A Study of Machine Learning Techniques for Daily Solar Energy Forecasting using Numerical Weather Models

Ricardo Aler, Ricardo Martín, José M. Valls, and Inés M. Galván

Carlos III University - Computer Science Department,
Avenida de la Universidad, 30 - 28911 Leganés (Madrid), Spain
aler@inf.uc3m.es

Abstract. Forecasting solar energy is becoming an important issue in the context of renewable energy sources and Machine Learning Algorithms play an important rule in this field. The prediction of solar energy can be addressed as a time series prediction problem using historical data. Also, solar energy forecasting can be derived from numerical weather prediction models (NWP). Our interest is focused on the latter approach. We focus on the problem of predicting solar energy from NWP computed from GEFS, the Global Ensemble Forecast System, which predicts meteorological variables for points in a grid. In this context, it can be useful to know how prediction accuracy improves depending on the number of grid nodes used as input for the machine learning techniques. However, using the variables from a large number of grid nodes can result in many attributes which might degrade the generalization performance of the learning algorithms. In this paper both issues are studied using data supplied by Kaggle for the State of Oklahoma comparing Support Vector Machines and Gradient Boosted Regression. Also, three different feature selection methods have been tested: Linear Correlation, the ReliefF algorithm and, a new method based on local information analysis.

1 Introduction

Photovoltaic systems are becoming important sources of energy in electricity networks. However, electric utility companies are required to guarantee electricity supply within certain ranges which is difficult given the fluctuating nature of weather conditions. Thus, accurate forecasts of solar radiation is becoming an important issue in the context of renewable energy sources. An approach to forecasting is to use statistical and machine learning techniques based on historical data of solar production [5]. With respect to the machine learning techniques, many works appear in the literature, that use for instance Artificial Neural Networks [11] or Support Vector Machines [15, 3]. However, for the prediction horizons required by photovoltaic plants (day-ahead), it has been shown that models based on Numerical Weather Prediction (NWP) systems, such as the Global Forecast System (GFS) and the European Centre for Medium-Range Weather Forecast (ECMWF), are a good alternative [5]. These global models

predict some meteorological variables for points in a low resolution grid. NWP predicted variables have been used as input for machine learning techniques mainly for wind power prediction [2, 12] and recently for solar energy forecasting [18, 8].

Here, we are interested on the problem of predicting incoming solar energy from NWP models computed from the NOAA/ESRL Global Ensemble Forecast System (GEFS). GEFS provides short-term forecasting for several meteorological variables, for different points or nodes located in a grid. For this paper, we use the data supplied by Kaggle [1] where the goal was to predict the total daily solar energy at 98 Oklahoma solar sites using 15 NWP variables every three hours for a 16×9 grid. In principle the closest grid nodes to the solar station should be the most relevant for prediction, but it can be useful to know how prediction accuracy improves as more and more GEFS grid nodes are used as input for the machine learning techniques. However, using the variables from a large number of grid nodes can result in many attributes which might worsen the generalization capabilities of the learning algorithms. Therefore, our second goal is to study the performance of different feature selection algorithms on prediction accuracy.

The rest of the article is structured as follows: Section 2 describes de data and the regression and feature selection methods used in this work. Section 3 shows the experimental results including the preliminary studies and parameter adjustment, the study of the influence of the number of grid nodes, and the study of feature selection methods. Finally, section 4 provides the conclusions and future work.

2 Data and Methods

2.1 Description of Data

The data available from the Kaggle website has been provided by the American Meteorological Society for the 2013-14 Solar Energy Prediction Contest. The goal is to predict the total daily incoming solar energy, measured in $J \times m^2$, at 98 sites of the Oklahoma Mesonet network. The input data for each day corresponds to the output of the numerical weather prediction model GEFS using 11 ensemble members and 5 forecast timesteps from 12 to 24 hours in 3 hour increments. Each ensemble member produces outputs for 15 different meteorological variables for each timestep and each point of a 16×9 uniform land-surface grid covering the State of Oklahoma. Some of the 15 meteorological variables are the following: accumulated precipitation ($kg.m^{-2}$), air pressure (Pa), downward and upward shortwave/longwave radiation ($W.m^{-2}$), cloud cover (%), temperature (K), etc. A more detailed information can be found in [1]. Thus, the number of attributes for each grid node is $11 \times 5 \times 15 = 875$. Since the number of grid points is $16 \times 9 = 144$, the total amount of available data for each day equals 118800. Data has been collected everyday from 1994 to 2007 (5113 days) in association with the corresponding accumulated incoming solar energy, which is the attribute

[1] https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest

to be predicted. This accumulated incoming solar energy (in $J \times m^2$) has been calculated by summing the solar energy measured by a pyranometer at each mesonet site every 5 minutes, from the sunrise to 23:55 UTC of the corresponding date.

From the total input-output available data covering 14 years, we have used the period 1997-2005 as the training set (4380 days), reserving the period 2006-2007 (733 days) for the testing set.

2.2 Regression Methods

Support Vector Machines (SVM) [4] is a class of supervised learning method extensively applied to classification and regression problems. SVMs basically construct maximum margin hyperplanes and use kernel functions to build non-linear models. The Kernel functions most used are linear, polynomial, and the Radial Basis Function (RBF) kernels. Accuracy is greatly influenced by the cost parameter C and the kernel parameters (σ in the case of the most commonly used kernel, the RBF). A more detailed information about SVM can be found in [16, **?**]. In this work, we have used the WEKA SVM implementation called SMO [9].

Gradient Boosted Regression (GBR) is a recent machine learning technique that has shown considerable success in predictive accuracy. The method was proposed by Friedman [6, 7] and it produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. GBR uses two algorithms: regression trees are from the classification and regression tree (decision tree) group of models, and boosting (an adaptive method for combining many simple models to give improved predictive performance) builds and combines a collection of models. Like SVM, the accuracy of GBR models depends on some parameters, as the number of trees used, the shrinkage (a regularization parameter) and the depth of trees. A more detailed description can be found in [14]. We have used for experiments the gbm package [17] from the R language [13].

2.3 Attribute Selection Methods

In this work, different attribute selection algorithms have been used. They are feature weighting algorithms because they assign weights to input attributes individually, depending on their relevance to the target, and rank them based according to these weights. Two of them are well known algorithms, linear correlation and the ReliefF algorithm [10]. The third one is a new algorithm based on local information analysis, which is described below.

The linear correlation attribute selection method ranks attributes according to their linear correlation with the target. The ReliefF algorithm [10] is also a feature weighting algorithm that estimates the quality of attributes in problems with strong dependencies between attributes. The estimation of the quality of attributes is made according to how well their values distinguish between instances that are near to each other. It evaluates an attribute by repeatedly sampling an

instance and considering the value of the given attribute for the nearest instance of the same and different class. In our work, the algorithm implementation for the Weka [9] tool (ReliefFAttributeEval) has been used as the attributes evaluator and Ranker method as search method, which ranks the attributes by their individual evaluations.

Attribute Selection Algorithm based on Local Information Analysis The algorithm divides the input space into a grid of fixed-size square regions, called cells. In this work, the algorithm maps all possible subsets of 1 and 2 attributes into grids of dimension *dim* 1 and 2, respectively. Attributes are ranked according to an evaluation function F_I that measures the information contained in the attributes in each of the cells. Information in a cell is measured as the number of patterns in the cell belonging to class $C1$ that cannot be explained by chance alone, assuming a binomial distribution with parameters (n, p), where n is the total number of patterns in the cell and p is the ratio of $C1$ instances in the whole training set. Thus, given a Cell and a value Conf of the confidence parameter, the information is measured as in Eq. 1 :

$$F_I(\text{Cell}, \text{Conf}) = \max(\text{Cell.C1} - \text{IDF}(\text{Cell.Total}, \text{Conf}, (N_{C1}/N)), 0.0) \quad (1)$$

where Cell.C1 and Cell.C0 are the number of patterns that belong to $C1$ and $C0$ in the cell, respectively; Cell.Total = Cell.C0 + Cell.C1; N_{C1} is the number of $C1$ patterns in the whole training set; N is the number of training patterns; and IDF is the inverse binomial distribution function with parameters n = Cell.Total and $p = N_{C1}/N$; Conf measures the confidence that the distribution of patterns within the cell has been generated by chance. The algorithm uses different values of the confidence parameter from ConfMin to ConfMax with a step $\Delta = 0.25$. The specific algorithm steps are as follows:

1. A vector VectorRanking with NAtrs dimension is initialized to zero, being NAtrs the total number of attributes in the problem.
2. A matrix MatrixInfo with dimension NAtrs $\times C$ is also initialized to zero, being C the number of confidence values used, i.e. $C = (\text{ConfMax} - \text{ConfMin})/\Delta$.
3. Starting with $dim = 2$, a grid of 4^{dim} cells is obtained by dividing the interval $[0, 1]$ in 4 parts. For each combination of *dim* attributes, the values of attributes are mapped into every cell of the grid.
4. The information provided by each combination of *dim* attributes in each cell is estimated for each confidence value by using $F_I(\text{Cell}, \text{Conf}_i)$ (Eq. 1), being $\text{Conf}_i = \text{ConfMin} + i * \Delta$ and $i = 1, ..., C$. That information is stored in MatrixInfo.
5. The attribute with highest information for confMax in MatrixInfo (the last column of matrix MatrixInfo) is assigned a rank of NAtrs. Next attribute is assigned a rank of NAtrs -1 and so on. This process is continued as long as information is strictly larger than zero. When information is zero, the next confidence value (ConfMax $- \Delta$) is used, and the process is repeated until all attributes have been ranked. The ranking is accumulated in VectorRanking.

6. Steps 2 to 5 are repeated for single attributes, i. e. $dim = 1$ (4 cells in the grid).
7. Attributes based on values stored in VectorRanking are ordered. Thus, they are ordered by decreasing relevance.

The algorithm assumes that the output is binary, i.e. patterns belong to two classes, $C0$ and $C1$. For regression, the problem is transformed into 10 binary problems, by discretizing the output value in 10 intervals. The attribute selection algorithm is applied to each problem and the ranking of the 10 set of attributes are combined.

3 Experimental Results

In this work, SMO and GBM have been used to approximate the solar energy production. First, for each solar station, models have been built using the information provided by the 16 nearest grid nodes and then, an attribute selection procedure is carried out. Before running the models, some preliminary studies have been done in order to decide aspects related with the information provided by GEFS and, also, to decide some important parameters of the machine learning algorithms.

3.1 Preliminary Studies and Parameter Adjustment

As it has been mentioned in section 2.1, data provided by GEFS includes 11 ensemble output forecasting models. Using the 11 ensembles as input variables to ML algorithms would imply to build up 11 regressors for each mesonet station. On the other hand, it is not obvious what ensemble model should be chosen. In this work, three different approaches to combine the 11 ensemble models have been considered: compute the mean of the 11 ensemble models, compute the median, and compute the mode. The three approaches have been run using the information provided by the 5 nearest grid points and the average of MAE (mean absolute error) for the 98 mesonet stations are 1940816, 1955128 and, 1979554, respectively. Therefore, we have decided to use the mean of the 11 ensemble models to summarize the information provided by all the ensembles.

On the other hand, the accuracy of SMO and GBM models depends highly of their parameters. To establish the optimum parameter values for each mesonet and for each possible number of grid nodes would involve a very heavy computation. Then, the parameters of models have been selected using only the first of the 98 mesonet (ACME station) and the five nearest grid nodes. A two-year validation dataset has been used to compare the different parameter combinations. An exhaustive grid search has been run to locate the optimal parameters (the cost parameter C and σ for SMO, and number of trees, shrinkage, and tree depth for GBM). Experiments established that for SMO with linear kernel (linear-SMO), the best parameter is $C = 0.03$. For SMO with RBF kernel (RBF-SMO) the best parameters are $C = 1$ and $G = 0.01$. For GBM models, they are: number of trees=5000, shrinkage=0.01, and tree depth=10. Those parameters have been used for all the experiments in the next sections.

3.2 Prediction accuracy with respect to the number of GEFS grid nodes

Figure 1 displays the evolution of MAE as the number of GEFS grid points is increased from 1 to 16 for linear-SMO, RBF-SMO, and GBM. Averaged train and test MAE for 98 solar sites are shown on the left and right figures of 1, respectively.

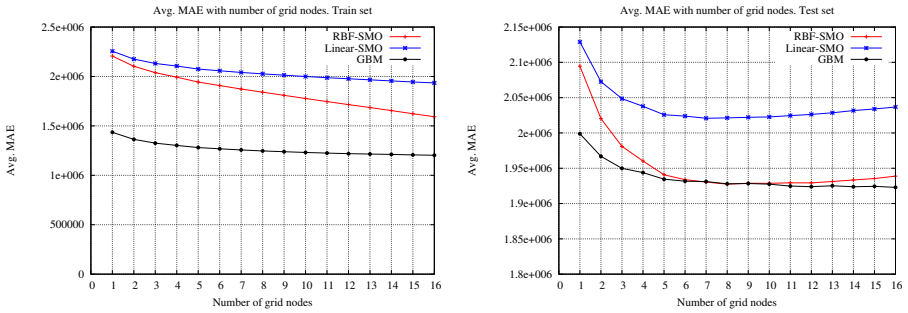


Fig. 1. Average MAE for different number of grid nodes, using linear-SMO, RBF-SMO, and GBM. Training and testing set

With respect to test MAE, it can be seen that the two non-linear models GBM and RBF-SMO perform significantly better than the linear one (linear-SMO). In all cases, it is observed that MAE tends to improve as the number of grid points increases. Both RBF-SMO and GBM obtain similar results when the number of grid points is large (8 or more). But GBM performs better when only a few grid points are used (from 1 to 4) and does not suffer from the slight overfitting observed for SMO for more than 10/11 GEFS grid points.

The main conclusions from this study are that non-linear models perform much better than the linear one, and that interestingly, the best results are obtained using more than the closest four or five grid nodes (i.e. the grid nodes surrounding the station): the minimum error is obtained from 8 grid points for RBF-SMO and from 16 points for GBM (although the gain obtained by GBM from 8 to 16 points is very small: a 0.26% decrease).

3.3 Study of feature selection methods

Here, the three feature selection algorithms have been applied to all the features present in 16 grid points (16*75=1200 features). The 1200 attributes are ranked and both RBF-SMO and GBM algorithms are trained and tested using the first 400, 500, 600, 800, 900, 1000 attributes, respectively. Figures 2 and 3 display the average MAE for training and testing for the 98 stations obtained using the different numbers of attributes.

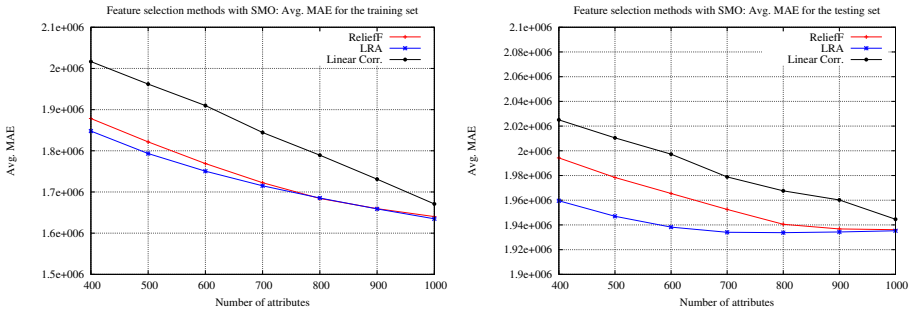


Fig. 2. Average MAE for different number of attributes, using RBF-SMO. Training and testing set

Results show that, surprisingly, although the original number of attributes is very large, the different attribute selection methods do not improve prediction error in general. Therefore, in this domain all 1200 attributes seem relevant to some degree. However, results also show that the number of attributes can be greatly reduced without loosing a significant accuracy. In the case of RBF-SMO, the local information analysis algorithm allows to reduce the number of attributes from 1200 to 600 and obtain the same error (1938241 with 600 features vs. 1938855 with all features). In this case, ReliefF and linear correlation obtain higher errors for the same number of (600) attributes (1965442 and 1997274, respectively). When GBM is used as regressor, the number of features cannot be reduced to the same extent but with 800 features, ReliefF and the local information algorithms are able to obtain quite similar errors compared to the full set of features (1926369 and 1932069, respectively, versus 1922594 for the 1200 features). In all cases linear correlation is not competitive with the other methods.

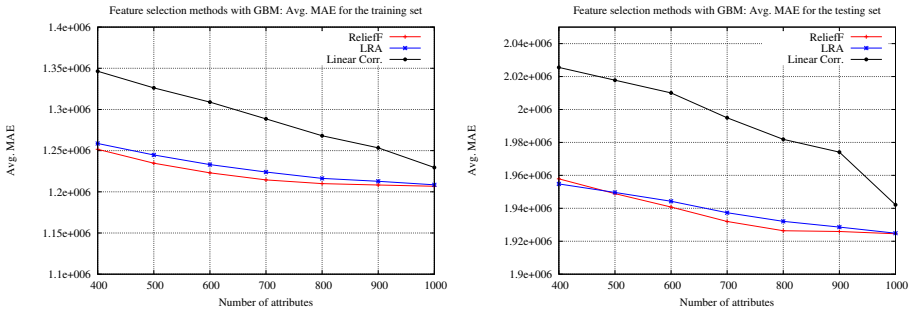
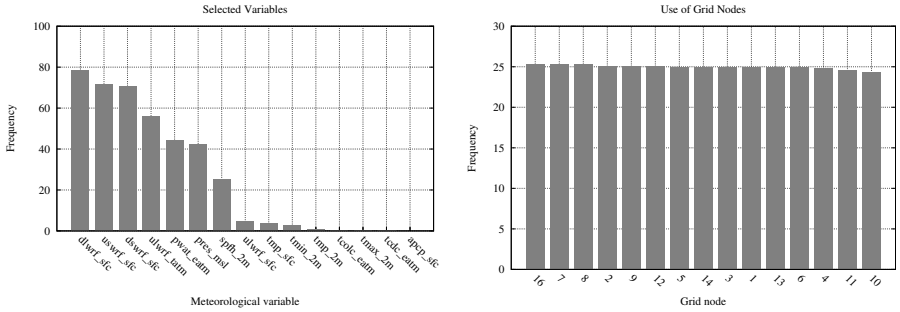


Fig. 3. Average MAE for different number of attributes, using GBM. Training and testing set

Finally, we will take advantage of the attribute ranking performed by the attribute selection methods in order to know which are the most relevant meteorological variables and the most relevant grid points. For this purpose, we have used the local information algorithm to select the 400 most relevant features. Figure 4 displays bar graphs of the variable names and the grid points used, from 1 (the closest) to 16, respectively. Figure 4 (a) shows a clear preference for some of the variables (downward long-wave radiative flux average at the surface, upward short-wave radiation at the surface, downward short-wave radiative flux average at the surface, and upward long-wave radiation at the top of the atmosphere, ...). However, the flatness of graph 4 (b) shows no preference for closer vs. farther away grid nodes: all grid points have about the same amount of attributes present in the 400 most relevant attributes.



(a) (b)

Fig. 4. (a) Bar graph of Meteorological Variables. (b) Bar graph of GEFS grid nodes

4 Conclusions

In this work, we have perform an study of different machine learning techniques in the context of solar energy forecasting using NWP models computed from the NOAA/ESRL Global Ensemble Forecast System (GEFS) for different nodes located in a grid. On one hand, three different regression methods (linear SVM, RBF-SVM, and GBM) have been used to build forecasting models and to study the influence of the grid nodes number on prediction accuracy. On the other hand, given the large number of features in this domain, three different attribute selection methods have been tested (linear correlation, ReliefF, and a local information analysis algorithm).

Experimental results show that the non-linear methods obtain lower errors than the linear one. GBM and RBF-SMO perform similarly, although RBF-SMO shows some slight overfitting when the number of grid points is large. Also, in the case of the best performing method (GBM), forecasting accuracy tends to improve as the number of GEFS grid nodes used as input increases, even beyond the 4 or 5 closest nodes. Contrary to what was expected, feature selection was

not able to improve solar energy prediction, although with RBF-SMO, the local information algorithm can obtain similar predictions with a half of the attributes.

In the future, it would be interesting to extend this study to other situations where geographical or meteorological features are different (surface elevations, different pressure levels grid nodes) or to other prediction problems within the renewable energy domain involving grid numerical weather prediction models.

References

1. Forecast daily solar energy with an ensemble of weather models, 2013.
2. Carlos M. Alaíz, Alberto Torres, and José R. Dorronsoro. Sparse linear wind farm energy forecast. In *ICANN (2)*, pages 557–564, 2012.
3. Ji-Long Chen, Hong-Bin Liu, Wei Wu, and De-Ti Xie. Estimation of monthly solar radiation from measured temperatures using support vector machines–a case study. *Renewable Energy*, 36(1):413–420, 2011.
4. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
5. Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27:65–76, 2013.
6. Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.
7. Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.
8. Y. Gala, A. Fernández, and J. R. Dorronsoro. Machine learning prediction of global photovoltaic energy in spain. In *International Conference on Renewable Energies and Power Quality*, 2014.
9. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
10. Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *Machine Learning: ECML-94*, pages 171–182. Springer, 1994.
11. Adel Mellit and Alessandro Massi Pavan. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected {PV} plant at trieste, italy. *Solar Energy*, 84(5):807 – 821, 2010.
12. C Monteiro, R Bessa, V Miranda, A Botterud, J Wang, G Conzelmann, et al. Wind power forecasting: state-of-the-art 2009. Technical report, Argonne National Laboratory (ANL), 2009.
13. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
14. Matthias Schonlau. Boosted regression (boosting): An introductory tutorial and a stata plugin. *Stata Journal*, 5(3):330, 2005.
15. Navin Sharma, Pranshu Sharma, David Irwin, and Prashant Shenoy. Predicting solar generation from weather forecasts using machine learning. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 528–533. IEEE, 2011.
16. Vladimir N Vapnik. Statistical learning theory (adaptive and learning systems for signal processing, communications and control series), 1998.

17. Greg Ridgeway with contributions from others. *gbm: Generalized Boosted Regression Models*, 2013. R package version 2.1.
18. Björn Wolff, Elke Lorenz, and Oliver Kramer. Statistical learning for short-term photovoltaic power predictions. In *DARE: Data Analytics for Renewable Energy Integration. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2013.