Online Solar Radiation Forecasting under Asymmetric Cost Functions

Seyyed A. Fatemi, Anthony Kuh, Matthias Fripp
Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, USA.
E-mails: (sfatemi, kuh, mfripp)@hawaii.edu

Abstract—Grid operators are tasked to balance the electric grid such that generation equals load. In recent years renewable energy sources have become more popular. Because wind and solar power are intermittent, system operators must predict renewable generation and allocate some operating reserves to mitigate errors. If they overestimate the renewable generation during scheduling, they do not have enough generation available during operation, which can be very costly. On the other hand, if they underestimate the renewable generation, they face only the cost of keeping some generation capacity online and idle. So overestimation of resources create a more serious problem than underestimation. However, many researchers who study the solar radiation forecasting problem evaluate their methods using symmetric criteria like root mean square error (RMSE) or mean absolute error (MAE). In this paper, we use LinLin and LinEx which are asymmetric cost functions that are better fitted to the grid operator problem. We modify the least mean squares (LMS) algorithm according to LinLin and LinEx cost functions to create an online forecasting method. Due to tracking ability, the online methods gives better performance than their corresponding batch methods which is confirmed using simulation results.

I. INTRODUCTION

Balance of load and generation is necessary for the electric grid. Power system balancing authorities (BAs) at each hour estimate the loads and schedule for generation of conventional power plants. Integration of renewable generation to the grid has been increasing in recent years. However, renewable sources such as solar and wind are intermittent. Since it takes time to start additional conventional power plants, grid operators must predict the intermittent generation as well as load and commit enough generation resources in advance. To mitigate forecasting errors, system operators may bias the renewable generation forecast downward, allocating some operating reserves to ensure that during operation, generation always meets load.

Underestimation means that true renewable generation during operation time is more than what the BA forecast. So the committed generation capacity is more than load. In this case during operation, output from conventional power plants is decreased below their maximum level, so that generation equals load [1].

On the other hand, overestimation means that true renewable generation during operation time is less than what the BA forecast. So the committed generation capacity is not enough to meet load. In market-based power systems, this can result in extremely high prices in the balancing market. If additional generation is not available, the overestimation may lead to an area control error (ACE), drawing unscheduled power from neighboring BAs. In extreme cases or on isolated power systems, the BA may need to disconnect firm loads. ACE or load shedding are very undesirable for the BA and/or customers [2]. These also correspond to high economic costs, manifest as fines paid by the BA for violating reliability standards, or the loss by customers of a valuable resource.

Put simply, in the case of overestimation of renewable generation, the BA will encounter shortages of generation and even may be forced to shed loads; however, in the case of underestimation of generation, they can curtail the excess generating capacity. So overestimation is a more serious error than underestimation.

Therefore the solar and wind generation forecasting problem in the BA’s view is not symmetric. However, the International Energy Agency (IEA) recommends the use of RMSE, Mean Bias Error (MBE), and Kolmogorov Smirnoff Integral (KSI) metrics; many authors who study solar irradiation forecasting tried to minimize the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) in their methods which are symmetric for both underestimation and overestimation [3].

In 1969, Granger mentioned that in many practical problems in economics the cost function is asymmetric. He introduced the LinLin function as an asymmetric piecewise linear function and suggested a useful although sub-optimal way for considering asymmetry by adding a constant bias value to the predictor [8]. The LinLin loss function is the simplest asymmetric cost function we can use to distinguish between overestimation and underestimation. However, it is unable to represent cases where the per-unit cost increases as the magnitude of the error increases.

The second popular asymmetric loss function is LinEx which was originally introduced for real estate assessment [9] and comprehensively discussed by Zellner. Many other applications require the use of asymmetric cost function. For example in dam construction underestimation of peak water level is more serious than overestimation [10]. In estimation of average life of the components of a spaceship, overestimation is usually more serious than underestimation [11]. In this study we also have that overestimation of renewable generation is more serious than underestimation. The LinEx function could be a useful proxy for the costs of generation shortfall, which may rise as the size of error increases. For example, as the size of the error increases, increasingly expensive emergency
resources may need to be brought online, or increasingly valuable loads may need to be shed (if load shedding is staged according to economic value).

Previously, we considered both LinLin and LinEx cost functions as more suitable functions for the utility problem. We showed that a batch forecasting method which optimizes directly against the loss function gives a better result than adding the optimal bias to an unbiased forecast [12].

Online algorithms are interesting because of suitability for real time applications, tracking gradual changes in statistics of input and hardware configuration [13],[14]. Hence in this paper, we implement online algorithms based on the stochastic gradient descent or least mean square (LMS) algorithm. Simulation results shows that the online methods give improvement over corresponding batch results. The rest of the paper is organized as follows. In section II, the forecasting problem is formulated as an optimization problem; the hypothesis model using solar zenith angle is discussed and the use of LinLin and LinEx cost functions are justified. Section III uses online methods in which we revise the LMS algorithm according to LinEx and LinLin to create an online method for forecasting under these asymmetric cost functions. Section IV is dedicated to the simulation results and discussion. A summary of results and conclusion is presented in section V.

II. PROBLEM STATEMENT

Our objective is to minimize expected loss by adjusting forecasting hypotheses parameters. Let the actual solar radiation at time \( n \) be \( x_n \) and the corresponding forecast be \( \hat{x}_n \). We are interested in \( k \) step ahead forecasting using a window of past observations. Then the optimization problem is

\[
\text{Minimize } \sum_{i=0}^{M} \text{Loss}(\hat{x}_{i+k} - x_{i+k})
\]

where \( \text{Loss} \) is loss function either LinLin or LinEx, \( m \) is the window size, and \( M \) is the total number of samples.

In the subsection A, a weight function that uses the solar zenith angle is explained. We justify the use of LinLin and LinEx that are asymmetric cost functions in the utility scheduling problem in subsection B.

A. Hypothesis Model

Solar irradiation is a random process which consists of some deterministic and some random parameters. One of the dominant deterministic parameters in solar radiation is sun position and in particular zenith angle. Zenith angle \( \theta_z \) is the angle between sun beam and perpendicular line on horizontal surface. Every day the sun rises when \( \theta_z = 90^\circ \) and gradually decreases until around noon and then starts to increase until again reaches \( \theta_z = 90^\circ \) when the sun sets. Time of sunrise and sunset is not constant during a year, similarly, zenith angle at specific time of the day changes during a year. For example, zenith angle at winter noon time is larger than summer noon time. Zenith angle is deterministic and calculated from the position of the earth with respect to the sun. The formula for the zenith angle is given by equation (1) which is discussed in [15]

\[
\cos \theta_z = \cos \phi \cos \delta \cos \omega + \sin \phi \sin \delta \quad (1)
\]

where \( \phi \) is latitude; \( \omega \) is solar time which is negative in mornings, zero at noon and positive in afternoons, and changes by 15°/hour rate; and \( \delta \) is declination angle given by [16]

\[
\delta = 23.45 \sin \frac{360(284 + d)}{365}
\]

where \( d \) is the Julian day number.

Note in Figure 1 the relationship between the cosine of the zenith angle \( \cos \theta_z \) and the solar irradiation for three different days (partly sunny day in summer, sunny day in winter, and a day with a sunny morning and cloudy afternoon). There is an approximately linear relationship between the \( \cos \theta_z \) and the solar irradiation. If we divide the solar irradiation by the cosine of the zenith angle we remove seasonal effects caused by the change of the sun position. (Seasonal effects caused by cloud patterns will not be removed.)

Hence we use forecasting method using zenith angle which was introduced in [17]. So our hypothesis \( h \) is linear combination of past data converted to the time of prediction:

\[
\hat{x}_{n+k} = (\alpha_0 + \frac{\alpha_1 x_n}{\cos \theta_z(n)} + ... + \frac{\alpha_{m} x_{n-m+1}}{\cos \theta_z(n-m+1)}) \cos \theta_z(n+k)
\]

(2)

Fig. 1. Relationship of \( \cos \theta_z \) to solar irradiation for three different days in Hawaii (for sunny days relationship between two quantities is linear).

where \( \theta_z(n) \) is solar zenith angle at time \( n \) and \( \alpha_0, \alpha_1, ..., \alpha_m \) are the weight parameters.

B. The Cost Functions

If the BA ignores all intermittent generation (i.e. forecasts zero output from these sources), it will schedule enough operating reserves at all times. However, unused reserves cost about 20% of per unit price of energy (i.e. in Hawaii about $0.05/kWh) [18]. So forecasting of intermittent generation is useful to avoid that cost. On the other hand if intermittent generation is overestimated, the BA may encounter shortage of generation and be forced to draw unscheduled power from neighboring BAs, or eventually shed loads. For the BA load
sheding costs are very expensive. The cost of unscheduled power transfers is difficult to assess. The value of lost load (VOLL) due to load shedding is reported to be around $8/kWh to $24/kWh [19][20][21]. For this study we assume VOLL to be $10/kWh.

Let $\epsilon$ be the forecast error given by

$$\epsilon_{n+k} = \hat{x}_{n+k} - x_{n+k}$$

So overestimation, which means the predicted value exceeds the actual value, corresponds to a positive error and underestimation ($0.06/kWh loss of revenue) and overestimation ($10/kWh penalty fee) as asymmetric trade off between underestimation ($0.06/kWh per unit revenue) and overestimation ($10/kWh penalty fee) lead to a LinLin loss function.

$$LinLin(\epsilon) = \begin{cases} C_1 \epsilon & \text{ if } \epsilon > 0, \quad C_1 \approx 10/kWh \\ -C_2 \epsilon & \text{ if } \epsilon \leq 0, \quad C_2 \approx 0.06/kWh \end{cases}$$

(3)

Fig. 2. LinLin ($C_1 = 10/kWh$, $C_2 = 0.06/kWh$) and LinEx ($b = 2$$/h , a = 0.03/kWh$) cost functions.

The system usually is robust so that it can tolerate small errors. Hence as a second case, we assume the load shedding cost is exponentially distributed among errors in the way that small errors pay less penalty fee but larger errors pay a more expensive penalty fee. In this case we have the LinEx loss function given by

$$LinEx(\epsilon) = b(e^{a\epsilon} - a\epsilon - 1)$$

(4)

The $a$ and $b$ constants are called shape factor and scale factor respectively. In Fig. 2 the two cost functions in equation (3) and (4) are shown. The shape and scale factor are selected so that they are consistent with LinLin for underestimation errors and exceeds the LinLin cost for overestimation errors more than a predetermined value for example 25%. (The 25% is reasonable since many utilities allow the percentage of distributed generation to load to be 15% or more [22].)

III. ONLINE METHODS

In many instances information about data is not complete and data statistics may be nonstationary. In these cases online learning algorithms which continually update their weights often give superior performance over the batch algorithms [23] that we formulated in [12]. The algorithmic simplicity of online algorithms is also an important issue when we deal with large scale problems [24] especially in real time. Online algorithms are also more preferable in terms of hardware implementation due to high modularity [13].

Therefore in this section we implement the online forecasting methods based on stochastic gradient descent. The first subsection is devoted to online formulation for LinLin cost function and the second subsection is dedicated to the LinEx cost function.

A. LinLin Cost Function

Similar to the least mean squares (LMS) algorithm which uses the instantaneous estimate of gradient vector for squared error cost, we use the instantaneous estimate of gradient vector for LinLin cost function.

$$\hat{J} = LinLin(\hat{x}_{n+k} - x_{n+k})$$

where $\hat{x}_{n+k}$ is computed using equation (2). Instantaneous estimate of gradient vector computed by following equations.

$$\nabla \hat{J} = [\frac{\partial \hat{J}}{\partial \alpha_0}, \frac{\partial \hat{J}}{\partial \alpha_1}, ..., \frac{\partial \hat{J}}{\partial \alpha_m}]^T$$

To compute the gradient let define function $g(x)$ using following equation

$$g(x) = \begin{cases} C_1 & \text{ if } x > 0 \\ 0 & \text{ if } x = 0 \\ -C_2 & \text{ if } x < 0 \end{cases}$$

The partial derivatives calculated using

$$\frac{\partial \hat{J}}{\partial \alpha_j} = \cos \theta_j(n+k) g(\hat{x}_{n+k} - x_{n+k})$$

In the same way, for $j = 0, 1, 2, ..., m - 1$

$$\frac{\partial \hat{J}}{\partial \alpha_{j+1}} = \frac{x_{n-j} \cos \theta_j(n+k)}{\cos \theta_j(n-j)} g(\hat{x}_{n+k} - x_{n+k})$$

Let $\alpha = [\alpha_0, \alpha_1, ..., \alpha_m]^T$. Then $\alpha$ is iteratively updated by following equation.

$$\alpha_{n+1} = \alpha_n - \eta \nabla \hat{J}$$

The total cost up to time $J(n)$ is cumulative sum of instantaneous cost, $\hat{J}$. So

$$J(n) = J(n-1) + \hat{J}(n)$$

In order to have comparable results we use per unit revenue. Let $R_{perfect}(n)$ be the cumulative revenue of perfect forecasting up to time $n$, then the per unit revenue is given by

$$R(n) = 1 - \frac{J(n)}{R_{perfect}(n)}$$
B. LinEx Cost Function

Similarly, instantaneous LinEx cost is given by
\[
\hat{J} = \text{LinEx}(\hat{x}_{n+k} - x_{n+k})
\]
where \( \hat{x}_{n+k} \) is computed using equation (2). Instantaneous estimate of gradient vector computed by following equations.
\[
\nabla \hat{J} = \left[ \frac{\partial \hat{J}}{\partial \alpha_0}, \frac{\partial \hat{J}}{\partial \alpha_1}, ..., \frac{\partial \hat{J}}{\partial \alpha_n} \right]^T
\]
\[
\frac{\partial \hat{J}}{\partial \alpha_0} = ab[\cos \theta_z(n+k)(e^{a(\hat{x}_{n+k} - x_{n+k})} - 1)]
\]
similarly for \( j = 0, 1, 2, ..., n-1 \)
\[
\frac{\partial \hat{J}}{\partial \alpha_{j+1}} = ab[\frac{x_{n-j} \cos \theta_z(n+k)}{\cos \theta_z(n-j)}(e^{a(\hat{x}_{n+k} - x_{n+k})} - 1)]
\]
The \( \alpha \) is iteratively updated by following equation.
\[
\alpha_{n+1} = \alpha_n - \eta \nabla \hat{J}
\]
Adding a momentum term to the learning rule could increase the learning rate; however, a constant momentum factor (\( \gamma \)) results in oscillation in the learning curve [25]. For this reason we tried decreasing the momentum factor such that at initial iterations the momentum factor is high and leads to faster learning but in later iterations the momentum factor decreases to zero to avoid over learning. As a result
\[
\gamma_n = \frac{\gamma_0}{(1 + \frac{n}{N})}
\]
where \( N \) is the number of samples per year.

The learning algorithm with momentum term is implemented using following equations:
\[
\Delta \alpha_{n+1} = \gamma_n \Delta \alpha_n - \eta \nabla \hat{J}
\]
\[
\alpha_{n+1} = \alpha_n + \Delta \alpha_{n+1}
\]

IV. SIMULATION RESULTS

For simulation, we retrieved solar irradiance data for three sites from http://www.nrel.gov/midcl. The names and details of the sites are shown in Table I. Resolution of the original data for LaOla and Los Angeles is one minute and for Elizabeth City is five minutes. The data removed night hours and low irradiance times in the morning and the evening. So only nine hours per day are considered. In order to get comparable results, we use per unit revenue which is the ratio of the annual revenue to maximum possible revenue which comes from perfect forecast.

In the steepest descent algorithm, several iterations are required to reach the optimal weight vectors and in each iteration all training samples are used; so each sample is used several times. In order to use each sample multiple times in the online method, we use a resampling technique i.e. we use nine yearly datasets of Elizabeth City seven times then the total 63 yearly datasets are randomly ordered and used as input to the online learning algorithm.

Fig. 3 compares the per unit average annual revenue for LinLin cost function using the online and batch forecasting methods with different taps. It reveals that increasing number of taps more than four taps does not give any performance improvement which is similar to our finding in batch method. The per unit revenue is 22% for four taps, which is 4% more than the 18% validation revenue of the corresponding batch method.

Fig. 4 shows the learning curves of the online method with different tap numbers under LinLin cost function; similar to Fig. 3 the difference between one tap and two taps is small and three taps give better results; The best performance is for four taps and more than four taps does not give any improvement. As shown in Fig. 5 the per unit average annual revenue
for LinEx cost function using the online forecasting method, is around 41% which is 3% better than the 38% for the corresponding batch methods. Also, the number of taps does not have a significant effect.

The learning curves of the online method with different tap numbers under LinEx cost function are shown in Fig. 6. While there is not a significant difference in final performance, the method which uses more taps learns faster than others.

As shown in Fig. 7 using a decaying momentum term in the learning rule increased the learning rate. However, using same momentum factor ($\gamma$) for all the taps is not always beneficial. For example here a single tap with $\gamma_0 = 1$ gives superior performance over nine taps with $\gamma_0 = 0.25$. A larger momentum factor for the nine taps method results in oscillation in learning curve due to over learning.

We repeated the same simulations for the datasets for both Hawaii and Los Angeles but with three years of data. For the Hawaii data set with LinLin cost function, we achieved 22% of maximum possible revenue using the linear programming method and the validation revenue decreases for more than one tap where as we could only achieve 9% by adding the optimal bias to the unbiased forecast. The online method gives us 22.5% which is similar to the batch result. For the case of LinEx cost function, 42% of maximum possible revenue is achieved by using two fold cross validation technique that is significantly more than 23% achievement given by adding the optimal bias to the unbiased forecast. The one tap online method with decaying momentum term gives around 45%.

For the Los Angeles data set with LinLin cost function, we earned 31% of maximum possible revenue using the linear programming method and the validation revenue decreases for more than four taps where as we could achieve 25% by adding the optimal bias to the unbiased forecast. The online method gives us 29% which is similar to the batch result. For the case of LinEx cost function, unlike the Elizabeth City and Hawaii results, there is small difference between direct biased forecast and adding the optimal bias to the unbiased forecast. The 47% of maximum possible revenue earned using the online method is similar to the batch results.

**V. CONCLUSION**

While many researchers have studied the problem of forecasting of solar radiation, they evaluated their methods using symmetric criteria like root mean square error (RMSE) or mean absolute error (MAE). However, grid operators have more concern about shortage of production rather than its abundance, i.e. overestimation of resources has more serious consequences than underestimation. So in the BA’s view the cost function is not symmetric. For this reason we discussed solar radiation forecasting under LinLin and LinEx as asymmetric cost functions which are better fitted to the grid operator problem. We revisited the least mean square (LMS) algorithm according to LinLin and LinEx cost functions to create an online method. Using decaying momentum term in the learning rule increases the rate of learning. The proposed online method gives an improvement over batch solutions due to better tracking ability. Simulation results also shows that higher revenue is achieved with LinEx cost function versus LinLin cost function.

Here we discussed forecasting methods for single sites using their past solar radiation observations. The forecasting method for multiple sites and incorporating exogenous data
like weather forecasts are also interesting and left for future research.

REFERENCES