# A temporal model of text periodicities using Gaussian Processes

**Daniel Preoţiuc-Pietro, Trevor Cohn**
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4DP, United Kingdom
{daniel,t.cohn}@dcs.shef.ac.uk

## Abstract

Temporal variations of text are usually ignored in NLP applications. However, text use changes with time, which can affect many applications. In this paper we model periodic distributions of words over time. Focusing on hashtag frequency in Twitter, we first automatically identify the periodic patterns. We use this for regression in order to forecast the volume of a hashtag based on past data. We use Gaussian Processes, a state-of-the-art bayesian non-parametric model, with a novel periodic kernel. We demonstrate this in a text classification setting, assigning the tweet hashtag based on the rest of its text. This method shows significant improvements over competitive baselines.

## 1 Introduction

Temporal changes in text corpora are central to our understanding of many linguistic and social phenomena. Social Media platforms and the digitalization of libraries provides a vast body of timestamped data. This allows studying of the complex temporal patterns exhibited by text usage including highly non-stationary distributions and periodicities. However, temporal effects have been mostly ignored by previous work on text analysis or at best dealt with by making strong assumptions such as smoothly varying parameters with time (Yogatama et al., 2011) or modelled using a simple uni-modal distri bution (Wang and McCallum, 2006). This paper develops a temporal model for classifying microblog posts which explicitly incorporates multimodal periodic behaviours using Gaussian Processes (GPs).

We expect text usage to follow multiple periodicities at different scales. For example, people on Social Media might talk about different topics during and after work on weekdays, talk every Friday about the weekend ahead, or comment about their favorite weekly TV show during its air time. Given this, text frequencies will display periodic patterns. This applies to other text related quantities like co-occurrence values or topic distributions over time, as well as applications outside NLP like user behaviour (Preoţiuc-Pietro and Cohn, 2013).

Modelling temporal patterns and periodicities can be useful to tasks like text classification. For example a tweet containing ‘music’ is normally attributed to a general hashtag about music like #np (now playing). However, knowing time, if it occurs during the (weekly periodic) air time of ‘American Idol’ it is more likely for it to belong to #americanidol or if its mentioned in the days building up to the Video Music Awards to be assigned to #VMA.

In NLP, temporal models have treated time in overly simplistic ways and without regard to periodicities. We propose a model that first broadly identifies several types of temporal patterns: a) periodic, b) constant in time, c) falling out of use after enjoying a brief spell of popularity (e.g. internet memes, news). This is performed automatically only using training data and makes no assumptions on the existence or the length of the periods we aim to model. We demonstrate the approach by modelling frequencies of hashtag occurrences in Twitter. Hashtags are user-generated labels included in tweets by their authors in order to assign them to a conversation and can be considered as a proxy for topics.

To this end, we make use of Gaussian Processes (GP) (Rasmussen and Williams, 2005), a

Bayesian non-parametric model for regression. Using the *Bayesian evidence* we automatically perform model selection to classify temporal patterns. We aim to use the most suitable model for extrapolation, i.e. predicting future values from past observations. The GP is fully defined by the covariance structure assumed between the observed points, and its hyperparameters, which can be automatically learned from data. We also introduce a new kernel suitable to model the periodic behaviour we observe in text: periods of low frequency followed by bursts at regular time intervals. We demonstrate that the GP approach is more general and gives better results than frequentist models (e.g. autoregressive models) because it incorporates uncertainty explicitly and elegantly, in addition to automatic model selection and parameter fitting.

To demonstrate the practical importance of our approach, we use our GP prediction as a prior in a Naïve Bayes model for text classification showing improvements over baselines which do not account for temporal periodicities. Our approach extends to more general uses, e.g. to discriminative text regression and classification. More broadly, we aim to establish GPs as a state-of-the-art model for regression and classification in NLP. To our knowledge, this is the first paper to use GP regression for forecasting and model selection within a NLP task.

All the hashtag time series data and the implementation of the PS kernel in the popular open-source Gaussian Processes packages GPML[1] and GPy[2] are available on the author's website[3].

## 2 Related Work

Time varying text patterns have been of particular interest in topic modelling. Griffiths and Steyvers (2004) analyse evolution of topics over time, but without modelling time explicitly. Extensions that model time make different assumptions, usually regarding smoothing proprieties in (Wang and McCallum, 2006; Blei and Lafferty, 2006; Wang et al., 2008; Hennig et al., 2012). Yogatama et al. (2011) proposed a regulariser for generalised linear models that encourages local temporal smoothness.

Modelling periodicities is one of the standard applications of Gaussian Processes (Rasmussen and Williams, 2005). Recent work by Wilson and Adams (2013) and Durrande et al. (2013) show how different periods can be identified from data. In general, methods that assume certain periodicities at daily or weekly levels were proposed e.g. in (McInerney et al., 2013). GPs were used with text by Polajnar et al. (2011) and for Quality Estimation regression in (Cohn and Specia, 2013; Shah et al., 2013).

Temporal patterns for short, distinctive lexical items such as hashtags and memes were quantitatively studied (Leskovec et al., 2009) and clustered (Yang and Leskovec, 2011) in Social Media. (Yang et al., 2012) studies the dual role of hashtags, of bookmarks of content and symbols of community membership, in the context of hashtag adoption. (Romero et al., 2011) analyses the patterns of temporal diffusion in Social Media finding that hashtags have also a persistence factor.

For predicting future popularity of hashtags, Tsur and Rappoport (2012) use linear regression with a wide range of features. (Ma et al., 2012; Ma et al., 2013) frame the problem as classification into a number of fixed intervals and applies all the standard classifiers. None of these studies model periodicities, although the former stresses their importance for accurate predictions. For predicting the hashtag given the tweet text, Mazzia and Juett (2011) uses the Naïve Bayes classifier with the uniform and empirical prior or TF-IDF weighting.

## 3 Gaussian Processes

In this paper we consider Gaussian Process (GP) models of regression (Rasmussen and Williams, 2005). GP is a probabilistic machine learning framework incorporating kernels and Bayesian non-parametrics which is widely considered as state-of-the-art for regression. The GP defines a prior over functions which applied at each input point gives a response value. Given data, we can analytically infer the posterior distribution of these functions assuming Gaussian noise. The kernel of the GP defines the covariance in response values as a function of its inputs.

We can identify two different set-ups for a regression problem. If the range of values to be predicted

[1] `http://www.gaussianprocess.org/gpml/code`

[2] `https://github.com/SheffieldML/GPy`

[3] `http://www.preotiuc.ro`

lies *within* the bounds of the training set we call the prediction task as interpolation. If the range of the prediction is *outside* the bounds, then our problem that of extrapolation. In this respect, extrapolation is considered a more difficult task and the covariance kernel which incorporates our prior knowledge plays a major role in the prediction.

There is the case when multiple covariance kernels can describe our data. For choosing the right kernel and its hyperparameters only using the training data we employ Bayesian model selection which makes a trade-off between the fit of the training data and model complexity. We now briefly give an overview of GP regression, kernel choice and model selection. We refer the interested reader to (Rasmussen and Williams, 2005) for a detailed introduction to GPs.

### 3.1 Gaussian Process Regression

Consider a time series regression task where we only have one feature, the value $x_t$ at time $t$. Our training data consists of $n$ pairs $\mathcal{D} = \{(t, x_t)\}$. The model will need to predict values $x_t$ for values of $t$ greater than those in the dataset.

GP regression assumes a latent function $f$ that is drawn from a GP prior $f(t) \sim \mathcal{GP}(m, k(t, t'))$ where $m$ is the mean and $k$ a kernel. The prediction value is obtained by the function evaluated at the corresponding data point, $x_t = f(t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is white-noise. The GP is defined by the mean $m$, here 0, and the covariance kernel function, $k(t, t')$.

The posterior at a test point $t_*$ is given by:

$$p(x_*|t_*, \mathcal{D}) = \int_f p(x_*|t_*, f) \cdot p(f|\mathcal{D}) \qquad (1)$$

where $x_*$ and $t_*$ are the test value and time. The posterior $p(f|\mathcal{D})$ shows our belief over possible functions after observing the training set $\mathcal{D}$. The predictive posterior can be solved analytically with solution:

$$\begin{aligned} x_* \sim \mathcal{N}(&k_*^T(K + \sigma_n^2 I)^{-1}\mathbf{t}, \\ &k(t_*, t_*) - k_*^T(K + \sigma_n^2 I)^{-1} k_*) \end{aligned} \qquad (2)$$

where $k_* = [k(t_*, t_1)...k(t_*, t_n)]^T$ are the kernel evaluations between the test point and all the training points, $K = \{k(t_i, t_j)\}_{j=1..n}^{i=1..n}$ is the Gram matrix over the training points and $\mathbf{t}$ is the vector of training points. The posterior of $x_*$ includes the mean response as well as its variance, thus expressing the uncertainty of the prediction. In this paper, we will consider the forecast as the expected value. Due to the matrix inversion in 2, inference takes $O(n^3)$ time where $n$ is the number of training points.

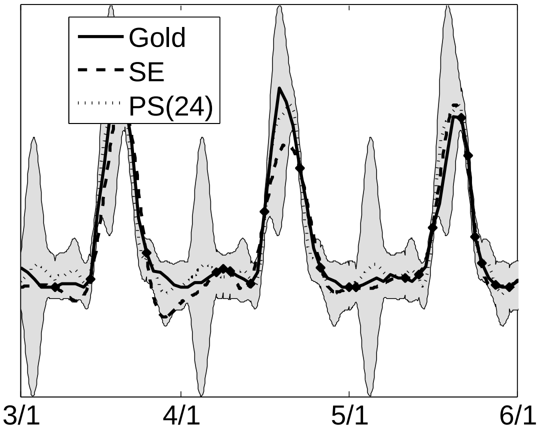

Figure 1: Interpolation for #goodmorning over 3 days with SE and PS(p=24,s=3) kernels. Prediction variance shown in grey for PS(24). Crosses represent training points.

### 3.2 Kernels

The covariance kernel together with its parameters fully define the GP (we assume 0 mean). The kernel induces similarities in the response between pairs of data points. Intuitively, if we want a smooth function, closer points should have high covariance compared to points that are further apart. If we want a periodic behaviour points at period length intervals should have the highest covariance. Usually, this is defined by an isotropic kernel, which means its invariant to all rigid motions.

For interpolation, a standard kernel (e.g. squared exponential) that encourages smooth functions is normally used. Figure 1 shows regression over 3 days for #goodmorning when only a random third of the values of the function are observed. We see that both the SE kernel and a periodic kernel (PS, see below) give good results.

However, for extrapolation, the choice of the kernel is paramount. The kernel encodes our prior belief about the type of function wish to learn. To illustrate this, in Figure 3, we show the time series for #goodmorning over 2 weeks and plot the regression for the future week learned by using different kernels.

In this study we will use multiple kernels, each most suitable for a specific category of temporal patterns in our data. This includes a new kernel inspired by observed word occurrence patterns. The kernels we use are:

**Constant (C):** The constant kernel is $k_C(t,t') = c$. Its mean prediction will always be the value $c$ and its assumption is that the signal is modeled only by Gaussian noise centred around this value. This describes the data best when we have a noisy signal around a stationary mean value.

**Squared exponential (SE):** The SE kernel or the Radial Basis Function (RBF) is the standard kernel used in most interpolation settings.

$$k_{SE}(t,t') = s^2 \cdot \exp\left(-\frac{(t-t')^2}{2l^2}\right) \qquad (3)$$

This gives a smooth transition between neighbouring points and best describes time series with a smooth shape e.g. a uni-modal burst with a steady decrease. However, its uncertainty grows with for predictions well into the future. Its two parameters $s$ and $l$ are the characteristic lengthscales along the two axes. Intuitively, they control the distance of inputs on a particular axis from which the function values become uncorrelated. Using the SE kernel corresponds to Bayesian linear regression with an infinite number of basis functions (Rasmussen and Williams, 2005).

**Linear (Lin):** The linear kernel describes a linear relationship between outputs.

$$k_{Lin}(t,t') = \frac{|t \cdot t'| + 1}{s^2} \qquad (4)$$

This can be obtained from linear regression by having $\mathcal{N}(0,1)$ priors on the corresponding regression weights and a prior of $\mathcal{N}(0,s^2)$ on the bias.

**Periodic (PER):** The periodic kernel represents a SE kernel in polar coordinates[4].

$$k_{PER}(t,t') = s^2 \cdot \exp \cdot \left(-\frac{2\sin^2(2\pi(t-t')/p)}{l^2}\right) \qquad (5)$$

It has a sinusoidal shape and is good at modelling periodically patterns that oscillate between low and high frequency. $s$ and $l$ are characteristic lengthscales as in the SE kernel and $p$ is the period.

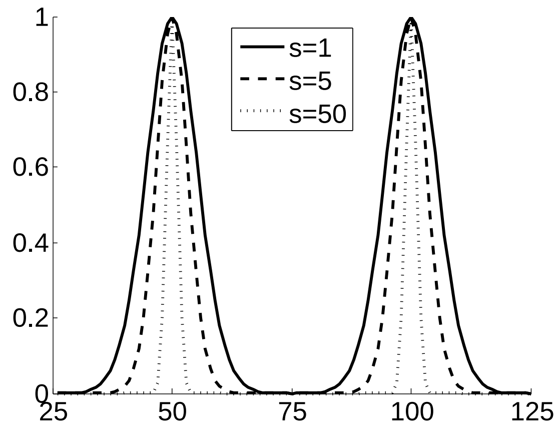

Figure 2: Behaviour of the PS kernel (p=50) with varying s. Values normalized in [0,1] interval.

**Periodic spikes (PS):** For textual time series, like word frequencies, we identify the following periodic behaviour: abrupt rise in usage, usually with a peak, followed by periods of low occurrence, which can be short (e.g. during the night) or long lived (e.g. the entire week except for a few hours). For modelling we introduce the following kernel:

$$\begin{aligned} k_{PS}(t,t') = &\cos\left(\sin\left(\frac{2\pi \cdot (t-t')}{p}\right)\right) \\ &\cdot \exp\left(\frac{s\cos(2\pi \cdot (t-t'))}{p} - s\right) \end{aligned} \qquad (6)$$

The kernel is parameterised by its period $p$ and a shape parameter $s$. The period indicates the time interval between the peaks of the function, while the shape parameter controls the *width* of the spike. The behaviour of the kernel is illustrated in Figure 2. We constrain $s \geq 1$.

In Figure 3 we see that the forecast is highly dependent on the kernel choice. We expect that for periodic data the PER and PS kernels will forecast best, maybe with the PS kernel doing a better job because it captures multiple modes of the daily increase in volume. We use for both kernels a period of 168 hours. This is because although a daily pattern exists, the weekly is stronger, with the day of the week influencing the volume of the hashtag. The NRMSE (Normalized Root Mean Square Error) in Table 1 on the held out data confirms this finding, with PS showing the lowest error.

[4] Equations in red corrected from the published version

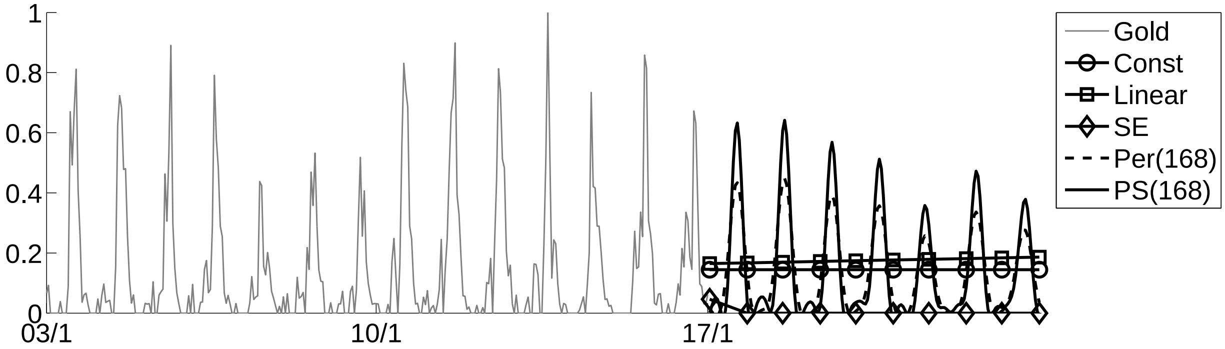

Figure 3: Extrapolation for #goodmorning over 3 weeks with GPs using different kernels.

| | Const | Lin | SE | PER | PS |
|---|---|---|---|---|---|
| **NLML** | -41 | -34 | -176 | -180 | -192 |
| **NRMSE** | 0.213 | 0.214 | 0.262 | 0.119 | 0.107 |

Table 1: Negative Log Marginal Likelihood (NLML) shows the best fitted model for the time series in Figure 3. NRMSE computed on the third unobserved week. Lower values are better in both cases.

### 3.3 Model selection and optimisation

We now briefly discuss the concepts of model selection in the GP framework, by which we refer to choosing the model (kernel) from a set $\mathcal{H}_i$ and optimising the model hyperparameters $\theta$. In our GP Bayesian inference scheme, we can compute the probability of the data given the model which involves the integral over the parameter space. This is called the *marginal likelihood* or *evidence* and is useful for model selection using only the training set:

$$p(x|\mathcal{D},\theta,\mathcal{H}_i) = \int_f p(x|\mathcal{D},f,\mathcal{H}_i)p(f|\theta,\mathcal{H}_i) \quad (7)$$

Our first goal is to fit the kernel by minimizing the negative log marginal likelihood (NLML) with respect to the kernel parameters $\theta$. This approximation is also known as type II maximum likelihood (ML-II). Conditioned on kernel parameters, the evidence of a GP can be computed analytically.

Our second goal is to use the evidence for model selection because it balances the data fit and the model complexity by automatically incorporating Occam's Razor (Rasmussen and Ghahramani, 2000). Because the evidence must normalise, complex models which can account for many datasets achieve low evidence. One can think of the evidence as the probability that a random draw of the parameter values from the model class would generate the dataset $\mathcal{D}$. This way, complex models are penalised because they can describe many datasets, while the simple models can describe only a few datasets, thus the chance of a good data fit being very low. This is for example the case of the periodic bursts in Figure 3. Although the periodic kernel can fit the data, it will incur a high model complexity penalty. The PS kernel in this respect is a simpler model and can fit the data and is thus chosen as the right model.

When the dataset is observed, the evidence can select between the models. More generally, the model choice actually gives us an *implicit classification* of the temporal patterns into classes: a steady signal with noise (C kernel), a signal with local temporal patterns (SE kernel), an oscilating periodic pattern (PER kernel) or a pattern with abrupt periodic peaks (PS kernel).

We use the NLML for optimising the hyperparameters only using training data. For optimising the hyperparameters of the kernel defined in Equation 6, it is important to first identify the right period. We consider as possible periods all integer values less than half the size of the training set, and then tune the shape parameter using gradient descent to minimise NLML. We then take the $argmin$ value of those considered. We show the NLML for a sample regression in Figure 4.

The likelihood shows that there are multiple canyons in the likelihood, which can lead a convex optimisation method to local optima. These appear when $p$ is equal or an integer multiple of the main period of the data, in this case 24. The lowest values are obtained when $p = 168$, allowing the model to accommodate the day of week effect. Our procedure is not guaranteed to reach a global optima, but

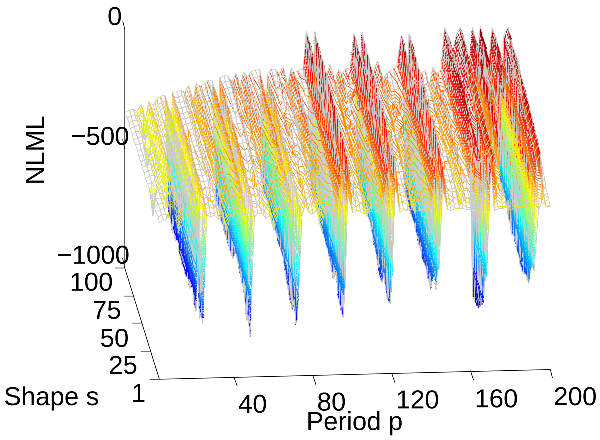

Figure 4: NLML for #goodmorning on the training set as a function of the 2 kernel parameters.

is a relatively standard technique for fitting periodic kernels (Duvenaud et al., 2013).

The flexibility of the GP framework allows us to combine kernels (e.g. $SE{\cdot}PS$ or $PS{+}Lin$) in order to identify a combination of trends (Duvenaud et al., 2013; Gönen and Alpaydin, 2011). Experiments on a subset of data showed no major benefits of combining kernels, but the computational time and model complexity increased drastically due to the extra hyperparameters. Because we will model a proportion of words within a limited time frame, there are few linear trends in the data. It might seem limiting that we only learn a single period, although we could combine periodic kernels with different periods together. But, as we have seen in the #goodmorning example (with overlapping weekly and daily patterns), if there is a combination of periods the model will select a single period which is the least common multiple.

## 4 Data

For our experiments we used data collected from Twitter using the public Gardenhose stream (10% representative sample of the entire Twitter stream). The data collection interval was 1 January – 28 February 2011. For simplicity in the classification task, we filtered the stream to include only tweets that have exactly one hashtag. These represent approximately 7.8% of our stream.

As text processing steps, we have tokenised all the tweets and filtered them to be written in English using the Trendminer pipeline (Preoţiuc-Pietro et al., 2012). We also remove duplicate tweets (retweets and tweets that had the same first 6 content tokens) because they likely represent duplicate content, automated messages or spam which would bias the dataset, as also stated by Tsur and Rappoport (2012). In our experiments we use the first month of data as training and the second month as testing. Note the challenging nature of this testing configuration where predictions must be made for up to 28 days into the future. We keep a total 1176 of hashtags which appear at least 500 times in both splits of the data. The vocabulary consists of all the tokens that occur more than 100 times in the dataset and start with an alphabetic letter. After processing, our dataset consists of 6,416,591 tweets with each having on average 9.55 tokens.

## 5 Forecasting hashtag frequency

We treat our task of forecasting the volume of a Twitter hashtag as a regression problem. Because the total number of tweets varies depending on the day and hour of day, we chose to model the proportion of tweets with the given tag in that hour. Given a time series of these values as the training set for a hashtag, we aim to predict the values in the testing set, extrapolating to the subsequent month.

Hashtags represent free-form text labels that authors add to a tweet in order to enable other users to search them to participate in a conversation. Some users use hashtags as regular words that are integral to the tweet text, some hashtags are general and refer to the same thing or emotion (#news, #usa, #fail), others are Twitter games or memes (#2010disappointments, #musicmonday). Other hashtags refer to events which might be short lived (#worldcup2022), long lived (#25jan) or periodic (#raw, #americanidol). We chose to model hashtags because they group similar tweets (like topics), reflect real world events (some of which are periodic) and present direct means of evaluation. Note that this approach could be applied to many other temporal problems in NLP or other domains. We treat each regression problem independently, learning for each hashtag its specific model and set of parameters.

### 5.1 Methods

We choose multiple baselines for our prediction task in order to compare the effectiveness of our ap-

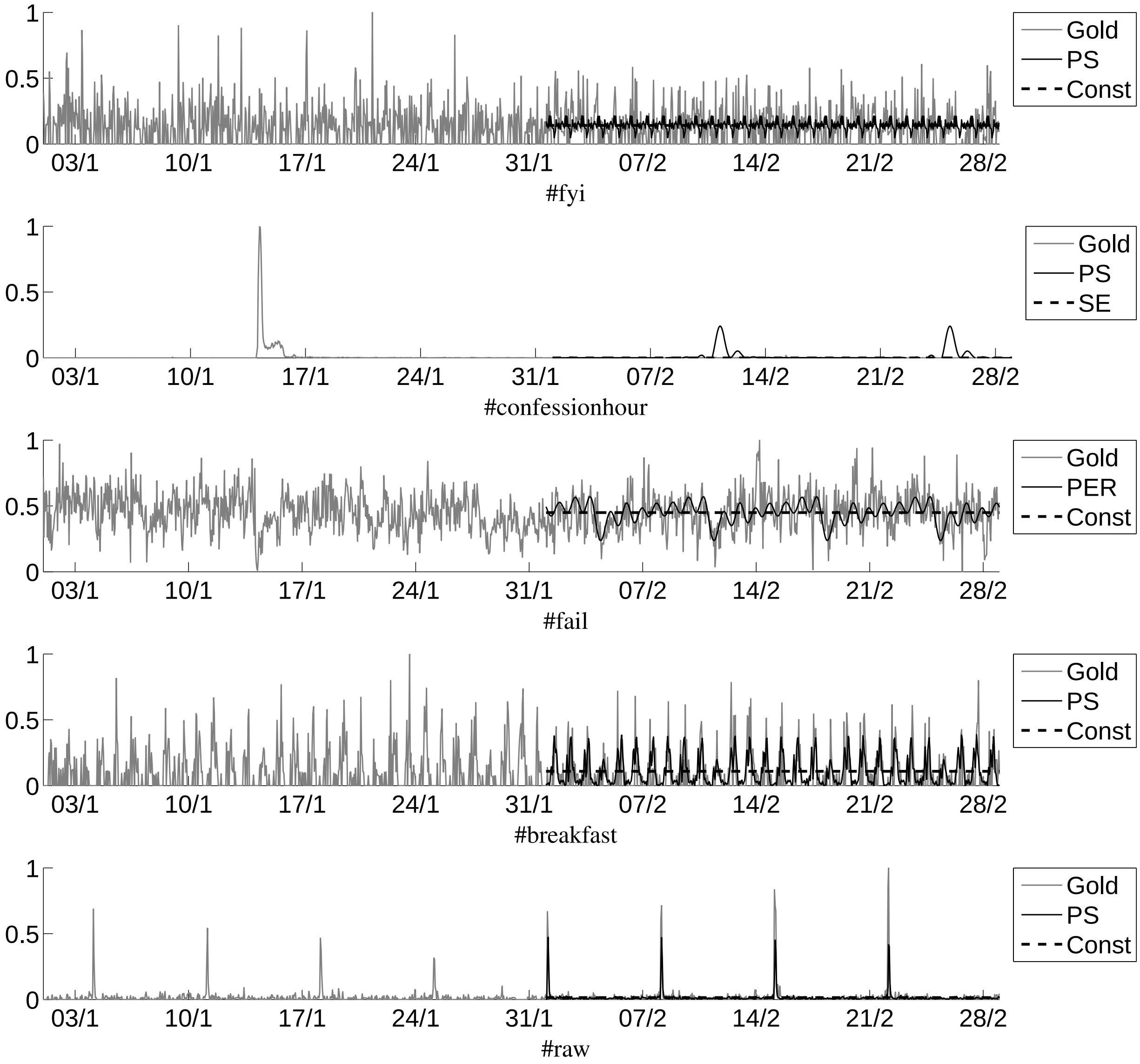

Figure 5: Sample regressions and their fit using different methods.

proach. These are:

**Mean value (M):** We use as prediction the mean of the values in the training set. Note that this is the same as using a GP model with a constant kernel (+ noise) with a mean equal to the training set mean.

**Lag model with GP determined period (Lag+):** The prediction is the mean value in the training set of the values at lag $\Delta$ where $\Delta$ is the period rounded to the closest integer as determined by our GP model. This is somewhat similar to an autoregressive (AR) model with all the coefficients except $\Delta$ set to 0. We highlight that given the period $\Delta$ this is a very strong model as it gives a mean estimate at each point. Comparing to this model we can see if the GP model can recover the underlying function that described the periodic variation and filter out the noise in the observations. Correctly identifying the period is very challenging as we discuss below.

**GP regression:** Gaussian Process regression using only the SE kernel (**GP-SE**), the periodic kernel (**GP-PER**), the PS kernel (**GP-PS**). The method that chooses between kernels using model selection as described in Section 3.3 is denoted as **GP+**. We will also compare to GP regression the linear kernel **(GP-Lin)**, but we will not use this as a candidate for model selection due the poor results shown below.

| Hashtag | **Lag(p)** | **Const** | | **SE** | | **PER** | | **PS** | |
|---|---|---|---|---|---|---|---|---|---|
| | NRMSE | NLML | NRMSE | NLML | NRMSE | NLML | NRMSE | NLML | NRMSE |
| **#fyi** | 0.1578 | **-322** | **0.1404** | -320 | 0.1898 | -321 | 0.1405 | -293 | 0.1456 |
| **#confessionhour** | 0.0404 | -85 | 0.0107 | **-186** | **0.0012** | -90 | 0.0327 | -88 | 0.0440 |
| **#fail** | 0.1431 | -376 | 0.1473 | -395 | 0.4695 | **-444** | **0.1387** | -424 | 0.1390 |
| **#breakfast** | 0.1363 | -293 | 0.1508 | -333 | 0.1773 | -293 | 0.1514 | **-367** | **0.1276** |
| **#raw** | 0.0464 | -1208 | 0.0863 | -1208 | 0.0863 | -1323 | 0.0668 | **-1412** | **0.0454** |

Table 2: NRMSE shows the best performance for forecasting and NLML shows the best model for all the regressions in Figure 5. Lower is better.

### 5.2 Results

We start by qualitatively analysing a few sample regressions that are representative of each category of time series under study. These are shown in Figure 5. For clarity, we only plotted a few kernels on each figure. The full evaluation statistics in NRMSE and the Bayesian evidence are show in Table 2.

For the hashtag #fyi there is no clear pattern. For this reason the model that uses the constant kernel performs best, being the simplest one that can describe the data, although the others give similar results in terms of NRMSE on the held-out testing set. While functions learned using this kernel never clearly outperform others on NRMSE on held-out data, this is very useful for interpretation of the time series, separating noisy time series from those that have an underlying periodic behaviour.

The #confessionhour example illustrates a behaviour best suited for modelling using the SE kernel. We notice a sudden burst in volume which decays over the next 2 days. This is actually the behaviour typical of ‘internet memes’ (this hashtag tags tweets of people posting things they would never tell anyone) as presented in Yang and Leskovec (2011). These cannot be modelled with a constant kernel or a periodic one as shown by the results on held-out data and the time series plot. The periodic kernels will fail in trying to match the large burst with others in the training data and will attribute to noise the lack of a similar peak, thus discovering wrong periods and making bad predictions. In this example, forecasts will be very close to 0 under the SE kernel, which is what we would desire from the model.

The periodic kernel best models hashtags that exhibit an oscillating pattern. For example, this best fits words that are used frequently during the day and less so during the night, like #fail. Here, the period is chosen to be one week (168) rather than one day (24) because of the weekly effect superimposed on the daily one. Our model recovers that there is a daily pattern with people tweeting about their or others’ failures during the day. On weekends however, and especially on Friday evenings, people have better things to do.

The PS kernel introduced in this paper models best hashtags that have a large and short lived burst in usage. We show this by two examples. First, we choose #breakfast which has a daily and weekly pattern. As we would expect, a big rise in usage occurs during the early hours of the day, with very few occurrences at other times. Our model discovers a weekly pattern as well. This is used mainly for modelling the difference between weekends and weekdays. On weekends, the breakfast tag is more evenly spread during the hours of the morning, because people do not have to wake up for work and can have breakfast at a more flexible time than during the week. In the second example, we present a hashtag that is associated to a weekly event: #raw is used to discuss a wrestling show that airs every week for 2 hours on Monday evenings in the U.S.. With the exception of these 2 hours and the hour building up to it, the hashtag is rarely used. This behaviour is modelled very well using our kernel, with a very high value for the shape parameter ($s = 200$) compared to the previous example ($s = 11$) which captures the abrupt trend in usage. In all cases, our GP model chosen by the evidence performs better than the Lag+ model, which is a very strong method if presented with the correct period. This further demonstrates the power of the Gaussian Process framework to deal with noise in the training data and to find the underlying function of the time variation of words.

In Table 3 we present sample tags identified as

| Const | SE | PER | PS |
|---|---|---|---|
| #funny | #2011 | #brb | #ff |
| #lego | #backintheday | #coffee | #followfriday |
| #likeaboss | #confessionhour | #facebook | #goodnight |
| #money | #februarywish | #facepalm | #jobs |
| #nbd | #haiti | #funny | #news |
| #nf | #makeachange | #love | #nowplaying |
| #notetoself | #questionsidontlike | #rock | #tgif |
| #priorities | #savelibraries | #running | #twitterafterdark |
| #social | #snow | #xbox | #twitteroff |
| #true | #snowday | #youtube | #ww |
| **49** | **268** | **493** | **366** |

Table 3: Sample hashtags for each category. The last line shows the total number of hashtags of each type.

| Lag+ | GP-Lin | GP-SE | GP-PER | GP-PS | GP+ |
|---|---|---|---|---|---|
| 7.29% | -3.99% | -34.5% | 0.22% | 7.37% | **9.22%** |

Table 4: Average relative gain over mean (M) prediction for forecasting on the entire month using the different models

being part of the 4 hashtag categories, and the total number of hashtags in each.

As a means of quantitative evaluation we compute the relative NRMSE compared to the Mean (M) method for forecasting. We choose this, because we consider that NRMSE is not comparable between regression tasks due to the presence of large peaks in many time series, which distort the NRMSE values. The results are presented in Table 4 and show that our Gaussian Process model using model selection is best. Remarkably, it consistently outperforms the Lag+ model, which shows the effectiveness of the GP models to incorporate uncertainty. The GP-PS model does very well on its own. Although chosen in the model selection phase in only a third of the tasks, it performs consistently well across tasks because of its ability to model well all the periodic hashtags, be they smooth or abrupt. The GP-Lin model does worse than the average, mostly due to uni-modal time series which don't have high occurrences in the testing part of the data.

### 5.3 Discussion

Let us now turn to why the GP model is better for discovering periodicities than classic time series modelling methods. Measuring autocorrelation between points in the time series is used to discover the hidden periodicities in the data and in building AR models. However, the downsides of this method are: a) the incapacity of accurately finding the correct periods, because all integer multiples of the correct period will be feasible candidates and b) it leads to incorrect conclusions when there is autocorrelated noise. The second case is illustrated in Figure 6 where #confessionhour shows autocorrelation but, as seen in Figure 5, lacks a periodic component.

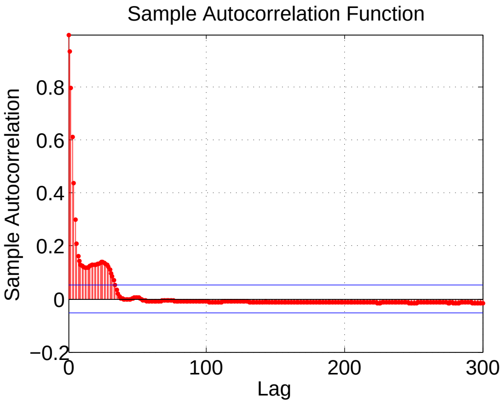

Figure 6: Sample autocorrelation for #confessionhour

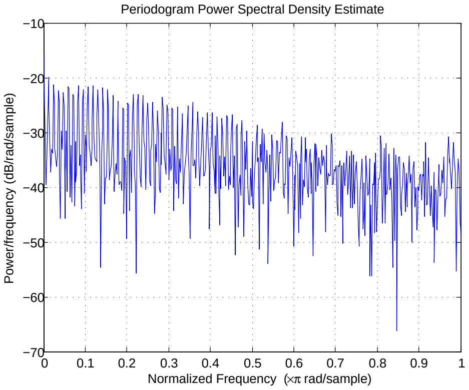

Figure 7: Power spectral density for #raw

Another approach to discovering periods in data is by computing the power spectral density. This has been used in the GP framework by Wilson and Adams (2013). For some time series, this gives a good indication of the period, as represented by a peak in the periodogram at that value. This fails to discover the correct period when dealing with large bursts like those exhibited by the #raw time series as shown in Figure 7. The lowest frequency spike corresponds to the correct period of 168, but also other candidate periods are shown as possible. The reason for this is its reliance on the Fourier Transform which decomposes the time series into a sum of oscillating patterns. These cannot model step-functions and other non-smoothly varying signals. A further discussion falls out of the scope and space constraints of this paper.

| Tweet | Time | Prior | Rank | Prediction |
|---|---|---|---|---|
| Bruins Goal!!! Patrice Bergeron makes it 3-1 Boston | 2-3am, 2 Feb 2011 | E: 0.00017<br>P: 0.00086 | 7<br>1 | #fb<br>#bruins |
| i need some of Malik people | 3-4am, 2 Feb 2011 | E: 0.00021<br>P: 0.00420 | 7<br>1 | #ff<br>#thegame |
| Alfie u doughnut! U didn't confront Kay? SMH | 7-8pm, 3 Feb 2011 | E: 0.00027<br>P: 0.00360 | 8<br>1 | #nowplaying<br>#eastenders |

Table 5: Example of tweet classification using the Naïve Bayes model with the two different priors (E - empirical, P - GP forecast). Rank shows the rank in probability of the correct class (hashtag) under the model. Time is G.M.T.

## 6 Text based prediction

In this section we demonstrate the usefulness of our method of modelling in an NLP task: predicting the hashtag of a tweet based on its text. In contrast to this classification approach for suggesting a tweet's hashtag, information retrieval methods based on computing similarities between tweets are very hard to scale to large data (Zangerle et al., 2011).

We choose a simple model for prediction, the Naïve Bayes Classifier. This method provides us with a straightforward way to incorporate our prior knowledge of how frequent a hashtag is in a certain time frame. This Naïve Bayes model (**NB-P**) uses the forecasted values for the respective hour as the prior on the hashtags.

For comparison we use the Most Frequent (**MF**) baseline and the Naïve Bayes with empirical prior (**NB-E**) which doesn't use any temporal forecasting information. Because there are more than 1000 possible classes we show the accuracy of the correct hashtag being amongst the top 1,5 or 50 hashtags as well as the Mean Reciprocal Rank (MRR). The results are shown in Table 6.

| | **MF** | **NB-E** | **NB-P** |
|---|---|---|---|
| **Match@1** | 7.28% | 16.04% | **17.39%** |
| **Match@5** | 19.90% | 29.51% | **31.91%** |
| **Match@50** | 44.92% | 59.17% | **60.85%** |
| **MRR** | 0.144 | 0.237 | **0.252** |

Table 6: Results for hashtag classification.

The results show that incorporating the forecasted values as a more informative prior for classification we obtain better predictions. The improvements are consistent in all the Match values. Also, we highlight that a 9% improvement in the forecasting task carries over to about a 2% improvement in classification. We show a few examples in which the GP learned prior makes a difference in classification in Table 5 together with the values for both priors.

With these experiments, we highlighted that there are performance gains even with only adding a more informative prior that uses periodicity information. This motivates future work to add this information to discriminative classifiers thus avoiding the need for the Naïve Bayes decomposition. The modelling framework offered by the GPs can accommodate classification, although scaling issues arise when using a large number of features or output classes. Efforts to scale GPs to a large number of variables are well understood (Candela and Rasmussen, 2005) and we will try to incorporate this in future work.

## 7 Conclusion

Periodicities play an important role when analysing the temporal dimension of text. We have presented a framework based on Gaussian Process regression for identifying periodic patterns and their parameters using only training data. We divided the periodic patterns into 2 categories: oscillating and periodic bursts by performing model selection using bayesian evidence. The periodicities we have discovered have proven useful in an NLP classification task.

In future work, we aim to model time continuously and to perform discriminative clustering in order to make better use of the learned periodicites. We will consider incorporating periodicities in other applications, such as topic models.

## Acknowledgements

This research was funded by the Trendminer project, EU FP7-ICT Programme, grant agreement no.287863. The authors would like to thank James Hensman, Nicolas Durrande and Neil Lawrence for advice on Gaussian Processes, Chris Dyer and Noah Smith for discussions about periodicities in NLP.

## References

David Blei and John Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International conference on Machine learning*, ICML '06.

Joaquin Quiñonero Candela and Carl Edward Rasmussen. 2005. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research (JMLR)*, 6:1939–1959, December.

Trevor Cohn and Lucia Specia. 2013. Modelling Annotator Bias with Multi-task Gaussian Processes: An Application to Machine Translation Quality Estimation. In *Proceedings of the Association of Computational Linguistics*, ACL '13.

Nicolas Durrande, James Hensman, Magnus Rattray, and Neil Lawrence. 2013. Gaussian Process models for periodicity detection. In *Submitted to JRSSb, http://arxiv.org/abs/1303.7090*.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. 2013. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the International Conference on Machine Learning*, ICML '13.

Mehmet Gönen and Ethem Alpaydin. 2011. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research (JMLR)*, 12:2211–2268, July.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, (Suppl 1):5228–5235, April.

Philipp Hennig, David H. Stern, Ralf Herbrich, and Thore Graepel. 2012. Kernel topic models. *Journal of Machine Learning Research (JMLR) - Proceedings Track*, 22:511–519.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International conference on Knowledge discovery and data mining*, KDD '09.

Zongyang Ma, Aixin Sun, and Gao Cong. 2012. Will this #hashtag be popular tomorrow? In *Proceedings of the 35th International ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12.

Zongyang Ma, Aixin Sun, and Gao Cong. 2013. On predicting the popularity of newly emerging hashtags in Twitter. *Journal of the American Society for Information Science and Technology*, 64(7):1399–1410.

Allie Mazzia and James Juett. 2011. Suggesting hashtags on Twitter. In *http://www-personal.umich.edu/ amazzia/pubs/545-final.pdf*.

James McInerney, Alex Rogers, and Nicholas R Jennings. 2013. Learning periodic human behaviour models from sparse data for crowdsourcing aid delivery in developing countries. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI '13.

Tamara Polajnar, Simon Rogers, and Mark Girolami. 2011. Protein interaction detection in sentences via Gaussian Processes: a preliminary evaluation. *International Journal Data Mining and Bioinformatics*, 5(1):52–72, February.

Daniel Preoţiuc-Pietro and Trevor Cohn. 2013. Mining User Behaviours: A Study of Check-in Patterns in Location Based Social Networks. In *Proceedings of the ACM Web Science Conference*, Web Science '13.

Daniel Preoţiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. 2012. Trendminer: An architecture for real time analysis of social media text. *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, Workshop on Real-Time Analysis and Mining of Social Streams*.

Carl Edward Rasmussen and Zoubin Ghahramani. 2000. Occam's razor. In *Advances in Neural Information Processing Systems*, NIPS 13.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning*. MIT Press.

Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. 2011. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20th International conference on World wide web*, WWW '11.

Kashif Shah, Trevor Cohn, and Lucia Specia. 2013. An Investigation on the Effectiveness of Features for Translation Quality Estimation. In *MT Summit '13*.

Oren Tsur and Ari Rappoport. 2012. What's in a hashtag? Content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM International conference on Web search and data mining*, WSDM '12.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International conference on Knowledge discovery and data mining*, KDD '06.

Chong Wang, David M. Blei, and David Heckerman. 2008. Continuous time Dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI '08.

Andrew Gordon Wilson and Ryan Prescott Adams. 2013. Gaussian Process covariance kernels for pattern discovery and extrapolation. In *Proceedings of the International Conference on Machine Learning*, ICML '13.

Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM International conference on Web search and data mining*, WSDM '11.

Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. 2012. We know what @you #tag: does the dual role affect hashtag adoption? In *Proceedings of the 21st International conference on World Wide Web*, WWW '12.

Dani Yogatama, Michael Heilman, Brendan O'Connor, Chris Dyer, Bryan R. Routledge, and Noah A. Smith. 2011. Predicting a scientific community's response to an article. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11.

Eva Zangerle, Wolfgang Gassler, and Gunther Specht. 2011. Recommending #-tags in twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web*, UMAP '11.