Regression based energy model and building energy saving calculation: analyzing the effect of temperature variable representation, and plan of next step

December 20, 2016

Contents

1 Goal statement

The broad goal of the study is to build a better (more accurate, or more interpretable, or both) building energy baseline model. The result will be useful in the application of building load / demand / consumption forcasting [4, 10, 15, 18, 26], calculation of energy savings from energy conservation retrofits [6, 7, 9, 11, 12], evaluating building energy performance [1], building parameter estimation [10], energy end use disaggregation [5, 7, 25], assist the "identification of energy saving opportunities and recommend the types of energy efficiency measures" [5].

The common input variable to the model include:

- Environmental variable
  - Temperature (measured, average, or categorical):
    - outdoor air temperature
      - as numerical: mean [4, 9–11], degree-day [7, 14, 17], Radio Basis Function Kernel (RBFs) [25], exact [3, 13, 27]
      - as categorical variables [26]
    - indoor air temperature [10]
  - Humidity
    - relative humidity (RH) [4]
    - dew point temperature [3]
    - exponential smoothing applied to humidity with time constant of 24h [2]
  - Solar:
    - solar radiation (W/m^2) [4, 10]
    - solar flux [13]
    - solar aperture (m^2) [10], different in different time of year
    - solar gains ($Q_S = SI$, unit: W) [10]
  - Wind
    - speed [13]
    - velocity [2]
- Occupancy
  - Number of occupants [26]

  - Operation schedules [17]
  - Occupancy ratio (ratio of occupied vs non-occupied days) [16]
- Industry type
- Building construction
  - Detached vs apartment, categorical [26]
  - Construction material: wooden vs non-wooden [26]
- Time
  - day type (every-day, weekday, weedend) [9]
  - hour of day ($[9, 25]$, [8] mean-week and day-time-temperature regression model)
  - day of week ([8] mean-week, day-time-temperature, and LBNL regression model)
  - time lag (k), the number of previous readings to include in the model [10]
  - unit circle representation of time of day, week, month, and year [2]
- Energy
  - power (W, it's an auto-regressive component: use energy to predict energy) [10]([15] has some experiment about prediction of different time horizon using different time resolution)
  - fuel type: Electric vs non-electric [26]
- Floor area [26]
- Building dynamics
  - Heat loss coefficient (W/m^2K) [26]
  - Equivalent leakage area (cm^2/m^2) [26]
- Retrofit type / time
  - pre-retrofit period [11]

The common output could be whole building energy such as whole building electricity [9] or gas, or single end use such as air handler unit (AHU) electricity [9], chilled water energy [9], chiller energy [1], condenser energy [1], hot water energy [1,9], AC-electric, appliance-electric, base-electric [25].

In literature, the topic is commonly referred to as: whole building or building baseline models [4,8,11,17,27], weather-adjusted index of consumption [7], energy signature model [10, 16], inverse (energy) model [1, 12, 27], data-driven (energy) models [27], energy prediction model [4, 15, 26].

2 Scope

The document presents the results of analyzing the influence of the temperature representation on model prediction accuracy.

In previous works, the most common representation of temperature used are:

- raw temperature
- average temperature

$$\text{average temperature} = E[T_i] \tag{1}$$

- degree-day

$$\text{cooling degree-day} = \sum_{i=1}^{n} (T_i - T_{base}) \mathbb{1}(T_i > T_{base}) \tag{2}$$

$$\text{heating degree-day} = \sum_{i=1}^{n} (T_{base} - T_i) \mathbb{1}(T_i < T_{base}) \tag{3}$$

- smoothed with exponential weighting function [2, 21]

$$s_0 = x_0; s_t = \alpha x_t + (1 - \alpha) s_{t-1} \tag{4}$$

$$\alpha = 1 - \exp(-\frac{\Delta T}{\tau}) \tag{5}$$

  [21]

- radio basis function [25]: defined to be some function ϕ such that $\phi(x) = \phi(\|x\|)$ [23], it's a non-linearly transformed degree-day representation to approximate / smooth the environment (temperature) input

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x} - \mathbf{x_i}\|) \tag{6}$$

  [23]

It is very common that the time resolution of the temperature data and the energy data is different: the common time step for temperature is usually hourly, while the energy data could be sub-hourly as in the case of smart meter measured values, or monthly in the case of utility bills. There is a tendency to unify the two variables into the same time resolution: when only monthly utility bills are available, the temperature is usually aggregated into a single value for each month, the average temperature, or the degree day.

However, such aggregation might lose some detailed information of the temperature changes, thus could affect the prediction accuracy of a model. The average monthly temperature approach does not record the temperature fluctuation, so a climate with very hot day and very cold night could have the same average temperature as the one with very stable temperature throughout the whole months, but the former would use a lot more heating and cooling energy comparing to the latter. The degree-day reflects more about the cooling and heating load, but the choice of T_{base} could affect the value of the degree-day. One way to choose a base temperature is to fit a linear regression model for a range of candidate T_{base} and choose the fit with the highest R^2. After choosing this single T_{base}, we assumed that the heating is only relevant to the temperature above that base (for cooling) or below that base for heating. Whether there's such a hard turning point is not certain.

Based on the previous observations, two general thoughts of how to improve the models using either aggregation method is: 1) can we remove the aggregation completely? 2) can we use some smarter aggregation method?

Thus the following approaches are proposed:

- No aggregation: Use the vector of hourly temperature for each month as the independent variable, with regularization.

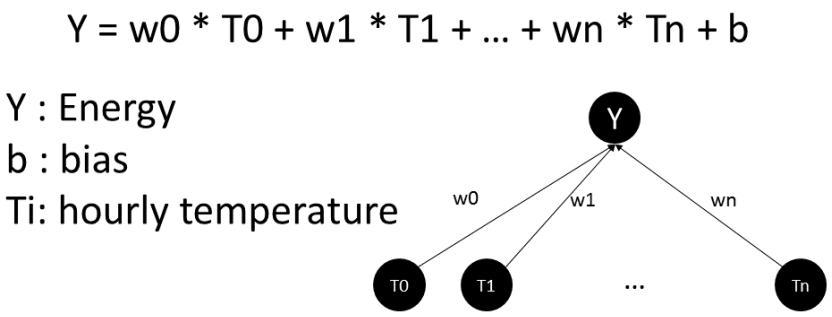


Figure 1: method 1 diagram

- Smarter aggregation

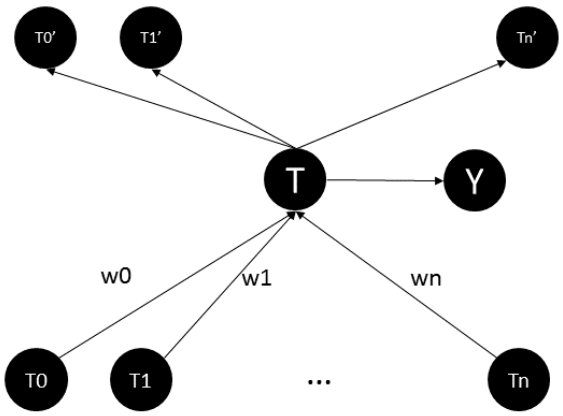


Figure 2: method 1 diagram

- Pre-process hourly temperature with PCA and use the projected input as the independent variable.
- Learn an auto-encoder of the hourly temperature to get some lower dimension hidden representation and use the lower dimensional representation as the independent variable.
- Keep many degree-day variables in the model, with a set of different bases, and regularize with fused lasso [20]

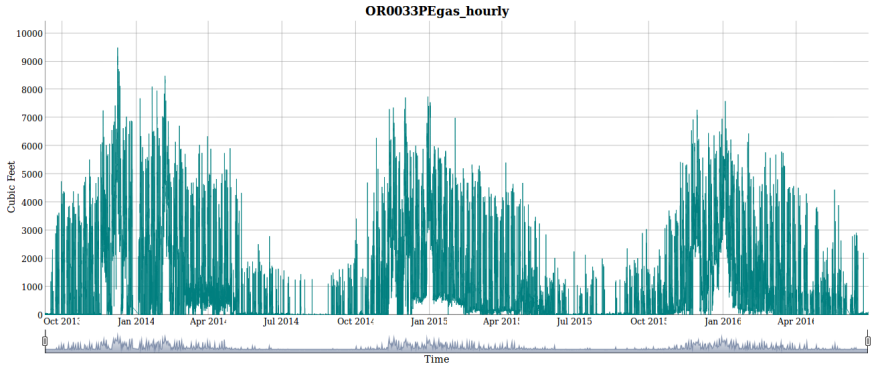
These methods are to be compared with two baseline methods: The piecewise regression model with monthly mean temperature as the independent variable; the best-fit (highest R^2) degree-day model.

3 Process

The data is randomly split into a train-development set, and a test set. This is done with the `sklearn.model_selection.train_test_split` For the ridge regression approach, the train-development set is used to tune the ridge parameter with cross validation and also learn model, the test set is used to report the performance.

Before performing regression, the X should be standardized by $x'_i = (x_i - E[X])/\sigma_X$, and Y should be centered by $y'_i = y_i - E[Y]$(??) [19]

The following result is based on the experiment with the building "OR0033PE".



3.1 Baseline method: piecewise linear regression with average month temperature

MSE: 2.07631642389

3.2 Proposed method 1: Hourly temperature with ridge term

Given the data $\mathcal{D} = \{(\mathbf{x^{(i)}}, y^{(i)})\}^n$, where $\mathbf{x^{(i)}} \in \mathbb{R}^{T_i}$ is the hourly temperature of the i-th month, $T^{(i)}$ is the number of hours in the i-th month. For simplicity, $T^{(i)}$ is fixed to $720 = 24 \times 30$. $y^{(i)} \in \mathbb{R}$ is the monthly whole building electricity or gas consumption.

The form of the regression mode is

$$\hat{y}_i = \theta^T \cdot \mathbf{x^{(i)}} \tag{7}$$

Note, since the data is pre-centered, there's no bias term in the model.

The model is learned by minimizing the loss function Equation 8.

$$J(\theta) = \frac{1}{2}(\theta^T \mathbf{x^{(i)}} - y^{(i)})^2 + \lambda \left\|\theta\right\|^2 \tag{8}$$

λ is the ridge parameter that controls the amount of regularization, it is tuned in the train-development set with cross-validation. `sklearn.linear_model.Ridge` is used for compute the ridge regression.

Cross-validation is performed by first partitioning the train-development data set into K folds. For the i-th fold, a model is learned on the rest K - 1 folds ($\{K_j \mid j \neq i\}$), the error is computed on the i-th fold. This process is repeated for K times. The error metric is mean squared error, the overall error is the average of the error of using the i-th fold as the test set. This is done with `sklearn.model_selection.cross_val_score`. In this experiment, 5-fold cross validation is used.

In the ridge regression, the hyper parameter λ needs to be tuned with cross-validation. A plot of MSE $- \lambda$ is shown in Figure 3

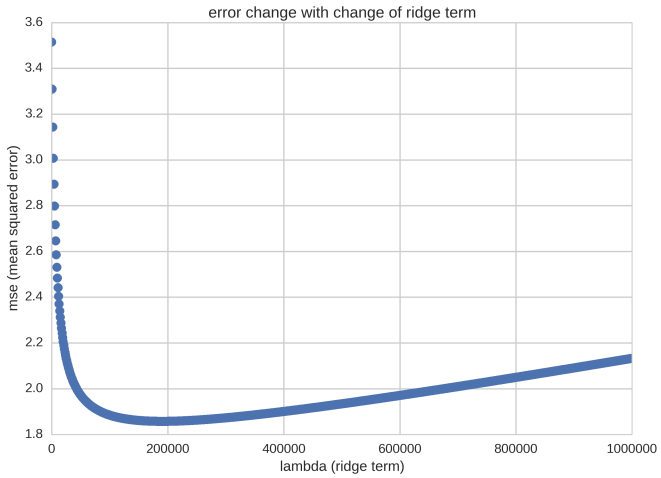


Figure 3: MSE vs lambda plot

From the plot, the λ that gives the lowest error is around 190000. Using this optimal λ to re-fit the model, we get the model:

```
[ -1.55822925e-04  -1.42539701e-04  -1.09100028e-04  -4.06291050e-06
   8.84344833e-05   1.78332926e-04   2.68478265e-04   2.65000351e-04
   2.02735707e-04   1.39508922e-04   1.30677887e-04   1.08732504e-04
   8.07017042e-05   8.77490889e-05   1.00807169e-04   9.19141490e-05
   8.23187429e-05   7.24149582e-05   4.88254898e-05   7.27015617e-05
   1.01496385e-04   1.02519860e-04   9.03311707e-05   6.98651205e-05
   6.75534592e-05   1.24527082e-04   1.25621821e-04   1.27268793e-04
   ...
]
```

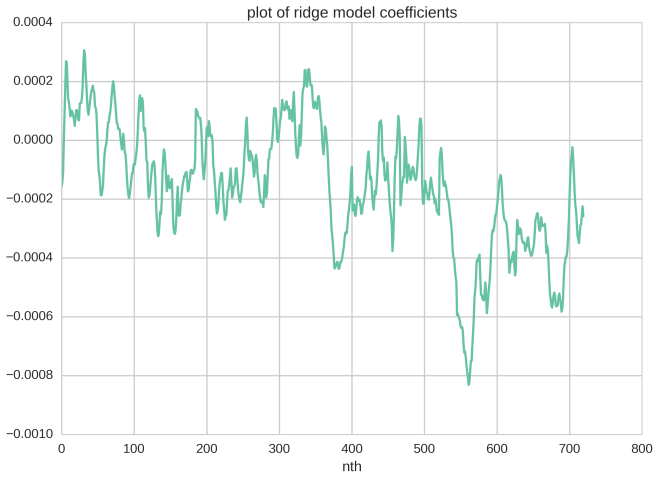


Figure 4: ridge regression parameter plot

From the plot of the parameters, the first two weeks seems to have larger impact on the gas consumption. The MSE of the building is: 1.37049790455

Later development will use Baysian Optimization will be used to tune the hyper parameters instead of the current grid search.

3.3 Proposed method 2: PCA

PCA regression [22] uses top PCs of the input data as the independent variable for the regression. It is "some kind of regularized procedure" [22].

3.3.1 result of analyzing weather station data

By analyzing the data from 483 weather stations in the U.S. used in the GSA project, the number of principal components with re-construction error of 95 percent (top) 99 percent (bottom) is shown in Figure 5

The input is a $720 \times m$ matrix X (720 hours per month, for simplicity), m is computed by `n / 720` (round down). Each column in the matrix X is a month worth of hourly temperature. The data is re-centered by $X - \mu$, where μ is the average of X.

The principal components analysis is conducted with the eigen decomposition of the covariance matrix, XX^T. The output include the eigenvectors and the eigenvalues. The eigenvectors are the principal components, and the ith eigenvalues are the variances accounted for by the ith principal components.

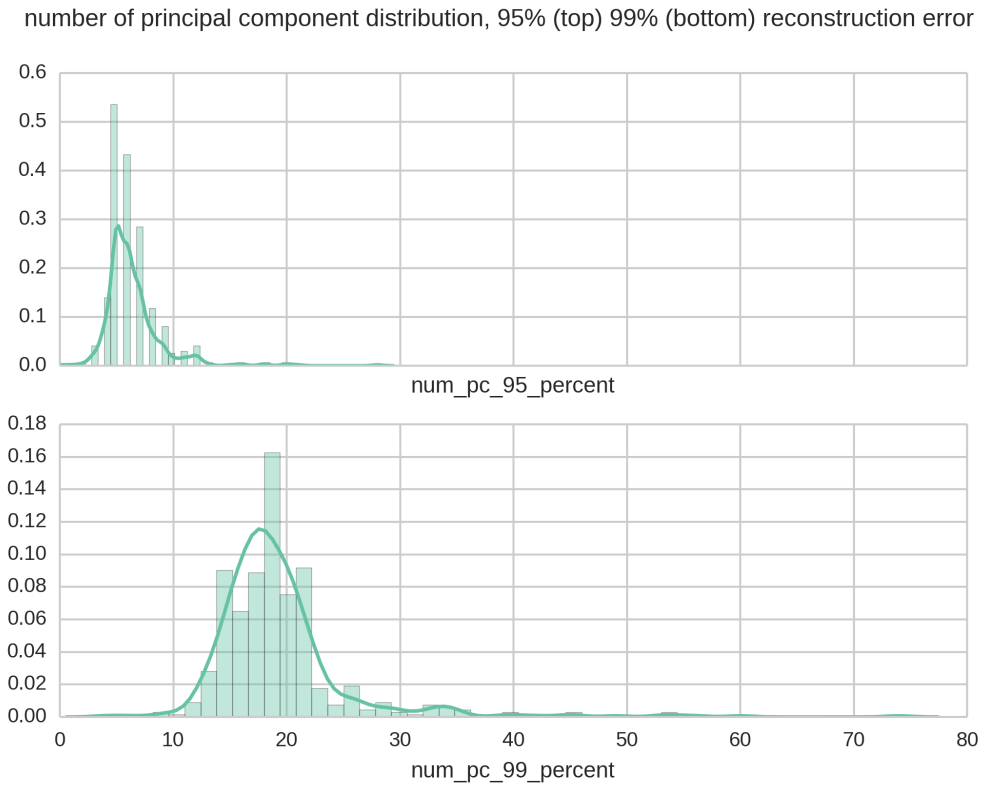


Figure 5: Number of principal components with 95 and 99 percent reconstruction error

```
Accounted for 95% variance
n_pc  n_building
5     145
6     117
7      77
4      38
8      32
9      22
3      11
...

Accounted for 99% variance
n_pc  n_building
18    64
17    60
20    51
19    46
15    45
```

```
16    44
21    35
...
```

Weather station and number of PCs

```
ICAO num_pc_95_percent num_pc_99_percent
-------------------------------------------------
KLCH 8                 22
KDDH 6                 19
KPLN 7                 20
KCXY 5                 17
...
```

For the experiment building "OR0033PE", a plot of the number of pcs and the varience accounted for are shown in Figure 6

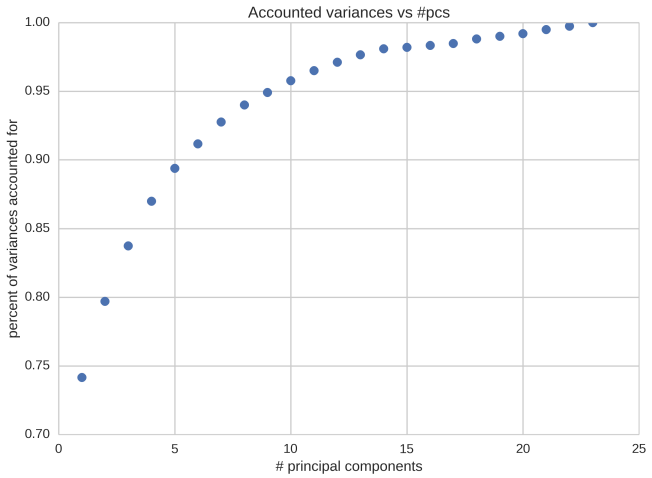


Figure 6: Accounted variance and the number of pcs

3.4 Use PCs as independent variables for the regression

For the test building "OR0033PE", using 1 to 10 PCs with ordinary least square linear regression, the error achieved are (note: this part is implemented using Python `sklearn.decomposition.PCA`, the "accounted variance" is different from the previous session when using `linalg.eig`)

```
num_PC accounted variance mse
1      0.765992063661     16.5442958987
2      0.81918354379      17.5088737364
```

```
3       0.852882590053      17.5473191338
4       0.879556033684      17.668850597
5       0.899650396008      17.9211677598
6       0.916688494142      17.9019603561
7       0.932492047507      18.0952033615
8       0.942872262315      18.1477157171
9       0.952484921675      18.1519075386
```

The smallest prediction error is when only using the first principal component (Figure 7), but it is still dramatically larger than the other methods.

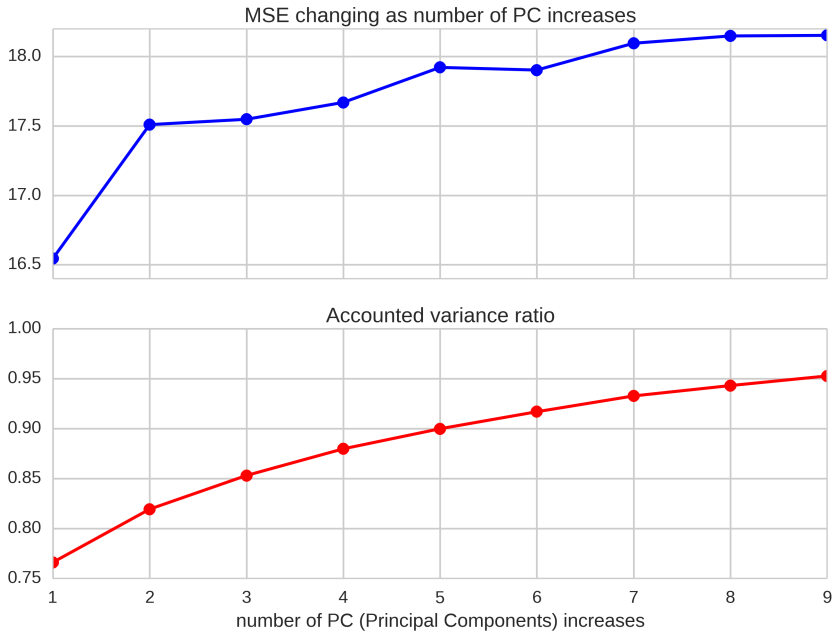


Figure 7: Error vs number of principal components

3.5 Proposed method 3: Auto-encoder with non-linear activation function

An auto-encoder contains an encoding function ϕ, and a decoding function ψ. It minimizes the re-construction error: $\|X - \psi(\phi)X\|^2$

To be implemented...

3.6 Proposed method 4: Fused lasso with degree day base 40F to 80F

Fused lasso is "a generalization that is designed for problems with features that can be ordered in some meaningful way" [20], penalizing the difference of adjacent coefficients.

4 Summary result

model	MSE
piecewise linear	2.07631642389
ridge regression	1.37049790455
PCA	16.5442958987
auto encoding	to be implemented...

Table 1: result of testing the methods above on building "OR0033PE"

5 next stage

5.1 Problem specification

- using a set of features to predict hourly energy
- using a set of features to predict monthly energy we know the duration of the energy record
- using a set of features to predict hourly energy we do not know the duration of the energy record

5.2 Broad idea

From the discussing with Professor Matt Gormley, the broad approach should be: first create a rich feature set with all potentially related features included, and use a non-linear model on the rich feature set so that the training data can be nearly perfectly predicted. Then applying some regularization to also drive down the test error. Finally try to interpret the model by evaluating the accuracy drop by leaving each feature out, or by incrementally adding a feature in random order and evaluate the accuracy gain by adding that feature.

5.3 Collect input variables and representation

A list of features to be included in the next stage model are

- Environmental variable
  - outdoor air temperature: exact hourly temperature, Radio Basis Function Kernel (RBFs), or temperature with exponential smoothing
  - Humidity
    - relative humidity (RH) as exact measurement or as with exponential smoothing
    - dew point temperature
  - solar radiation (W/m^2)
  - Wind speed (scaler)
- Occupancy
  - Operation schedules
  - Occupancy ratio (ratio of occupied vs non-occupied days)
- Building type
- Time
  - weekday vs weedend
  - hour of day
  - day of week
  - time lag (k), the number of previous readings to include in the model
  - unit circle representation of time of day, week, month, and year
- Energy
  - power (W, it's an auto-regressive component: use energy to predict energy) of previous k time steps.
  - fuel type: Electric vs non-electric
- Floor area
- pre-retrofit and post-retrofit period

5.4 Non-linear models

- Neuron Network with 1-2 hidden layer
- Support Vector Regression with RBF kernel
- Random forest regression [24]
- piecewise linear regression as baseline (it's a simple non-linear model, but not expressive enough)

References

[1] Bass Abushakra et al. An inverse model to predict and evaluate the energy performance of large commercial and institutional buildings. In *Building Simulation*, volume 3, pages 403–410, 1997.

[2] Matthew Brown, Chris Barrington-Leigh, and Zosia Brown. Kernel regression for real-time building energy analysis. *Journal of Building Performance Simulation*, 5(4):263–276, 2012.

[3] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.

[4] Bing Dong, Cheng Cao, and Siew Eang Lee. Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings*, 37(5):545–553, 2005.

[5] Melissa Donnelly, Jim Kummer, and Kirk Drees. Lean energy analysis using regression analysis to assess building energy performance. `http://www.eia.gov/tools/faqs/faq.cfm?id=58&t=8`, 3 2013. Accessed: 2016-10-20.

[6] IP Edition. Ashrae handbook-fundamentals, 2013.

[7] Margaret F Fels. Prism: an introduction. *Energy and Buildings*, 9(1-2):5–18, 1986.

[8] Jessica Granderson. Evaluation of the predictive accuracy of five whole building baseline models, 2014.

[9] JS Haberl and S Thamilseran. A bin method for calculating energy conservation retrofit savings in commercial buildings, 1994.

[10] Stig Hammarsten. A critical appraisal of energy-signature models. *Applied Energy*, 26(2):97–110, 1987.

[11] John Kelly Kissock. *A methodology to measure retrofit energy savings in commercial buildings*. PhD thesis, UMI, 2008.

[12] John Kelly Kissock, Jeff S. Haberl, and David E. Claridge. Inverse modeling toolkit: Numerical algorithms for best-fit variable-base degree day and change point models., 2003.

[13] David JC MacKay. Bayesian non-linear modeling for the prediction competition. In *Maximum Entropy and Bayesian Methods*, pages 221–234. Springer, 1996.

[14] Energy Star Portfolio Manager. Climate and weather. `https://portfoliomanager.energystar.gov/pdf/reference/Climate%20and%20Weather.pdf`. Accessed: 2016-10-13.

[15] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, 6:91–99, 2016.

[16] Ari Rabl and Anne Rialhe. Energy signature models for commercial buildings: test with measured data and interpretation. *Energy and buildings*, 19(2):143–154, 1992.

[17] T Agami Reddy, Namir F Saman, David E Claridge, Jeff S Haberl, WD Turner, and AT Chalifoux. Baselining methodology for facility-level monthly energy use-part 1: Theoretical aspects. *TRANSACTIONS-AMERICAN SOCIETY OF HEATING REFRIGERATING AND AIR CONDITIONING ENGINEERS*, 103:336–347, 1997.

[18] David M Solomon, Rebecca Lynn Winter, Albert G Boulanger, Roger N Anderson, and Leon Li Wu. Forecasting energy demand in large commercial buildings using support vector machine regression. *Department of Computer Science, Columbia University, Tech. Rep. CUCS-040-11*, 2011.

[19] Rob Tibshirani. Regularization: Ridge regression and the lasso, 2006.

[20] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[21] Wikipedia. Exponential smoothing. `https://en.wikipedia.org/wiki/Exponential_smoothing`. Accessed: 2016-12-17.

[22] Wikipedia. Principal component regression. `https://en.wikipedia.org/wiki/Principal_component_regression`. Accessed: 2016-12-07.

[23] Wikipedia. Radial basis function. `https://en.wikipedia.org/wiki/Radial_basis_function`. Accessed: 2016-12-17.

[24] Wikipedia. Random forest. `https://en.wikipedia.org/wiki/Random_forest`. Accessed: 2016-12-17.

[25] Matt Wytock and J Zico Kolter. Contextually supervised source separation with application to energy disaggregation. *arXiv preprint arXiv:1312.5023*, 2013.

[26] Zhun Yu, Fariborz Haghighat, Benjamin C.M. Fung, and Hiroshi Yoshino. A decision tree method for building energy demand modeling. *Energy and Buildings*, 42(10):1637 – 1646, 2010.

[27] Yuna Zhang, Zheng O'Neill, Bing Dong, and Godfried Augenbroe. Comparisons of inverse modeling approaches for predicting building energy performance. *Building and Environment*, 86:177 – 190, 2015.