Wind Power Prediction with Cross-Correlation Weighted Nearest Neighbors

Nils André Treiber, Oliver Kramer[1]

Abstract

A precise wind power prediction is important for the integration of wind energy into the power grid. Besides numerical weather models for short-term predictions, there is a trend towards the development of statistical data-driven models that can outperform the classical forecast models [1]. In this paper, we improve a statistical prediction model proposed by Kramer and Gieseke [5], by employing a cross-correlation weighted k-nearest neighbor regression model (x-kNN). We demonstrate its superior performance by the comparison with the standard u-kNN method. Even if different pre-processing steps are considered, our regression technique achieves a comparably high accuracy.

1. Introduction

In the past, Kramer and Gieseke developed a prediction model that is exclusively based on wind power time series measurements [5]. In this model the predictions task is formulated as multivariate regression problem that considers the time series of neighboring turbines for a particular target turbine. In this line of research, we employed linear regression, support vector regression and k-nearest neighbor (kNN) regression. The main result is that there is in general no regression technique, which provides the highest accuracy for all cases. Rather, it depends on the surrounding terrain and the wind conditions, which method provides the most accurate prediction. In addition, the selection of appropriate features has an important part to play [7].

In this paper, we introduce a new method, called x-kNN model. The cross-correlation between the particular neighboring and the target turbine defines its contribution to the prediction in kNN regression. We compare our model with the uniform kNN method (here called u-kNN), whose most appropriate features have been found by testing all possible pattern combinations. Moreover, the exclusive consideration of the cross-correlation of the turbines is discussed. In addition, we show that our model is also superior if the inputs of the classical regression model have been reduced by a principle component analysis (PCA).

2. Related Work

In his thesis, M. Hall demonstrates that an appropriate feature selection can be performed via a correlation analysis [3]. Thereby, features should be highly correlated with the label, but uncorrelated with the label. A correlation-based kNN algorithm for classification tasks is introduced by Xinran and Xiang [6]. In a more general way, the contribution of the neighbors for classification and regression tasks can be weighted with regard to the distance in feature space, so that nearer neighbors contribute more than more distant ones. This implementation is discussed in the paper written by Dudani [2].

[1] University of Oldenburg, 26111 Oldenburg, Germany, nils.andre.treiber@uni-oldenburg.de and oliver.kramer@uni-oldenburg.de, Department of Computing Science

3. Wind Data Set and Prediction Model

3.1. General Time Series Model and u-kNN Regression

We formulate the prediction as regression problem. Let us first assume we want to predict the power production of a target turbine only with its time series: The wind power measurement $\boldsymbol{x}^i = p_t(t^i)$ (pattern) is mapped to the power production at target time $y^i = p_t(t^i + \lambda)$ (label). For our regression model, we assume to have *N* of such pattern label pairs that are basis of our training set T={$(\boldsymbol{x}^1, y^1)$, …, $(\boldsymbol{x}^N, y^N)$} and allow via a regression to predict the label for unknown patterns. One can assume that this model generates better predictions, if more information of the time series is used. For this reason, we extend the pattern by adding past measurements $p(t-1), \ldots, p(t-\mu)$ with $\mu \in \mathbb{N}^+$.

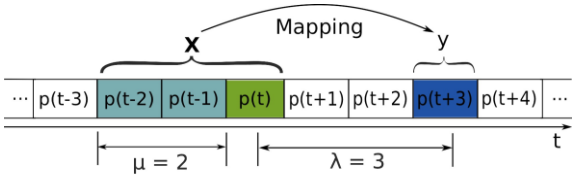

Figure 1: General time series model. The parameter λ describes the time horizon of the prediction.

To catch some spatial-temporal dependencies, we also take the features of the times series of *m* neighboring turbines into account, which are generated in the same way as for the target turbine.

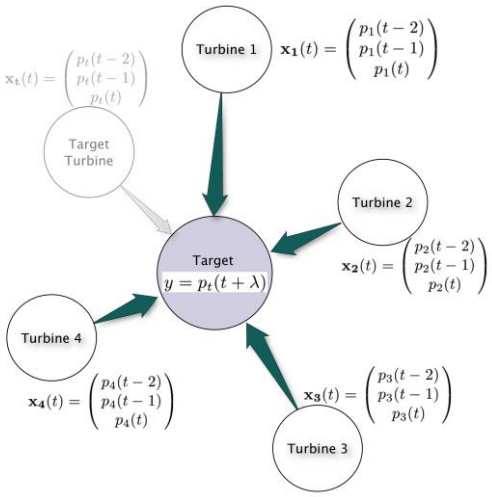

Figure 2: Setup with 4 neighboring turbines and 3 past measurements. This results in a $15-$*dimensional pattern* $[(4+1)\cdot 3 = 15]$.

After defining the mapping, the goal is now to find a regression function f that provides good predictions to an unseen pattern $\boldsymbol{x}'$. In the u-kNN regression method, the output of $\boldsymbol{x}'$ depends on the k-nearest patterns in the training set, found by calculating the Euclidian distance $dist(\boldsymbol{x}^i, \boldsymbol{x}')$ to all existing patterns. Finally, the label is given as arithmetic average of the corresponding *k* labels:

$$f_{kNN}(\boldsymbol{x}') = \frac{1}{k}\sum\nolimits_{i \in \mathcal{N}_k(\boldsymbol{x}')} y_i$$

with the set $\mathcal{N}_k(\boldsymbol{x}')$ that contains the k-nearest neighbors of $\boldsymbol{x}'$.

3.2. Data Set

For our experiments, we use the freely available wind data set of the National Renewable Energy Laboratory (NREL), which is part of an integration study for wind and solar energy in the western part of the United States [8]. On the NREL website, a GUI allows to download the time series of the generated power with a time resolution of 10 minutes for 32,043 turbines. In this paper, we decide to predict the output of three arbitrarily selected wind turbines near Casper (WY, ID: 23167), Comanche (WY, ID: 8419) and Tehachapi (CA, ID: 4155).

4. Cross-Correlation Weighted kNN

Before we go into detail, it is worth mentioning that the measured power measurements p_k of all turbines are in the same interval $p_k \in [0\,MW,\ 30MW]$. Therefore, a normalization preprocessing has not to be done, regardless of the particular kNN implementation. In our x-kNN variant, the inputs do not contribute equally to the label, but are weighted according to the cross-correlation with the target:

$$x_{corr}(p_t, p_k) = \frac{\sum_{i=1}^{N} p_k\left(t^i\right) p_t(t^i + \lambda)}{\sqrt{\sum_{i=1}^{N}\left(p_k(t^i)\right)^2 \cdot \sum_{i=1}^{N}\left(p_t(t^i + \lambda)\right)^2}}$$

If the cross-correlation coefficient is high, the turbine gets a major influence for the prediction by expanding the corresponding dimension in the regression model. Each feature of the pattern is weighted with the cross-correlation coefficient to the power of l:

$$x_k = p_k \cdot |x_{corr}(p_t, p_k)|^l$$

with an index k representing a neighboring turbine $j = 1, 2, \ldots, m$ or the target itself. The power l controls the strength of the cross-correlation weighting.

5. Experimental Details and Results

5.1. Training and Evaluation Details

Every tested kNN model is trained by using data from the year 2004. To speed up the training process, only every fourth time step is taken into account. Despite a smaller training set it is guaranteed that wind conditions at different seasons are included. For finding an appropriate value of the parameter *k*, we implement a two fold cross validation and test values of *k* in the interval *k=[10, 20, 30, …, 130].* For the evaluation, we test our models on the year 2005 and determine the mean square error (MSE) of the forecasts y^i with the measured power outputs $p_t(t^i)$ for the N forecasts:

$$MSE = L_2\left(p_t(t^i) - y^i\right) = \frac{1}{N}\sum_{i=t_{start}}^{t_{end}}\left(p_t(t^i) - y^i\right)^2$$

We compare the MSE of our prediction models with the persistence model assuming that the wind speed does not change in the forecast horizon. This naïve model is quite successful for short time horizon predictions and not easy to beat [9].

In the experiments, we do not include past measurements, i.d. $\mu = 0$. The additional neighboring turbines are selected with an automatic algorithm that determines turbines around the target, which are spatially well distributed. With a focus on one hour predictions and given mean wind speeds of

about 8.5 m/s, we consider neighboring turbines in a distance of 30 km from the target turbine. Finally, we determine *m=12* neighboring turbines for each target.

5.2. Measurement of the Cross-Correlation between Target and Neighboring Turbines and the Target Itself

Before we compare the different regression models, we determine the cross-correlation coefficient $x_{corr} \in [-1, 1]$ between the time series of the target and the neighboring turbines $x_{corr}(p_t, p_j)$ and between the target itself $x_{corr}(p_t, p_t)$ for all measurements in year 2004.

Table 1 shows that there are high correlations $x_{corr}(p_t, p_j)$ for the turbines in Comanche. In contrast, the correlation in Tehachapi varies quite strongly.

turbine	$x_{corr}(p_t, p_t)$	$x_{corr}(p_t, p_j)$ min	$x_{corr}(p_t, p_j)$ max	$x_{corr}(p_t, p_j)$ mean
Casper	0.902	0.529	0.800	0.678
Tehachapi	0.930	-0.029	0.775	0.555
Comanche	0.873	0.785	0.826	0.815

Table 1: Cross-correlation coefficient for the target itself $x_{corr}(p_t, p_t)$ and for the target and the neighboring turbine $x_{corr}(p_t, p_j)$. The coefficient is shown for the turbine *j* with the lowest/highest correlation. The last column includes the average correlation coefficient.

5.3. Results for x-kNN Regression

After computing the cross-correlation coefficient, we employ our x-kNN implementation and MSE of an one hour ahead prediction for the three target turbines. Hereby, the pattern includes the features of all 12 available neighboring turbines. Table 2 shows the results, achieved with a cross-correlation weighting parameter of $l = 5$.

	Casper	Tehachapi	Comanche
x-kNN: MSE $[MW]^2$	22.963	17.803	17.525
Pers.: MSE $[MW]^2$	26.066	20.715	24.274

Table 2: Results for the introduced x-kNN regression. For comparison, the results of the persistence model are also shown.

One can observe that the x-kNN prediction achieves a considerably higher accuracy than the persistence model. The increase in accuracy for the turbine near Comanche is very high (27.8%). In the following, we compare x-kNN to three further approaches: (1) u-kNN with naïve input set tuning, (2) u-kNN with input sets tuned with regard to cross-correlation, and (3) dimensionality reduction as pre-processing for u-kNN.

5.4. Results for u-kNN Regression with Naïve Input Set Tuning

Table 3 shows the MSE achieved with the u-kNN regression and with a) no neighboring turbine, b) all 12 turbines, and c) with the best subset of turbines determined with an exhaustive search of all possible input combinations, i.e. $\sum_{m=0}^{12} \binom{12}{m} = 4096$.

u-knn	Casper	Tehachapi	Comanche
MSE $[MW]^2$: a)	24.884	20.006	22.869
MSE $[MW]^2$: b)	24.948	20.226	17.673
MSE $[MW]^2$: c)	23.258	17.588	17.501

Table 3: Results for the u-kNN regression with particular input sets: a) univariate times series model (no neighboring turbines), b) multivariate model considering all additional turbines in the neighborhood and c) multivariate model with most appropriate inputs.

One can observe that taking into account a particular subset of neighboring turbines is useful for the prediction. But even with the best subset the u-kNN method only achieves in mean a comparable high accuracy with our x-kNN.

5.5. Results for u-kNN Regression with Inputs Selected with Regard to Cross-Correlation

In the following experiments, we analyze the accuracy of the u-kNN when taking into account only the turbines with a high cross-correlation with the target. The results are given in Table 4.

turbine	l=1	l=3	l=5	l=7	l=9	l=11
Casper	23.905	23.538	24.030	24.424	24.889	24.917
Tehachapi	19.625	18.970	18.806	19.759	20.515	20.217
Comanche	19.539	18.627	18.549	18.482	17.637	17.649

Table 4: MSE in $[MW]^2$ for the u-kNN regression with the l highest cross-correlated neighboring turbines. The target turbine measurement is always part of the pattern, because it has the highest correlation with the prediction value, see Table 1.

It can be seen that this u-kNN variant does not allow good predictions. In particular, for the turbines in Casper and Tehachapi, the accuracy is much worse in comparison to the results achieved with x-kNN.

5.6. Results for u-kNN Regression with Various Numbers of Principle Components

In further experiments, we employ PCA [9] as preprocessing method and test a different number of components for the u-kNN prediction model. We determine the principle components of the measurements corresponding to the year 2004 and train the regression model with these features. In the evaluation part, we have to transform the measurements according to the principle components we computed on the train set, before we make the prediction on the test set. Table 5 shows the results.

turbine	c=1	c=3	c=5	c=7	c=9	c=11
Casper	42.345	30.478	25.658	24.556	24.702	24.836
Tehachapi	39.029	29.578	24.964	24.784	20.558	20.004
Comanche	18.199	17.842	17.648	17.680	17.671	17.677

Table 5: MSE in $[MW]^2$ for the u-kNN regression with PCA reduced features. Parameter c identifies the number of components taken into account.

It can be observed that the PCA in general does not yield competitive results. While the results for Comanche are still quite accurate, they are comparable for Casper and Tehachapi with the relatively inaccurate results achieved with the univariate u-kNN model. For both turbines the forecast fails completely, if only few components are taken into account.

6. Conclusion

Wind power prediction is a key technology for the successful integration of wind energy into the grid, because it allows to plan reserve plants, battery loading strategies and scheduling of the different authorities. In this paper, we present a special kNN regression method based on a weighting of inputs with regard to the cross-correlation between neighboring turbines and the target turbine.

We demonstrate that the exclusive consideration of the cross-correlation for u-kNN is not sufficient for a good prediction. In contrast, our efficient implementation provides robust and precise predictions, which can only be achieved for equally weighted inputs after an extensive pre-selection of turbines.

Acknowledgements

We thank the presidential chair of the University Oldenburg and the EWE research institute NextEnergy for partly supporting this work. Further, we thank the US National Renewable Energy Laboratory (NREL) for providing the wind data set.

References

[1] Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen and H., Feitosa, E., "A review on the young history of the wind power short-term prediction", *Renewable and Sustainable Energy Reviews* 12(6), 2008

[2] Dudani, S. A., "The distance-weighted k-nearest-neighbor rule", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-6 , pp. 325-327, 1976

[3] Hall, M. A., "Correlation-based feature selection for machine learning", *Doctoral Thesis*, University of Waikato, 1999

[4] Jackson, J. E., "A user's guide to principle components", *John Wiley and Sons*, 1991

[5] Kramer, O. and Gieseke, F., "Short-term wind energy forecasting using support vector regression", *Proc. SOCO*, pp. 271-280, 2011

[6] Li, X. and Xiang C., "Correlation-based k-nearest neighbor algorithm", *Proc. ICSESS*, pp-185-187, 2012

[7] Treiber, N.A. and Kramer, O., "Evolutionary turbine selection for wind power predictions", *Proc. KI*, 2014

[8] Potter, C.W., Lew, D. McCaa, J., Cheng, S., Eichelberger, S., and Grimit, S., "Creating the dataset for the western wind and solar integration study", *Wind Engineering 32(4)*, pp. 325-338, 2008

[9] Wegley, H., Kosorok, M., Formica, W., "Subhourly wind forecasting techniques for wind turbine operations", *Technical Report, Pacific Northwest Lab.*, 1984