5_Empirical_Studies

Nicholas Mitchell

April 1, 2016

Contents

1 Empirical studies

1.1 Financial market data acquisition

Here, a brief summary of the traditional financial data used in this study is presented. Daily data for numerous market segments was collected, using free online sources[1]. Market data is freely available for time-spans that greatly surpass that of Twitter data (and social media data in general[2]), which amounts to the fact that the scraped Twitter data is the limiting factor regarding the time-series length used for this study. Table 1 summarises the final market data, categorised by the asset class from which each variable stems. All obtained market data was of daily frequency. The timeline used ranges from 14^{th} January 2013 until 11^{th} September 2015, which equates to a total of 971 days, with a total of 695 *weekdays*. The term *weekdays* is intentionally specified, in place of *business days*, as bank holidays in North America remained part of the timeline for modelling[3].

A total of seven asset classes were included, covering the majority of traditional stock markets (although futures markets were not included). Options are indirectly incorporated via several indicators for market volatility, e.g. the VIX volatility index (ID 26 in Table 1) is a metric derived from the implied volatility of highly liquid options[4]. The single exchange traded fund (ETF) (ID 12) was chosen to represent emerging markets, with the majority of other assets prevalent only in developed economies. The gold and copper spreads (with IDs 27 and 28) were derived as the nominal spread between the spot and three month prices, as a supplementary proxy to short term volatility. The **Reference** column provides the unique *ticker* or name that is used by the corresponding **Data Provider**. *Quandl* uses many sources of data (see footnotes for more information), whereas (for this study) the *quantmod* package in R was used as a convenient interface to Yahoo Finance data. In several cases, only one reference is given, which returns the data for several of the variables. For example, all zero-coupon bond data is returned for all maturities from one database call. This is indicated in Table 1 as appropriate.

1.2 Data preparation

1.2.1 Data transformations

The combined output of Chapters `chapter-twitter-mining` and `sentiment-analysis-chapter` consists of the sentiment of individual tweets for each of the thirteen search terms over a two year eight month period. This forms one half of the data set to be used, which necessitated further manipulation before being combined with financial market data for modelling. Aggregating to daily frequency was a requirement, and there were several ways to to do this. This following sub-sections outline how this was achieved.

1.2.1.1 Re-scaling sentiment scores The results that were returned from the five sentiment analysis models (at the individual tweet level) were all on slightly different scales. Although within the same orders of magnitude, ± 10, they were re-scaled to be spread over the same range for consistency's sake. This also facilitated the combination of data from the different models, as discussed in the Section 1.2.3.2. When re-scaling the sentiment scores, it was important to retain the meaning that the scores conveyed, i.e. a positive value conveyed a positive sentiment, and vice versa. Therefore it was not an option to simply normalise the data, giving it a mean value of zero and a desired variance, as this would have inevitably meant that some individual scores would cross the *zero-boundary*, thereby changing their sign and losing their true meaning. The method that was therefore used, was to merely reduce their magnitudes, so that the maximum score within one set of tweets[5] was equal to one. This was achieved by simply dividing the scores for each data set by the maximum scores in that set. In the special case of the SentiStrength data, which produces a binary output, the two response were first averaged - creating a single score for each tweet - before being re-scaled to the same range as the other sentiment scores: $[-1, 1]$.

[1] A mixture of Yahoo Finance† and Quandl† was used - please visit them for more information on their original sources. Interfaces were provided by the R packages *quantmod*† and *Quandl*†, respectively.

[2] Market data can be obtained for many indices and assets deep into the last century, whereas social media data older than ten years old is extremely rare.

[3] A detailed listing of the bank holidays can be found in Appendix `pub-holidays`.

[4] The exact methods of calculation can be found in the relevant white paper from the Chicago Board Options Exchange†.

[5] One set signifies the tweets for one search term, for one sentiment analysis model. Thirteen search terms and five models gave sixty-five initial data sets.

ID	Asset class	Asset	Data source	Reference
1	Commodities	Gold spot	Quandl	LBMA/GOLD
2		Gold 3M		(as above)
3		Copper spot		LME/PR_CU
4		Copper 3M		(as above)
5		Oil (WTI)		CHRIS/ICE_T1
6		Natural gas		OFDP/FUTURE_NG1
7	Currency pairs	USD-AUD	Quandl	CURRFX/USDAUD
8		USD-CAD		CURRFX/USDCAD
9		USD-EUR		CURRFX/USDEUR
10		USD-GBP		CURRFX/USDGBP
11		USD-JPY		CURRFX/USDJPY
12	Exchange traded fund	MSCI Emerging Markets	quantmod	EEM
13	Fixed income (US bonds)	Zero-coupon 1Y	Quandl	FED/SVENY
14		Zero-coupon 2Y		(as above)
15		Zero-coupon 5Y		(as above)
16		Zero-coupon 10Y		(as above)
17		Zero-coupon 15Y		(as above)
18		Zero-coupon 20Y		(as above)
19	Stock indices	DAX (Germany)	quantmod	^GDAXI
20		Dow Jones (U.S.)		^DJI
21		FTSE100 (U.K.)		^FTSE
22		Nikkei 225 (Japan)		^N225
23		S&P500 (U.S.)		^GSPC
24		Shanghai SE (China)		000001.SS
25	Volatility indicators	VXD (Dow Jones)	Quandl	CBOE/VXD
26		VIX (S&P500)		YAHOO/INDEX_VIX
27		Gold spread		(derived)
28		Copper spread		(derived)

Table 1: A summary of the financial market data to be paired with sentiment analysis results. Daily frequency was obtained for all data, meaning no interpolation was necessary. Imputation was performed using the LOCF method.

1.2.1.2 Weighted aggregation of sentiment scores

The second form of aggregation that was necessary to perform on the Twitter data, was to ensure that each of the thirteen search terms provided a *single* score for each day (from each of the five sentiment models). This aggregation was a necessary step to match the frequency of the financial market data. In this step, the additional meta data that was *scraped* for each tweet was brought into use. Instead of computing the average score for each day over all tweets, each tweet's individual score was first weighted, by using both the number of times that it was *retweeted* as well as the number of times it was marked as a *favourite* by another user. The reasoning behind this step may be explained as follows: if one tweet has a score of e.g. +2, this means one person has a positive sentiment concerning the matter that was tweeted. If then five additional people retweeted or favourited that tweet, they thereby show their agreement with that tweet, and so the underlying sentiment is magnified in its interpretation. Therefore, a tweet that has been retweeted and favourited many times should logically carry more weight into the required average for that day, as it represents the opinion of a greater number of people. Let a *tweet event* be defined as one additional opinion, i.e. one retweet or one marking as a favourite. Given this, the methodology used for weighting a single tweet may be summarised as follows:

$$score_{weighted} = \left(\frac{\tau}{\sum_{i=1}^{N} \tau_i} \right) \cdot score_{original} \tag{1}$$

where τ is the total events for the tweet being weighted and N is the number of tweets on the day in question. This means the denominator represents the total sum of tweet events on that day, for that specific search term. Using this, the sentiment score for each day reflects the sentiment found on Twitter with an added amount of precision.

1.2.2 Imputation

As was touched upon in Section `final-output`, there was a negligible proportion of missing data in the sentiment analysis results (< 1 %). The majority of missing data, throughout both the financial market and sentiment analysis data sets (weekends having been removed) could be contributed to public holidays. As the vast majority of the data listed in Table 1 was obtained from American markets, only official public holidays from America were considered. Over the 695 period timeline there were a total of 25 public holidays (≈ 3.6 % of the periods), which are detailed in Table `tab:public-holidays`, Section `pub-holidays`. The weekends were removed from the sentiment analysis data, in order to be combined with the market data.

Even though the percentage of missing data within the entire data set (including sentiment data and market data) fell below 1 % after removing weekends, the component-wise boosting models do not (by default) tolerate missing data. Therefore it was necessary to use a method of imputation; the method selected was that of *last observation carried forward* (LOCF). This must be carried out before the log-returns are computed[6]. Using LOCF implies then that the log-return is simply equal to zero on days where no data was received, as the difference would be zero between the imputed day and its preceding day. Additionally, imputing the data instead of removing the data was not able to create a large bias, as there are so few missing data points. Other methods of imputation that were considered include splines, k-Nearest Neighbour and variable modelling[7]; however, as the proportion of missing data was so low, these more complicated methods were not warranted. As the public holidays were imputed, a dummy variable was created to make use of the information, if possible[8].

1.2.3 Derived variables

1.2.3.1 Weekend sentiment

The sentiment analysis data was collected over a total of 971 days, in continuous time, without any breaks. This means there was sentiment data available from the weekends that could not be directly modelled alongside market data, which is only for weekdays. In an attempt to capture sentiment from the weekend, a new variable was created that incorporated sentiment data from

[6] Otherwise the result from that computation would be twice as many missing data points, when using the `diff()` function within R.

[7] This involves modelling each individual variable in a way that allows one to impute the variable using its own distribution.

[8] See Section 1.2.3.3, where the inclusion of public holidays and other deterministic factors were included in the model.

each weekend, and was used in modelling the following Monday's returns[9].

In order to extract the weekend sentiment, the values from each Friday, Saturday and Sunday were grouped into one mean value, which replaced the original value for the same Friday[10]. This final Friday value was the value to be used in predicting the next day, i.e. the immediately following Monday. By not using any information at or further ahead of in time than the outcome variable, there is no violation of temporal information flow. For this new variable, only three of the thirteen search terms were selected; "Dow Jones", "federal reserve" and "stock prices". This is because the interpretation regarding the underlying sentiment becomes rather difficult once more are combined. The three terms were chosen due to their obvious correlation in underlying sentiment with market movements. Figure 1 illustrates the combined sentiment score of the newly derived variable against movements of the DJIA - the weekends are shaded, and the lack of market data is noticeable by zero returns on the shaded regions. It can be observed that the movement of the sentiment over the weekend coincides with a market movement in the same direction for the second and third of the three weekends. Furthermore, it can be seen that the sentiment rises over all three Mondays (the segments directly following the shaded regions), which reflects the market movements on the first Monday and precedes the Market's actual upwards movement in the second and third weeks.

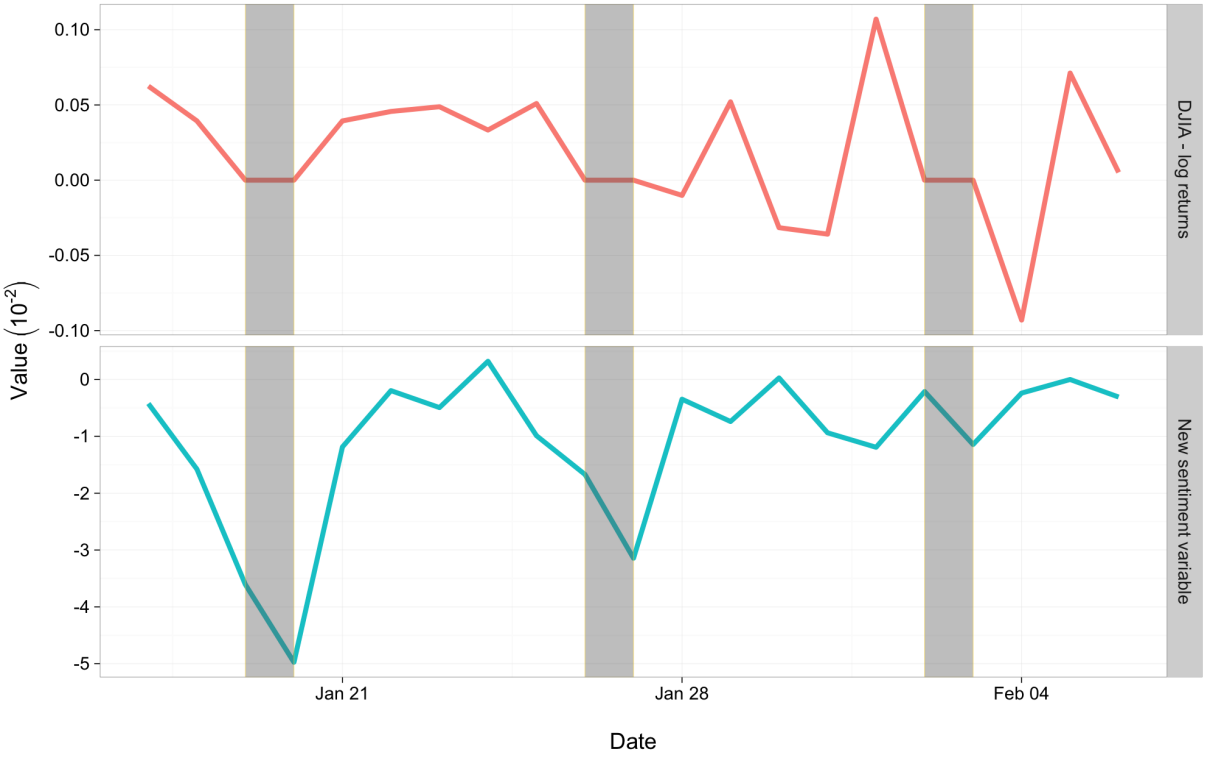

Figure 1: A comparison between the derived weekend sentiment variable (an average of three single sentiment variables) against the log returns of the DJIA. Dates are taken from 2013, with weekends highlighted by grey rectangles.

1.2.3.2 Combined sentiment models

As mentioned above, five sentiment analysis models were used to score the tweets. A last set of variables were created in order to condense the sentiment data into fewer variables. This was performed simply by using the mean over all five sentiment analysis models, for each of the thirteen search terms. These variables were used in the creation of more succinct data sets, which can be seen and compared to the other in Section 1.2.4.

[9]This variable was constructed out of the intuition that it makes sense that sentiment scores from the weekend should reflect the opinions of people who may make trades on the following Monday. However, no thorough statistical analysis was performed in the construction of this new variable.

[10]One could imagine a more complicated method of combining the sentiment scores for future work. For example, it is theoretically possible to weight the tweets, giving more weight to those closer to the Monday, to reflect a belief that *younger* sentiment scores reflect more relevant opinions, and so exponential smoothing could be used, for example.

1.2.3.3 Dummy variables

Certain facets of time-series data are deterministic: these dummy variables relate to the days of the week. Two dummy variables (DVs) were created in the hope that the component-wise boosting methodology would be able to use them; at the same time knowing, however, that they would not be promoted by the model if no effect was noticeable due to their inclusion. Both DVs relate to days and dates, with the first returning either a 1 or 0, stating whether the day is a Monday or not, respectively. The second DV answers a similar question, making use of the known public holidays (as discussed in Section 1.2.2), stating simply whether a day was a public holiday or not.

When creating the lagged variables for each data set (described in Section 1.2.4.1), these two DVs were able to, additionally, be included alongside the outcome variable, without being lagged. This is because, if it is believed that stocks perform badly on Mondays, it is known in advance that Monday is coming and so traders may adjust their strategies accordingly. The information is deterministic. For more ideas into the realm of deterministic cyclic trading patterns, see [**?**] and [**?**].

1.2.4 Final subsets for comparative modelling

The last step in data pre-processing was to organise the data in a way that allowed a direct measure of the benefit that adding sentiment data to market data supplied. To do this, five data sets were defined - their names, contents and reasoning are explained in Table 6. The data sets displayed were then later increased in size when predictor variables were lagged. The largest data set was the *combined* data set, with 693 predictor variables, which is two greater than the number of observations, 691[11].

[11] Without any lagged variables, the timeline includes 695 days; however, when creating lagged variables, the timeline is reduced in length by one day for each additional lag.

Subset name	$\sum$	**Contents**	**Reasoning**
$\text{traditional}_{\text{small}}$	6	A selection of six of market data variables: S&P500, gold (spot), oil, USD-EUR, and 1o year zero-coupon bond yield	A benchmark model that should perform satisfactorily, and allow for fair comparison with fitting methods that do not have inherent method of variable selection.
$\text{traditional}_{\text{large}}$	36	All market data, as listed in Table 1, plus dummy variables (Section 1.2.3.3)	To combine and showcase component-wise boosting, being able to select the strongest candidates for a model with traditional predictors.
$\text{sentiment}_{\text{small}}$	22	Sentiment results for each search term averaged over the five sentiment models, plus dummy variables (Section 1.2.3.3)	A compact collection of the sentiment analysis results. as with $\text{traditional}_{\text{small}}$, a benchmark model to show the predictive power of sentiment scores by themselves.
$\text{sentiment}_{\text{large}}$	100	All sentiment results: each search term passed through each sentiment model, plus dummy variables (Section 1.2.3.3)	To allow boosting to select the very best components of the sentiment analysis results. Also to facilitate direct comparisons to the combined data set.
combined	142	All variables: the summation of $\text{traditional}_{\text{large}}$, $\text{sentiment}_{\text{small}}$ and $\text{sentiment}_{\text{large}}$, plus dummy variables (Section 1.2.3.3)	To demonstrate the effect of combining sentiment analysis results with traditional market data. A large data set, exploiting the model's variable selection.

Table 2: The five different subsets created from the financial market and sentiment analysis data. The subset names, total number of variables ($\sum$), a summary of the constituents as well as a description of each data set is given. The last column, Reasoning, explains how the subsets were chosen to (1) demonstrate the ability of component-wise boosting, (2) to highlight the impact of social media data on predictive accuracy, and (3) to allow for comparisons to other models.

1.2.4.1 Lagged subsets

After the five different subsets were defined, several variations were made for each of them to include lagged predictor variables. For each of the subsets, four additional subsets were created, including lags of two to five with respect to the outcome variable; the DJIA. Each further degree of lagged variables was appended to the previous lagged subset. Using a fictional data set with a univariate outcome, y, and only one predictor, x, the resulting five subsets (including the base) may be illustrated as follows:

$$\begin{aligned}
&\textbf{Base subset:} && y_t = x_{t-1} \\
&\textbf{Second lag:} && y_t = x_{t-1} + x_{t-2} \\
&\textbf{Third lag:} && y_t = x_{t-1} + x_{t-2} + x_{t-3} \\
&\textbf{Fourth lag:} && y_t = x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4} \\
&\textbf{Fifth lag:} && y_t = x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4} + x_{t-5}
\end{aligned}$$

where intercepts and parameter coefficients have been omitted for simplicity. This was performed for each of the five subsets defined in Section 1.2.4, which means a total of 25 sets of data were defined. Each of these subsets was used for each of the parameter configurations within the component-wise boosting modelling phases - for more information, refer to Section 1.4.

1.2.5 Pairwise correlation reduction

As was discussed within Chapter `chapter-gradient-boosting`, the variable selection ability of component-wise gradient boosting does have limits. If two variables are highly correlated and so produce similar approximations to the negative gradient of the loss function, the model will have no way to really distinguish exactly which is the best. In such a case, this would lead to almost random selection between the two variables. In order to minimise the likelihood of this occurring during the modelling, as well as to improve the numerical stability of the gradient descent, a method to remove correlation within the data sets was devised; specifically, pairwise correlation was targeted.

The method used to purge pairwise correlation from the data set is detailed by Algorithm (1). The removal of variables was performed iteratively, re-calculating the remaining correlation within the data set after each individual variable was removed. This method presents a more systematic means of removing only those variables, which may impede the overall performance within the boosting procedure, described in Section `comp-alg`.

Algorithm 1: Iteratively removing pairwise correlation within a data set

Input: correlation matrix of data set, $\mathcal{C}$; maximum allowed pairwise correlation, κ
Output: data set with reduced pairwise correlation

1 **while** $\max \mathcal{C} > \kappa$ **do**

- Step 1. Identify the two variables exhibiting the highest pairwise correlation
- Step 2. Compute which has the greatest cumulative pairwise correlation over the entire data set
- Step 3. Remove this variable from the pair
- Step 4. Re-calculate the correlation matrix, having removed one variable

2 **end**
3 **return** *data set with* $\max \mathcal{C} \leqslant \kappa$

The maximum level of correlation, κ, to choose for each model is not something that can be analytically decided upon. Depending on the levels chosen, the number of variables that are removed from a data set can change rather drastically. Figure 2 illustrates the number of variables that are removed

from both the $traditional_{large}$ and *combined* data sets, as a function of the correlation threshold, κ. The left y-axis shows the number of predictors that are removed for a given κ, whereas the right y-axis shows that number as a percentage of the total number of predictors in that data set. The error bars highlight how many predictors are removed as κ crosses that specific threshold (reading κ from low to high). It can be seen that, even choosing a relatively high value for κ, e.g. ≈ 80 %, removes approximately 25 % of predictors for the $traditional_{large}$ data set, whereas more than 35 % of predictors are removed from the *combined* data set at the same level of κ. This shows that the level of correlation within the *combined* data set is larger than that of the (smaller) $traditional_{large}$ data set. This might be expected, as the *combined* data set contains e.g. five values (and so five predictors) of sentiment for each search term, one for each sentiment model - these should be highly correlated by nature.

As is outlined in Section 1.4.1, a selection of threshold values, κ, were used for modelling, meaning the effect of correlation within the data is able to be considered when inspecting the collated results. Both of the curves in Figure 2 are approximately linear. In the case of the $traditional_{large}$ data set, one may loosely keep in mind that the value of κ roughly equates to the percentage of original predictors that remain in the data set for modelling.

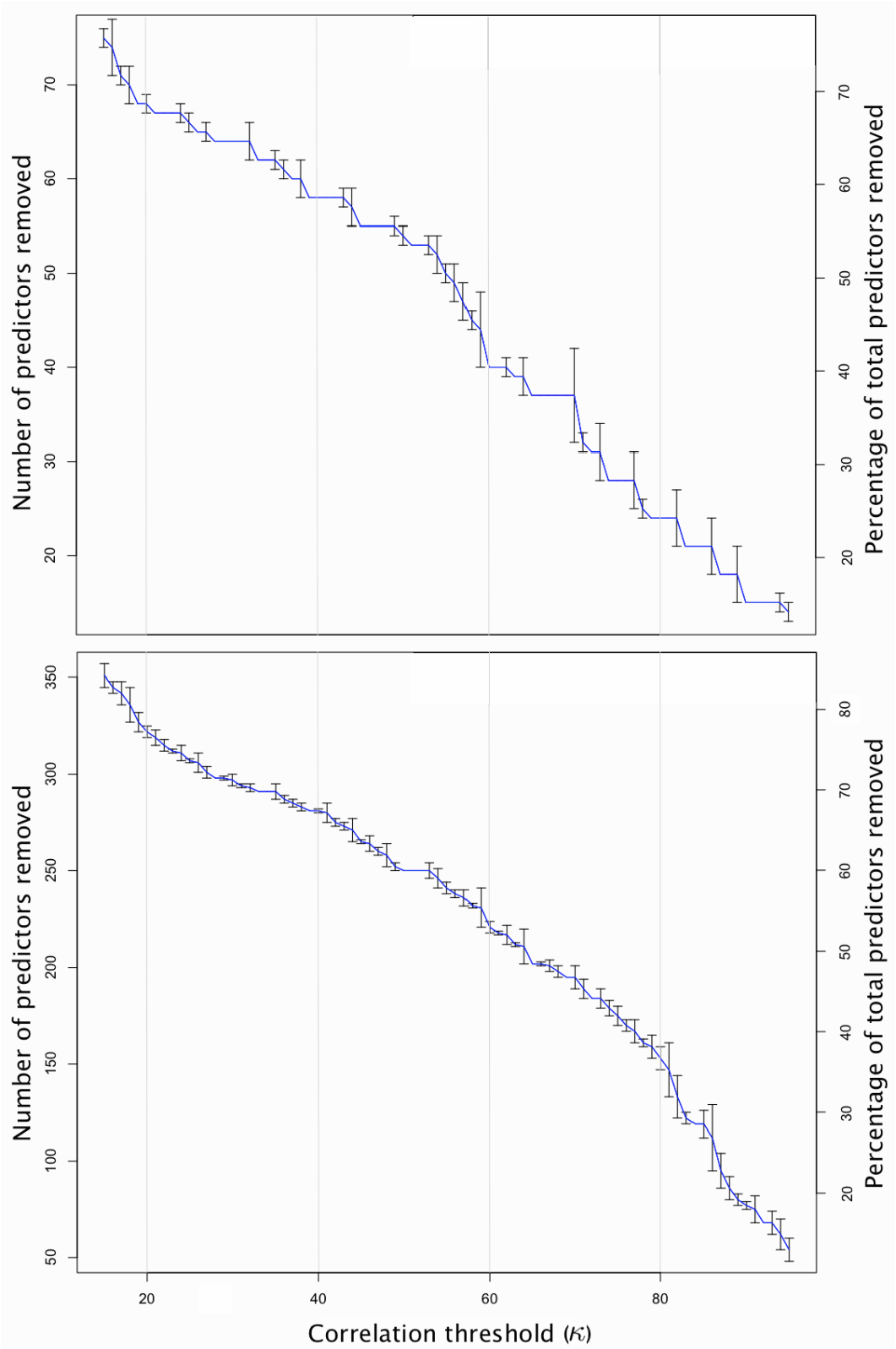

Figure 2: The number of predictors that are removed as a function of the correlation cutoff, κ. (Top) the $traditional_{large}$ data set with three lags; (bottom) the *combined* data set with three lags. The vertical bars placed along the curve reflect how many predictors are removed for the corresponding value of κ.

1.3 Data exploration

Before modelling commenced, the obtained, cleaned and collated data was explored and visualised in order to better understand the structure, and perhaps to gain some insights that may help with making modelling decisions as well as interpreting results. The main outcomes are presented here, allowing the reader to become acquainted with the data set. As the social media data and the sentiment analysis thereof is the novel segment, which this study aims to leverage, the presentation of the data will focus on this area, as well as its relationship to features of the outcome variable: the Dow Jones Industrial Average (DJIA) stock index.

1.3.1 Macro view with examples

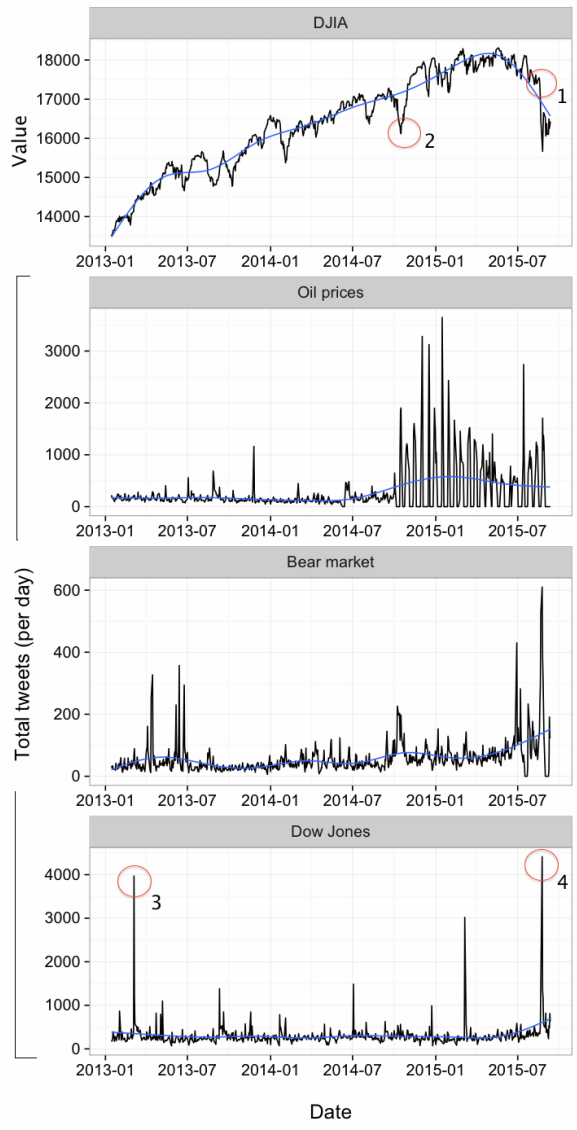

Figure 3: Individual plots for the DJIA and the tweet counts of three search terms (given in the facet titles), plotted over the entire time, each with a blue trendline. Several key events are highlighted and numbered.

As was presented in Section `final-output`, the total number of tweets obtained from Twitter was 2,350,217. Figure 3 gives a facet view of how the tweets for several of the thirteen search terms are dispersed over the timeline[12], i.e. the frequency with which the search terms appeared on Twitter. The blue line on each of the facet plots highlights the trend for that given variable, several points of interest are highlighted with red circles. The DJIA rises for the majority of the timeline, until it plateaus at the end of 2014, with a free-fall drop (labelled "1") at the end of summer 2015. An interesting correspondence is that between the astonishing increase in "oil prices" tweets, just **before** the DJIA takes a short dip and begins to plateau (labelled "2"). Additionally, the number of tweets containing "bear market" begins to rise almost a month before the sharp fall (marked "1"). The two circles peaks (labelled "3" and "4") on the bottom facet more than likely signify reactionary tweets to extraordinary market movements. The first can be traced to 28$^{\text{th}}$ March 2013, where the DJIA closed at a record high†. The circled peak "4" clearly aligns with circle "1", on 24$^{\text{th}}$ August 2015, on which day the DJIA plummeted over 1,000 points on negative news† regarding China's economy. These signs illustrate that there is a two-way relationship between the movements of the DJIA and the activity on Twitter. Some larger trends seem to be visible through the number of tweets (and likely through the resulting sentiment scores), whereas other features highlight purely the reactive nature of Twitter users to market events. It is the former, which the modelling is aimed at exploiting; the periods of momentum ought to be captured. How this is targeted is discussed further in Section 1.4.1.

1.3.2 Micro view

The previous section showed large peaks in the tweet count at potentially important events in the markets timeline; however, here a closer look is taken at the same relationship by inspecting the day-to-day movements of the market versus the

[12]The tweet data is aggregated to daily sums of tweets - see Section 1.2.1 for more information.

activity on Twitter. Figure 4 shows the relationship between the DJIA, log returns thereof and number of tweets computed using tweets containing "Dow Jones". The level of correlation between the log returns and the Twtter data are clear, with moves in Twitter data reflecting, and in the days immediately following 16th February, preceding those of the market.

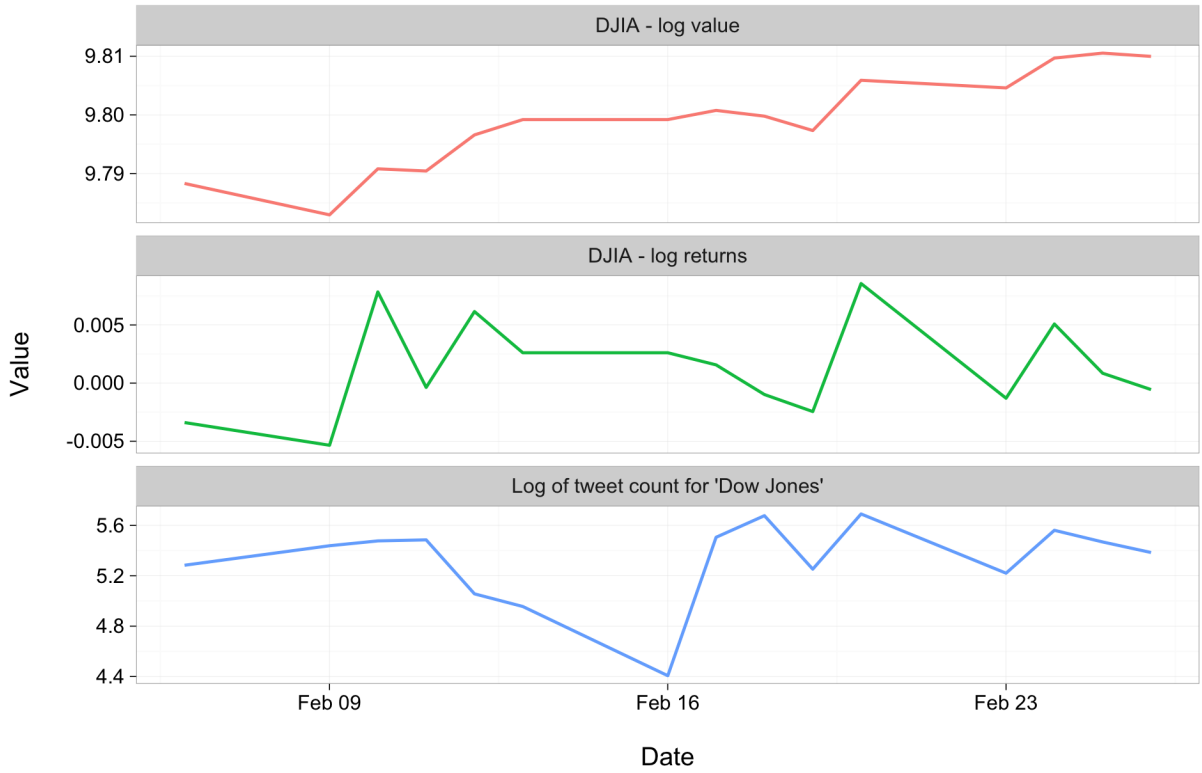

Figure 4: Daily movements from 2015 in the DJIA - as well as the log returns thereof - are plotted against the log of sentiment scores for the same period from the "Dow Jones" search term.

1.4 Generalised linear models

In this section, both a discussion of model parameters as well as results of the two main generalised linear models (GLMs) are presented, followed by a short comparison. Comparisons between the two GLM models, using component-wise boosting, and several differing models are provided in Section `results-summary`.

1.4.1 Parameter grid

Parameters specific to the individual boosting step of the modelling (using the `mboost` package) are the learning rate, ν, and the maximum number of iterations, m_{max}. A wide range were tested, resulting in a learning rate of $\nu = 0.05$, coupled with a maximum number of iterations $m_{max} = 2000$, which together sufficed for the algorithm to converge on all data sets. These values were therefore used, consistently, through all modelling variations. Using the bootstrap cross-validation method, outlined in Section `mstop`, meant that the optimal number of iterations m_{stop} could be determined in each individual case, tailoring the final model used for prediction to each data set[13].

Taking a step back from each gradient descent problem, the next level of abstraction for the models in general concerns the *time-series* nature of the data. It is desirable to make as many predictions as possible, allowing for the predictive accuracy and its errors to be computed accurately. For this reason, a final parameter was defined, namely the *frame-size*, which describes how many days were used to approximate the optimal function f^*, using the pre-defined model parameters ν and m_{stop} - the approximation function was then used to make one single prediction. As an example, using a frame-size equal to 40 means 40 periods of data were taken (40 days from the total 695), the boosting method generated the approximation function, which was subsequently used to predict the outcome variable on the 41^{st} day.

One further modelling decision had to be made, namely whether the frame-size should be held constant (shifting along the timeline, one period at a time, with each shift making one prediction), or whether the start point be anchored, thereby allowing the number of periods used in finding the approximation function to grow over the timeline. When conducting time-series analysis, there are no hard-and-fast rules governing how many time-periods must be included to guarantee model robustness[14]. It is a question whose answer changes depending on the data being used. There is a trade-off to be found between three main components: the number of periods available, the number of covariates used (i.e. the number of model parameters to be estimated) and lastly the level of noise within the data set.

There are additional factors that must be taken into consideration within the context of financial markets, and those are of trends and cycles - not to be confused with seasonal effects, tackled through time-series decomposition. There are times in which an asset (e.g. a single company stock, a commodity or an entire index) tends to move in one direction, i.e. it exhibits some level of momentum. The event of such a cycle changing may be labelled a *fraction* or *break* in the asset's price-path[15]. The approach taken here to deal with this facet of financial time-series is to make use of our final parameter, frame-size, which would ideally be matched in length to those of the trends and cycles. This is difficult (perhaps impossible) to know ahead of time - and cannot be embedded into the model via the analysis of historic data[16] without creating a bias in the predictions, as the information was not available at the time. Taking this into consideration, the choice was made to use a fixed frame-size for each model, shifting it along the timeline with each prediction. However, in order to test for ours model's sensitivity to the chosen frame-size, several values for this parameter were included in the parameter grid, namely 40 and 60 days.

In summary, the final set of parameters that were worked through included pair-combinations for the pairwise correlation threshold, κ, and the frame-size. For each of these pairs, every single subset (as defined in Section 1.2.4) was analysed and used to make predictions. The results are summarised in the following sections.

[13] In many cases, these values of ν and m_{stop} were unnecessarily high. The disadvantage of this being computational cost, whereas the risk that was being circumnavigated was that of non-convergence. The latter is not a problem that cross-validation could have solved.

[14] As illustrated in the case of ARIMA modelling by R. Hyndman†.

[15] For work on modelling assets in such a fashion, refer to Mandelbrot's Multifractal Model of Asset Returns (MMAR) **[?]** **[?]** **[?]**.

[16] It is not possible using the data of this study. One can, however, envisage using older DJIA time-series data to estimate an optimal *frame-size*.

1.4.2 Gaussian family

The Gaussian family utilises the L_2 loss function (as depicted in Section `naive-boosting` by (**??**) and (**??**)). Additional to several standard error measurements, we define a measurement, we shall call *predictive accuracy*, which is a simple test that produces a binary response, indicating whether the *direction* of movement was predicted correctly or not - the numerical discrepancy between the true and predicted values are not taken into consideration for this measure. In Figure 5, the column *Predictive acc.* reports the percentage of correct predictions measured by the sign accuracy, over the (695− *frame-size*) predictions that were made for each model. Looking at the upper row of facet plots, with the frame-size fixed at 40 days, the *combined* subset generally has the best performance, consistently appearing at the higher end of the predictive accuracy spectrum. The condensed $sentiment_{small}$ subset performed particularly well in cases with a lag of 1. As did the $traditional_{small}$ subset; however, that subset's performance decreased rather drastically with the increase of lag value, for all levels of κ. With frame-size set to 60 in the lower row of plots, it is the $sentiment_{large}$ subset that dominates the group, producing some of the highest predictive accuracies, approaching 57 %. The *combined* subset performs similarly for lag values $\geqslant 2$.

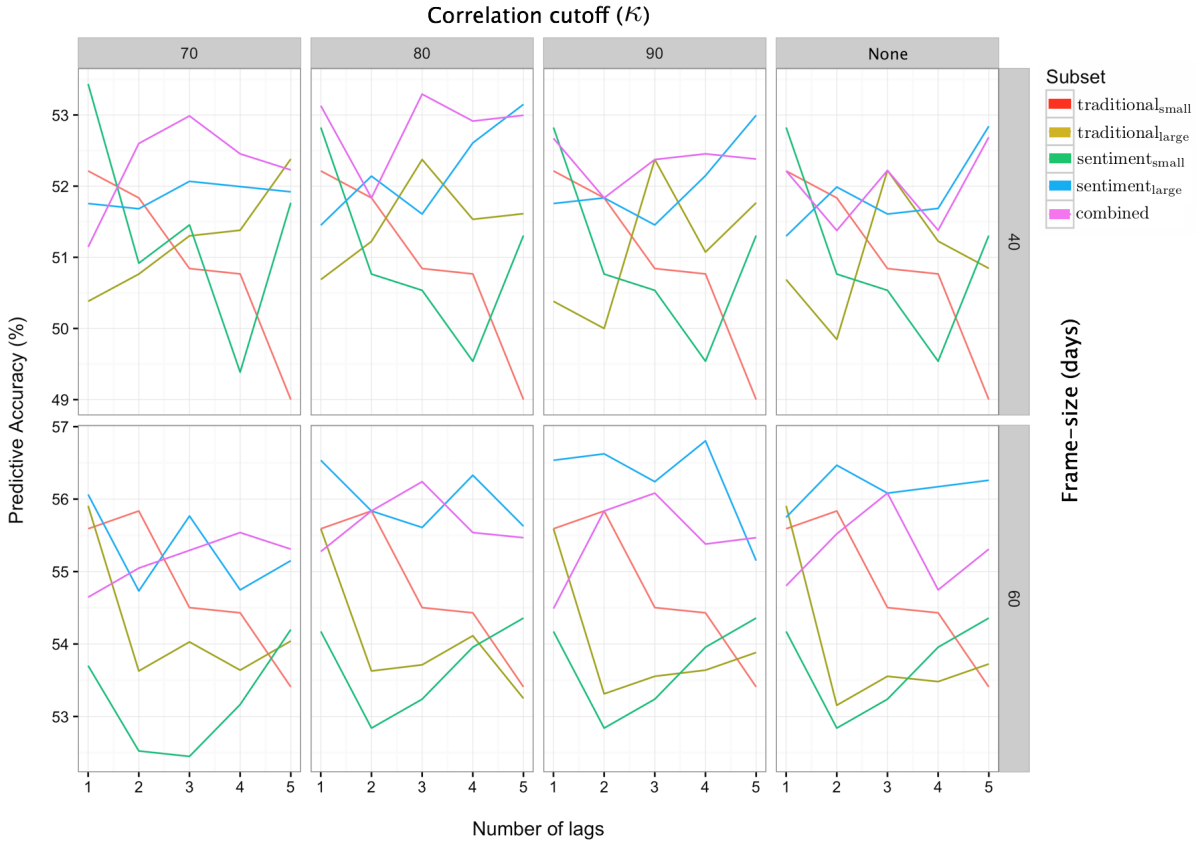

Figure 5: For each of the parameter-pairs, correlation cutoff κ and frame-size (upper row 40 days, lower row 60 days), the average predictive accuracy is plotted for each subset, for each of its five lagged variants.

In Figure 6, the mean-squared error (MSE) is given for each of the predictive accuracies presented in Figure 5. Each MSE value (just as with the predictive accuracy values) is the average value over the (695− *frame-size*) predictions that were made for each subset. Comparing the upper and lower rows of plots, the errors are further spread out from one another within the upper row, with frame-size 40. In the facets where $\kappa = 70$ % and 80 % (the two left-most columns), the errors of the *combined* subset are at least as good as for all other subsets, with low dispersion between lags and low absolute values - they are very comparable to those of the $traditional_small$ subset. However, comparing the predictive accuracy of the two subsets in Figure 5, it can be seen that the *combined* subset outperforms the $traditional_{small}$ subset in all but the first lag. In the plots where $\kappa = 90$ % and no correlation reduction was used (the two columns furthest to the right), there is a clear increase in error for both the $sentiment_{large}$ and *combined* data sets. This is likely due to them containing a large number of predictors, which compounds as higher

orders of lag are utilised (see Section 1.2.4 for the number of predictors each subset contains). It appears to be in the first lag, where the two subsets are penalised most heavily for their size, relative to the other subsets.

The errors of the $sentiment_{small}$ subset in the upper row, with a lag of 3, are noticeably larger than all others. A reason for the relatively high error value is unknown; however, the *consistency* of the error (all values seemingly identical) may be explained by the fact that very few predictors are removed from this subset during the correlation reduction step described in Section 1.2.5 - not unexpected, given the small number of predictors it begins with.

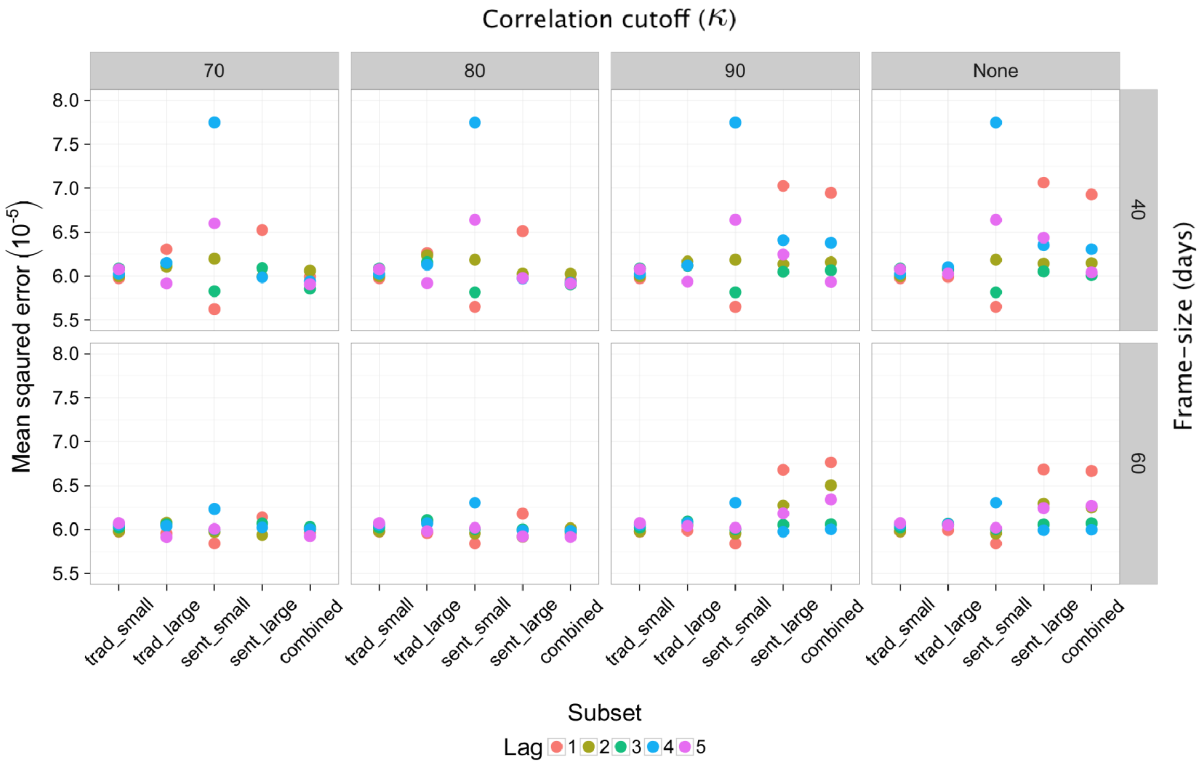

Figure 6: The mean-squared error for each of the results presented in Figure 5. Facets separate the frame-size and the correlation threshold κ (upper row: 40 days, lower row: 60 days). The errors for each lag value of each subset are grouped onto one vertical line.

1.4.3 Binomial family

Each of the models that were presented in the previous section were also completed using the binomial family within the `mboost` package. Using this family meant that the outcome variable automatically produced a binary response, $\{0, 1\}$, which corresponded to the model predicting whether the market moves upwards or downwards on the following day. This is somewhat of a simplification in comparison to the models using the Gaussian family, as the question of magnitude is no longer a concern. The goal, therefore, was to increase the defined metric on performance, i.e. the predictive accuracy, by slightly reducing the requirements of the model. When predicting a binomial response, the ways in which error can be recorded are naturally confined. Numerical residuals cannot be measured for each prediction, as in the previous section. The two methods will, therefore, later be compared solely via their predictive accuracies.

As was done for the Gaussian family results, all parameter combinations for the binomial family are presented in Figure 7. Inspecting the upper row, which corresponds to a frame size of 40 days, it is clear that the *combined* data set returns the greatest predictive accuracy for the majority of parameters combinations. The performance levels of the two *small* data sets are equally low, barely breaching 51.5 % predictive accuracy between them. The $traditional_{large}$ data set performs overall best in case where the lag was equal to five.

The lower row tells a more convincing story, with clear separation between the data sets that include

sentiment analysis results and those that don't. The *combined* and $sentiment_{large}$ subsets are clear victors across all values of κ and lag; the $sentiment_{small}$ data as performing markedly better than the two *traditional* data sets. The *combined* data set shows the highest predictive ability overall, with the $sentiment_{large}$ data set performing equally well when a larger frame-size was used.

The models using a frame size of 60 days consistently outperform those using 40 days, with a relatively large improvement in predictive accuracy for almost every subset. The maximum predictive accuracy achieved in the latter equals that of the the worst in the former ($\approx$ 53 %). These results validate the relationship pointed out in Section 1.3.1, namely that long-term market momentum having ties with activity in social media. The inclusion of social media data, in this case, unquestionably improved the performance of the model.

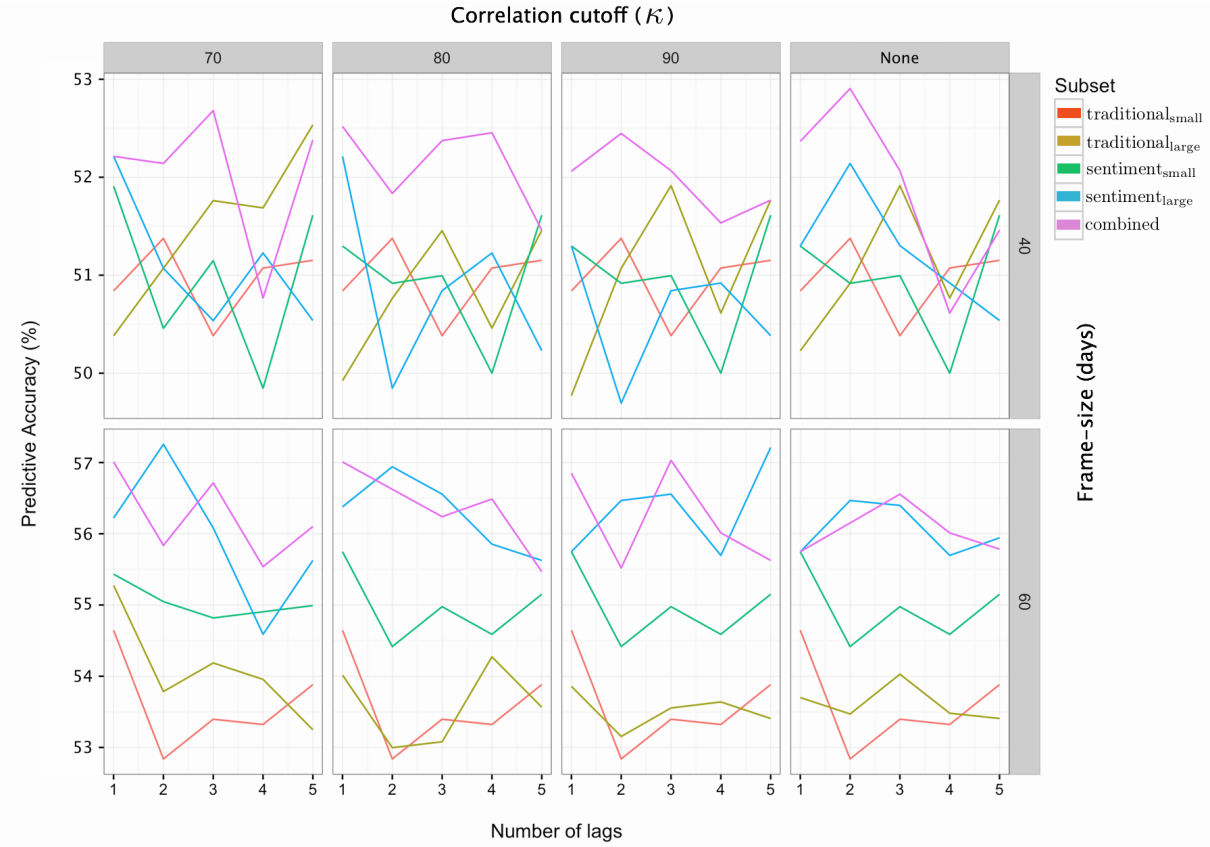

Figure 7: The predictive accuracies are plotted for the binomial model, separated by the frame-size and correlation threshold, κ (upper row: 40 days, lower row: 60 days).

The top 50 results over all subsets and parameters combinations are given in Table 3, ordered according to predictive accuracy. The first *traditional* subset appears at position 49. One thing that is highlighted by the results in that table is that the frame-size value of 60 dominates the results. This indicates that cycles of momentum in the DJIA (discussed briefly in Section 1.4.1) appear to be best captured over a phase of 60 days[17]. There is less consistency in the further parameters, κ and lag. There is a general tendency for lower values of lag (1 to 3) to perform better.

[17]In the same table for the Gaussian results, the 60 day frame-size also dominates. The highest position a *traditional* subset reached was 15^{th}; a total of 12 *traditional* subsets appeared in the top 50.

Rank	Frame-size	κ (%)	Lag	Subset	Pred. accuracy (%)
1	60	70	2	$\text{sentiment}_{\text{large}}$	57.26
2	60	90	5	$\text{sentiment}_{\text{large}}$	57.21
3	60	90	3	combined	57.03
4	60	70	1	combined	57.01
5	60	80	1	combined	57.01
6	60	80	2	$\text{sentiment}_{\text{large}}$	56.94
7	60	90	1	combined	56.85
8	60	70	3	combined	56.71
9	60	80	2	combined	56.62
10	60	80	3	$\text{sentiment}_{\text{large}}$	56.56
11	60	90	3	$\text{sentiment}_{\text{large}}$	56.56
12	60	none	3	combined	56.56
13	60	80	4	combined	56.49
14	60	90	2	$\text{sentiment}_{\text{large}}$	56.47
15	60	none	2	$\text{sentiment}_{\text{large}}$	56.47
16	60	none	3	$\text{sentiment}_{\text{large}}$	56.40
17	60	80	1	$\text{sentiment}_{\text{large}}$	56.38
18	60	80	3	combined	56.24
19	60	70	1	$\text{sentiment}_{\text{large}}$	56.22
20	60	none	2	combined	56.15
21	60	70	5	combined	56.10
22	60	70	3	$\text{sentiment}_{\text{large}}$	56.08
23	60	90	4	combined	56.01
24	60	none	4	combined	56.01
25	60	none	5	$\text{sentiment}_{\text{large}}$	55.94
26	60	80	4	$\text{sentiment}_{\text{large}}$	55.85
27	60	70	2	combined	55.84
28	60	none	5	combined	55.78
29	60	80	1	$\text{sentiment}_{\text{small}}$	55.75
30	60	90	1	$\text{sentiment}_{\text{small}}$	55.75
31	60	90	1	$\text{sentiment}_{\text{large}}$	55.75
32	60	none	1	$\text{sentiment}_{\text{small}}$	55.75
33	60	none	1	$\text{sentiment}_{\text{large}}$	55.75
34	60	none	1	combined	55.75
35	60	90	4	$\text{sentiment}_{\text{large}}$	55.70
36	60	none	4	$\text{sentiment}_{\text{large}}$	55.70
37	40	80	3	$\text{sentiment}_{\text{small}}$	55.65
38	40	90	3	$\text{sentiment}_{\text{small}}$	55.65
39	40	none	3	$\text{sentiment}_{\text{small}}$	55.65
40	60	70	5	$\text{sentiment}_{\text{large}}$	55.63
41	60	80	5	$\text{sentiment}_{\text{large}}$	55.63
42	60	90	5	combined	55.63
43	60	70	4	combined	55.54
44	60	90	2	combined	55.52
45	60	80	5	combined	55.47
46	60	70	1	$\text{sentiment}_{\text{small}}$	55.43
47	40	70	3	$\text{sentiment}_{\text{small}}$	55.31
48	40	90	1	$\text{sentiment}_{\text{large}}$	55.29
49	60	70	1	$\text{traditional}_{\text{large}}$	55.28
50	40	80	4	$\text{sentiment}_{\text{small}}$	55.24

Table 3: The top 50 predictive accuracies from the binomial models. The results are dominates by subsets containing sentiment analysis data. The best performing traditional model appears at position 49.

1.4.4 Family comparisons

Inspecting first the predictive accuracy results in Figures 5 and 7, as well as the detailed results found in Table 3 for the binomial results, we see that the best performers are unequivocally those containing social media and sentiment analysis data. The magnitude of the predictive accuracies are very similar between the Gaussian and binomial sets of results, ranging from 50 % to 53 % for frame sizes of 40, and from 53 % to just over 57 % in the case of a 60 day frame-size. The clear distinguishing feature between the two families is the separation of performance between models containing social media data and those that didn't, that the binomial family was able to accentuate. Other than this difference, the two models are not easy to distinguish between in terms of their performance. Due to the binomial model not returning MSE values, it is not possible to compare the errors of the two models.

Almost each model outperformed a naive (random selection) model giving 50% predictive accuracy. Only in the cases of the small data sets; $traditional_{small}$ and $sentiment_{small}$ in lag values four and five, with a frame size of 40 days, were any results worse than 50 % found.

1.4.5 Prediction-generated price paths

Figure 8 visualises the results in direct comparison to the DJIA itself, by plotting its price-path over the entire timeline against those price-paths that were predicted (on a day-by-day basis) by several different models. In order to have measurable magnitudes of price movement, the results from the Gaussian GLM models are used. The prices follow from a nominal base-value of 100. We take the *combined* data set that performed the best over all parameters configurations, namely for: frame-size = 60, $\kappa = 80$ % , with a lag of 3. In comparison, we take the $traditional_{large}$ data set (thereby containing all the same financial market data, but without the social media data) for the same parameter combinations, plus a third data set, which is the best performing *traditional* subset. That used the parameters frame-size = 60, $\kappa = 70$ %, with a lag of 1.

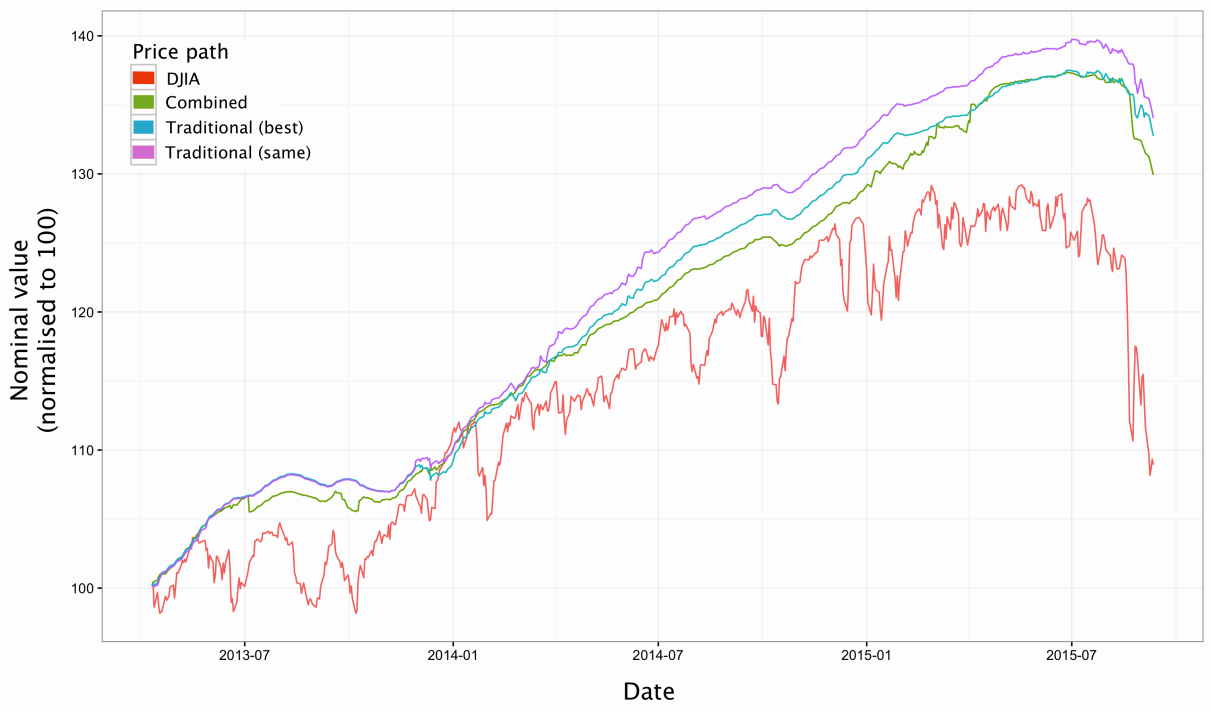

Figure 8: Four price-paths are plotted over the entire timeline: the DJIA plus those of the three best performing *combined* and *traditional* data sets, as described in Section 1.4.5.

All three of the predicted returns manage to map to the movements of the market fairly well - following similar patterns of peaks and troughs. It is the *combined* data set that most closely resembles the price-path of the the DJIA, more intimately tracking movements and reacting more sharply to large jumps the DJIA made. For example, all three subsets move as expected on the news of the huge drop on 24th August 2015 (discussed in Section 1.3.1); however, it is the *combined* data set, including social media data, which falls earliest and most rapidly, thereby most closely reflecting the DJIA. All three

of the subsets produce paths that somewhat resemble a moving average of the DJIA itself, all with an upward bias and without the high levels of granularity exhibited by the DJIA.

1.4.6 Increasing frame-size

Throughout the modelling presented thus far, the frame-size used to train an approximation function was fixed at either 40 or 60 days. The reasons for which are discussed in Section 1.4.1. Here we present a set of predictive accuracy results that were created - for the Gaussian and binomial models - using an *increasing* frame-size. An initial frame-site of 60 days was used; however, instead of shifting the frame along one day into the future to train a new approximation function and, from that, make one more prediction, the frame-size was increased by one day and **not shifted** - keeping the start of the frame-size anchored at day 1 in the timeline. This means that, for each prediction, the maximum amount of information available within the data set was used, i.e. all days in the timeline up until the day before the prediction. This is to test the assumption that the frame-size corresponds to general trends and phases within the price development of the DJIA. The approach presented here has the advantage that the maximum possible amount of information is used for every prediction (in terms of the timeline), but has perhaps a disadvantage in that it does not necessarily capture the prevailing momentum of the market at the time of the prediction.

In addition to an (initial) frame-size of 60 days being used (as with all other comparisons made to the results from Section 1.4), a correlation threshold of $\kappa = 80$ % was used. All other model parameters, such as shrinkage, $\nu = 0.05$, and maximum number of boosting iterations, $m_{max} = 2000$, were kept identical to those used in Section 1.4, for both the Gaussian and binomial models. Figure 9 presents the predictive accuracies of the two comparisons, with the left column displaying the results of the binomial model and the right column those of the Gaussian model. In both columns, the upper row gives the results obtained earlier from a fixed frame-size of 60 days, whilst the lower row gives those for an increasing frame-size.

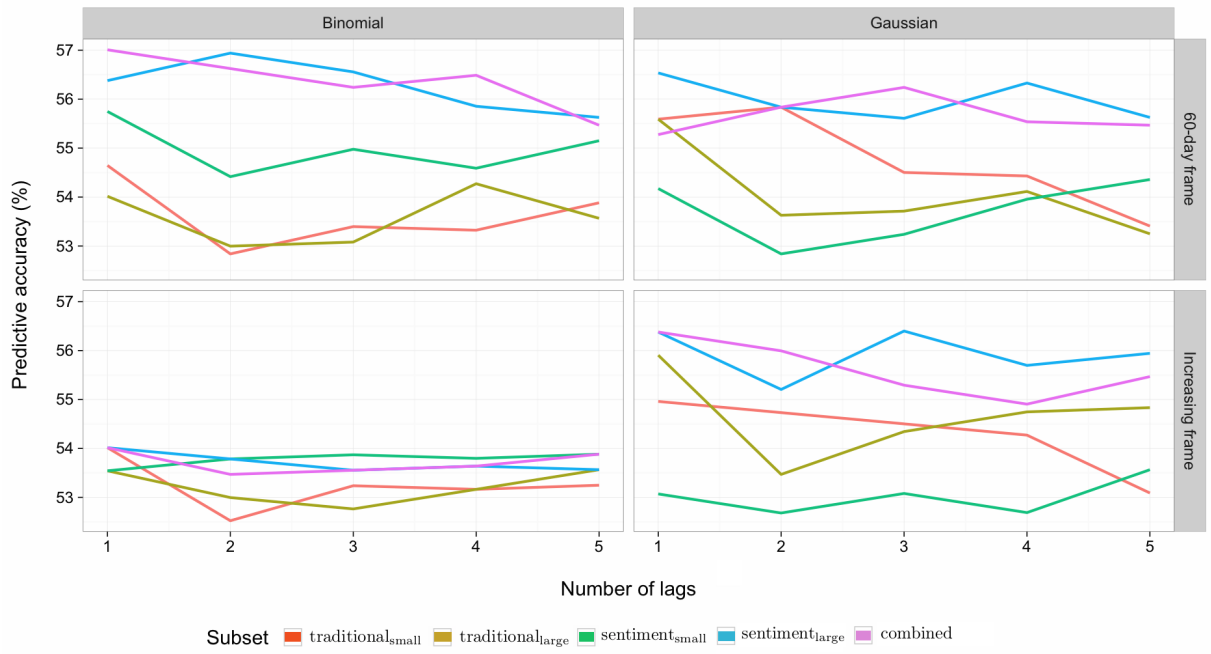

Figure 9: A comparison of predictive accuracies between models with fixed frame-size and increasing frame-size. In both cases, results using binomial (left) and Gaussian (right) modelling are plotted. Each facet contains the results for all lagged variants of all five subsets.

Inspecting first the binomial results, it is evident that the use of an increasing frame-size significantly diminishes the performance of the model. All predictive accuracies are compressed to below 54 %, with the *traditional* subsets still showing weaker performance than those subsets containing social media data. While still performing worse than in the case of a fixed frame-size, the $sentiment_{small}$ subset did manage to surpass the other subsets in all but the first lagged variant, when using the increasing frame-size. This may be due to more information being provided (due to a higher average number of periods used for

each prediction), while still being compactly included in fewer covariates than in the other models.

In the case of the Gaussian models, the performance is very similar between the models using fixed frame-size and those with increasing frame-size. The predictive accuracy of the *combined* data set increases in the first lag, from 55.5 % with fixed frame-size to 56.5 % with increasing frame-size. It is indistinguishable in the second lag ($\approx$ 56 %), while the performance is reduced for lags three, four and five. It is perhaps juxtapositional to state that the high number of variables, coupled with ever increasing numbers of observations overwhelms the boosting algorithm because, contradictorily, the performance of the $sentiment_{large}$ data set actually improves along the same dimension of comparison - it shows higher performance in lags three and five for the increasing frame-size method than that of fixed frame-size. The $sentiment_{small}$ data set performs consistently worse using an increasing frame-size that fixed, which may reflect the model losing its ability to explain market momentum when presented with longer time-frames, using only its succinct collection of predictors. Sentiment seems only to aid with prediction in the shorter term. This can also be seen when looking at the simulated price-paths illustrated in Section 1.4.5, where the *combined* data set was better attuned to market movements than its $traditional_{large}$ counterpart in the short-term.

Comparing the binomial and Gaussian models, it seems the Gaussian regression methods was better able to cope with the higher influx of data at each iteration, whereas the binomial model was more greatly, negatively, affected.

1.5 Stochastic gradient boosting

1.5.1 Parameter tuning

This section presents results obtained from stochastic gradient boosting, which was performed for a binary response variable, i.e. whether the market rose or fell. This makes for good comparison to the results presented in Section 1.4.3, however the methodology does differ in more than one way. A detailed discussion on stochastic gradient boosting may be found in Section 1.5. It must be made very clear that the algorithm used here is **not** component-wise gradient boosting, but rather the *batch* gradient boosting, outlined in Section `naive-boosting`. As this stochastic methodology is quite different, the aim here is merely to achieve the best *predictive accuracy* possible using the stochastic approach. Results are then be compared to those presented in Section 1.4.3 via this metric, which is validated by the choice of input data (see below). Using the *batch* gradient descent method does raise concerns about performance in the face of wide data sets, as there is no in-built variable selection. We aim to test the abilities the method has through its added factor of random predictor selection, which essentially replaces that of the more systematic and statistically tuneable way of sequential variable selection, carried out in component-wise gradient boosting. The added randomisation does not compensate directly for this difference, however it is designed (adapted to utilise properties of random forests) to have the ability of variance reduction within a model (see section 1.5).

In order to compare the performance of a stochastic implementation of gradient boosting as fairly as possible with that of the GLM models using component-wise boosting, an identical subset of this study's data set was used. Namely that with a correlation threshold, $\kappa = 80$ %, with lag values one through five. The methodology followed here does mean that the *frame-size* parameter is lost. This is because, instead of using a rolling-window approach, train a test sets were created instead of sequentially fitting a model and making a prediction at one-period intervals along the timeline[18]. The 25 data sets - the five lags of each of the five subsets - were each divided into 65 % training data and 35 % test data, using stratified sampling. If it is assumed that the market rose and fell with a ratio of e.g. 8:10 over all periods, then *stratified* sampling simply signifies that the training and test sets created also contain this ratio of days, on which the market rose and fell. A model was then trained on the training set (of 452 days) and the test set (243 days) was used to make predictions. The predictive accuracy was then defined as the percentage of days on which the movement of the market was correctly predicted, out of the 243 predictions made.

Using this method introduces several new paramters to the model, the most interesting of which being the fraction of predictors[19] that are randomly selected at each iteration of the gradient descent process, π. This paramter generally leads to a slower descent of the loss function, meaning a larger number of

[18]This was also because the limitations of the R package `gbm`, which did not to be function when using the narrower of the data sets, e.g. $traditional_{small}$ and $sentiment_{small}$.

[19]In his original paper [**?**], Friedman used π to symbolise a random permutation of the data, from which the fraction was taken. We use π simply to symbolise the fraction, or sub-sample, of of predictors used to fit the base learner in a given step.

iterations are required. Furthemore, it allows several other *new* hyperparamters to be tuned. To optimise for these, a new parameter tuning grid was first defined to test over a wide range of combinations[20]. The metric used to tune the paramters was the area under the receiver operator characteristic† (AUROC) [**?**]. This tuning was performed individually for each of the data sets, as their differences in size may have been better suited to different variations to the model parameter space. Table 4 shows the values that were tested for, as well the the values for each parameter that were given as optimal in the case of the *combined* data set with lag value of 5 - the largest data set. The optimal random selection fraction, π, was found to be 0.5, which coincides with the range suggestion made by Friedman [**?**]. The maxmimum number of boosting iterations, m_{max} was given to be 3000, which is 50 % more than was found to be a reasonable maximum for the component-wise boosting. Again, the number of iterations required to reached the loss-function's minimum is expected to increase, as the introduction of a stochastic process perturbs the path of steepest descent - the added benefit (as discussed in Section 1.5) being that variance in results is likely reduced. An optimal learning rate of $\nu = 0.01$ further reflects that the algorithm is required to be a slow learner with the involvement of a stochastic process.

Parameter	**Value range**	**Desctiption**	**Optimum**
π	0.3, 0.5, 0.7	The fraction of parameters selected at each iteration	0.5
m_{max}	100 - 5000 (intervals of 500)	The number of iterations	3000
ν	0.01, 0.05, 0.1	The learning rate, or shrinkage	0.01

Table 4: The tuning grid used to find optimal parameters for the stochastic gradient boosting algorithm, for binary response modelling.

1.5.2 Comparison to component-wise boosting

The results of the stochastic gradient boosting are compared to those of the GLM model using component-wise boosting, originally presented in Section 1.4.3. As was mentioned there, the two methods are similar, but not the same. Therefore, any comparisons deeper than that of the pure outcome, the *predictive accuracy*, are not possible within the scope of this limited superficial differentiation of their nuances.

It can be seen from Figure 10 that the results obtained from the stochastic boosting are not as strong as those from component-wise boosting, with several models failing to beat even a naive model with predictive accuracy of 50%, with the performance of the $traditional_{large}$ subset dropping below 47.5 % in one case (in lag 1). Furthermore, the stochastic boosting results do not show any signs of reduced variance[21], when compared to the component-wise boosting results. The relatively large spread in performance within the stochastic boosting results is clear to see, with almost 10 % difference between the best ($sentiment_{small}$ in lag 2) and worst ($traditional_{large}$ in lag 1) performers, which is larger than any other set of boosting results found in this study.

The ability of stochastic boosting to deal with the larger data sets seems to also be inferior to that of component-wise boosting. This is illustrated by the superior performance of the smaller data sets, with both the $sentiment_{small}$ and the $traditional_{small}$ subsets outperforming their *large* equivalents in four out of five lag variants. The largest subset: *combined*, performs unexpectedly poorly, given its comparatively high levels of predictive accuracy in all parameter combinations used in component-wise boosting, depicted in Sections 1.4.2 and 1.4.3.

[20] As was mentioned, the frame-size no longer plays a role, and the value of κ remained fixed at 80 %.

[21] The reduction in variance through the addition of a stochastic process is usually detectable through lower errors (and variance thereof) on predictions; however, these are not available here, having performed logistic regression.

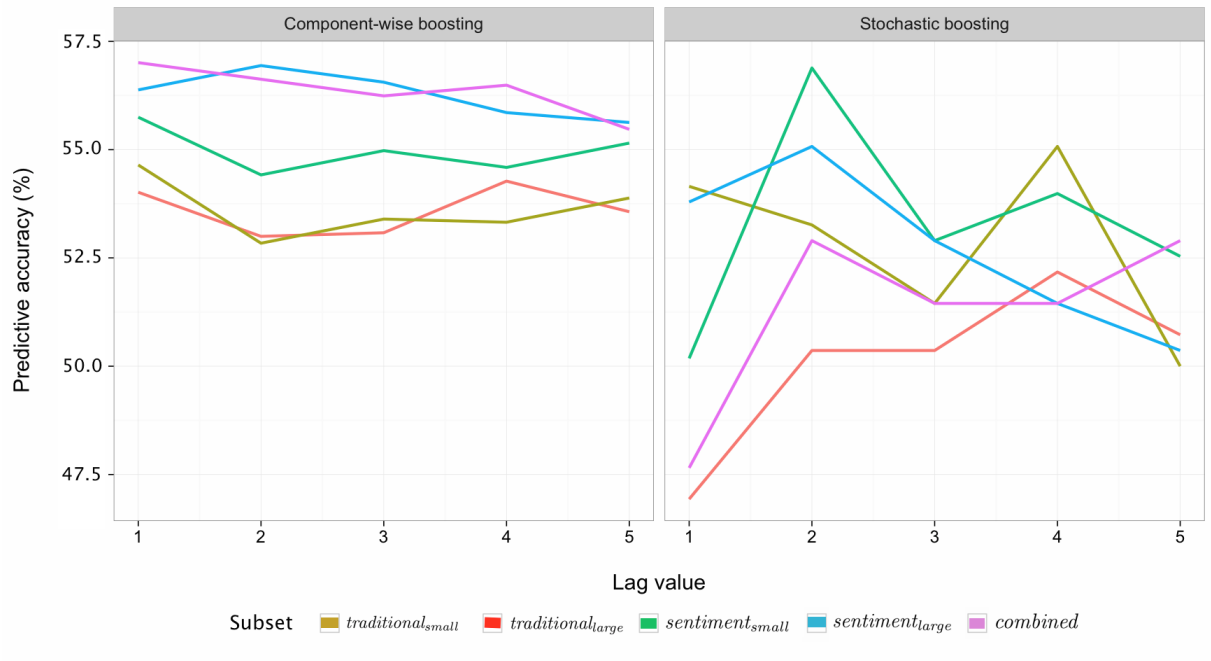

Figure 10: Predictive accuracies for component-wise boosting (left) are plotted alongside those using stochastic gradient boosting (right). Both results stem from identical input data sets, with pairwise correlation threshold, $\kappa = 80$ %.