Project ideas, version 2

Prabir Barooah

October 28, 2018

1 Simultaneous identification of a system model and disturbance using a Kalman filter

Data-driven modeling, or system identification, is frequently used to model complex engineering systems. In the simplest case, we assume the system in question is linear, and we can measure inputs and outputs of the system. The job is to estimate the model parameters from input-output measurements. Let use a simple model structure, say, a transfer function in discrete time, $H(z)$. In that case the problem becomes choosing an appropriate order of $H(z)$ and determining the coefficients in the numerator and denominator polynomials of $H(z)$. If the system is MIMO, then one has to do this for all the entries of the matrix $H(z)$. If the system is modeled in the state-space form: $x_{k+1} = Ax_k + Bu_k, y_k = Cx_k$, then the job is to figure out the matrices A, B, C.

The basic approach - or even the not so basic approaches - essentially relying on figuring out the model so that the prediction error of the model, $e = \sum_k \|y_k - \hat{y}_k\|$ is small, where $\hat{y}_k$ is the output of the estimated model for the same input u that was measured.

An important assumption here is that all the inputs to the system can be measured, or that if there are other inputs (disturbances) that cannot be measured, the effect of those inputs are small. When that assumption is violated, the problem has infinite number of solutions. The difficulty is best described in transfer function domain. Suppose because of the disturbance, the model is $y(z) = H(z)u(z) + w(z)$, where w is the disturbance. You can pick any H, say H_0, and claim that is the plant model, and then say that the disturbance is $w_0(z) := y(z) - H_0(z)u(z)$. The pair H_0, w_0 will perfectly satisfy the relation $y(z) = H(z)u(z) + w(z)$.

Additional information, or constraints, must be imposed to determine the plant and the disturbance simultaneously in such as situation. Usually you must know something about the system and/or disturbance for you to impose these additional constraints. Otherwise it is a hopeless situation.

This particular problem is faced in modeling thermal dynamics of buildings. Model-based predictive control of building's heating, ventilation, and air-conditioning (HVAC) systems - especially for large buildings - has become a hot research topic in the last few years because of the large energy use of buildings, and the possible role they can play in managing the highs and lows of electricity generation from solar and wind[1].

Any model based control needs a model of the thermal dynamics of a building. The input to such a model is the various sources of "heat gains", such as ambient temperature, solar irradiance, and

[1] It is not possible for me to explain why here; that'll be too much of a digression. If you are interested, let me know and I'll send you some references you can read.

heat injected into the building by the HVAC system[2], and the main output is the indoor temperature of the building. An important input is the heat produced by occupants and the equipment used by occupants. Every person produces approximately 100 W of heat, and of course our computers, monitors, printers and lights and coffee makers produce heat as well. All of these are collectively called the "occupant induced heat gain". This input is impossible to measure except in a very carefully and specially instrumented building. So this plays the role of the unknown disturbance w mentioned above.

But we know something about the occupant induced heat gain: it does not change wildly with time. In fact, people come into a commercial building at around the same time, leaving temporarily for lunch perhaps, and then leave for the day at around the same time. So this disturbance is approximately piecewise constant. In other words, its derivative is close to 0 most of the time, differing significantly from 0 only at specific times (we do not know those times though). So instead of modeling it as an exogenous disturbance, we can model it as as a state x_{n+1} (assuming the basic model had n states), with dynamics:

$$\dot{x}_{n+1}(t) = 0 + w(t)$$

where $w(t)$ is a 0 mean stochastic process that captures the error in the "$x_{n+1}(t) = \mathit{contant}$" model. In discrete time, this becomes

$$x_{n+1}[k+1] = x_{n+1}[k] + w[k]$$

where again where $w[k]$ is a 0 mean random sequence.

A recent paper [1] used this idea to simultaneously estimate a plant model and the disturbance signal (the state x_{n+1}) from input-output data collected from a building. The disturbance (the state x_{n+1}) was estimated using a Kalman filter. An outer loop optimization was used to estimate the parameters of the plant model (which was in state-space form).

There are several possible projects based on this idea, some of which I describe below. To understand what follows and to assess how difficult or easy the suggested projects will be, you will need to (1) familiarize yourself with thermal dynamics of buildings, and (2) the paper [1]. A handout on one specific type of thermal dynamic model, the so-called "resistor-capacitor network models", or RC network models for short, has been prepared to help you do item (1). You can find the handout in "Files > FinalProjectRelatedDocs > papers"; the document is named "ThermalModelsExplained.pdf".

1. Extend the algorithm proposed in [1] to make it applicable to a building with multiple rooms by lumping them into one. In [1], the method was applied to a single zone of a building, specifically to the auditorium in Pugh Hall at the UF campus. Because the auditorium is one single space, there is one single output - the average indoor temperature. Similarly, the input signals - such as heating power delivered by the HVAC system - are also unambiguous since all of that power is delivered to one space. Same with the occupant induced heat gain. When you consider a building with multiple rooms, there are multiple room temperatures and each of them is an output. Similarly, there are multiple inputs since the heating power injected by the HVAC system into every room is distinct. You can still lump the entire building into one plant, with the total HVAC power as one of the inputs, and take the average of the various room temperatures as the single output. The advantage is that now there is again a single unknown disturbance affecting the entire system that needs to be estimated.

[2]In Florida the HVAC system usually provides cooling, in which case the heat injected by the HVAC system is negative.

2. Extend the algorithm proposed in [1] to make it applicable to a building with multiple rooms, by dividing the building into multiple sections (depending on its geometry perhaps) and then estimate the model and the disturbance for each section. In terms of the value of the resulting model to use in real-time control, such a model is more useful than the structure described above. But the estimation problem is harder. There are more degrees of freedom, and the multiple disturbance signals can absorb all the model mismatch, leading to poor estimates.

3. Extend the algorithm proposed in [1] to a different model structure. For instance, a black-box LTI model rather than the gray-box RC network model. Or, you can add one more path for solar radiation, which will increase the model parameters (two effective areas rather than one).

4. Assessment rather than extension: Use the same method as in [1], but apply it to data from multiple buildings. (See the note on data below.)

1.1 Data

There are at least two data sets available to you to apply your algorithm to.

1. *Data set 1: Pugh Hall data:* It is the same data set that the method in [1] was applied to. The data comes from the large auditorium in Pugh Hall in the UF campus. The data is available in the folder "Files > FinalProjectRelatedDocs > PughHallSingleZoneData" in the form of two .mat files. There is a matlab script in the same folder which plots the data in the mat files. The mat files contain data for two weeks. Which variable corresponds to which signal is easy to see once you run the plotting script and examine the axis labels.

   If you want to take on the problem of "multi-zone model identification", let me know and I will provide you data from other sections of Pugh Hall that you can use. As I mentioned earlier, you should first reproduce the results for the single-zone case reported in [1] before you attack the multi-zone problem.

2. *Data set 2: United Way College building in Singapore* The data is available here: `http://www.datadrivenbuilding.org/United-World-College-Campus`. The site provides an IPython notebook for loading and visualizing the data. You can also manually download the data files and use MATLAB© to extract and plot the data, but then figuring out which variable corresponds to what signal collected at what location in the building will become a bit more tedious. One advantage of this data set is that it is for a much longer period than 2 weeks.

2 Ensemble Kalman filter

The Kalman filter computes the best linear estimate of the state x_k, the $E*[x_k| \ldots]$. When everything is Gaussian, this is the conditional mean. It also computes the covariances of that estimate. Since the mean and covariance completely defines the pdf of a Gaussian random vector, in the Gaussian case the Kalman filter completely specifies the conditional pdf of the state. But when any of the assumptions used in the derivation of the Kalman filter fails, such as linearity of the process dynamics, or the noise processes are not Gaussian, it no longer provides the conditional pdf. Especially when the process dynamics are not linear, the conditional pdf of the state may be multi-modal. In that case it is desirable to estimate the conditional pdf directly instead of just its mean. Two "Monte-Carlo" extensions of the Kalman filter that estimates the conditional pdf are (1) the Ensemble Kalman Filter (EnKF) and (ii) the particle filter.

I would not recommend picking up the particle filter for this project; it requires many tricks to make it work.

The EnKF is both good enough and simple enough that you can take that up as a project topic. This filter was originally proposed by Evensen in [2]. You are however better off by reading the tutorial paper [3] than the original reference in order to learn the filter.

Another useful resource is the website `http://enkf.nersc.no/`.

You will have to choose a problem to apply the EnKF to. You are free to choose any problem, but make sure you have good quality data and the target problem is a good candidate for the EnKF. If all you have is a linear model that can be simulated to generate data, that is not a good source of data for such a project. The simple Kalman filter will be good enough for such a problem.

One possibility for a target problem is the one mentioned in the previous section, especially the one with the non-linear RC-network model.

No matter what target problem you choose, do not forget to implement the basic Kalman filter and compare the performance with the EnKF.

3 Solar energy/power generation prediction

This is another system identification, or "data driven modeling", type project. The goal is to identify a model that can be used to predict solar power generation from a PV installation in one particular location. Imagine you work for a power company that has a large PV (photo voltaic) farm. For planning and operation of the power grid, it is essential to know how much power the solar array is going to produce in the next few minutes, hours, days and weeks[3]. For a specific solar installation at one specific location, solar power production varies with time because of its dependence myriad phenomena such as change in atmospheric conditions through which the sun's rays travel, cloud cover, ambient temperature[4] etc. So, although it is easy to make qualitative predictions such as "the same PV array in Arizona will have higher generation than one in Germany", it is difficult to make quantitative predictions, such as "a 10 MW solar array in Gainesville will produce 2 MW at 9.15 am on Nov 5, 2018".

Generation data from 3 solar PV arrays installed in UF's Diamond Village apartments are made available to you in the `DiamondSolarData` folder in a csv file. The three PV arrays are on three different buildings, #300, #304, and #306, and the rows are named accordingly. The data is sampled approximately every 5 minutes. Figure 1 shows the monthly energy generated by the three PV arrays over a period of approximately 2 years. You can see the reduced conversion efficiency of the solar cells during hot summer months clearly in this plot.

One idea to help in prediction is to come up with a model for solar generation, whose structure is decided based on some guesses or prior knowledge, and then estimate the unknown coefficients in the model from past solar generation data collected from the very facility for which a predictive model is

[3] If you are wondering why it is important to know this, please see me and I'll provide references. I wont be able to explain this here as it will take us too far afield. Suffice it to say that blackouts and other bad things will happen if the power grid operators do not have good predictions of future generation from uncontrollable energy sources such as solar and wind.

[4] Efficiency with which a photovoltaic cell converts light to electricity varies inversely with ambient temperature, so the cells are less efficient at hotter climates.

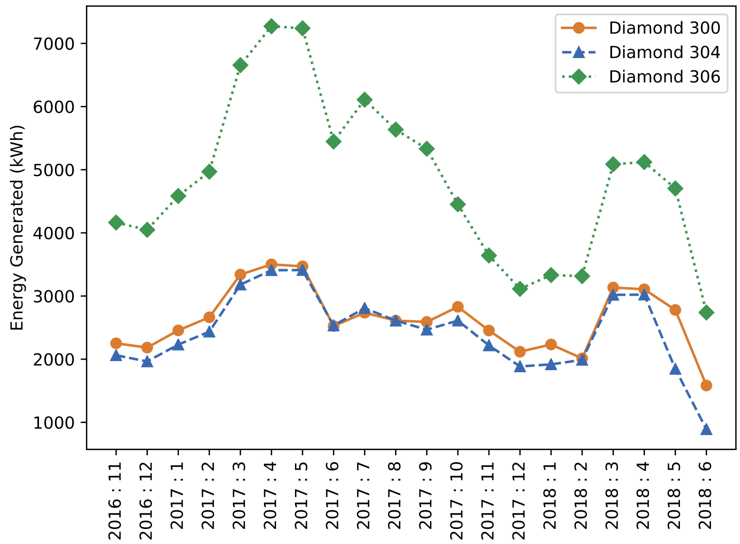

Figure 1: Energy Generation for each month. The three curves correspond to the three PV arrays in Diamond Village.

sought. Especially if one uses long-term data for a specific PV installation, one can hope that all the factors such as which angle the PV array is tilted, if there are shades from nearby trees or structures that affect its generation, etc. are already contained in the data, and one does not have to model those affects. Of course, random changes in cloud cover etc. cannot be captured, but perhaps predictable changes in weather in that location (such as summer afternoon rain clouds in Gainesville) will again be contained in the data, and a well fitted model will be able to predict the average generation.

The paper [4] uses an ARMA (autoregressive moving average) model to fit a model to solar generation data from a small scale PV array. The paper also contains a review of alternative methods for solar generation prediction.

ARMA models are part of a general class of modeling techniques called *time series modeling*. The key idea to remember in time-series modeling is that if you want to predict some variable y, but do not know what inputs are producing y as the output of some unknown system, then you use time series modeling: y_k is modeled as a function of y_{k-1}, y_{k-2} etc. and some correlated noise as input. The hope is that by using sufficient memory (the past outputs), the model will capture the behavior of the unknown system and unknown inputs.

Another way of thinking about this is: $y(s) = P_1(s)\ u(s)$, and both the plant P_1 and the input u is unknown, but the unknown input u is the output of some unknown system driven by noise: $u(s) = P_2(s)w(s)$ (w is noise) and so $y(s) = P(s)w(s)$, and an ARMA model is a model of P.

In some cases it makes sense, in some cases it is plain silly to try. It is very common in the field of economics and finance. Whether that says anything about time series modeling or of these fields, I leave that to you.

A good introduction to ARMA and its variants is in its Wikipedia page.

Specific project ideas:

1. Fit an ARMA model to Diamond Solar data. Assess its predictive power at various time-scales: 15 minutes, 1 hour, 2 hours, 3 hours,...24 hours, 48 hours, etc. Use various techniques to fit an

ARMA model, and compare the performance of various methods, such as ML, least squares. You can also use some fancy technique from MATLAB©'s system identification technique.

2. Fit and ARMAX model: A weakness of using an ARMA model for this application is that some known features of solar generation are not included. For instance, we know that solar cell efficiency is a function of ambient temperature. Therefore it makes sense to use ambient temperature as one of the input signals to the model. Since ambient temperature predictions are available from weather services, the resulting model can still be used for predicting solar power generation. Many of the other factors cannot be measured or predicted, such as cloud cover, so a time series approach still makes sense. So, you can combine these two ideas and come up with a hybrid model: and this is called an ARMAX model.

3. Any other model of your choice. The goal is the same: prediction of solar generation (kW, not kWh).

No matter what method you use, first divide the data into two sets, calibration and validation sets, or in-samples and out-of-sample data[5]. Fit your model to calibration data only, and assess its performance with validation data only.

If you go with an ARMAX model, you'll need to collect weather data for the Gainesville. This can be done by downloading data from wunderground.com or by from national weather services by using their APIs.

References

[1] A. Coffman and P. Barooah, "Simultaneous identification of dynamic model and occupant-induced disturbance for commercial buildings," *Building and Environment*, vol. 128, no. 153-160, 2018.

[2] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/94JC00572

[3] J. Mandel, "A brief tutorial on the ensemble Kalman filter," *ArXiv.org*, 2009, arXiv:0901.3725.

[4] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar generation prediction using the arma model in a laboratory-level micro-grid," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2012, pp. 528–533.

[5]Remember the homework on in-sample and out-of-sample data?