SVR in Load Prediction

Chuan Lu, Fudan University

Nov 11, 2016

Outline

Goal

To predict the max load of the following 1/3/7,... days with information we can get from history data and/or forecast of the next days.

SVR

From Wikipedia:

SVR is a version of Support Vector Machine (SVM) used for regression, proposed by Vladimir N. Vapnik in 1996. It is a supervised learning method, equivalent to solving

$$\textit{minimize} \quad \frac{1}{2}\|w\|^2$$

$$\textit{subject to} \begin{cases} y_i - < w, x_i > -b \leq \varepsilon \\ < w, x_i > +b - y_i \leq \varepsilon \end{cases}$$

SVR

It can be used to do linear or unlinear regressions using the support vectors, especially when the characters of data are linearly non-separable. In this case we can use different kernels to map data points into a higher-dimensional space. The most commonly used kernel is Gaussian Kernel (or called Radial Basis Function)

$$K(x, x_c) = e^{-\frac{\|x-x_c\|^2}{2\sigma^2}}$$

Parameters

In the given dataset, the only useful information is the dependent variable **load**. But from publications we can find many other parameters which could be used in our model.

Parameters

- MAX Load of yesterday
- MAX Load of 3, 5, 7, ... days before
- Weather information: temperature, wind speed/direction, humidity, ...
- Calendar information: weekends, weekday, holiday, ...
- Special Case: G20 summit, Regulation, ...
- ...

Parameters in practical use

After Cross-Validation, we finally chose the following parameters:

- MAX Load of past 6 days (Scaled: Divided by 10000)
- Temperature of tomorrow (Scaled: Divided by 29)
- Calendar info: Weekdays (By 6 Bi-digits), Holidays (By -1, 0, 1)

Dependency

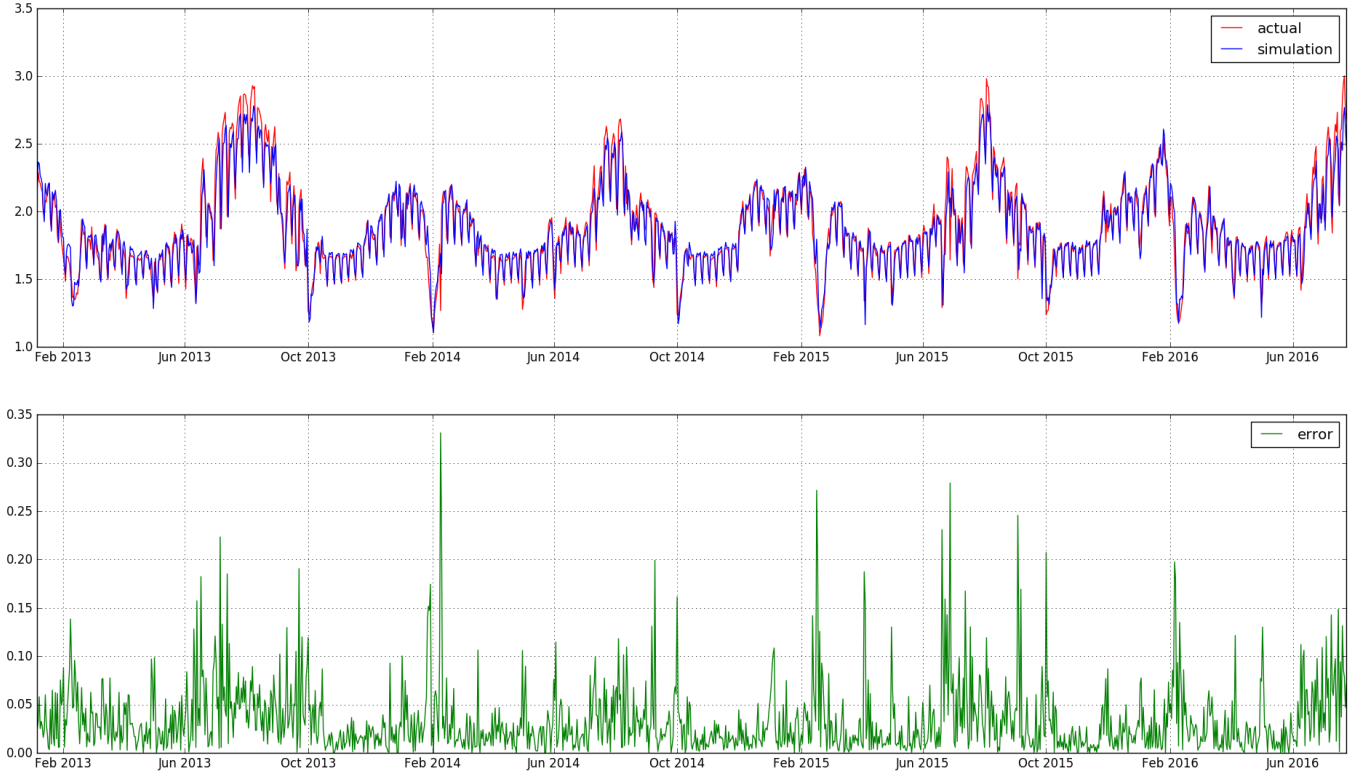
Written and Tested By Python 3.5. We wish the module could be deployed in Linux-based systems.

- Python 3+ (3.4+ preferred)
- Python-numpy (The latest version)
- Python-scipy
- Python-scikit-learn
- Python-pandas
- Python-matplotlib

Usage

See Readme.html

Prediction Result



Conclusion

Use 2 years' data as training set, 3.5 years' data as testing set.s

- Total average error is 3.1%.
- MAX error is 33%.
- Those over 20% are almost caused by holidays.
- The prediction is somehow too convergent.

Conclusion

There are some points for further study and improvement:

- Holidays could be marked artificially, we can adjust simply by multiplying a constant.
- We can bind the results of different methods, for example, Kalman Filters always produce radical results and SVR produce moderate ones; If we use the average of these two methods, there could be a more accurate one.