

Electricity Demand Forecasting using Gaussian Processes

Manuel Blum and Martin Riedmiller

University of Freiburg, Department of Computer Science
Georges-Koehler Allee 079
79110 Freiburg, Germany

Abstract

We present an electricity demand forecasting algorithm based on Gaussian processes. By introducing a task-specific, custom covariance function k_{power} , which incorporates all available seasonal information as well as weather data, we are able to make accurate predictions of power consumption and renewable energy production. The hyper-parameters of the Gaussian process are optimized automatically using marginal likelihood maximization. There are no parameters to be specified by the user. We evaluate the prediction performance on simulated data and get superior results compared to a simple baseline method.

Introduction

Electricity demand forecasting is an important aspect of the control and scheduling of power systems. Demand predictions ensure the reliability of supply, but are also a vital tool for retail brokers, buying and selling electricity on modern, liberalized wholesale markets. Energy is typically traded on future markets, allowing brokers to buy and sell quantities of energy for future delivery. Therefore, the broker needs to know in advance how much electricity will be needed at a later point in time. In electricity markets, supply and demand have to be perfectly balanced. If a broker fails to keep a neutral net energy budget, the Distribution Utility (DU) exercises capacity controls at prices that are normally much less attractive than the prices in the wholesale market.

The electricity demand is mainly influenced by *meteorological conditions* and daily and weekly *seasonal effects*. Nevertheless, short term forecasting for timeframes of a day or less, is often done using only univariate prediction models. This is justified with the fact that weather conditions tend to change smoothly and therefore should already be reflected in the load data (Taylor 2003). A range of univariate methods have been successfully applied to electricity demand forecasting, such as ARIMA models, exponential smoothing or neural networks. For an extensive review and comparison see (Taylor, Menezes, and McSharry 2006).

Figure 1 shows the simulated electricity demand of different consumers as well as renewable energy providers over a



Figure 1: Two weeks of energy usage data for different customer types in a simulated environment.

two week period. A daily seasonality is immediately apparent in all of the energy profiles even the wind turbines. But clearly not all effects can be explained using only this information. Less prominent, but still noticeable, is a weekly seasonality in the consumers, which is obviously not present in the producers. The producers mainly depend on current weather conditions, like cloud cover or wind speed. Weather information is widely available through weather reports and weather predictions and an advanced prediction algorithm should be able to utilize this information, in order to provide the best forecasts possible.

In this paper, we present a prediction algorithm based on *Gaussian processes* (GP), a powerful non-parametric machine learning method for regression. GPs have been successfully applied to tasks like prediction of atmospheric CO_2 concentration (Rasmussen and Williams 2006) and model learning in robotics (Nguyen-Tuong, Seeger, and Peters 2009). We adapt Gaussian processes to electricity demand forecasting by formulating the task as a regression problem: Fit a function f that approximates the electricity demand dependent on time t and additional features z using

observed data from the past:

$$f(t_1 \dots t_T, z_1 \dots z_T, f_1 \dots f_{T-1}) \approx f_T \quad (1)$$

Our method is similar to the work of (Leith, Heidl, and Ringwood 2004), who have been using Gaussian processes for electricity load forecasting before. However, our method is able to incorporate additional available information and automatically adapt to specific customer profiles. No parameters need to be specified, making the approach easily applicable to more complex domains.

Gaussian Processes

The forecasting problem in Eq. 1 is modeled by a collection of jointly Gaussian distributed random variables, one for each time-slot. The covariance of the random variables is specified by a parametrized covariance function $k(\mathbf{x}, \mathbf{x}')$. For notational simplicity the mean of the process is set to zero. In order to make predictions about future time-slots, the joint distribution of the training outputs \mathbf{f} and the test output \mathbf{f}_* (Eq. 2) is conditioned on the observations and the expected value is obtained according to Eq. 3.

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \quad (2)$$

$$\mathbb{E}[\mathbf{f}_* | X, X_*, \mathbf{f}] = K(X_*, X)K(X, X)^{-1}\mathbf{f} \quad (3)$$

X denotes the matrix of training inputs, X_* is the matrix of test inputs and K denotes the kernel matrix which is obtained by pairwise evaluation of the covariance function for the given inputs. If we use the shorthand notation $\mathbf{k}_* = K(X_*, X)$ and assume that we are only able to observe noisy measurements $\mathbf{y} = \mathbf{f}(x) + \epsilon$ of the true function values \mathbf{f} , we can write Eq. 3 as

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} = \mathbf{k}_*^T \boldsymbol{\alpha} \quad (4)$$

where σ_n^2 is the variance of *iid* Gaussian noise ϵ (Rasmussen and Williams 2006). The computation of the kernel matrix inverse in Eq. 4 is an $\mathcal{O}(n^3)$ operation, but once the inverse is known, predictions can be made in linear time using a simple dot product. If more training data is added, the inverse can be updated using the Woodbury matrix identity, which is much faster than recomputing and inverting the entire kernel matrix (Hager 1989). This is especially useful in a time forecasting framework, where each data point is added to the training data as it is observed in order to improve future predictions.

Covariance Functions

The covariance function encodes prior beliefs about the function to be learned. It is often defined in terms of the distance $r = |\mathbf{x} - \mathbf{x}'|_2^2$ between input points, since function values are expected to be similar if their inputs are near to each other. Every kernel function that gives rise to a positive definite kernel matrix qualifies for modeling the covariance of the Gaussian process. The most frequently used kernel function is the squared exponential covariance function

$$k_{SE}(r) = \sigma_f^2 \exp\left(-\frac{r}{2l^2}\right) \quad (5)$$

with noise term σ_f^2 and characteristic length scale l . The characteristic lengthscale controls the distance from which on two points are uncorrelated and thereby influences the impact of the input patterns. In order to incorporate information about the periodicity of power usage, we do not use the time index directly as input to the covariance function, but a mapping to a two-dimensional input given by

$$\mathbf{x}(t) = \begin{bmatrix} \cos(g(t)) \\ \sin(g(t)) \end{bmatrix}, \quad g(t) = \frac{2\pi t}{p} \quad (6)$$

resulting in a covariance function for a periodic function with period p (MacKay 1998).

We use the fact that the sum of two kernels is also a kernel, to construct a special covariance function for power usage forecasting:

$$k_{power} = k_{daily} + k_{weekly} + k_{weather} \quad (7)$$

k_{daily} and k_{weekly} are periodic squared exponential covariance functions as defined in Eq. 5 and 6, with $p_1 = 24 = 168$ and $p_2 = 7 \cdot 24$ respectively. They model the daily and weekly correlations between power usages, that are present in most customer characteristics.

$k_{weather}$ is a sum of four squared exponential covariance functions (Eq. 5), which receive as an input the temperature, wind speed, wind direction and cloud cover respectively. It models the influence of different weather conditions on power usage. This covariance function could be easily extended to more complex features, if more information was available.

Hyper-parameter Optimization

With k_{power} we are able to integrate all available information into the power usage model. Depending on the hyper-parameters of this kernel function, we will obtain predictions that are correlated with already observed values that happened recently, at the same time on another day, at the same time and day of another week or those with similar weather conditions. However, there are different customer characteristics and not all the data is informative for each of them. The influence of different features on power usage is defined by the hyper-parameters of the covariance function, which can be learned by marginal likelihood optimization. The log-marginal likelihood is defined as

$$\log p(\mathbf{y}|X, \theta) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \quad (8)$$

where the first term is the data-fit, the second term is a complexity penalty and the last term is a normalizing constant with n being the number of training samples.

Since the kernel k_{power} is differentiable the gradient of the marginal likelihood with respect to the hyper-parameters θ can be computed. We use Rprop, a fast and accurate gradient-based optimization technique originally designed for neural network learning, to maximize the objective function in Eq. 8 (Blum and Riedmiller 2013).

Besides providing a good trade-off between data-fit and model complexity, the obtained hyper-parameters also allow for a straightforward interpretation of the data. For customers that are not affected by wind speed for example,

Algorithm 1: Prediction algorithm

Input: training data (X, \mathbf{y}) , time t , weather report z_t , weights α , hyper-parameters θ, σ_n^2

Output: prediction \bar{f}_t

- 1 $\mathbf{x}_* \leftarrow [t, z_t]$;
 - 2 $\mathbf{k}_* \leftarrow K_{power}(\mathbf{x}_*, X)$;
 - 3 $\bar{f}_t \leftarrow \mathbf{k}_* \alpha$;
-

the corresponding length scale parameter will go to infinity, thereby effectively removing the influence of wind from the prediction.

The Prediction Algorithm

The prediction procedure is summarized in Algorithm 1. Given some initial training inputs X , corresponding observed power usages \mathbf{y} , the weights $\alpha = (K + \sigma_n^2 I)^{-1} \mathbf{y}$, hyper-parameters θ and σ_n^2 , Algorithm 1 predicts the power usage for time t by first incorporating the time information and weather information into one input vector \mathbf{x}_* . Then the kernel function k_{power} is evaluated between \mathbf{x}_* and all training inputs X and values are aggregated in \mathbf{k}_* . Obviously, when predicting power usage for a future time-slot, the real weather conditions are not known yet and weather predictions have to be used instead.

If a new data point is available, the real weather conditions at that time are observed and the aggregated training pattern is appended to the training set. Afterwards $(K + \sigma_n^2 I)^{-1}$ is updated accordingly. In regular intervals the hyper-parameters are retrained using marginal likelihood optimization. Note, that after changing the hyper-parameters the entire kernel matrix has to be recomputed and inverted.

The PowerTAC Competition

We evaluate our approach using data from the Power Trading Agent Competition (PowerTAC) (Ketter et al. 2012). PowerTAC is a competitive simulation, where agents offer energy services to customers through tariff contracts. Balancing the energy production and consumption of the subscribed customers is achieved by placing orders on a wholesale market. In order to do this successfully, a trading agent needs access to accurate predictions of future energy demands.

Customer models include households and a variety of commercial and industrial entities, which either produce or consume energy. Each time step, the power consumption and production of the customers is reported and brokers have the chance to place orders to buy or sell electricity for future delivery.

There is a sample implementation of a PowerTAC broker available for download, which already implements the most important parts of a retail broker¹. We use the provided electricity demand algorithm as a baseline algorithm

¹<https://github.com/powertac/>

for our experiments. The PowerTAC default broker divides a week into 168 time-slots, one for each hour and uses exponential smoothing of previously observed values from the same time-slot for training the predictor. Forecasts for time t are done by simply looking up the corresponding entry in a table. While this prediction method, which will be denoted *baseline* method for the remainder of this paper, is only able to reflect weekly seasonality, its computation is efficient and predictions can adapt to changing conditions.

Experiments

We simulated four weeks of a typical PowerTAC game to evaluate our prediction algorithm. The tariff market was completely ignored, i.e. we are making predictions for the entire customer population. We trained one model for each customer using the first two weeks of data, which is exactly the same data accessible to agents before a tournament game starts. For the remaining two weeks of data, we first perform predictions from one up to 24 hours into the future and add the observed data point to the training set before moving to the next time-slot. We evaluated two different GP models, one using all available information and one using only time information while discarding weather information. Different classes of customers are evaluated separately.

Figure 2 compares the R^2 measures of the Gaussian process models to the baseline method for the different prediction horizons. Both GP methods perform better than the baseline method for all customer profiles. As expected, the differences in performance are most pronounced for the renewable energy producers, since solar and wind production cannot be reliably predicted without using weather information. For these scenarios the GP using k_{power} gives by far the best results.

For the consumers weather information does not seem to improve the prediction errors and both GP methods achieve comparable performance. Furthermore, Figure 2 shows that both Gaussian processes and the baseline method are able to provide the same level of prediction accuracy over all different prediction horizons. This enables a trading agent to buy electricity early when prices are low. Table 1 summarizes the results of one-step ahead predictions for the four evaluated customer types. We report the average hourly load, as well as the RMSE of the three prediction methods. Prediction using Gaussian processes with weather data performs best in all categories.

Summary and Outlook

We presented a machine learning technique for electricity demand forecasting based on Gaussian processes. By specifying a suitable covariance function we obtain a very flexible prediction model, that achieves very high accuracy on simulated data for different types of customers. Our method can incorporate arbitrary features describing weather conditions and considers daily and weekly periodicity of the data.

Gaussian processes could also be used to obtain error-bars for the predictions, which could help to improve the buying and selling strategies on the wholesale market. The agent could for example compute the lower bound of energy

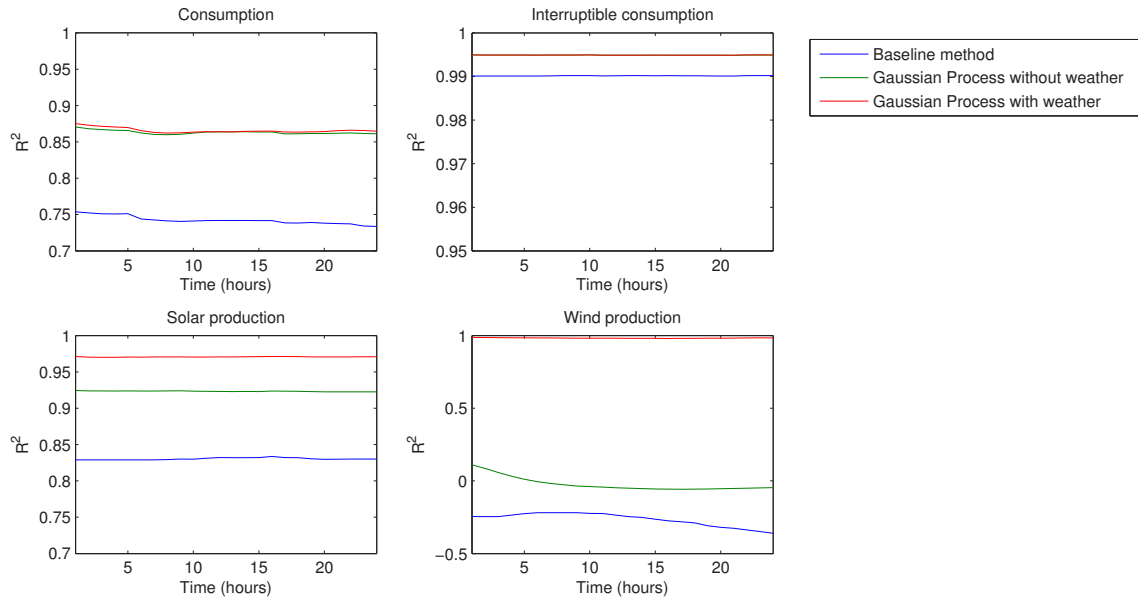


Figure 2: R^2 measures of the Gaussian process models compared to the baseline method for the different customer types.

Power type	average hourly load (MWh)	GP imbalance (MWh) (without weather)	GP imbalance (MWh) (with weather)	baseline imbalance (MWh)
Consumption	-31.71 \pm 8.23	2.96 \pm 2.01	2.91 \pm 2.08	4.08 \pm 2.66
Interruptible consumption	-12.22 \pm 2.00	0.14 \pm 0.09	0.14 \pm 0.09	0.20 \pm 0.13
Solar production	2.96 \pm 2.57	0.71 \pm 0.51	0.44 \pm 0.25	1.06 \pm 0.94
Wind production	4.51 \pm 2.96	2.89 \pm 1.70	0.33 \pm 0.25	3.30 \pm 2.42
Population	-36.46 \pm 8.36	4.10 \pm 2.70	3.00 \pm 2.17	5.13 \pm 3.32

Table 1: Average hourly usage and root mean squared prediction errors for the different customer types.

that will be needed with a probability of 95% and buy this amount early, when prices are low.

We also plan to use GP forecasting to predict energy prices in the trading module of our agent. Solid price forecasts allow the agent to buy low and sell high. Finally, we are participating in the 2013 PowerTAC, where our methods will have to prove themselves in a competitive environment.

References

Blum, M., and Riedmiller, M. 2013. Optimization of Gaussian Process Hyperparameters using Rprop. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Hager, W. W. 1989. Updating the Inverse of a Matrix. *SIAM Review* 31(2):221–239.

Ketter, W.; Collins, J.; Reddy, P.; and Weerdt, M. D. 2012. The 2012 Power Trading Agent Competition. *ERIM Report Series Reference No. ERS-2012-010-LIS*.

Leith, D.; Heidl, M.; and Ringwood, J. 2004. Gaussian

process prior models for electrical load forecasting. In *8th International Conference on Probabilistic Methods Applied to Power Systems*, 112–117.

MacKay, D. 1998. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences* 133–166.

Nguyen-Tuong, D.; Seeger, M.; and Peters, J. 2009. Model Learning with Local Gaussian Process Regression. *Advanced Robotics* 23(15):2015–2034.

Rasmussen, C. E., and Williams, C. K. 2006. *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.

Taylor, J.; Menezes, L. D.; and McSharry, P. 2006. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting* 44.

Taylor, J. 2003. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society* 44:799–805.