LASSO-Based Forecasting of Financial Time Series on the Basis of News Headlines

Adrian Waltenrath*

1 Introduction

In this paper, I carry out an interdisciplinary approach which is rather new and has drawn increasing attention recently. Since, theoretically, news articles contain all relevant information affecting stock prices, it is reasonable to use them to predict stock market volatility. All events causing stock price movements should be reflected in a news story and therefore be incorporated in the data. A challenge lies in the processing of news from their textual form to numerical quantities, which can be handled by mathematical methods to detect potential relations.

This approach contains elements from different fields of study such as finance, econometrics and computer sciences. Two subfields of the latter are especially interesting in this context: natural language processing and machine learning. Natural language processing generally deals with the processing of human language in its spoken or textual form by computers, while machine learning addresses the development of methods to categorize observations and/or recognize patterns for prediction. As pointed out by Varian (2014), machine learning offers a variety of methods which can be beneficial to econometricians in related applications. Although the focus of this paper lies on the econometrics involved, I borrow and apply methods developed by the other fields.

From the perspective of a computer scientist, the given problem is tackled in a rather basic way: I use headlines to predict market volatility by simply counting the number of occurrences for single words-stems over time. This means that I calculate the *term frequency* (TF), while term, in this paper, refers to a single word-stem.[1] Using single

*Before entering the master's programm of the University of Bonn, Adrian Waltenrath earned a B.Sc. in Economics from the University of Mannheim. He received his M.Sc. in Economics in fall 2015 and is now employed at DZ BANK AG.

[1] Since every field of study has its own special vocabulary it is not always trivial to please all of them. The expression *term frequency* (TF) is commonly used in relevant literature to describe the number of occurrences in a document or a set of documents.

words implies dividing texts into collections of words without taking into account any linguistic connections between them. This technique is common in relevant literature and called *bag-of-words* or *1-grams*. The reason to focus solely on headlines follows the argumentation that they are more to-the-point than news stories as they have a higher proportion of signal words, which might have explanatory power (Peramunetilleke and Wong 2002, Huang et al. 2010 and Nassirtoussi et al. 2015).

2 Literature Review

Due to the limitation in space and since most of the research done on this topic tackles the problem from a different perspective than I intend to, I cease from an extensive review. Detailed reviews are provided by Nassirtoussi et al. (2014), Hagenau, Liebmann, and Neumann (2013) as well as Nikfarjam, Emadzadeh, and Muthaiyah (2010). In addition, an overview can be found in appendix 8.1

3 Data Description

3.1 VIX

Data on opening and closing prices are taken from Datastream for every trading day from Jan 1, 2014 to Jul 31, 2015. The CBOE Volatility Index (VIX) measures the implied volatility of S&P 500 and is computed by the Chicago Board Options Exchange. It is quoted in percentage points and intends to estimate the annualized expected volatility of the S&P 500 within the next 30 days.[2]

3.2 News

News headlines are taken from *The New York Times*. They are gathered via the *New York Times Article Search API*[3]. To reduce noise I only gather news articles belonging to the sections *World*, *Business*, *Business Day* and *U.S.* I gather data

[2] Further information on the construction of the VIX can be accessed via http://www.cboe.com/micro/vix/vixintro.aspx. Detailed information aubout the S&P 500 can be found in the methodology document available under http://us.spindices.com/indices/equity/sp-500.

[3] The New York Times (2015): http://developer.nytimes.com/docs/read/article_search_api_v2.

from Jan 1, 2014 to Jul 31, 2015[4]. In total 243 750 articles are collected.[5]
As mentioned before, the news have to be processed in a way that maps words into numbers. To deal with the structure of natural language, a few more steps have to be performed, aiming to keep noise at a minimum: First, all headlines are converted to lower case. Second, all special (meaning all non-alphabetic) characters are deleted. The remainder can be viewed as a vector of lower-case words for every headline. From this vector I remove all so-called stopwords. The list of stopwords applied in my analysis is taken from the R-package *tm* (Feinerer and Hornik 2015) and is presented in Appendix 8.2. Next a stemming algorithm is applied. A common choice is the algorithm created by Porter (1980) which maps words back to a stem by applying transformations to the words suffix. This algorithm is commonly used in literature and has proven to work reasonably well. The algorithm is applied via the R-package *SnowballC* (Bouchet-Valat 2014), an example of its performance is shown in the appendix (reproduced online) .
Finally, the stemmed words were counted.[6] In total, there exist 47 023 different stemmed words, leaving me with the same number of potential predictors to include in the model. Since I try to forecast the closing price prior to market opening and trading hours are from 9:30 a.m. to 4:00 p.m. Eastern Time[7], I treat the period from 9:30 a.m. (the day before) to 9:29 a.m. as one time interval.

3.3 Holidays and Weekends

Holidays and weekends are ignored. That means I always use the news released within 24 hours prior to market opening, no matter if the prior day has been a trading day or not.[8]

[4] From Jan 1, 2014 onwards news releases increase heavily due to the inclusion of additional sources.

[5] Headlines are not always unique. Sometimes an update on the news is performed, leading to a repost of the same news. In addition, there are recurrent articles, each time having the same headline and different content. In order to avoid noise, articles whose exact headline occurs ten or more times over the whole period are deleted. In addition, I delete news if the same headline already occurred in a period of seven days prior to the news-release.

[6] Tables presenting the most frequently occurring words as well as empirical quantiles of frequencies are provided in Appendix 8.4.

[7] UTC−5 in winter and UTC−4 in summer. As illustrated in Appendix 8.3, time originally measured in Coordinated Universal Time (UTC) is converted to Eastern Time (ET) while the news were processed.

[8] This includes the assumption that news lying more than 24 hours in the past are already fully incorporated in the opening price, whereas news from the past 24 hours are assumed to have some predictive power for the performance over the upcoming trading day.

4 Methodology

4.1 LASSO

When trying to estimate the impact of the different (stemmed) words on financial time series, a high-dimensional problem is created. Reducing dimensions to a moderate number of explanatory variables that can be assumed to have predictive power is crucial in my analysis. This is done by applying different variations of the LASSO (*least absolute shrinkage and selection operator*): The standard LASSO which was proposed by Tibshirani (1996), the relaxed LASSO by (Meinshausen 2007) as well as the adaptive LASSO by Zou (2006). The adaptive LASSO is carried out in two variations. One uses the first stage estimates as weights during the second step (aLASSO-L), the other uses OLS estimates calculated on the subset which is selected by the first stage LASSO-regression (aLASSO-O). In Addition I analyse the performance of a simple OLS forecast based on the subset selected by the first stage LASSO-regression (LASSO-OLS). The mathematics of these procedures are described in more detail in appendix 8.5.

4.2 Parameter Selection: Cross-Validation

It is a common approach to determine the tuning parameters by cross-validation (CV). CV in general is considered, e.g., in Hastie, Tibshirani, and J. Friedman (2009) and Arlot and Celisse (2010). Given the time series character of the data at hand, its application is not trivial. The topic of CV in a time-series environment with dependent data is extensively studied by Bergmeir and Benítez (2012). Although they do not find any practical issues with standard k-fold CV, they suggest to use a blocked form of k-fold CV and to additionally control for stationarity. In my analysis non-stationarity is not an issue, since all variables are assumed to be stationary. This assumption is supported by performing augmented Dickey-Fuller tests[9] (ADF-tests) on the VIX-returns. In addition, ADF-tests are performed for the ten most frequent word stems as well as for ten more words, which are randomly drawn. All tests reject non-stationarity at 1% such that the assumption of stationarity is justified. Following Bergmeir and Benítez (2012) I implement a blocked form of k-fold CV, while dropping 20 observations at the borders of each training set. This is done to obtain approximate independence between folds. They argue that the presented method makes full use of the data[10], while – by retaining the time-series structure –

[9] Tests are performed with seven lags. This is the lag length chosen by default by the R-package *tseries* (Trapletti and Hornik 2015), which is used to compute the test-statistics.

[10] In contrast to the use of a single block as testing set. This is another method considered by Bergmeir and Benítez (2012).

delivering robust error estimates. The implemented procedure is outlined in detail in the appendix 8.6.

Choosing the Cross Validation Parameter: Bias-Variance Trade-off

It is widely known that, when applying k-fold CV, there exists a trade-off between bias and variance as a small k gives upward biased error estimates possessing a low variance, whereas a large k reduces bias at the cost of a higher variance (c.f. Hastie, Tibshirani, and J. Friedman 2009 or James et al. 2013). Leave-one-out CV (LOOCV) with $k = N$ delivers unbiased error estimates but suffers from high variance and thus possibly leads to a poor choice of λ. In addition, LOOCV is computationally intense since the model is fitted $k = N$ times on each of the training sets. By choosing a smaller value for k, the computational burden is reduced proportionally.

The variance of the error estimates increases in k because of the increasing similarity of the training sets: If k is chosen large, less observations are removed for the construction of the training sets, which leads to greater overlapping between any two training sets. Therefore, as k approaches N, the estimated models become very similar and the CV-error is computed as the average over positively correlated quantities and hence possesses a higher variance than the average computed from less-correlated quantities. As pointed out by Hastie, Tibshirani, and J. Friedman (2009), common choices are $k = 5$ and $k = 10$ since they have shown to provide a reasonable balance between bias and variance in empirical applications.

In fact, the number of folds is crucial in my analysis as the tuning parameter is extremely sensitive to the assignment of the folds. I therefore pay serious attention to the selection of k in Section 6.1.

4.3 Model Setup

Forecasts are computed for each trading day from Jan 1, 2015 to Jul 31, 2015, which results in 146 predictions. The model is re-estimated at each prediction date using all observations of the previous 12 months. Depending on the trading days, this gives a database of 250 to 252 observations to estimate the model on. The model horizon of 12 months as well as the prediction horizon are chosen rather arbitrarily. Nevertheless, given the fact that, becuse of the smaller amount of data, it is not feasible to make use of the news before Jan 1, 2014, I argue that this is a reasonable choice. In Section 7, I discuss this choice critically.

Re-estimating the model for each prediction date is computationally very intense but necessary since, due to the instability of the tuning parameter, it is not appropriate to estimate the model just once and apply the same model over the whole prediction horizon.

In this case, one lucky (or unlucky) result for the optimal tuning parameter could heavily bias the analysis. In addition, I argue that the focus of financial markets changes over time such that the predictive power of some features is changing as well. It is thus necessary to continuously re-fit the model.

5 Simulation

To verify that the presented methods can detect a small set of meaningful variables within a huge amount of noise, I conduct a simple simulation which is shown in the appendix (reproduced online). In short, these results show that the proposed methods can indeed detect the majority of true predictors within a vast amouunt of noise. However, the selected models are too large as they also pick some of the noise variables. Still, as most coefficients estimated for these noise variables are small and alternate around zero, the estimated models are expected to have some predictive power.

6 Results

6.1 Choosing the Cross Validation Parameter: Sensitivity of the Tuning Parameter

As mentioned in Section 4.2 the optimal tuning parameter is sensible to the assignment of the folds and therefore to the value chosen for k. Shifts in the fold assignment can lead to very different results. The problem of the instability of the LASSO procedure for $p \gg N$ is assessed by Zhang and Yang (2015) as well as Roberts and Nowak (2014). Their recommendations are not applicable in the context of blocked CV but briefly discussed in Section 7. Zhang and Yang (2015) state that, in highly instable cases with $p \gg N$, bias increases severely for small k, while variance decreases monotonically. They argue that choosing $k \in \{5, 10\}$ as a general rule can be misleading and find that, in these cases, $k \leq 10$ can perform significantly worse than LOOCV, i.e. $k = N$.
Another perspective, which should also be taken into account, is the available number of observations. Since I deal with a relatively small dataset (250 to 252 observations), it might not be adequate to choose a small k because this results in training sets which are considerably smaller than the set the model is finally estimated on. As described in Hastie, Tibshirani, and J. Friedman (2009, p. 243), the choice of k depends on the *learning curve* of the model. It would be adequate to choose $k = 5$ if the model estimated on 200 observations performed nearly as good as the model estimated on 250 observations. On the other hand, if the model performed quite

poor for 200 observations, but notably gained performance from the additional 50 observations, it would be appropriate to choose a large k. The drawback is that a larger k comes with a higher variance. To assess this issue for the application at hand, I take a look at the CV-error-curves for different k. Appendix 8.10 shows error curves from the (first stage) standard LASSO procedure for a subset of randomly drawn dates. Theoretically, one would expect the error curves to be lower with increasing k since the bias decreases. In turn, error curves are expected to be instable for high k due to the increasing variance. The Figure partly confirms these expectations. Bias seems to drop for $k > 5$, whereas it is hard to detect any decrease for $k > 20$. Since none of the curves are highly instable, I argue that $k = 20$ provides a reasonable balance between bias and variance at a moderate level of computational costs. This is in line with the findings of Zhang and Yang (2015). In addition – since I deal with a relatively small number of observations – models are estimated on a considerably larger database as for $k \in \{5, 10\}$. Obviously, the presented error curves only constitute a small fraction of the 146 prediction dates. The remaining error curves look similar and allow for the same interpretation.

6.2 Empirical Results

In this section, I present the results obtained when the analysis described in the previous chapters is carried out to forecast the VIX. All results presented are computed under the *mean absolute error* (MAE) loss, the corresponding results for *mean squared error* MSE are shown in the appendix. When comparing results for MAE and MSE, MAE seems to prevail. This might originate from the fact that the MAE weighs small and large deviations equally and is thus less affected by outliers. Such outliers can be observed in case of some event whose market impact dominates all other effects. This scenario is not unlikely for the given application of stock price volatility. Following this argumentation, using the mean absolute error is considered the better choice since it is robust to these situations.

Since the proposed approach uses the *bag-of-words* technique, it is not able to capture any semantics. Take, for example, the word *sanction*, which can have a positive or negative impact depending on its context (whether sanctions are tightened or eased). In any case, *sanction* is expected to cause volatility. 5.1 shows the proportion of correct directions as well as the hypothetical profit achieved, when investing according to the predicted directions at the opening price and evening out the position each day at the closing price. Note that this profit is purely hypothetical since it is not possible to directly invest in the VIX.[11] Still, it helps to evaluate the

[11] The VIX is indirectly investible via VIX-futures or via buying/selling options on the S&P 500. Both strategies do not produce a perfect correlation with the VIX. In addition, VIX futures possess a negative roll yield which causes additional costs. Constructing an option-based strategy is also non-trivial and beyond the scope of this paper.

Table 5.1: Results under MAE

	Proportion of Correct Directions				Hypothetical Profit in %			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	63%	62%	63%	61%	134.5	114.6	137.5	97.7
LASSO-OLS	62%	61%	62%	62%	122.7	90.3	128.5	133.4
rel. LASSO	63%	62%	60%	62%	123.9	98.1	111.3	123.0
aLASSO-O	64%	61%	62%	60%	136.5	90.3	131.4	112.5
aLASSO-L	62%	60%	61%	62%	113.7	88.1	122.0	97.1

prediction system since in contrast to the proportion of correct directions it is not purely binomial.

The system predicts the correct direction in at least 58% percent of the cases. The hypothetical profit is positive but, as explained, can never be achieved in practice. Theoretically, a long-term investment in the VIX generates a performance of -31.76% over the whole horizon. Buying the VIX each morning and selling it in the evening yields -142.04%. A naive trader, short-selling the VIX, could therefore generate a profit of 31.76% from a long-term investment and 142.04% from investing repeatedly each morning. She would be correct in the sense of directions in 64% of the cases. As it can be seen in 5.1, the proposed system can outperform the long-term investment but hardly beats the 64% achieved by the naive trader.

The estimated model sizes are presented in 5.2. Introducing a threshold or investing only if a non-degenerate model is estimated does not improve performance. Nevertheless, I take a closer look at the performance of the non-trivial models[12]. The

[12] Tables illustrating the performance for investing with a threshold of 0.8% are presented in Appendix 8.11.

Table 5.2: Model Size (MAE)

	5 Folds	10 Folds	20 Folds	40 Folds
std. LASSO	3.56 - 55/146	3.29 - 59/146	3.32 - 59/146	3.11 - 59/146
LASSO-OLS	3.56 - 55/146	3.29 - 59/146	3.32 - 59/146	3.11 - 59/146
rel. LASSO	3.54 - 55/146	3.20 - 54/146	3.32 - 59/146	3.11 - 59/146
aLASSO-O	3.03 - 53/146	2.98 - 52/146	3.03 - 58/146	2.86 - 57/146
aLASSO-L	3.16 - 55/146	3.02 - 53/146	3.06 - 58/146	2.87 - 57/146

This table summarizes the estimated model size for different k. The first value corresponds to the average number of non-zero coefficients (including the intercept). The value after the minus sign shows the number of times a non-trivial model (with at least one additional predictor) is estimated. 146 is the length of the prediction horizon.

Table 5.3: Results under MAE for Non-Degenerate Models

	Proportion of Correct Directions				Hypothetical Profit in %			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	64%	63%	63%	56%	46.3	30.3	57.4	17.4
LASSO-OLS	62%	59%	59%	58%	34.5	6.0	48.4	53.1
rel. LASSO	64%	59%	56%	58%	35.6	9.0	31.2	42.7
aLASSO-O	66%	60%	59%	53%	60.6	15.3	47.4	37.7
aLASSO-L	62%	57%	57%	58%	25.5	0.2	37.9	22.3

inferior performance is not surprising since the non-degenerate model invests in less than half of the trading days. In addition, the degenerate model always recommends a short position[13], which is correct in the majority of cases. In terms of correct directions, the non-degenerate models perform worse than the 64% achieved by the naive trader for most specifications. Still, it can beat the benchmark of a long-term investment in some cases. For further analysis, I focus on one of the presented specifications. Although much randomness is involved, I argue that the aLASSO-O method performs well over all considered k. In addition, aLASSO-O yields a good performance during the simulation.

As pointed out in Section 6.1, $k = 20$ provides a reasonable balance between bias and variance. Although the choice of $k = 5$ gives better results for the VIX, I stick to $k = 20$ as this choice is better founded and expected to suffer from less instability. The good results for $k = 5$ are suspected to be coincidental. The choice is in line with the findings of Zhang and Yang (2015), who investigate CV in the context of the LASSO for the case of $p \gg N$.

6.3 Predictors and Estimated Coefficients

In this section, I take a closer look at the estimation results of the aLASSO-O method with $k = 20$ under the mean absolute error loss. 5.1 summarizes all predictions made by this system. It also illustrates whether predictions are based on a degenerate model and whether the direction is predicted correctly. Interestingly, at each prediction date before May 1, 2015 a degenerated model is estimated. Keeping in mind that estimation is always carried out on the last 12 months prior to the prediction date, this cannot be led back to an increasing database. Instead, it implies that during the first four months the system is not able to detect any meaningful predictors from the given database. Looking at the performance of the VIX, which is presented in Appendix 8.12, does not show any peculiarities in the dependent variable that could have dropped out or joined the database around May 1, 2014 or May 1, 2015,

[13] It always predicts a value in $[-0.8, -0.1]$.

respectively. Instead, this has to be interpreted as the result of a process. Possibly, at that point, enough information about some topic(s) joined the database such that a pattern is recognized and a non-degenerated model is estimated.

It is also possible that the effects of topics change over time, which lowers the predictive power of the corresponding feature and makes it harder to detect a pattern. Take, for example, the stem *ukrain*: Surprisingly, a negative impact of this feature on volatility is estimated if it is included in the model. Nevertheless, the impact of the corresponding news obviously heavily depends on the context and for sure has not always been negative over the last year. It is likely that at the beginning of the Ukraine crisis the stem *ukrain* was a driver of volatility and that it adopted a calming affect in the recent past as the crisis passed its climax such that news were reducing, rather than causing, uncertainty.

According to this argumentation, a feature's impact can depend on the context such that it cannot easily be detected by the given approach. This is a drawback and discussed in Section 7. Another issue is the twelve-month calibration interval, which is a long horizon in fast-moving financial markets. I again refer to Section 7, where I discuss this issue in detail.

So far, I have only investigated one feature, namely *ukrain*. 5.4 shows the number of times each feature is included in the model along with some more detailed information. Additionally, 5.2 illustrates the estimated coefficients for the eleven most frequent features over time. The feature which is most often included in the model, is *obama*

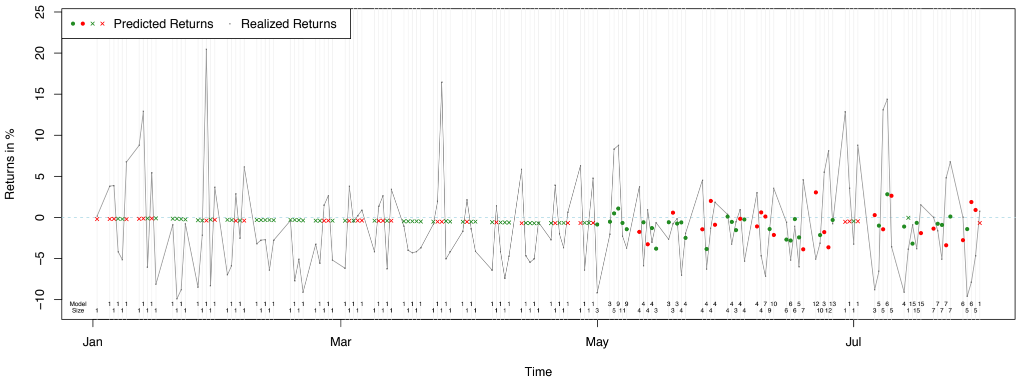

Figure 5.1: This figure shows the true realized returns (black) of the VIX along with the predictions (red/green) created by the aLASSO-O procedure for $k = 20$ under the absolute error loss. Colors indicate whether the direction is predicted correctly. Predictions are marked with dots if they come from a non-trivial model. An X is drawn if the underlying model is degenerated. In addition, at the bottom of the figure, the size of the estimated model is presented. Vertical lines are drawn at each date a model is estimated.

Table 5.4: Frequency of Features and Summary of Corresponding Coefficients

Feature	Freq.	Pos.	Neg.	Min.	Max.	Avg.	Corresponding Words
(Intercept)	146	9	137	−3.51	0.50	−0.84	
obama	58	0	58	−0.25	−0.13	−0.19	obama, obamas
sanction	49	49	0	0.25	0.47	0.4	sanctions, sanction, sanctioned, sanctioning
report	36	36	0	0.11	0.28	0.23	report, reports, reported, reporting, reporter, reporters
leader	25	25	0	0.31	0.45	0.38	leader, leaders
ukrain	18	0	18	−0.18	−0.06	−0.13	ukraine, ukraines
china	17	17	0	0.04	0.18	0.13	china, chinas
iran	17	0	17	−0.16	−0.01	−0.05	iran, irans
crash	14	14	0	0.05	0.12	0.09	crash, crashes, crashing, crashed
gaza	14	14	0	0.08	0.19	0.12	gaza, gazas
iraq	12	0	12	−0.24	−0.08	−0.18	iraq, iraqs
greek	10	0	10	−0.17	−0.10	−0.15	greek, greeks
death	5	0	5	−0.19	−0.15	−0.17	deaths, death
mai	5	0	5	−0.32	−0.29	−0.31	may, mais, mays
polic	5	0	5	−0.10	−0.07	−0.09	police, policing, polices
deal	3	0	3	−0.07	−0.07	−0.07	deal, deals, dealings, dealing
japan	3	0	3	−0.22	−0.21	−0.21	japan, japans
vote	3	0	3	−0.13	−0.12	−0.12	vote, votes, voting, voted
cuba	1	0	1	−0.17	−0.17	−0.17	cuba, cubas
take	1	1	0	0.40	0.40	0.40	takes, take, taking
u	1	0	1	−0.03	−0.03	−0.03	us, u

This table shows the number of times (stemmed) words are included in the model. It also shows the number of times the estimated coefficients are positive or negative. In addition, it shows the maximum, minimum and average of all coefficients for each word.

and has a negative impact on volatility. This is not immediately intuitive as a U.S. president's actions or wording could have severe effects on financial markets. On the other hand, Obama as well as the American government is certainly not interested in highly volatile markets, especially not when facing a period of fragile economic growth. It is therefore reasonable that he might have chosen his actions and wording to reduce uncertainty, enforcing markets to stay calm. According to the presented results this has been successful to some extent.

Other features with negative coefficients are *ukrain* and *greek*. As pointed out before, this has not been expected beforehand. The negative coefficients suggest that news including the corresponding words were reducing uncertainty rather than containing

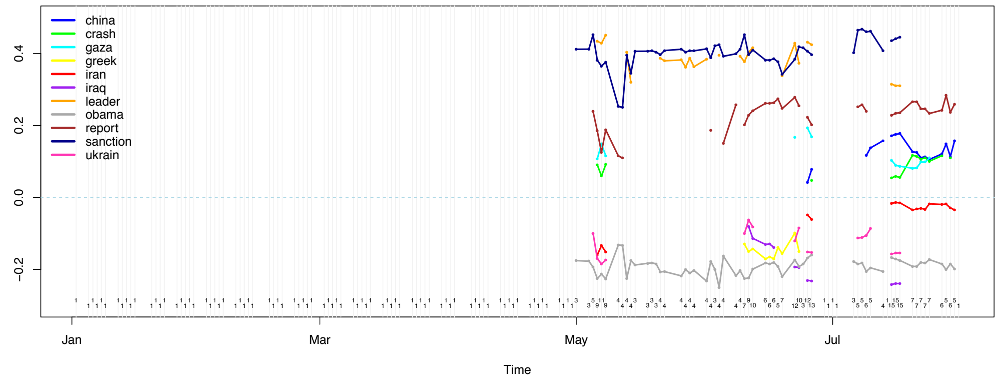

Figure 5.2: This figure illustrates the coefficients for the eleven most frequent features estimated by the aLASSO-O procedure for $k = 20$ under the absolute error loss. At the bottom of the figure the size of the estimated model is presented. Vertical lines are drawn at each date a model is estimated.

new, unanticipated information. However, both features are not persistently included in the model, such that one should not give too much credit to this interpretation. As explained above, the impact of those features on volatility is probably very context dependent. This might contribute to the poor prediction performance.

Among others, positive coefficients are estimated for the features *sanction*, *report* and *leader*. The positive coefficients of *sanction* are plausible and most likely connected to the Ukraine crisis and the sanctions which were introduced by various countries against the Russian government. Sanctions – no matter if they are introduced, tightened or eased – should cause volatility as they raise uncertainty about their economic impact on all involved countries. They should therefore have a positive impact on volatility.

The other features are harder to interpret. The feature *report* corresponds to a greater number of words. It includes forms of the verb *to report* as well as the nouns *reporter*, *reporters* and *report*. Here, another drawback of the approach arises. The words *reporter* and *reporters* do not fit to the other words that can be linked to the release of a financial report or some economic key figures, causing the positive coefficient. The words *reporter* and *reporters* are expected to be mentioned in a different context such that noise is caused.

At first sight, the feature *leader* is puzzling. Looking at some corresponding headlines shows that these words are often used for the leader of some country or organization like for instance *Iran Leader*, *E.U. Leaders* and *Greek Leader*. Thus, *leader* typically refers to an influential person or groups of persons, who can affect financial markets by their actions and statements and therefore cause volatility. Interestingly, *leader* in general seems to cause volatility, whereas *obama* reduces it. Some reasoning for this has been given above when assessing the intention of the American government. Since the remaining features are less persistent and/or possess smaller coefficients, I cease from a detailed interpretation. In general, a positive coefficient indicates that the topic has been a driver of volatility and that the corresponding news raised uncertainty. In turn, a negative coefficient implies a calming effect such that news containing the corresponding words reduced uncertainty and did not contain much unexpected information. As mentioned above, one should keep in mind that the coefficients are estimated on data from the past 12 months, such that recent shifts in the impact of features can hardly be reflected by the coefficients.

7 Criticism and possible Extensions

At various points I have been referring to this section, in which I want to discuss the approach critically. Due to the approach's interdisciplinarity there exists a wide range of possibilities for modifications and improvements.

First and foremost, there are two factors which are key in my opinion: The prediction interval and the horizon the model is calibrated on. Both characteristics are obviously connected and cannot be chosen independently as the length of the prediction interval affects the number of observations in the calibration horizon.

In my opinion, the poor prediction performance likely originates from the long prediction interval and the large model horizon. Financial markets are fast-moving and twelve months seem to be a long time for patterns to persist. Also, markets, especially those as liquid as the S&P 500, are fast in processing new information such that a prediction interval from market opening to market closing is probably not adequate. The same holds for the interval that allocates the headlines to the prediction dates. 24 hours seem to be a lot if markets are fast-moving. It is likely that news released in the beginning of the interval are already fully incorporated in the opening price.

Therefore, it would be interesting to repeat the analysis with smaller intervals, such as 1 hour or even 30 minutes, while taking the news released during the prior interval as independent variables. This modification yields up to 13 observations per trading day. In turn, it drastically decreases the number of news falling into each of the intervals such that it would be necessary to find another source which is providing more frequent news releases[14]. This increase in observations per day allows to shorten the prediction horizon drastically. In this paper, the model was calibrated on a 12-month-horizon with 250 to 252 observations. With 13 observations per day, twice the amount can be reached by using data from the prior two months. This larger number of observations could be beneficial for the estimation of the

[14] A candidate could be the news-ticker provided by Bloomberg terminals as it is customizable and delivers real-time news from various sources.

tuning parameter and improve stability. In addition, the shorter calibration horizon is expected to better reflect the fast-moving character of financial markets, where focus can move rapidly and topics' impacts might change over time. Whether such a system is able to yield a better prediction performance is still to be investigated.
Next, I turn to the instability of the CV procedure. Some research has already addressed this topic. For one, there exists the one-standard error rule (Hastie, Tibshirani, and J. Friedman 2009, p. 244), according to which one should choose the largest λ whose CV-error lies within one standard deviation of its minimum. The intention is to choose the simplest model possessing an accuracy that is comparable to the best model. Unfortunately, for the given approach, this procedure is unrewarding as it always leads to a degenerate model. This implies that the training-set performance of the chosen model is never significantly better than the performance of a degenerate model. From that finding the predictive power of the given system can be questioned. As pointed out before, this can hopefully be solved by adjusting prediction and calibration horizons.

Another workaround has been proposed by Roberts and Nowak (2014). Since they do not assume any time-series context, their approach relies on standard CV where folds are assigned randomly. They introduce a procedure called *percentile-LASSO*, which repeatedly performs CV to get a distribution of the optimal λ and takes a particular percentile from that distribution for LASSO estimation. Similar to the one-standard error rule, they state that the models chosen by CV tend to be too large and show that choosing values grater or equal to the 0.75-percentile can improve performance. They suggest to use the 0.95-percentile, which is supported by their simulations. According to Roberts and Nowak (2014), this approach can also be implemented together with the one-standard error rule.

Zhang and Yang (2015) investigate CV for model selection. They state that choosing k according to a general rule can be misleading and find that in highly instable cases with $p \gg N$, choosing $k \in \{5, 10\}$ can lead to a significantly larger CV-error than LOOCV. Overall, they state that LOOCV and repeated k-fold CV[15] with $k = 20$ or $k = 50$ perform best in this context. They conclude that since k-fold CV can be instable, repeated k-fold CV seems most promising for prediction.

The just described procedures by Roberts and Nowak (2014) and Zhang and Yang (2015) cannot easily be implemented in this paper since the blocked form of CV, which is used to reflect the time-series character of the data, does not involve randomness. As pointed out by Bergmeir and Benítez (2012), standard CV – although theoretically less adequate – also works well in time-series contexts such that one could cease from using the blocked form and instead implement standard CV. Then, one of the just proposed modifications could be implemented. This may improve stability and lead to better prediction performance at the cost of a theoretically less accurate estimation of λ^*. Nevertheless, if gains obtained from improved stability

[15] Repeated k-fold CV chooses the optimal λ such that it minimizes the CV-error over all repetitions.

prevail, this can lead to an estimate of λ^* which performs better in practice. Whether this holds obviously depends on the application at hand and cannot be answered in general. An appropriate simulation reflecting the particular application could help to give a recommendation.

Another starting point for modifications is the processing of news. As discussed in Section 6.3, a major drawback is the approach's inability to detect semantics. Even the incorporation of simple semantics could severely improve performance since the impact of words can depend heavily on their context. There exist some simple methods that can provide this improvement. In this paper, single words (*1-grams*) are taken as separate features. Correspondingly, possible alternatives are *2-grams* or *3-grams*, which take two or three subsequent words as single features such that features correspond to short expressions. The drawback of these alternatives is that they heavily increase the number of potential predictors as there exist much more three-word combinations than single words. In addition, a particular three-word combination is generally observed less often than a single word such that the majority of features will contain mostly zeros. Other techniques that can be used are *two-word combinations* and *noun phrases*. They are described in more detail, e.g., in Hagenau, Liebmann, and Neumann (2013). Also, one could consider the use of a dictionary to identify features or to capture news-sentiment. Especially the latter is interesting as it approaches the prediction problem from a different perspective.

Lastly, one could not only process the headlines. Instead, one could also use the news-body and group news according to their topics. This can be done by using a topic model such as *latent dirichlet allocation* (LDA), which was proposed by Blei, Ng, and Jordan (2003). Topic models estimate the probabilities of a particular text to belong to a number of prespecified topics[16]. The estimated frequency of news in each topic can then be used to predict some financial time series. In addition, some topics can be manually discarded by the researcher if they are considered to be irrelevant for forecasting. This is especially helpful if the database contains general news rather than news already focusing on financial topics. When dealing with general news, one could apply LDA two times. Once, with a small number of topics, to identify relevant news and a second time to identify different subjects within the group of relevant news. I find it promising to investigate whether this approach is better suited to explain stock market volatility.

Using a different feature representation can also be considered. A basic overview over possible feature representations is given in Nassirtoussi et al. (2014). In particular, using the multiplier of *term frequency* and *inverse document frequency* TF×IDF for measurement seems interesting as it takes into account that some words are more common than others.

To summarize, there exist a lot of starting points for potential improvements. Further research is needed to investigate whether some of the suggested modifications can im-

[16] By adjusting the total number of topics one can implicitly set the scope for each topic.

prove the system's prediction performance. It remains unclear whether LASSO-based methods are suited to be applied in this context.

8 Conclusion

In this paper I have applied various LASSO-based methods to forecast stock market volatility from news headlines. It is found that the system in its proposed form cannot achieve a forecasting performance which is better than chance. Nevertheless, it yields some insights about the topics that have recently been driving financial markets. The system comes with a whole lot of possibilities for potential improvements, which have been discussed extensively in Section 7. Due to its interdisciplinarity, the approach remains both interesting and challenging at the same time.

The increasing amount of data and computational powers provide great opportunities for future text-based analyses. Due to the growing digitalization, this development is bound to accelerate such that this area will likely become even more important. As pointed out by Varian (2014), collaborations between computer scientists and econometricians are promising in this context. Especially machine learning can provide helpful tools to researchers working in this area.

Although the approach proposed in this paper does not yield any remarkable prediction performance, I do not wish to discard it. Instead, further research is needed to determine whether the suggested modifications are able to improve performance. The approach is still considered to be promising and worth investing further effort.

Bibliography

Antweiler, Werner and Murray Z Frank (2004). “Is all that talk just noise? The information content of internet stock message boards”. In: *The Journal of Finance* 59.3, pp. 1259–1294.

Arlot, Sylvain and Alain Celisse (2010). “Survey of cross-validation procedures for model selection”. In: *Statistics Surveys* 4, pp. 40–79.

Bergmeir, Christoph and José M. Benítez (2012). “On the use of cross-validation for time series predictor evaluation”. In: *Information Sciences* 191, pp. 192–213.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). “Latent dirichlet allocation”. In: *The Journal of machine Learning research* 3, pp. 993–1022.

Bollen, Johan, Huina Mao, and Xiaojun Zeng (2011). “Twitter mood predicts the stock market”. In: *Journal of Computational Science* 2.1, pp. 1–8.

Bouchet-Valat, Milan (2014). *SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library.* R package version 0.5.1. http://CRAN.R-project.org/package=SnowballC.

Burman, Prabir, Edmond Chow, and Deborah Nolan (1994). “A Cross-Validatory Method for Dependent Data”. In: *Biometrika* 81.2, pp. 351–358.

Butler, Matthew and Vlado Kešelj (2009). “Financial forecasting using character n-gram analysis and readability scores of annual reports”. In: *Advances in artificial intelligence.* Springer, pp. 39–51.

Chatterjee, A and S. N. Lahiri (2013). “Rates of convergence of the Adaptive LASSO estimators to the Oracle distribution and higher order refinements by the bootstrap”. In: *The Annals of Statistics* 41.3, pp. 1232–1259.

Das, Sanjiv R. and Mike Y. Chen (2007). “Yahoo! for Amazon: Sentiment extraction from small talk on the web”. In: *Management Science* 53.9, pp. 1375–1388.

Fama, Eugene F. (1965). “Random Walks in Stock Market Prices”. In: *Financial Analysts Journal* 21.5, pp. 55–59.

— (1970). “Efficient capital markets: A review of theory and empirical work”. In: *The Journal of Finance* 25.2, pp. 383–417.

Feinerer, Ingo and Kurt Hornik (2015). *tm: Text Mining Package.* R package version 0.6-2. http://CRAN.R-project.org/package=tm.

Frank, Ildiko E. and Jerome H. Friedman (1993). “A Statistical View of Some Chemometrics Regression Tools”. In: *Technometrics* 35.2, pp. 109–135.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent”. In: *Journal of Statistical Software* 33.1. http://www.jstatsoft.org/v33/i01/, pp. 1–22.

Groth, Sven S and Jan Muntermann (2011). “An intraday market risk management approach based on textual analysis”. In: *Decision Support Systems* 50.4, pp. 680–691.

Hagenau, Michael, Michael Liebmann, and Dirk Neumann (2013). “Automated news reading: Stock price prediction based on financial news using context-capturing features”. In: *Decision Support Systems* 55.3, pp. 685–697.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer Texts in Statistics.

Huang, Chenn-Jung et al. (2010). “Realization of a news dissemination agent based on weighted association rules and text mining techniques”. In: *Expert Systems with Applications* 37.9, pp. 6409–6413.

James, Gareth et al. (2013). *An Introduction to Statistical Learning: With Applications in R.* Springer Texts in Statistics.

Jin, Fang et al. (2013). “Forex-foreteller: Currency trend modeling using news articles”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, pp. 1470–1473.

Knight, Keith and Wenjiang Fu (2000). “Asymptotics for Lasso-Type Estimators”. In: *The Annals of Statistics* 28.5, pp. 1356–1378.

Kraemer, Nicole, Juliane Schaefer, and Anne-Laure Boulesteix (2009). “Regularized Estimation of Large-scale Gene Association Networks Using Graphical Gaussian Models”. In: *BMC Bioinformatics* 10.1.

Meinshausen, Nicolai (2007). “Relaxed Lasso”. In: *Computational Statistics and Data Analysis* 52.1, pp. 374–393.

Mittermayer, Marc-André (2004). “Forecasting intraday stock price trends with text mining techniques”. In: *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on.* IEEE.

Nassirtoussi, Arman Khadjeh et al. (2014). “Text mining for market prediction: A systematic review”. In: *Expert Systems with Applications* 41.16, pp. 7653–7670.

— (2015). “Text mining of news-headlines for FOREX market prediction: A Multilayer Dimension Reduction Algorithm with semantics and sentiment”. In: *Expert Systems with Applications* 42.1, pp. 306–324.

Nikfarjam, Azadeh, Ehsan Emadzadeh, and Saravanan Muthaiyah (2010). “Text mining approaches for stock market prediction”. In: *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on.* Vol. 4. IEEE, pp. 256–260.

Osborne, Michael R., Brett Presnell, and Berwin A. Turlach (2000). “On the lasso and its dual”. In: *Journal of Computational and Graphical statistics* 9.2, pp. 319–337.

Peramunetilleke, Desh and Raymond K. Wong (2002). “Currency exchange rate forecasting from news headlines”. In: *Australian Computer Science Communications* 24.2, pp. 131–139.

Porter, M.F (1980). “An Algorithm for Suffix Stripping”. In: *Program* 14.3, pp. 130–137.

R Core Team (2014). *R: A Language and Environment for Statistical Computing.* http://www.R-project.org/. R Foundation for Statistical Computing. Vienna, Austria.

Racine, Jeff (2000). “Consistent cross-validatory model-selection for dependent data: HV-block cross-validation”. In: *Journal of Econometrics* 99.1, pp. 39–61.

Riloff, Ellen and Janyce Wiebe (2003). “Learning extraction patterns for subjective expressions”. In: *Proceedings of the 2003 conference on Empirical methods in natural language processing.* Association for Computational Linguistics, pp. 105–112.

Roberts, S. and G. Nowak (2014). “Stabilizing the lasso against cross-validation variability”. In: *Computational Statistics & Data Analysis* 70, pp. 198–211.

Schumaker, Robert P et al. (2012). “Evaluating sentiment in financial news articles”. In: *Decision Support Systems* 53.3, pp. 458–464.

Taşcı, Şerafettin and Tunga Güngör (2013). “Comparison of text feature selection policies and using an adaptive framework”. In: *Expert Systems with Applications* 40.12, pp. 4871–4886.

Tetlock, Paul C. (2007). “Giving content to investor sentiment: The role of media in the stock market”. In: *The Journal of Finance* 62.3, pp. 1139–1168.

Tetlock, Paul C., Maytal Saar-Tsechansky, and Sofus Macskassy (2008). “More than words: Quantifying language to measure firms’ fundamentals”. In: *The Journal of Finance* 63.3, pp. 1437–1467.

The New York Times (2015). *New York Times Article Search API v2.* http://developer.nytimes.com/docs/read/article_search_api_v2.

Tibshirani, Robert (1996). “Regression Shrinkage and Selection Via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288.

Trapletti, Adrian and Kurt Hornik (2015). *tseries: Time Series Analysis and Computational Finance.* R package version 0.10-34. http://CRAN.R-project.org/package=tseries.

Varian, Hal R. (2014). “Big Data: New Tricks for Econometrics”. In: *The Journal of Economic Perspectives* 28.2, pp. 3–27.

Wiebe, Janyce and Ellen Riloff (2005). “Creating subjective and objective sentence classifiers from unannotated texts”. In: *Computational Linguistics and Intelligent Text Processing.* Springer, pp. 486–497.

Wuthrich, B. et al. (1998). “Daily stock market forecast from textual web data”. In: *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics.* Vol. 3. IEEE, pp. 2720–2725.

Zhang, Yongli and Yuhong Yang (2015). “Cross-validation for selecting a model selection procedure”. In: *Journal of Econometrics* 187.1, pp. 95–112.

Zou, Hui (2006). “The Adaptive Lasso and Its Oracle Properties”. In: *Journal of the American Statistical Association* 101.476, pp. 1418–1429.

Zou, Hui, Trevor Hastie, and Robert Tibshirani (2007). “On the “degrees of freedom” of the lasso”. In: *The Annals of Statistics* 35.5, pp. 2173–2192.

Online Appendix to: LASSO-Based Forecasting of Financial Time Series on the Basis of News Headlines by Adrian Waltenrath

8.1 Related Literature

In this section I want to give a short overview on some literature related to this paper. More extensive reviews are provided by Nassirtoussi et al. (2014), Hagenau, Liebmann, and Neumann (2013) as well as Nikfarjam, Emadzadeh, and Muthaiyah (2010). As Nassirtoussi et al. (2014), I start with briefly addressing the Efficient Market Hypothesis (Fama 1965 and Fama 1970), which states that markets reflect all relevant information at all time. According to this hypothesis, market behavior is not predictable at all. In practice, however, the Efficient Market Hypothesis does not hold since market participants seldom possess knowledge about all available information. Also, new incoming information is not processed immediately by markets but over time. Therefore, prediction is theoretically possible during that process.

The most common approach to predict movements from textual data involves the usage of classifiers like *support vector machines* (SVM) or *naive Bayes* (c.f. Hastie, Tibshirani, and J. Friedman 2009) to assign news to categories according to their market impact. Such a procedure is applied e.g. by Mittermayer (2004) who uses three categories (*Buy*, *Sell* and *Neutral*) to predict single stock movements. Usually, data is divided into a training and a validation set, whereas the elements of the training set have to be assigned to the previously defined categories. Mittermayer (2004) does this according to the market performance within 15 minutes after the news release. The classifier – a SVM in case of Mittermayer (2004) – is then calibrated on the training set and predictions are done by using the classifier on the validation set to assign the news to the class they most likely belong to. This concept is followed by the majority of research related to this topic.

Another indirect, behavioral-economic argumentation states that news do not affect stock prices directly but influence the sentiment of investors, which in turn affects demand and therefore prices. Estimating the sentiment of textual data is often referred to as *sentiment analysis* or *opinion mining*. An example for this approach is the work of Schumaker et al. (2012) who use financial news to estimate market-sentiment and try to predict future movements from an estimated sentiment score. For computing the sentiment, they use a prespecified dictionary, which assigns scores for implied (positive or negative) sentiment to each word. Making use of such a dictionary is convenient as it allows to put different weights on different features[17]. As a drawback, one relies on a decent dictionary. Schumaker et al. (2012) and others use *OpinionFinder* (Riloff and Wiebe 2003 and Wiebe and Riloff 2005), a tool to evaluate the sentiment of whole sentences. Other examples for dictionaries that can be used for sentiment analysis are the Harvard IV-4 psychosocial dictionary, used e.g. by Tetlock, Saar-Tsechansky, and Macskassy (2008), or the Google-Profile of Mood States (GPOMS), used by Bollen, Mao, and Zeng (2011). Wuthrich et al. (1998), who were one of the first to use news articles to predict financial data, make use of a dictionary of word sequences, which was especially designed by an expert. As pointed out before, prediction is mostly performed by using some kind of classification. However, when sentiment is computed as an intermediate step, there have been some regression-based approaches (Tetlock, Saar-Tsechansky, and Macskassy 2008, Bollen, Mao, and Zeng 2011 and Jin et al. 2013).

Various news sources have been used by different authors. They range from print news (e.g. Tetlock 2007 and Tetlock, Saar-Tsechansky, and Macskassy 2008) over digital news (e.g. Schumaker et al. 2012, Wuthrich et al. 1998 and Nassirtoussi et al. 2015), ad-hoc announcements (e.g. Groth and Muntermann 2011) to internet stock messages boards (e.g. Antweiler and M. Z. Frank 2004 and Das and Chen 2007) and even cover social media platforms such as twitter (e.g. Bollen, Mao, and Zeng 2011).

Previous work also differs substantially in the prediction horizon, the predicted quantity as well as the definition of a feature. The considered prediction horizon ranges from several minutes (e.g. Mittermayer 2004) to one year (Butler and Kešelj 2009), while most of the research focuses on short term predictions of less than 24 hours. The predicted quantities are typically single stocks or stock indices. Some authors also focus on volatility or currency exchange rates.

In this work, a feature corresponds to a single word-stem. This can and has been generalized to expressions containing two or more words to capture some of the syntax. However, the technique of treating each word as a separate feature is applied by the vast majority of researchers and called *bag-of-words* or *1-grams*. Some alternatives, like *noun-phrases* or *n-grams*, are briefly discussed in Section 7. At this point, I again refer to the review by Nassirtoussi et al. (2014), who provide detailed tables about the characteristics of most of the references discussed here.

Authors typically evaluate their approach by assessing the proportion of correctly predicted directions. Success ranges between 50 and 70 percent, while everything

[17] Feature is the machine-learning equivalent to explanatory variable. The term feature is more general as it can apply to regressions as well as classifiers and other techniques.

above 55 is considered to be report-worthy (Nassirtoussi et al. 2014). Often, a trading strategy is derived, allowing to evaluate the approach by the hypothetical profit achieved by this strategy. In most cases and if there is no intermediate step, such as the estimation of sentiment, feature selection and therefore dimension reduction is done using the *term frequency* (TF). This means selecting features according to their occurrences, i.e. dropping all features which occur less than a certain threshold. When using a dictionary, selection is done implicitly by assigning zero scores to all words not contained in the dictionary.

TF can also be used as feature representation like it is done in this work. The term feature representation refers to the measurement of observations, i.e. the unit variables are represented in, which in this work is the number of occurrences per day. Another method is called *inverse document frequency* (IDF), which is defined as the logarithm of the total number of documents over the number of documents containing the term. This implies putting additional weight on rare expressions by assuming that they contain more essential information. Often, as a combination, their multiplier TF×IDF is used. This measure increases in the number of times a feature appears in the document and is offset by the feature's overall frequency. It therefore takes into account that some words are more common than others (c.f. Taşcı and Güngör 2013 and Mittermayer 2004). A basic overview over possible representations is given, e.g., in Nassirtoussi et al. (2014).

In short, most of the research done on this topic tackles the problem from a different perspective than I do. To the best of my knowledge, LASSO-based procedures have not yet been applied in this context, although they are expected to provide reasonable performance when facing the problem of feature selection from a large number of possible predictors.

8.2 List of Stopwords

The following table shows the list of stopwords applied in my analysis. It is taken from the R-package *tm* (Feinerer and Hornik 2015):

i	me	my	myself	we	our	ours
ourselves	you	your	yours	yourself	yourselves	he
him	his	himself	she	her	hers	herself
it	its	itself	they	them	their	theirs
themselves	what	which	who	whom	this	that
these	those	am	is	are	was	were
be	been	being	have	has	had	having
do	does	did	doing	would	should	could
ought	i’m	you’re	he’s	she’s	it’s	we’re
they’re	i’ve	you’ve	we’ve	they’ve	i’d	you’d
he’d	she’d	we’d	they’d	i’ll	you’ll	he’ll
she’ll	we’ll	they’ll	isn’t	aren’t	wasn’t	weren’t
hasn’t	haven’t	hadn’t	doesn’t	don’t	didn’t	won’t
wouldn’t	shan’t	shouldn’t	can’t	cannot	couldn’t	mustn’t
let’s	that’s	who’s	what’s	here’s	there’s	when’s
where’s	why’s	how’s	a	an	the	and
but	if	or	because	as	until	while
of	at	by	for	with	about	against
between	into	through	during	before	after	above
below	to	from	up	down	in	out
on	off	over	under	again	further	then
once	here	there	when	where	why	how
all	any	both	each	few	more	most
other	some	such	no	nor	not	only
own	same	so	than	too	very	

8.3 Stemming

5.5 and 5.6 show some examples of the outcome.

Table 5.5: Example of Stemming

orig. word	**stem**	**orig. word**	**stem**	**orig. word**	**stem**
walk	walk	sensitiveness	sensit	controlling	control
walks		sensitivity		control	
walked		sensitization		controller	

Table 5.6: Empirical Example of Headline Stemming

Timestamp (UTC)	Original Headline
2014-05-28T01:04:10Z	China Sacks Senior Energy Official Amid Corruption Crackdown: Xinhua
2015-01-26T18:54:53Z	China's Li Says to Create 10 Million Jobs in 2015: China Daily
2015-07-20T16:54:38Z	Rates Rise at Weekly US Treasury Auction

Date	Local Time	Stemmed Headline
2014-05-27	21:04:10	china, sack, senior, energi, offici, amid, corrupt, crackdown, xinhua
2015-01-26	13:54:53	china, li, sai, creat, million, job, china, daili
2015-07-20	12:54:38	rate, rise, weekli, u, treasuri, auction

8.4 Most Frequently Occurring Words and Empirical Quantiles

5.7 shows the ten most frequently occurring words, their number of occurrences as well as the corresponding unstemmed words. Note that these are not all possible words leading to a particular stem but the words empirically found in the news headlines over the whole horizon. These words are already converted to lower case. Also special characters like apostrophes are already dropped. In addition, 5.8 shows empirical quantiles of feature frequencies. As expected, the distribution is heavily left-skewed.

Table 5.7: Most Frequently Occurring Words

Stemmed Word	Occurrences	Corresponding Words
u	17198	us[18], u
sai	12935	says, say, saying
new	9083	new, news
kill	6766	kill, killed, killing, kills, killings
china	5847	china, chinas
polic	5351	police, policing, polices
bank	4617	banks, bank, banking, bankings, banked
man	4521	man, mans, manning, mannings
ukrain	4521	ukraine, ukraines
court	4421	court, courts, courting, courted
state	4305	state, states, stated, stately
obama	4285	obama, obamas
deal	4218	deal, deals, dealings, dealing
case	3699	case, cases, casings, casing
year	3636	year, years
russia	3540	russias, russia
charg	3533	charges, charge, charged, charging, charg
plan	3519	plans, planned, plan, planning
talk	3396	talks, talk, talking, talkative, talked
eu	3257	eu, eus

[18] Since special characters are already removed *us*, represents the personal pronoun *us* as well as the abbreviation for the United Stated *U.S.*

Table 5.8: Empirical Quantiles of Frequencies

50%	60%	70%	80%	90%	92.5%	95%	97.5%	100%
1	2	3	7	28	47	91	232	17198

8.5 LASSO Procedures

Standard LASSO

The LASSO (in its standard form hereafter abbreviated as std. LASSO) is a form of penalized regression and was originally proposed by Tibshirani (1996). It is described fairly well in James et al. (2013) and Hastie, Tibshirani, and J. Friedman (2009), whereas the latter is being more precise in handling the topic than the first one. LASSO is closely related to ridge regression as both procedures belong to the group of bridge estimators. In fact, bridge estimators were originally developed by I. E. Frank and J. H. Friedman (1993) as a generalization of ridge regression and generally apply a penalty of order $q > 0$. Ridge regression and LASSO are special cases and apply a L_2 and L_1–penalty, respectively. The L_1–penalty of the LASSO results in the fact that, in contrast to ridge regression, LASSO-coefficients are not only shrunken towards but exactly to zero[19], meaning that some sort of variable selection is done implicitly. Formally, the LASSO solves the following minimization problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2 \right\} \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j|^q \leq t. \tag{1}$$

Or equivalently in Lagrangian form:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}, \tag{2}$$

with $q = 1$ to represent the L_1–penalty, $\boldsymbol{X} = [\boldsymbol{x_{\cdot 1}}, \ldots, \boldsymbol{x_{\cdot p}}]$ being the predictor matrix with $\boldsymbol{x_{\cdot k}} = (x_{1k}, \ldots, x_{nk})^T \ \forall \ k \in \{1, \ldots, p\}$, $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ being the dependent variable, N being the number of observations and p being the number of independent variables, i.e. the number of potential predictors. The parameters t and λ are often referred to as tuning or shrinkage parameters since they control the amount of shrinkage and implicitly determine the number of variables included in the estimated model. Therefore, the selection of these parameters is crucial. I deal with this in detail in Section 4.2.

The given minimization problem as well as all subsequent ones is solved using the R-package *glmnet*[20] (J. Friedman, Hastie, and Tibshirani 2010), which possesses

[19] This can be visualized nicely by a graph showing the contours of the error and constraint functions for both procedures. Such a graph is presented in all three references mentioned above: Tibshirani (1996), James et al. (2013) and Hastie, Tibshirani, and J. Friedman (2009)

[20] Note that the *glmnet* package needs an input matrix with at least two columns (excluding the intercept). Therefore, in my analysis if a model of size two (the intercept and one predictor) is

a fast performing algorithm that uses cyclical coordinate descent to successively optimize the objective function over each parameter while keeping the others fixed[21]. One drawback of the standard LASSO is that not only the coefficients of predictors which are not included in the model are shrunken (to zero), but that the remaining nonzero coefficients are also biased towards zero. This leads to the fact that the estimated coefficients in general are not consistent (Hastie, Tibshirani, and J. Friedman 2009, p. 91). Since I focus on prediction performance rather than on obtaining the correct estimates, this is not a deal-breaker. Nevertheless, it is inconvenient. An obvious workaround, which is also suggested in Hastie, Tibshirani, and J. Friedman (2009, p. 91), is to perform LASSO to identify a subset of non-zero predictors and in a second step estimate OLS on that subset. I abbreviate this procedure as LASSO-OLS. As pointed out by Hastie, Tibshirani, and J. Friedman (2009, p. 91), this not feasible if the selected subset is large. Another common approach is the so-called *relaxed LASSO*, which aims to reduce bias and therefore mitigates the problem of inconsistency. This procedure is described next.

Relaxed LASSO

The relaxed LASSO (rel. LASSO) was proposed by Meinshausen (2007) and is a two-step-procedure. First, the standard LASSO is performed to determine a subset of non-zero predictors. Then, to estimate the model, LASSO is applied again to the subset of non-zero predictors identified during the first step. The idea is that the optimal amount of shrinkage applied in the second step should be smaller due to the smaller number of noise variables. Therefore, the estimated coefficients should suffer from less bias compared to the first step solution, which is equal to the standard LASSO. Sticking to the Lagrangian notation of equation (2), the problem solved by the relaxed LASSO estimator can be written in the following way:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} \left(x_{ij}\beta_j \cdot \mathbf{1}_{\{\beta_j \neq 0\}} \right) \Big)^2 + \phi\lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{3}$$

with $\mathbf{1}_{\{\beta_j \neq 0\}}$ being an indicator-function which takes the value 1 if β_j is non-zero in the first stage estimation and 0 otherwise. The second stage LASSO-parameter ϕ controls the amount of shrinkage applied during the second step. It is defined on $(0, 1]$, while $\phi = 1$ corresponds to the first stage solution, i.e. the standard LASSO. For $\phi \to 0$ the coefficients are estimated as an unconstrained solution equal to

estimated, it is not possible to carry out estimation for the two-step procedures described below. In these situations, as a workaround, I drop the predictor and only estimate an intercept for all procedures. These situations are rare such that this practice – if at all – should have a minor impact on the results.

[21] The intuition is described, e.g., in Hastie, Tibshirani, and J. Friedman (2009, p. 92)

performing OLS on the subset of non-zero predictors (LASSO-OLS). As mentioned before, the relaxed LASSO procedure reduces bias by applying less (or at most equal) shrinkage than the standard LASSO, nevertheless (since $\phi > 0$) it yields inconsistent estimators. Another variation of the LASSO, which can deliver consistent estimates, is the *adaptive LASSO* outlined next.

Adaptive LASSO

A technique that implicitly performs variable selection like the standard LASSO and can deliver consistent estimates is the adaptive LASSO proposed by Zou (2006). To obtain consistency, it allows for different shrinkage factors by assigning individual weights w_j to the amount of shrinkage applied to each of the coefficients. It solves the following problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2 + \lambda \sum_{j=1}^{p} |w_j \beta_j| \right\}. \tag{4}$$

Zou (2006) shows that adaptive LASSO estimators are consistent if $\boldsymbol{w} = \frac{1}{|\hat{\boldsymbol{\beta}}|^{\gamma}}$ and $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T$ is a root-n consistent estimator[22]. He suggests to use OLS estimates for $\hat{\boldsymbol{\beta}}$ and to determine $\gamma > 0$ together with λ by two-dimensional cross-validation[23]. In the present case, since $p > n$, it is not feasible to use OLS estimates as weights. This limitation for high-dimensional problems is also pointed out by Zou (2006). As a solution he suggests the use of ridge regression estimators.[24] Unfortunately, due to the huge amount of noise caused by the 47 023 predictors, ridge regression estimates can suffer from instability, which makes me cease from this idea.

Instead, two other variations are performed. For one, I use the standard LASSO estimates to plug in for $\hat{\boldsymbol{\beta}}$ (aLASSO-L). Predictors whose standard LASSO coefficients are equal to zero are excluded in the adaptive LASSO step. This implies the application of infinite shrinkage to those predictors. Using the standard LASSO estimates for $\hat{\boldsymbol{\beta}}$ is a common workaround in literature in case of $p > n$ (c.f. Chatterjee and Lahiri 2013 and Kraemer, Schaefer, and Boulesteix 2009).

As a second variation, I use the OLS coefficients, which have been estimated on the subset of non-zero predictors (LASSO-OLS). Again, predictors whose standard (first stage) LASSO coefficients are equal to zero are excluded. Both variations yield

[22] Note that root-n consistency of $\hat{\boldsymbol{\beta}}$ is not necessarily required since this condition can be weakened. See Zou (2006) for details.

[23] Cross-validation is discussed in detail in Section 4.2

[24] Note that this modification raises the need to estimate an additional tuning parameter λ^{ridge} to determine the best ridge regression fit.

consistent estimates[25] and, in a way, combine the advantages of relaxed and adaptive LASSO.

[25] OLS is root-n consistent. Root-n consistency of the LASSO estimates is shown by Knight and Fu (2000). The relevant Lemma is also stated in Zou (2006).

8.6 Blocked k-fold cross-validation

Due to the time-series character of the data, observations can be numbered consecutively from 1 to N. To create training and testing sets, the data is divided into k equally large, non-overlapping sets of subsequent observations (blocks).[26] Each block constitutes a fold and serves as testing set once, while the other sets are preserved as the corresponding training set.

To eliminate dependence between folds, h observations at the inner borders[27] of all k training sets are dropped. The advantage of using blocks rather than assigning the folds randomly is illustrated by the fact that this procedure minimizes the loss of data due to dropping h observations around the members of the testing set.[28]

Note that h should at least be set to one as a 24-hour-interval prior to market opening is used to predict daily returns. This setup implies that, in general, news released during the prior trading period can be used to forecast the next day's returns such that some information in the t-th row of $\boldsymbol{X}$ could have had influence on y_{t-1} and thus contradicts independence between folds for $h = 0$. In addition, as pointed out by Burman, Chow, and Nolan (1994), h should be set according to the order of autocorrelation of the data. They argue that, to ensure independence, h should be large and suggest $h = \frac{N}{6}$ as a rule of thumb. This results in removing one third of the data, which can be problematic, especially in small samples. Therefore, a trade-off between sample size and the degree of dependence between folds arises. Racine (2000) gives the same arguments as Burman, Chow, and Nolan (1994) and shows that even small values of h can significantly improve cross-validation performance.[29]

The 5.3, 5.4 and 5.5 show empirical autocorrelations for the VIX-returns, the S&P 500-returns, the ten most frequent features as well as ten randomly drawn features. Note that the randomly drawn features are the same as those that have been tested for stationarity. After observing these autocorrelations I set $h = 20$ for the entire analysis. I argue that this is more than adequate for the vast majority of predictors, while not being too costly in the sense of dropping too much information. It is acknowledged that this does not eliminate autocorrelation for some predictors since this would require h to be much greater. Nevertheless, in the given application it is necessary to find a compromise that serves the prediction performance.

The optimal tuning parameter λ (or the optimal vector (γ, λ) in case of two-dimensional cross-validation) is chosen from a prespecified grid[30] such that it mini-

[26] Note that in contrast to standard cross-validation there is no randomness involved. For certain modifications this can be a drawback as discussed later in Section 7.

[27] *Inner* means that only observations lying *between* two folds are dropped. The first and the last h observations are never dropped since there are no other folds to which dependence could occur.

[28] When using blocked cross-validation at most $2h$ observations are dropped. For standard cross-validation this can be up to $2\lceil\frac{N}{k}\rceil h$, which is much larger, as long as k is not close to N.

[29] Note that Burman, Chow, and Nolan (1994) introduce a correction term to take into account the loss of data. This correction is not applied by Racine (2000), nor is it in this paper.

[30] The grids λ and γ are chosen from, are described in Appendix 8.7.

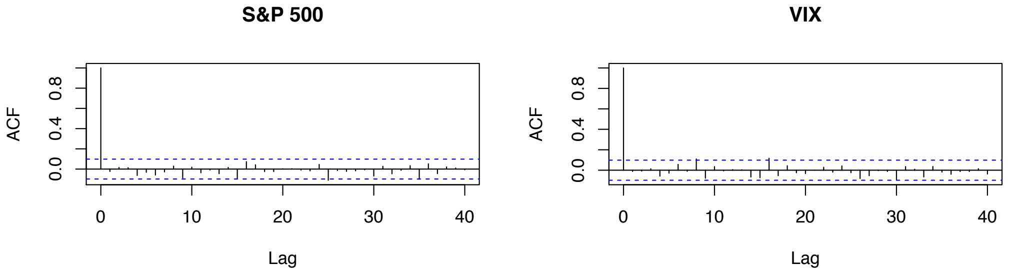

Figure 5.3: Empirical autocorrelations for the returns of S&P 500 and VIX from Jan 1, 2014 to Jul 31, 2015.

mizes the average cross-validation error, given a particular Loss-function L. To write this down mathematically, I introduce the following notation: Let K_1, K_2,. . . ,K_k denote the sets of observations, which represent the folds. Let n_1, n_2,. . . ,n_k be the corresponding numbers of observations in each of the folds. In addition, set $n_0 = 0$. Therefore, $\underline{n_j} = \sum_{i=0}^{j-1}(n_i)+1$ is the number of the first observation in K_j and, equivalently, $\overline{n_j} = \sum_{i=0}^{j} n_i$ corresponds to the number of the last observation in K_j. Also, let $\boldsymbol{x_{i\cdot}}$ denote the i-th row of the regressor matrix $\boldsymbol{X}$ and $X_{(\underline{n_j}:\overline{n_j})} = \{\boldsymbol{x_{i\cdot}} : \underline{n_j} \leq i \leq \overline{n_j}\}$ denote the set containing all observations from $\underline{n_j}$ to $\overline{n_j}$ such that $K_j = X_{(\underline{n_j}:\overline{n_j})}$. Correspondingly, let $K_j^c = X_{-(\underline{n_j}:\overline{n_j})} = \{\boldsymbol{x_{i\cdot}} : 1 \leq i < \underline{n_j}\} \cup \{\boldsymbol{x_{i\cdot}} : \overline{n_j} < i \leq N\}$ be the complement, i.e. all observations except those in fold j. Using this notation, the

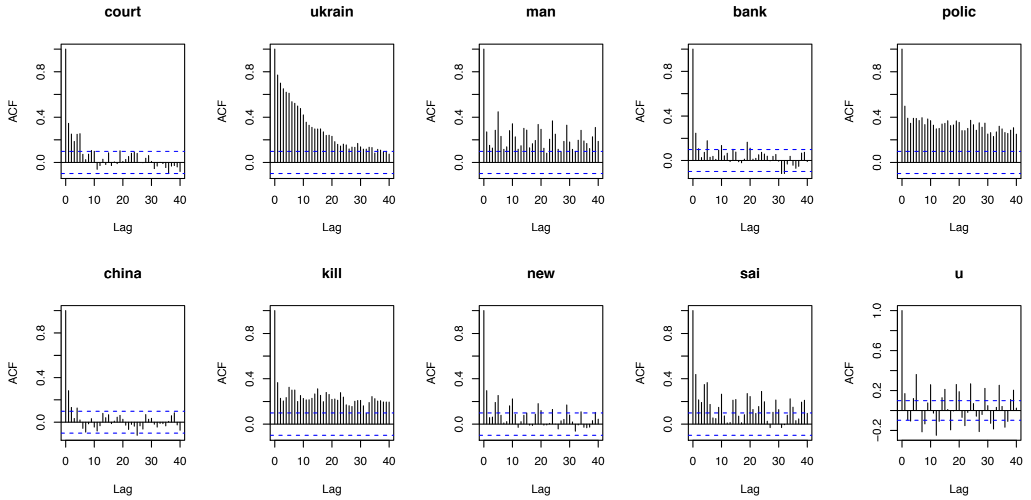

Figure 5.4: Empirical autocorrelations for the ten most frequent features from Jan 1, 2014 to Jul 31, 2015.

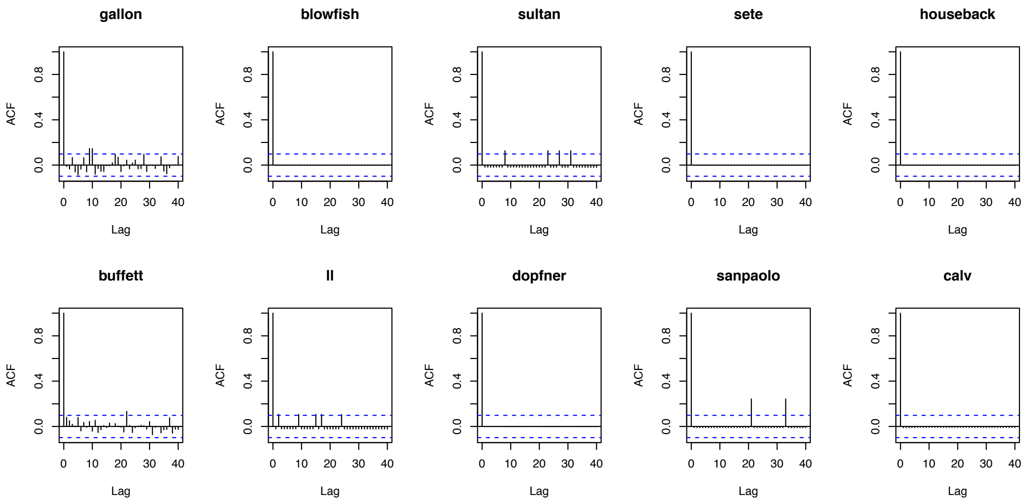

Figure 5.5: Empirical autocorrelations for ten randomly drawn features from Jan 1, 2014 to Jul 31, 2015.

implemented cross-validation procedure can be written down as follows:

$$\min_{\lambda} \frac{1}{N} \sum_{j=1}^{k} \sum_{i=\underline{n_j}}^{\overline{n_j}} L\big(y_i, \hat{f}^{\mathcal{X}_j}(\boldsymbol{x}_{\boldsymbol{i}\cdot}, \lambda)\big), \tag{5}$$

with $\mathcal{X}_j = X_{-(\underline{n_j}-h:\overline{n_j}+h)}$ representing the set $\hat{f}$ is estimated on.[31]
This can easily be extended to the two-dimensional case giving

$$\min_{\gamma} \min_{\lambda} \frac{1}{N} \sum_{j=1}^{k} \sum_{i=\underline{n_j}}^{\overline{n_j}} L\big(y_i, \hat{f}^{\mathcal{X}_j}(\boldsymbol{x}_{\boldsymbol{i}\cdot}, \lambda, \gamma)\big). \tag{6}$$

Note that the described method is closely related to the h-block cross-validation procedure proposed by Burman, Chow, and Nolan (1994) and the hv-blocked cross-validation by Racine (2000). It can be seen as a k-fold-version of h-block cross-validation or an incomplete version of hv-blocked cross-validation.[32]

Error measure

As pointed out by Bergmeir and Benítez (2012) as well as Hastie, Tibshirani, and J. Friedman (2009, p. 219), different error measures can be used. The most common

[31] Being even more accurate it should be written as $\mathcal{X}_j = X_{-(\max(1,\underline{n_j}-h):min(\overline{n_j}+h),N)}$ since for the first and the last observation it does not make sense to subtract or add some $h > 0$.

[32] Both procedures, h-block cross-validation as well as hv-blocked cross-validation, are not suitable for my analysis as they are computationally too intense.

choices are the squared error loss, which gives the mean squared error (MSE), and the absolute error loss, yielding the mean absolute error (MAE):

$$L(Y, \hat{Y}) = \begin{cases} (Y - \hat{Y})^2 & \text{MSE} \\ |Y - \hat{Y}| & \text{MAE.} \end{cases} \tag{7}$$

The latter does not put additional weight on large deviations and is therefore less affected by outliers.

Standard Errors

As pointed out by Tibshirani (1996) as well as Knight and Fu (2000), it is non-trivial to compute standard errors for LASSO-type estimators. Tibshirani (1996) and Osborne, Presnell, and Turlach (2000) provide some approximations which are considered unsatisfactory by Knight and Fu (2000). Another suitable approach, which is also suggested by Tibshirani (1996) as well as Knight and Fu (2000), is to use the bootstrap to obtain valid standard errors. Since I focus on prediction, the computation of standard errors is not pursued here.

8.7 Grids for λ and γ

The Parameter λ

The Parameter λ is determined on a grid of length 100 which is chosen by the default option of the *glmnet*-package (J. Friedman, Hastie, and Tibshirani 2010) and depends on N, p as well as $\boldsymbol{w}$. For more details I refer to the corresponding help files in R or equivalently to the *glmnet*-reference manual, which is available under https://cran.r-project.org/web/packages/glmnet/glmnet.pdf.

The Parameter γ

The parameter γ is is determined on a grid from 0.025 to 10 with increasing distances between the single values. From 0.025 to 2 the next value increases by 0.025. From 2 to 4 the next value increases by 0.05. From 4 to 6 the next value increases by 0.25. From 6 to 10 the next value increases by 0.5. This results in the following grid of length 136:

$\gamma \in$ {0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45, 0.475, 0.5, 0.525, 0.55, 0.575, 0.6, 0.625, 0.65, 0.675, 0.7, 0.725, 0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925, 0.95, 0.975, 1, 1.025, 1.05, 1.075, 1.1, 1.125, 1.15, 1.175, 1.2, 1.225, 1.25, 1.275, 1.3, 1.325, 1.35, 1.375, 1.4, 1.425, 1.45, 1.475, 1.5, 1.525, 1.55, 1.575, 1.6, 1.625, 1.65, 1.675, 1.7, 1.725, 1.75, 1.775, 1.8, 1.825, 1.85, 1.875, 1.9, 1.925, 1.95, 1.975, 2, 2.05, 2.1, 2.15, 2.2, 2.25, 2.3, 2.35, 2.4, 2.45, 2.5, 2.55, 2.6, 2.65, 2.7, 2.75, 2.8, 2.85, 2.9, 2.95, 3, 3.05, 3.1, 3.15, 3.2, 3.25, 3.3, 3.35, 3.4, 3.45, 3.5, 3.55, 3.6, 3.65, 3.7, 3.75, 3.8, 3.85, 3.9, 3.95, 4, 4.25, 4.5, 4.75, 5, 5.25, 5.5, 5.75, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10}.

8.8 Simulation

To verify that the presented methods can indeed detect a small set of meaningful variables within a huge amount of noise, I conduct a simple simulation. As a first step, in order to create a problem similar to the one in my analysis, I generate time-series for 47 020 variables and 252 time points $t \in \{1, \dots, 252\}$[33]. They are modeled as an autoregressive process of order one with a low coefficient:

$$x_t = 0.2x_{t-1} + \varepsilon_t \quad \forall\, t, \tag{8}$$

while ε_t is standard normally distributed: $\varepsilon_t \sim N(0,1)\;\forall\, t$. To add meaning to some variables I construct the dependent variable $\boldsymbol{y} = (y_1, \dots, y_{252})^T$ from the first 20 of the just generated variables and add a great amount of noise $\varepsilon_t^* \sim N(0,10)\;\forall\, t$. With $\boldsymbol{x}^i$ denoting the i-th generated variable, $\boldsymbol{y}$ is obtained from

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}^*, \tag{9}$$

with $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^{20}]$ and $\boldsymbol{\varepsilon}^* = (\varepsilon_1^*, \dots, \varepsilon_{252}^*)^T$. The variance of the error terms is chosen large to reflect a situation where $\boldsymbol{y}$ is affected by a lot of unmodeled factors. This situation is expected to be given in the application below. Coefficients are chosen as follows:

$$\begin{aligned}\boldsymbol{\beta} = (&-0.2, -0.4,\quad 0.6, -0.8,\quad 1.0,\quad 1.2,\quad 1.4, -1.6,\quad 1.8, -2.0,\\ &2.2, -2.4,\quad 2.6, -2.8,\quad 3.0, -3.2,\quad 3.4, -3.6,\quad 3.8, -4.0)^T.\end{aligned} \tag{10}$$

Note that the elements of $\boldsymbol{\beta}$ sum up to 0. 5.9 and 5.10 summarize the performance of all considered methods on the generated data. The results are presented for different k and computed under the mean absolute error with $h = 20$. Results for the squared error loss are presented in Appendix 8.9. The first value displayed in 5.9 corresponds to the number of non-zero coefficients among the first 20 variables. These variables, which were used to generate $\boldsymbol{y}$, are from now on referred to as the true predictors. The second value presented in 5.9 is the total number of non-zero coefficients. In addition, the true predictors that are not detected by the model, i.e. the variables whose coefficients are incorrectly estimated as zero, are presented. One can see that for given k the adaptive LASSO with OLS weights performs best in the sense of including the smallest number of noise variables. Thus, this method estimates the smallest models. As a consequence, the adaptive LASSO with OLS weights runs a higher risk of dropping one of the true predictors, as seen, e.g., for $k = 40$ and $k = 20$. An exception to this is the case of $k = 5$, where aLASSO-O estimates the smallest model while including 17 of the first 20 variables.[34]

[33] Since the majority of models is estimated on 252 observations.

[34] The statement of a higher probability of dropping additional true predictors is additionally supported by the results under the MSE presented in Appendix 8.9.

Table 5.9: Simulation Results MAE - Model Size

	5 Folds	10 Folds
std. LASSO	17/189 - V1, V2, V3	16/120 - V1, V2, V3, V4
LASSO-OLS	17/189 - V1, V2, V3	16/120 - V1, V2, V3, V4
rel. LASSO	16/180 - V1, V2, V3, V4	15/112 - V1, V2, V3, V4, V5
aLASSO-O	17/151 - V1, V2, V3	15/ 88 - V1, V2, V3, V4, V5
aLASSO-L	16/175 - V1, V2, V3, V4	16/110 - V1, V2, V3, V4

	20 Folds	40 Folds
std. LASSO	16/134 - V1, V2, V3, V4	16/115 - V1, V2, V3, V4
LASSO-OLS	16/134 - V1, V2, V3, V4	16/115 - V1, V2, V3, V4
rel. LASSO	16/130 - V1, V2, V3, V4	16/110 - V1, V2, V3, V4
aLASSO-O	15/111 - V1, V2, V3, V4, V5	15/ 89 - V1, V2, V3, V4, V5
aLASSO-L	16/128 - V1, V2, V3, V4	16/100 - V1, V2, V3, V4

The first values correspond to the number of non-zero coefficients for the first 20 variables. The second number corresponds to the total number of non-zero coefficients i.e. the size of the estimated model. The variables, whose coefficients are incorrectly estimated to be zero, are displayed after the minus sign.

5.10 summarizes the quality of the estimated coefficients. It shows the sum of absolute deviations of the estimated coefficients to the true coefficients, i.e. the coefficients used to generate the data. This sum is presented separately for the true predictors and the noise variables. Note that all estimated models fit an intercept, which is not included in the calculation of deviations.

When considering the coefficients for the first 20 variables, the adaptive LASSO procedures clearly perform best, while OLS weights seem to be superior to those obtained from the standard LASSO. The picture is different for the noise variables, since the standard LASSO performs much better than all two-step procedures. This originates from the greater amount of shrinkage applied to the coefficients and illustrates this exact property which was the reason for introducing the relaxed LASSO as well as the other two-step procedures in the first place. For the noise-variables, whose true coefficients are zero, the greater shrinkage leads to better estimates. In turn – again following the argumentation for introducing the relaxed LASSO – the standard LASSO performs badly for the first 20 variables, as these coefficients are also shrunken towards zero. Therefore, although this method provides the smallest total deviation, it might not be the best choice for forecasting. Considering only the two-step methods the adaptive LASSO procedures again yield the most promising results. In addition, using OLS weights seems to prevail slightly.

To summarize, these results show that the proposed methods can indeed detect

Table 5.10: Simulation Results MAE - Deviations of Coefficients

$k = 5$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.35	16.52	16.32	**15.82**	15.92
others	**18.45**	27.10	24.77	24.87	24.96
total	**34.80**	43.62	41.08	40.69	40.87

$k = 10$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.93	14.63	14.84	**12.95**	14.21
others	**10.47**	22.73	20.74	19.09	20.07
total	**27.40**	37.36	35.58	32.04	34.28

$k = 20$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.73	15.97	16.01	**15.23**	15.73
others	**12.09**	25.05	23.62	22.85	23.11
total	**28.82**	41.03	39.64	38.08	38.84

$k = 40$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	17.03	14.70	14.85	**13.39**	13.93
others	**9.96**	21.62	20.73	19.23	19.99
total	**26.99**	36.32	35.58	32.62	33.92

This table shows the sums of absolute deviations of the estimated coefficients to the true coefficients i.e. the coefficients used for generating the data. The first row displays the sum of deviations for the first 20 variables. The second row corresponds to the remaining 47000 variables. The third row shows the sum over all variables.

the majority of true predictors, while having difficulties detecting those with low coefficients. However, all selected models are too large as they also pick some of the noise variables. Still, as most coefficients estimated for these noise variables are small and alternate around zero, the estimated models are expected to have some predictive power.

It is acknowledged that the proposed methods are far from detecting the correct model. Nevertheless, reducing the predictors from 47 020 to less than 200 while retaining most of the true predictors can be regarded as (partial) success.

8.9 Simulation Results – MSE

The following tables show the simulation results computed under the squared error loss. The simulation was performed using the same data as for the tables presented in the text.

Table 5.11: Simulation Results MSE - Model Size

	5 Folds	10 Folds
std. LASSO	17/209 - V1, V2, V3	16/120 - V1, V2, V3, V4
LASSO-OLS	17/209 - V1, V2, V3	16/120 - V1, V2, V3, V4
rel. LASSO	16/198 - V1, V2, V3, V4	15/112 - V1, V2, V3, V4, V5
aLASSO-O	15/142 - V1, V2, V3, V4, V5	15/ 89 - V1, V2, V3, V4, V5
aLASSO-L	16/158 - V1, V2, V3, V4	15/111 - V1, V2, V3, V4, V5

	20 Folds	40 Folds
std. LASSO	16/129 - V1, V2, V3, V4	16/140 - V1, V2, V3, V4
LASSO-OLS	16/129 - V1, V2, V3, V4	16/140 - V1, V2, V3, V4
rel. LASSO	15/121 - V1, V2, V3, V4, V5	16/133 - V1, V2, V3, V4
aLASSO-O	15/ 92 - V1, V2, V3, V4, V5	15/115 - V1, V2, V3, V4, V5
aLASSO-L	15/118 - V1, V2, V3, V4, V5	16/124 - V1, V2, V3, V4

The first values correspond to the number of non-zero coefficients for the first 20 variables. The second number corresponds to the total number of non-zero coefficients i.e. the size of the estimated model. The variables, whose coefficients are incorrectly estimated to be zero, are displayed after the minus sign.

Table 5.12: Simulation Results MSE - Deviations of Coefficients

$\boldsymbol{k = 5}$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.36	16.01	16.19	**14.94**	15.71
others	**20.25**	26.85	24.21	24.22	24.70
total	**36.61**	42.86	40.40	39.16	40.40

$\boldsymbol{k = 10}$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.93	14.63	14.84	**13.11**	14.39
others	**10.47**	22.73	20.74	19.32	20.37
total	**27.40**	37.36	35.58	32.43	34.76

$\boldsymbol{k = 20}$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.84	15.62	15.54	**13.58**	14.75
others	**11.00**	24.61	22.35	20.14	21.43
total	**27.84**	40.23	37.89	33.73	36.18

$\boldsymbol{k = 40}$	std. LASSO	LASSO-OLS	rel. LASSO	aLASSO-O	aLASSO-L
V1–V20	16.69	16.03	16.05	15.17	**15.14**
others	**12.62**	25.07	23.44	22.97	22.54
total	**29.32**	41.10	39.48	38.14	37.67

This table shows the sums of absolute deviations of the estimated coefficients to the true coefficients i.e. the coefficients used for generating the data. The first row displays the sum of deviations for the first 20 variables. The second row corresponds to the remaining 47000 variables. The third row shows the sum over all variables.

8.10 VIX – Cross-Validation Error Curves

The figure shows the cross-validation error curves using the VIX returns as dependent variable for nine randomly drawn dates.
Curves correspond to the (first stage) standard LASSO procedure. The optimal parameter chosen for the standard LASSO is most crucial as it affects all considered models. Note that curves for the other LASSO procedures are not easily comparable since the models estimated in the first stage differ in k such that estimation is carried out on different data. The *glmnet* package does rescale λ according to the number of variables. One could instead plot the cross-validation error against the degrees of freedom implied by each λ. This is not expected to yield further insights. Additionally note that, since not only the variables which are included in the final model were used for LASSO-estimation, the degrees of freedom are not adequately quantifying the complexity of the model. For this reason, the concept of effective degrees of freedom has been proposed by Zou, Hastie, and Tibshirani (2007).

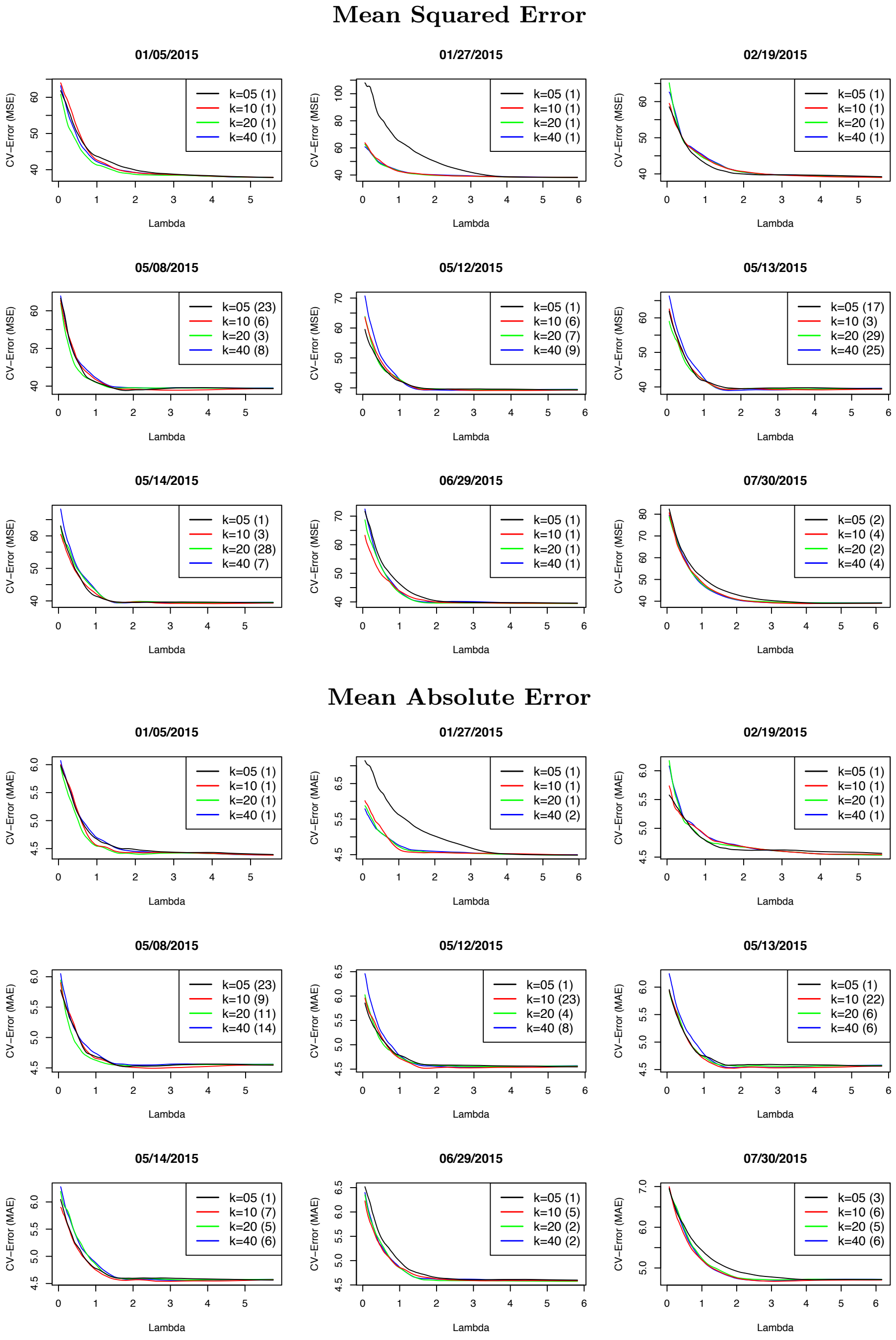

Figure 5.6: Cross-Validation Error Curves for the two Error measures and nine randomly drawn dates with VIX returns as dependent variable. For each k the number in brackets corresponds to the size of the estimated model implied by the optimal parameter. For (1) only an intercept is estimated.

8.11 Additional Performance Tables

Results under MSE

Table 5.13: Results under MSE

	Proportion of Correct Directions				Hypothetical Profit in %			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	63%	63%	62%	64%	129.7	121.8	115.4	126.7
LASSO-OLS	66%	61%	60%	59%	156.5	69.9	65.2	55.9
rel. LASSO	66%	61%	60%	59%	156.5	69.9	65.2	55.9
aLASSO-O	66%	61%	60%	59%	156.5	72.4	80.3	68.9
aLASSO-L	66%	60%	60%	58%	149.6	66.3	78.1	34.7

Table 5.14: Results under MSE for Non-Degenerate Models

	Proportion of Correct Directions				Hypothetical Profit in %			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	59%	63%	59%	65%	23.7	35.7	28.4	53.6
LASSO-OLS	71%	57%	53%	51%	50.5	−16.2	−21.9	−17.2
rel. LASSO	71%	57%	53%	51%	50.5	−16.2	−21.9	−17.2
aLASSO-O	71%	57%	51%	51%	50.5	−13.7	−6.8	−4.2
aLASSO-L	68%	54%	53%	47%	43.6	−19.8	−8.9	−38.4

Model Size (MSE)

Table 5.15: Model Size (MSE)

	5 Folds	10 Folds	20 Folds	40 Folds
std. LASSO	2.72 - 41/146	2.81 - 46/146	3.63 - 49/146	3.71 - 51/146
LASSO-OLS	2.72 - 41/146	2.81 - 46/146	3.63 - 49/146	3.71 - 51/146
rel. LASSO	2.70 - 41/146	2.81 - 46/146	3.62 - 49/146	3.70 - 51/146
aLASSO-O	2.52 - 41/146	2.60 - 46/146	3.18 - 49/146	3.34 - 51/146
aLASSO-L	2.51 - 41/146	2.66 - 46/146	3.32 - 49/146	3.49 - 51/146

This table summarizes the estimated model size for different k. The first value corresponds to the average number of non-zero coefficients (including the intercept). The value after the minus sign shows the number of times a non-trivial model (with at least one additional predictor) is estimated. 146 is the length of the prediction horizon.

Investing with Threshold

In the following I present tables for the proportion of correct directions and the hypothetical profit with an investing-threshold. As threshold 0.8% is chosen since the degenerate models always predict in $[-0.1, -0.8]$. Therefore it is only invested if a decrease of more than 0.8% or an increase of more than 0.8% is predicted by the system. The first table shows the number of investments taken. Note that the prediction horizon consists of 146 trading days. Introducing this threshold decreases the number of investments severely. Basically, it reduces the investments by all predictions of degenerate models plus those that are close to zero.

Table 5.16: Number of Investments with Threshold 0.8%

	MAE				MSE			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	17	19	25	21	12	17	16	18
LASSO-OLS	37	35	39	37	20	29	28	32
rel. LASSO	34	30	37	34	21	30	27	32
aLASSO-O	38	35	38	34	21	26	26	31
aLASSO-L	38	35	39	36	21	28	28	31

Table 5.17: Proportion of Correct Directions with Threshold 0.8%

	MAE				MSE			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	65%	53%	60%	48%	58%	47%	50%	50%
LASSO-OLS	57%	54%	56%	49%	70%	62%	57%	47%
rel. LASSO	56%	53%	57%	53%	67%	60%	56%	47%
aLASSO-O	61%	51%	50%	44%	67%	54%	54%	45%
aLASSO-L	61%	51%	56%	56%	67%	54%	54%	42%

Table 5.18: Hypothetical Profit in % with Threshold 0.8%

	MAE				MSE			
	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 5$	$k = 10$	$k = 20$	$k = 40$
std. LASSO	−8.2	−4.4	20.7	4.0	6.2	−18.1	−11.4	−5.0
LASSO-OLS	12.9	−0.8	30.9	37.6	26.7	4.3	8.2	−8.5
rel. LASSO	1.1	5.0	26.8	40.0	23.0	0.8	−0.5	−8.5
aLASSO-O	29.9	−10.0	18.3	16.4	21.2	−10.5	7.1	−9.5
aLASSO-L	17.7	−13.9	24.7	35.8	21.2	−13.7	1.4	−26.2

8.12 Performance over Whole Horizon

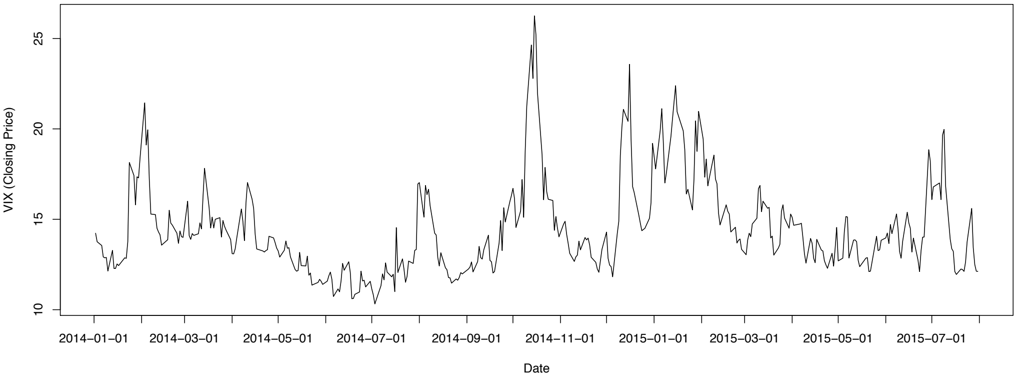

Figure 5.7: This figure shows the VIX in levels over the whole horizon.

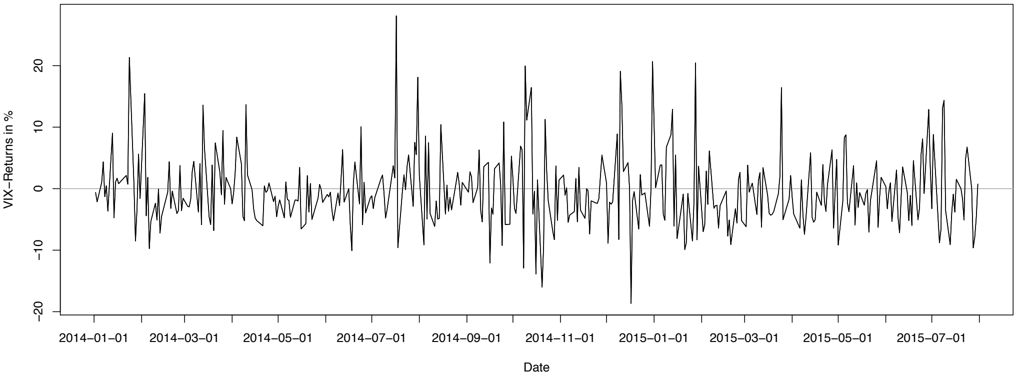

Figure 5.8: This figure shows the VIX-Returns in % over the whole horizon.