Reducing CO_2 Levels Using Genetic Algorithms

Cristian Lincu

This project aims to provide a solution framework for minimizing CO_2 emission levels in a national power grid, through machine learning methods. The data is collected via Energinet's public API which provides real-time data about Denmark's power system[1]. The dataset prepared for this research consists of data recorded every 5 minutes for the past year.

The main elements of this project are: a regression tree for inferring CO_2 levels, XGBoost regressors for predicting demand and renewable energy production (trained on past year's data) and genetic algorithms for finding the optimal energy distribution with respect to CO_2 emission minimization. The goal is to infer the optimal resource combinations that minimize CO_2 emissions, depending on future demand and renewables production. The methodology is described below:

I. A decision tree regressor infers CO_2 levels taking into account 12 features: renewable energy production (solar and wind), energy produced by power plants with installed capacity greater or equal to 100 MW, energy produced by power plants with installed capacity less than 100 MW and energy exchanges between Denmark and other countries or areas (DK1-DE, DK1-NL, DK1-GB, DK1-NO, DK1-SE, DK1-DK2, DK2-DE, DK2-SE, Bornholm-SE); positive exchange values are imports, while the negative ones are exports. The hyperparameters of the regressor are tuned using grid search and the model is evaluated through 5-fold cross-validation.

II. The XGBoost regressors provide multi-step forecasting for energy demand and renewable production for the next 5 time points. The hyperparameters are tuned using grid search. These forecasts are important because ultimately, the optimal power distribution will have to match demand constantly, in order to keep the grid frequency at 50 Hz (within a margin of

[1] https://www.energidataservice.dk/tso-electricity/PowerSystemRightNow

0.1 Hz, translated into ±1% of the total demand). Considering that renewable energy depends on weather conditions, we will need a forecasted amount of distributable energy whose allocation will be optimized among the mentioned sources (except for renewables, of course); this quantity will be represented by the difference between forecasted demand and forecasted renewable energy.

III. For each of the 5 forecasted steps, the optimal combination of sources is inferred using genetic algorithms. In order to escape eventual local optima and increase the method's success, this is done by employing a genetic ensemble with 25 optimizers. Populations of solutions are initialized stochastically and transformed by evolutionary processes to arrive at the "fittest" distribution, exploring a search space of approximately $2 \cdot 10^{32}$ combinations. The restrictions are the ranges of power plant production and the interconnector limits for exchanges (as inferred from past year's data). These limits are considered at initialization time and respected throughout the process, during random mutation events. Another condition, as we have seen, is that the distributable energy matches the difference between forecasted demand and forecasted renewables; this is addressed directly in the fitness function.

The first generation is of the form

$$G = \begin{bmatrix} - & x^{(1)} & - \\ - & x^{(2)} & - \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ - & x^{(m)} & - \end{bmatrix} \in \mathbb{R}^{m \times d}$$

where m is the size of the population, d is the number of genes within each chromosome $x^{(j)}$ and each $x^{(j)}$ represents a vector comprising representations of genes, whose values have to be optimized through selection, crossover and mutation. Each chromosome is evaluated using the regression tree to infer the CO_2 emission level implied by the solution it represents, while solutions that do not balance the power grid are penalized by an augmentation that will significantly reduce their coupling chances. Some simplifying approaches are in place, as the model does not take into consideration other factors that might influence the quantities, like exchange contracts with neighboring countries or energy prices in the area. However, the fitness function also includes a term whose role is to maximize the similarity between proposed solutions and the latest distribution in the dataset. This makes the optimal

solutions for the next 5 time points more realistic and also makes sure there is a certain coherence between them, as much as possible, avoiding crazy swings of production or exchange values, 5 minutes apart. Thus, each optimal solution minimizes the fitness function composed of these elements: grid imbalance penalty, negative resemblance and CO_2 level.

$$x^* = \underset{x^{(j)}}{\operatorname{argmin}} \left[\xi \left(\mathbf{I} \left(\left| D_{XG} - R_{XG} - \sum_{i=1}^{d} x_i^{(j)} \right| > 0.01 D_{XG} \right) - \cos\left(x^{(j)}, \lambda\right) \right) + \zeta \Delta_{CO_2} \right]$$

where:

ξ is the penalty hyperparameter,

$\mathbf{I}$ is the indicator function,

D_{XG} is the demand forecast provided by the XGBoost regressor,

R_{XG} is the renewable energy forecast provided by the XGBoost regressor,

$\cos\left(x^{(j)}, \lambda\right)$ measures the cosine similarity between the chromosome and the latest energy distribution in the dataset,

Δ_{CO_2} is the CO_2 emission level inferred by the decision tree regressor.

The indicator function $\mathbf{I}$ checks whether the solution balances the grid within a margin of 1%. If not, the chromosome is penalized with ξ, so that the probability of coupling for crossover is reduced considerably. The genetic heuristic encourages this term to be 0.

The second term, $\cos\left(x^{(j)}, \lambda\right)$, computes the cosine similarity between $x^{(j)}$ and the latest distribution λ, defined as their dot product divided by their norm: $\frac{x^{(j)} \cdot \lambda}{\|x^{(j)}\| \, \|\lambda\|}$, giving results between -1 (completely opposite vectors) and 1 (identical vectors). This term is taken with a negative sign in the fitness function, so that the similarity is maximized. In this case, the role of ξ is to augment the cosine result to make it detectable in the optimization process.

The third term minimizes the CO_2 level inferred by the regression tree, augmented by ζ.

Ultimately, every 5 minutes, this model produces 5 forecasted time points presenting values for power plant production and exchanges that minimize CO_2 levels, balance the grid and, at the same time, maximize the resemblance with the latest distribution in order to increase realism and coherence. This model has provided solutions that represented potential 50%-75% CO_2 emission reductions.

Visit the Flask web app deployed on Heroku (desktop view only), which includes visuals showcasing a dynamic comparison between past hour's emissions and minimized CO_2 levels in the next 5 time points (inferred by the model presented herein), as well as the optimal energy resource distributions leading to these minimized emission levels. Check out the Python scripts on GitHub.