CS760: Forecasting Power Production

For The Askaryan Radio Array

Matthew Newcomb & Greig Hazell
University Of Wisconsin
mnewcomb@wisc.edu & hazell@wisc.edu

Abstract

This paper studies forecasting wind turbine power production for use in the Askaryan Radio Array at the South Pole. As physical hardware does not yet exist, first a model for the power production of a wind turbine from historical meteorology data is developed.

1. Introduction

The Askaryan Radio Array (ARA) is a widely distributed neutrino telescope being designed for deployment at the South Pole. Neutrinos, neutral subatomic particle with a mass close to zero, usually pass smoothly through matter without stopping or interacting. This “low cross section” makes their detection extremely difficult and thus rare. To be effective a huge radio-transparent and radio quiet detector is needed. How huge is huge? The ideal design places one detector every kilometer over six hundred square kilometers, each drawing over three hundred watts! Luckily the ice at the South Pole meets all requirements admirably.

However the remoteness of the South Pole brings many unique challenges. The United States Antarctic Program has had a continuous presence at the South Pole since 1957. The base is called the Amundsen Scott South Pole station and sits at 9306 feet above mean sea level all on glacial ice. The sun is up twenty four hours a day for the summer, but it disappears for six months of winter! Winter temperatures can dip to -117F. The day to day atmospheric pressure varies enough to make the pressure altitude vary by almost 3,000 ft. With all of these extremes, winds are surprisingly light, averaging only 10.7 knots with the highest measured wind speed of 48 knots [3]. There are generators run by the support staff on the station but there is not enough excess capacity to power this array. Cabling even a small 20 station array to these generators would cost an estimated quarter of a million dollars. The current design calls for wind turbines at each detector with a small battery bank to provide the required power. The problem is that the winds at the south pole are light and are not reliable enough to provide a continuous uninterrupted power source.

The naive control algorithm currently proposed shuts a station down when the batteries reach a low state of charge, only turning it back on again once the charge level has been restored. The drawback of this control algorithm is that as all stations are likely to be exposed to the same wind speed they will likely all power off at the same time and recover at the same time. This is not desirable as a complete detector outage will eliminate any chance of catching a possibly rare event. Improvements to this control algorithm may be possible by forecasting future power production of the wind turbines!

The South Pole Meteorology Office has recorded wind speed, wind direction, temperature, and atmospheric pressure once a minute, every minute since 2005 and has kindly made that data available to us [2]. As no final turbine has been selected for ARA, our technique has to be generic for any given wind turbine and has to work from the weather data supplied.

2. Previous Work

Previous work has focused on forecasting future wind speeds instead of future power production[1][10]. This technique is not sufficient for use at the south pole for a number of reasons. First, maintaining the necessary meteorological instrumentation in a remote harsh environment is difficult and expensive. Power production measurement is simple and requires no moving parts. Second, due to the unique weather conditions at the south pole with severe air density changes (thousand foot changes in physiological altitude overnight are not uncommon) forecasting wind speed alone is insufficient. Thirdly forecasting wind speed alone does not take individual turbine variations into account. Snow drifting is a major problem at the south pole resulting in wind turbine towers becoming shorter by several feet a year. The meteorology office actually mounts some of their instrumentation on sleds that can be repositioned as needed to be on the snow surface. Finally, some work augments forecasting technology [9] that is not available in Antarctica.

3. Methodology

3.1 Turbine Modeling

As the physical hardware for the ARA array has not yet been selected we had to devise a method for modeling wind turbine power production based on meteorology data. The South Pole Meteorology Office has kindly provided us with wind speed, direction, temperature and atmospheric pressure sampled every minute from 2005 through the present[2].

$$Power = \frac{1}{2}(swept\_area)(air\_density)(velocity^3) \quad (1)$$

3.2 Tower Height Corrections

The wind speeds provided by the South Pole Met office are taken at the top of a 10 meter tower. ARA may use a different tower height. The wind speed varies at different heights in a logarithmic profile depending on the surface roughness (z_0). Luckily Jackson & Carroll present a model for z_0 for the South Pole that depends

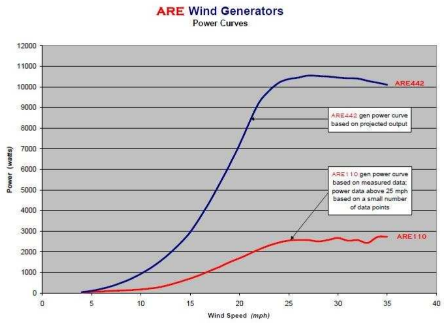

Figure 1. Power Production Curve For ARE Wind Generators

on wind direction. Given the calculated z_0 we can correct our wind speed for new tower height with the following formulas:

$$u = speed_{measured} \frac{0.4}{log(\frac{height_{measured}}{z_0})} \tag{2}$$

$$speed_{corrected} = \frac{u}{0.4} log(\frac{height_{new}}{z_0}) \tag{3}$$

3.3 Power Production Curve

A wind turbine manufacturer will provide a set of data points relating wind speed to power output at sea level at standard pressure and temperature. These data points can be fit to a smooth equation [8]. For an example of such a curve see fig 1.

3.4 Density Correction

As the power production curve is for a turbine at sea level we have to correct our estimate for production at altitude. Air density (ρ) is expressed as $\rho = \frac{pressure}{specificgasconstant*Tinkelvin}$. $\rho_{sea-level}$ is defined as 1.2041183[11]. Using the MET office weather data we calculate an instantaneous $\rho_{altitude}$ and de-rate the turbine output by a factor of $\frac{\rho_{altitude}}{\rho_{sea-level}}$.

3.5 Verification

Raytheon Polar Services Corporation installed an ARE110 2.5 KW wind turbine on top of a 12 meter tower. A technician manually records the total number of kilowatt hours produced by the turbine since last reset. For the month of April 2010 there where no logged problems with the turbine and the logs where obtained for that month. To evaluate our model we ran the weather data for that month through the model described above and then integrated to produce total kw/h's produced per day to match the logs provided by Raytheon. One source of uncertainty is that it is not known what time of day the technician recorded the daily readout. The total actual production for the month is within 1.2% of the production estimated by our model in fig 2.

4. Forecasting

4.1 Preprocessing

Transforming the weather data to power production data results in a time series of watts per minute. Directly using this data proved to be problematic due to the training time required. For instance training a multi-layer perceptron with forty-five minutes of previous history required almost an hour of training time. After a discussion with an ARA personnel it was decided that any stations should only be turned on or off once an hour. As a result the time series of watts per minute was integrated down to watts per hour.

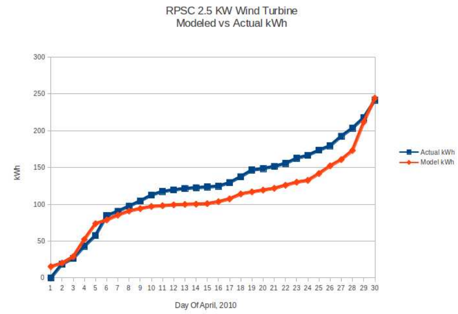

Figure 2. Actual Versus Model Comparison

5. Machine Learning

5.1 Forecasting

We would like to be able to forecast power production up to ten hours into the future. Three algorithms capable of numerical prediction were selected for evaluation.

1. Multi-layer Perceptrons
   - Settings
     - Learning Rate 0.3
     - Momentum 0.2
     - Number of Epochs 500
2. SVM By Sequential Minimal Optimization (SMO)
   - Settings
     - Complexity Constant 1.0
     - ϵ 1.0E-12
3. Linear Regression
   - Settings
     - M5 Attribute Selection Method
     - Ridge parameter 1.0E-8

All three algorithms were easily available in Weka version 3.6.0. All were tested using data from the entire winter of 2008 with 10 fold cross-validation.

5.2 Past History

Two methods were compared to decide the amount of past history required for an effective prediction. First we plotted the RMSE error over the folds of the cross validation. Figures 3, 4, and 5 are plots of the Weka reported RMS error over the folds of the training data.

In all three cases it should be noted that providing four hours of past history to the learning algorithm being evaluated yielded approximately the same results as providing eleven hours of past history.

As a further test we performed a sign test on a test set consisting of data from the winter of 2009. The error of a prediction with eleven hours of past history was compared with the error of a prediction with a lesser amount of past history. The fraction of time the error of the eleven hours was best was then plotted (fig 6). The results demonstrate that using four hours of previous history yields results that are significantly similar to using eleven hours of past history.

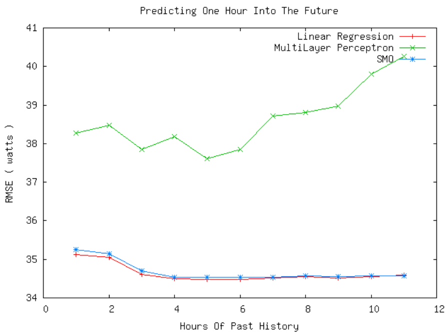

Figure 3. Predicting One Hour Into The Future

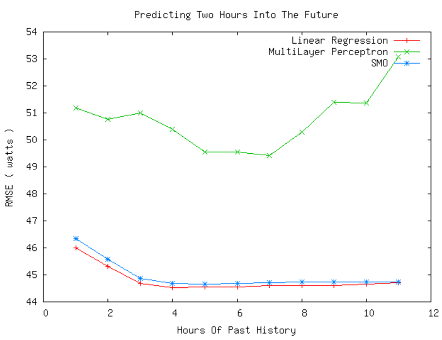

Figure 4. Predicting Two Hours Into The Future

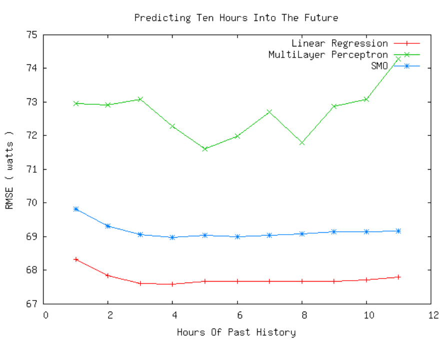

Figure 5. Predicting Ten Hours Into The Future

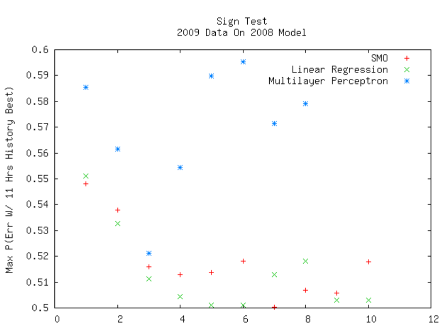

Figure 6. Sign Test

```
GetReward(state, action, detector_array)
     reward = 0
     for each detector d
       for each power prediction p
         if actiond,p = on
           reward += 1
           if first time grid is active at time t
             reward += 1
     return reward
```

Figure 7. Reward Function

5.3 Schedule Learning

Once we are able to predict the future power available to a detector, the next phase of the project involves using this information to schedule the activation and deactivation of each detector within the array. Our aim is to maximize both the total number of hours each detector is active and the total time the array is in an active state. The array is considered to be in an active state if at least one of its detectors is currently active.

In order to determine the optimal schedule of detector activations and deactivations, we will make use of a reinforcement learning algorithm. Reinforcement learning algorithms in general have successfully been used in various optimization problems including (control) scheduling.

For the project we used PyBrains™reinforcement learning package. PyBrains NFQ package is a feed forward, neural network learning algorithm. There are several advantages in using a neural network learning algorithm over a traditional state-action table learning algorithm. Amongst these advantages are the condensed representation of the environments state space and the possibility of converging faster. Given the exponential nature of our environments state, the size of which is calculated as

$$(bf2)^n$$

where b is the detectors battery level, f is the number of future predictions, and n is the size of the array. A neural network learner is likely the only feasible solution for this problem.

While a few minor extensions were necessary to the NFQ package in order to be applied to our problem, the details of these extensions are neither necessary nor advantageous to the reader. Instead we turn our attention to the details of the environment and the reward function used.

The environment state space consists of the current battery level as a percentage of the maximum possible charge discretized within 10% intervals along with the future power predictions available for each detector within the array. To further reduce the size of the state space each power prediction was represented as a fraction of the maximum possible power output generated by a turbine correct to two decimal places.

The details of the reward function are shown in figure 7. In order to meet our objectives of maximizing a single detector's active status as well as that of the array, two separate rewards are provided to the learner. A reward is received each time a detector is successfully activated within a given timeslot. An additional reward is also received if the time slot is previously empty. That is, no other detector has been currently scheduled within the time slot. The aim of the second reward is to encourage the learner to stagger each detector's schedule against the remaining detectors.

6. Conclusions

The RMS error across the folds of the training set (winter 2008) showed that linear regression out performed the other methods (fig

Figure 8. 2008 RMS Error On Cross Folds

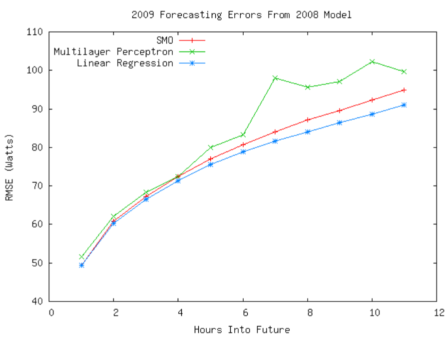

Figure 9. 2009 RMS Error On 2008 Model

8). The comparison held when all three algorithms were run on a test set of data from the winter of 2009 (fig 9).

As an approximation for how far in the future these methods would effectively predict power production we performed the following procedure. The average power production for our training set of the winter of 2008 was 81 watts with a standard deviation of 92 watts. If we simply predict average power production we get an RMS error of 125 watts. Applying the same techniques proposed in this paper allows us to predict power production up to thirty hours in the future before the RMS error of any of our predictive techniques exceeds that of naively predicting the average.

Because of time constraints, a few (minor) sacrifices were made as discussed below in order to evaluate the Q Learning Algorithm. Ideally, a Q learner should be trained on simulated data and allowed to run as long as necessary until convergence (or the user is happy with its performance). However, because of limited time, we were forced to restrict the amount of data sampled and as a result deliberately limited training data to that available for the years 2005 through 2010. The data for 2011 was reserved for testing purposes. Each training iteration comprised of 3 future power predictions, an array of size 2 and a specified number of passes to be made over the data. Values of 1, 2, 5 and 10 were used.

Unfortunately, the results were not as expected. The table below shows the number of hours active for a given detectors as well as the array out of a maximum of fifteen hundred hours. After analyzing the data, we observed that the problem lied primarily in the reward function. Given a state such as $[b_1, p_{1,1}, p_{1,2}, p_{1,3}]$, $[b_2, p_{2,1}, p_{2,2}, p_{2,3}]$ where b represents the battery state and p_i's are the future power predictions, we were unable to determine the appropriate reward balance for scheduling a detector within the first time slot and maximizing the number of time slots covered. Consequently, we would see an action such as 0, 1, 0 for detector 1 and 0, 0, 1 for detector 2. Consequently both detectors are delayed for scheduling until the next hour. Receiving the next hourly update we will see a similar action with the activation being delayed until a later hour. This will occur more frequently as the number of iterations in the learning process increases.

Number passes	detector 0	detector 1	array active
1	321	373	641
2	25	260	279
5	84	134	201
10	0	0	0

References

[1] T.G. Barbounis, J.B. Theocharis, M.C. Alexiadis, and P.S. Dokopoulos. Long-term wind speed and power forecasting using local recurrent neural network models. *Energy COnversion, IEEE Transactions On*, 21, 2006.

[2] South Pole Meterology Department. Weather data. `ftp://amrc.ssec.wisc.edu/pub/southpole/surface_observations/one_minute/`.

[3] South Pole Meterology Department. Met handout. `http://amrc.ssec.wisc.edu/usap/southpole/met-handout.pdf`, May 2011.

[4] Erich Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics*. Springer; 2nd edition, 2005.

[5] B.S & Carroll J.J Jackson. Aerodynamic roughness as a function of wind direction over asymmetric surface elements. *Boundary-layer Meteorology*, 1978.

[6] Donald E. Knuth. *The TEXbook*. Addison-Wesley, 1984.

[7] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[8] James R. Phillips. Zunzun curve fitting website.

[9] April 30) Plataforma SINC (2009. Neural networks used to improve wind speed forecasting. *Science Daily*, 2009.

[10] K & Ramakanthkumar P. Sreelakshmi. Neural networks for short term wind speed prediction. *World Academy of Science, Engineering and Technology*, 42, 2008.

[11] Wikipedia. Density of air, 2011.