Structure in the Tweet Haystack: Uncovering the Link between Text-Based Sentiment Signals and Financial Markets

Axel Groß-Klußmann*
Quoniam Asset Management GmbH

Markus Ebner
Quoniam Asset Management GmbH

Stephan König
University of Applied Sciences and Arts, Hannover

This version: October 2015

Abstract

We examine the relationship between signals derived from unstructured social media microblog text data and financial market developments. Employing statistical language modeling techniques we construct directional user sentiment and non-directional topic disagreement metrics and link these to S&P 500 index returns and volatility. Based on an extensive five year sample of Twitter messages our study shows that both unsupervised and supervised statistical learning methods successfully identify subsets of expert users with distinct finance focus. This allows to filter out the substantial noise associated with social media. Accounting for salient properties of the time series in ARMA models we document significant effects of expert disagreement signals on current and future S&P volatility. Moreover, we detect a significant contemporaneous relation between expert sentiment signals and S&P returns.

Keywords: Natural Language Processing, Sentiment Analysis, Unstructured Social Media Data, Big Data

[*] Corresponding author. Email: axel.grossklussmann@gmail.com. We acknowledge support from the equities and multi asset teams of Quoniam Asset Management, in particular we thank Thomas Kieselstein and Volker Floegel. The views and opinions expressed herein are those of the authors and do not necessarily reflect the views of Quoniam Asset Management GmbH.

1 Introduction

The last years have witnessed a steady rise in information shared through social media venues on the worldwide web. In consequence, more and more research is devoted to processing the vast volumes of unstructured data in order to gain deeper understanding of a wide range of real life aspects of its users. Important examples include attempts to track influenza epidemics (Broniatowski et al. (2013), Lamb et al. (2013)) as well as improving election forecasts (Tumasjan et al. (2010)). Antenucci et al. (2014), for instance, construct a social media job market indicator that supports the assessment of jobless claims in the USA. However, messages posted via social media websites are often informal by nature, thus complicating statistical inference substantially.

This paper addresses the challenge of relating author sentiment as well as author disagreement indicators from unstructured large-scale volume social media text data to financial market developments. Such indicators are devised to quantify the underlying tone or user disaccord in documents based on, e.g., counting negative and positive words. The fundamental objective of this study is to assess the potential of these social media signals for explaining and predicting broad financial market developments. Specifically, we link S&P 500 index returns and volatility to signals based on microblog messages, so-called tweets, on the Twitter network (www.twitter.com). In a novel approach to the problem we employ techniques from computational linguistics to automatically identify network users with a distinct focus on financial contexts. Thereby we reduce the considerable amount of irrelevant information associated with social media platform messages. Another feature of the present study is the extensive five year sample of tweets, which allows for robust conclusions about the relationship between time series of text-based metrics and financial market developments.

In the present study we aim to address the following research questions:

(i) Are broad financial market movements in terms of index returns and volatility significantly reflected in directional sentiment and non-directional disagreement signals derived from tweets?

(ii) Can we reduce the off-topic noise in tweets by automatically classifying twitter users into experts with finance focus producing strong text signals and non-experts with distinctly weaker signals?

Due to the large amount of data that need to be processed, research question (i) is typically addressed with time series shorter than a year and still lacks an empirical long-term analysis (see, e.g., Sprenger et al. (2014) amongst others). We answer question (i) in a time series analysis for financial market variables. To further clarify the interplay between tweets and financial markets we propose a novel systematic way to deal with the noise inherent in tweets, answering (ii). In specific, we address question (ii) with the help of methods from the natural language processing and machine learning literature. The classification into experts and non-experts is solely based on the linguistic similarity of the tweet text to expert vocabulary. Together, these questions provide first evidence about the potential of large-scale data mining approaches on social networks to become building blocks in financial applications.

Robust empirical support exists for the hypothesized interdependence between financial time series and structured text data like, e.g., earnings press releases for companies. Loughran and McDonald (2011), Li (2008) as well as Davis et al. (2012), for instance, quantify the text in official company earnings disclosures and document a positive relationship between linguistic signals and the financial performance of the respective companies. In a similar vein, Tetlock (2007), Tetlock et al. (2008), Groß-Klußmann and Hautsch (2011) and Gurun and Butler (2012) utilize numerical measures from news stories to analyze the interaction between the media and the stock market. Antweiler and Frank (2004) and Das and Chen (2007) are early analyses linking financial market activity to signals based on textual data from internet stock message boards. More recently, attention has been drawn to the Twitter platform, with Bollen et al. (2011) and Sprenger et al. (2014) among others reporting conflicting evidence about the predictive content of tweet sentiment measures for stock returns. In contrast to the typically unfiltered and noisy message stream in most text-mining studies, Chen et al. (2014) explicitly focus on financial expert communities on www.seekingalpha.com. They detect an economically meaningful relation between social media mood and future stock returns. Notably, the study indicates the need to identify finance experts on social media platforms in order to reduce the noise in the messages.

Using 14.4 million tweets from a finance-related subset of all tweets posted from January 01, 2010 through December 31, 2014 we derive indicators capturing the sentiment of tweets and disagreement among users on a given day. To reduce the noise stemming from off-topic messages, we first identify financial experts before subsequently computing sentiment and disagreement measures for the expert sets. Based on a small pre-labeled training sample of tweets from expert users we employ supervised learning methods like Naive Bayes and Support Vector Machines to classify all tweets into expert and non-expert tweets. To avoid spurious results due to an arbitrarily selected training sample of experts we show that meaningful subgroups of twitter users naturally emerge from a cluster analysis via the K-means algorithm applied to the text data. In correlation and time series analyses we relate the directional author sentiment metrics to financial market returns, while the non-directional author disaccord measure is linked to market volatility.

A major finding of our study is a significant relationship between expert sentiment and expert disagreement measures and the aggregate stock market return and volatility. In the course of the study we show that the noise in social media can be dramatically reduced through clustering and the classification of experts whose text passages exhibit close linguistic proximity to the finance topic. The relation between financial markets and text-based signals is shown to be robust and in line with economic theory. Past and concurrent non-directional expert user disaccord metrics are significant regressors in autoregressive moving average (ARMA) models of stock price volatility. Finally, ARMA models for stock price returns reveal a significant contemporaneous relationship between expert sentiment metrics and returns.

The remainder of the paper is organized as follows. In the next section, we describe the data mining and data processing. Section (3) covers the text data model and the machine learning approaches used for the classification and clustering. In Section (4), we outline the classification and clustering results as well as the correlation and time series analysis of stock market developments and twitter signals. Section (5) concludes.

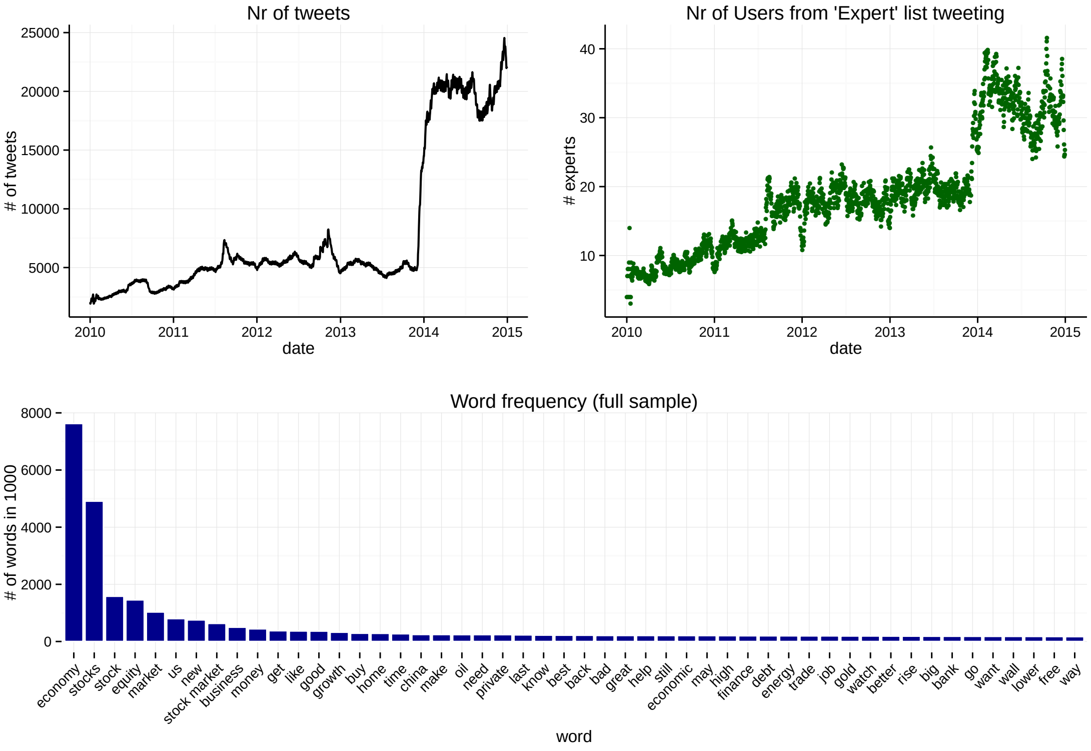

Figure 1: Top left: Evolution of tweet numbers (smoothed). Top right: Evolution of number of users from expert list sharing tweets. Bottom panel: distribution of top 50 words

2 Data and Initial Processing

2.1 The Tweet Sample

The textual data used for the present study consists of microblog messages with up to 140 characters on the Twitter social media platform. To concentrate more on the finance topic and reduce the amount of data to be processed, we focus on tweets containing at least one of the words in the searchterm list {'stocks', 'stock market', 'economy'}. In addition to the pure text, we obtain meta information on the user sharing the message as well as the exact time and language.

Using the Java programming language to connect to the Twitter streaming API we automatically downloaded tweets from November 30, 2013 to December 31, 2014 in real time. The public stream is constrained to 10 % of all available messages on a given day (sampling scheme 'GardenHose'). However, due to the search term list restriction this constraint is not relevant to our analysis. Moreover, we augment the data by historic tweets during January 01, 2010 to November 29, 2013 obtained from the Gnip, Inc., company, now a subdivision of Twitter. In this second subsample, however, the daily amount of tweets is constrained to 1 % of the overall tweet volume (sampling scheme 'Spritzer'). This is achieved through random sampling of the tweets containing one of the searchterms.

The data amounts to 14,420,749 tweets, with 6,6689,642 tweets in the Gnip sample before Nov. 2013 and 7,731,107 tweets from the Twitter streaming API. Considering the focus on pre-defined searchterms, the volume of data is remarkable. The top left panel of Figure 1 shows the evolution of daily tweet numbers over time. The histogram of words occurring in tweets is given in the lower panel.

A visual inspection of the tweets reveals that the searchterm restriction is not sufficient to isolate tweets with finance focus. The example below shows two typical tweets in the sample.

@User1; 2013-08-31 10:53:30.000: 'hi @flyPAL is it possible to book a business class fare only in one way fare? my return flight is economy class? thanks.'
@User2; 2013-08-31 05:01:42.000: 'New post: US economy expands at stronger rate in second quarter, figures show http://t.co/YGlgQtBCjw&Hannamiller777'

To identify subgroups of finance tweets we therefore aim to discover finance 'experts', i.e. users who tend to share tweets on the finance subtopic exclusively. Several websites portray such twitter experts for finance. Screening these websites we identified 190 experts and additionally inspected individual tweets on Twitter to confirm the selection. The appendix 6.1 gives the websites used in this study through 12/31/2014.

Most experts unfortunately post tweets irregularly, which results in a range of only three to a maximum of forty-two experts issuing tweets per day. The top right panel of Figure 1 shows the expert users by day. Signals computed based on such a small text sample are typically overlaid by noise. Moreover, in view of the millions of Twitter users, handpicking more experts is practically infeasible. In order to make it possible to still derive precise 'expert' signals from the tweet text data in an automated way we enlarge the list of twitter experts with the help of techniques from linguistic pattern recognition.

2.2 Text Data Processing

Sentiment measures of a text typically utilize positive and negative word lists as contained, e.g., in the Harvard General Inquirer IV-4 psychological dictionaries available at http://www.wjh.harvard.edu/~inquirer/. The Inquirer word lists are built upon categories of the well-known Harvard psychosocial dictionary and can be seen as established tools for the content analysis of textual data (see Stone et al. (1966)). The words and attributes are organized in comma-separated spread sheets as follows.

ENTRY	......	POSITIVE	......	NEGATIVE	......	......	......	ECON	......
account		-		-				econ	
accurate		positive		-				-	
accursed		-		negative				-	
......									

The Harvard IV-4 dictionary covers a broad range of topics beyond positive and negative word lists. In the course of the study, we additionally make use of word lists on the topic 'economy' (column 'Econ' in the spreadsheet example) when inferring topic contexts for user subgroups.

Upon noticing that neutral words like 'tax' occur in a negative context in the Harvard IV-4 dictionary, Loughran and McDonald (2011) put forward alternative lists of negative and positive words adapted to financial contexts. To keep the number of words as large as possible while still accounting for the financial context we therefore integrate the Loughran and McDonald (2011) word lists and the respective word polarities into the Harvard IV-4 dictionary.

The tweet data is processed by means of the Python programming language. In specific, we employ the Python Natural Language Toolkit (http://www.nltk.org) to remove punctuation and stop words like 'the', for instance. Hashtags, i.e. links to trending topics on Twitter, are treated as common words after stripping the '#'-sign. All characters are changed to lower cases. The resulting so-called normalized words are denoted *terms* in the natural language processing literature. Moreover, to reduce the fraction of noise induced by informal language on Twitter we discard all words not contained in the full modified Harvard IV-4 dictionary (7315 words). We refrain from a stemming procedure, i.e. reducing derived or inflected words to their stem. Stemming was prone to error in our setup.

Using the Porter (1980) stemming algorithm, for instance, we find 3% of the positive word list after stemming to be identical to stemmed words from the negative word list.

2.3 Financial Time Series

The aggregate stock market index representing important worldwide developments is taken to be the S&P 500 price index. Log returns r_d are constructed from prices p_d at the end of trading day d as $r_d = \log(p_d) - \log(p_{d-1})$.

To obtain a simple, yet robust measure of price volatility we follow Parkinson (1980) and compute the scaled high-low range

$$r_d^{vol} = \frac{1}{4\log(2)}(\log(p_d^h) - \log(p_d^l))^2, \tag{1}$$

where p_d^h is the highest observed price on d and p_d^l the lowest price. In contrast to the squared return measure based on open and close prices, the scaled range is not as heavily contaminated by so-called market microstructure noise and thus represents a better estimator of latent financial market volatility (see Alizadeh et al. (2002)).

3 Natural Language Processing Methods

3.1 The general Signal Extraction and Denoising Strategy

Based on a numerical representation of textual data we employ statistical methods to identify subgroups of users with high proximity to financial topics. The small set of preselected known 'experts' (see section 2.1) augmented by a non-expert set allows to utilize supervised learning methods for the classification of all remaining Twitter users into an expert group, henceforth G_e, and non-experts, G_n. To avoid dependence on the preselected sample we further employ unsupervised learning methods and identify twitter user groups as clusters in the tweet data. Subsequently we characterize the clusters in terms of their proximity to the economy word list contained in the Harvard-IV dictionary (see section 2). Intuitively, both statistical approaches rely on the similarity of the vocabulary used by tweet authors to expert word lists from the finance context.

Counting negative and positive word mentions we next construct straightforward metrics of author sentiment and disagreement. To denoise the tweet language, the metrics are explicitly computed for expert subsets of Twitter users as identified by the statistical methods. Ultimately, the resulting time series of expert sentiment and disagreement measures are linked to stock market return and volatility series.

3.2 The Vector Space Document Representation

We view a tweet, denoted by T, as a *concatenation* of ordered character strings. The strings originate from a universal word list $\mathcal{L}$ containing n distinct terms. Let $\{t_1, t_2, ..., t_{N(T)}\} \subset \mathcal{L}$ represent an ordered set of terms occurring in a tweet T with $N(T)$ terms. We define tweets formally as the concatenation of the individual terms,

$$T := \mathrm{c}(\ \{t_1, t_2, ..., t_{N(T)}\}\ ) = t_1 t_2 ... t_{N(T)}, \tag{2}$$

where $\mathrm{c}(\cdot)$ represents a concatenation operator for an ordered set of strings. As such, tweets are formally given as tuples $t_1 t_2 ... t_{N(T)}$, ordered from left to right and written without the punctuation. Let us assume that we can identify individual terms in the concatenated tweets. Hence, we write $t \in T$ if for term t it holds that $t \in \{t_1, t_2, ..., t_{N(T)}\}$.

In the following, we operate on two sets of documents comprised of tweet-based text. First, let $T^u_{d\tau}$ denote the tweet for user u from a user universe $\mathcal{U}$, at a time τ from a grid of time stamps Δ_d on a specific day $d = 1, .., \mathcal{D}$. The original full set of single tweets retrieved through Twitter is given by

$$\mathcal{T} = \{\ T^u_{d\tau} \mid u \in \mathcal{U},\ \tau \in \textstyle\bigcup_d \Delta_d,\ d \in \mathbb{N}_{\mathcal{D}}\ \}. \tag{3}$$

Second, in order to identify subgroups of users with common characteristics we aggregate the individual history of tweet text. For each user u sharing a message on a day d we append all text data shared by u up to and on day d to the latest tweet and define the user-concatenated tweets

$$T^{u,con}_d := \mathrm{c}(\ \{\ T^u_{x,\tau} \in \mathcal{T} \mid 0 \leq x \leq d\ \}\ ) = T^u_{1\tau_1} T^u_{1\tau_2} ... T^u_{d\tau_1} T^u_{d\tau_2} ... \tag{4}$$

Aggregated tweets $T_d^{u,con}$ are typically multiple times bigger than their short 140 character counterparts $T_{d,\tau}^u$. The main goal of this user aggregation is to enlarge the vocabulary fingerprint for each user such that classification into experts and non-experts becomes easier. Note that we are still true to the information set available on day d, avoiding forward-looking bias in (4). The full sample of aggregated tweets is now

$$\mathcal{T}^{con} = \{ \; T_d^{u,con} \mid u \in \mathcal{U}, \; d \in \mathbb{N}_{\mathcal{D}} \; \}, \tag{5}$$

where individual tweet times τ on a day are no longer relevant. Let $T_d := \{T_x^{u,con} \in \mathcal{T}^{con} | x = d\}$ denote the tweet collections for day d. In addition, set N_d as the number of tweets on d. Unless explicitly noted otherwise, the statistical methods for textual data are based on the aggregation $\mathcal{T}^{con}$ throughout the study.

Many approaches in natural language processing use word frequency representations as outlined in Manning et al. (2008), Salton and Buckley (1988) as well as Aizawa (2000). A crucial concept in this context is the term-frequency-inverse-document-frequency (tf-idf). The tf-idf is based on the term frequency tf which gives the number of occurrences of a term in a tweet,

$$\mathrm{tf}(t, T) = |\{i \in \mathbb{N}_{N(T)} | \; t_i \in T \text{ and } t_i = t\}|, \tag{6}$$

where $|\cdot|$ denotes the cardinality. The term frequency is typically weighted by the inverse document frequency idf,

$$\mathrm{idf}(t, d) = \log\left(\frac{N_d}{|\{u \in \mathcal{U} | \; t \in T_d^{u,con}\}|}\right), \tag{7}$$

where the denominator of the fraction is the number of user-aggregated tweets on d containing term t. The idf attains its lowest value 0 in case of a term occurring in all tweets on a day, i.e. in case $|\{u \in \mathcal{U} | \; t \in T_d^{u,con}\}| = N_d$. In summary, the idf scales down word frequencies of words that are contained in many tweets and thus likely have low discriminatory power for, e.g., the classification of documents. Combining the term frequency (6) and weights (7), the tf-idf representation for terms is given as

$$\mathrm{tf\_idf}\;(t, T, d) = \mathrm{tf}(t, T) \cdot \mathrm{idf}(t, d), \qquad T \in \mathcal{T}^{con}. \tag{8}$$

Tabulating tf-idf-weighted occurrences of terms contained in the n words of list $\mathcal{L}$ for each tweet on a day d we obtain the daily *term document matrix* (tdm) M,

$$M = \begin{pmatrix} M'_{1\cdot} \\ \vdots \\ M'_{N_d\cdot} \end{pmatrix} = (M_{i_T=1,\dots,N_d,\ k_t=1,\dots,n}) = \text{tf\_idf}\,(t,T,d), \quad t \in \mathcal{L}, \quad T \in T_d, \tag{9}$$

where the row index i_T denotes the tweet position on day d, $i_T \in \mathbb{N}_{N_d}$. The column index k_t stands for the position of term t in the word list $\mathcal{L}$, $k_t \in \mathbb{N}_n$. Identifying row vectors $M_{i_T\cdot}$ of the tdm with vectors in $\mathbb{R}^n$, the term document matrix gives rise to the notion of a vector space for documents. Each row of M represents a tweet with its corresponding tf-idf-values forming the vector in $\mathbb{R}^n$. To minimize effects of different tweets lengths, all rows of M are normalized. Adding row and column names to the tdm, an illustrative example of M looks as follows.

	ACCOUNT	ACCURATE	ACCURSED	.....
Tweet of user 1235, $T_d^{1235,con}$	0	0.5	0	.....
Tweet of user 2257, $T_d^{2257,con}$	0.12	0	0	.....
.....	.....	.....	.....	.....

Due to the 140-character restriction of tweets, the resulting matrices M are typically extremely sparse which allows for parsimonious memory usage. A major advantage of the tdm representation of documents is that standard statistical tools can be applied.

In order to group documents by their underlying topic, Zhao and Karypis (2004), Sebastiani (2002) and Boyack et al. (2011) apply and review similarity approaches for document spaces. Specifically, the vector space for documents suggests to consider tweets to be more similar the smaller the angle between corresponding vectors. The similarity between tweets $T, T^* \in \mathcal{T}^{con}$ as represented by the rows of M can be computed via the cosine similarity

$$\text{sim}(T,T^*) = \frac{||M_{i_T\cdot} \ \cdot \ M_{i_{T^*}\cdot}||^2}{||M_{i_T\cdot}|| \cdot ||M_{i_{T^*}\cdot}||} \tag{10}$$

with $||M_{i_T\cdot}|| = \sqrt{M^2_{i_T,1} + M^2_{i_T,2} + \dots + M^2_{i_T,n}}$. Another intuitive similarity metric for normalized vectors in $\mathbb{R}^n$ is given by the Euclidean distance. Note further that upon including a tweet $\mathcal{L}^*$ consisting of a concatenated topic word list as row in M we can straightforwardly compute the similarities between tweets and a particular topic as $\text{sim}(T, \mathcal{L}^*)$.

Implementations of vector space transformations as well as machine learning algorithms applied to the vectorized text data in the present study are taken from the Python scikit-learn library put forward by Pedregosa et al. (2011).

3.3 Unsupervised Learning Methods for Language Modeling

Unsupervised machine learning methods like clustering can be used to detect hidden structure in textual data. An important aspect of unsupervised learning is that it does not require labeled *responses* (*dependent* variable observations) to learn the latent structure. In specific, unsupervised learning methods for text documents detect structure solely based on the tdm M. Clustering methods applied to the document vector space spanned by the rows of M are thought of as grouping the data into clusters of similar topics.

Machine learning methods derive results from *features*, i.e. measurable properties of the data under consideration. In the context of models for documents, the features are given by the tf-idf values for the terms used in the documents. However, the large dimension of the row space of M impairs the unsupervised learning quality considerably. This is the 'curse of dimensionality' problem encountered in numerous statistical procedures. Consequently, dimension reduction or feature selection methods in the vector space of documents are typically applied prior to the learning procedure to improve the performance. In the following we first describe the latent semantic analysis (henceforth LSA) as an unsupervised learning method for dimensionality reduction before we turn to the K-Means clustering algorithm.

3.3.1 Dimension Reduction via Latent Semantic Analysis

Introduced by Deerwester et al. (1990), the LSA or latent semantic indexing is based on a singular value decomposition (SVD) of the transposed tdm, M'. In contrast to a standard principal component transformation of the data, however, the LSA explicitly does *not* operate on a data matrix with column-wise mean zero. The reason is to ultimately maintain the sparsity of the tdm and hence the good storage properties. The purely data-driven extraction of latent features from documents make the LSA especially attractive for dimension reduction in unsupervised clustering applications as outlined in Dong et al. (2006) and Schütze and Silverstein (1997).

The SVD of the transposed tdm, M', is given by

$$M' = S\Lambda^{\frac{1}{2}}R', \tag{11}$$

where $SS' = I$ and $R'R = I$. $\Lambda^{1/2}$ contains square roots of eigenvalues from S in descending order. The columns of S are orthonormal eigenvectors of $M'M$, while columns of R are orthonormal eigenvectors of MM'. In the end we are interested in a mapping of M' to the eigenspace of MM' such that much of the characteristic structure in M is preserved. This can be achieved by first rearranging (11) to get $R = (\Lambda^{-1/2}S'M')'$.

To finally lower the dimensionality of the transformed tdm we compute a *reduced* singular value decomposition to approximate the transformation R. In the first step, we retain only the top k eigenvalues of Λ and obtain $\Lambda_k^{1/2}$. Second, we eliminate corresponding row vectors of S to get S_k and consequently compute the approximation

$$R_k = (\Lambda_k^{-1/2}S_k'M')'. \tag{12}$$

One can show that the reduced SVD $M_k := S_k\Lambda_k^{1/2}R_k'$ is the best rank-k approximation to M in terms of the Frobenius norm (see Manning et al. (2008)).

The tweet T, identified with the tdm row $M_{i_T,\cdot}$ is thus mapped to the i_T-th row of R_k via the projection

$$M_{i_T,\cdot} \to R'_{i_T,\cdot} = (\Lambda_k^{-1/2}S_k'M_{i_T\cdot})', \tag{13}$$

with dimension $(1, k)$ instead of $(1, n)$ and $k \ll n$. Intuitively, the row R'_{i_T} consists of linear combinations of terms (represented by tf-idf values) considered to have similar meaning. In consequence, the LSA is thought to reduce the variance of the text data induced by polysemy, multiple words with similar meaning, and synonymy, one word having multiple meanings.

3.3.2 K-Means Clustering

The K-Means algorithm of MacQueen (1967) is a heuristic method to partition data into K clusters such that the data are maximally close to the cluster centers according to a distance measure. In order to find clusters of users with similar shared Twitter content we apply the K-Means algorithm to the tweet vector space. However, to mitigate the curse of dimensionality due to a high-dimensional M we conduct a LSA transformation prior to the K-Means and set $M := R_k$ henceforth.

Clusters for normalized rows of M are based on the squared Euclidean distance measure $||M_{i\cdot} - M_{l\cdot}||^2 = \sum_{j=1}^{n}(M_{ij} - M_{lj})^2$. We assign the i-th tweet vector to its closest cluster with mean m_k via $i \to C(i) = \arg\min_{1 \leq k \leq K} ||M_{i\cdot} - m_k||^2$. A measure of fit for the cluster assignment function C is the total cluster variance according to

$$V(C, m_1, .., m_k) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||M_{i\cdot} - m_k||^2, \tag{14}$$

where $N_k = \sum_{i=1}^{n} \mathbb{1}_{\{C(i)=k\}}$. Hastie et al. (2009) describe a K-Means scheme consisting of the following steps given an initial cluster assignment

1. Minimize (14) with respect to $m_1, .., m_K$ for a given cluster assignment C.
2. Given $m_1, .., m_K$, minimize (14) with respect to the cluster assignment C.
3. Repeat first and second step until the assignments do not change.

Although this scheme poses a hard computational problem, effective algorithm implementations exist in the scikit-learn Python library. The library allows to randomize initial cluster assignments to avoid initial value dependence.

Furthermore, to determine the number of clusters K, our analysis uses the statistical procedure outlined in Tibshirani et al. (2001).

3.4 Supervised Learning Methods for Document Classification

Let a so-called training set of features based on tweets from $\mathcal{T}^{con}$ for a given day and corresponding *known* expert and non-expert group labels be given as tuples $(M_{1\cdot}, G_1)$,

$(M_{2\cdot}, G_2)$,.., $(M_{k\cdot}, G_k) \in \mathbb{R}^n \times \{G_e, G_n\}$. In our context, supervised learning methods are devised to utilize the labeled training data to learn (estimate) the unknown true function $f\colon\ M_{i\cdot} \to \{G_e, G_n\}$ assigning group affiliations to individual tweets based on the features $M_{i\cdot}$. The resulting estimate $\widehat{f}$ of the classification function is used to ultimately predict group affiliations of the remaining unlabeled tweet data.

One of the most prominent supervised learning methods included in our study is the Naive Bayes model, popularized, e.g., by Maron and Kuhns (1960) and Lewis (1998) for text mining applications. More complex supervised classification methods are given by the linear and nonlinear Support Vector Machines (SVMs). Joachims (1998) and Burges (1998) document the good performance of SVMs in text classification exercises.

In contrast to unsupervised learning approaches, dimension reduction methods in the supervised case exploit the additional label information to select a feature subset carrying useful information for the classification of unlabeled data. An example of these methods is the χ^2-statistic.

3.4.1 Dimension Reduction with the χ^2-Statistic

To select terms t with high discriminatory power between two groups, Schütze et al. (1995) propose to use a χ^2-test for independence of the events $\{t \in T\}$ and $\{T \in G\} :=$'T issued from a user of group G'. The classical χ^2-test in the text data context is applied to a contingency table formed from the events $\{t \in T\}$, $\{t \notin T\}$ and $\{T \in G\}$, $\{T \notin G\}$. It is based on the squared difference between observed joint term and group occurrences and expected occurrences under the null hypothesis of independence of term and group. High deviations resulting in large values of the χ^2-statistic indicate a dependency of term and group occurrence and thus a high discriminatory power of the term for group classification.

Instead of directly applying the statistical test, however, the χ^2 test statistic is simply used to rank the features and keep the top k features. Manning et al. (2008) note that this is due to the multiple testing problem arising for tests applied to a large set of terms which renders p-values for the test decision invalid.

3.4.2 The Naive Bayes Classifier

Our analysis employs the Bernoulli Naive Bayes model, which has attractive empirical properties in case of smaller documents like tweets (McCallum and Nigam (1998)). Utilizing the Bayes theorem, the probabilistic Naive Bayes learning method maximizes the probability of a tweet being from an expert or non-expert group according to

$$\begin{aligned} G = \underset{G \in \{G_e, G_n\}}{\arg\max}\ \mathbb{P}(T \in G \mid T) &= \underset{G \in \{G_e, G_n\}}{\arg\max}\ \frac{\mathbb{P}(T \mid T \in G)\ \mathbb{P}(T \in G)}{\mathbb{P}(T)} \\ &= \underset{G \in \{G_e, G_n\}}{\arg\max}\ \mathbb{P}(T \mid T \in G)\ \mathbb{P}(T \in G). \end{aligned} \tag{15}$$

We model the conditional probability $\mathbb{P}(T \mid T \in G)$ in (15) as

$$\mathbb{P}(T \mid T \in G) = \mathbb{P}\left(\bigcap_{t \in \mathcal{L}} \{U_t = \mathbb{1}_{\{T\}}(t) | T \in G\} \right) \overset{(*)}{=} \prod_{t \in \mathcal{L}} \mathbb{P}(U_t = \mathbb{1}_{\{T\}}(t) | T \in G), \tag{16}$$

where the occurrences of terms t in T conditional on T being from group G are modelled as independent Bernoulli random variables U_t. Specifically, we are interested in the probability of U_t assuming values $\mathbb{1}_{\{T\}}(t)$, i.e. 1 if $t \in T$ and zero else given that $T \in G$. The independence assumption $(*)$ for the U_t reduces the number of parameters to be estimated substantially and directly results in the right-hand side of (16).

Daily estimates for the conditional probabilities are readily available based on

$$\widehat{\mathbb{P}}(U_t = 1 |\ T \in G) = \frac{\sum_{T \in G} \mathbb{1}_{\{T\}}(t)}{N_d} \tag{17}$$

for tweets from the daily set T_d as the estimated probability that t will occur in a tweet of group G. Equation (17) is used to construct $\widehat{\mathbb{P}}(T | T \in T)$ via (16). Further,

$$\widehat{\mathbb{P}}(G) = \frac{|\{u \in G \mid \exists T_d^{u,con}\ \}|}{N_d}$$

is the daily estimate for $\mathbb{P}(G)$. Consequently, the decision function value yielding the estimated group for T, represented by the tdm row $M'_{i_T\cdot}$, is given by

$$\widehat{f}(M_{i_T\cdot}) = \arg\max_{G} \widehat{\mathbb{P}}(G)\widehat{\mathbb{P}}(T|T \in G).$$

3.4.3 Support Vector Machines

First introduced in the 1970s by Vapnik (1979) in the context of statistical learning, support vector machines gained popularity in the 90s as classifiers for a wide range of applications. The fundamental idea behind SVMs is to find a hyperplane in the data vector space such that the data are separated according to the class labels. The hyperplane derived from labeled training data represents a linear decision function to be used for the classification of unlabeled data. In an important extension of the basic setup, transformations of the data can be shown to produce nonlinear decision functions for the original data. Vapnik (1998), Schölkopf and Smola (2001) and Cristianini and Shawe-Taylor (2000) amongst others give overviews of the topic.

We operate on the document space in $\mathbb{R}^n$ spanned by the N_d rows of M. Let $y_i \in \{-1, 1\}$ represent the expert or non-expert class $\{G_n, G_e\}$ of the tweet given by row $M_{i\cdot}$. Furthermore, let $\phi : \mathbb{R}^n \to \mathbb{H}$ define a mapping of the data to the (possibly higher-dimensional) space $\mathbb{H}$. The data transformation via ϕ allows to identify a linear separating hyperplane in $\mathbb{H}$ which in turn can form a *nonlinear* decision boundary in $\mathbb{R}^n$. The classification itself is based on the set $\{x \in \mathbb{R}^n \mid g(x) = \omega'\phi(x) + b = 0\}$ representing a hyperplane in $\mathbb{H}$. Individual data points (tweets) x are classified into $\{-1, 1\}$ according to the decision function $f(\cdot) = \text{sign}(g(\cdot))$, which gives the location of transformed features $\phi(x)$ with respect to the hyperplane. In an ideal scenario, the hyperplane perfectly separates the classes, while in general, classes are allowed to overlap. The weight vector $\omega \in \mathbb{H}$ and the bias $b \in \mathbb{R}$ are given as solutions to an optimization problem where the margin between the separating hyperplane and the two classes is maximized. The reason for the margin maximization is to ultimately reduce the classification error on unseen data after training the SVM on a labeled training sample.

The SVM optimization problem can be posed as

$$\min_{\omega,b,\xi} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N_d} \xi_i, \tag{18}$$

$$s.t. \quad y_i \cdot (\omega'\phi(M_{i\cdot}) + b) \geq 1 - \xi_i, \tag{19}$$

$$\xi_i \geq 0, \quad i = 1, ..., N_d, \tag{20}$$

where C is a cost parameter that controls the tradeoff between the complexity of the decision function and the fraction of tweets misclassified. The ξ_i are so-called slack variables which allow data points to scatter freely on either side of the hyperplane such that classes do not have to be perfectly separable.

The optimization problem (18) can further be transformed to an equivalent dual problem, given as

$$\begin{aligned} \max_{\alpha} L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \cdot \underbrace{\phi(M_{i\cdot})'\phi(M_{j\cdot})}_{=K(M_{i\cdot},M_{j\cdot})} \\ s.t. \quad 0 \leq \alpha_i \leq C, \quad i = 1, ..., N_d, \\ \sum_{i=1}^{N_d} \alpha_i y_i = 0. \end{aligned} \tag{21}$$

The dual problem formulation allows to rewrite the inner product $\phi(M_{i\cdot})'\phi(M_{j\cdot})$ in terms of a kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,

$$\phi(M_{i\cdot})'\phi(M_{j\cdot}) = K(M_{i\cdot}, M_{j\cdot}). \tag{22}$$

The *kernel trick* (22) does not hold for arbitrary ϕ and K, thus effectively imposing constraints on the choice of ϕ. However, note that in this case the mapping ϕ does not have to be specified explicitly for (21). The terms involving ϕ can be replaced by a kernel function, which is a huge advantage. Kernel functions are important tools in nonparametric statistics used to capture complex nonlinearities. A flexible transformation of the data can be obtained, for instance, with the Gaussian radial basis function (RBF) kernel,

$$K(M_{i\cdot}, M_{j\cdot}) = \exp(-\gamma ||M_{i\cdot} - M_{j\cdot}||^2), \quad \gamma \in \mathbb{R}, \tag{23}$$

where γ is a hyperparameter defining its shape.

Problem (21) represents a convex optimization problem which can be solved by standard quadratic programming methods. First order conditions imply $\omega = \sum_{i=1}^{N_d} y_i \alpha_i \phi(M_{i\cdot})$ for the optimum such that the decision function f can be rewritten in terms of the kernel function. After numerical optimization, the estimated group of a tweet x, represented by a row in M, according to the SVM is given by

$$\widehat{f}(x) = \text{sign}(\widehat{\omega}'\phi(x) + \widehat{b}) = \text{sign}\left(\sum_{i=1}^{N_d} \widehat{\alpha}_i y_i K(M_{i\cdot}, x) + \widehat{b}\right), \tag{24}$$

where the hats denote optimal parameters. Details on convex optimization are given in, e.g., Boyd and Vandenberghe (2004).

Our analysis uses two specifications of the SVM. First, we train the nonlinear SVM based on the RBF kernel. Second, we train a linear SVM which is subsumed in the above outline when imposing $\phi(x) = x$, i.e. no kernel replacement in the dual problem (21) is needed. In the first case we need to calibrate both the cost parameter C governing the variance-bias tradeoff and the shape parameter γ. The second specification requires the calibration of the cost parameter C.

3.4.4 Model Evaluation and Calibration of Classifiers

We report several metrics to assess the quality of the classification. Let $G \in \{-1, 1\}$ again represent the groups $\{G_n, G_e\}$ to be discriminated.

The accuracy is given as $\frac{TP+TN}{P+N}$, where TP represents the number of true positives, i.e. correct predictions of values 1. Correspondingly, TN is the number of true negatives. P and N are the total number of positive and negative sample elements, respectively. However, the accuracy has drawbacks as a model assessment metric as it obscures the classification performance for data with an unbalanced fraction of positives and negatives.

The within class accuracies are captured by the specificity, $\frac{TN}{N}$, and the sensitivity,

$\frac{TP}{P}$. Further refining the classification assessment of the positive class by including misclassifications leads to the precision metric, $\frac{TP}{TP+FP}$, where FP denotes false positives. The so-called $F1$ metric is a popular metric in the machine learning community. It combines the precision and sensitivity via harmonic weighting, which yields

$$F1 = \frac{2P}{2TP + FP + FN}, \tag{25}$$

where FN are the false negatives. Model selection based on the $F1$ metric mitigates the problems associated with the accuracy measure and individual class metrics like sensitivity, specificity and precision.

To calibrate the parameters of the classifiers we split the pre-labeled fraction of the tweet data (see section 2) into training and test samples on a daily frequency, with each sample containing both expert and non-expert labels for tweets. Model training is carried out on the training set exclusively. The pre-labeled test sample, unseen throughout the calibration, is used to report the performance metrics outlined above. Separating model training and assessment in this way helps to avoid over-fitting the data.

The parameters are chosen from a discrete grid of possible values such that a criterion is maximized. A key concept in training a classifier is the k-fold cross-validation (CV). The k-fold CV splits the training set into k subsets in the first step. In a second step, each of the k subsets is used to evaluate the model performance while the remaining k-1 subsets are used to fit the model for a given parameter from the grid. The criterion to be maximized is the sum of the individual criterion values for the k CV iterations. As optimal parameters we select the parameters that maximize the $F1$-measure for the 10-fold CV of the respective model on the training set.

3.5 Measuring Author Sentiment and Disagreement

The construction of signals from tweets is based on the set of individual tweets $\mathcal{T}$. Consequently, in contrast to the user-aggregated tweet set $\mathcal{T}^{con}$ employed in the classification and clustering of users, tweets $T^{u}_{d\tau}$ on day d contain text data from day d only and occur at times $\tau \in \Delta_d$.

We denote the number of terms in a tweet $T^u_{d\tau}$ stemming from an arbitrary word list $\mathcal{L}^*$ intuitively as

$$\#\mathcal{L}^*(d,u,\tau) := |\{\ i \in \mathbb{N} \mid t_i \in T^u_{d\tau} \text{ and } t_i \in \mathcal{L}^*\}|.$$

Let now $\mathcal{L}^{pos} \subset \mathcal{L}$ and $\mathcal{L}^{neg} \subset \mathcal{L}$ denote positive and negative word subsets of the universal word list $\mathcal{L}$, respectively. We compute a daily directional sentiment measure as the daily *negative* fraction of negative words of the universal word list according to

$$S_d = (-1) \cdot \frac{\sum_{\tau\in\Delta_d} \#\mathcal{L}^{neg}(d,u,\tau)}{\sum_{\tau\in\Delta_d} \#\mathcal{L}(d,u,\tau)}, \qquad u \in G, \tag{26}$$

where G is a user group like, e.g., expert and non-expert user groups $\{G_e, G_n\}$ or clusters identified in the data. Dating back to Tetlock et al. (2008), negative fractions are considered more precise in financial contexts, as positive words are often negated. Note that the fraction of negative words is multiplied by -1. By construction, we hence expect high values of (26), i.e. a low fraction of negative terms, to reflect positive developments on financial markets, resulting in an expected positive correlation between returns and twitter sentiments S_d.

In addition to capturing the polarity of tweets we also quantify the disagreement or disaccord among users. Let the $(N_d \times 1)$ vector of daily differences of positive and negative word counts in individual tweets at times $\tau_1, ..., \tau_{N_d}$ be defined as

$$\vec{D}_d \;=\; (\ \#\mathcal{L}^{pos}(d,u,\tau) - \#\mathcal{L}^{neg}(d,u,\tau)\ )_{\tau_1,\ldots,\tau_{N_d}}, \qquad u \in G. \tag{27}$$

As a non-directional measure of user sentiment disagreement we propose to use the standard deviation of (27),

$$D_d = \mathrm{sd}(\vec{D}_d). \tag{28}$$

Price volatility on financial markets is typically caused by dissemination of new information. As a measure of tweet topic volatility, (28) captures information asymmetry on twitter and is thus expected to be positively correlated to financial market volatility. Whenever needed, we further add a superscript to the measures D_d and S_d indicating the specific twitter user group we operate on.

4 Results

Our analysis consists of two major stages. First, we identify daily 'expert' and 'non-expert' user subsets via supervised learning. Likewise, unsupervised learning methods are used to detect and characterize meaningful clusters of users in M. Throughout the first stage we operate on the tweet aggregation $\mathcal{T}^{con}$ and construct tdms M from the daily subsets T^d.

Second, we use the original tweet set $\mathcal{T}$ to construct author sentiment and disagreement metrics for the user subgroups and clusters derived in the first stage. We examine the contemporaneous correlations between return data r_d and sentiment signals as well as volatility measures r_d^{vol} and topic disagreement signals. In this context we assess the properties of the statistical learning approaches. Finally, we model both r_d and r_d^{vol} as ARMA processes. To obtain robust results about the link between S&P market developments and signals based on expert user groups we include lags of sentiment and disagreement signals as regressors in the econometric models.

4.1 The Contemporaneous Correlation between Sentiment Measures and the Financial Market

4.1.1 Unsupervised Learning of Experts

We use the K-Means algorithm to uncover hidden structure in the tweets by clustering $\mathcal{T}^{con}$. As the K-Means clustering is prone to the curse of dimensionality we first reduce the dimension of the tweet vector space via the latent semantic indexing (3.3.1). Transforming each daily tdm according to the LSA mapping (13) we cut the vector space dimension from 7315, the size of the universal word list, to 100. The exact size of the reduced space was ultimately not important to the results as qualitatively similar results can, for instance, be obtained for dimensions 50, 80, 150 and 300.

In the next step we derive daily clusters in the LSA-transformed aggregated tweet space via the K-Means algorithm 3.3.2. Due to the construction of $\mathcal{T}^{con}$ as tweet aggregation per user, the emerging clusters directly correspond to subsets of twitter users. Beginning on the first day of the sample, the K-parameter of the K-Means is calibrated bi-weekly with the gap statistic put forward by Tibshirani et al. (2001). The numbers of daily clusters

	Av. Similarity to economy lexicon	Av. Number of users	Two sample t-test (av. p-val.)
Top Cluster	0.052	1799	0.000
Mid Clusters	0.028	5697	0.000
Lowest Cluster	0.021	5590	0.000

Table 1: Characteristics of cluster solutions. Top and lowest cluster correspond to the clusters with highest and lowest similarity to the economy word list. Mid Clusters refers to the group of remaining clusters. P-values are given for two sample t-test of the similarity measure equality for adjacent clusters. The two sample t-test value for the mid cluster group tests against the the top group only. All numbers are averages per day.

of the K-Means are time-varying and range from 3 to 9 clusters with an average of 3.5 clusters. Figure 6 in the appendix 6.2.1 shows the corresponding time series evolution of the K.

A typical concept in clustering is the characterization of resulting clusters after the convergence of the algorithm. We characterize the clusters in terms of the dominant topic of the underlying text. In this context, the cosine similarity metric (10) can be used to compute the average similarity of tweets from the individual clusters to word lists with financial focus. Without changing the cluster outcome we re-label the resulting clusters in an order corresponding to their user's tweet similarity to the economy word list of the Harvard dictionary, henceforth denoted $\mathcal{L}^{econ}$ (see section 2). This allows us to sort clusters according to their proximity to economical contexts in descending order and finally to assess sentiment and disagreement measures based on this grouping.

To deal with the time-varying number of clusters we divide the full range of clusters into three groups. We report results for the two groups containing all daily clusters with the numerically highest and lowest cluster labels, respectively. After the re-labeling, these are the clusters with overall highest and lowest similarity to the economy word list per day. In addition, we group together the remaining clusters to form a third group of clusters with moderate proximity to the economy topic. Table 1 gives the means of the day-to-day average of cosine similarities $\text{sim}(T, \mathcal{L}^{econ})$ for tweets from each cluster group as well as average user numbers for the three groups. Two-sample t-tests show that the average similarities to the economy word list are significantly different from the averages for the

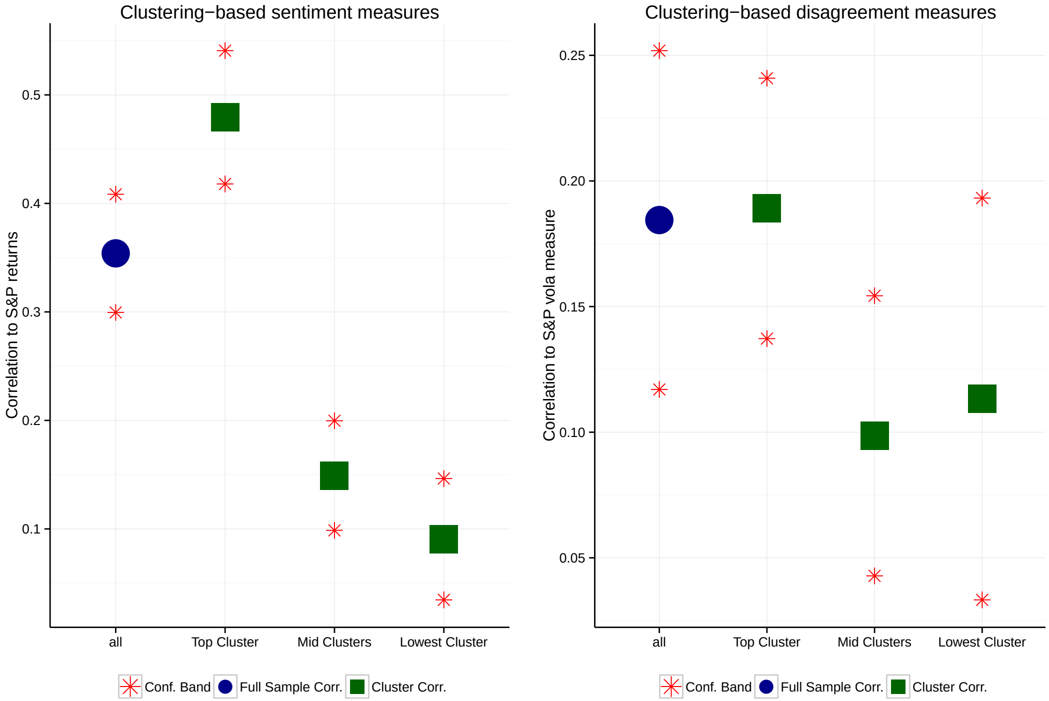

Figure 2: Correlation of unsupervised clustering-based sentiment measures S_d and D_d to daily returns and squared returns, respectively. Left panel: S_d vs. r_d, right panel: D_d vs. r_d^{vol}. Confidence bands are $2\widehat{\sigma}$ intervals, with $\widehat{\sigma}$ obtained via bootstrapping the correlation distribution. Notation as in Table 1. 'all' stands for the signals based on the full tweet sample.

adjacent group (p-values in Table 1).

After collecting tweets from $\mathcal{T}$ for users from within the three cluster groups we construct daily sentiment measures S_d as well as user disagreements D_d for the user groups. Moreover, we also include measures S_d and D_d constructed for the full sample of users. Finally, the corresponding time series are correlated to the series r_d and r_d^{vol}. Figure 2 illustrates the resulting contemporaneous Pearson correlations $\text{Corr}(r_d, S_d)$ and $\text{Corr}(r_d^{vol}, D_d)$.

Evidently, signs of the correlations are in line with expectations. First, the daily negative fraction of negative words among words used in tweets, given by S_d, is positively correlated with daily S&P returns. This shows that the polarity in sentiments reflects return movements. Second, the disagreement of user sentiments D_d is positively associated with broad market volatility. We conclude that the volatility of individual tweet sentiment polarities corresponds to financial market volatility. Third, we further observe that the order of the correlations matches the closeness of the tweets of users to financial topics. Correlations for signals of the top cluster are always higher than signals for remaining user clusters. In case of directional sentiment measures, the resulting correlations for the sample of users from the top cluster are stronger than for measures based on the full sample of tweets (data point 'all' in Figure 2). In specific, we thus find subgroups of users ('expert' clusters) which allow to construct particularly strong sentiment measures. This indicates that a purely data-driven statistical language approach yields user groups clustered together by conversation topic such that informative signals can be separated from noisy signals.

4.1.2 Supervised Learning of Experts

Even for an user subsample of 190 handpicked expert users, the number of users actually issuing tweets per day can be extremely low (cf. Figure 1) such that daily measures S_d and D_d computed for the expert subsample are based on very few terms and thus become overly volatile. To automatically increase the sample of experts we employ supervised statistical classification methods to classify unlabeled user tweets represented by rows of a tdm on a given day into experts and their complement.

However, in order to learn, supervised classification models require both a pre-labeled positive sample (experts) *and* a pre-labeled negative sample (non-experts). To obtain the negative non-expert sample we again rely on the similarity of tweets to the Harvard 'economy' word list. On a daily basis we choose users associated with tweets having a similarity measure smaller than the lowest quintile of its empirical distribution as the non-expert set. Due to the up to 25,000 tweets collected per day, the portion of such chosen non-experts is substantially greater than the expert counterpart even for very low quantiles of the similarity measure. Unfortunately, the classification performance of standard machine learning algorithms is typically diluted by drastically imbalanced data. Drawing on He and Garcia (2009) as well as Haerdle et al. (2009) we thus down-sample the non-expert set by selecting twice as much non-expert users than expert users per day to better balance the data. The reason for keeping more non-expert data lies in the generally larger text of experts. In addition, on each day we always include the full history of aggregated user tweets in the training and test samples to maintain large pre-labeled test and training sets.

Figure 3 shows the histogram of the words occurring in the expert tweets from combined

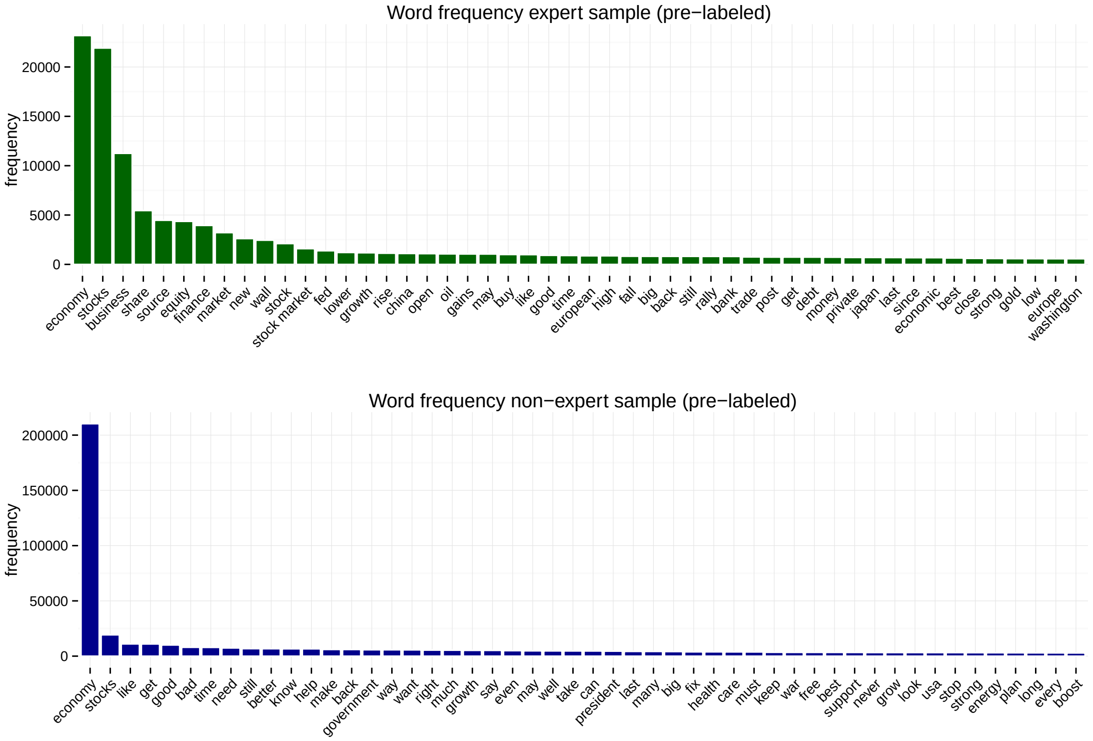

Figure 3: Distribution of top 50 words for expert and non-expert set of the combined test and training sample

training and test data as well as the histogram of words from non-expert tweets for test and training data as given by the similarity approach. The histograms show that the expert set exhibits a word distribution skewed towards terms from the financial context, whereas the non-expert words are more evenly distributed. Except for the large occurrence of the term 'economy', the non-expert tweets contain several terms from everyday life topics.

Moreover, we mitigate the curse of dimensionality problem via the χ^2-feature selection procedure (see 3.4.1). Ranking the terms accordingly we retain the 250 terms with the highest value of the χ^2-statistic. The results for this reduced term document space were found to be representative for alternative selections of the top 100, 150, 300 and 400 terms, respectively.

We train the supervised learning methods outlined in section 3.4 on the training sets on a daily basis. Hyperparameters are calibrated through maximization of the F1-measure (25) using 10-fold cross-validation (see 3.4.4). Details on the time series of hyperparameters are given in the appendix 6.2. The classification performance of the Bernoulli Naive Bayes as well as the linear and nonlinear SVM classifier can be evaluated on the pre-labeled test

	BNB	SVM	SVM (nonlin.)
Specificity	0.992	0.985	0.924
Sensitivity	0.813	0.791	0.821
Accuracy	0.947	0.937	0.898
Precision	0.949	0.938	0.904
F1 measure	0.948	0.939	0.927
% experts	31.7	26.6	33.1

Table 2: Accuracy of classification on the test set of experts (6.1). Abbreviations are BNB: Bernoulli Naive Bayes, SVM: Support Vector Machine, SVM (nonlin.): SVM based on RBF kernel. % experts denotes the fraction of experts as predicted for the remaining unclassified data. All numbers are averages over the five year sample.

sample of experts and non-experts. Table 2 gives the classification metrics averaged across the sample days. After learning the decision rule $f : M_{i\cdot} \rightarrow \{G_e, G_n\}$ on the training set and obtaining a daily estimate $\widehat{f}$ we classify the yet unseen remaining tweet data consisting of tweets not contained in the test and training expert and non-expert samples.

The overall accuracy of the classifiers on the test set exceeds 89% with differences, however, within classes. While non-experts are classified with a high accuracy, i.e. the specificity being above 92%, the sensitivity as well as precision measures reflect a worse performance of the classifiers for the classification of the experts group. Notably, the resulting expert fraction for the completely classified tweet sample on a given day spans 26.6% (SVM) to 33.1% of users (nonlinear SVM) on average.

Finally, we compute measures S_d and D_d for the full daily expert and non-expert user subsets as identified by the three classifiers trained before. The resulting time series are again linked to daily returns r_d and volatility r_d^{vol}. Figure 4 shows the contemporaneous correlations $\text{Corr}(r_d, S_d)$ and $\text{Corr}(r_d^{vol}, D_d)$ for expert and non-expert sets based on the individual classifiers.

Several results emerge from the correlation analysis. First, it is possible to separate distinct finance tweets which can be used to compute strong signals. There are notable differences in the correlation for expert signals compared to the complementary non-expert signals across all models. Moreover, expert-based correlations are always higher than the correlation for the full tweet sample ('all'). Second, there are differences in the correlations

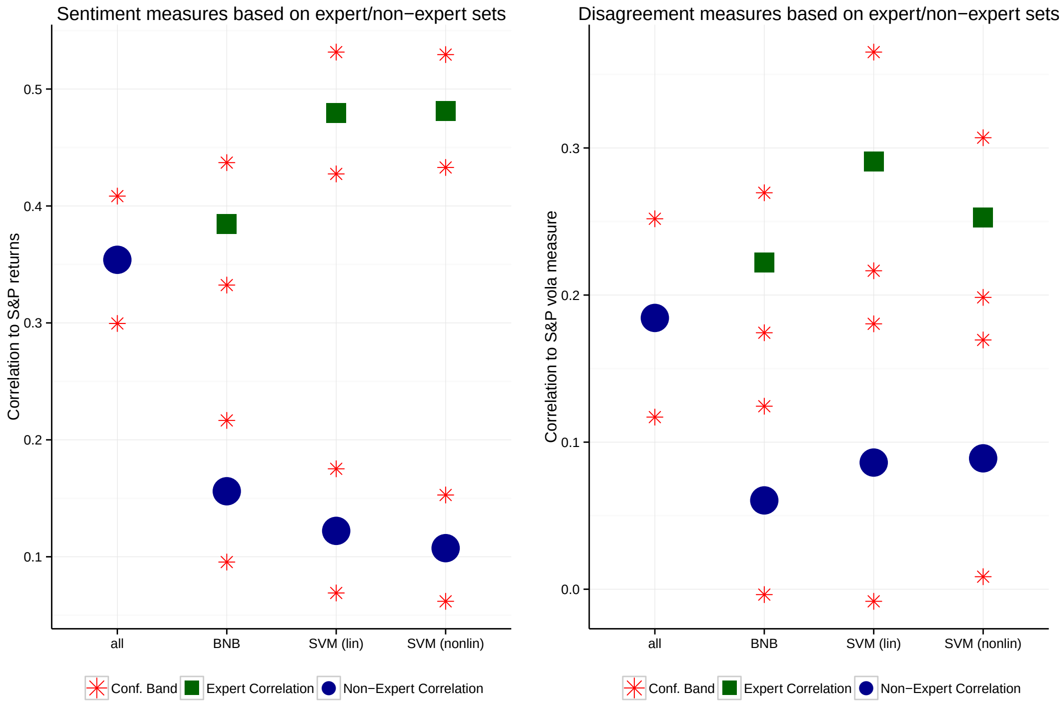

Figure 4: Correlation of supervised learning-based sentiment measures S_d and D_d to daily returns and squared returns, respectively. Left panel: S_d vs. r_d, right panel: D_d vs. r_d^{vol}. Abbreviations as in Table 2. 'all' stands for measures S_d and D_d derived from the full tweet sample. Confidence bands are $2\widehat{\sigma}$ intervals, $\widehat{\sigma}$ obtained via bootstrapping.

for different classifiers, as indicated already by the differing classification performances reported in Table 2. In specific, while the Bernoulli Naive Bayes exhibits the lowest correlations of corresponding measures S_d and D_d to returns and squared returns we observe consistently high correlations for the signals from the linear SVM. Third, the signs of the correlations are in line with expectations, confirming results for the signals derived via unsupervised learning (section 4.1.1). Fourth, the additional information provided by the pre-classified training set of experts and non-experts lead to stronger disagreement signals than for the unsupervised classification exercise (cf. section 4.1.1).

4.2 A Time Series Analysis of the Relation between Twitter Expert Signals and the Financial Market

For the time series analysis we concentrate on the expert signals as identified by the classification methods as well as on signals based on the clusters with the highest correlations

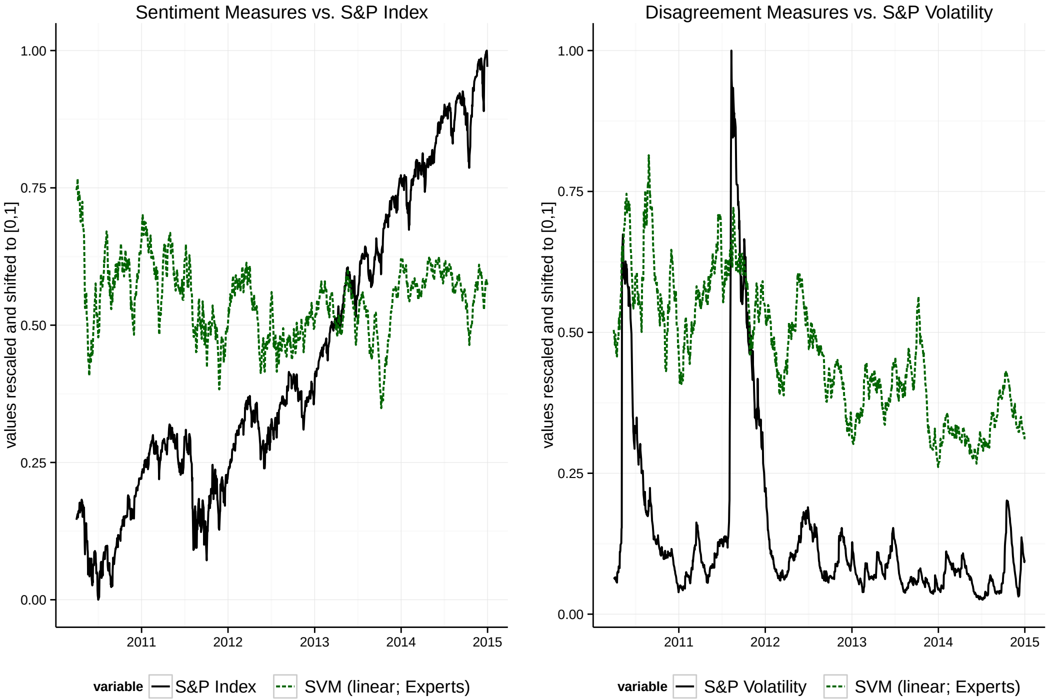

Figure 5: Example evolution of smoothed expert sentiment and disagreement measures (linear SVM classification) vs. S&P 500 index and volatility r_d^{vol}. 'all' stands for the full sample signals. Sentiment and disagreement measures for day d are averages based on the previous 31 days.

to the economy word list. Hence, for sake of brevity we neither include non-expert-based signals nor signals derived from clusters with low average correlation to the economy-word list. Still, to compare expert signals to less precise signals we report results for the signals for the full sample of tweets ('all').[1]

Figure 5 exemplifies the time series S_d^{SVM} and D_d^{SVM} for experts identified via the linear SVM classification. In addition we show the full sample signals next to the time series p_d (S&P index) and r_d^{vol}. The contemporaneous linkage of text-based signals (see subsections 4.1.1 and 4.1.2) and the broad market is reflected in the time series evolution. Spikes in the S&P index and S&P volatility tend to line up with corresponding twitter signals. In contrast to the time of these events, the magnitude of spikes in sentiment and disagreement metrics often does not correspond to the financial market variables, indicating topic exaggeration on social media platforms.

The descriptive statistics in Table 3 show that all time series under consideration in the present study are covariance stationary as evidenced by the augmented Dickey Fuller

[1] To give the complete picture, main results for non-experts are given in the appendix 6.3.

	mean	max	min	sd	adf (P-val.)	ρ_1	ρ_2	ρ_3
r_d	0.00	0.05	-0.07	0.01	-10.92 (0.01)	-0.06	0.06	-0.07
r_d^{vol}	0.69	30.01	0.01	1.41	-6.42 (0.01)	0.43	0.43	0.32
S_d^{all}	-0.43	-0.27	-0.55	0.03	-5.72 (0.01)	0.65	0.50	0.44
S_d^{BNB}	-0.46	-0.25	-0.61	0.05	-5.73 (0.01)	0.62	0.47	0.42
S_d^{SVM}	-0.45	-0.33	-0.60	0.04	-7.83 (0.01)	0.40	0.16	0.11
S_d^{SVMnl}	-0.44	-0.28	-0.57	0.04	-8.32 (0.01)	0.41	0.17	0.12
S_d^{topCl}	-0.47	-0.31	-0.69	0.06	-8.33 (0.01)	0.40	0.17	0.12
D_d^{all}	1.44	1.93	1.32	0.08	-4.00 (0.01)	0.83	0.78	0.77
D_d^{BNB}	1.45	2.03	1.27	0.12	-4.12 (0.01)	0.83	0.79	0.78
D_d^{SVM}	1.42	1.92	1.24	0.08	-7.42 (0.01)	0.52	0.47	0.42
D_d^{SVMnl}	1.41	1.90	1.24	0.08	-6.92 (0.01)	0.52	0.47	0.42
D_d^{topCl}	1.38	1.71	1.19	0.08	-7.66 (0.01)	0.38	0.24	0.24

Table 3: Descriptive statistics of expert sentiment and dispersion measures. Abbreviations are *sd* for the standard deviation and *adf* for the augmented Dickey Fuller test and ρ_i denotes the sample autocorrelation of order i. Furthermore, measures with the $SVMnl$-superscript are based on the nonlinear SVM algorithm. *topCl* denotes the K-Means clusters with the highest correlations to the economy lexicon.

test statistic for the null hypothesis of a unit root in the data. The same result can be inferred from alternative unit root tests like the KPSS or Phillips-Perron test. Nevertheless, the realized volatility measure r_d^{vol} exhibits the typical pronounced autocorrelation pattern (first three sample autocorrelations given).

Table 4 shows Pearson correlations of signals S_d and D_d to leads and lags of returns and volatility, respectively. The signs of the lead and lag sample correlations widely confirm sections 4.1.1 and 4.1.2. However, while the directional sentiment signals are positively correlated to past and current returns, they do not carry significant information for future returns. This is in line with economic findings about market efficiency and the impossibility of return prediction, as documented, amongst others, in Timmermann (2008). Furthermore, the correlations peak at the contemporaneous case. Nevertheless, while sentiment signals S_d are uncorrelated to future S&P returns we observe persistently significant positive correlations of disagreements D_d to the leads and lags of the S&P volatility metric.

The highly persistent autocorrelations in realized variance measures like r_d^{vol} (Table 3) are typically captured in time series models put forward by, e.g., Andersen et al. (2003). In order to avoid spurious and overstated results about the relationship between text-based

Corr(·,·)	l=−2	l=−1	l=0	l=1	l=2	l=3	l=4
r_{d+l}, S_d^{all}	**0.08**	**0.29**	**0.35**	0.00	-0.02	-0.05	-0.07
r_{d+l}, S_d^{BNB}	0.06	**0.30**	**0.38**	-0.01	-0.02	-0.05	**-0.08**
r_{d+l}, S_d^{SVM}	**0.09**	**0.38**	**0.48**	-0.02	-0.03	-0.06	**-0.09**
r_{d+l}, S_d^{SVMnl}	**0.09**	**0.37**	**0.48**	-0.02	-0.04	-0.06	**-0.09**
r_{d+l}, S_d^{topCl}	**0.08**	**0.36**	**0.48**	0.00	-0.02	-0.05	-0.06
r_{d+l}^{vol}, D_d^{all}	**0.12**	**0.18**	**0.18**	**0.20**	**0.19**	**0.18**	**0.16**
r_{d+l}^{vol}, D_d^{BNB}	**0.16**	**0.22**	**0.22**	**0.24**	**0.23**	**0.22**	**0.21**
r_{d+l}^{vol}, D_d^{SVM}	**0.17**	**0.25**	**0.29**	**0.28**	**0.27**	**0.25**	**0.24**
r_{d+l}^{vol}, D_d^{SVMnl}	**0.14**	**0.24**	**0.25**	**0.26**	**0.25**	**0.22**	**0.21**
r_{d+l}^{vol}, D_d^{topCl}	**0.07**	**0.18**	**0.19**	**0.20**	**0.20**	**0.19**	**0.17**

Table 4: Correlations of expert sentiment and disagreement to S&P returns and volatility. Abbreviations are as in Table 3. Significant sample correlations for $\alpha = 0.05$ under a normality assumption are bold-faced.

signals and the financial variables we explicitly take the time series dependencies into account. We propose to model both the S&P return and volatility measure in an ARMA model with explanatory variables (ARMA(X)).

In the following, let y_d represent the dependent variable r_d or r_d^{vol} and let x_d stand for a Twitter-based signal as covariate. For sake of brevity we concentrate on ARMA models including either the contemporaneous D_d or S_d or one-day lagged D_{d-1} or S_{d-1} as additional covariates x_d. The ARMA(X) model with order (p,q) is given as

$$y_d = c + \sum_{i=1}^{p} \alpha_i y_{d-i} + \sum_{j=1}^{q} \beta_j \varepsilon_{d-j} + \gamma x_d + \varepsilon_d, \quad d = 1, ..., \mathcal{D}, \quad \varepsilon_d \sim \mathcal{N}(0, \sigma^2). \tag{29}$$

The dependencies in y_d are captured via lags of y_d and lags of the errors ε_d. Model specification focuses on choosing p and q governing the lag structure. The ARMA model can account for high persistence in time series even with a parsimonious parameterization, i.e. with low values p and q (see Hamilton (1994) for details).

Solving (29) for ε_d and iterating on

$$\varepsilon_d = y_d - c - \gamma x_d - \sum_{i=1}^{p} \alpha_i y_{d-i} - \sum_{j=1}^{q} \beta_j \varepsilon_{d-j} \tag{30}$$

we can write the conditional log likelihood function for (29) in terms of the errors according to

$$\log f(y_{\mathcal{D}}, y_{\mathcal{D}-1}, ..|\mathbf{y}_0, \boldsymbol{\varepsilon}_0, x_0; c, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma, \sigma) = -\frac{\mathcal{D}}{2}\log(2\pi) - \frac{\mathcal{D}}{2}\log(\sigma^2) - \sum_{d=1}^{\mathcal{D}} \frac{\varepsilon_d^2}{2\sigma^2}, \qquad (31)$$

where f is the joint density of the observations y_d conditional on initial values $\mathbf{y}_0$=$\{y_0$, y_{-1},..,$y_{-p+1}\}$, x_0, $\boldsymbol{\varepsilon}_0 = \{\varepsilon_0$,..,$\varepsilon_{-q+1}\}$. The initial values can be set to zero or their theoretical expected values (Hamilton (1994)). Optimal parameter estimates for c, σ, $\boldsymbol{\alpha} = \{\alpha_1, .., \alpha_p\}$, $\boldsymbol{\beta} = \{\beta_1, .., \beta_2\}$ and γ are obtained through maximization of the likelihood function via standard numerical procedures.

We select the optimal ARMA model parameterization from all feasible combinations (p, q) up to $(p, q) = (5, 5)$ according to the Akaike information criterion. Table 5 gives specification details and estimation results of the ARMA models for r_d and r_d^{vol} as dependent variables. We document ARMA coefficients only up to order 1. Ljung-Box statistics confirm that the ARMA models always account for the dependence structure in both r_d and r_d^{vol}.

The following findings emerge. First, estimates for the coefficient of the contemporaneous sentiment S_d in the model for r_d are significant across all models, while the coefficient estimate for the lagged S_{d-1} is insignificant. This means we detect a significant contemporaneous relation of returns and directional expert sentiment measures, but no predictive property of the directional sentiment metric for returns. Second, a significant contemporaneous interaction of disagreement measures D_d and the S&P volatility r_d^{vol} can be confirmed for measures based on the linear and nonlinear SVM classification as well as for the top cluster set. The estimates for the lagged disagreement measures D_{d-1} are significant in the equation for r_d^{vol} at all conventional testing levels for all users sets except for the top cluster. Hence, we detect significant information value in the lagged D_{d-1} for the modeling of the realized volatility at d. The likelihood ratio test of $H_0 : \gamma = 0$ as an alternative to the t-test (yet asymptotically equivalent) confirms the significance results for the variables. Third, coefficient signs are identical to the signs of the correlations (see Table 4) and thus in line with economic theory. Fourth, differences in classification quality as documented in subsections 4.1.1 and 4.1.2 are reflected in the significance of coefficients γ. In specific, we find strongest results for the SVM classifiers. Most notably, the significant contempo-

y_d ; x_d	$\widehat{c}$	$\widehat{\alpha}_1$	$\widehat{\beta}_1$	$\widehat{\gamma}$	LR	LB_{20}
	Contemporaneous relation $r_d \leftrightarrow S_d$,			ARMA(2,2)		
r_d ; S_d^{all}	**0.06** (0.00)	0.18 (0.11)	**-0.27** (0.11)	**0.16** (0.01)	0.00	0.16
r_d ; S_d^{BNB}	**0.06** (0.00)	0.20 (0.11)	**-0.32** (0.11)	**0.14** (0.01)	0.00	0.20
r_d ; S_d^{SVM}	**0.05** (0.00)	0.16 (0.09)	**-0.33** (0.09)	**0.13** (0.01)	0.00	0.37
r_d ; S_d^{SVMnl}	**0.05** (0.00)	0.19 (0.10)	**-0.37** (0.10)	**0.13** (0.01)	0.00	0.29
r_d ; S_d^{topCl}	**0.04** (0.00)	-0.95 (0.53)	0.79 (0.53)	**0.10** (0.00)	0.00	0.22
	Contemporaneous relation $r_d^{vol} \leftrightarrow D_d$,			ARMA(1,4)		
r_d^{vol} ; D_d^{all}	-0.01 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	0.12 (0.07)	0.09	0.41
r_d^{vol} ; D_d^{BNB}	-0.01 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	0.09 (0.05)	0.09	0.39
r_d^{vol} ; D_d^{SVM}	**-0.03** (0.01)	**0.97** (0.01)	**-0.74** (0.03)	**0.23** (0.05)	0.00	0.52
r_d^{vol} ; D_d^{SVMnl}	-0.02 (0.01)	**0.97** (0.01)	**-0.73** (0.03)	**0.16** (0.05)	0.00	0.50
r_d^{vol} ; D_d^{topCl}	-0.01 (0.01)	**0.98** (0.01)	**-0.73** (0.03)	**0.09** (0.04)	0.03	0.40
	Lagged explanatory variable, $r_d \leftrightarrow S_{d-1}$,			ARMA(1,1)		
r_d ; S_{d-1}^{all}	0.00 (0.00)	**-0.84** (0.08)	**0.78** (0.09)	0.00 (0.01)	1.00	0.09
r_d ; S_{d-1}^{BNB}	0.00 (0.00)	**-0.84** (0.08)	**0.78** (0.09)	0.00 (0.01)	1.00	0.09
r_d ; S_{d-1}^{SVM}	0.00 (0.00)	**-0.85** (0.07)	**0.79** (0.09)	0.00 (0.01)	1.00	0.08
r_d ; S_{d-1}^{SVMnl}	0.00 (0.00)	**-0.84** (0.07)	**0.78** (0.09)	0.00 (0.01)	1.00	0.09
r_d ; S_{d-1}^{topCl}	0.00 (0.00)	**-0.84** (0.08)	**0.77** (0.09)	0.00 (0.01)	1.00	0.08
	Lagged explanatory variable, $r_d^{vol} \leftrightarrow D_{d-1}$,			ARMA(1,4)		
r_d^{vol} ; D_{d-1}^{all}	-0.02 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	**0.16** (0.07)	0.03	0.33
r_d^{vol} ; D_{d-1}^{BNB}	-0.01 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	**0.14** (0.05)	0.01	0.31
r_d^{vol} ; D_{d-1}^{SVM}	-0.02 (0.01)	**0.98** (0.01)	**-0.73** (0.03)	**0.15** (0.05)	0.01	0.30
r_d^{vol} ; D_{d-1}^{SVMnl}	-0.02 (0.01)	**0.97** (0.01)	**-0.73** (0.03)	**0.15** (0.05)	0.00	0.24
r_d^{vol} ; D_{d-1}^{topCl}	0.00 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	0.06 (0.04)	0.17	0.30

Table 5: ARMA model results for dependent variables r_d and r_d^{vol}. Notation as before. Standard errors in parentheses. Significance at the 5% level bold-faced. LR gives the p-value of the likelihood ratio test of $H_0 : \gamma = 0$. LB_{20} gives the p-value of the Ljung-Box test for autocorrelation in first $K = 20$ lags based on the null $H_0 : \rho_1 = .. = \rho_K = 0$.

raneous influence of the author disagreement on r_d^{vol} can only be observed for expert user subsets.

In summary, the results show that the links between financial market metrics and tweet-based expert sentiment and topic disagreement measures hold even when time series properties of returns and volatility measures are accounted for. In specific, current market swings in terms of returns are captured by the contemporaneous directional topic sentiment of tweets. Further, we conclude that the opinion disagreement conveyed by expert tweets reflects information diffusion or asymmetry, which is a driver of current and future financial volatility. This confirms economic models of information dissemination on financial markets by, e.g., Harris and Raviv (1993) and Kandel and Pearson (1995). In these models, increasing differences in opinions and higher information asymmetry correspond to increasing price volatility.

4.3 On Robustness and Reproducibility

Several robustness checks underscore the validity of our study. A supplementary web appendix at https://www.dropbox.com/s/3m7l108drrsxaln/bigdataappendix.pdf?dl=0 shows that results are robust with respect to a change of the underlying word lists and to a change of the volatility measure. In addition, we find that results hold for alternative stock market indices like the EUROSTOXX 50 and the NASDAQ index.

Moreover, the present study is reproducible due to the heavy and exclusive use of open source software and public word lists. Streaming tweet data can be freely obtained for academic purposes via Twitter's streaming API.

5 Conclusions

Constructing signals for financial market applications from text messages posted on social media platforms presents a major challenge as most messages are overlaid by substantial noise. Specifically, the text data are typically unstructured, heterogeneous and heavily contaminated by informal language and irrelevant information. As yet another challenge, the noise reduction through larger sample sizes requires the processing of huge amounts of data, which makes it difficult to obtain long historical time series. Together, these obstacles obscure the information value of social media data and impede the identification of a possible relationship between signals derived from text data and financial market developments.

Making use of a unique data set of microblogs containing finance-related keywords, this study presents robust results on the relationship between signals from Twitter messages (tweets) and broad market returns and volatility. To reduce the impact of noise, we use statistical text processing tools in a first step to identify expert subsets of Twitter users based on their linguistic fingerprint. Having identified such user subgroups we construct simple and established daily sentiment and topic disagreement metrics for the user groups. While the sentiment measure represents a directional measure of the polarity of tweet topics, the non-directional disagreement measure is devised to capture the volatility in opinions disseminated on Twitter. In a second step we examine the relation of sentiment measures and S&P 500 returns as well as between disagreement measures and the S&P 500 volatility.

First, we show that supervised and unsupervised statistical learning methods applied to text data are capable of identifying expert users with a more pronounced finance focus. Among statistical methods, support vector machines emerge as the most attractive classifier. Second, we document that signals based on expert groups allow to uncover links of text-based signals to stock market returns and volatility. Third, a time series analysis reveals that relations between text-based signals and global market developments are in line with economic theory. In specific, directional expert user sentiments reflect the contemporaneous direction of returns. Further, disagreement among expert users explains current and future return volatility, which confirms theoretical models postulating a positive relation between opinion differences and financial market volatility.

Overall, we find that statistical methods for natural language are successful in structuring and categorizing the noisy message stream on the Twitter social media platform. This allows to deeply investigate in how far financial market developments are reflected in tweets from Twitter users with finance focus. Moreover, the findings provide first 5-year evidence that social media-based disagreement metrics are useful covariates in time series models of returns and realized financial market volatility.

References

Aizawa, A. (2000): "The Feature Quantity: An Information Theoretic Perspective of Tfidf-like Measures," in *Proc. of ACM SIGIR 2000*, SIGIR, 104–111.

Alizadeh, S., M. W. Brandt, and F. X. Diebold (2002): "Range-Based Estimation of Stochastic Volatility Models," *The Journal of Finance*, 57, 1047–1091.

Andersen, T. G., T. Bollerslev, F. X. Diebold, and P. Labys (2003): "Modeling and Forecasting Realized Volatility," *Econometrica*, 71, 579–625.

Antenucci, D., M. Cafarella, M. Levenstein, C. Ré, and M. D. Shapiro (2014): "Using Social Media to Measure Labor Market Flows," Working Paper 20010, National Bureau of Economic Research.

Antweiler, W. and M. Z. Frank (2004): "Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards," *The Journal of Finance*, 59, 1259–1294.

Bollen, J., H. Mao, and X. Zeng (2011): "Twitter mood predicts the stock market," *Journal of Computational Science*.

Boyack, K., D. Newman, R. Duhon, R. Klavans, J. Patek, M.and Biberstine, B. Schijvenaars, A. Skupin, N. Ma, and K. Boerner (2011): "Clustering More than Two Million Biomedical Publications: Comparing the Accuracies of Nine Text-Based Similarity Approaches," *PLoS ONE*, 6.

Boyd, S. and L. Vandenberghe (2004): *Convex optimization*, Cambridge University Press.

Broniatowski, D. A., M. J. Paul, and M. Dredze (2013): "National and Local Influenza Surveillance through Twitter: An Analysis of the 2012-2013 Influenza Epidemic," *PLoS ONE*, 8, e83672.

Burges, C. J. C. (1998): "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 2, 121–167.

Chen, H., P. De, Y. J. Hu, and B.-H. Hwang (2014): "Wisdom of Crowds: The Value of Stock Opinions Transmitted Through Social Media," *Review of Financial Studies.*

Cristianini, N. and J. Shawe-Taylor (2000): *Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press.

Das, S. R. and M. Y. Chen (2007): "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Management Science*, 53, 1375–1388.

Davis, A. K., J. M. Piger, and L. M. Sedor (2012): "Beyond the Numbers: Measuring the Information Content of Earnings Press Release Language," *Contemporary Accounting Research*, 29, 845–868.

Deerwester, S., S. T. Dumais, G. W. F. T. K. Landauer, and R. Harshman (1990): "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, 41, 391–407.

Dong, Q., X. Wang, and L. Lin (2006): "Application of latent semantic analysis to protein remote homology detection," *Bioinformatics*, 22, 285–290.

Gross-Klussmann, A. and N. Hautsch (2011): "When machines read the news: Using automated text analytics to quantify high frequency news-implied market reactions," *Journal of Empirical Finance*, 18, 321–340.

Gurun, U. G. and A. W. Butler (2012): "Don't Believe the Hype: Local Media Slant, Local Advertising, and Firm Value," *The Journal of Finance*, 67, 561–598.

Haerdle, W., Y.-J. Lee, D. Schaefer, and Y.-R. Yeh (2009): "Variable selection and oversampling in the use of smooth support vector machines for predicting the default risk of companies," *Journal of Forecasting*, 28, 512–534.

Hamilton, J. D. (1994): *Time Series Analysis*, Princeton, NJ: Princeton University Press.

Harris, M. and A. Raviv (1993): "Differences of opinion make a horse race," *The Review of Financial Studies*, 6.

Hastie, T., R. Tibshirani, and J. H. Friedman (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer, 2nd ed.

He, H. and E. A. Garcia (2009): "Learning from Imbalanced Data," *IEEE Trans. on Knowl. and Data Eng.*, 21, 1263–1284.

Joachims, T. (1998): "Text categorization with support vector machines: Learning with many relevant features," in *Proc. ECML*, Heidelberg: Springer, 137–142.

Kandel, E. and N. Pearson (1995): "Differential interpretation of public signals and trade in speculative markets," *The Journal of Political Economy*, 103.

Lamb, A., M. J. Paul, and M. Dredze (2013): "Separating fact from fear: Tracking flu infections on Twitter," in *In NAACL*.

Lewis, D. D. (1998): "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," in *Proc. ECML*, London, UK: Springer, 4–15.

Li, F. (2008): "Annual report readability, current earnings, and earnings persistence," *Journal of Accounting and Economics*, 45, 221 – 247, economic Consequences of Alternative Accounting Standards and Regulation.

Loughran, T. and B. McDonald (2011): "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *The Journal of Finance*, 66, 35–65.

MacQueen, J. B. (1967): "Some methods for classification and analysis ofmultivariate observations," in *Proc. Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, 281–297.

Manning, C. D., P. Raghavan, and H. Schütze (2008): *Introduction to Information Retrieval*, New York, NY, USA: Cambridge University Press.

Maron, M. E. and J. L. Kuhns (1960): "On relevance, probabilistic indexing, and information retrieval," *Journal of the ACM*, 7, 216–244.

McCallum, A. and K. Nigam (1998): "A Comparison of Event Models for Naive Bayes Text Classification," in *AAAI ICML Workshop on Learning for Text Categorization*, 41–48.

Parkinson, M. (1980): "The Extreme Value Method for Estimating the Variance of the Rate of Return," *The Journal of Business*, 53, 61–65.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011): "Scikit learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830.

Porter, M. F. (1980): "An algorithm for suffix stripping," *Program*, 14, 130–137.

Salton, G. and C. Buckley (1988): "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, 24, 513–523.

Schölkopf, B. and A. J. Smola (2001): *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press.

Schütze, H., D. A. Hull, and J. O. Pedersen (1995): "A Comparison of Classifiers and Document Representations for the Routing Problem," in *Proc. SIGIR*, ACM Press, 229–237.

Schütze, H. and C. Silverstein (1997): "Projections for Efficient Document Clustering," in *Proc. SIGIR*, ACM Press, 74–81.

Sebastiani, F. (2002): "Machine Learning in Automated Text Categorization," *ACM Comput. Surv.*, 34, 1–47.

Sprenger, T. O., A. Tumasjan, P. G. Sandner, and I. M. Welpe (2014): "Tweets and Trades: the Information Content of Stock Microblogs," *European Financial Management*, 20, 926–957.

Stone, P. J., D. C. Dunphy, M. S. Smith, and D. M. Ogilvie (1966): *The general inquirer: A computer approach to content analysis*, Cambridge, MA: The MIT Press.

Tetlock, P. C. (2007): "Giving Content to Investor Sentiment: The Role of Media in the Stock Market," *The Journal of Finance*, 62, 1139–1168.

Tetlock, P. C., M. Saar-Tsechansky, and S. Macskassy (2008): "More Than Words: Quantifying Language to Measure Firms' Fundamentals," *The Journal of Finance*, 63, 1437–1467.

Tibshirani, R., G. Walther, and T. Hastie (2001): "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 411–423.

Timmermann, A. (2008): "Elusive Return Predictability," *International Journal of Forecasting*, 1–18.

Tumasjan, A., T. O. Sprenger, P. G. Sandner, and I. M. Welpe (2010): "Election Forecasts With Twitter: How 140 Characters Reflect the Political Landscape," *Social Science Computer Review*.

Vapnik, V. N. (1979): *Estimation of Dependencies based on empirical Data*, Moscow: Nauka.

——— (1998): *Statistical Learning Theory*, Wiley-Interscience.

Zhao, Y. and G. Karypis (2004): "Empirical and theoretical comparison of selected criterion functions for document clustering," *Machine Learning*, 55, 311–331.

6 Appendix

6.1 Twitter Users with Finance Focus

websites:
http://www.businessinsider.com/the-best-finance-people-on-twitter-2012-4?op=1#!Cj8ed
http://finansakrobat.com/blog2/2013/2/5/who-to-follow-on-twitter-the-finance-edition
http://blogs.sap.com/innovation/financial-management/top-50-financial-twitter
-influencers-019897
http://www.marketfolly.com/2012/06/top-finance-people-to-follow-on-twitter.html
http://www.moneysense.ca/must-read/borzykowski-follow-these-finance-related-twitter
-feeds

The following lists summarize selected twitter users as publicly listed on the above websites through December 31st, 2014. We split 190 users into training and test sample as follows.

A) Training Sample

@BloombergMrkts, @BloombergNews, @Forbes, @Reuters, @BW, @ftfinancenews, @CNNMoney, @CNBCFastMoney, @WSJ, @YahooFinance, @CNBCWorld, @MarketWatch, @IBDinvestors, @MSN_Money, @FoxBusiness, @RealMarketNews, @NYSEEuronext, @nyse, @TabbFORUM, @EconBizFin, @CNBCClosingBell, @dowbands, @FXStreetNews, @Street_Insider, @HedgeWorld, @cr_harper, @globeinvestor, @PhilipEtienne, @CapitalObserver, @LongShortTrader, @MebFaber, @bespokeinvest, @StockJockey, @ritholtz, @footnoted, @Legacy_Trades, @zerohedge, @AllAboutAlpha, @abnormalreturns, @StockTwits, @Benzinga, @tradefast, @researchpuzzler, @Retail_Guru, @montoyan, @MicroFundy, @StoneStAdvisors, @MarioGabelli, @samgadjones, @herbgreenberg, @TheStalwart, @Alphal_pha, @BergenCapital, @fundmyfund, @mojoris1977, @BarbarianCap, @VolSlinger, @TechStockRadar, @jennablan, @vitaliyk, @Greg_Speicher, @DanZanger, @OptionsHawk, @cperruna, @calculatedrisk, @katie_martin_FX, @ReformedBroker, @selaco, @felixsalmon, @FGoria, @chrisadamsmkts, @Pawelmorski, @Stotty67, @LorcanRK, @SimoneFoxman, @PeterCWarren, @GuyJohnsonTV, @Merimack1, @howardlindzon, @moorehn, @kitjuckes,

@CarbonBubble, @moveyourmoneyuk, @triodosuk, @interfluidity, @Lavorgnanomics, @TFMkts, @GoldmanSachs, @EddyElfenbein, @nanexllc, @DavidSchawel, @ppearlman, @pcdunham

B) Test Sample

@EnisTaner, @conorsen, @allstarcharts, @RiskReversal, @IvanTheK, @yvessmith, @InterestArb, @mark_dow, @auaurelija, @stlouisfed, @barnejek, @dvolatility, @JustinWolfers, @PIMCO, @brianmlucey, @fwred, @Stovall_SPCapIQ, @groditi, @Fullcarry, @mbusigin, @prchovanec, @AswathDamodaran, @TheArmoTrader, @ukarlewitz, @stt2318, @Arbeter_SPCapIQ, @economistmeg, @cbk_chi, @retheauditors, @spbaines, @sspencer_smb, @jfahmy, @Ralph_Acampora, @agwarner, @hsilverb, @valuewalk, @TradeDesk_Steve, @MorganStanley, @MS_Econ, @DanBTIG, @IanShepherdson, @ianbremmer, @JimPethokoukis, @jkrinskypga, @gusbaratta, @BLS_gov, @cullenroche, @paul_vestact, @EU_Eurostat, @WilliamsonChris, @D_Blanchflower, @toby_n, @tomkeene, @matt_levine, @FinancialTimes, @izakaminska, @ericgplatt, @PreetaTweets, @politico, @morningmoneyben, @CNBC, @KateKellyCNBC, @EamonJavers, @carney, @carlquintanilla, @diana_olick, @moneymorning, @AAAMPblog, @ColinTWilliams, @Nouriel, @BenChu_, @alessiorastani, @finplan, @susanweiner, @LewisFinancial, @daily_finance, @Wealthfront, @allanschoenberg, @MoneyNing, @MichaelKitces, @EconomyUS, @CBSMoneyWatch, @BethKobliner, @CNBCnow, @tim, @deborahgage, @FinanceFeed, @SunFinancial, @davidkwaltz, @TheStreet, @ryanavent, @jaredwoodard, @SPDJIndices

6.2 Model Selection

6.2.1 The K in K-Means

Consider the intra-cluster distances between points in a given cluster C_k,

$$D_k = \sum_{x_i \in C_k} \sum_{x_j \in C_k} ||x_i - x_j||^2 = 2n_k \sum_{x_i \in C_k} ||x_i - \mu_k||^2, \tag{32}$$

where n_k is the number of data points in cluster k. A measure of compactness of the clustering is given as

$$W_k = \sum_{k=1}^{K} \frac{1}{2n_k} D_k. \tag{33}$$

The idea of the gap statistic is to compare the compactness metric $\log W_k$ to the expected value of the $\log W_k$ distribution under the null hypothesis of no clustering in the data. The more compact the clustering, the better the fit of the clustering method and, hence, the more likely do we have a correctly specified model.

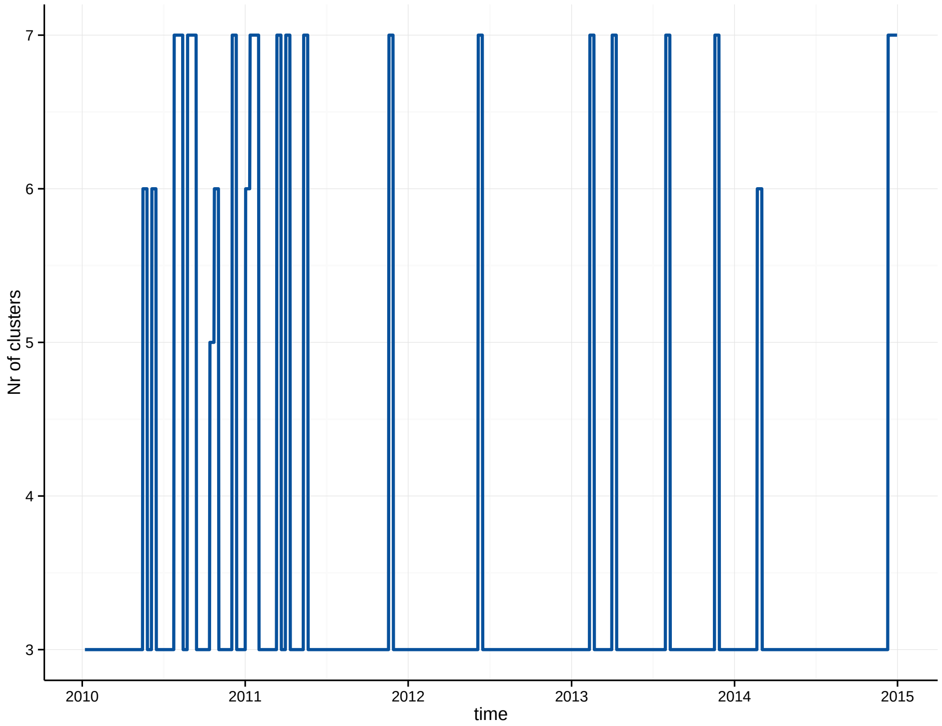

Figure 6: Evolution of the cluster number for the K-Means.

In specific, we consider the difference of the expected value under H_0 and the observed value according to

$$Gap_n(k) = \mathbb{E}_n^*[\log W_k] - \log W_k. \tag{34}$$

$\log W_k^*$ is obtained via Monte Carlo simulation based on B draws from the reference uniform distribution. With $sd(k)$ denoting the standard deviation of the simulated W_k^* we compute

$$s_k = \sqrt{1 + 1/B}\text{sd}(k). \tag{35}$$

The optimal number of clusters is chosen to be the smallest k such that $Gap(k) \geq Gap(k+1) - s_{k+1}$.

Figure 6 shows the cluster over time. We hold the cluster number constant for two weeks.

6.2.2 Hyperparameters for the supervised learning algorithms

Figure 7 gives the evolution of hyperparameters for the machine learning models applied to the vector space of words. We calibrate on a daily basis based on the F1 measure. Parameters are chosen from a grid of values such as to maximize the F1 measure through 10-fold cross validation on the daily training samples. The grids are as follows.

Linear SVM: We choose $C \in \{0.01,\ 0.1,\ 1,\ 2,\ 5,\ 10,\ 100,\ 600,\ 1000,\ 10000,\ 100000,\ 1000000,\ 10000000,\ 100000000,\ 1000000000,\ 10000000000\}$.
Nonlinear SVM: The grid for C is the same as for the linear SVM. The additional γ is chosen from $\{$ 1e-09, 1e-08, 1e-07, 1e-06, 1e-05, 1e-04, 1e-03, 1e-02, 1e-01, 1e+00, 1e+01, 1e+02, 1e+03 $\}$

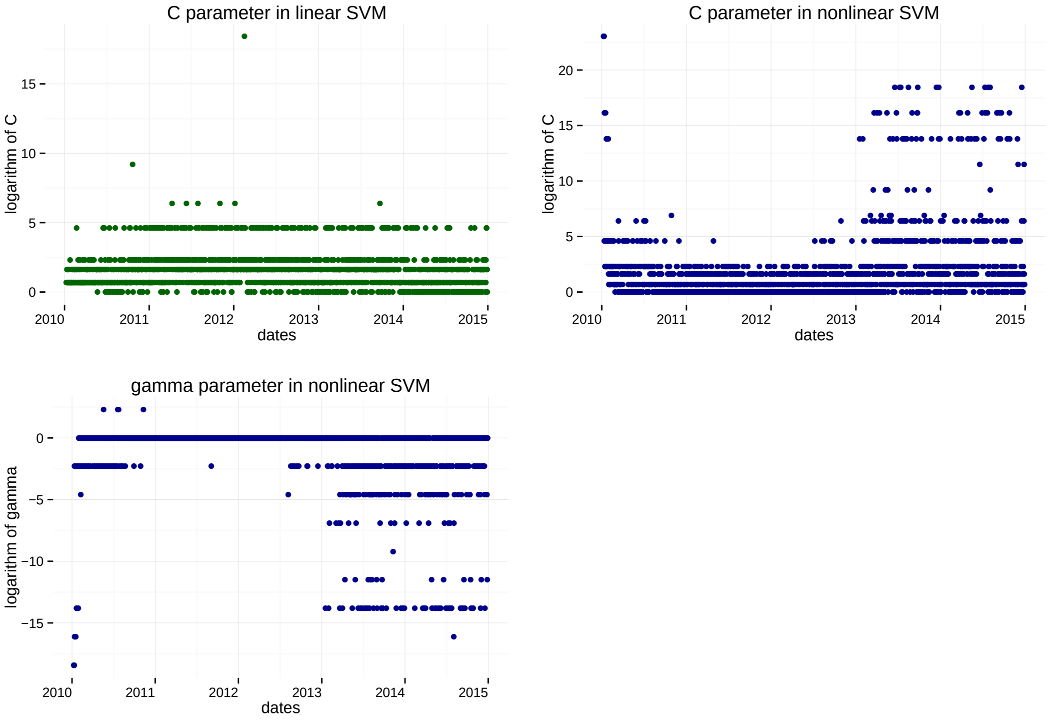

Figure 7: Evolution of parameters as calibrated by the Cross-Validation based on the ROC AUC criterion.

y_d ; x_d	$\widehat{c}$	$\widehat{\alpha}_1$	$\widehat{\beta}_1$	$\widehat{\gamma}$	LR	LB_{20}
			Contemporaneous relation $r_d \leftrightarrow S_d$			
r_d ; S_d^{all}	**0.06** (0.00)	0.18 (0.11)	**-0.27** (0.11)	**0.16** (0.01)	0.00	0.16
$r_d;S_d^{BNBN}$	**0.02** (0.00)	-0.13 (0.21)	0.06 (0.21)	**0.06** (0.01)	0.00	0.22
$r_d;S_d^{SVMN}$	**0.01** (0.00)	-0.15 (0.25)	0.09 (0.25)	**0.03** (0.01)	0.00	0.25
$r_d;S_d^{SVMnlN}$	**0.01** (0.00)	-0.13 (0.25)	0.08 (0.25)	**0.03** (0.01)	0.00	0.26
$r_d;S_d^{lowCl}$	**0.01** (0.00)	0.09 (0.14)	-0.15 (0.15)	**0.01** (0.00)	0.00	0.26
			Contemporaneous relation $r_d^{vol} \leftrightarrow D_d$			
r_d^{vol} ; D_d^{all}	-0.01 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	0.12 (0.07)	0.09	0.41
$r_d^{vol};D_d^{BNBN}$	-0.02 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	**0.19** (0.09)	0.04	0.4
$r_d^{vol};D_d^{SVMN}$	0.02 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	-0.09 (0.06)	0.19	0.32
$r_d^{vol};D_d^{SVMnlN}$	0.01 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	-0.04 (0.06)	0.50	0.34
$r_d^{vol};D_d^{lowCl}$	0.00 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	0.08 (0.04)	0.06	0.30
			Lagged explanatory variable, $r_d \leftrightarrow S_{d-1}$			
r_d ; S_{d-1}^{all}	0.00 (0.00)	**-0.84** (0.08)	**0.78** (0.09)	0.00 (0.01)	1.00	0.09
$r_d;S_{d-1}^{BNBN}$	0.00 (0.00)	0.09 (0.18)	-0.15 (0.18)	0.00 (0.01)	1.00	0.25
$r_d;S_{d-1}^{SVMN}$	0.00 (0.00)	0.07 (0.18)	-0.13 (0.18)	0.00 (0.01)	1.00	0.25
$r_d;S_{d-1}^{SVMnlN}$	0.00 (0.00)	0.08 (0.20)	-0.13 (0.20)	0.00 (0.01)	1.00	0.25
$r_d;S_{d-1}^{lowCl}$	0.00(0.00)	0.09 (0.18)	-0.14 (0.18)	0.00 (0.00)	1.00	0.26
			Lagged explanatory variable, $r_d^{vol} \leftrightarrow D_{d-1}$			
r_d^{vol} ; D_{d-1}^{all}	-0.02 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	**0.16** (0.07)	0.03	0.33
$r_d^{vol};D_{d-1}^{BNBN}$	0.00 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	0.02 (0.09)	0.86	0.34
$r_d^{vol};D_{d-1}^{SVMN}$	0.00 (0.01)	**0.97** (0.01)	**-0.72** (0.03)	0.06 (0.06)	0.31	0.35
$r_d^{vol};D_{d-1}^{SVMnlN}$	0.00 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	0.01 (0.06)	0.79	0.35
$r_d^{vol};D_{d-1}^{lowCl}$	0.00 (0.01)	**0.98** (0.01)	**-0.72** (0.03)	0.02 (0.04)	0.67	0.34

Table 6: ARMA model results for dependent variables r_d and r_d^{vol}. Notation widely as before. However, we add a N-superscript to indicate that we operate on the non-expert groups. The cluster superscript 'lowCl' stands for the cluster with lowest correlation to the economy lexicon.

6.3 Results for the Non-Expert group of Users

The following table gives the ARMA-model results for the non-expert user groups. Specification as in Table 5 except for the case $r_d \leftrightarrow S_d$. Here we use $(2, 3)$ as ARMA-specification to better account for autocorrelation structure in the errors.

Overall, we find weaker relations of financial markets and sentiment and disagreement signals. While lagged expert disagreement signals have a significant influence on future volatility measures (see Table 5), this cannot be confirmed by the corresponding non-expert groups.