Dank gilt meinem Professor Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser, welcher meine Arbei und somit auch mich betreut hat. Du standest mir fachlich und inhaltlich stets zur Seite und hast mir viel Deiner wertvollen Zeit geschenkt. Dadurch hast Du einen wesentlichen Beitrag zur Enstehung dieser Arbeit geleistet. Ich danke Dir.

Ich möchte diese Gelegenheit auch nützen meiner Familie zu danken. Danke für eure Geduld und eure bedingungslose Unterstützung. Insbesondere möchte ich meinen Schwestern danken, ihr wart immer mein moralischer Rückhalt.

Zuguterletzt möchte ich mich bei meinen Kollegen für die gemeinsame Zeit und Freundschaft bedanken.

Abstract

In this master thesis we want to identify outliers in time series concerning sales data by a two-step procedure. In the first step we extract the signal of the underlying series with the methods ARIMA modeling and Kalman filtering. In the second step we want to explain the remaining structure within the residuals of step one. For this purpose we compress the underlying information of financial indicators into a few latent components. For identification of the latent variables we apply Principal Component Regression, Partial Least Squares Regression and Sparse Partial Least Squares Regression. Concerning the last two methods we will also present robust approaches using Robust M-Regression. The optimal number of components is determined by Cross Validation (CV) and repeated CV. Within a regression model the scaled residuals of step two are regressed on the latent variables of step two.

For outlier detection the scaled regression residuals are monitored by means of tolerance bands, calculated with prediction errors from the Cross Validation. Observations beyond these bands are identified as outliers.

Zusammenfassung

In dieser Master Arbeit stellen wir ein zweistufiges Verfahren zur Ausreißererkennung in Absatzzeitreihen vor. Im ersten Schritt extrahieren wir das Signal der Zeitreihe mit Hilfe von ARIMA Modellen und Kalman Filter. Die verbleibende Struktur in den Residuen analysieren wir im zweiten Schritt mit Hilfe von Finanzindikatoren. Dazu schätzen wir latente Variablen, welche die komprimierte Information aus den Finanzindikatoren enthalten. Die dafür verwendeten Verfahren sind Haupt-komponentenregression, Partial Least Squares Regression und Sparse Partial Least Squares Regression, wobei bei für die letzten beiden auch robuste Varianten, basie-rend auf Robuster M-Regression, verwendet werden. Die optimale Anzahl an Komponenten wird mittels Cross Validation (CV) beziehungsweise wiederholter CV ermittelt. In einem Regressionsmodell werden die skalierten Residuen aus Schritt eins auf die latenten Variablen regressiert.

Zur Erkennung von Ausreißern werden die skalierten Residuen der Regression mit ihren entsprechenden Toleranzbändern dargestellt. Beobachtungen, die außerhalb dieser Toleranzbänder liegen, werden als Ausreißer klassifiziert.

Contents

1 Introduction

The investigations of this master thesis are motivated by the problem of a company that is globally conducting its business in the industry. Time series describing sales data of certain products should be analysed on structural breaks or rather structural irregularities. The aim is to provide a model that is capable of classifying incoming sales observations as regular points or outliers.

The main idea is to find a model that extracts the signal of the underlying time series as accurate as possible. If we have a model that is able to capture the structure correctly, it can be applied for predictive purposes and outliers would then indicate abnormalities. Within the scope of finding an accurate model we follow a stepwise approach.

First we give a short overview in Chapter 2 of the data that is used for the analysis. The time series we analyse describes the sales of advance good producers in Germany and is provided by the German Central Bank. Furthermore, we will make use of economic indicators that are mostly provided by national banks or private institutions.

Chapter 3 deals with the first step, where we try to extract a signal from the time series by means of its own history. For this purpose we consider two different models, the ARIMA model from the Box Jenkins framework and a state space model. ARIMA models are regarded as standard models for time series modeling due to their simple structure. Their major weakness are the strong assumptions that are made. More precisely, stationarity is required, or at least stationarity of differenced time series, which is hardly ever given from an empirical point of view. An assumption violation could affect the results considerably. In contrast thereto we also take the approach of state space modeling, where we work with Kalman filtering. State space models have the advantage of being very general, and a wide class of problems, including ARIMA models, can be formulated as special cases.

Considering the residuals of both approaches will show that they still contain some structure, and only performing the methods of Chapter 3 is not sufficient. For this reason, we induce a second modeling step in Chapter 4 and try to explain the remaining structure by means of exogenous variables. For this purpose, it seems reasonable to use financial indices, as we assume that business figures are among others influenced by movements of the global market. These movements are displayed by means of financial indicators, for this analysis overall 232 indices are used. We will face the issue that they poorly contain information, as they are more or less linear combination of each other. For this reason, we try to compress the underlying information into a couple of latent variables that supply the most relevant information. For identification of the latent variables we will make use of Principal Components Regression (PCR), Partial Least Squares Regression (PLSR) and Sparse Partial Least Squares Regression (SPLSR). Concerning PLSR and SPLSR we will also present robust approaches. The optimal number of latent variables is determined by Cross Validation (CV) and repeated Double Cross Validation (rDCV). Then the identified latent variables are used as exogenous variables for the regression, where the structure of the residuals from step 1 is tried to be explained.
The residuals of the regression are then monitored by means of confidence bands, com-

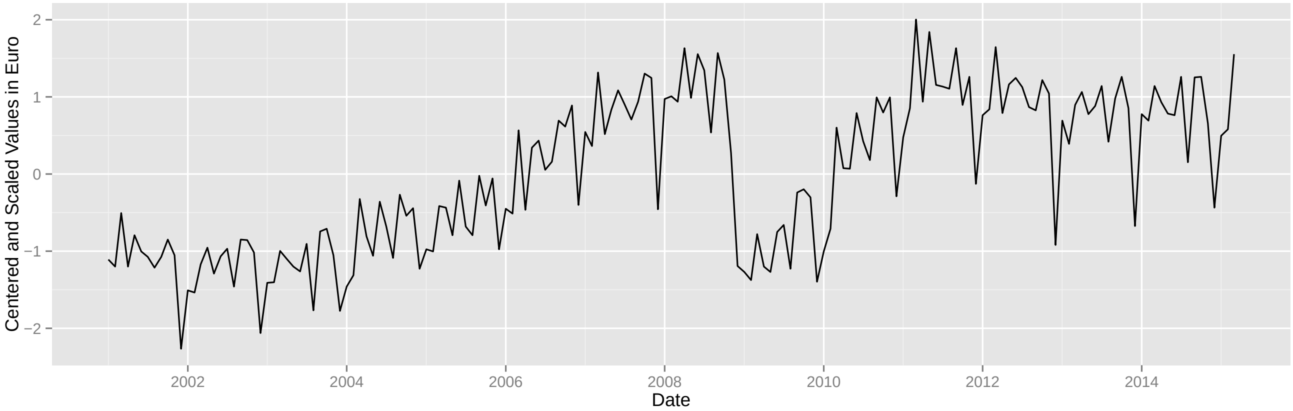
puted with the standard error obtained from the CV training data.

Within the practical part in Chapter 5 we apply the methods presented in Chapter 4 in R. We provide plots of the fitted values coming from these models and a monitoring plot, with the appropriate residuals and confidence levels of $\pm 2\sigma$ or $\pm 3\sigma$, which enable outlier detection. Values that are beyond these bands are classified as outliers.
Chapter 6 provides some summarizing conclusions.

In the Appendix we list a summary of the R code used in Chapter 5 for implementing the routines of Chapter 4, but also the routines for Chapter 4 are presented.

2 Description of Variables

Within the scope of this thesis we analyze a time series provided at the homepage of the German Central Bank. It is an index which describes sales of advance good producers in Germany. The values are listed monthly from January 2001 till August 2014 and are neither trend nor seasonally adjusted. We work with the demeaned and by its sample standard deviation scaled series.



The demeaned and scaled series shows an upwards trend till 2008 and seasonal slumps. The impact of the subprime crisis Shiller (2008) becomes obviously remarkable at the end of 2008, 2009 appears as year of a great recession, from 2010 the sales start rising again. Beginning from 2007, we see very distinctive peaks each year.
Besides the subprime crisis, the dot-com bubble Kindleberger and Aliber (2005) can be taken as benchmark to proof whether the methods recognise these years as outliers.

With regards to the used economic indicators, we use 232 indicator variables overall. Among others we have the Economic Sentiment Indicator, several confidence indicators, producer price indizes, stocks, volatility indices and gross domestic product figures. Many of them are provided by central banks, while other can be bought from different private providers.

3 Signal Extraction

Given an economic time series, we want to extract its signal as accurate as possible. The choice of the model depends on the properties of the underlying series, for example stationarity.Furthermore, we have to distinguish between methods that are capable of modeling exogenous information or not.

As we are following a step-wise approch in signal extraction, we want to focus in this step on extracting the signal from the time series own history. In the next step we will proceed with extracting remaining structure in the residuals by means of exogenous variables.

In this chapter we want to present two widely used methods for modeling univariate time series in terms of past realizations.

3.1 Autoregressive Integrated Moving Average Processes (ARIMA)

The notation and definitions of this chapter are manly based on Box and Jenkins (1976), Madsen (2007) and Scherrer (2014). Many empirical time series do not meet the stationarity property. They either have a local level, trend or seasonalities. Within the scope of ARIMA modeling we consider time series that, except for a trend or local level, behave homogenously over time.
A crucial property is that the d^{th} difference is a stationary mixed auto regressive moving average. In this way Box and Jenkins created a generalization of ARMA models, so that after taking differences we obtain a stationary ARMA process.

For better understanding we want to introduce some concepts that are required for discussing further models:

- **Stationary processes:** A univariate stochastic process $\{Y_t\}$ is weakly stationary, if $\forall t, s \in \mathbb{Z}$ holds
  - $\mathbb{E}Y_t^T Y_t < \infty$
  - $\mathbb{E}Y_t = \mathbb{E}X_s \quad \forall t, s \in \mathbb{Z}$
  - $\mathbb{E}Y_t Y_s^T = \mathbb{E}Y_{t+k} Y_{s+k}^T$ (or equivalently $\text{Cov}(Y_t, Y_s) = \text{Cov}(Y_{t+k}, Y_{s+k})$) $\forall t, s, k \in \mathbb{Z}$

  In the further formulation we will use just "stationary" meaning weakly stationary

- **White noise:** A process $\{\epsilon_t\}$ is called white noise, if
  - $\mu_t = \mathbb{E}[\epsilon_t] = 0$
  - $\sigma_t^2 = \text{Var}[\epsilon_t] = \sigma_\epsilon^2$
  - $\gamma_\epsilon(k) = \text{Cov}[\epsilon_t, \epsilon_{t+k}] = 0 \qquad \text{for} \quad k \neq 0$

  where σ_ϵ^2 is a constant value.

- **Backshift operator:** the backshift operator B is defined by

$$B(Y_t) = Y_{t-1} \quad \forall t > 1$$

- **z-Transform:** for a sequence $\{Y_t\}$ the z-transform is defined as

$$Z(\{Y_t\}) = \sum_{t=-\infty}^{\infty} Y_t z^{-t}$$

and defined for all $z \in \mathbb{C}$ for which this series is convergent.

In the following we want to introduce the basic models on which ARIMA models are based on.

3.1.1 MA(q)

The process $\{Y_t\}$ given by

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} \tag{1}$$

where $\{\epsilon_t\}$ is white noise, is called moving average process of order q denoted by MA(q) and $\theta_1, ..., \theta_q$ are parameters which determine the model.

3.1.2 AR(p)

The process $\{Y_t\}$ given by

$$Y_t + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} = \epsilon_t \tag{2}$$

where $\{\epsilon_t\}$ is white noise, is called autoregressive process of order p denoted by AR(p) and $\phi_1, ..., \phi_p$ are parameters which determine the model.

3.1.3 ARMA(p,q)

The solution $\{Y_t\}$ of the ARMA(p,q) system given by

$$Y_t + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} \tag{3}$$

with $\{\epsilon_t\}$ white noise, is called ARMA(p,q) process.

By means of using the backshift operator B we can use polynomials to write the ARMA(p,q) system in the following form

$$\phi(B) Y_t = \theta(B) \epsilon_t,$$

where ϕ and θ are polynomials of order p and q, respectively. Applying the z-transformation we obtain the notation

$$\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p$$

and

$$\theta(z) = 1 + \theta_1 z + ... + \theta_q z^q.$$

Now we can define following properties for ARMA(p,q) systems:

- **Stationarity condition:**

  $$\phi(z) \neq 0 \quad \forall |z| = 1$$

  The ARMA system has a unique, stationary solution if the stationarity condition is fulfilled. The solution is a regular MA(∞) process

  $$Y_t = \phi^{-1}(B)\theta(B)\epsilon_t = \sum_{j=-\infty}^{\infty} \tilde{\theta}_j \epsilon_{t-j}$$

  where $\tilde{\theta}(B) = \phi^{-1}(B)\theta(B)$ denotes a filter polynomial with coefficients $\tilde{\theta}_j$.

- **Stability condition:**

  $$\det(\phi(z)) \neq 0 \quad \forall |z| \leq 1$$

  If the stability condition is fulfilled, $\{Y_t\}$ is a causal MA(∞) process given by

  $$Y_t = \phi^{-1}(B)\theta(B)\epsilon_t = \sum_{j \geq 0} \tilde{\theta}_j \epsilon_{t-j}.$$

- **Minimum phase condition:**

  $$\theta(z) \neq 0 \quad \forall |z| < 1$$

- **Strict minimum phase assumption:**

  $$\theta(z) \neq 0 \quad \forall |z| \leq 1$$

  Under the strict minimum phase condition we obtain the AR(∞) representation of the process:

  $$\epsilon_t = \sum_{j \geq 0} \tilde{\phi}_j Y_{t-j} = \theta^{-1}(B)\phi(B)Y_t$$

  where $\tilde{\phi}(B) = \theta^{-1}(B)\phi(B)$ denotes a filter polynomial with coefficients $\tilde{\phi}_j$.

- **Co-primeness condition:** $\phi(z), \theta(z)$ are co-prime if and only if they have no common roots.

For every regular ARMA process exists an ARMA system, which fulfills the stability, minimum phase and co-primeness condition. In the scalar case, these conditions are sufficient to define the system uniquely. In the multivariate case further assumptions are required for uniqueness.

3.1.4 ARIMA(p,d,q)

The integrated form of an ARMA process is called ARIMA(p,d,q) process with p representing the order of the AR proccess, q the order of the MA process and d how many differences need to be taken to obtain a stationarity ARMA process.

The process $\{Y_t\}$ is called an integrated autoregressive moving average or ARIMA(p,d,q) process, if it can be written in the form

$$\phi(B)\Delta^d Y_t = \theta(B)\epsilon_t, \qquad (d \in \mathbb{N}) \tag{4}$$

where $\{\epsilon_t\}$ denotes a white noise process, $\phi(z^{-1})$ a polynomial of order p and $\theta(z^{-1})$ a polynomial of order q. Both polynomials have all roots inside the unit circle. Δ^d is the d^{th} difference of a process, where Δ denotes the difference operator, defined as $\Delta Y_t = Y_{t+1} - Y_t$.

From definition (4) we see that the process defined by

$$W_t = \Delta^d Y_t$$

is a stationary and invertible ARMA (p,q) process, where invertibility means that the roots of $\theta(z^{-1})$ lie within the unit circle. Hence we see that Y_t is obtained only by d summations of the stationary and invertible process W_t.This is why we use the notation "integrated". Once we have the differenced stationary process W_t we can proceed with ARMA(p,q) modeling methods.

3.1.5 Parameter Estimation

Here we want to present the maximum likelihood approach for parameter estimation. Concerning the required distributional assumption we take (ϵ_t) as white noise, meaning they are normally distributed with zero mean and $\text{Var}[\epsilon_t] = \sigma_\epsilon^2$. For the ARMA(p,q) system in (3) we introduce the parameter vectors

$$\Theta^T = (\phi_1, ..., \phi_p, \theta_1, ..., \theta_q)$$

and

$$Y_t^T = (Y_t, Y_{t-1}, ..., Y_1).$$

The general formula for the likelihood function for time series data is then given by the joint distribution of all observations given the parameters Θ and σ_ϵ^2:

$$\begin{aligned} L(Y_N; \Theta, \sigma_\epsilon^2) &= f(Y_N|\Theta, \sigma_\epsilon^2) && (5)\\ &= f(Y_N|Y_{N-1}, \Theta, \sigma_\epsilon^2) f(Y_{N-1}|\Theta, \sigma_\epsilon^2) && (6)\\ &= \left(\prod_{t=p+1}^{N} f(Y_t|Y_{t-1}, \Theta, \sigma_\epsilon^2) \right) f(Y_p|\Theta, \sigma_\epsilon^2) && (7) \end{aligned}$$

Here N denotes the number of observations utilized and f the density function of Y_t (see Madsen (2007)). From this expression we can derive the conditional likelihood function, conditioned on Y_p:

$$\begin{aligned} L(Y_N; \Theta, \sigma_\epsilon^2) &= \prod_{t=p+1}^{N} f(Y_t|Y_{t-1}, \Theta, \sigma_\epsilon^2) && (8)\\ &= (\sigma_\epsilon^2 2\pi)^{-\frac{N-p}{2}} \exp\left(-\frac{1}{2\sigma_\epsilon^2} \sum_{t=p+1}^{N} \epsilon_t^2(\Theta) \right) && (9) \end{aligned}$$

Taking the logarithms, differentiating the log-likelihood with respect to σ_ϵ^2, and putting it equal to zero we see that the ML estimate $\hat{\theta}$ for Θ is obtained by minimizing

$$S(\Theta) = \sum_{t=p+1}^{N} \epsilon_t^2(\Theta), \qquad (10)$$

and the ML estimate for σ_ϵ^2 is obtained by

$$\hat{\sigma}_\epsilon^2 = \frac{S(\hat{\Theta})}{N-p}. \qquad (11)$$

3.1.6 Order Estimation

The order p and q (and d) has to be estimated in a way that makes the model fit the data as accurate as possible but avoiding overfit. Most commonly used tools for determining the order are:

- **Akaike's Information Criteria (AIC):** The usual form is

  $$AIC = -2\text{log (max. likelihood)} + 2n$$

  with n determining the number of estimated parameters. For ARMA(p,q) models this results in

  $$AIC = N\text{log}\hat{\sigma}_\epsilon^2 + 2(p+q) \qquad (12)$$

  where N denotes the number of used observations and $\hat{\sigma}_\epsilon^2 = \frac{S(\hat{\Theta})}{N}$, see (11). p and q are chosen in a way to minimize (12), but AIC basically tends to allow for too many parameters.

- **Bayesian Information Criterion (BIC):** This criterion is more restrictive with the number of parameters:

$$BIC = N\log\hat{\sigma}_{\epsilon}^{2} + (p+q)\log N \tag{13}$$

The order d for ARIMA (p,d,q) models is taken as the smallest number of differences required for obtaining a stationary time series. This means that after each differencing a test on stationarity is applied, e.g. KPSS test Maddala and Kim (1998).

3.1.7 Plots of ARIMA Fitted Values and Residuals

The R function auto.arima from package forecast Hyndman et al. (2015) gives an ARIMA (2,1,2) model

Coefficients	**AR1**	**AR2**	**MA1**	**MA2**
Estimate	-0.6656	0.3325	0.2219	-0.7383
Standard Error	0.1192	0.1199	0.0768	0.0793

with $\hat{\sigma}_{\epsilon}^{2} = 0.2783$ and Log Likelihood$= -132.51$, AIC$= 276.28$ and BIC $= 291.96$

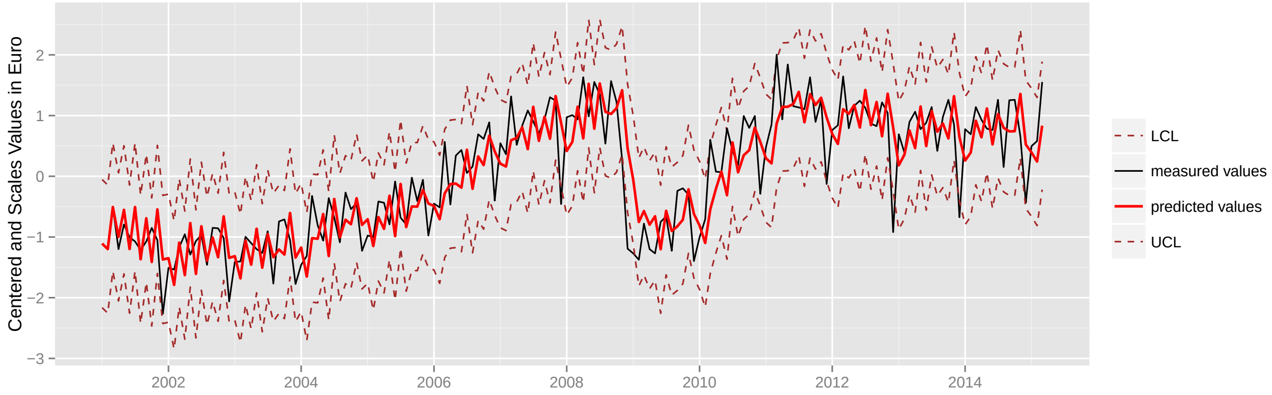


Figure 1: ARIMA fitted values and confidence levels for industrial revenues.

with LCL and UCL denoting the lower and upper confidence level respecitvely of the estimated values, see Figure 1. We see that the estimated values are generally following the trend, but with a slight shift. Furthermore, the confidence bands are quite wide, due to a big standard error. This means the model is not very precise.

The scaled residuals $\tilde{r}_i = \frac{r_i}{\hat{\sigma}_{\epsilon}}$, where r_i are the residuals of the ARIMA(2,1,2) model, get small due to the big standard error. Therefore only very extreme outliers will appear as such, see Figure 2. As a result, we will recognize only very large outliers, which ought to display a potential crisis. But still we clearly see that the subprime crisis reaches the German industry in 2009.

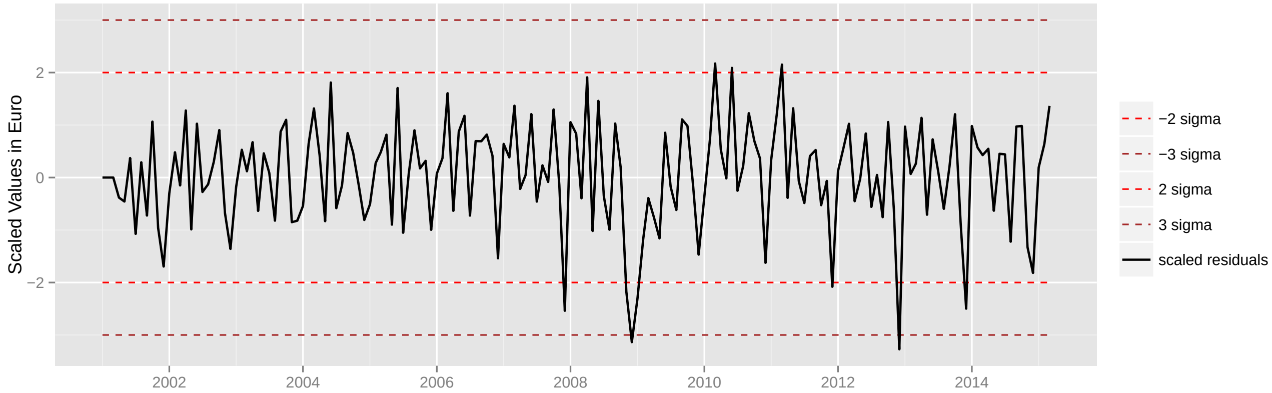


Figure 2: Scaled ARIMA residuals with 2- or 3- $\tilde{\sigma}_\epsilon$ confidence levels

The *sigma* in Figure 2 denotes $\tilde{\sigma}_\epsilon$ which is the standard deviation of $\tilde{r}_i$. As $\tilde{r}_i$ is scaled we have $\tilde{\sigma}_\epsilon = 1$. The confidence levels are constructed as the intervals $[-2\tilde{\sigma}_\epsilon, 2\tilde{\sigma}_\epsilon]$ and $[-3\tilde{\sigma}_\epsilon, 3\tilde{\sigma}_\epsilon]$ respectively. Values of $\tilde{r}_i$ that are beyond these intervals, significantly distinguish from 0 and can be considered as outliers.

3.2 State Space Models

The structure and notation in this chapter refer to Shumway and Stoffer (2011), Kalman (1960) and Durbin and Koopman (2001). State space models go back to the groundbreaking results of Kalman's paper in 1960, where he formulates the Wiener problem, a special case of a Gaussian process, from a state point of view. This enables a very general treatment of a wide class of problems. Therefore many known models, like ARIMA for instance, can be considered as special cases of state space models and be put into this form.

3.2.1 Model Formulation

State space models or dynamic linear models (DLM) consist of two equations

- **State equation:**
  Basically, the state equation characterizes the underlying process the examined time series is following. This means it determines the way the value today depends on its past. In its basic form we assume a vector autoregressive process of order one.

  $$x_t = \Phi x_{t-1} + w_t \tag{14}$$

  x_t and $x_{t-1} \in \mathbb{R}^{px1}$ represent the state vectors with time points $t \in \{1, ..., n\}$. $\Phi \in \mathbb{R}^{pxp}$ is the coefficient matrix determined by the underlying stochastic process. We assume $w_t \in \mathbb{R}^{px1}$ to be independant and identically distributed noise components with zero mean and covariance $Q \in \mathbb{R}^{pxp}$, meaning $w_t \overset{i.i.d}{\sim} N(0, Q)$. As starting values we take

  $$x_0 \overset{i.i.d}{\sim} N(\mu_0, \Sigma_0)$$

- **Observation equation:**
  The introduction of an observation equation contains a new modeling aspect: as measurements are mostly afflected with errors, we cannot observe the process directly. Therefore we take the approach of observing a linear transformation of it with an additional noise component:

  $$y_t = A_t x_t + \nu_t \tag{15}$$

  Here the observed data is denoted with $y_t \in \mathbb{R}^{q\text{x}1}$, where q can either be smaller, equal to or larger than p, depending on the problem. $A_t \in \mathbb{R}^{q\text{x}p}$ denotes the measurement or observation matrix and the error term ν_t is white noise (see Chapter 3.1) with covariance $R \in \mathbb{R}^{q\text{x}q}$.

For simplicity we assume v_t and w_t to be uncorrelated, but this assumption is not necessary. Furthermore we can also include exogenous information into modeling the equations, resulting in

$$x_t = \Phi x_{t-1} + \Upsilon u_t + w_t \tag{16}$$

and

$$y_t = A_t x_t + \Gamma u_t + \nu_t \tag{17}$$

for input vectors $u_t \in \mathbb{R}^{r\text{x}1}$ and appropriate coefficient matrices $\Upsilon \in \mathbb{R}^{p\text{x}r}$ and $\Gamma \in \mathbb{R}^{q\text{x}r}$. Here we will proceed with the formulation (14) and (15), as exogenous variables will be modelled as part of Chapter 4.

3.2.2 Filtering

From a practical point of view our aim is to find estimates for the unobserved signal x_t given the data $Y_s = \{y_1, ..., y_s\}$. Depending on s we can distuingish between the following cases:

- $s < t$: prediction
- $s = t$: filtering
- $s > t$: smoothing

Within the scope of our analysis we want to focus on filtering. In this case all measurements till time point t are provided by the information set Y_s and we have to reconstruct the signal x_t at this point. For this purpose we want to introduce the following notations

$$x_t^s = \mathbb{E}[x_t | Y_s] \tag{18}$$

and

$$P_{t_1,t_2}^s = \mathbb{E}[(x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)^T]. \tag{19}$$

As we assume the noise to be Gaussian, we get very nice properties for the expressions above. The expectation in (18) can be considered as the projection operator rather than an expectation and (19) as the corresponding mean-squared error, see Shumway and Stoffer (2011). We also see that the conditional error covariance

$$P^s_{t_1,t_2} = \mathbb{E}[(x_{t_1} - x^s_{t_1})(x_{t_2} - x^s_{t_2})^T|Y_s]$$

equals (19). This is due to the orthogonal projection in (18) that causes the difference $(x_t - x^s_t)$ to be orthogonal on the Y_s plane and therefore independent (due to normality). In the same way we obtain that the conditional distribution of $(x_t - x^s_t)$ given Y_s equals the unconditional distribution of $(x_t - x^s_t)$, see Shumway and Stoffer (2011).

The Kalman filter is used for filtering and forecasting purposes. It is called filter, as x_t can be written as filter of the observations $y_1, ..., y_t$, namely

$$x_t = \sum_{s=1}^{t} B_s y_s$$

with adequate filter coefficient matrices $B_s \in \mathbb{R}^{pxq}$. The filter is considered as very powerful, as it shows how the filter x^{t-1}_{t-1} has to be updated to x^t_t when the new observation y_t is obtained, without reprocessing the whole data $\{y_1, ...y_t\}$. This recursive nature enables simple implementation of the Kalman filter on a computer.

Properties of the Kalman filter

For the state space model with equations (14) and (15) and initial conditions $x^0_0 = \mu_0$ and $P^0_0 = \Sigma_0$ for $t = 1, ..., n$ we have the following prediction equations,

$$x^{t-1}_t = \Phi^{t-1}_{t-1} + \Upsilon u_t \tag{20}$$

and

$$P^{t-1}_t = \Phi^{t-1}_{t-1} \Phi^T + Q \tag{21}$$

with the required filtering equations

$$x^t_t = x^{t-1}_t + K_t(y_t - A_t x^{t-1}_t - \Gamma u_t) \tag{22}$$

and

$$P^t_t = [I - K_t A_t] P^{t-1}_t \tag{23}$$

where

$$K_t = P^{t-1}_t A^T_t [A_t P^{t-1}_t A^T_t + R]^{-1} \tag{24}$$

denotes the Kalman gain, which determines how much information is provided by the new observation y_t.

3.2.3 Maximum Likelihood Estimation

As the parameters specifying the state space model (14) and (15) are unknown, they have to be estimated given the observations $y_1, ..., y_s$. We collect the unknown parameters, namely the initial mean and covariance μ_0 and Σ_0, the transition matrix Φ, the covariance matrices Q and R, and the coefficient matrices Υ and Γ, into the parameter vector $\Theta = \{\mu_0, \Sigma_0, \Phi, Q, R, \Upsilon, \Gamma\}$. The coefficient matrix A_t in (15) has to be determined by the user, as we are specifying the observation system. A simple choice would be taking the identity.

Under the normality assumption of the initial state vector $x_0 \sim N(\mu_0, \Sigma_0)$ and the errors $\{w_1, ..., w_n\}$ and $\{\nu_1, ..., \nu_n\}$ to be uncorrelated we use maximum likelihood for parameter estimation. Therefore we define the innovations of the process $\{\epsilon_1, ..., \epsilon_n\}$ as follows,

$$\epsilon_t = y_t - A_t x_t^{t-1} - \Gamma u_t,$$

which are independent Gaussian vectors with zero mean and covariances denoted by $\Sigma_t = A_t P_t^{t-1} A_t^T + R$. Except for a scaling constant we obtain the maximum likelihood function depending on the parameter vector Θ:

$$-\ln L_Y(\Theta) = \frac{1}{2}\sum_{t=1}^{n} \log|\Sigma_t(\Theta)| + \frac{1}{2}\sum_{t=1}^{n} \epsilon_t(\Theta)^T \Sigma_t(\Theta)^{-1} \epsilon_t(\Theta) \tag{25}$$

As this equation is non-linear and a complicated function of the unknown parameters, minimizing the negative log-likelihood can be very challenging. The common procedure for solving the problem would be to initialize x_0 and develop a set of recursions for the likelihood function. Then the parameters can be updated successively by using a Newton-Raphson algorithm. For further details we refer to Shumway and Stoffer (2011).

3.2.4 Plots of Kalman Fitted Values and Residuals

The Kalman filter can be computed in R with the package dlm Petris (2014). We assumed the model to follow a random walk model, this means setting Φ and A_t constantly equal to 1 in equations (14) and (15). We get following estimates:

Parameter	$\boldsymbol{\Phi}$	$\boldsymbol{A}$	**Q**	**R**	$\boldsymbol{x_0}$	$\boldsymbol{\Sigma_0}$
Estimate	1	1	0.0481	0.1938	0	10000000

Application of the Kalman filter on our data leads to the following result presented in Figure 3.

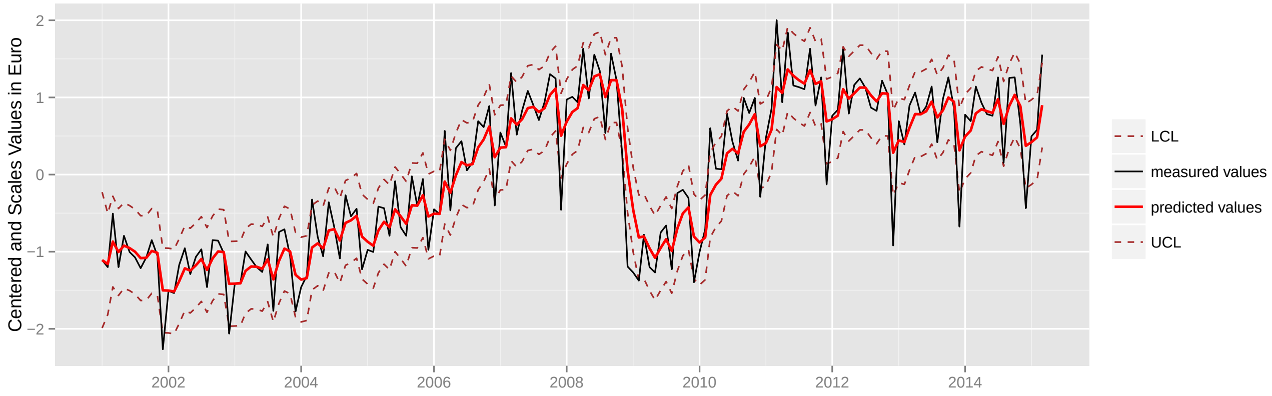


Figure 3: Kalman filter and confidence levels for industrial revenues

with LCL and UCL denoting the lower and upper confidence level, respecitvely, of the filter values. We see that the filter nicely captures the trend of the series, but still having difficulties with large outliers. in contrast to the ARIMA approach the confidence bands are narrower now, meaning we have a smaller standard error. Fur this purpose we have a look again at the scaled residuals in Figure 4.

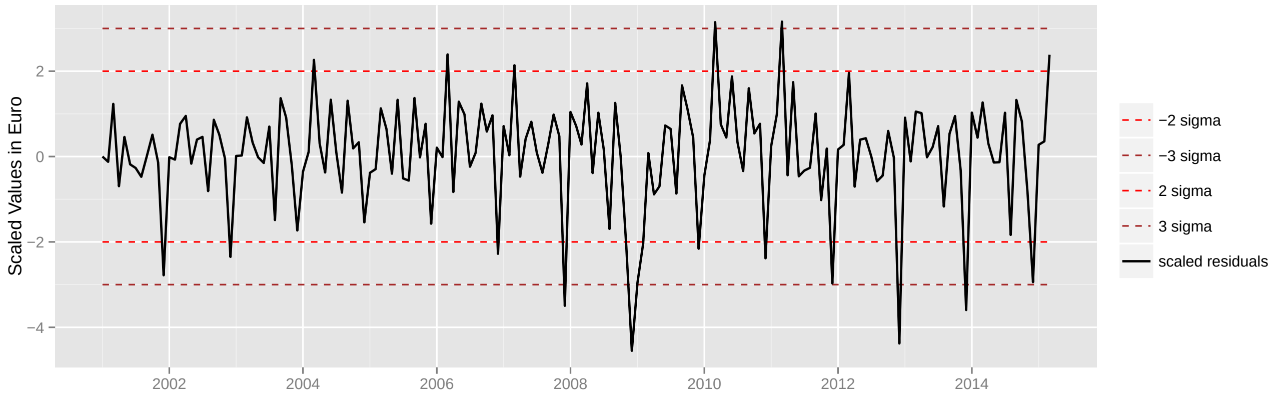


Figure 4: scaled residuals of the Kalman filter with 2- or 3- σ confidence levels

The standard deviation taken for scaling the Kalman residuals is derived from (19). As σ is the standard deviation of the scaled residuals it is equal to 1. We clearly recognise the effects of the subprime crisis in the German industry in 2009. Although our model in Figure 3 recognises the downwards swing, it is not capable of capturing its effects entirely, which leads to an overproportional outlier after all. Furthermore we see extreme outliers in 2010, 2011 and 2013 which might result from the subprime crisis. In constrast to Figure 2 we seem to have more extreme values, but this is due to the smaller standard error. This fact makes the plot more ”sensitive” towards outliers.

3.3 Comparison between ARIMA and State Space Models

The motivation for state space models was actually out of research concerning aerospace. Later on it found wide application in other fields like economics and soil sciences. Main contributions in application of state space methods in economics came from Durbin and Koopman Durbin and Koopman (2001). They saw huge advantages in the state space model: This approach is based on a structural analysis of the problem, which distinguishes between the dynamics of the process, which is determined by trend, seasonal components and exogenous information, on the one side and on the other side, side effects and features of the process in particular situations, which have to be determined by the investigator. In contrast thereto Box Jenkins methods appear not to capture the deeper structure of the data generating process.

Furthermore, state space models are very flexible. Due to their Markovian property problems can be solved recursively, which facilitates the implementation on a computer and makes them very flexible towards structural changes over time. This enables handling of very large problems. They capture a wide class of problems, including ARIMA models, due to their generality. Furthermore, state space models are capable of handling missing values and forecasting does not require a seperate theory, in contrast to Box Jenkins models. On the contrary, Box Jenkins models are homogenous over time, due to the stationarity assumption (for the differenced time series). But exactly the stationarity assumption is the weakness of the Box Jenkins framework. With regards to economic and social time series, empirically observed, hardly any series meets this assumption, no matter how many differences are taken.

In 2001, Durbin and Koopman stated: ”In our opinion, the only disadvantages are the relative lack in the statistical and econometric communities of information, knowledge, and software regarding these models.”

4 Residual Analysis

This chapter features the second step of our precedure. As the residuals of the ARIMA and State Space approach still contain some structure, we try to explain or rather eliminate it by means of a second modeling step. The hypothesis is, that this structure is driven by some exogenous variables that were not modeled in the first step, as the first step only makes use of the time series own history. More precisely, we assume that fluctuations of the financial market have significant influence on our sales variables. The movement of the market can be described via financial market indicators, among others macroeconomic key figures. We will face the issue, that these indicators are highly correlatet as they are mostly linear combinations of each other. For this reason, even a wealth of indicators still contains very little information. Therefore we firstly have to compress the information of all provided financial indicators and then regress the scaled residuals of the sales variables on these latent variables in order to explain their influence on the sales.

In the next sections we will discuss some multivariate techniques that are typical representatives for solving these issues. In case of principal component regression and (robust) partial least squares we refer to Varmuza and Filzmoser (2009) and Serneels et al. (2005).

4.1 Principal Component Regression

Principal Component Regression (PCR) is a method which enables reducing the number of regressor variables and removing multicollinearity Varmuza and Filzmoser (2009). In contrast to variable selection, we no more have the original regressor variables but linear combinations thereof. These linear combinations are obtained via Principal Components Analysis (PCA), therefore PCR can be considered as a combination of PCA and multiple linear regression.

4.1.1 Principal Component Analysis

PCA is a frequently used tool for dimension reduction by computation of latent variables. Basically, it is a method that computes a new orthogonal coordinate system, which consists of these latent variables. The transformation is done with the requests that the new axes represent the directions with highest variance, therefore the first principal component represent the direction with the largest variance contained in the original data.

As we are interested to find these directions with highest variance explained, PCA can be formulated as a maximization problem with constraints. Since the principal components can then be written as linear combinations of the variables $(x_1, ..., x_k)$ we have

$$t_i = x_1\gamma_{1i} + \ldots + x_k\gamma_{ki}, \tag{26}$$

where the unknown coefficients correspond to the loadings vector $\gamma_i = (\gamma_{1i}, ..., \gamma_{ki})^T$, and the t_i's represent the so called scores. At this point we also want to mention that PCA is typically applied to centered and scaled data.

As t_i should have maximum variance, we can formulate the following maximization problem:

$$\max\ Var(t_i) \qquad s.t. \qquad \gamma_i^T \gamma_i = 1 \tag{27}$$

Rearranging the variance expression of the scores leads to

$$Var(t_i) = Var(x_1\gamma_{1i} + \ldots + x_k\gamma_{ki}) = \gamma_i^T Cov(x_1, ..., x_k)\gamma_i = \gamma_i^T \Sigma \gamma_i. \tag{28}$$

In this way we obtain for the Langrangian form for the maximization problem in (27):

$$L(\gamma_i, \lambda_i) = \gamma_i^T \Sigma \gamma_i - \lambda_i(\gamma_i^T \gamma_i - 1) \qquad \text{for} \quad i = 1, ..., k \tag{29}$$

The solution for the Langrangian problem is obtained by taking the first derivative with respect to γ_i and setting it equal to zero. This leads to an equation also known as the eigenvalue problem

$$\Sigma\gamma_i = \lambda_i \gamma_i \qquad \text{for} \quad i = 1, ..., m \tag{30}$$

The γ_i's are the eigenvectors of Σ, representing the loadings, and the λ_i's are the corresponding eigenvalues, which represent the variances of the principal components. As the eigenvalues are ordered decreasingly, the corresponding variances of the principal components are decreasing

$$Var(t_i) = \gamma_i^T \Sigma \gamma_i = \gamma_i^T \lambda_i \gamma_i = \lambda_i \tag{31}$$

In this way we obtain the solution of the Langrangian problem by computing the eigenvectors and the corresponding eigenvalues.

PCA Sample Version

For a concrete sample we have the data matrix $X \in \mathbb{R}^{nxk}$ with the sample mean vector $\hat{\mu}$ and sample covariance $\hat{\Sigma}$. The principal components are obtained with

$$T = (X - 1\hat{\mu}^T)\Gamma,$$

where 1 denotes a vector of one's, Γ denotes the loadings matrix, which contains the eigenvectors of $\hat{\Sigma}$, and T is the scores matrix, which contains the principal components.

4.1.2 Multiple Linear Regression

We have seen that PCA decomposes any centered data matrix X into scores T and loadings Γ, taking only a certain number of components, usually less than $rk(X)$, where $rk(\cdot)$ denotes the rank of the matrix. Thus we have

$$T = X\Gamma + E \tag{32}$$

where the scores represent the l latent variables with most important information ($l < k$). As we want to do regression with the principal components, we first consider a standard linear regression model

$$y = Xb + u, \tag{33}$$

where y is the endogenous variable, in our case the scaled ARIMA/Kalman residuals, b denotes the regression coefficient and u the error term. According to the decomposition in (32) we can replace X with the first PCA scores and therefore only take the most relevant information for regression

$$y = Xb + u = (T\Gamma^T)\tilde{b} + u_T = T\alpha + u_T \tag{34}$$

where α represents the new regression coefficients and u_T a new error term. Now we can apply OLS regression in order to obtain estimators for these coefficients,

$$\hat{\alpha} = (T^T T)^{-1} T^T y, \tag{35}$$

and the final regression coefficients for model (33) are

$$\hat{b}_{PCR} = \Gamma\alpha. \tag{36}$$

This procedure solves the issue of high collinearity as the major information of the data is compressed to a few orthogonal scores. This makes the OLS estimator in (35) numerically stable.

4.2 Partial Least Squares Regression

While principal component regression only considers information of the X variables, partial least squares (PLS) regression includes information about the response variables Y in the course of computing the scores T that should be related to the scores of Y Varmuza and Filzmoser (2009). This means that the mostly applied criterion for computing the latent variables of PLS is to maximize the covariance between the scores in the X and Y-space. Thereby we combine high variance of the scores in the x-space, which leads to stability of the model, and high correlation with the response variable of interest, which enhances its modeling. Therefore PLS can be considered as a compromise between PCR and OLS. This means that PLS basically summarizes highly correlated predictor variables to a set of latent variables, which are uncorrelated, contain maximal variance in the x-space and have maximal covariance to the dependent variables. Then the dependent variable is regressed on these latent variables Serneels et al. (2005).

In this way, PLS can also be written as maximization problem, where the objective function is the covariance between the X and Y-scores subject to the constraint that the scores have to be orthogonal. We consider the data matrix $X \in \mathbb{R}^{nxk}$ and the response matrix $Y \in \mathbb{R}^{nxq}$, both are mean centered.
Actually we are interested in finding a linear relation between X and Y, using regression coefficients B (in the case of univariate y this simplyfies to (33)). But instead of finding this relation directly, we take the approach of modeling X via latent variables,

$$X = T\Gamma^T + E_X, \tag{37}$$

respectively

$$Y = U\Theta^T + E_Y. \tag{38}$$

The scores of X and Y are then connected via the following relationship,

$$U = T\Delta + H, \tag{39}$$

where T and U are the score matrices which give good information summaries of X and Y, respectively. Γ and Θ are the loadings matrices, which have l colums where $l \leq min(k, q, n)$. The number of colums represents the number of considered PLS components. Δ is a diagonal matrix with elements $\delta_1, ..., \delta_l$, and H is the residual matrix with colums h_j, where $j = 1, ..., l$. In the case we have a univariate response, no scores can be computed in the y-space, therefore Equation (39) reduces to

$$y = Td + h, \tag{40}$$

with the coefficients d and the error term h. The goal of PLS is to maximize the covariance between X and Y scores.

Hence we can formulate following maximization problem

$$\max_{\text{a,c}} \quad \text{Cov}(Xa, Yc) \tag{41}$$

under the constraints

$$\|t\| = \|Xa\| = 1 \quad \text{and} \quad \|u\| = \|Yc\| = 1. \tag{42}$$

The solution gives us the wanted loading vectors a and c. Concerning the estimation of the covariance there are multiple approaches possible. In the classical case, we take the sample covariance $\frac{1}{n-1}t^T u$ but also robust estimators can be used. This would lead to a robust PLS, which will be discussed later. The constraints are required in order to provide a unique solution. Within the scope of solving the maximization problem, we consider the covariance of the scores. As we use the sample covariance we can write

$$t^T u = (Xa)^T Yc = a^T X^T Yc \quad \rightarrow \quad \max \tag{43}$$

under the constraints in (42). The vectors a and c can be computed via multiple algorithms. One possible approach is using singular value decomposition for finding the optimal a and c. In that case the solutions correspond to the largest singular values of $X^T Y$. Other widely used algorithms are the Kernel Algorithm, NIPALS, SIMPLS, O-PLS which also provide orthogonal scores, while the eigenvector method results in orthogonal loadings. For further details we refer to Varmuza and Filzmoser (2009).

The solution of (41) delivers the first scores t_1 and u_1 of the x-space and y-space, respectively. For computing further scores we have to introduce further constraints with regard to the scores to provide orthogonality,

$$t_i^T t_j = 0 \qquad \text{and} \qquad u_i^T u_j = 0 \qquad \text{for} \quad 1 \leq i \leq j \leq l. \tag{44}$$

In this way each additional score covers new variability of the data and improves explanatory power of the model.

4.3 Robust Methods

Working with empirical data, we often face data points that do not follow the main structure of the data, so called outliers. With regard to our problem, where we try to model structure in economic time series by means of financial indices and thereby find time points whith anomalous structure, we actually assume the existence of outliers and try to model them. Therefore it might appear obvious to work with robust methods. In this chapter we want to give a short overview about robust regression methods and apply them in the respective robustified versions of PCR and PLS, based on the results of Hastie et al. (2008), Varmuza and Filzmoser (2009), Serneels et al. (2005), Li et al. (2004) and Scherrer (2012).

4.3.1 Robust Regression

We have the following standard assumptions for the classical linear regression model

$$y = X\beta + u.$$

1. $X \in \mathbb{R}^{nxk}$ is deterministic
2. rk(X)=k
3. $\mathbb{E}[u] = 0 \in R^n$
4. $\mathrm{Var}[u] = \sigma^2 I_n, \sigma^2 \geq 0$
5. every $\beta \in \mathbb{R}^k$ is a priori feasible

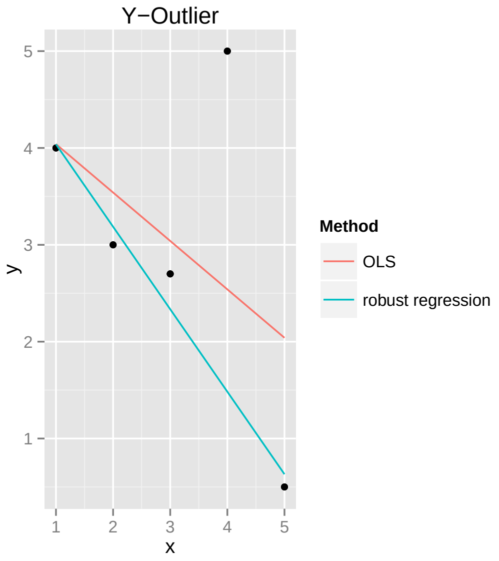
If these assumptions hold, the OLS estimator

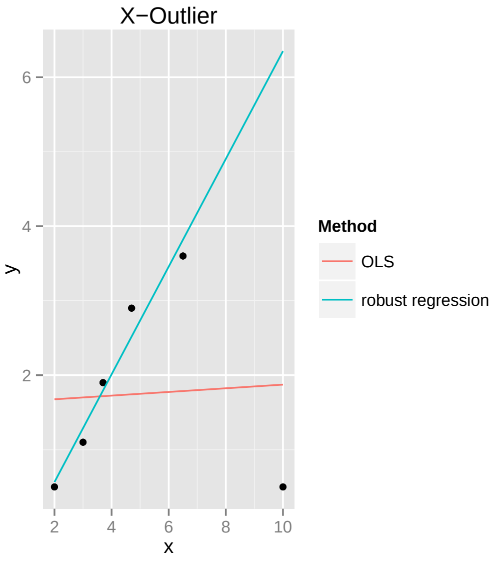
$$\hat{\beta}_{OLS} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 \tag{45}$$

is the best linear unbiased estimator according to Gauss Markov Theorem. This means that $\hat{\beta}_{OLS}$ has the least variance under all linear, unbiased estimators. Here,
$y = (y_1, ..., y_n)^T$, and $x_i \in \mathbb{R}^k$ form the rows of X.

Outliers in the data set are a typical issue that leads to assumption violation. We can distinguish between the following types of outliers:

- *x-outliers*: data points that are outliers in the x-space
- or *y-outliers*: data points that are within the range of the x-variables, but are outliers in the y-space





We see that a single outlier can have such a strong effect on OLS estimation, that it becomes more or less useless. In this case even diagostic plots are unreliable. In particular we face this issue with x-outliers, so called leverage points.

In other words, outliers lead to violations of the classical assumptions and therefore $\hat{\beta}_{OLS}$ loses its optimality property. In this case robust estimators will outperform OLS estimators. Examples for robust estimators are regression MM-estimators or least trimmed sum of squares (LTS) regression, see Rousseeuw and Leroy (1987). Here we want to give a short overwiev over regression M-estimators.

Regression M-estimators

Instead of minimizing the sum of squared residuals like in (45) we introduce a more general loss function

$$\hat{\beta}_M = \underset{\beta}{argmin} \sum_{i=1}^{n} \rho(y_i - x_i^T \beta). \tag{46}$$

Assumptions on the loss function ρ are that it has to be symmetric and non-decreasing. For higher robustness towards outliers we also require a bounded loss function. If we set $\rho(u) = u^2$, we obtain the OLS estimator as a special case. Furthermore, we can express the robust estimator in (46) as weighted least squares estimator with weights depending on β. Therefore we define the following weights for the i-th observation,

$$w_i^r = \frac{\rho(r_i)}{r_i^2}, \tag{47}$$

where $r_i = y_i - x_i^T \beta$ is defined as the residual in (46). Plugging then (47) into (46) gives

$$\hat{\beta}_M = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^r (y_i - x_i^T \beta)^2. \tag{48}$$

With the aid of this formulation, the estimator $\hat{\beta}_M$ can be computed by means of an iteratively reweighted least squares algorithm.

In the formulation of (47), $\hat{\beta}_M$ would only be robust with regard to vertical outliers. In order to be capable of covering also x-outliers, we have to introduce additional weights w_i^x, which leads to

$$\hat{\beta}_{RM} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^r w_i^x (y_i - x_i^T \beta)^2. \tag{49}$$

The weights w_i^x will be close to 1 if the i-th observation is following the majority of the data in the x-space, while outliers in the x-space, more precisely leverage points, will have weights zero. This formulation covers both horizontal and vertical outliers now, therefore we talk about robust M-estimators (RM).

4.3.2 Partial Robust M-regression

As mentioned in the previous chapter, a possible way of robustifying the PLS method is to use a robust estimator for the covariance in (41) rather than the sample covariance. Here we want to refer to the partial robust M-regression method proposed by Serneels et al. (2005), which is following the partial least squares approach, but uses robust M-regression instead of OLS.

Again we start with the idea of a latent variable regression model. In the case we have a high number of exogenous variables compared to the sample size, we want to compress the contained information to a couple of latent variables and regress the dependent variable on them. For a matrix $X \in \mathbb{R}^{nxk}$ this means that its l latent variables t_j with $j \in 1, ..., l$, are put together in a scores matrix $T \in \mathbb{R}^{nxl}$ and the latent regression model can be written as

$$y_i = t_i^T \alpha + \epsilon_i \tag{50}$$

where t_i is the i^{th} row of T_i and α are the regression coefficients. Now we can make use of the robust M-estimator to estimate α, but here we define the residuals required for the computation of the weights w_i^r as $r_i = y_i - t_i^T \alpha$. If we only use these weights for downweighting large residuals, that is setting $w_i = w_i^r$, we would obtain the Partial M-estimator (PM). But as mentioned previously, we also want to consider leverage points. Therefore we compute appropriate weights w_i^x. Instead of using the original x-variables for computing them, we take the scores t_j. In this way we obtain weights for both vertical and horizontal outliers,

$$w_i = w_i^r w_i^x. \tag{51}$$

At this point the scores t_i are still unknown. In order to obtain them we take an approch mainly following the one in Section 4.2. For getting the scores t_j we have to compute the appropriate loadings vectors a_j with $j \in \{1, ..., l\}$ sequentially:

$$a_j = \underset{a}{\operatorname{argmax}} \ \operatorname{Cov}_w(y, Xa) \tag{52}$$

subject to the constraints

$$\|a\| = 1 \quad \text{and} \quad \mathrm{Cov}_w(Xa, Xa_h) = 0 \quad \text{for} \quad 1 \leq h < l. \tag{53}$$

In contrast to contraints (42) in Section 4.2, we here require the loadings to have length one and their orthogonality. Note that Cov_w stands for the weighted covariance, meaning for any vectors $x, y \in \mathbb{R}^n$ we have

$$\mathrm{Cov}_w(x, y) = \frac{1}{n} \sum_{i=1}^{n} w_i x_i y_i. \tag{54}$$

After computation of all loadings a_j we can put them togheter in a loadings matrix $A \in \mathbb{R}^{kxl}$ and write the scores matrix as $T = XA$. Now we can go back to (50) and finally compute $\hat{\alpha}$ by means of robust M-estimation. The final estimate for β is given by

$$\hat{\beta} = A\hat{\alpha} \tag{55}$$

We see that PLS can be considered as a special case of this formulation if we take equal weights w_i for all observations, leading to a non-robust estimator.

Algorithm

The PRM algorithm is based on the iteratively reweighted partial least squares (IRPLS) algorithm, see Serneels et al. (2005). The IRPLS basically takes some appropriate starting values for the weights w_i, to obtain a first approximation of $\hat{\alpha}$. Afterwards, the weights can be recomputed by using the previously obtained estimator. This enables a second approximation step, where $\hat{\alpha}$ is obtained by applying weighted PLS again.

In contrast to IRPLS, PRM makes following significant extensions:

- *robust staring values:* Prevent the risk of converging to a local minimum, as the objective function used for the partial (robust) M-estimator may have local minima. A local minimum would correspond to a non-robust estimate.
- *consideration of leverage points:* The weights used by IRPLS are only dependent on the residuals after each iteration step. Here we introduce additional weights depending on the values of the scores, in this way capturing leverage points in the x-space.

For defining the additionally required weights w_i^r in Equation 47, several weight fuctions can be used. The ones provided in the appropriate R package sprm, see Serneels and Hoffman (2013), are:

- *Fair:*

  $$w_i^r = f(\nu, c) = \frac{1}{\left(1 + |\frac{\nu}{c}|\right)^2}, \tag{56}$$

  where c is a tuning constant and $\nu = \frac{r_i}{\hat{\sigma}}$, and $\hat{\sigma}$ is a robust estimate of the residual scale. According to the results in Serneels et al. (2005), taking $c = 4$ leads to a good compromise between robustness and statistical efficiency. Letting c go to infinity makes $f(\nu, c)$ flatter and the PRM estimator becomes more alike PLS.

- *Hampel:*

$$w_i^r = f(\nu) = \begin{cases} 1, & |\nu| \le a \\ \frac{a}{|\nu|}, & a < |\nu| \le b \\ a\frac{c-|\nu|}{(c-b)|\nu|}, & b < |\nu| \le c \\ 0, & |\nu| > c, \end{cases}$$

  where $\nu = \frac{r_i}{\hat{\sigma}}$ and for the parameters a, b and c it holds that $0 < a < b < c < \infty$.

- *Huber:*

$$w_i^r = f(\nu) = \begin{cases} 1, & |\nu| \le c \\ \frac{c}{|\nu|}, & |\nu| > c, \end{cases}$$

  where $c > 0$, $\nu = \frac{r_i}{s}$ and the scale parameter s is defined as

$$s = \begin{cases} \hat{\sigma}_c, & |\nu| \le c \\ \hat{\sigma}, & |\nu| > c, \end{cases}$$

  where $\hat{\sigma}_c$ denotes the empirical standard deviation and $\hat{\sigma}$ a robust scale estimator.

For all these weight functions $\hat{\sigma}$ is taken as a robust estimate, more precisely the median absolute deviation,

$$\hat{\sigma} = MAD(x) = MAD(r_1, ..., r_n) = 1.4826 \cdot \underset{i}{median}|r_i - \underset{j}{median}(r_j)|$$

We will stick to the Fair function for weights definition to be consistent with Serneels et al. (2005).

Weights referring to leverage points concerning the scores t_j are then computed as

$$w_i^x = f\left(\frac{\|t_i - med_{L1}(T)\|}{median_i\|t_i - med_{L1}(T)\|}\right) \tag{57}$$

In this expression we consider the distance between each score vector t_i and the center of the data cloud of the score vectors, which are collected in the matrix T. As we want to estimate the center robustly, we take the L_1-median estimator as multivariate version of the sample median, see Serneels et al. (2005). Another way of getting a multivariate robust estimate would be computing the median component-wisely.

Now we can sum up the procedure for computing the partial robust M-estimator to a few points

1. *Computation of robust starting values for $w_i = w_i^r w_i^x$:*

   - for the residual weights w_i^r we take $r_i = y_i - median_j y_j$
   - for the leverage weights w_i^x we take (57) and insert the x_i's instead of the scores, as we have not computed any at that point

2. *Perform PLS on the weighted variables:*

- weighted observations $\tilde{X}$ and $\tilde{y}$ are obtained by multiplying the rows of X and the cells of y with $\sqrt{w_i}$
- PLS is performed with $\tilde{X}$ and $\tilde{y}$, giving an update for the scores matrix T and thus also for $\hat{\alpha}$. The scores also have to be weighted.

3. *Update weights $w_i = w_i^r w_i^x$:*
   We use Equations (56) and (57) again, residuals for (56) are $r_i = y_i - t_i\hat{\alpha}$.

4. *Redo step (2) and (3) until convergence of $\hat{\alpha}$:*
   Convergence is achieved if in the k^{th} iteration step we have $\|\hat{\alpha}_k - \hat{\alpha}_{k-1}\| < \epsilon$ for an arbitrarily small ϵ, where $\hat{\alpha}_k$ is the value of the estimate $\hat{\alpha}$ in the k^{th} iteration step.

5. *Compute $\hat{\beta}_{PRM}$:*
   The limit of $\hat{\alpha}$ is taken for computation of $\hat{\beta}_{PRM}$ according to PLS.

4.4 Sparse Partial Least Squares Regression

The results of this section are mainly based on Chun and Keles (2010). The idea behind sparse partial least squares (SPLS) is to impose an L_1 constraint on the loadings vector and putting the contribution of certain directions equal to zero. This approach often makes sense for problems with a large number of exogenous variables, where estimates often have many components close to zero. These variables have hardy impact on the endogenous variable but still require estimation, which increases prediction uncertainty. SPLS avoids this and leads to dimension reduction and variable selection.

4.4.1 SPLS Model

The actual PLS problem would be modified in this way:

$$\max_{a} \quad a^T M a \qquad \text{s.t.} \qquad a^T a = 1, \quad |a| \leq \lambda, \tag{58}$$

where $M = X^T Y Y^T X$, a denotes the loadings vector and λ determines the amount of sparsity. This formulation tends not to be sparse enough. Therefore we introduce a surrogate direction vector c that is close to a and impose the L_1 penalty on it. In a Lagrangian formulation we get the general form for multivariate Y

$$\min_{a,c} - \kappa a^T M a + (1-\kappa)(c-a)^T M (c-a) + \lambda_1 |c|_1 + \lambda_2 |c|_2 \quad \text{s.t} \quad a^T a = 1, \tag{59}$$

where the L_1 penalty covers the issue of sparsity and the L_2 constraint is introduced due to potential singularity in M when solving for the direction vector. κ controls the effect of the concave part of the equation as there might be issues with local solutions. The rescaled direction vector c of length one is then used as estimated direction vector.

The solution of (59) is obtained by alternatively iterating between solving a for a fixed c or vice versa, which gives us these objective functions:

$$\min_{a} - \kappa a^T M a + (1-\kappa)(c-a)^T M (c-a) \quad \text{s.t} \quad a^T a = 1, \tag{60}$$

and for $0 < \kappa < \frac{1}{2}$ (this choice avoids local solution issues) Equation (60) rearranges to

$$\min_a (Z^T a - \kappa' Z^T c)^T M (Z^T a - \kappa' Z^T c) \quad \text{s.t} \quad a^T a = 1, \tag{61}$$

where $Z = X^T Y$ and $\kappa' = (1-\kappa)/(1-2\kappa)$. (61) can then be solved via the Lagrange method, with the solution given by $a = \kappa'(M + \lambda^* I)^{-1} Mc$. The multiplier λ^* is the solution of $c^T M (M + MI)^{-2} Mc = (\kappa')^2$, see Chun and Keles (2010). Solving for c results in

$$\min_c (Z^T c - Z^T a)^T M (Z^T c - Z^T a) + \lambda_1 |c|_1 + \lambda_2 |c|_2. \tag{62}$$

Especially in the univariate case of Y, a large λ_2 is required for solving (62). In this case we take λ_2 to be ∞ which yields in a solution with a soft penalty. Furthermore, for univariate Y no iteration betweed a and c is required. Instead, we threshold the original PLS direction vector, and get the solution of (59) with $\hat{c} = (\tilde{Z} - \lambda_1/2) + sign(\tilde{Z})$, where $\tilde{Z} = X^T y / \|X^T Y\|$ is the first direction vector of PLS, see Chun and Keles (2010).

4.4.2 SPLS Algorithm

Basically we optimize (59) which gives us relevant variables, so called active variables. These variables are used for PLS then. We can sum the algorithm up to a couple of points, concerning the notation we have $\mathcal{A}$ as index set of active variables, l denotes the number of components and $X_{\mathcal{A}}$ the matrix with the variables defined in $\mathcal{A}$.

1. As starting values we take $\hat{\beta}_{PLS} = 0, \mathcal{A} = \{\}, h = 1$ and $Y_1 = Y$.
2. Until we reach the maximum number of components l we do:
   - Find the direction $\hat{a}$ by solving equation (59) with setting $M = X^T Y_1 Y_1^T X$.
   - The set of active variables $\mathcal{A}$ is taken as $\{i : \hat{a}_i \neq 0\} \cup \{i : \hat{\beta}_i^{PLS} \neq 0\}$, where $\hat{a}_i$ are the components of $\hat{a}$ and $\hat{\beta}_i^{PLS}$ are the components of $\hat{\beta}^{PLS}$.
   - Fit PLS (or robust PLS for robust SPLS) by means of $X_{\mathcal{A}}$ using k latent components.
   - Update the estimate $\hat{\beta}_{PLS}$ through the PLS estimates, $Y_1 \leftarrow (Y - X\hat{\beta}_{PLS})$ and $h \leftarrow h + 1$

4.4.3 Choosing the tuning parameters

According to Equation (59) we would assume to have 4 tuning parameters, namely $\lambda_1, \lambda_2, \kappa$ and l. As we have univariate Y, the problem is independent of κ and as discussed, λ_2 is set to ∞, yielding our problem only to depend on λ_1 and l.

For determining the penalty λ_1 a soft and a hard thresholding approach can be taken. Here we want to stick to the soft approach, where we define a soft thresholded direction vector

$$\tilde{a} = \left(|\hat{a}| - \eta \max_{1 \le i \le p} |\hat{a}_i| \right) I \left(|\hat{a}| \ge \eta \max_{1 \le i \le p} |\hat{a}_i| \right), sign(\hat{a})$$

where p denotes the number of predictors and η plays the role of the sparsity parameter λ_1 with $0 \leq \eta \leq 1$. The single tuning parameter η can now be determined via cross validation (CV). Tuning η for each direction separately is avoided due to very high computational effort.

The number of components l is also tuned by CV. For this reason, CV becomes a function of two parameters in the case of soft thresholding.

4.5 Sparse Partial Robust M-Regression

This section is mainly based on the results of Hoffman et al. (2015). They state that, up to their knowledge, the sparse partial robust M-regression (SPRM) estimate is the first one to combine these three characteristics:

- it is based on projection onto latent structures
- it is integrally sparse concerning both regression coefficients and direction vectors
- it is robust with respect to both vertical outliers and leverage points

4.5.1 SPRMS Model

We want to introduce the SPRM estimate as a sparse version of the PRM estimate. Going back to the PRM regression approach in section 4.3.2 we introduced the weighted variables $\tilde{X} = \Omega X$ and $\tilde{y} = \Omega y$, where Ω is a diagonal matrix with diagional elements $\sqrt{w_i}$. The weigths w_i were defined as $w_i = w_i^r w_i^x$, where w_i^r and w_i^x take care of vertical outiers and leverage points, respectively. Now we can take these weighted variables $\tilde{X}$ and $\tilde{y}$ and plug them into the SPLS formulation (58), yielding:

$$\max_a \quad a^T \tilde{M} a \qquad \text{s.t.} \qquad a^T a = 1, \quad |a| \leq \lambda \tag{63}$$

where $\tilde{M} = \tilde{X}^T \tilde{y} \tilde{y}^T \tilde{X}$. Again we have to impose sparseness on a surrogate direction vector c to achieve sufficiently sparse estimates, resulting in

$$\min_{a,c} -\kappa a^T \tilde{M} a + (1-\kappa)(c-a)^T \tilde{M}(c-a) + \lambda_1 |c|_1 \quad \text{s.t} \quad a^T a = 1 \tag{64}$$

(see equation (59)) and the final estimate of a is given by $a = \frac{\hat{c}}{\|c\|}$, where $\hat{c}$ is the surrogate vector that minimizes (64).

Now we have to find the solution of this SPLS problem. According to Hoffman et al. (2015) it is given by

$$w_h = \Big(|z_h| - \eta \max_i |z_{ih}|\Big) \odot I\left(|z_h| - \eta \max_i |z_{ih}| > 0\right) \odot \text{sign}(z_h) \tag{65}$$

where w_h denotes the h^{th} computed sparse PLS direction vector, z_h the nonsparse PLS direction vector of the deflated X matrix, see Hoffman et al. (2015), z_{ih} the corresponding

components of z_h and the index i runs within the given number of components, $\eta \in [0, 1)$ the sparsity parameter, $I(\cdot)$ the indicator function, giving a vector whose components are equal to 1 if the corresponding argument is true, otherwise zero and $\odot$ denotes the componentwise product.
The computation of the robust M estimator in (49) is then done by iteratively reweighting the least squares estimator.

4.5.2 Choosing the tuning parameters

As we have a univariate Y we only have 2 parameters left to optimize, namely the optimal number of latent variables h_{opt} and the sparsity parameter λ_1. Therefore a grid of values for η and components and $h_{opt} = 1, ..., H$ is sampled and a robust CV searches the best parameter combination. We use a robust criterion, where the mean squared error of the fraction $1 - \alpha$ of the smallest validation residuals is used, the remaining part α of the residuals are assumed to be outliers.

5 Implementation in R

In this chapter we apply the methods discussed in Chapter 4 on residuals obtained in Chapter 3 by using R. We will apply the methods on both the ARIMA and the Kalman residuals and afterwards compare the results. For the implementation in R we use the functions provided by the packages pls Mevik et al. (2013), spls Chung et al. (2013), chemometrics Filzmoser and Varmuza (2014) and sprm Serneels and Hoffman (2013), when necessary functions of the packages were modified to make them applicable to our problem (e.g. in terms of cross validation for time series data) and provide specific output.

5.1 Cross Validation (CV)

A crucial issue in the modeling step is the choice of a "fair" model. This means we do not only want to reconstruct the given data as accurate as possible, the model should be capable of fitting new incoming data well. As mostly only one data set is provided in the modeling step, we can make use of resampling methods. Here we present cross validation (CV) Filzmoser (2013). We use it for determining the number of used latent variables in each model and for determining the amount of sparsity for SPLS models.

For CV the given data set is split up into k segments. As we have time series data, the segments cannot be arranged randomly because this would lead to too pessimistic results. We rather choose consecutive segments in order to retain time behavior within a segment. As some R functions do not have preimplemented consecutive segments for CV, we had to construct them properly.

For $j = 1, ..., k$ we omit the j^{th} segment, which becomes the test set, and fit the model, denoted with $\hat{f}^{-j}$, for the remaining $k - 1$ segments, which become the training set. This is done k times, so that each segment is treated as test data once. Therefore $\hat{f}^{-j}$ is different, depending on which segment is treated as test data. We see that each x_i^j, the i^{th} observation contained in the j^{th}segment, will be part of the test data once and we can define the prediction for this value with $\hat{y}_i^j = \hat{f}^{-j}(x_i^j)$. Hence we get n predictions $\hat{y}_i^j$ for $i = 1, ...n$.

In this way we can define a mean squared prediction error:

$$\widehat{\mathrm{ERR}}_{CV} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i^j) \tag{66}$$

where $L(y, \hat{f}(x))$ denotes a loss function, taken as the squared residuals: $L(y, \hat{f}(x)) = (y - \hat{f}(x))^2$. We choose the model, respectively the number of hidden latent variables, which minimizes (66).

The choice of the number of segments k also plays an important role. An extreme case called "leave one out CV" is taking $k = n$, meaning that each observation is a segment itself. This is computationally expensive and leads to high variance due to the similarity of the training data sets. A popular choice is $k = 10$, but we chose to split the segments in a way that each segments covers the period of 2 years. In this way each segments covers a trend within and over some years.

5.1.1 Double Cross Validation (DCV)

Double Cross Validation (DCV) Filzmoser et al. (2009) basically works like a multistage CV. It works in 2 loops, where the outer loop performs CV as described above. The inner loop then runs CV again but on the test set of the outer loop. In this way we get ”test data from the test data”. An extension would be repeated DCV (rDCV) where DCV is repeatet a couple of times. In the additional loop the data is then split in a different way.

The aim of (r)DCV is to obtain more predicted values from test sets. This leads to a more realistic evaluation of the prediction performance.

5.2 Explanation and Notation

Now we apply the methods presented in Chapter 4 on the ARIMA residuals and Kalman residuals, respectively. In the next sections one will find two plots and a short summary for each method. This should help capturing the main features of the applied methods. Following, there will be an overall summary of the models, where we will compare them according to some indicators. First we give a short overview over the setup of the plots and the summaries.

5.2.1 Model Abbreviations

We shortly want to present the abbreviations used for denoting the methods of Chapter 4 in the plots:

- PCR: principal components regression
- PCR.DCV: principal components regression with rDCV
- PLS: partial least squares regression
- PLS.DCV: partial least squares regression with rDCV
- RPLS: partial robust M-regression
- PRMS: partial robust M-regression (but with another CV criterion, see details below)
- SPLS: sparse partial least squares regression
- SPRMS: sparse partial robust M-regression

5.2.2 Plot of Fitted Values

The first plot features the residuals from the first modeling step (ARIMA and Kalman filtering, respectively) and the fitted values from the corresponding methods of Chapter 4. A vertical line in the middle of the year 2012 divides the plot into training data on the left side and test data on the right side. In our procedure we took 15% of all available observations as test data.

5.2.3 Plot of Residuals

The second plot shows the residuals from the upper plot, namely the differnce between the ARIMA or Kalman residuals and the fitted values of the appropriate model. Within the area of the training residuals we will see two curves. One is displaying the CV residuals, more precisely the prediction errors for the observations in the test segments of the cross validation. The other curve shows the simple training residuals, these are the values we obtain, if we apply the optimal model from the cross validation on all training data points and take the residuals. In the right segment we see the prediction errors for the test data.

Furthermore, all residuals are scaled with an estimated standard deviation $\hat{\sigma}_{CV}$. For its estimation we take the CV residuals as they already display prediction errors of the test segments of the CV. For this reason, taking them tends to be more realistic than taking the simple train residuals. Hence in all plots it will be observable that the CV residuals tend to be bigger than the train residuals.

Furthermore, the ± 2 and $\pm 3\tilde{\sigma}$ tolerance bands are marked horizontally. $\tilde{\sigma}$ denotes the standard deviation of the ARIMA/Kalman residuals which are scaled with $\hat{\sigma}_{CV}$, for this reason $\tilde{\sigma}$ is equal to 1 and the values of the tolerance bands are exactly ± 2 and ± 3 respectively. In the case of normally distributed residuals, the $\pm 3\tilde{\sigma}$ tolerance bands should contain 99.7% of the distribution. For this reason they help us indicating outliers. If the value of the residuals is placed between the bands, the value could be classified as regular point, if it is placed outside the $\pm 3\tilde{\sigma}$ we could give a warning. This value is neither following the model, based on the times series' own history, nor any information provided by financial indicators. Therefore we see that something irregular is happening there, maybe a crisis.

5.2.4 Summary

The summary contains indicators, which should enable the comparison of the models.

- *mse.te*: indicates the estimated mean squared error of the test residuals
- *mse.tr*: indicates the estimated mean squared error of the training residuals, but based on the CV residuals. The square root of this value is taken as $\hat{\sigma}_{CV}$ for scaling.
- *mse.rat*: this value is the ratio between *mse.tr* and *mse.te*, namely $\frac{mse.tr}{mse.te}$. It gives an idea of the models' ability to handle new data. This would be the case if the value is close to one. If the prediction performance is much worse than the training

performance, the value gets closer to zero. In some exceptional cases it is also possible that the value is bigger than one, this means that the prediction errer is lower than the training error.

- *alpha*: within the scope of the used robust methods the optimal model is determined by fitting the best $(1-\alpha)$ values. In oder to make the robust models comparable with other non-robust methods, we compute all indicators on both, all values and the trimmed values. For the robust models the trimming facotor $\alpha = 0.15$ is used. For directly comparing the robust with the non-robust methods we consider the values with $\alpha = 0.15$.

- *e.var*: stands for explained variance. This value can be considered as an R^2 for the data, which shows the goodness of fit. We applied a weighted version, namely

$$R_w^2 = \frac{\sum_{i=1}^n w_i(y_i - \bar{y}_w)^2 - \sum_{i=1}^n w_i(y_i - \hat{y}_w)^2}{\sum_{i=1}^n w_i(y_i - \bar{y}_w)^2}$$

where $\bar{y}_w$ denotes the weighted mean of the observations y_i, $\hat{y}_w$ the weighted fitted values and w_i some appropriate weigths; we used a 0/1 encoding for w_i. E.g., if $\alpha = 0.15$ we take the 15% biggest absolute residuals and assign their observations a weight of 0 and their complement a weight of 1. If $\alpha = 0$ there is no cut-off and all observations are weighted with 1. As these weights do not match the actual weights computed by the robust procedures, we do not obtain an "exact" R^2, therefore its value no longer needs to be $\in [0,1]$ necessarily.

- *ncomp*: denotes the optimal number of components determined by the CV. We applied different criteria for different methods for chosing the number of components. For PCR and PLS we applied the minimum rule, which takes the number of components that minimizes the mean squared prediction error (MSPE).

  For PCR.DCV and PLS.DCV we applied the Hastie rule. This rule basically takes the number of components with minimal MSPE and adds one standard error of this component number. The optimal number of components is then determined by taking a smaller number of components whose MSPE is below this bound, see Filzmoser and Varmuza (2014).

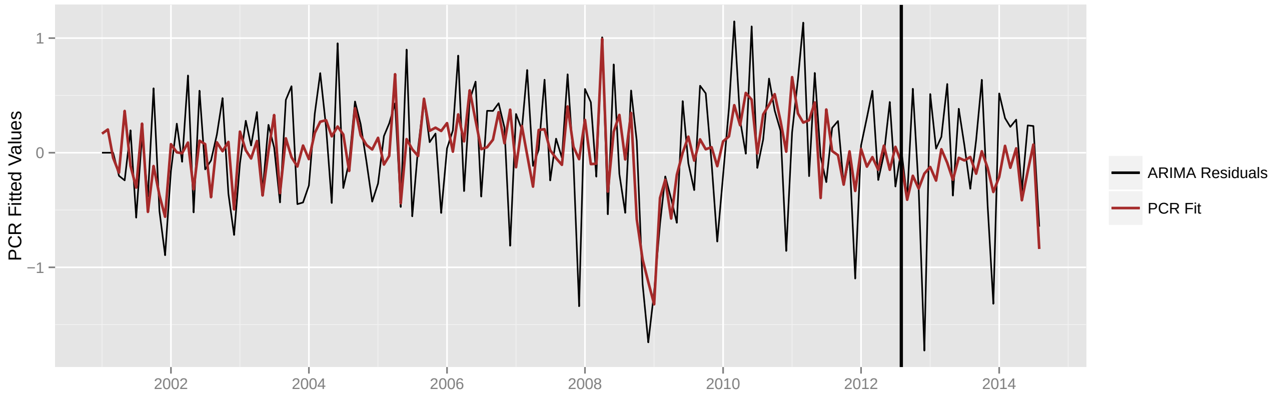
  SPLS and SPRMS work in the way described in Section 4.4.3 and 4.5.2, respectively. Basically, they choose the parameter combination of components number and η that minimizes MSPE.

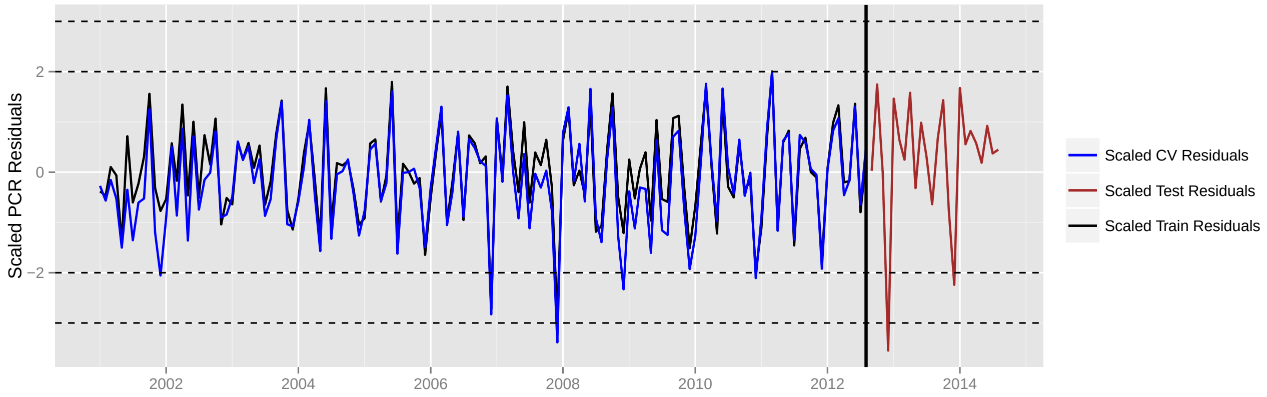
  RPLS and PRMS are both methods based on partial robust M-regression, but distinguish concerning the used CV criterion. For RPLS the mean of the trimmed standard error of prediction (SEP) $\tilde{e}$ is computed for each number of components, as well as their standard errors $\hat{e}$. To the minimum of $\tilde{e}$ a factor $c \cdot \hat{e}$ is added, where c is set equal to 2 in our case. The optimal number of components is the most parsimonious model that is below this bound, see Filzmoser and Varmuza (2014). This matches the Hastie rule, but the Hastie rule does not use trimmed SEP but includes all errors. In contrast thereto PRMS uses the α trimmed MSEP over all observations as robust criterion to choose the optimal model, see Serneels and Hoffman (2013).

- *eta*: for sparse methods we also consider η giving the degree of sparsity. $\eta = 0$ denotes a model where all variables are used, the other extreme with $\eta = 1$ denotes an empty model.

5.3 ARIMA Residuals

5.3.1 PCR



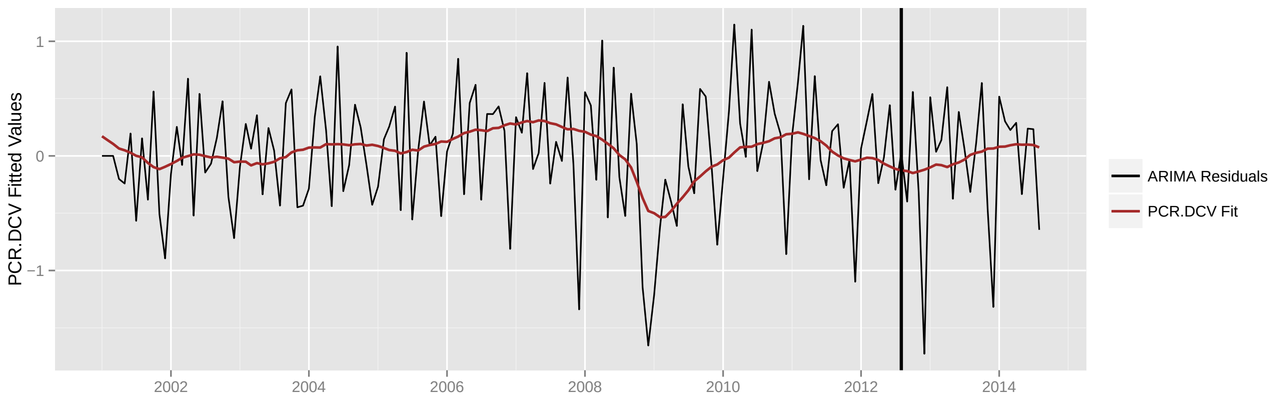


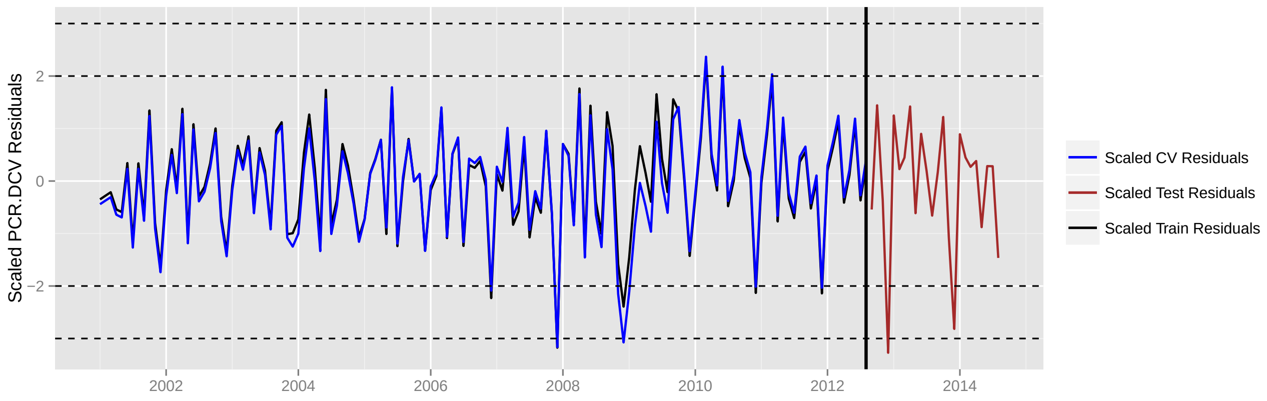
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.282	0.189	0.671	0.388	0	15
0.136	0.101	0.741	0.562	0.15	15

Applying PCR results in a high number of components. Within the procedure one can determine the maximal number of components to be considered for CV. Here the given maximum of 15 components is attained. Therefore we also see that the model is fitting the data quite well.

We can clearly observe the effects of the subprime crisis, starting in 2007 we see that the residuals in the first plot get bigger each year, reaching a peak in 2009. Only the peak in 2009 can be recognized with the aid of financial indices. Therefore the years 2007 and 2008 are stated as outliers in the second plot. Furthermore, the irregularity in 2013 and 2014 is recognised and we have a bigger outlier in 2002, perhaps the after-effect of the dot-com bubble in 2000. The value of the explained variance is not bad and obviously increases by considering the best 85% of the data. The *mse.rat* values indicate a performance decline for the test data, but less extreme for the trimmed data. This could indicate the presence of more extreme outliers in the test data than in the training data.

PCR with rDCV



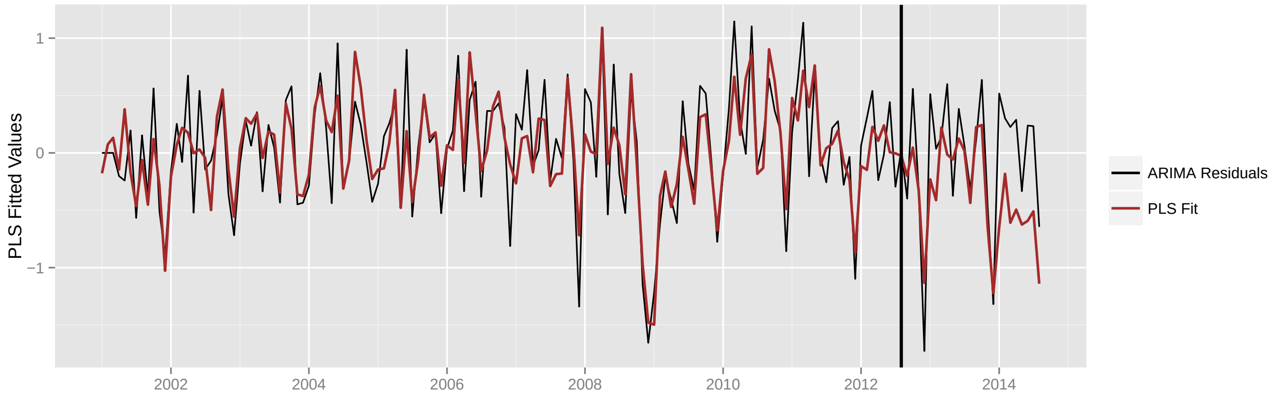


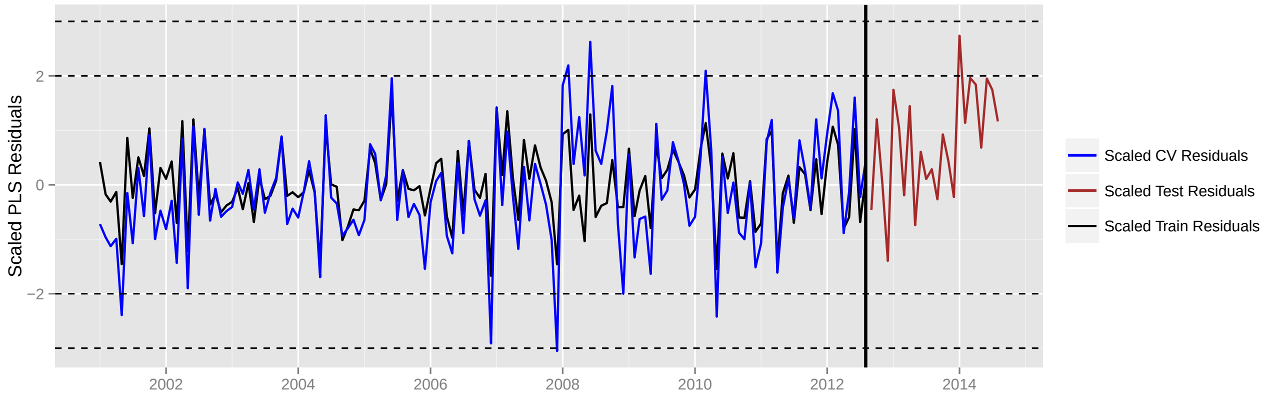
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.337	0.241	0.714	0.116	0	1
0.147	0.121	0.819	0.085	0.15	1

Applying the repeated CV (here with 20 replications) gives a completely different result, using only 1 component for the model. The curve of the fitted values is obviously much smoother than before, also having a clear downturn in 2009. The residual in 2002 is within the normal range now, but 2008, 2009, 2013 and 2014 still indicate outliers.

Due to the smaller number of components, the explained variance is much smaller now. But the *mse.rat* improved in contrast to normal CV.

5.3.2 PLS

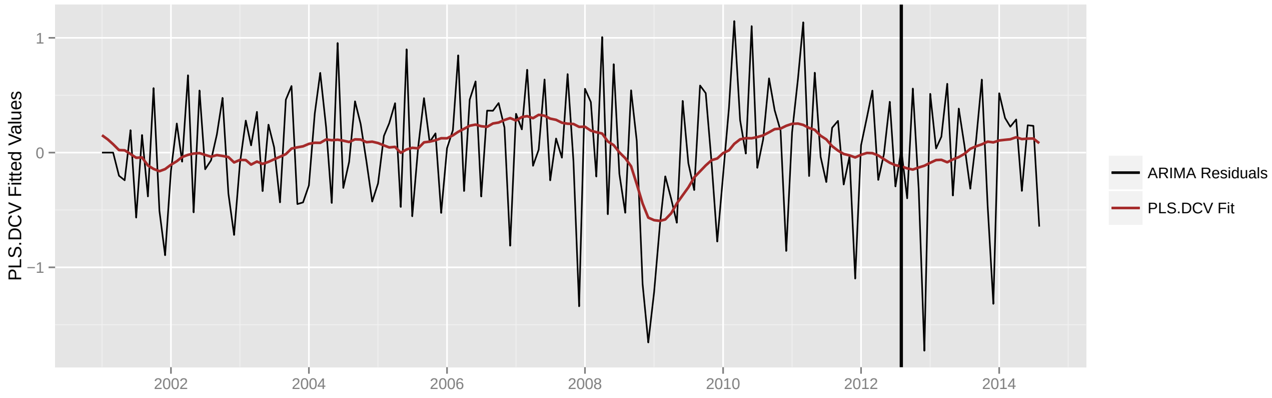


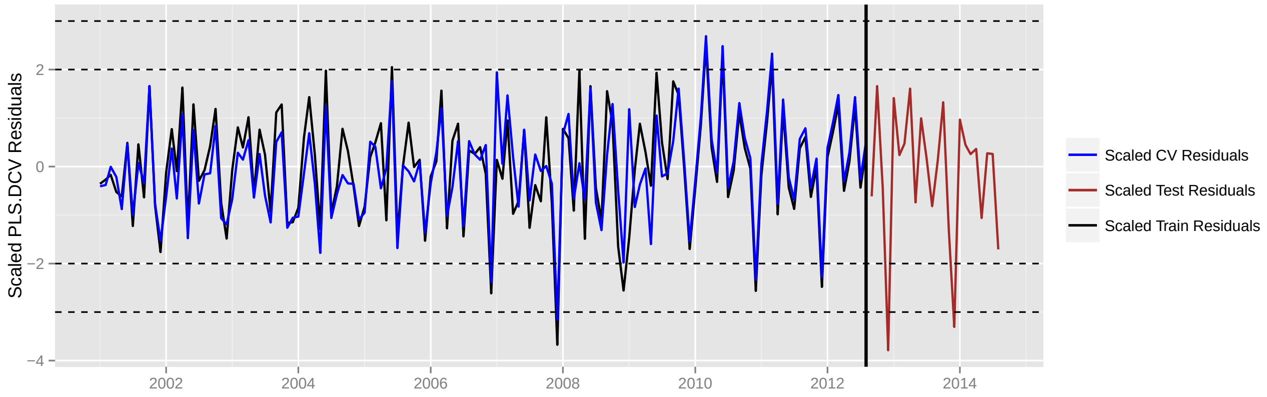


mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.277	0.181	0.653	0.689	0	6
0.186	0.083	0.443	0.820	0.15	6

Here we clearly see the big advantage of PLS: Although it requires less components, the explained variance is much higher than with PCR. Omitting the biggest residuals we have a very good R^2. Furthermore, the fitted values nicely follow the data.
Concerning the residuals we see that normal residuals and outliers are better distinguishable than in the PCR case, as the normal residuals are now much smaller. Here we see that our model reacts quite early on a crisis: in the normal data plot we primaly see the huge effect of the subprime crisis with the downswing in 2009. This is also the point, where the financial indices are influenced and therefore the effect is also captured by the model. Therefore, the residual in 2009 does not appear as an outlier. But in the early years of the subprime crises 2007 and 2008, where there is less impact on the indices, our model already displays the irregularities as outliers.
Similar behavior is observable for the dot-com bubble, where the time series shows an effect in 2002, but the residuals of the model already indicate the outlier in 2001.
We graphically see some difficulties with handling the test data, this also manifests in the $mse.rat$.

PLS with rDCV





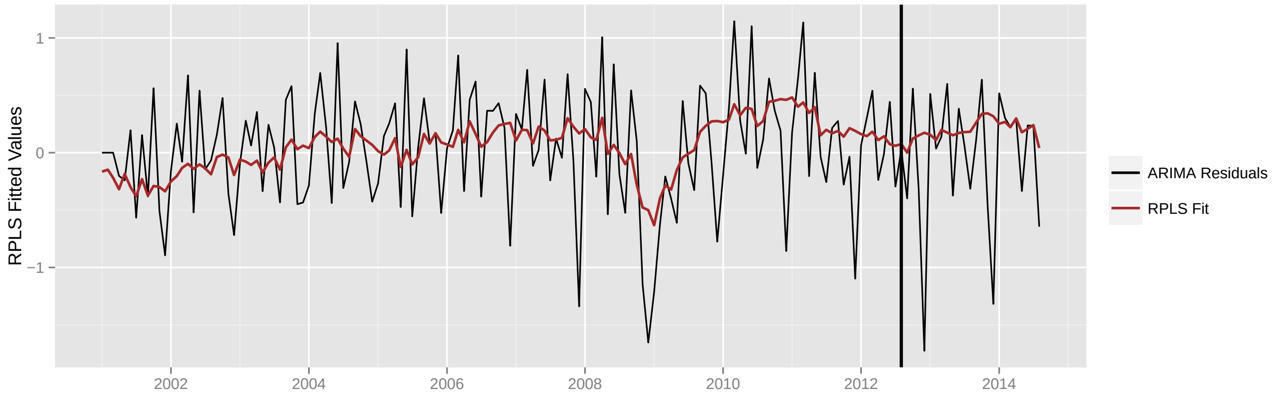
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.338	0.181	0.537	0.141	0	1
0.143	0.086	0.601	0.196	0.15	1

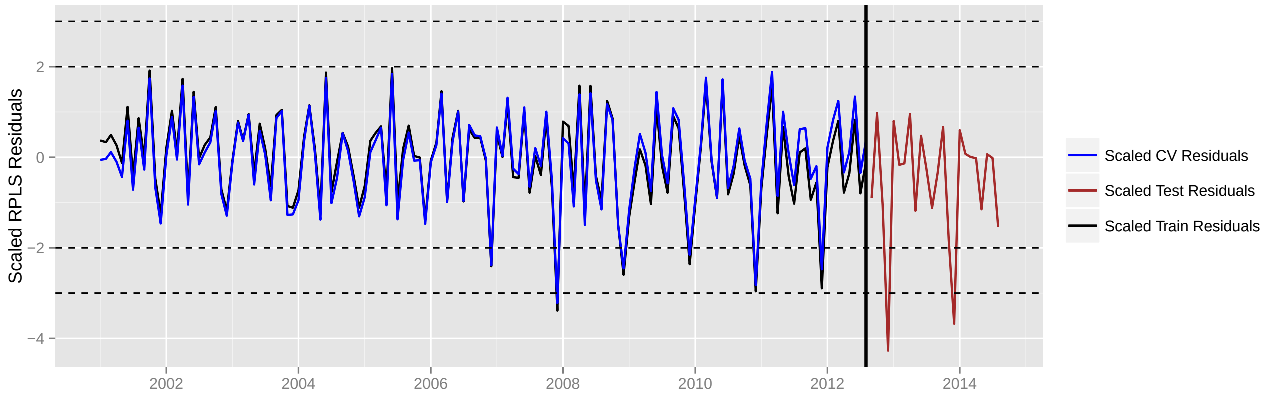
Applying rDCV again results in less components, wherefore the fitted curve is again smoother than with more components and the explained variance decreases, but is still higher than with rDCV applied on PCR.

The downturn in 2009 is again captured with the fitted values and the residuals display again the outliers of 2007 and 2008. But in contrast to simple CV we no more recognise the crisis in the early years of 2000. Nevertheless, the model indicates outliers in 2013 in 2014.

5.3.3 Robust PLS

RPLS





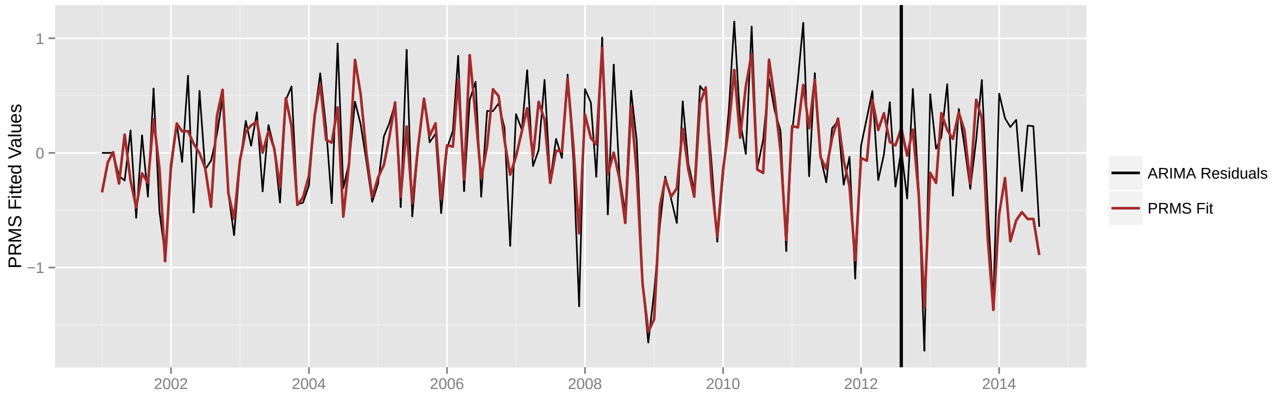
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.388	0.198	0.510	0.218	0	2
0.113	0.101	0.892	0.266	0.15	2

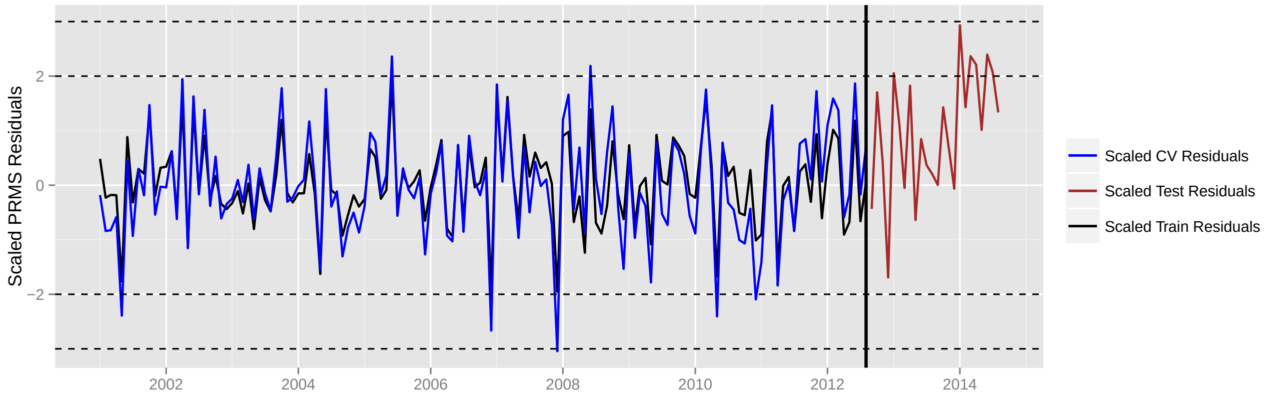
The fitted values of the robustified PLS appear similar to the PLS with rDCV, but less smooth and stronger following the peaks of each year. In contrast to normal PLS, less components are required. This shows again, how much more information is required to capture the effects of outliers.

The effects of the subprime crisis again manifest in the outliers, but we continuously keep having very large residuals after 2007. This might show up some long term effects of the crisis.

Due to robustness and using only 2 components, the model does not capture very much of the overall variance. For the trimmed data the *mse.rat* value indicates a quite good prediction performance.

PRMS



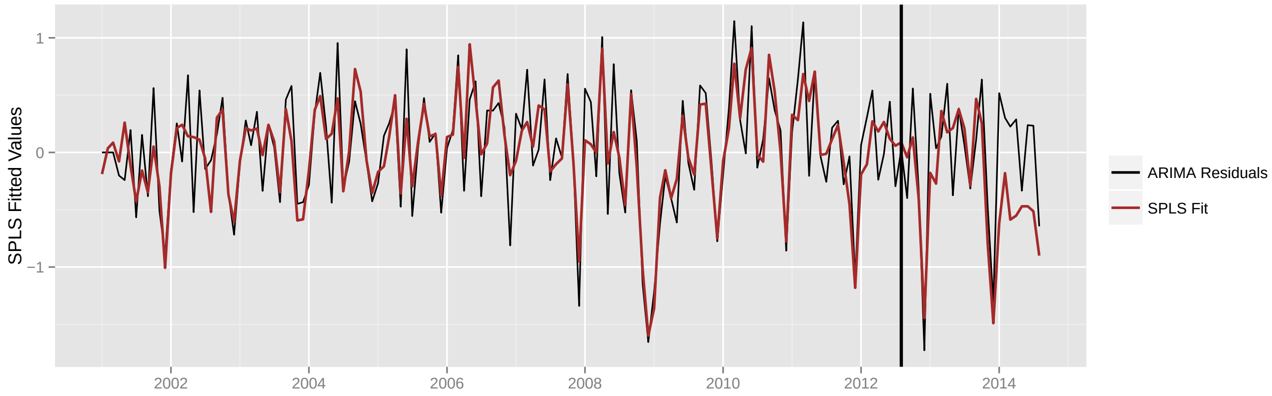


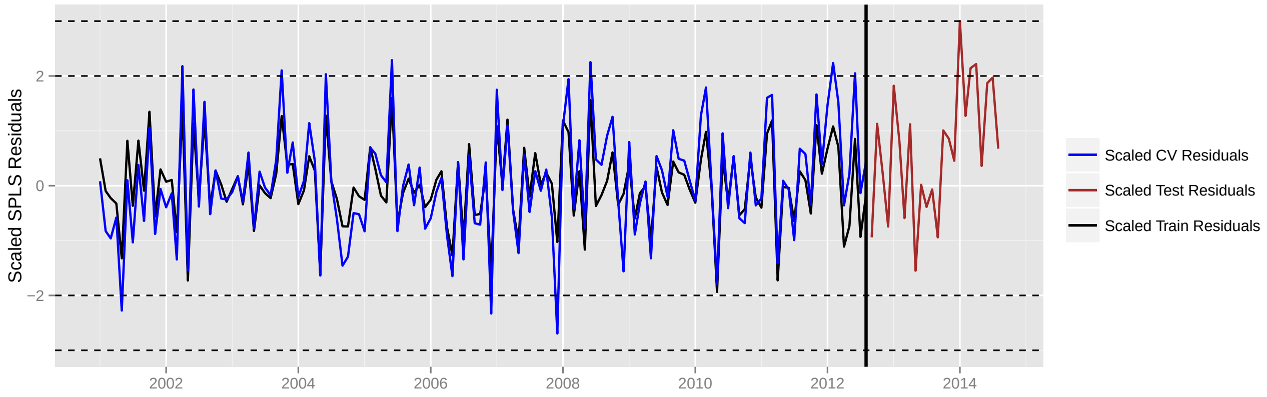
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.320	0.146	0.456	0.669	0	6
0.227	0.067	0.297	0.808	0.15	6

The CV criterion used for PRMS results in a higher number of components, which leads to a better fit towards the data and therefore to higher *e.var*.
We can observe difficulties with modeling the test data, which manifests in the *mse.rat* value. Cutting off big residuals even leads to a decrease of *mse.rat*. A possible explanation is that we see in the upper plot that the fitted values capture the huge residual in 2013, but has difficulties with ”normal” values. If the big residual is then cut off, the moderate fit to the other data has a bad impact. On the other side this could even indicate a crisis, as these values appear unexpected for the model.

Again we see in the lower plot that normal residuals and outliers are well distinguishable. PRM is able to capture the effects of the dot-com bubble and the subprime crisis very early. Furthermore, we have an outlier in 2014.

5.3.4 SPLS

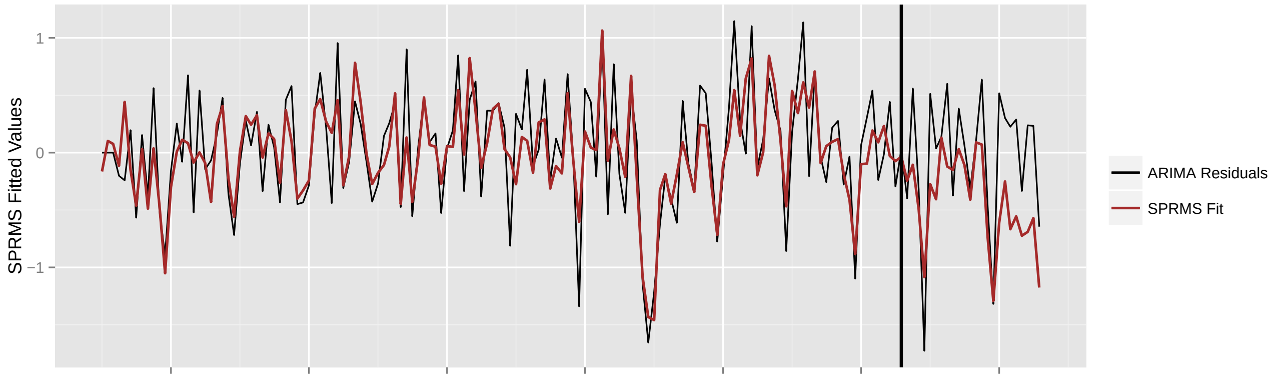


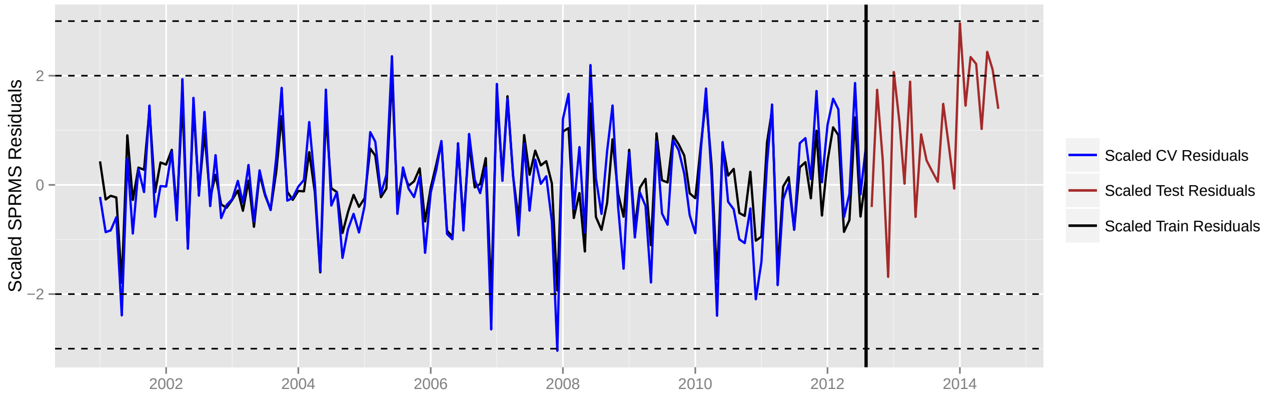


mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
0.250	0.144	0.574	0.723	0	7	0.5
0.159	0.069	0.432	0.858	0.15	7	0.5

Sparse PLS uses 7 components, hence we have a big *e.var* and the structure of the data is fitted well, but again with deviations in the test data.
The lower plot shows big residuals in many years, but clear outliers in 2001, 2002, 2007, 2008 and 2014.
Furthermore, SPLS imposes sparsity on the parameters, namely an $\eta = 0.5$, which leads to using 207 among 232 financial indices.

5.3.5 Robust SPLS





mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
0.328	0.145	0.442	0.663	0	6	0
0.236	0.067	0.286	0.809	0.15	6	0

The robustified SPLS requires less components, but also shows a good value for *e.var*. Again we see a deviation of the ARIMA residuals and the fitted values in the test data, the big residuals are then displayed as outliers in the lower plot. Furthermore, the years 2001, 2007 and 2008 are obviously outliers. Again we recognize that the robust methods makes normal and extreme residuals better distinguishable.

Furthermore, the robust SPLS imposes no sparsity in contrast to SPLS.

5.4 Summary

Here we want to provide a short summary and some conclusions on the previous methods and plots

Method	mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
PCR	0.282	0.189	0.671	0.388	0	15	-
	0.136	0.101	0.741	0.562	0.15	15	-
PCR.DCV	0.337	0.241	0.714	0.116	0	1	-
	0.147	0.121	0.819	0.085	0.15	1	-
PLS	0.277	0.181	0.653	0.689	0	6	-
	0.186	0.083	0.443	0.820	0.15	6	-
PLS.DCV	0.338	0.181	0.537	0.141	0	1	-
	0.143	0.086	0.601	0.196	0.15	1	-
RPLS	0.388	0.198	0.510	0.218	0	2	-
	0.113	0.101	0.892	0.266	0.15	2	-
PRMS	0.320	0.146	0.456	0.669	0	6	-
	0.227	0.067	0.297	0.808	0.15	6	-
SPLS	0.250	0.144	0.574	0.723	0	7	0.5
	0.159	0.069	0.432	0.858	0.15	7	0.5
SPRMS	0.328	0.145	0.442	0.663	0	6	0
	0.236	0.067	0.286	0.809	0.15	6	0

We clearly see that PCR is the model with most required components, but is outperformed by PLS methods, both robust and/or sparse. Methods with the least number of components are those which make use of rCV.

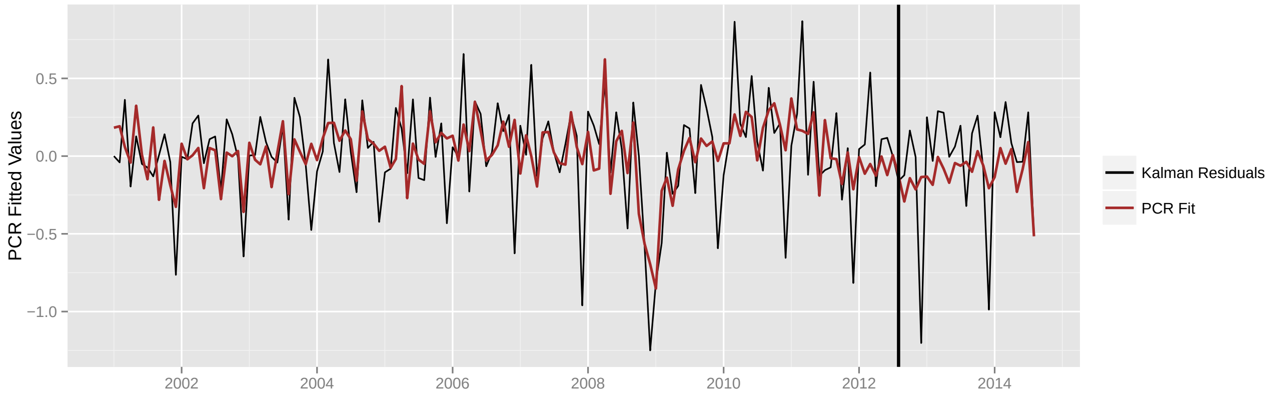
Concerning *mse.rat* values, PCR shows the best performance, while SPLS performs best concerning *e.var*.

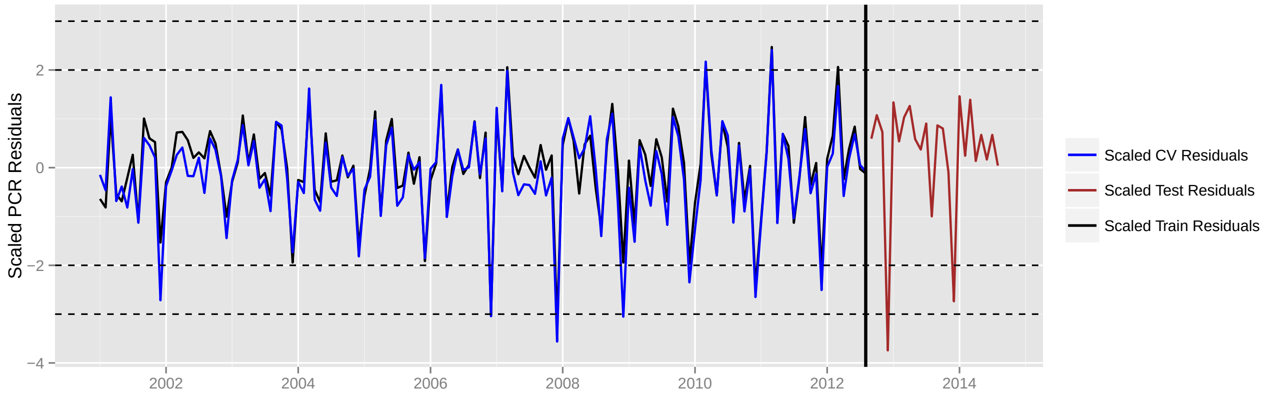
Comparing the robust PLS methods, PRMS uses more components and has therefore a higher *e.var* value and lower mean squared errors. Only the *mse.rat* lags a bit behind RPLS. But all in all PLS, PRMS and SPRMS appear to have similar performance.

In contrast thereto SPLS imposes sparsity and outperforms SPRMS with regard to all key figures. Furthermore, we see that the robust SPLS does not impose any sparsity, meaning the results are almost the same as in PRMS. The minimal differences can be due to different implementations of the algorithm in both functions.

5.5 Kalman Residuals

5.5.1 PCR



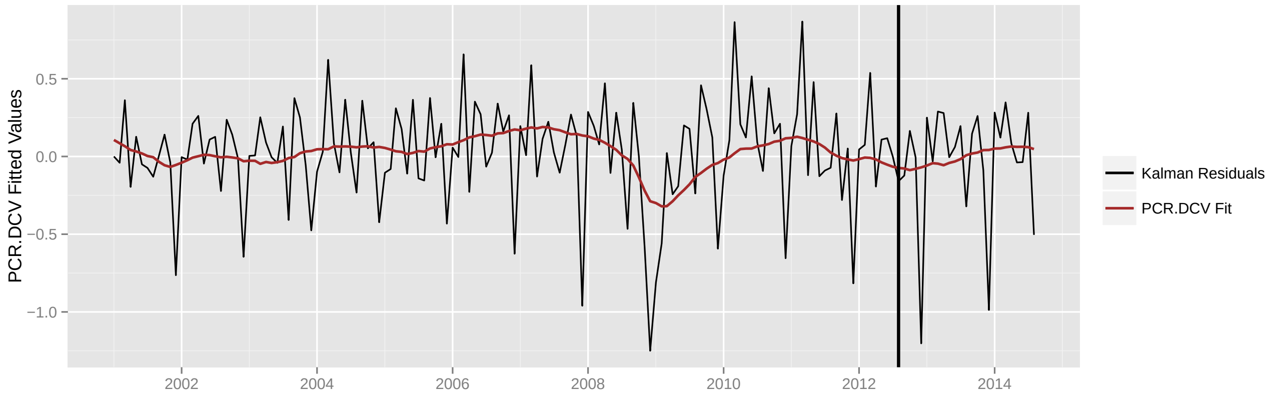


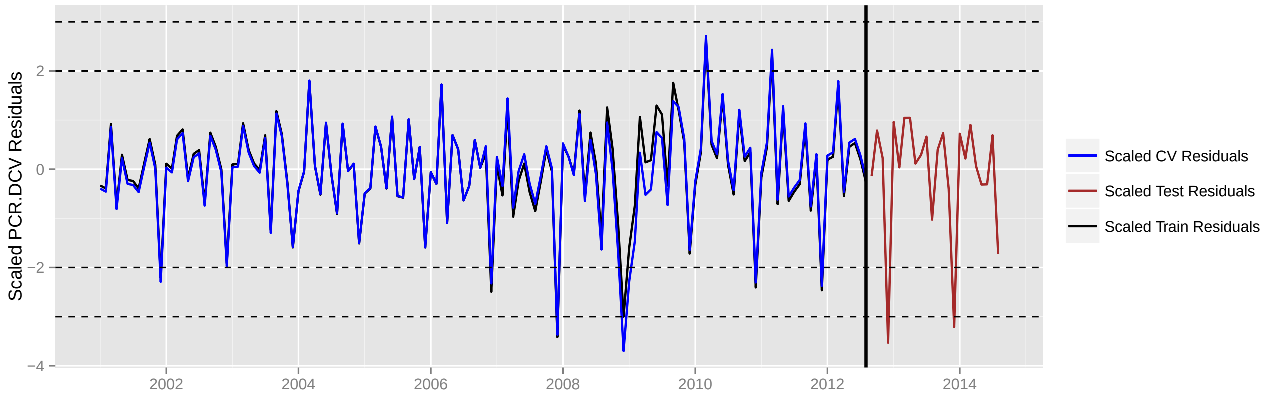
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.126	0.081	0.648	0.369	0	15
0.052	0.027	0.526	0.462	0.15	15

In constrast to the ARIMA residuals, the Kalman residuals tend to have stronger peaks each year, the one in 2009 is also well captured by the fitted values. The others result in bigger residuals in the lower plot, which displays many years as outliers - the most extreme are in 2008 and 2013. But we see that the subprime and dot-com crisis become apparent through the plot.

Again PCR attains the maximum of allowed components, namely 15, but explains the variance only moderately.

PCR with rDCV





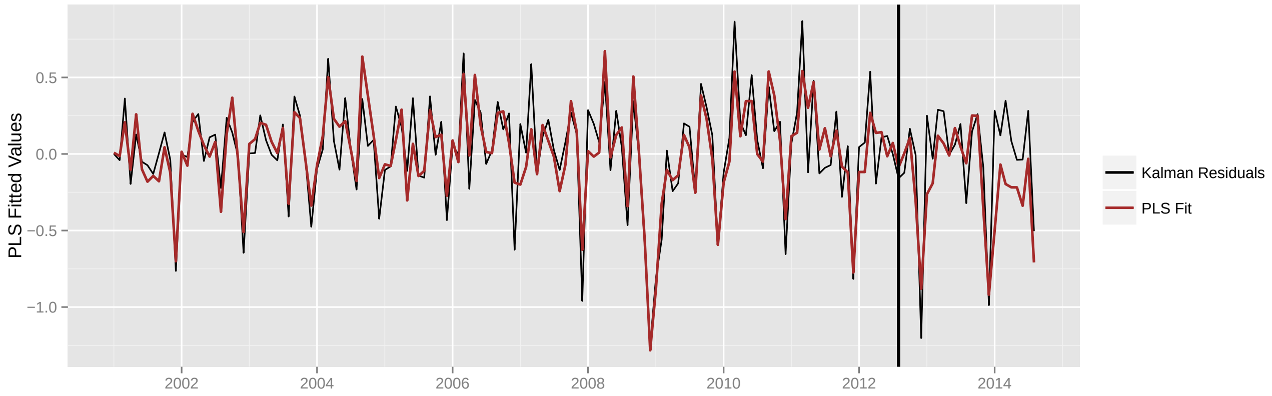
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.1459	0.103	0.705	0.010	0	1
0.041	0.036	0.880	0.105	0.15	1

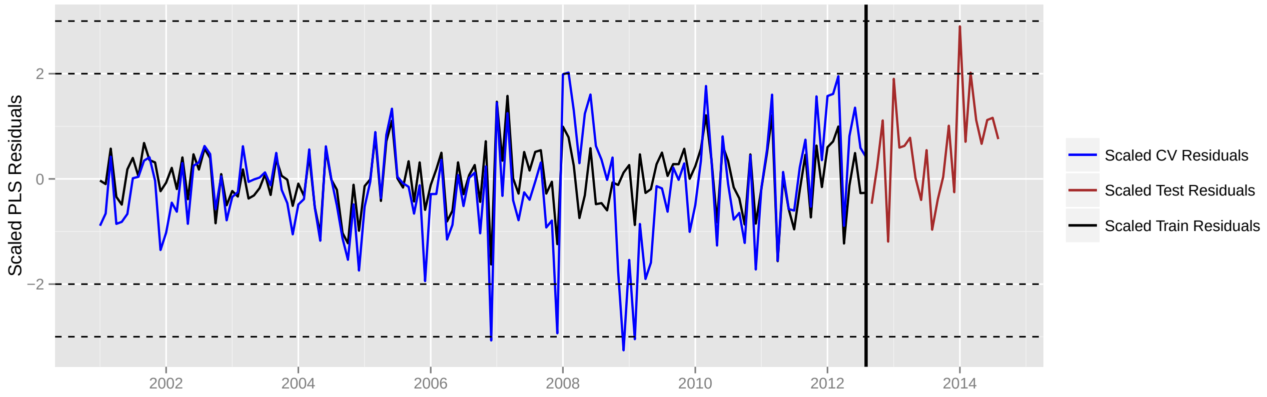
Applying repeated DCV results again in only 1 component, this reduces the values of *e.var*, but the *mse.rat* value suggest a quite good prediction performance.

The curve of the fitted values is quite smooth, due to only using one component, but captures the downswing in 2009.
Considering the residuals we see that the residuals increase within the course of the subprime crisis, having an extreme outlier in 2009, but also later in 2013 and 2014. The effect of the dot-com bubble is now less intense on the plot.

Using one component also diminishes the value of the variance explained by the model.

5.5.2 PLS



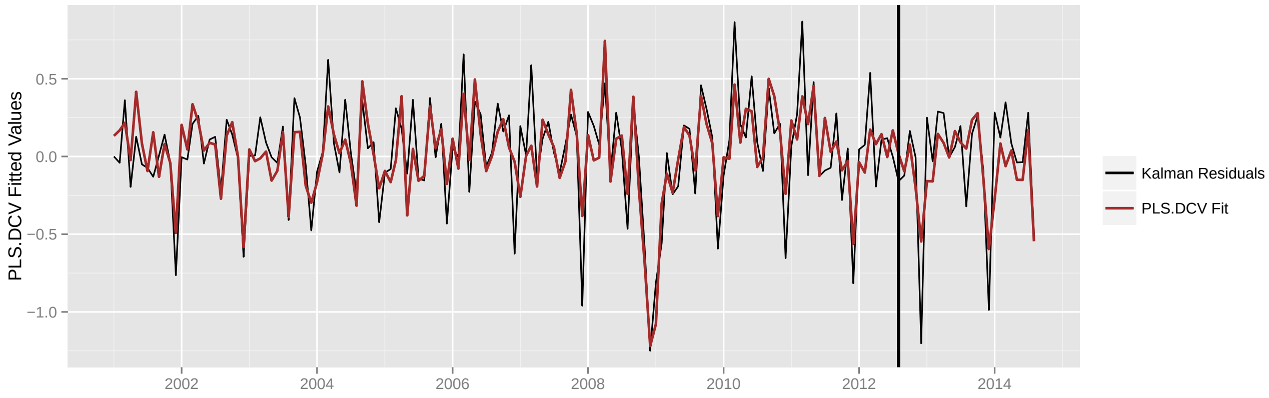


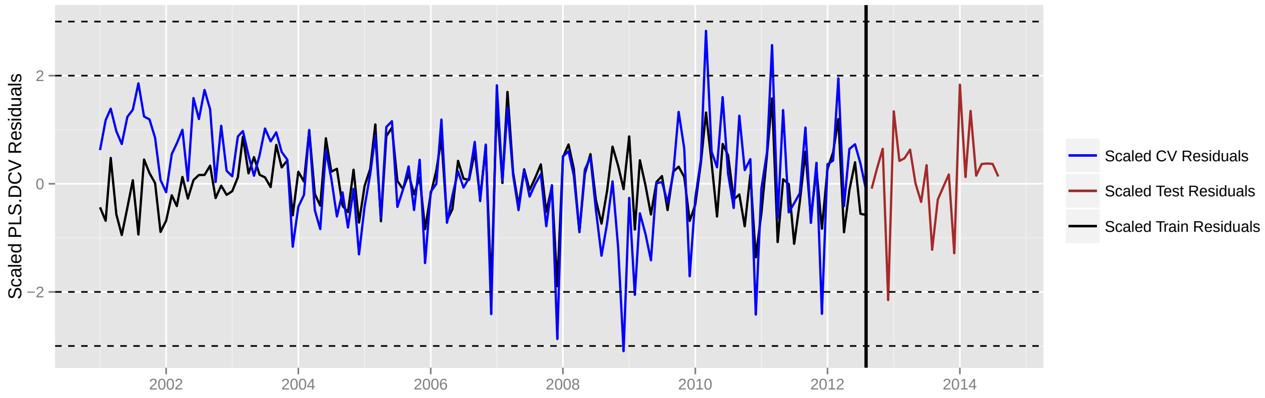
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.086	0.073	0.845	0.776	0	6
0.043	0.0309	0.726	0.868	0.15	6

PLS results as assumed in a higher number of components, which manifests in a very good fit in the first plot and high ratio of explained variance. Furthermore, we have a good *mse.rat*, which indicates good prediction preformance.

Considering the residuals we see that only the subprime crisis is captured and each year equally intense. Furthermore, we have an outlier in 2014, while there is no visible impact of the dot-com bubble. It is also observable, that the irregularities within the subprime crisis are only captured by the CV residuals and not by the training residuals.

PLS with rDCV





mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.065	0.093	1.422	0.674	0	5
0.0312	0.041	1.310	0.782	0.15	5

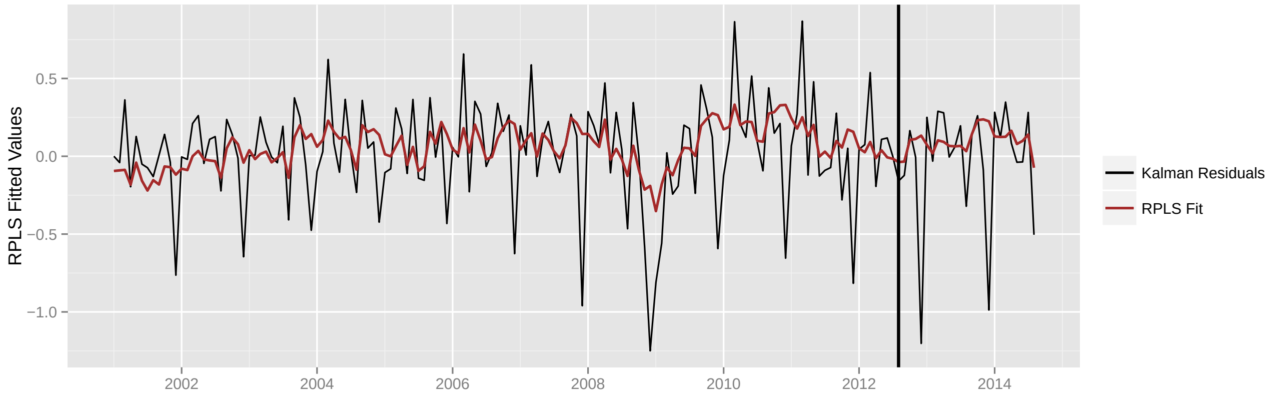
Application of repeated DCV does not diminish the number of components that dramatically now. Using 5 components also leads to a good fit, the downswing in 2009 seems to be captured entirely. Therefore, the explained variance is quite high.

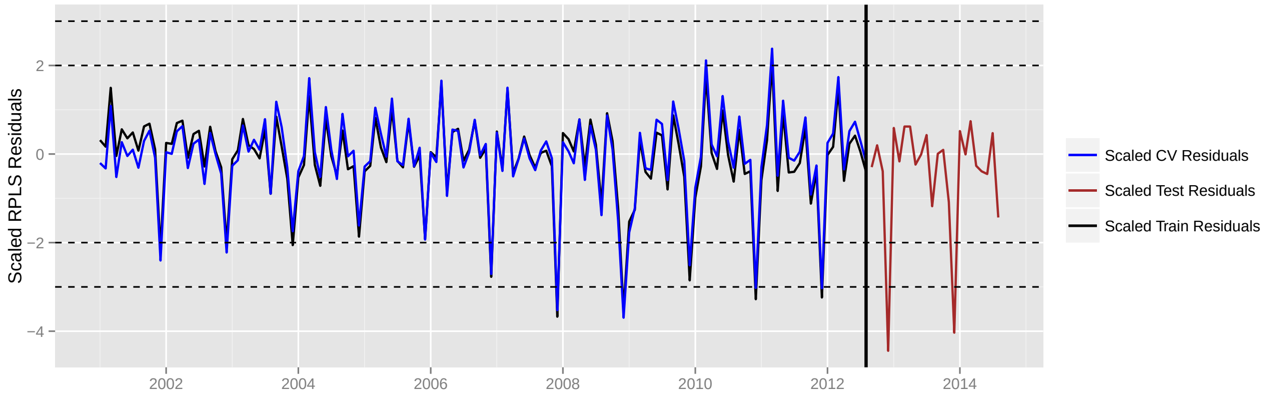
As discussed it is also possible to have a *mse.rat* bigger than one, right here we exactly have this case: the model seems to fit the test data better than the training data.

Considering the residuals plot we have very large residuals starting from 2007, while the test data does not indicate extreme outliers, but 2013 and 2014 still having large residuals. The first couple of years show a rather difference between rDCV and simple training residuals.

5.5.3 Robust PLS

RPLS





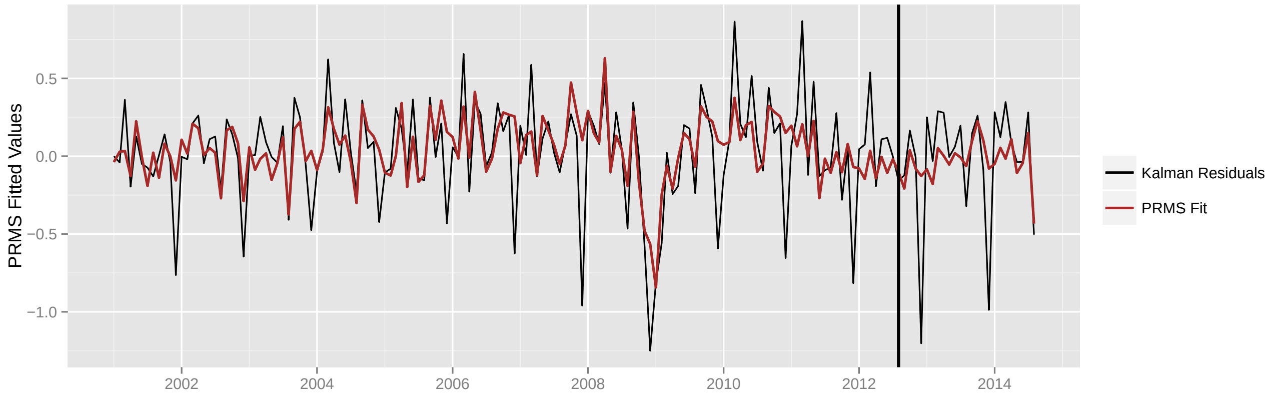
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.165	0.090	0.548	0.194	0	2
0.025	0.024	0.981	0.399	0.15	2

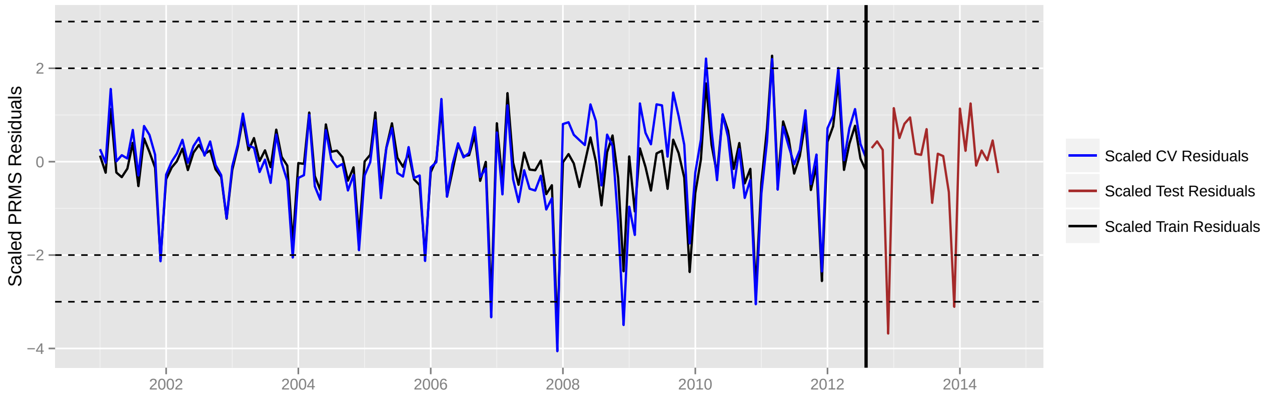
The robustified version of PLS applies only 2 components, which reduces the *e.var* value in comparison to its non-robust counterpart. Taking a 1-component model, the PCR with rDCV, we see what a huge impact switching to PLS and applying robust procedures can make. Adding one component and applying RPLS increases the *e.var* from 0.01 to 0.19.

Considering the trimmed values we also find a very good *mse.rat* for this model.

The residuals for this model generally scatter quite widely and starting from 2008 we find almost every residual exceeding the -3 σ tolerance band.

PRMS



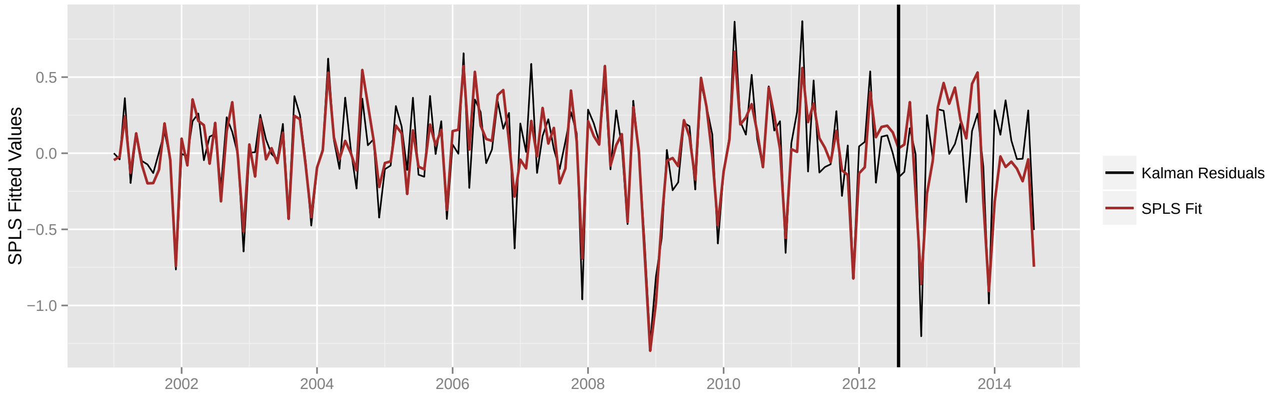


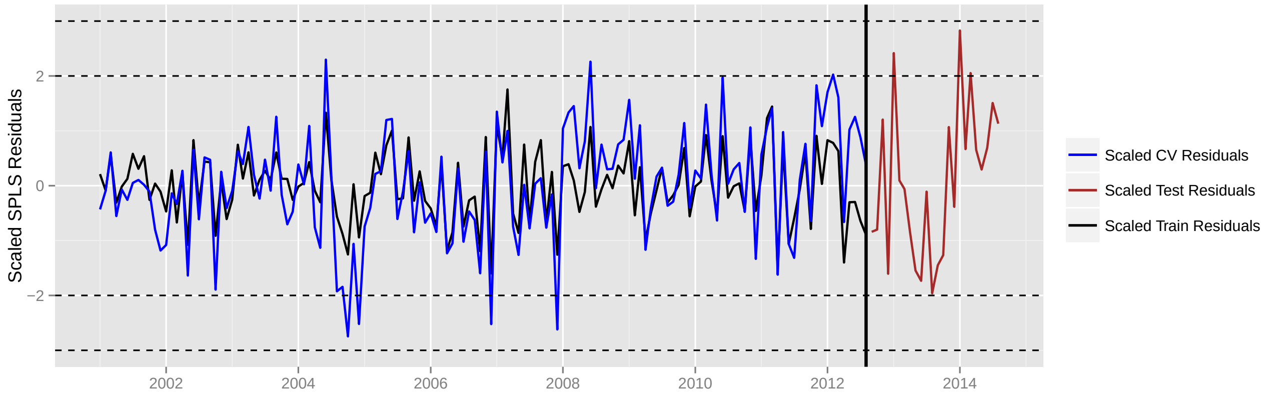
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp
0.113	0.085	0.757	0.427	0	4
0.028	0.026	0.930	0.692	0.15	4

The CV criterion for PRMS results again in a higher number of components, which leads to a more accurate fit of the model to the Kalman residuals and an increasing *e.var* value. Furthermore, the *mse.rat* is quite good, especially for the trimmed values.

Here the residuals during the subprime crisis appear more than before.

5.5.4 SPLS



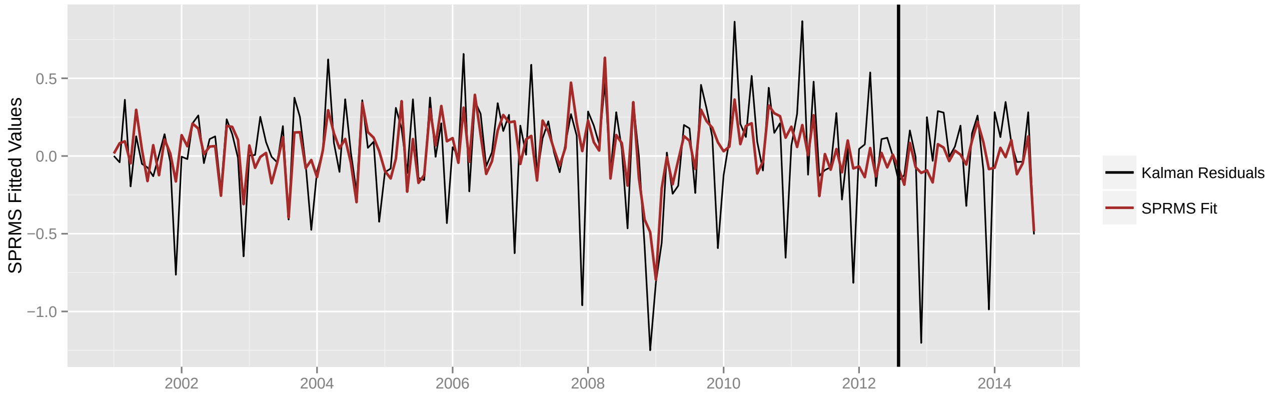


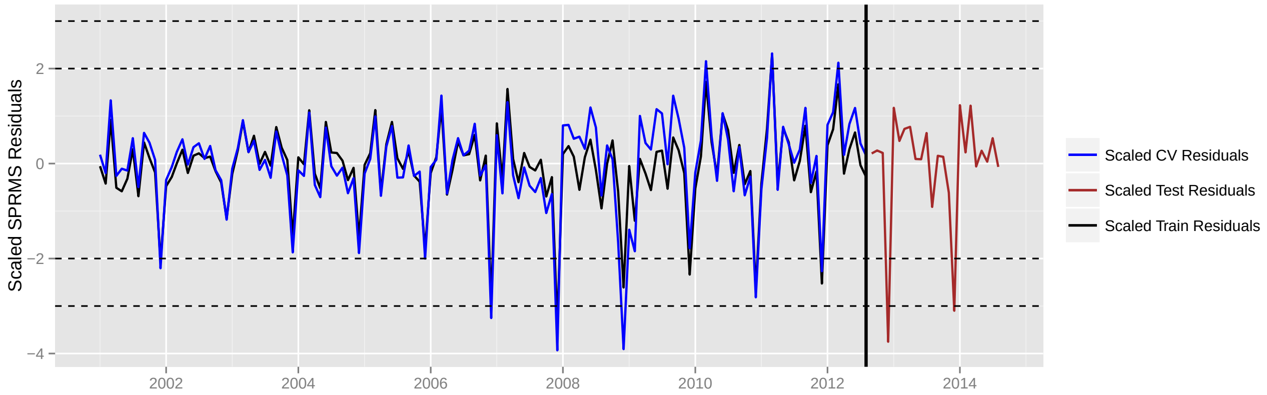
mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
0.082	0.046	0.553	0.828	0	8	0.7
0.055	0.022	0.406	0.894	0.15	8	0.7

Applying SPLS requires more components than the previous model, but at the same time it imposes high sparsity with $\eta = 0.7$. This means the model uses only 99 of overall 232 financial indices.

The model fits the data very good, leading to a good *e.var* value for both trimmed and non trimmed residuals. Considering the residuals we see that besides the typical outliers during the subprime crisis and in 2013 and 2014, the SPLS model indicates outliers in 2004 and 2005.

5.5.5 Robust SPLS





mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
0.112	0.085	0.757	0.443	0	4	0.1
0.026	0.025	0.948	0.677	0.15	4	0.1

Robustifying SPLS requires less components, wherefrom the *e.var* value is suffering, but the *mse.rat* remains good. Robust SPLS imposes sparsity with $\eta = 0.1$ leading to 217 used indices.

The residual plot also shows the usual outliers during the subprime crisis and in the test data. In contrast to simple SPLS it captures the effects of the dot-com bubble.

5.6 Summary

Method	mse.te	mse.tr	mse.rat	e.var	alpha	ncomp	eta
PCR	0.126	0.081	0.648	0.369	0	15	-
	0.052	0.027	0.526	0.462	0.15	15	-
PCR.DCV	0.1459	0.103	0.705	0.010	0	1	-
	0.041	0.036	0.880	0.105	0.15	1	-
PLS	0.086	0.073	0.845	0.776	0	6	-
	0.043	0.0309	0.726	0.868	0.15	6	-
PLS.DCV	0.065	0.093	1.422	0.674	0	5	-
	0.0312	0.041	1.310	0.782	0.15	5	-
RPLS	0.165	0.090	0.548	0.194	0	2	-
	0.025	0.024	0.981	0.399	0.15	2	-
PRMS	0.113	0.085	0.757	0.427	0	4	-
	0.028	0.026	0.930	0.692	0.15	4	-
SPLS	0.082	0.046	0.553	0.828	0	8	0.7
	0.055	0.022	0.406	0.894	0.15	8	0.7
SPRMS	0.112	0.085	0.757	0.443	0	4	0.1
	0.026	0.025	0.948	0.677	0.15	4	0.1

We see like in the case of ARIMA residuals that the PCR method requires most components and repeated DCV leads to a decrease of the component number, but now less dramatically in the case of PLS.

Compared to all other models, PCR does not seem recommendable. PLS and PLS with rDCV appear quite comparable, but PLS with rDCV having a better performance on test data, which justifies the approach of rDCV.
The PRMS method slightly outperforms the RPLS. SPLS uses twice as many components as SPRMS, but imposes high sparsity, while the SPRMS gets along with less components and having a better *mse.rat* value and smaller *mse.te* for the trimmed values. Due to a higher component number SPLS captures more variance.

Comparing the robust with the non-robust models we have to consider the trimmed values. With regard to them, PLS (rDCV), PRMS and both sparse PLS methods appear quite good.

6 Conclusion

In Chapter 3 we extracted the signal of the underlying time series by means of ARIMA and Kalman filtering. ARIMA suggests the process to be integrated and depending on both past observations and past noise terms. The Kalman filter showed a smaller standard error than the ARIMA model, which suggests a more precise fit to the data. Furthermore, we stated the general applicability of state space models.

The implementation in R of the discussed methods of Chapter 4 was shown in Chapter 5. A comparison of all methods did not reveal a method that clearly outperforms the others concerning all discussed key figures, which were mean squared error of training and test data, explained variance, or number of used latent variables; in the case of sparse methods we added the sparsity parameter η. It appeared obvious that the performance of PCR lagged behind all others. It required throughout the highest number of components but still captured less variance than other methods with less components. Application of double cross validation, which ought to improve prediction performance, led to a decrease of required components in all cases. We also saw that the different implementations of robust Partial M-Regression led to different numbers of components, as they apply different criteria for component selection (see Chapter 4). The PRMS criterion generally results in a higher number of components.

Robust and non-robust sparse PLS methods result in different sparsity, robust methods are either hardly or even not sparse. For non-robust SPLS the highest induced sparsity of $\eta = 0.7$ only reduced the number of used financial indicators from 232 to 99. This shows that the structure in the data is not driven by a few particular indicators. We also recognized that the models to not impose a very high number of components, with exception of PCR, 8 components maximally. Therefore one could consider including other exogenous variables into the models.

Concerning the outlier detection performance we saw that the subprime crisis was indicated by all models already in 2007, although the dramatic slump in the original data is firstly visible in 2009. Some models showed difficulties in recognizing the effects of the dot-com bubble. In the case of ARIMA residuals those were PCR and PLS with rDCV and robust PLS. In the case of Kalman residuals, PLS and SPLS could not recognize the dot-com bubble effects. All models stated some irregularities in the test data around the year 2014. Overall, the models appear to be capable of detecting outliers in the time series.

Appendix

R Code for Sections 4 and 5

```
#LOAD REQUIRED DATA
load("DataPz.Rdata") #standardized & demeaned financial indicators
load("salesz.Rdata") #standardized & demeaned time series

#ARIMA MODEL ESTIMATION
bj<-auto.arima(sales.z[which(!is.na(sales.z[,2])),2], max.p=10,
               max.q=10) #fixed max p,q values
sales.bjr<-sales.z
sales.bjr[!is.na(sales.bjr[,2]),2]<-bj$residuals

#STATE SPACE MODEL ESTIMATION
sales.ssr<-sales.z
dat<-sales.z[which(!is.na(sales.z[,2])),2]
loclevel <- function(p)
{dlmModPoly(1, dV=exp(p[1]), dW=exp(p[2]))}
fit <- dlmMLE(dat, parm=c(0,0), build=loclevel)
dlm.mod <- loclevel(fit$par)
StructTS(dat, type="level")
filter <- dlmFilter(dat, mod=dlm.mod)

v <- sqrt(dropFirst(unlist(dlmSvd2var(filter$U.C,
                                      filter$D.C))))
kuf <- dropFirst(filter$m) + 2*(v)
klf <- dropFirst(filter$m) - 2*(v)
f.res<-dat-dropFirst(filter$m)
sales.ssr[which(!is.na(sales.ssr[,prod])),prod]<-f.res

#MERGING THE ARIMA/STATE SPACE RESIDUALS
#WITH THE FINANCIAL INDICATORS
DatCombi<-merge(sales.bjr,DataP.z,by.x="Date",
                by.y="Date", all.x=T) #or
DatCombi<-merge(sales.ssr,DataP.z,by.x="Date",
                by.y="Date", all.x=T)

#APPLYING METHODS
# Principal Components Regression
##PCR
sales.pcr<-pcr(Response~ FinancialD, data=train.l,ncomp=15,
      validation="CV",segments=floor(length(train.l$Date)/24),
      segment.type="consecutive" ,center=T, scale=T)
rmsep.pcr<-RMSEP(sales.pcr, estimate="adjCV")
```

```
ncomp.pcr<-which.min(rmsep.pcr$val)-1

##PCR.DCV
tmp.pcr.dcv<-my.mvr_dcv(Response~ FinancialD,data=train.l,scale=T,
      center=T, ncomp=15,segments=floor(length(train.l$Date)/24),
      segment.type="consecutive",segments0=4,
      segment0.type="consecutive", method="svdpc", repl=20,
      selstrat=c("hastie"))
ncomp.pcr.dcv<-tmp.pcr.dcv$afinal

# Partial Least Squares Regression
## PLS
sales.plsr<-plsr(Response~ FinancialD,data=train.l,validation="CV",
          ncomp=15,segments=floor(length(train.l$Date)/24),
          segment.type="consecutive", center=T, scale=T)
rmsep.plsr<-RMSEP(sales.plsr, estimate="adjCV")
ncomp.plsr<-ifelse(which.min(rmsep.plsr$val)==1,1,
                   which.min(rmsep.plsr$val)-1)


## PLS.DCV
tmp.plsr.dcv<-my.mvr_dcv(Response~ FinancialD,data=train.l,scale=T,
      center=T, ncomp=15, segments=floor(length(train.l$Date)/24),
      segment.type="consecutive",segments0=4,
      segment0.type="consecutive",method="simpls", repl=20,
      selstrat=c("hastie"))
ncomp.plsr.dcv<-tmp.plsr.dcv$afinal


## RPLS
sales.rplsr.dcv<-prm_cv(train.l$FinancialD,train.l$Response, a= 15,
                         segments=floor(length(train.l$Date)/24),
                segment.type="consecutive", opt="median", plot=F )
ncomp.rplsr<-sales.rplsr.dcv$optcomp
sales.rplsr<-prm(train.l$FinancialD,train.l$Response,a=ncomp.rplsr,
                 opt="median", usesvd=T)

## PRMS
sales.prms<- my.prmsCV(V1~.,data=as.data.frame(cbind(
        train.l$Response,
        train.l$FinancialD)), as=seq(1,15,1), fun = "Fair",
        probp1 = 0.95, hampelp2 = 0.975,hampelp3 = 0.999,
        center = "median", scale = "qn", usesvd = FALSE,
        numit = 100, prec = 0.01, plot=T)
```

```
#Sparse Partial Least Squares
## SPLS
spls.par<-my.spls(x=train.l$FinancialD, y=train.l$Response,
          fold=10,eta=seq(0,0.9,0.1), K=seq(1,15,1))
sales.spls<-spls(x=train.l$FinancialD, y=train.l$Response,
                 K=spls.par$K.opt, eta=spls.par$eta.opt)


## SPRMS
sales.sprms<-my.sprmsCV(V1~.,data=as.
      data.frame(cbind(train.l$Response,train.l$FinancialD)),
      as=seq(1,10,1), etas=seq(0,0.9,0.1),
      nfold = 10, fun = "Fair", probp1 = 0.95, hampelp2 = 0.975,
      hampelp3 = 0.999, center = "median", scale = "qn",
      plot = TRUE, numit = 100, prec = 0.01, alpha = 0.15)
```

References

G. E. P. Box and G. M. Jenkins. *Time Series Analysis forecasting and control.* Holden-Day, San Francisco, 1976.

H. Chun and S. Keles. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society*, 72:3–25, 2010.

D. Chung, H. Chun, and S. Keles. *spls: Sparse Partial Least Squares (SPLS) Regression and Classification*, 2013. URL `http://CRAN.R-project.org/package=spls`. R package version 2.2-1.

J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods.* Oxford University Press, New York, 2001.

P. Filzmoser. Klassifikation und Diskriminanzanalyse. Skriptum zur Vorlesung, Technische Universität Wien, Wien, 2013.

P. Filzmoser, B. Liebmann, and K. Varmuza. Repeated double cross validation. *Journal of Chemometrics*, 23, 2009.

P. Filzmoser and K. Varmuza. *chemometrics: Multivariate Statistical Analysis in Chemometrics*, 2014. URL `http://CRAN.R-project.org/package=chemometrics`. R package version 1.3.9.

T. J. Hastie, R. J. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, New York, 2^{nd} edition, 2008.

I. Hoffman, S. Serneels, P. Filzmoser, and C. Croux. Sparse partial robust M regression. *Chemometrics and Intelligent Laboratory Systems*, 149, 2015.

R. J. Hyndman, G. Athanasopoulos, D. Razbash S. Schmidt, Z. Zhou, Y. Khan, C. Bergmeir, and E. Wang. *forecast: Forecasting Functions for Time Series and Linear Models*, 2015. URL `https://cran.r-project.org/package=forecast`. R package version 5.9.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

C. P. Kindleberger and R. Aliber. *Manias, Panics and Crashes.* John Wiley & Sons, Hoboken, 5^{th} edition, 2005.

H. Li, X. Jiang, and Z. Li. Robust estimation in Gaussian filtering for engineering surface characterization. *Precision Engineering*, 28:186–193, 2004.

G.S. Maddala and I.M. Kim. *Unit Roots, Cointegration and Structural Change.* Cambridge University Press, Cambridge, 1998.

H. Madsen. *Time Series Analysis.* Chapman & Hall, London, 2007.

B. H. Mevik, R. Wehrens, and K. H. Liland. *pls: Partial Least Squares*, 2013. URL `http://CRAN.R-project.org/package=pls`. R package version 2.4-3.

G. Petris. *dlm: Bayesian and Likelihood Analysis of Dynamic Linear Models*, 2014. URL `http://CRAN.R-project.org/package=dlm`. R package version 1.1-4.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* Wiley-Interscience, New York, 1987.

W. Scherrer. Grundlagen der Ökonometrie. Skriptum zur Vorlesung, Technische Universität Wien, Wien, 2012.

W. Scherrer. Stationäre Prozesse und Zeitreihenanalyse. Skriptum zur Vorlesung, Technische Universität Wien, Wien, 2014.

S. Serneels, C. Croux, P. Filzmoser, and P.J. Van Espen. Partial robust M-regression. *Chemometrics and Intelligent Laboratory Systems*, 79, 2005.

S. Serneels and I. Hoffman. *sprm: Sparse and Non-Sparse Partial Robust M Regression*, 2013. URL `http://CRAN.R-project.org/package=rgl`. R package version 0.93.991.

R. J. Shiller. *The Subprime Solution: How Today's Global Financial Crisis Happened, and What to Do about It.* Princeton University Press, Princeton, 2008.

R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications, With R Examples.* Springer, New York, 2011.

K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics.* Taylor & Francis - CRC Press, Boca Raton, FL, 2009.