A Survey of Regression Models for Predicting Photovoltaic Cell Power Production

Jack McShane
Indiana University, jamcshan@iu.edu

ABSTRACT

In recent decades, alternative means of energy production have become increasingly important with the advent of climate change. In particular, photovoltaic cells have become increasingly accessible to the public. In response, this project employs linear regression, ridge regression, decision tree regression, and ensemble regression learning models to predict the power output of localized solar cell installations with hopes for the models' later incorporation into automated systems geared at maximizing the generation and utilization of solar power generation.

Key Terms: Photovoltaic Cells, Machine Learning, Linear Regression, Ridge Regression, Decision Tree Regression, Tree Regression, Ensemble Learning.

INTRODUCTION

In recent decades, photovoltaic (PV) cells as an alternative means of energy production have become increasingly important with the advent of climate change. As advances in these technologies continue to increase individuals' access to them, small-scale residential installations of PV cells have become far more common. This project aims to evaluate the performance of a variety of less resource intensive machine learning (ML) algorithms on the task of power output prediction for small scale PV cell installations. The goal was to find machine learning models that can accurately predict PV cell power generation while being resource un-intensive to a large enough degree such that they can affordably be incorporated as an integral part of an automated system for maximizing PV cell power usage.

DATA COLLECTION

Data collection for this undertaking proved a challenge. While there are many data sets for solar power generation, few of them contain information about the weather conditions for the area in which the data was collected. The majority of solar power generation data sets that were encountered tracked power production for large areas, often on the scale of counties, states, or provinces. Weather conditions for localized regions were therefore rarely present. Instead, features such as longitude, latitude and region were the defining characteristics of these data sets. As well, many of the data sets that were found for more localized areas were incredibly small, often containing only a few hundred data points, making them impossible to use in machine learning applications.

The data set finally selected for use in the project comes from the University of California, Berkeley. The data set was curated by a PhD candidate by the name of Alexandra Constantin and describes the power generation of the University of California, Berkeley's campus solar installation over the course of 14 months in 3 hour intervals. The installation consists of several rooftops upon which the PV cells sit. The final data set contained weather condition measurements such as: the distance of the sun from solar noon, whether there was daylight during the course of the recorded period, and the average temperature throughout the day. This is the type of data that will be useful in the prediction of power output for localized solar installations rather than the aggregate power output of a region, and aligns well with the aim of this project.

METHODS

As the focus of the project was to survey the performance of various models in their capacity for predicting solar power generation, naturally, many different models were constructed. Through the course of the project, predictions for PV cell power output were generated using Linear Regression, Ridge Regression, and Decision Tree Regression models. The best performing approach was then used to instantiate the subsequent ensemble approach in an effort to glean the performance improvements could be gained with said ensemble approach. To evaluate the performance of the models and allow apt comparison, the error metric selected was the root mean squared error (RMSE).

I. Baseline

As a performance measure against which to compare the various models used, a baseline predictor needed to be established. Far from complex, the baseline predictor simply predicts the mean of the power generated by the solar cell installation during the course of the data collection, ie. the mean power generated of the whole data set.

II. Linear Least Squares Regression

The fist approach taken in the course of the project was simple linear regression. Linear regression, also referred to as linear least squares regression, is a commonly applied approach in machine learning that attempts to formulate the relationship between the independent variables of the data set and the dependent variable, in this case, the power generated. The model calculates the line of best fit for the n^{th}-dimensional space that the data set exists within.

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

(1) General Form of the Best Fit Line

Figure (1) gives the generalized form of this equation. The algorithm achieves this best-fit line by finding the coefficients for the line that minimize the sum of the squared errors between the power generated and the prediction of the line.

	Coefficient
Is Daylight	-8705.258894
Distance to Solar Noon	11773.938201
Average Temperature (Day)	-1835.599545
Average Wind Direction (Day)	1701.829704
Average Wind Speed (Day)	-1881.603239
Visibility	-1363.844411
Relative Humidity	-10035.518381
Average Wind Speed (Period)	6236.679093
Average Barometric Pressure (Period)	2764.010503
Hours from Noon	-39398.875078
Month_2	2260.623497
Month_3	4749.285515
Month_4	5736.662600
Month_5	7773.628081
Month_6	9256.990288
Month_7	8734.787885
Month_8	7346.301808
Month_9	4949.205767
Month_10	1855.363900
Month_11	304.722494
Month_12	-1261.865391
Sky Cover_1	787.244721
Sky Cover_2	207.118316
Sky Cover_3	-831.597379
Sky Cover_4	-3691.230064

(2) Table of Coefficients for the Line of Best Fit

Using five-fold cross validation to verify consistent performance over the course of training, a linear least squares regression model was trained on the training set. Figure (2), above, contains the coefficients of the best fit line found by the algorithm. The newly learned regression model was subsequently used to make predictions on the testing set for evaluation.

III. Ridge Regression

Ridge regression is an adaptation of linear least squares regression. Ridge regression uses the same approach for learning, namely minimizing the sum of the squared error between the true value and the prediction of the line, but its error calculation includes an additional term. This term is often referred to as the regularization term and introduces a penalty for the magnitude of the coefficients present in the equation of the estimator, or put another way, the slope of the best-fit line.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

(3) Error Equation for Ridge Regression

This additional penalty term can be seen in (3), above, where the first summation term is a simple error calculation and the second term is the penalty for the coefficient magnitudes. The additional penalty of the squared coefficients term is referred to as the regularization term because it improves the generalization of the model in that it reduces the model's sensitivity to changes in the input it receives, ie. it is less fine-tuned to the training data specifically.

	Root Mean Squared Error (RMSE)
Ridge Model 1	5134.958962
Ridge Model 2	5135.124769
Ridge Model 3	5136.927083
Ridge Model 4	5162.459400
Ridge Model 5	5357.103195

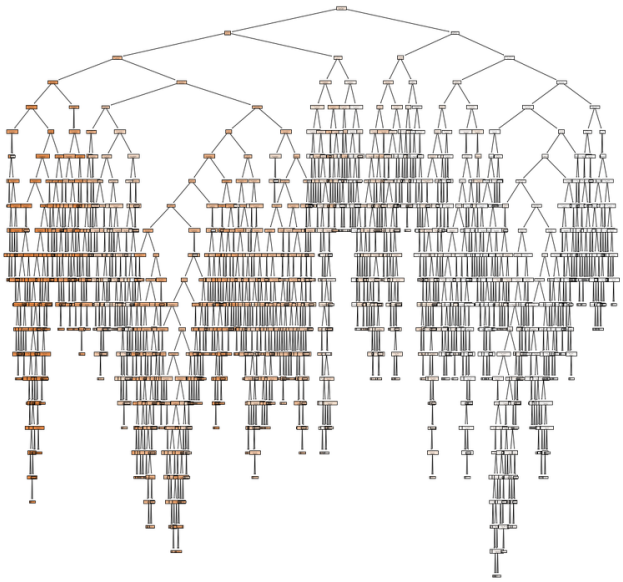
(4) Table of Ridge Regression Models' Performance

Several ridge regression models with lambda terms ranging from .001 to 10 were trained and tested. The scores of the resulting models can be seen in (4) above. Surprisingly, the least regulated model is the one which performed the best.

IV. Decision Tree Regression

Regression trees are an adaptation of the well known decision tree classifier. Fundamentally, the two work in the same manner. Both find splits in the data based on the maximum entropy among the features, ie. based on the feature that provides the most similarity between data points on either side of the proposed split. For example, the data set used contains a feature labeled 'Is Daylight'. In examining the feature's correlation with the amount of power generated, it was found that whenever it was daylight, there was absolutely no power generation on the part of the PV cells. When it was daylight, there was a non-zero value for the power generated by the cells. This would be a great feature to split on, as all values on one side of the split have a non-zero value and on the other, all have zero values for the generated power.

Where decision tree classifiers and regression trees differ is in their output. While the classifiers have to output a class label based on the most prominent class in a particular split, the regressor outputs a continuous value that captures the value of similar points. Figure (5) below shows the decision tree regressor that resulted from training.



(5) Figure of Decision Tree Regressor Model

The decision tree model is obviously quite complex and one of the big concerns with decision tree models is their aptitude for overfitting the training data. To alleviate that to some degree, cross validation was used when training the model. Firstly, because it helped to alleviate overfitting by slightly reducing the magnitude of training data seen and secondly, because it allowed monitoring of the learning process by providing validation scores for the model as it learned.

V. Gradient Boosted Decision Tree Regressors

Following the application of the above regression approaches, the performance of each variation was evaluated. The approach that fared best against the testing set was then selected for use in an ensemble approach. Of the three approaches above, linear regression, ridge regression, and tree regression, the best performing approach was the decision tree.

Using Scikit-learn's GradientBoostingRegressor, several ensemble models using regression trees as the base estimator and utilizing early stopping were trained against the data set. The variations trained were based on changes to three key parameters of the model: the max depth for each individual tree regressor in the ensemble, the learning rate, and the maximum number of regression trees to be in the ensemble. The results of the trials are depicted below in (6).

	Model Specifics	Root Mean Squared Error (RMSE)
DT Model 1	d=2, lr=0.001, ntrees=255	8482.699481
DT Model 2	d=2, lr=0.01, ntrees=255	3863.669837
DT Model 3	d=2, lr=0.1, ntrees=75	3490.682823
DT Model 4	d=2, lr=1, ntrees=16	3630.410824
DT Model 5	d=4, lr=0.001, ntrees=255	8281.428641
DT Model 6	d=4, lr=0.01, ntrees=255	3444.204324
DT Model 7	d=4, lr=0.1, ntrees=75	3178.610645
DT Model 8	d=4, lr=1, ntrees=16	3857.703332

(6) Table of Decision Tree Regressor Ensemble Performance

RESULTS

As was mentioned earlier in the paper, the metric used to evaluate the performance of each learned model was the root mean squared error (RMSE). Below, (7) shows the equation for calculating RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}}$$

(7) Root Mean Square Error (RMSE) Equation

As a metric, the root mean squared error is a measurement for how distant a model's predictions are from the recorded value. With this metric, the performance of different models can be compared for selection of the best one. Below, (8) shows the RMSE of the best performing models from each approach.

	Root Mean Squared Error (RMSE)
Baseline	10264.067487
Linear Regression	5134.940713
Ridge Regression	5134.958962
Decision Tree	4147.503699
DT Ensemble	3178.610645

(8) RMSE of Best Resulting ML Regression Models

DISCUSSION AND FUTURE WORK

As can be seen in (8) the original linear least squares regression model cut the original error measurement by a factor of 2 and is a drastic improvement over the baseline.

As mentioned earlier, the ridge regression approach did not perform any better than the linear regression approach. This is surprising as regularization of the simple linear regression model often leads to better generalization, and subsequently scoring, on the test data. This may be an indicator that the relationship between the

independent variables of the data set and the dependent variable are not, in fact, linear. Hence, the poorer performance of the linear models.

The decision tree regression approach offered significant improvement over the performance of both the linear and the ridge regression approaches. If you assume that the relationship of the produced power to the independent variables is not linear, something which is clearly suggested by performance of the linear models, then said non-linear relationship may well be the reason for the performance improvement seen in the regression tree approach.

While the individual regression trees that made up the ensemble were kept simple relative to the original singular regression tree, the performance of the ensemble still outshone all other approaches. This result was the hope of the project, as ensemble approaches tend to outperform singular machine learning models. In this case, the selection of regression trees for the base model helped instantiate significant improvement in regards to the other approaches.

REFERENCES

[1] Detyniecki, Marcin, et.al. “Weather-Based Solar Energy Prediction” 2012. Web Accessed.
[2] Sharma, Navin, et. al. “Predicting Solar Generation from Weather Forecasts Using Machine Learning”, Web Accessed.
[3] Nespoli, Alfredo, et. al. “Day-Ahead Photovoltaic Forecasting: A Comparison of the Most Effective Techniques”, 2019. Web Accessed.
[4] “Theocharides, Spyros, et. al. “Comparative Analysis of Machine Learning Models for Day-Ahead Photovoltaic Power Production Forecasting”, 2018. Web Accessed.
[5] Martin, R., Aler, R., Valls, J. M., and Galvan, I. M. “Machine learning techniques for daily solar energy prediction and interpolation using numerical weather models”, 2016. Web Accessed.
[6] Carrera, Berny and Kim, Kwanho. “Comparison Analysis of Machine Learning Techniques for Photovoltaic Prediction Using Weather Sensor Data”, 2020. Web Accessed.