Prediction for night-time ventilation in Stanford's Y2E2 building

Balthazar Donon
Stanford University

December 16, 2016

I. Introduction

Indoor ventilation

In the United States, around 40% of the energy consumption is due to buildings, and a significant part of it comes from temperature management. In some areas, the climate is convenient to natural ventilation. California's mild climate is well suited for a night-time natural ventilation. Instead of using a large amount of energy in air-conditioning, and forced ventilation, the idea is to simply use the fact that the outside temperature is low during the night. Then one just needs to open the appropriate windows during the right amount of time to efficiently renew and cool the air inside the building.

The Y2E2 building has been built in 2008 and implements this smart indoor ventilation strategy. There are a few large windows on top of it that can be opened for ventilation. For now the windows opening strategy is quite simple and simply reproduce some pattern periodically. But if the opening strategy could predict how much time we want to keep the windows open to get to a specific temperature, it could also decrease the use of additional temperature management systems. This will be the primary objective of this project.

Many sensors were implemented in the Y2E2 building, giving us access to valuable informations such as the temperature at different points, the state of the windows (open/closed), the outdoor temperature, etc.

Wind Engineering Group

The Wind Engineering group, which is part of the Civil and Environmental Engineering department at Stanford University, is currently working on a tool for architects that could allow them to efficiently implement a night-time natural ventilation. This tool, called box-model, aims at predicting the evolution of the temperature in a building, depending on its size, the material used, the outdoor temperature, etc. This box-model is being refined through different strategies : CFD and theoretical modeling.

The temperature in the building obeys a differential equation whose coefficients are not precisely known. Using the large amount of data gathered since a few years could also help us estimate the coefficients of the differential equation. Then, instead of directly predicting the temperature in the building, we would be able to predict the differential equation it satisfies, and then the temperature. This will be the secondary objective of this project.

II. Data and Preprocessing

Predicting with estimated values

In this work we try to predict the indoor temperature of the Y2E2 building depending on many other parameters, such as the outdoor temperature for instance. One could argue that it doesn't make sense to predict on day d the indoor temperature for the day $d+1$ because you don't have access to the corresponding outdoor temperature, which is true. Instead we will consider that we have access to weather forecasts that we will consider to be equivalent to the real datas of $d+1$. The following assumption is made:

$$T_{outdoor}(d+1) = T_{predicted}(d+1) \quad (1)$$

Natural Features

The Y2E2 sensors record the following informations:

- Indoor Temperature at the 1st, 2nd and 3rd floor of each Atria (A,B,C,D);

- Outdoor Temperature at 3 different points (to simplify, we will average over the 3 points);
- Windows state (open/closed);
- Total Solar Irradiation;
- Wind Speed at 2 points (we will average over the 2 points);
- Wind Direction at 2 points (we will average over the 2 points);
- Time.

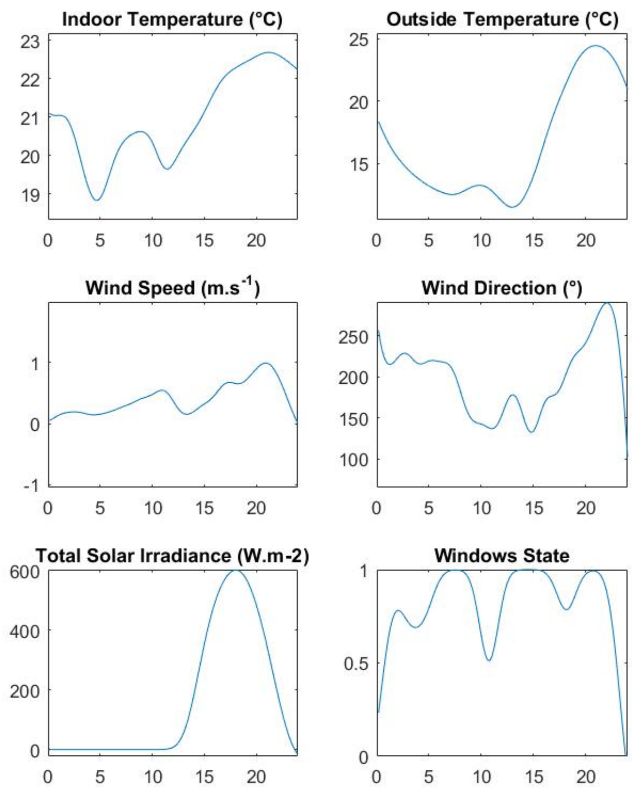

Figure 1: *Example of data recorded in the Y2E2 building*

Preprocessing

The measurements come from a small software called SEE-IT. Measurements between 01/01/2010 and 12/31/2015 with a 10-minutes time-step have been used. One of the main problems that was encountered with the data is that it was absolutely not well suited for Matlab. There was a lot to do to convert each temperature and date to the right format. Then I had to split the datas in days (between 6pm on day i and 6pm on day i+1). Many days (60% of them) were not complete and did not have the 145 points that they should have (6 points per hour, 24 hour; The first temperature of the following day is also included). Instead of interpolating the datas for the days that lacked only a few points, I have decided to do without all these days. This reduces a lot the number of days available. Anyway, the 859 remaining days should be enough to do some accurate predictions.

There was a lot of noise on the data, and since variation that occur in less than 30 minutes are not valuable informations, some smoothing has been done using a locally weighted linear regression. The parameter τ that has been chosen is $\tau = 5 \times \Delta t = 50min$.

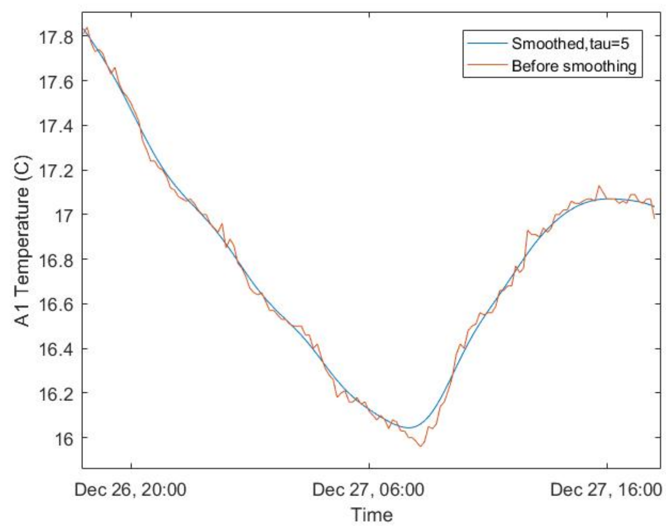

Figure 2: *Smoothing of the Indoor Temperature*

III. Features

Selecting Training and Testing Sets

We have a total set S of 859 days. For each of these days, we have one measurement for every 10 minutes of :

- Indoor Temperature;
- Outdoor Temperature;
- Windows state;
- Total Solar Irradiation;
- Wind Speed;
- Wind Direction;

Every time we launch a linear regression or a weighted linear regression, 90% of the set S is randomly selected and becomes S_{train}, while the 10% remaining becomes the set S_{test}. This random selection is done everytime a regression is made. That means that if we launch a linear regression 10 times using this code, it will produce 10 different sets of predictions. When we want to estimate the accuracy of a type of regression, we will run the same code 10 to 100 times and average the loss values.

Time-related Features It appears that when we try to predict the temperature in the building on December 20 for instance, we give the same importance to temperature measurements that were made during summer as those that were made during winter. Intuitively, we would want our algorithm to be able to model the impact of the moment of the year. The problem with date is that there are more days between December 20, 2010 and December 20, 2011 than between December 20, 2010 and July 20, 2011. That means that we have to consider a feature that would be periodical, and not simply linear. We will use the following two feature to model the impact of the date:

$$newFeature1 = \cos(\frac{2\pi}{365} \times Day)$$
$$newFeature2 = \sin(\frac{2\pi}{365} \times Day)$$

The value of Day is one for January 1, while it is 365 for December 31. Implementing these two new features in the linear regression gave a 10% improvement.

Loss The loss has been defined the following way:

$$Loss = \sqrt{\frac{1}{|S_{test}| * 145} \sum_{y^{(i)} \in S_{test}} (y^{(i)} - h(x^{(i)}))^2}$$

From a physical point of view this is an estimation in $°C$ of the error between the measured Indoor Temperature and the predicted values.

IV. Linear Regression

Linear Regression Let's say that we want to predict only the temperature in the Atrium A at the 3rd floor (A3). We will first use a linear regression assuming a normal distribution on the error ϵ (defined by $\epsilon^{(i)} = y^{(i)} - \theta^T x^{(i)}$). The average loss estimated is 0.64 degrees Celsius. This seems to be a pretty accurate prediction at first, but let's see if we can reduce the loss estimation under 0.5.

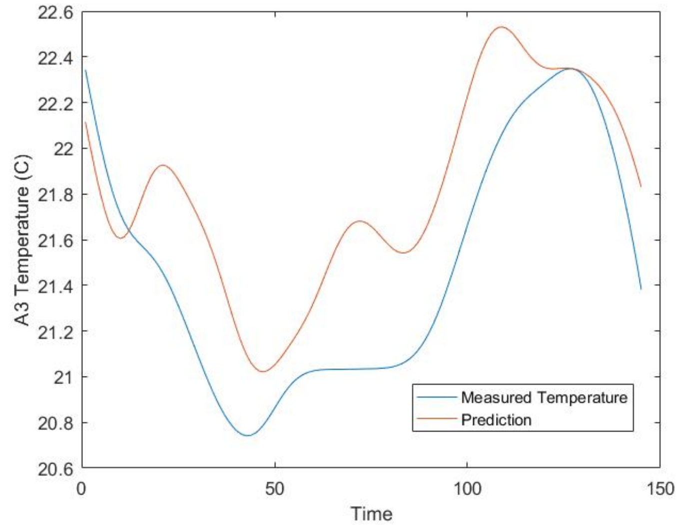

Figure 3: *Prediction using a Simple Linear Regression*

Locally Weighted Linear Regression One can expect much better results using a locally weighted linear regression. The weight is defined as :

$$w^{(i)} = \exp(-\frac{(x - x^{(i)})^2}{2\tau^2}) \quad (2)$$

This has been done for several values of τ and it appears that we get the best results for $tau = 10^8$. It might seem high, but no scaling has been done on the data.

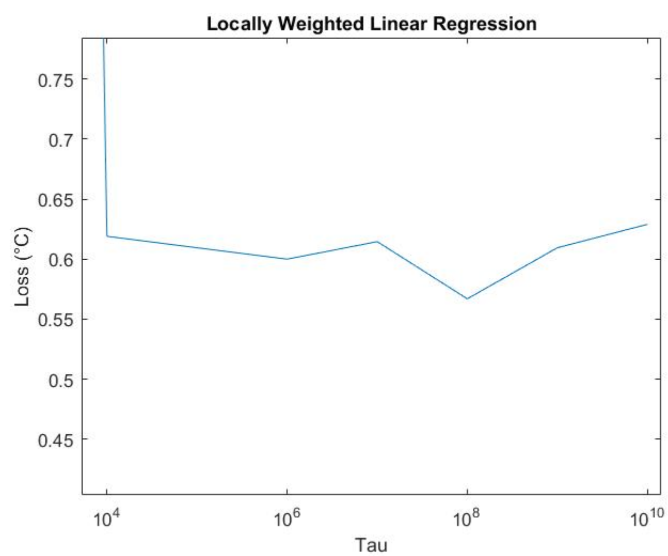

Figure 4: *Loss vs. Tau*

The minimum Loss obtained using this method is 0.58°C which is slightly better than with the simple Linear Regression.

V. Functional Regression

Naive Approach We define the following distance function:

$$d(x^{(i)}, x^{(j)}) = \|x^{(i)} - x^{(j)}\|_2^2$$

Using this distance, we have performed a functional regression, and the estimated loss is $0.806°C$, which is not even as good as a linear regression. The problem is that the input data has not been scaled. The order of magnitude of the Windows State is 1, while the order of magnitude of the Total Solar Irradiance is 10^3.

Parametric Functional Regression This time we will split the distance function between the outside temperature part, wind speed part, wind direction part, etc. That means that we will weight the distance depending on the nature of the data.

$$\begin{aligned} d(x^{(i)}, x^{(j)}) =& \theta_{outTemp}\|x^{(i)}_{outTemp} - x^{(j)}_{outTemp}\|_2^2 \\ &+ \theta_{wind}\|x^{(i)}_{wind} - x^{(j)}_{wind}\|_2^2 \\ &+ \theta_{solar}\|x^{(i)}_{solar} - x^{(j)}_{solar}\|_2^2 \\ &+ ... \end{aligned}$$

In the following example, only the θ related to the window state is modified. By optimizing the Loss with regards to each component of θ, we managed to get a $0.498°C$ error.

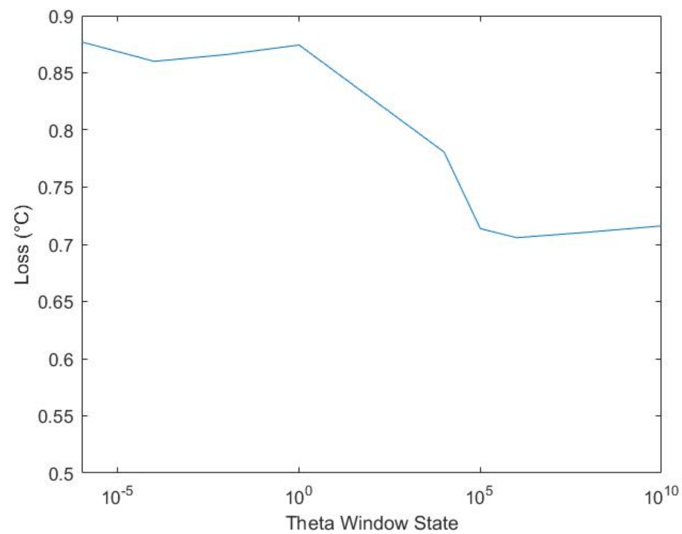

Figure 5: *Parametric Functional Regression - Loss for different values of* $\theta_{WindowsState}$

VI. Differential Equation

Idea The estimations that we get using linear and functional regressions are quite good but lack of "coherence". It seems like the temperature at t is not related at all to the temperature at $t-1$. The system predicts what happens at t based on every single measurements that has been done.

Another interesting approach is to model the time evolution of the indoor temperature using a differential equation. The parameters of this differential equation can then be estimated using a neural network. What is interesting about this strategy is that it makes more sense from a physical point of view. The parameters of the differential equation have a physical meaning.

Implementation We will model the behaviour of our indoor temperature the following way :

$$\frac{dT_{indoor}}{dt} = f(t, T_{indoor}(t), Inputs(t))$$

Our $Inputs$ are actually the features that were defined above: Outdoor Temperature, Total Solar Irradiance, etc. The approach is a little different than before: we will consider the indoor temperature as an input, and its derivative as the output. To construct the time derivative we use the following scheme:

$$\frac{dT_{indoor}}{dt} = \frac{T_{indoor}(t+1) - T_{indoor}(t-1)}{2 \times \Delta t}$$

To clearly define our problem, we set $y^i = \frac{dT_{indoor}}{dt}(t^i)$, $x^{(i)} = [t^i, T_{indoor}(t^i), I_{solar}(t^i), ...]$, and we will use a neural network to estimate the function f so as to minimize:

$$\sum_i \|y^i - f(x^i)\|_2^2$$

Once f is estimated, we can compute the indoor temperature simply by solving the differential equation, and the $Loss$ is defined as before.

This strategy has been achieved using the "Neural Net Fitting" app of Matlab. 40 neurons and one neuronal layer have been used. The average Loss is $0.56°C$. This is not as good as the Parametric Functional Regression, but this time the neural network has a physical meaning and can be used to better understand how the physical system works.

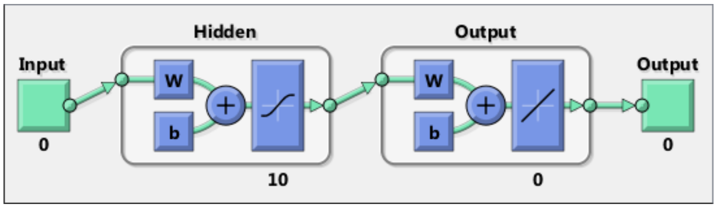

Figure 6: *Matlab Neural Network*

VII. Conclusion

In order to get a better idea of how the predictions look like, here are the smoothed data and three predictions achieved through different methods.

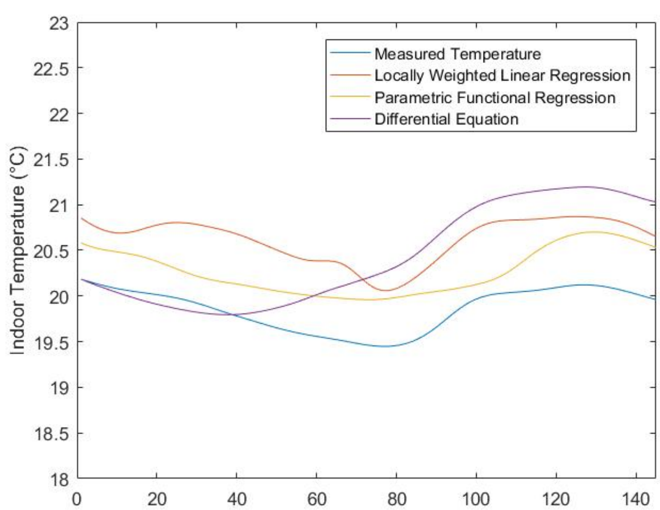

Figure 7: *Indoor Temperature Predictions*

Linear Regression	$0.606°C$
Locally Weighted Linear Regression	$0.580°C$
Naive Functional Regression	$0.806°C$
Parametric Functional Regression	$0.498°C$
Differential Equation and Neural Network	$0.560°C$

Table 1: *Prediction Performance*

It appears that the Parametric Functional Regression gives us the best results.

Having an efficient tool to predict the temperature inside the building allows us to try different strategies for the opening of the windows. Moreover, it could be interesting to implement a reinforcement learning strategy for the windows. This can be a very interesting tool for architects and energy consumption optimization.

VIII. Future Work

The Differential Equation part might be very useful to the development of the "box model". I believe some improvement can be made and can help us better understand the impact of different parameters on the indoor temperature. I think that it might be interesting to focus on a specific day rather than trying to estimate a differential equation that would work for every day.

IX. Acknowledgement

I would like to thank Pr. Catherine Gorlé and the Wind Engineering Group (Stanford, CEE departement) for this project idea and for the dataset.

References

[Nejib Smaoui and Suad Al-Enezi, 2004] Journal of Computational and Applied Mathematics 170 (2004) 27-58.*Modelling the dynamics of nonlinear partial differential equations using neural networks*

[The MathWorks, Inc.] Machine Learning and Statistics Toolbox Release 2015b, Natick, Massachusetts, United States.

[The MathWorks, Inc.] Neural Network Toolbox Release 2015b, Natick, Massachusetts, United States.