Predicting Solar Panel Deployment Using US Census Data

Jeremy Chen, Eddie Sun, Brett Szalapski
{czm, eddiesun, brettski} @stanford.edu

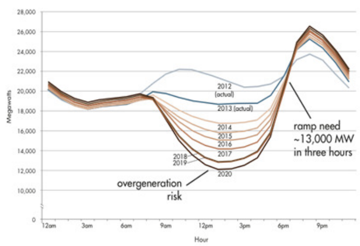
Motivation

The recent surge in the number of intermittent energy generation facilities requires improved monitoring and control methods for the electric grid due to increased supply-side uncertainty.

One major component of supply-side uncertainty comes from residential solar panel installations. Today, installing solar panels on residential homes is easy and affordable. As a result, it is difficult to know how many solar panels exist and supply power to the grid.

For this project, we developed an SVR and a neural network (NN) to predict solar panel deployment from US census data. We also used PCA and the NN to gain some insight on the data.

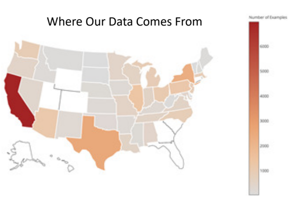
Daily Power Generation in California

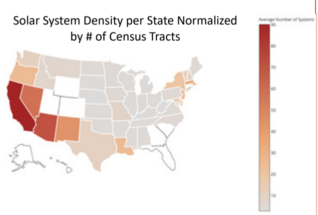


Dataset

The features of the dataset contains US census data from the 2015 American Community Survey [1]. The labels of the data (# of solar panel systems) comes from a previous project [2] in ES's research lab which used conv-nets to count solar panels from satellite images (not yet published).

The dataset contains labeled data from 35,698 census tracts and is split 80/10/10 between train/dev/test sets. The data was preprocessed by removing categorical columns and rows with invalid numeric values before feeding the data into the SVR and NN.

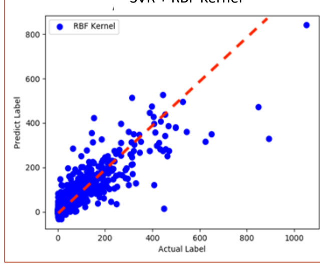




Support Vector Regression

Support Vector Regression (SVR) with several different kernels—Gaussian RBF, Linear, Polynomial—was implemented with SKLearn. SVR with the RBF kernel has the highest performance of the three, with a correlation coefficient of R^2 = 0.79. The linear kernel does not perform well, which is expected based on the PCA plot. The polynomial kernel does not converge.

SVR + RBF Kernel



Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

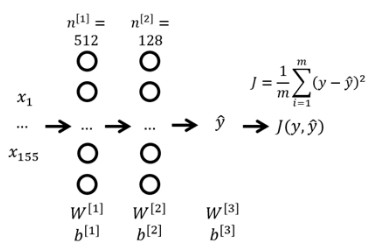
Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$

Gaussian (Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$

Neural Network

The neural network (NN) is a standard feed-forward NN coded in Keras with a Tensorflow backend. The final model architecture and hyperparameters were tuned using random search.



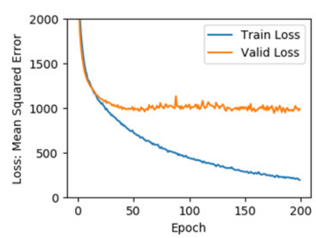
Forward Prop

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$
$$A^{[1]} = ReLU(Z^{[1]})$$
$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$
$$A^{[2]} = ReLU(Z^{[2]})$$
$$\hat{y} = W^{[3]}A^{[2]} + b^{[3]}$$

Back Prop

$$W^{[3]} := W^{[3]} - \frac{\partial J}{\partial W^{[3]}}$$
$$b^{[3]} := b^{[3]} - \frac{\partial J}{\partial b^{[3]}}$$

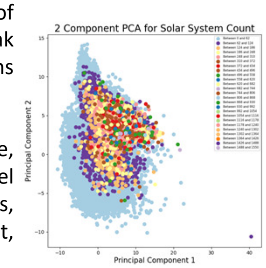
etc.



Hyperparameter	Value
# Hidden Layers	2
# Neurons	[512, 128]
Batch Size	512
Dropout Prob. [3]	[0.25, 0.25]
Learning Rate	0.0003

Principal Component Analysis

Principal Component Analysis (PCA) of the dataset demonstrates that a weak radial trend is present, which explains the SVR + RBF performance.



Top components: average income, education level, cost of housing, fuel cost, race, electricity prices, incentives, democratic voting percentage, frost, humidity.

Equation: $PC_1 = \underset{u:u^Tu=1}{\operatorname{argmax}} \ u^T \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}x^{(i)T}\right)u$

Discussion

Model Performance

Model	Train MAE	Val MAE	Test MAE	Train R^2	Val R^2	Test R^2
SVR (Lin. Kernel)	29.1	29.9	29.3	0.26	0.16	0.22
SVR (RBF Kernel)	4.2	17.4	18.1	0.93	0.79	0.78
NN	7.3	10.5	18.5	0.95	0.79	0.71

*MAE = Mean Absolute Error = $\frac{1}{m}\sum_{i=1}^{m}(y - \hat{y})$

Feature Correlation: The numerical gradients of the predictions w.r.t. the features were calculated using the NN weights and correspond well with the PCA features above. The table below shows notable influential features.

Pos. Correlated Features		Neg. Correlated Features	
% of Dem. Voters (2012)	0.85	Population Density	-1.00
Median House Value	0.81	% of Rep. Voters (2012)	-0.64
% of Dem. Voters (2016)	0.67	% of Rep. Voters (2016)	-0.63
Yearly Solar Irradiance (KWh)	0.66	# Frost Days	-0.29
Sales Tax Rate	0.46	Poverty Level	-0.24

Future Work

- Separating residential vs. commercial installations
- Transfer learning to other similar tasks (e.g. predicting electric car ownership)
- Error analysis, further hyperparameter tuning

References

[1] American Community Survey. 2015. 2015 ACS 1-year estimates [data file]. Retrieved from http://factfinder.census.gov
[2] Yu, Jiafan, Wang, Zhecheng et al. "DeepSolar: A Machine Learning Framework to Efficiently Construct Solar Deployment Database in the United States." Joule (2019), accepted
[3] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.