Machine Learning

Modelling Consumption of Smart Grid using Regression Methods

M.Sc. student,
Aggelos Aggelidakis

Semester Project
2014

Smart Grid

1. An intelligent electricity delivery system
   - Energy suppliers and consumers interconnect through a network
2. Smart meters: monitor energy consumption
   - Installed at home and business
   - Monitor energy consumption
   - Give feedback to energy providers
3. Energy providers
   - Renewable sources (wind-turbines, photovoltaic panels)
   - Track energy consumption
   - Reduce energy consumption when demand gets too high
4. Storage devices
   - High storage batteries
   - Deposit energy for future use

Any Advantages?

- Reduce energy consumption
- Reduce cost of consumers
- Help business reduce carbon emissions
- Reduce cost of blackout
- New opportunities for tech companies – more jobs

Any needs for predictive models?

- When demand gets too high
- When prices per kilo-watt-hour increases
- When production is larger than consumption, storage needs

Predict consumption given price of electricity

- Observe behaviour of consumers proportional to price
- Large consumption during low-cost periods of time
- Predictive model using
  - Gaussian Process Regression
  - Bayesian Linear Regression

Why Bayesian Linear Regression?

Assume

- linearity
- independence
- constant variance

Benefits

- Avoid over-fitting problem of maximum likelihood
- Lead to automatic methods of determining model complexity using training data alone

Building Model

Basis function

- polynomial

Evidence Approximation Learning α, β

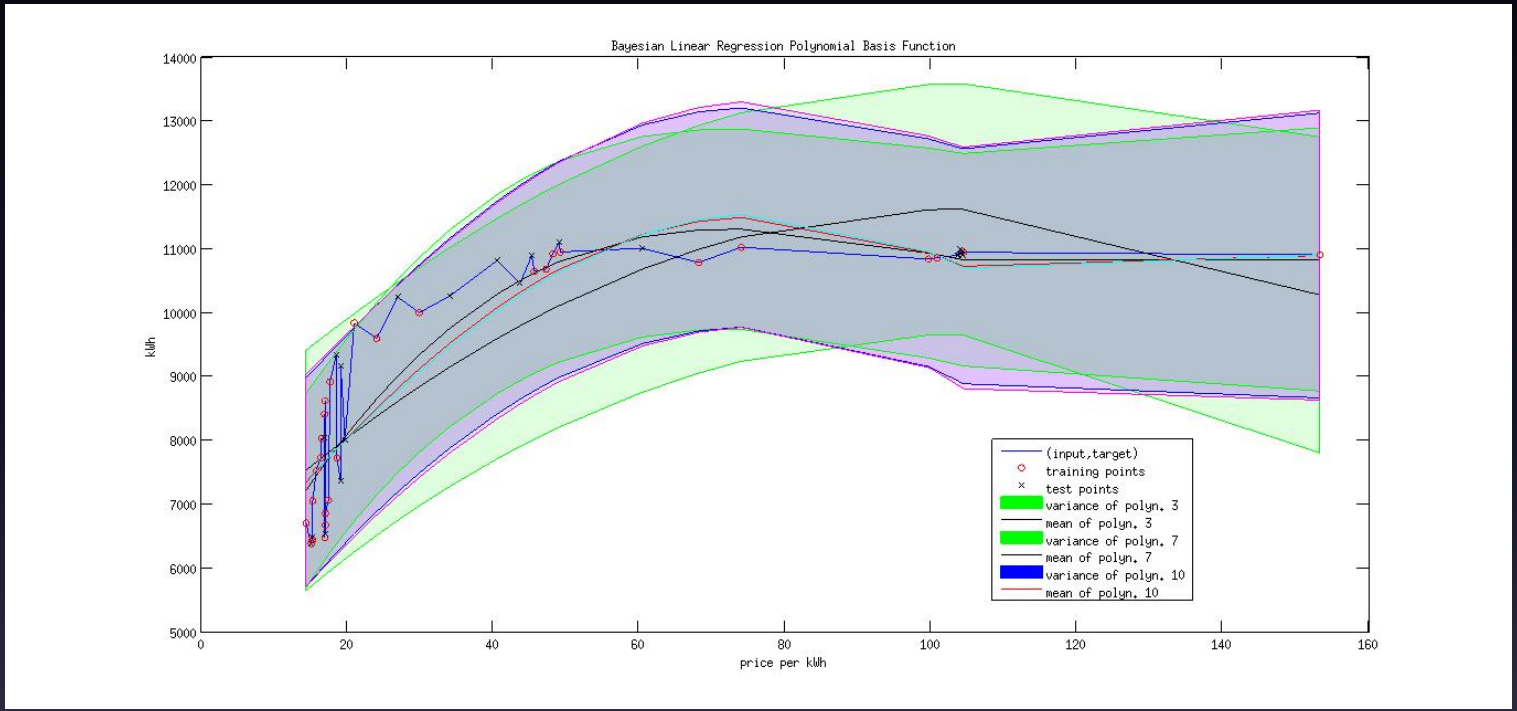
- Predictive Model: $p(t|x, t, \alpha, \beta) = N(t|m_N^T \phi(x), \sigma_N^2(x))$
- α and β are set to values determined by maximizing marginal likelihood functions obtained by integrating over parameters w
- *log marginal likelihood* $:=$
  $\frac{N}{2} \ln(\alpha) + \frac{M}{2} \ln(\beta) - E(m_n) - \frac{1}{2} \ln|A| - \frac{N}{2} \ln(2\pi)$

Cross-Validation

- 80% training data and 20% testing data
- degree of polynomial with the minimum average MSE is D=3

Degree of Polynomial	MSE
1	0.012617
2	0.007474
3	0.003761
4	0.004039
5	0.004260
6	0.004426
7	0.004549
8	0.004638
9	0.004703
10	0.004751
11	0.004787
12	0.004814

Table : Mean Square Error bayesian linear regression


Bayesian Linear Regression Polynomial Basis Function
kWh
price per kWh
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
0
20
40
60
80
100
120
140
160
(input,target)
training points
test points
variance of polyn. 3
mean of polyn. 3
variance of polyn. 7
mean of polyn. 7
variance of polyn. 10
mean of polyn. 10

Gaussian Process Regression

Kernels

$$2^{nd}\,degree\ polynomial : k(x_n, x_m) = \theta_0 + \theta_1(x^T z) + \theta_2(x^T z)^2 \qquad (1)$$

$$gaussian : k(x_n, x_m) = \theta_0\ \exp\left(-\frac{(x-z)^T(x-z)}{2(\theta_1)^2}\right) \qquad (2)$$

- evaluate the predictive distribution $p(t_{N+1}|t_N)$
- joint distribution $p(\mathbf{t}_{N+1}) = N(t_{N+1}|0, C_{N+1})$

$$C_{N+1} = \begin{bmatrix} C_N & k \\ k^T & c \end{bmatrix}$$

- where C_N is the N $\times$ N covariance matrix
- vector k has elements $k(x_n, x_{N+1})$ for $n = 1, \ldots, N$
- scalar $c = k(x_{N+1}, x_{N+1}) + \beta^{-1}$

$p(t_{N+1}|t)$ is also **Gaussian distribution**

$$m_{x_{N+1}} = k^T C_N^{-1} t \qquad \sigma^2(x_{N+1}) = c - k^T C_N^{-1} k$$

Learning Hyperparameters

- maximize log likelihood function
  $ln\ p(t|\theta) = -\frac{1}{2} ln|C_N| - \frac{1}{2} t^T C_N^{-1} t - \frac{N}{2} ln(2\pi)$
- gradient of loglikelihood
  $\frac{\partial}{\partial \theta_i}\ ln\ p(t|\theta) = -\frac{1}{2} Tr(C_N^{-1} \frac{\partial C_N}{\partial \theta_i}) + \frac{1}{2} t^T C_N^{-1} \frac{\partial C_N}{\partial \theta_i} C_N^{-1} t$

Maximization of the log likelihood can be done using efficient gradient-based optimization algorithms

Choose $\mathbf{x}$;
Compute $\bigtriangledown f(\mathbf{x}), f(\mathbf{x})$;
while $\| \bigtriangledown f(\boldsymbol{x}) > tol \|$ **do**
  $t := 1$;
  while $f(\boldsymbol{x} + t\Delta\boldsymbol{x}) > f(\boldsymbol{x}) + \alpha t \bigtriangledown f(\boldsymbol{x})^T \Delta x$ **do**
    $t := \beta t$
  end
  Update $x := x - t \bigtriangledown f(x)$;
  Compute $\bigtriangledown f(\mathbf{x}), f(\mathbf{x})$;
end

Algorithm 1: Backtracking line search

Choose $\mathbf{x}$;
Compute $\bigtriangledown f(\mathbf{x}), f(\mathbf{x})$;
while $\| \bigtriangledown f(\boldsymbol{x}) > tol \|$ **do**
  Compute $t_{start} > 0$ that minimizes $f(x - t \bigtriangledown f(x))$;
  Update $x := x - t \bigtriangledown f(x)$;
  Compute $\bigtriangledown f(\mathbf{x}), f(\mathbf{x})$;
end

Algorithm 2: Gradient descent method

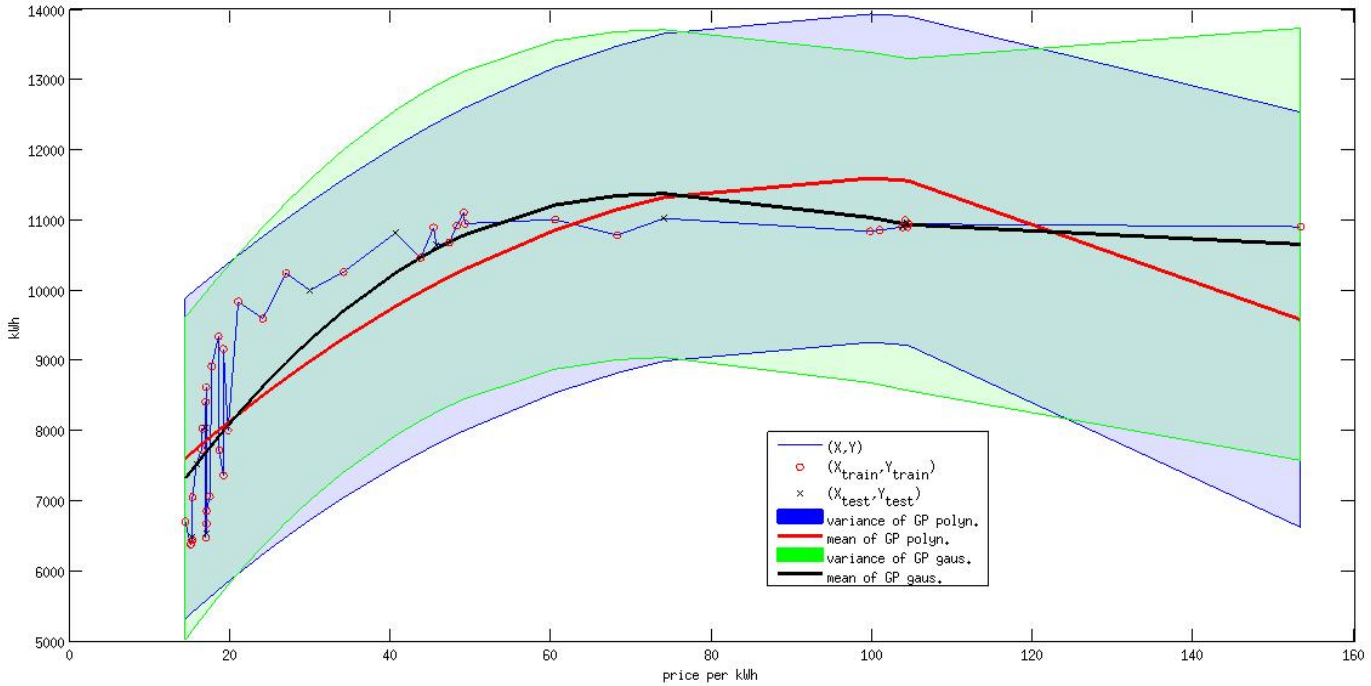
Cross-Validation

- Polynomial or Gaussian?

MSE polynomial	MSE gaussian
0.0055	0.00531
0.0037	0.0159
0.0070	0.0059
0.0044	0.0042
0.0083	0.0040
0.0073	0.0148
0.0089	0.0036
0.0068	0.0036
0.0063	0.0045
0.0053	0.0052

GP polynomial kernel	0.006698
GP gaussian kernel	0.006359

Table : Average Mean Square Error

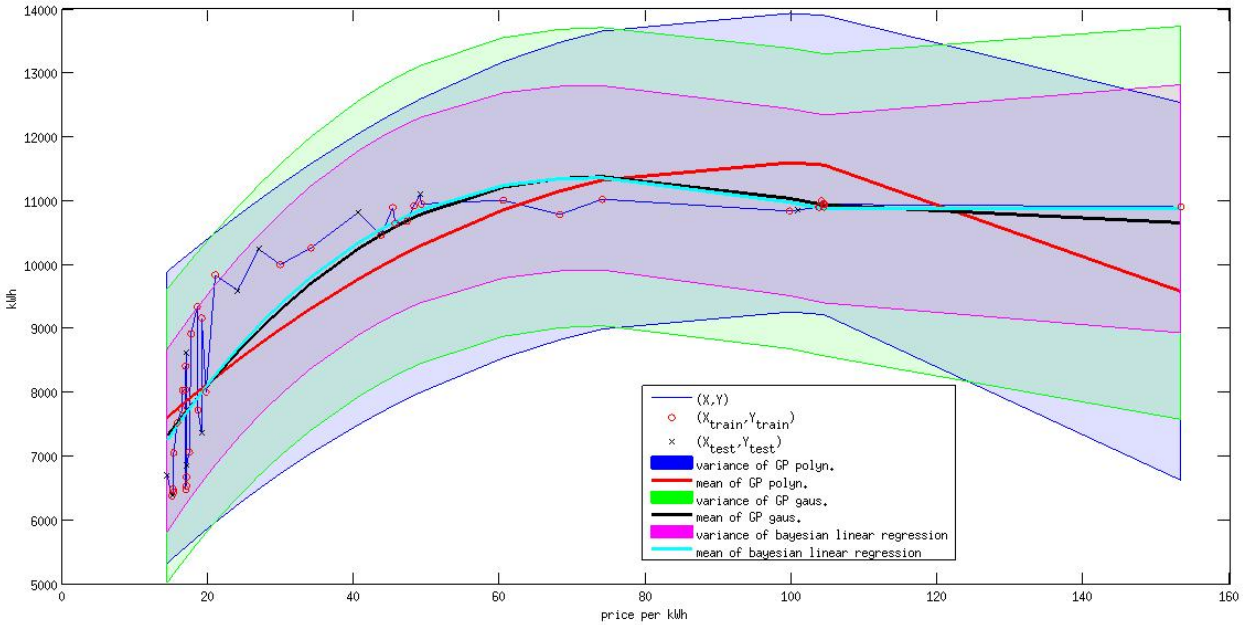

14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
kWh
0
20
40
60
80
100
120
140
160
price per kWh
(X,Y)
(X_train,Y_train)
(X_test,Y_test)
variance of GP polyn.
mean of GP polyn.
variance of GP gaus.
mean of GP gaus.

GP (gaussian, polynomial kernels) vs. Bayesian Linear Regression

- Gaussian or Bayesian Linear Regression?

Regression Models	MSE
Bayesian Linear Regression	0.004468
GP polynomial kernel	0.006698
GP gaussian kernel	0.006359

Table : Mean Square Error


14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
kWh
0
20
40
60
80
100
120
140
160
price per kWh
(X,Y)
(X_train,Y_train)
(X_test,Y_test)
variance of GP polyn.
mean of GP polyn.
variance of GP gaus.
mean of GP gaus.
variance of bayesian linear regression
mean of bayesian linear regression

Thank you
Any Questions?