Modelling the Working Week for Multi-Step Forecasting using Gaussian Process Regression

Pasan Karunaratne, Masud Moshtaghi, Shanika Karunasekera, Aaron Harwood, Trevor Cohn
Department of Computing and Information Systems, The University of Melbourne, Australia
pkarunaratne@student.unimelb.edu.au, {masud.moshtaghi, karus, aharwood, tcohn}@unimelb.edu.au

Abstract

In time-series forecasting, regression is a popular method, with Gaussian Process Regression widely held to be the state of the art. The versatility of Gaussian Processes has led to them being used in many varied application domains. However, though many real-world applications involve data which follows a working-week structure, where weekends exhibit substantially different behaviour to weekdays, methods for explicit modelling of working-week effects in Gaussian Process Regression models have not been proposed. Not explicitly modelling the working week fails to incorporate a significant source of information which can be invaluable in forecasting scenarios. In this work we provide novel kernel-combination methods to explicitly model working-week effects in time-series data for more accurate predictions using Gaussian Process Regression. Further, we demonstrate that prediction accuracy can be improved by constraining the non-convex optimisation process of finding optimal hyperparameter values. We validate the effectiveness of our methods by performing multi-step prediction on two real-world publicly available time-series datasets - one relating to electricity Smart Meter data of the University of Melbourne, and the other relating to the counts of pedestrians in the City of Melbourne.

1 Introduction

Forecasting time series is an important problem, with applications in fields as diverse as financial markets, robotics and electricity generation. Many real-world forecasting tasks involve data which exhibit periodic structure and which relate to a Monday - Friday working week. In many scenarios, the working week contains a great deal of information which can be utilised to improve forecasting accuracy. In this work we focus on making multi-step predictions of up to 24 hours into the future, with an emphasis on methods of incorporating prior information of periodicities and effects of the working week in order to improve the accuracy of the forecasting.

Regression is a popular method to perform forecasting, for which Gaussian Process Regression [Rasmussen, 2006] is considered the state of the art. Gaussian Processes are Bayesian nonparametric models and are popular for their support for intrinsic feature selection and their versatility in modelling complex functions. They are fully defined by their underlying mean and covariance functions, and derive their versatility by the encoding of assumptions of the structure of the data through varied combinations of covariance functions.

This specification of the structure of the data through combinations of covariance functions is not a straightforward exercise. A simple formulaic specification of a common covariance function will fail to fully capture the unique characteristics of the data in most real-life scenarios. Such generic model specifications will fall short in prediction, since the prior knowledge incorporated by the covariance functions chosen in Gaussian Processes plays a major role in their prediction accuracy [Preotiuc-Pietro and Cohn, 2013].

On the other hand, though more elaborate models potentially could model the data more closely, the wide variety of combinations possible and the specificity of Gaussian Process model specifications to the given problem domain results in this model specification being done by experts. Examples of work in which the main contribution involves experts encoding structure relevant to a given problem by using kernel combinations include [Klenske *et al.*, 2013] [Preotiuc-Pietro and Cohn, 2013] and [Senanayake *et al.*, 2016]. Too often though, simple models are built which do not fully utilize the flexibility of Gaussian Processes [Hachino and Kadirkamanathan, 2007] [Kolter and Ferreira, 2011]. This is particularly apparent given the challenges posed in using more elaborate models with multiple kernels, where the high number of parameters increases computational complexity as well as the likelihood of reaching local optima in the optimisation process.

Our work focuses on effective model specification relating to not only one particular application, but to the general problem of modelling the working week in time-series data using Gaussian Processes. In this work we introduce methods to build Gaussian Processes which model complex periodic structure relating to the working week. For example, in Figure 1 we observe the working week effect of Monday-Friday (high values) against weekends (low values), and also the fact that the days in the second week have higher values than the days in the first week. We propose multiple methods to encode varied prior beliefs on periodicity, especially when such multiple periodic structure based on knowledge of the working week is to be encoded in the same model for better forecasting accuracy.

We list our contributions below.

1. We provide novel kernel-combination methods to explicitly encode weekday and weekend effects in a working week for improved accuracy in multi-step prediction. These methods are flexible enough to be easily extended to model holiday effects.
2. We suggest methods to mitigate the effects of convergence to local optima in the optimisation process over hyperparameters, which is especially important in the case of low volume of training data or complex models.
3. We illustrate the effectiveness of our approaches on two real-world time-series datasets relating to electricity consumption and the counts of pedestrians in a city.

2 Related Work

Gaussian Process Regression is a highly flexible method and has been used in a wide variety of forecasting problems. For example, [Chen *et al.*, 2013] forecasts power generation in wind farms, [Bickel *et al.*, 2008] makes predictions on the effects of combinations of drugs, and [Sturm and Burgard, 2013] [Williams *et al.*, 2009] build models to represent the kinematics and adaptive control functions to be used by robotic manipulators. Work related to robotic mapping has also been undertaken using spectral analysis to model periodic environment processes [Krajnik *et al.*, 2014]. Work with similar contributions of analysis of kernel combinations in different application domains include identifying the effects of additive errors in control systems [Klenske *et al.*, 2013], forecasting the number of tweets with a particular hashtag [Preotiuc-Pietro and Cohn, 2013], and modelling the propagation of seasonal influenza [Senanayake *et al.*, 2016]

In relation to the application domains that this work focuses on, much work has been done on the prediction of electricity load data. Some popular approaches include variations on auto-regressive methods [Conejo *et al.*, 2005] [Alzate and Sinn, 2013], and Artificial Neural Networks [Guan *et al.*, 2013]. Gaussian processes have been utilised in [Kolter and Ferreira, 2011] [Hachino and Kadirkamanathan, 2007] [Leith *et al.*, 2004], but they either do not consider the periodic structure in the data, or do not explicitly incorporate weekend and weekday information in their models. Further, prior beliefs on parameter values are not encoded. In relation to the application domain of pedestrian data, little work exists in the literature. The work in [Doan *et al.*, 2015] performs anomaly detection on pedestrian data, but to the best of our knowledge no attempt at predicting future counts has been done.

3 Introduction to Gaussian Process Regression

3.1 Gaussian Processes

A Gaussian Process is formally defined as a collection of random variables where any subset of the random variables taken together jointly form a (multivariate) Gaussian distribution. A useful intuition is to view a Gaussian Process as defining a distribution over functions (function-space view [Rasmussen, 2006].)

Since a Gaussian Process is a distribution over functions, sampling from a Gaussian Process results in the draw of a single function. It is possible to specify prior belief in the general properties we expect to see in these functions drawn (e.g. whether the function is continuous). Further, Gaussian Processes follow the Bayesian paradigm of updating prior beliefs based on observed data to form posterior distributions. In the case of Gaussian Processes, these prior and posterior distributions are distributions over functions, and therefore the Bayesian inference that takes place occurs in function space.

A Gaussian Process is fully defined by its mean function ($m(\boldsymbol{t})$) and covariance (kernel) function ($k(\boldsymbol{t},\boldsymbol{t'})$) (where $\boldsymbol{t}$ and $\boldsymbol{t'}$ are two separate input vectors). Therefore we can write the Gaussian Process as $f(\boldsymbol{t}) \sim GP(m(\boldsymbol{t}), k(\boldsymbol{t},\boldsymbol{t'}))$

In our tasks our data consists of n pairs $D = (t_i, y_i)$, where y_i is the value of the time series at time t_i. We define without loss of generality a *prior* mean function of 0 and covariance function $k(t, t')$, with the resulting general form of the Gaussian Process $f(t) \sim GP(0, k(t, t'))$.

3.2 Regression

Our goal is to predict values y_* at time t_* given training data D. We assume a latent function f, which provides the values for each data point according to $y_t = f(t) + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$ is Gaussian noise. Noting that we perform Bayesian inference with regards to functions f, we can write the posterior predictive distribution as follows:

$$p(y_*|t_*, D) = \int_f p(y_*|t_*, f) \cdot p(f|D) \quad (1)$$

In the case of Gaussian Processes, we are able to solve for this predictive posterior *analytically* with the solution given below:

$$y_* \sim N(k_*^T(K + \sigma^2 I)^{-1}\boldsymbol{y}, \\ k(t_*, t_*) - k_*^T(K + \sigma^2 I)^{-1}k_*) \quad (2)$$

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 covariance matrix formed by the evaluations of the kernel function between all pairs of training points, and $\boldsymbol{y}$ is the vector of training outputs.

Therefore, it can be seen that the posterior includes both the mean and the variance of the response at the required prediction time points. In this paper the expected value (i.e. mean) of the response is considered as the forecast of the prediction.

3.3 Covariance (Kernel) Functions

Covariance (kernel) functions define the similarity between any two input points. That is, a kernel function outputs a real-valued similarity score for any given pair of input points. Evaluating the kernel function over all pairs of input points results in a conventional covariance matrix.

The choice of kernel function determines the overall structure of the functions drawn from the Gaussian Process. For example, consider the popular Squared Exponential (SE) kernel function defined below:

$$k_{SE}(t,t') = \sigma^2 \exp\Big(-\frac{(t-t')^2}{2l^2}\Big) \tag{3}$$

A kernel function defines the covariance or similarity between pairs of points. In this example, when the two points t and t' are close to each other they will have high covariance compared to points far apart. This encoding of higher similarity to points closer to each other results in smooth functions being drawn from the Gaussian Process defined by this kernel. We are further able to change the properties of these smooth functions by changing the values of the hyperparameters σ (output variance) and l (length scale).

3.4 Combinations of Kernel Functions

Combinations of kernel functions can be used to model a number of effects together. For example, a squared exponential kernel coupled with a periodic kernel can induce a recency effect to the periodic kernel.

Kernels may be combined by either addition or subtraction. The addition of two kernels can be thought of as a logical OR operation, in that the final value of the addition will be high if either one of the two kernels being added outputs a high value. Similarly, the multiplication of two kernels is similar to a logical AND operation, where the final value is high only if both the two base kernels output a high value.

3.5 Optimisation over Hyperparameters

The nature of the functions drawn from the Gaussian Process depends on the type of kernel chosen, as well as the values of the kernel hyperparameters. These hyperparameters are optimised to fit the training data.

The marginal likelihood of a model provides a measure of how likely the data was generated by the given model. Therefore, maximising the marginal likelihood of the model would improve the fit of the model to the data. In Gaussian Processes, it is possible to formulate an analytic solution to the marginal likelihood. Therefore, the Type II maximum likelihood estimate of the marginal likelihood is found by using gradient ascent with respect to the model hyperparameters in the training process. It is important to note that the marginal likelihood is non-convex in the hyperparameter values, and therefore the optimisation process is at risk of converging to local optima.

4 Methodology

We detail the process we followed in building the Gaussian Process models. This includes the choice of the nature of the covariance functions, the setting of the hyperparameter values of the chosen covariance functions, specifying hyperpriors on the hyperparameters to control the optimisation process, and finally our methods to explicitly model the working-week characteristics in the data.

4.1 Choice of Kernel Function

We use periodic kernel functions in this work, since our focus is on time-series data that exhibit some degree of periodicity.

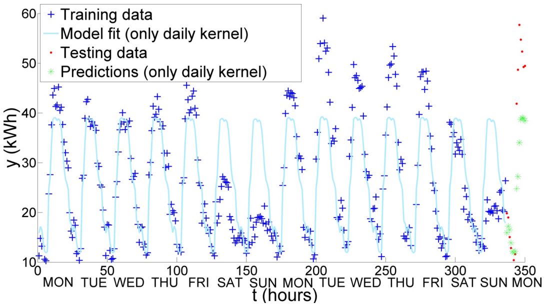

Figure 1: Modelling using only daily kernel on Smart Meter data

Periodic kernels allow the modelling of functions that repeat themselves exactly. The distance between the repetitions can be set by setting the periodicity hyperparameter in the kernel to the required value.

The periodic kernel is defined as:

$$k_{Per}(t,t') = \sigma^2 \exp\Big(-\frac{2\sin^2(\pi \mid (t-t') \mid /p)}{l^2}\Big) \tag{4}$$

where σ is the output variance and l is the length-scale (just as in the Squared-Exponential kernel), and p is the periodicity hyperparameter.

For example, if we have data sampled at an hourly frequency, and we wish to enforce a daily repetition structure in the functions drawn from the Gaussian Process, we can achieve this by choosing a periodic kernel and setting its periodicity hyperparameter value to 24. Similarly, we can choose a periodic kernel with a periodicity value of 168 (i.e. 24*7) to encode a weekly repetition structure in data sampled hourly.

We investigate periodic kernels with periodicity values set to both 24 (referred to as the 'daily kernel') and 168 (referred to as the 'weekly kernel'). We utilise plain-vanilla versions of these kernels for prediction, as well as more elaborate combinations (Section 4.4) to model periodic effects that are more complicated than simple repetition, with one or more of these periodic kernels in the mix.

4.2 Setting of Hyperparameters

The optimal values of hyperparameters which effectively model the given time series are found by minimising the *negative* log marginal likelihood (NLML). However, given that the marginal likelihood objective function is commonly non-convex, this optimisation procedure is sensitive to the initial starting values of the hyperparameters provided. Therefore, different starting values for the hyperparameters were experimented with, along with different numbers of iterations for the optimisation process. Reasonable initial values were found by using a manual grid-search-like process. The initial parameter values chosen are given in Table 1.

4.3 Specifying a Hyperprior

We observed that the parameter values arrived at after optimisation of the initial parameter values did not necessarily turn out to be values which would help reduce prediction error. This was especially so in the case of the periodicity, which was 'tuned' to values far away from 24 (daily) or 168

Table 1: Initial hyperparameter values chosen

Hyperparameter	Initial Value
Length Scale	0.03
Periodicity	24 or 168
Output Variance	0.1
ARD Bias a_1 and b_2	0.9
ARD Bias a_2 and b_1	0.1
Noise Variance	0.01

(weekly) set. However, visual inspection of the time-series plots as well as knowledge of the problem domain gives us strong confidence that the periodicities should indeed be either 24 or 168. Therefore, we decided to encode our strong prior belief by putting a prior over the hyperparameter values (a hyperprior), in this case by clamping the value at either 24 or 168 and preventing the optimisation process from modifying these values.

4.4 Modelling Combined Weekday and Weekend Effects

It is frequently required to model daily and weekly variations in a single model. For example, most data exhibits correlations on a weekly periodicity, in that the values for Wednesday this week would be highly correlated to that of the Wednesday last week. In addition, especially in cases where the response variable values might have different levels in different weeks, the values of recent previous days would carry valuable information.

Naive approach

A naive approach to model both these correlations would be to combine two periodic kernels, one with a daily periodicity and the other with a weekly periodicity using either addition or multiplication of the kernel functions. However, this poses the problem of Mondays being correlated to Sundays and Saturdays being correlated to Fridays. We propose a method to solve the problem of combining periodicities elegantly, and with additional scope for more flexible models.

New features

We enhance the feature representation of the time series as follows. Represent the current representation as *(t, y)*, where *t* is time and *y* is the corresponding output value. We augment this representation with two features which represent whether the date *t* is a weekday *(w)* or a weekend *(n)*, resulting in the representation *(t, w, n, y)*. For example, a value of 554.2 on Monday at 11 am might be represented by (143, 1, 0, 554.2) and a value of 742.1 on Sunday at 10 pm might be (183, 0, 1, 742.1).

Consider two days represented by *(t, w, n, y)* and *(t', w', n', y')*. The dot product *w.w'* would be 1 if both the days *t* and *t'* are weekdays, and would be 0 if either of the days are weekends. Conversely, considering the dot product *n.n'*, this would be 1 if both days are weekends, and 0 if either of the days are weekdays. We are able to calculate dot products by utilising linear kernels, which enables the elegant expression of the mathematical operation in terms of kernel functions. For example, $covLIN(w, w')$ is equivalent to the dot product between w and w' $(w \cdot w')$.

Kernel combinations

Now consider the expression:

$$k(t,t') = covLIN(w,w')covPer_{daily}(t,t') + covLIN(n,n')covPer_{weekly}(t,t') \quad (5)$$

It can be seen that the values generated by $covPer_{daily}$ will be activated only when both t or t' are weekdays (i.e. $w = w' = 1$), and the values generated by $covPer_{weekly}$ will be activated only when both t or t' are weekends (i.e. $n = n' = 1$). Therefore, scenarios similar to Fridays being considered correlated to Saturdays will not arise. This simplest example of combining the two kernels with different periodicities essentially provides a switching mechanism between either $covPer_{daily}$ or $covPer_{weekly}$ based on the two dates t or t' under consideration.

More elaborate and flexible models may also be constructed. For example, we are able to provide a configurable level of influence of each kernel function as follows:

$$k(t,t') = (a_1 covLIN(w,w') + b_1 covLIN(n,n'))covPer_{daily}(t,t') + (a_2 covLIN(w,w') + b_2 covLIN(n,n'))covPer_{weekly}(t,t') \quad (6)$$

In this expression the weights b_1 and a_2 would be close to 0 to represent the low level of influence we desire from the daily kernel on weekends and the weekly kernel on weekdays. What is noteworthy is that we do not need to completely discard one type of kernel, but can instead weight the kernels according to our requirements.

This method can be equivalently expressed using a linear Automatic Relevance Determination (ARD) kernel, which provides the weighted sum of the outputs of the regular linear kernels when applied separately on each dimension.

$$k(t,t') = covLINard([w;n],[w';n'])covPer_{daily}(t,t') + covLINard([w;n],[w';n'])covPer_{weekly}(t,t') \quad (7)$$

This allows the use of the regular optimisation mechanism to tune the weights in accordance with our data.

A further variation on this theme is the expression:

$$k(t,t') = covLIN(w,w')covPer_{daily}(t,t') + covPer_{weekly}(t,t') \quad (8)$$

which prevents the weekly periodic kernel being suppressed to 0 for weekdays, but does so for the daily periodic kernel for weekends.

This model is also further extensible to seamlessly model public holidays to the model with a feature representing whether the date is a public holiday which could be combined with a yearly periodic kernel.

5 Experiments and Results

5.1 The Data

The data used for electricity load forecasting comes from Smart Meters installed in the Parkville campus of the University of Melbourne[1]. This data consists of the electricity

[1] http://sustainablecampus.unimelb.edu.au

load recorded in kWh at each point in time in the Smart Meters of 21 buildings. The data is available at a granularity of 15-minutes, which was aggregated to be hourly data. We perform 24-hour ahead predictions of the electricity load values in each of the 21 buildings.

The pedestrian data set was obtained through the Open Data initiative of the City of Melbourne[2]. The data is the output of a 24-hour system which monitors pedestrian movement at key locations in Melbourne, Australia and provides hourly pedestrian counts for each day. We use our methods to perform 24-hour ahead predictions of pedestrian counts at 10 key locations in the City of Melbourne.

5.2 The Error Metric

We use as our metric the mean absolute error normalised by the average magnitude of the actual load / prediction count values for the prediction period, and expressed as a percentage. The normalisation step is required to compare error rates between different buildings / sensors, which would have different levels of electricity usage / pedestrian counts. In general form this metric (Mean Error Relative to $\overline{y}$ - MER) is expressed as follows:

$$Error(MER) \quad = \quad 100 \;\cdot\; \frac{1}{N} \sum_{h=1}^{N} \frac{|\hat{y}_h - y_h|}{\overline{y}} \qquad (9)$$

where $\hat{y}_h$ is the predicted value at hour h, y_h is the actual value at hour h, and $\overline{y}$ is the mean consumption in the period considered (a day in this context), and N is the number of hours predicted (24 in this scenario).

5.3 The Experimental Setup

All models discussed were implemented using the GPML Matlab Toolbox[3]. Initial hyperparameter values were set to values that were observed to provide reasonable prediction accuracy and did not result in pathological cases. These values are outlined in Table 1. Hyperparameter optimisation was done via conjugate gradient ascent on the log marginal likelihood function with a maximum iteration limit of 100. Predictions were made for the month of June 2014 for the electricity load forecasting data and the month of October 2016 for the pedestrian forecasting with two weeks of data used for training.

We also compare the results of Gaussian Process Regression with ARIMA, which is the most widely-used forecasting technique based on regression. The Box-Jenkins methodology was used in the tuning of parameters, which resulted in an ARIMA(1,1,0) (number of time lags, degree of differencing and order of moving average model respectively) model being selected.

5.4 The Results

Periodic kernels

We begin our exploration with a periodic daily kernel. However, as we see in Table 2 the daily kernel does not provide us with an acceptable degree of prediction accuracy. One reason for this low prediction accuracy is that there is not much of a daily signal in the data. Further, just using a daily kernel means that the function is constrained to have a constant pattern across weekends and weekdays, as well as across weeks that may have different value levels. For example, we see in Figure 1 where the day to be predicted was a Sunday (actual values in red, prediction output in green), that the prediction was significantly inaccurate. Therefore this low accuracy is unsurprising. It is interesting to observe, even in this case, that adding a hyperprior helps with accuracy, providing notable improvements in accuracy especially for the campus Smart Meter dataset.

We observe better accuracy with the use of a periodic kernel with a weekly periodicity, and much better accuracy when the weekly kernel is coupled with a hyperprior. This agrees with our intuition that this kind of data will have a strong weekly signal. The weekly kernel with the hyperprior also turns out to be the simplest model that improves on the predictive accuracy of the ARIMA model.

Adding hyperpriors

It can be seen in Table 2 that adding a hyperprior improves accuracy drastically in all the combinations under consideration. Especially in the case of pedestrian count prediction, it is seen that when coupled with a hyperprior the use of the weekly kernel in place of the daily kernel provided a very substantial improvement. This could be attributed to the fact that the pedestrian count data has even less of a daily periodicity compared to the Smart Meter data. Further, fixing some parameters reduces the search space and makes it more likely that good values are found for the parameters being optimised over.

Figures 2- 5 show the accuracy changes across a range of parameter values, which vary across a number of orders of magnitude, when predicting with and without using a (hyper)prior on the periodicity hyperparameter. It is seen that the prediction accuracy is significantly better when using a hyperprior across all parameter values. This is true regardless of the dataset or the hyperparameter being varied.

Further, it is worth noting that in most cases, even the parameter values that yield the worst accuracy when used with a hyperprior are still better or close to the level of accuracy gained with the hyperparameter values that perform best without using a hyperprior. For example, in campus dataset, the absolute worst prediction accuracy using a hyperprior is 22.35%, compared with the absolute best prediction accuracy without using a hyperprior being 19.31%.

Modelling weekday and weekend effects

Our initial attempt of modelling weekday and weekend effects was Combination I:

$$k(t,t') = \mathit{covLIN}(w,w')\mathit{covPer}_{\mathit{daily}}(t,t') + \\ \mathit{covLIN}(n,n')\mathit{covPer}_{\mathit{weekly}}(t,t')$$

It is seen in Table 2 that this combination results in a noticeable improvement in the electricity load prediction scenario, but not so for the prediction of pedestrian counts. This can be attributed to the fact that the pedestrian count dataset

[2] http://www.pedestrian.melbourne.vic.gov.au/

[3] http://www.gaussianprocess.org/gpml/code/

Table 2: Error rates for different kernel combinations

Kernel Combination	Smart Meter Error (MER)	Pedestrian Count Prediction Error (MER)
Daily	36.05	72.63
Weekly	22.10	63.65
Daily with Hyperprior	17.12	70.30
Weekly with Hyperprior	9.33	17.98
Daily*LIN + Weekly*LIN (*Combination I*)	**8.73**	20.68
Daily*LIN + Weekly (*Combination II*)	9.26	**16.53**
Daily*LINard + Weekly*LINard (*Combination III*)	9.06	18.29
ARIMA	16.40	40.11

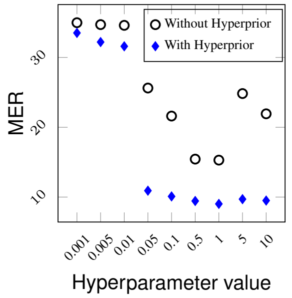

Figure 2: Campus Smart Meter - Output Variance Hyperparameter

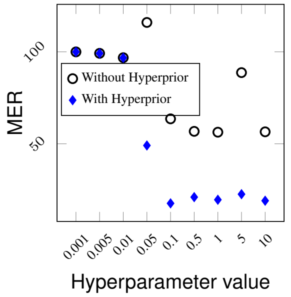

Figure 3: Pedestrian - Output Variance Hyperparameter

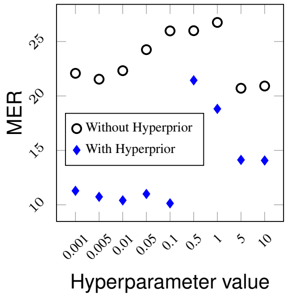

Figure 4: Campus Smart Meter - Noise Hyperparameter

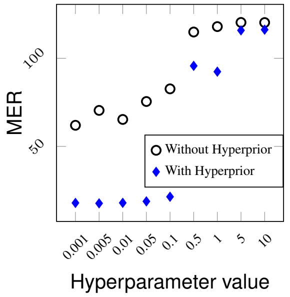

Figure 5: Pedestrian - Noise Hyperparameter

has a strong weekly signal, both for weekdays as well as for weekends. The Combination I model forces the weekly kernel to zero for weekdays, which prevents the modelling of the weekly effect for weekdays.

We therefore modified the model specification to Combination II:

$$k(t,t') = covLIN(w,w')covPer_{daily}(t,t') + covPer_{weekly}(t,t')$$

which prevents the forcing to zero of the weekly kernel for weekdays (i.e. the weekly kernel is always used in the mix) while still allowing the separate modelling of weekdays and weekends. This modified model specification resulted in an improvement in performance as seen in Table 2 for the pedestrian count prediction. This is in line with our previous observation where the pedestrian counts had a high weekly periodic signal compared to the daily signal. This model allows us to utilise the weekly periodicity in the prediction of week days as well, which is what would have resulted in the increased accuracy.

The model specification in Combination III:

$$k(t,t') = covLINard([w;n],[w';n'])covPer_{daily}(t,t') + covLINard([w;n],[w';n'])covPer_{weekly}(t,t')$$

allows the setting of relative weights for the daily and weekly kernels. Given that this model specification would not set the kernels to exactly zero or one, and given the strong weekly signal present in both datasets under consideration, it is unsurprising that the accuracy of this model is between the Combination I and Combination II models for both datasets. This version of the model would be most useful in a dataset which might have varying daily and weekly signal strengths for weekends and weekdays.

Other observations

For all the kernel combinations discussed above, experiments were run with and without adding a noise kernel to the combination. Adding a noise kernel did not have any significant effect. Further, though the product of a squared exponential kernel with a periodic kernel would result in a recency effect being modeled along with the periodic effect, we did not observe improvements in prediction accuracy. This could be attributed to the modeled recency effect not being significant enough to yield prediction accuracy gains, especially when burdened with the need to now optimise over a higher number of hyperparameters.

6 Conclusions

We provide multiple mechanisms to model the weekend and weekday effects in periodic time series data using Gaussian Processes. Our results indicate significant improvements when hyperpriors are introduced to both daily and weekly kernels. Further improved predictions were observed using mechanisms introduced to model daily and weekly periodic kernels in concert. All results were obtained on two real-world publicly available datasets with widely different applications. This is highly suggestive of the potential generalisation of the methods to other use cases with time series data with similar working-week structure.

References

[Alzate and Sinn, 2013] Carlos Alzate and Mathieu Sinn. Improved electricity load forecasting via kernel spectral clustering of smart meters. In *2013 IEEE 13th International Conference on Data Mining*, pages 943–948. IEEE, 2013.

[Bickel *et al.*, 2008] Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on Machine learning*, pages 56–63. ACM, 2008.

[Chen *et al.*, 2013] Niya Chen, Zheng Qian, Ian T Nabney, and Xiaofeng Meng. Short-term wind power forecasting using gaussian processes. In *IJCAI*, 2013.

[Conejo *et al.*, 2005] Antonio J Conejo, Miguel A Plazas, Rosa Espinola, and Ana B Molina. Day-ahead electricity price forecasting using the wavelet transform and arima models. *IEEE transactions on power systems*, 20(2):1035–1042, 2005.

[Doan *et al.*, 2015] Minh Tuan Doan, Sutharshan Rajasegarar, Mahsa Salehi, Masud Moshtaghi, and Christopher Leckie. Profiling pedestrian distribution and anomaly detection in a dynamic environment. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1827–1830. ACM, 2015.

[Guan *et al.*, 2013] Che Guan, Peter B Luh, Laurent D Michel, Yuting Wang, and Peter B Friedland. Very short-term load forecasting: wavelet neural networks with data pre-filtering. *IEEE Transactions on Power Systems*, 28(1):30–41, 2013.

[Hachino and Kadirkamanathan, 2007] Tomohiro Hachino and Visakan Kadirkamanathan. Time series forecasting using multiple gaussian process prior model. In *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 604–609. IEEE, 2007.

[Klenske *et al.*, 2013] Edgar D Klenske, Melanie N Zeilinger, Bernhard Schölkopf, and Philipp Hennig. Nonparametric dynamics estimation for time periodic systems. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 486–493. IEEE, 2013.

[Kolter and Ferreira, 2011] J Zico Kolter and Joseph Ferreira. A large-scale study on predicting and contextualizing building energy usage. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[Krajnik *et al.*, 2014] Tomas Krajnik, Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett. Spectral analysis for long-term robotic mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3706–3711. IEEE, 2014.

[Leith *et al.*, 2004] Douglas J Leith, Martin Heidl, and John Ringwood. Gaussian process prior models for electrical load forecasting. *Probabilistic Methods Applied to Power Systems*, pages 112–117, 2004.

[Preotiuc-Pietro and Cohn, 2013] Daniel Preotiuc-Pietro and Trevor Cohn. A temporal model of text periodicities using gaussian processes. In *EMNLP*, pages 977–988, 2013.

[Rasmussen, 2006] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[Senanayake *et al.*, 2016] Ransalu Senanayake, Simon O'Callaghan, and Fabio Ramos. Predicting spatio–temporal propagation of seasonal influenza using variational gaussian process regression. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Sturm and Burgard, 2013] Jürgen Sturm and Wolfram Burgard. Learning probabilistic models for mobile manipulation robots. In *IJCAI*, 2013.

[Williams *et al.*, 2009] Christopher Williams, Stefan Klanke, Sethu Vijayakumar, and Kian M Chai. Multi-task gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems*, pages 265–272, 2009.