Power Curve Modeling

Haiyang Zheng
Intelligent Systems Laboratory
The University of Iowa

Outline
- k-NN (k-nearest neighbor) algorithm
- Wind farm power modeling based on k-NN and PCA (Principle Component Analysis)
- Parametric models of wind farm power

k-NN algorithm
- Represent each instance in a multi-dimension space.
- Divide the entire data set into training and test data sets.
- Given a test instance, a distance metric is computed between the test instance and all training instance, then the k nearest neighbors are selected from the training data.
- Compute the average values of the k nearest neighbors. This value becomes the predicted value for the test instance.

Small Example of k-NN
Assume x1 and x2 are the two measurements of the final product quality, how to use k-NN for product quality judgment?

<table>
<thead>
<tr>
<th>No</th>
<th>X1= Strenght</th>
<th>X2= Smooth</th>
<th>Y= Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>Bad</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
<td>Good</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>Bad</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>?</td>
</tr>
</tbody>
</table>

Algorithm Schemes
- Different distance metrics are used, e.g. Euclidean, Manhattan.
- The parameter k is significant in k-NN algorithm and its best value depends on the data structure and conditions.
- Euclidean distance metric is selected and k is set to 100 based on the model’s prediction accuracy.
Euclidean distance

- Two data points P and Q represented in Euclidean n-space,

\[ P = (p_1, p_2, \ldots, p_n) \]
\[ Q = (q_1, q_2, \ldots, q_n) \]

The Euclidean distance between P and Q is expressed as following:

\[ d(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \]

K=1, 1-nearest neighbor

- Data set 1-4 are the training data set, and data set 5 is the test data set.
- Compute the distance between the test data and training data.

\[ D_1 = \sqrt{(5 - 2)^2 + (5 - 3)^2} = 3.6 \]
\[ D_2 = \sqrt{(1 - 2)^2 + (2 - 3)^2} = 1.4 \]
\[ D_3 = \sqrt{(6 - 2)^2 + (7 - 3)^2} = 5.6 \]
\[ D_4 = \sqrt{(3 - 2)^2 + (5 - 3)^2} = 2.2 \]
- \( D_2 \) is the smallest, and thus data point 2 is chosen as the 1-nearest neighbor.
- Thus, the quality is bad.

K=3, 3-nearest neighbors

\[ D_5 = \sqrt{(5 - 2)^2 + (5 - 3)^2} = 3.6 \]
\[ D_2 = \sqrt{(1 - 2)^2 + (2 - 3)^2} = 1.4 \]
\[ D_3 = \sqrt{(6 - 2)^2 + (7 - 3)^2} = 5.6 \]
\[ D_4 = \sqrt{(3 - 2)^2 + (5 - 3)^2} = 2.2 \]
- Data points 1, 2, and 4 are chosen as 3-nearest neighbors. Two training data vote for bad, and one for good. The quality of 5 is still bad.
- Assume the value of good is 1, and bad is 0.

\[ y_5 = \frac{(1 + 1 + 0)}{3} = 0.6 \]

A typical power curve of a single turbine

Cumulative power curve of the wind farm

The wind farm contain 100 turbines, 89 turbines with good quality data are selected

Data Description

<table>
<thead>
<tr>
<th>Data set</th>
<th>Start Time Stamp</th>
<th>End Time Stamp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/1/06 12:00 AM</td>
<td>1/31/06 11:50 PM</td>
<td>Total data set; 4547 observations</td>
</tr>
<tr>
<td>2</td>
<td>1/1/06 12:00 AM</td>
<td>1/25/06 6:20 AM</td>
<td>Training data set; 3476 observations</td>
</tr>
<tr>
<td>3</td>
<td>1/25/06 6:30 PM</td>
<td>1/31/06 11:50 PM</td>
<td>Test data set; 871 observations</td>
</tr>
</tbody>
</table>

10-minute SCADA data
Data Format

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>…..</th>
<th>X89</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind speed of Turbine 1</td>
<td>Wind speed of Turbine 2</td>
<td>Wind speed of Turbine i</td>
<td>Wind speed of Turbine 3</td>
<td>Wind farm power</td>
</tr>
</tbody>
</table>

Preprocess the data and prepare them into the format for Data Mining Software, e.g., Weka and Statistica.

k-NN (k=100)

MAE (Mean absolute error): 2872 kW
Std (Standard deviation of absolute error): 2949 kW

Small Example of Wind Farm Power Modeling

<table>
<thead>
<tr>
<th>Time Stamp</th>
<th>Turbine_1</th>
<th>Turbine_2</th>
<th>Turbine_3</th>
<th>Turbine_4</th>
<th>Total Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/06 12:00 AM</td>
<td>7.96</td>
<td>8.92</td>
<td>8.78</td>
<td>7.17</td>
<td>3556.85</td>
</tr>
<tr>
<td>1/1/06 12:10 AM</td>
<td>8.35</td>
<td>8.49</td>
<td>9.06</td>
<td>6.89</td>
<td>3621.85</td>
</tr>
<tr>
<td>1/1/06 12:20 AM</td>
<td>8.5</td>
<td>8.4</td>
<td>9.12</td>
<td>7.02</td>
<td>3499.33</td>
</tr>
<tr>
<td>1/1/06 12:30 AM</td>
<td>8.34</td>
<td>8.4</td>
<td>9.12</td>
<td>7.02</td>
<td>?</td>
</tr>
<tr>
<td>1/1/06 12:40 AM</td>
<td>7.98</td>
<td>8.5</td>
<td>9.44</td>
<td>6.75</td>
<td>?</td>
</tr>
</tbody>
</table>

K=2, Euclidean distance

<table>
<thead>
<tr>
<th>Euclidean distance</th>
<th>1/1/06 12:00 AM</th>
<th>1/1/06 12:10 AM</th>
<th>1/1/06 12:20 AM</th>
<th>1/1/06 12:30 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/06 12:40 AM</td>
<td>0.89</td>
<td>0.58</td>
<td>0.67</td>
<td>0.56</td>
</tr>
</tbody>
</table>

The data at time 1/1/06 12:10 AM and 1/1/06 12:30 AM are chosen as 2 nearest neighbors. The average total power at these two time points is the prediction value of wind farm power.

Total power = (3514.91 + 3499.33) / 2 = 3507.12 kW

The actual total power at time 1/1/06 12:40 AM is 3512.05 kW.

Feature reduction and transformation

- The k-NN model built before has 89 inputs, and thus the input dimensionality need to be reduced.
- The principal component analysis (PCA) was chosen to do feature transform and reduction.
- The PCA expresses the variance-covariance structure of a set of variables by a few linear combinations.
Basic steps of PCA

- Compute a correlation matrix.
- Compute the eigenvectors and eigenvalues of the correlation matrix.
- Select the components to form an eigenvector.
- Derive the new data comprised of the principal component of the original data.

<table>
<thead>
<tr>
<th>Value Number</th>
<th>Eigen value</th>
<th>Total Variance (%)</th>
<th>Cumulative Eigen value</th>
<th>Cumulative (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85.17</td>
<td>95.70</td>
<td>85.17</td>
<td>95.70</td>
</tr>
<tr>
<td>2</td>
<td>0.54</td>
<td>0.61</td>
<td>95.72</td>
<td>96.31</td>
</tr>
<tr>
<td>3</td>
<td>0.36</td>
<td>0.41</td>
<td>96.09</td>
<td>96.73</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.23</td>
<td>96.29</td>
<td>96.96</td>
</tr>
<tr>
<td>5</td>
<td>0.18</td>
<td>0.20</td>
<td>96.48</td>
<td>97.17</td>
</tr>
<tr>
<td>6</td>
<td>0.13</td>
<td>0.17</td>
<td>96.63</td>
<td>97.34</td>
</tr>
<tr>
<td>7</td>
<td>0.18</td>
<td>0.15</td>
<td>96.77</td>
<td>97.40</td>
</tr>
<tr>
<td>8</td>
<td>0.13</td>
<td>0.14</td>
<td>96.90</td>
<td>97.64</td>
</tr>
<tr>
<td>9</td>
<td>0.09</td>
<td>0.11</td>
<td>97.11</td>
<td>97.88</td>
</tr>
</tbody>
</table>

Plot of Eigenvalues of correlation matrix

k-NN-P (integrated k-NN and PCA) Model

\[ p_i = \sum \hat{c}_i x_i \]

- \( p_i \): the PF (principal factor)
- \( \hat{c}_i \): the wind speed of 89 turbines
- \( x_i \): the weight for feature transformation

Wind farm power model based on k-NN-P algorithm

MAE: 2255 kW; Std: 2174 kW

Power curve based on PCA

Observed power curve as the function of the principal component derived from 89 wind speeds.
Nonlinear parametric modeling of wind farm power curves

- Logistic function

\[ y = f(x; \theta) = \frac{1 + e^{-\theta_1 x_{\tau}}} {1 + e^{-\theta_1 x_{\tau}}} \]

\[ \theta = (a, m, n, c) \]

Learning Parametric model from training data

\[ y = f(x; \theta) = \frac{1 + e^{-\theta_1 x_{\tau}}} {1 + e^{-\theta_1 x_{\tau}}} = \theta = (a, m, n, c) \]

\( x \): is the principal component of 89 wind speeds

\( y \): the power of the wind farm

\( \theta \): is a 4-dimension vector parameter of logistic function that determines the shape of the power curve

\[ S_{\text{error}} = \sum_{i=1}^{N} \left( \frac{1 + e^{-\theta_1 x_{\tau}-\theta_2 x_{\tau} - \theta_3 x_{\tau} + \theta_4 x_{\tau}}}{1 + e^{-\theta_1 x_{\tau}}} - y_i \right)^2 \]

\[ \hat{\theta} = \arg\min_{\theta} S_{\text{error}}(x(1), y(1), \ldots, x(N), y(N)|a, m, n, c) \]

The basic steps of the evolutionary strategy algorithm

1. Initialize \( N \) individuals (candidate vector parameter) to form the initial parent population.
2. Repeat until the stopping criteria are satisfied.
   2.1: Select from the parent population and recombine \( \lambda \) times to generate \( \mu \) children.
   2.2: Mutate \( \mu \) children.
   2.3: Select the best \( N \) individuals from the children and parent pool based on the fitness function values.
   2.4: Use these \( N \) selected individuals as parents for the next generation.
3. Apply one of the stopping criteria, e.g., the allowable number of generations.

Convergence process of the fitness function

Power curve generated from the observed and the parametric model