CS 229: Machine Learning
June 12, 2019

Predicting Solar Power Generation from Weather Data
Project Report

Alex Kim, *Mathematical and Computational Science* alexkim7@stanford.edu
Dane Stocks, *Physics* dstocks@stanford.edu

https://github.com/alexkim/solar-forecasting

---

1 Motivation

In this project, we aim to predict solar intensity for a given area 48 hours into the future using local time-series weather observation data. Specifically, we will use data from the National Solar Radiation Database (NSRDB)[1], which conveniently includes both weather and solar intensity measurements at continual 30-minute intervals for a localized geographic area.

Our motivation for predicting solar intensity is that it is directly proportional to solar power generation; if we are able to accurately model future solar intensity given current weather data for a specified area, then that area's solar generation output in the near future can be estimated with greater accuracy. Working toward more accurate prediction of solar power generation addresses one of the obstacles facing widespread integration of renewable energy into the national power grid. The highly variational nature of renewable energy production puts stress on conventional (fossil fuel based) power generation. Currently, most grid power facilities are forced to alter the rate of conventional power production in accordance with near-real-time levels of renewable production (i.e., throttle conventional output when it is particularly clear and bright on one day, or increase conventional output when there is not a gust of wind on another day). One significant issue that grid power sites face occurs when predicted renewable production varies inversely with consumer demand for electricity. These disparities produce "ramping" periods where conventional production is quickly increased or decreased, which is a costly operation for grid sites. This phenomena is explained in detail and graphically shown in a 2014 publication of the National Renewable Energy Laboratory[2].

In an attempt to mitigate this problem, we follow the advice of a 2011 update from the American Physical Society's Panel of Public Affairs[3], and investigate the possibility of providing high-confidence forecasts of solar generation (via solar intensity) using simple, readily-available weather data. By limiting the uncertainty of predicted solar forecasts, such a model has the potential to allow grid sites to reduce production of conventional reserves (which at the moment remain high, on average, due to large variance in forecasts). Additionally, accurate forecasts would enable sites to dampen the economic impact of ramping periods by being more prepared to switch between conventional and renewable sources. In summary, we aim to lay the groundwork for constructing (ideally adaptive) models that could be dispatched to various regions, incorporate that geographic location's weather data, and output accurate predictions for that area's solar power production up to 48 hours in the future.

[1] https://nsrdb.nrel.gov/
[2] https://www.nrel.gov/docs/fy14osti/61721.pdf
[3] https://www.aps.org/policy/reports/popa-reports/upload/integratingelec.pdf

2 Data

2.1 General Information

The NSRDB includes both observed weather data (temperature, relative humidity, cloud cover, etc.) and solar intensity data, measured in watts per square meter. The database actually includes several types of solar radiation measurements; we specifically choose to analyze *global horizontal irradiance* (GHI), as it incorporates both direct incident radiation and ambient solar radiation reflected from nearby surfaces and atmospheric particles. This makes it a good indicator for solar panel readings[4].

The NSRDB covers the entire United States at a spatial resolution of one measuring station per every four square kilometers. Its data is measured once every 30 minutes, stretching back to 1998. For the purposes of this project, we chose to investigate a single location with qualitatively strong solar intensity: Las Vegas, Nevada, at the coordinates (36.17°N, −115.14°W). Additionally, we limit our analysis to the data collected over the entirety of 2016 and 2017. This leaves us with 35,088 distinct observations, each with the 14 features shown in Table 1 and a corresponding measure of GHI.

Year	Day	Hour-Minute	Surface Albedo	Cloud Type	Dew Point	Zenith Angle
Month	Pressure	Wind Direction	Wind Speed	Relative Humidity	Temperature	Precip. Water

Table 1: The 14 weather (and time) parameters we chose for our input features.

2.2 Data Considerations

In the process of training different models on our data set, the results of which are explained in the proceeding section, we chose to use the exact same train/development/test split for each model. In our implementation of k-means clustering, we shifted our view of the data so that a single "observation" was composed of the 48 readings of solar intensity over a day. We then ran a k-means algorithm to identify the five (and then ten) most common solar intensity trends throughout a day, and mapped half-hour observations to these clusters during testing using GDA. Importantly, this process only makes logical sense when we take care not to split the half-hour observations of a single day between the train/development/test data sets — otherwise the corresponding day-length observations have no physical significance. To allow us to directly compare performance between k-means and our other regression models, we decided to split our overall data set along day boundaries for all training and testing.

Furthermore, given our plentiful 35,088 observations, we opt for a conservative 40/10/50 ratio for our train/development/test split. We choose this split to ensure that we have high confidence in the reliability of our test performance.

Given that some of our features are on the scale of hundreds while others rarely exceed the value 3, we standardize all of our features before constructing any models. This allows us to ensure that no features are undervalued simply because of their scale.

Looking at Table 1, one may observe that we did not include current solar intensity as a predictor for the solar intensity at a future time. The aim of this project was to develop a model to accurately predict future solar intensity using basic and *ubiquitously recorded* weather data. While the NSRDB has extensive measurements of solar intensity, this is not a common datum for the average weather station. We focus our efforts on common data such as temperature, wind direction and speed, and relative humidity because of their simplicity and widespread use for traditional forecasting purposes.

[4] https://www.3tier.com/en/support/glossary/#ghi

3 Models and Results

To formalize our task, our goal is to have predicted all solar intensity values up to 48 hours in the future at any given time point. Note that this is not limited to predicting the single solar intensity value exactly 48 hours in the future. While we do use this method in some models, there are also alternative ways to approach the problem, such as predicting entire batches of solar intensity values simultaneously. We explore one such method in this project.

Because there are multiple ways to approach our task, we do not have universal pairing of past weather observations to future solar intensity observations—we can choose to predict solar intensity using the weather data from any time point in the past, or even from multiple time points combined. The only constraint is that we predict at least 48 hours into the future.

3.1 Linear Regression (Baseline)

For our simple baseline model, we perform linear regression on individual weather observations (14 features) to predict the solar intensity exactly 48 hours in the future. In other words, we impose the artificial pairing of each weather observation to the solar intensity measurement from 48 hours in the future. After running a standard linear regression on this pairing, we achieve the following results:

Train R^2: `0.773`
Test R^2: `0.766`

Given the simplicity of this model, we find the R^2 to be fairly impressive. Furthermore, the closeness of the train and test R^2 indicates that this model generalizes well and does not overfit.

3.2 Linear Regression with Multiple Weather Observations

As an extension of our baseline linear regression model, we now incorporate multiple weather observations into our feature set for predicting solar intensity. We maintain the original pairing of weather data to the solar intensity value 48 hours in the future; however, we now we expand the feature space to include additional weather observations from previous time points. Namely, we expand our feature set to include the weather observations from the adjacent past time points.

1 Expansion (28 Features)		**3 Expansions (56 Features)**		**10 Expansions (140 Features)**		**30 Expansions (420 Features)**	
Train R^2:	`0.789`	**Train R^2:**	`0.826`	**Train R^2:**	`0.895`	**Train R^2:**	`0.910`
Test R^2:	`0.780`	**Test R^2:**	`0.814`	**Test R^2:**	`0.888`	**Test R^2:**	`0.900`

In our tests, we evaluated a larger range of expansions, but we feel that the above expansions provide a representative picture. After 30 expansions, we did not see any additional gain in test R^2. Unsurprisingly, as we add more features, we can see that our training R^2 increases. Furthermore, even with 30 predictors, the gap between train and test R^2 remains small, so there are no concerns of overfitting.

3.3 Linear Regression with Quadratic Feature Expansion

Expanding on our previous model, we now apply a mathematical feature expansion. Namely, we follow the same approach as the previous section, but now expand our feature set to include the square of each feature, as well as the pairwise interaction between each pair of features. Because the pairwise interactions expand our feature set considerably, we generally work with smaller starting feature sets (e.g.

2 expansions), as this still provides an ample expanded feature space.

No Expansions (217 Features)		**1 Expansion (874 Features)**	
Train R^2:	`0.929`	**Train R^2:**	`0.937`
Test R^2:	`0.914`	**Test R^2:**	`0.738`

As we can see, the test R^2 immediately drops when we introduce one expansion (one more weather observation) to the starting feature set. Since our training R^2 remains high, we suspect overfitting, and thus repeat this experiment with ridge regularization applied. Below are our results.

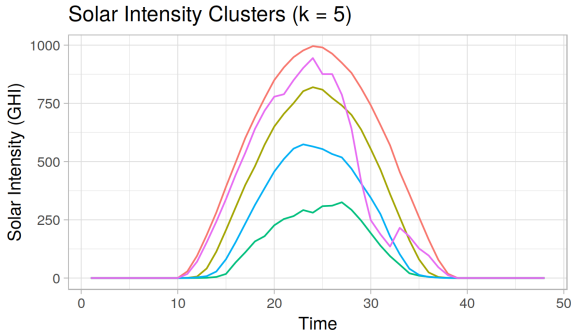
No Expansions (217 Features)		**1 Expansion (874 Features)**		**2 Expansions (1972 Features)**		**3 Expansions (3511 Features)**	
Train R^2:	`0.912`	**Train R^2:**	`0.922`	**Train R^2:**	`0.927`	**Train R^2:**	`0.933`
Test R^2:	`0.903`	**Test R^2:**	`0.909`	**Test R^2:**	`0.911`	**Test R^2:**	`0.913`

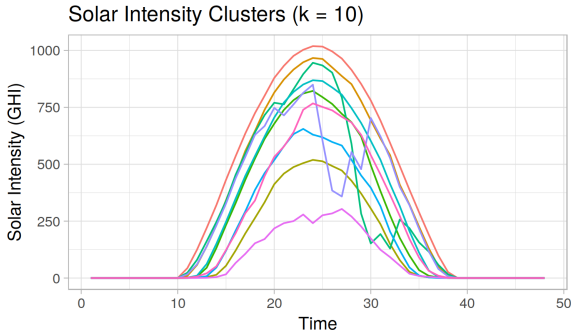
3.4 K-Means Clustering with Gaussian Discriminant Analysis

3.4.1 Unsupervised Clustering

"Please God look at your data." — Professor Chris Ré

In order to gain visual insights into the patterns of solar intensity, we apply k-means clustering to the daily trajectories of solar intensity measurements. In doing this clustering, we ignore all predictors and instead focus only on the response variable. We also group the solar intensity measurements by day, allowing us to treat the solar intensity at each time as a feature of the day. Since each day contains 48 solar intensity measurements, we cluster over a 48-dimensional space, which we can conveniently visualize as a time series. Below are visualizations of the clusters of solar intensities, both for $k = 5$ (left) and $k = 10$ (right).





As we can see, solar intensity generally follows an inverted paraboloid shape, and is consistently valued at zero near the beginning and end of the day. Some clusters exhibit a very smooth trajectory, whereas others exhibit some roughness and irregularity. Despite these differences, it is helpful to know that the daily solar intensity trajectories generally lie within a predictable range.

3.4.2 Regression via Gaussian Discriminant Analysis

Aside from just gaining visual insights, we also leverage k-means clustering to initialize supervised classification on our data. Namely, we apply Gaussian discriminant analysis to classify our weather data

examples, using the cluster assignments as labels.

Note that we cannot immediately perform this task, as there is some inconsistency between the features and the labels; our original features are weather observations measured at 30-minute intervals (48 per day), whereas our labels are cluster assignments corresponding to each entire day (1 per day). One solution might be to simply assign each day's label to each of the 48 weather observations corresponding to that day; however, at prediction time, this might lead to multiple different predicted labels for any given day (since we will be predicting on all 48 weather observations).

To solve this problem, we naively throw away all weather data examples except for last one in each day. This leaves us with one weather example per day, giving us a one-to-one correspondence of weather observations to labels. From here, we train a supervised classification model using Gaussian discriminant analysis. At prediction time, we predict the cluster assignment for each day using only the last weather observation in the day. Once we obtain the predicted cluster assignments, we are able to predict the solar intensity value at each 30-minute interval of the day by referring to the centroid corresponding to the assigned cluster. By following this process, we are able to predict the solar intensity value at each time point for an entire day up to 48 hours in the future. Below are the R^2 values we attain from this process.

1 Cluster		**5 Clusters**		**10 Clusters**	
Train R^2:	`0.815`	**Train R^2:**	`0.925`	**Train R^2:**	`0.932`
Test R^2:	`0.803`	**Test R^2:**	`0.925`	**Test R^2:**	`0.923`

As we can see, using 10 clusters provides no discernible gain in test set performance, compared to using 5 clusters. However, 5 clusters does provide a noticeable gain over using 1 cluster (i.e. the average solar intensity trajectory). It may also be worth looking into methods to reduce the gap between train and test set R^2, perhaps through some sort of regularization method.

4 Conclusion

Ultimately, we found that our combination of k-means clustering and Gaussian discriminant analysis provided the optimal test R^2 of **0.925**. Following closely was ridge regression with quadratically expanded features, attaining a test R^2 of **0.913**. All other linear regression models yielded lower test R^2, but still demonstrated considerable improvements over our baseline test R^2 of **0.766**.

Overall, we are surprised that the k-means/GDA combination performed as well as it did. One of us had briefly learned about this general approach being used for predicting a consumer's energy consumption, so we are glad to have been apply it here in another energy-related problem. Moving forward, it might be worth exploring additional clustering schemes (e.g. hierarchical clustering), as well as other classification methods (e.g. support vector machines).

5 Contributions

Alex Kim, *Chief Stack Exchange Debugger* — Converted ideas and models into code; ran models and logged results. Devised the combined method of k-means clustering and Gaussian discriminant analysis.

Dane Stocks, *Executive LaTeX Wrangler* — Led efforts to formalize results and observations in written reports. Provided rigorous oversight and verification of all methods and models.

Special thanks to Professor Dorsa Sadigh, for providing excellent insight into machine learning in CS 221; and to Alex Laskey of Opower, for inspiring the time-series clustering method.