Wind Energy Prediction with Machine Learning

- Nils André Treiber (CI Uni-Oldenburg) -

December 7, 2012
Overview

1. Computational-Intelligence Group


3. Different ways of Modeling: Physical and statistical models

4. Used data set

5. Excursion: SVM/SVR

6. Our approach
   - Implementation
   - First Results
   - Modification

7. Future work
The Computational-Intelligence Group

Research:

- Machine Learning
- Evolutionary Optimization
- Energy Systems (Wind & Solar)

Jun.-Prof. Dr. Oliver Kramer
Dr. Fabian Gieseke
Dipl.-Phys. Nils Treiber
B.Sc. Björn Wolff
Wind Energy Prediction

Motivation:

- strong increase in renewable energy
- but: integration of wind turbines into the power grid results in some problems caused by high fluctuations in production
- short term wind energy prediction is important to schedule spinning reserves and reserve energy
Modeling

There are two different approaches for wind prediction:

Physical models:
- often very complex differential equations
- computationally heavy to run
- difficult to calibrate

Statistical models:
- different tools and methods to analyze spatio-temporal data
- algorithms derive functional dependencies directly from the observations (only data-driven)
- in complex terrains usually the only successful possibility of making wind prediction
Wind power: Data from turbines

Observation: Still quite difficult to get data

Used Datasets:
- data from NREL, which are designed for a wind integration study in the western part of the US
- based on about 32,000 simulated wind turbines (farms)
- with a time-resolution of ten minutes => 52,560 entries per year

http://www.nrel.gov
Excursion: Support Vector Machine

- Classifier: divides a set of multidimensional input patterns \( x_i \) in two possible classes \( y_i \in \{-1, +1\} \) with a gap that is as wide as possible:

\[
\begin{align*}
\text{Space of Features} & \rightarrow \text{Space of Labels} \\
X \subseteq \mathbb{R}^d & \rightarrow Y \\
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} & \rightarrow y \in \{-1, +1\}
\end{align*}
\]

- after training with labeled patterns, the SVM can predict the output for unseen ones

Example for a 2-dim input vector
Inseparable Patterns
For non-linearly separable problems there is a special trick:

- the basic idea is to map the input vector $\mathbf{x}$ into a higher dimensional feature space by a nonlinear mapping function \textit{(kernel function)} and then to perform a linear classification in this feature space
- the ratio between a good representation of the data and the complexity of the model is set by \textit{regularization parameter} $\lambda$.

Support Vector Regression (SVR)

- for regression scenarios, the space $\mathcal{Y}$ is given by $\mathbb{R}$
- the goal of the learning process consists in finding a prediction function $f : \mathcal{X} \rightarrow \mathbb{R}$ that maps the unseen patterns to reasonable real-valued labels

![Images showing different values of $\lambda$ with corresponding regression results]
Implementation

- formulate the wind forecasting task as regression problem
- for the prediction of the production of the target $y$ at time $t^* = t + t_h$ take in account the past wind energy measurements at time $t - 1, t - 2, ..., t - \mu$ with $\mu \in \mathbb{N}$ of the surrounding turbines $x_i$
- train the model with a set $T = \{(x_1^1, y_1^1), ..., (x_N^1, y_N^1)\}$

**Figure**: Example: Setup with 5 inputs, each with his considered 3 past measurements. This results in 15-dim input vector $x$. 
Experimental Settings

- each time horizon requires a selection of neighbored turbines with adequate distances
- to automatize this selection, all wind turbines within a given radius are picked
- then dividing the circle into segments and rings
- selection of the wind turbines, which are closest to the center of the cells
- determine the best parameters for radius and the number of rings and segments with a manual parameter study

Figure: red points: available turbines; blue points: centre of each cell; green points: selected turbines
Training & Results

- use of linear and rbf-kernels
- perform a crossvalidation for the regularization parameter $\lambda$ and for the bandwidth (rbf-kernel)
- training of the SVR on the first 9 months of 2006 and evaluation on the last quarter of 2006
- comparison with the 'naive' prediction, based on the assumption, that the wind blows in e.g. 20 min as strong as now
Quantitative Results

- determination of the square error of the forecasts:

\[ \sum_{i=t_{\text{start}}}^{t_{\text{end}}} \left( p_{\text{measured}}^i - p_{\text{predicted}}^i \right)^2 \]

- SVR-based prediction significantly better than the naive prediction

<table>
<thead>
<tr>
<th>terrain</th>
<th>improvement of L2-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>complex</td>
<td>10.9 %</td>
</tr>
<tr>
<td>flat</td>
<td>33.2 %</td>
</tr>
</tbody>
</table>

Table: 30 min forecast

- further experiments show: predictions for up to 2 h lead to good results
Feature Selection

Question: Can the prediction be improved, if only the best windturbines are used as input?

Approach: Iterative adding of turbines
- Find the best input $x_b$, that alone leads to the best prediction
- then add a further turbine $x_f$, so that you get the best prediction with this input-ensemble $(x_b, x_f)$
- add more and more turbines

![Graph showing abs. L2-error vs ensemble size for SVR with rbf-kernel, complex terrain]
Future Work:

Tuning of the statistical model:

- prediction with specials kernels
- tests with other methods/algorithms [k-Nearest-Neighbors,...]

Finally: Build up a hybrid-model

- combination of statistical and physical(meteorological) models