# A New Recursive Dynamic Factor Approach to Solar Radiation Forecasting with Multi-Cluster Learning and Bayesian Model Averaging

YUAN Yang

MSc. THESIS

THE UNIVERSITY OF HONG KONG

Abstract of thesis entitled

# A New Recursive Dynamic Factor Analysis in Solar Radiation Forecasting with Multi-Cluster Learning and Bayesian Model Averaging

Submitted by

YUAN Yang

For the degree of Master of Science

At the University of Hong Kong

In August 2012

With extremely abundant resources and environmentally- friendly characteristics, the solar energy is widely welcomed. However, solar energy is variable and its integration to the conventional power systems presents some difficulties. So it is desirable to predict how much solar energy will be available in the next day so that the required fossil fuels and other energy sources can be estimated to meet the customers' need. Because the information about the amount of global solar radiation (GSR) is essential to the solar energy development and the meteorology research [1], in this study, we propose a new recursive dynamic factor analysis (RDFA) algorithm to forecast the solar radiation and update the dataset every day. We compare the proposed RDFA algorithm with the existing methods proposed in [2] using public available dataset and found that the RDFA method has better forecasting results. Not only can it give satisfactory prediction within 4 hours, it can also yield good results in a whole day with an estimated confident interval. The latter is very important in solving the unit commitment problem for allocating the required energy sources to meet the users' demand. Another advantage of the proposed RDFA method is its low arithmetic complexity and simple real-time updating, which is different from other conventional algorithms. Then, based on the nature of solar radiation and the RDFA algorithm, we put forward another method based on multi-cluster learning and combine their prediction results by the Bayesian model averaging. Forecasted results using existing database show that this new method can further improve the forecasting accuracy.

# Declaration

I hereby declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

YUAN Yang

August 2012

# Acknowledgements

Foremost, I most like to express my sincere gratitude to my supervisor Prof. S. C. Chan and Mr. Andrew Wu for the continuous support and guidance in writing this dissertation, for their patience, motivation, enthusiasm, and immense knowledge. This thesis would not have been finished without their help.

Beside my supervisor, I would like to thank one of my teachers Mr. Andrew Wu, who gave the most care during my Msc study. I would like to thank everybody in Digital Signal Processing Laboratory CYC 722 for their insightful comments and encouragement, and the jubilant companion. They are all great and helpful.

My sincere thanks go to my friends who conduct the Msc study in EEE with me and share the most unforgettable moments with. Last but not least, I would thank my family, for their strong belief, their endless supports and love no matter what happens.

## Table of Content

# List of Figures

# List of Tables

# List of Abbreviations

| Abbreviations | Full Name |
|---|---|
| AR | Auto Regression |
| ANN | Artificial Neural Networks |
| ARIMA | Autoregressive Integrated Moving Average |
| DFA | Dynamic Factor Analysis |
| DFM | Dynamic Factor Model |
| ED | Eigen-decomposition |
| FA | Factor Analysis |

| | |
|---|---|
| FCM | Fuzzy C-means |
| KF | Kalman Filter |
| LMS | Least Mean Square |
| MAPE | Mean Absolute Percentage Error |
| MDL | Minimum Description Length |
| MLE | Maximum Likelihood Estimation |
| MMSE | Minimum Mean Square Error |
| MSE | Mean Squared Error |
| OPASTr | Othonormal Projection Approximation Subspace Tracking with Rank-1-modification |
| PCA | Principal Component Analysis |
| RLS | Recursive Least Square |
| VAR | Vector Auto Regression |
| RDFA | Recursive Dynamic Factor Analysis |
| UCM | Unobserved Components Models |

# Chapter 1 Introduction

The energy shortage and environment problem became increasingly serious in recent years. Therefore, there is an urgent need to look for renewable and clean energy as the representatives of fossil energy. Solar energy, with its advantage of abundant reserves, no transportation problem, and cleanness, was widely welcomed, and photovoltaic power is an effective mean how solar energy utilization is utilized. The solar radiation, as the main decisive factor of power generated main from the solar, has a significant effect on the site selection, energy control and planning. As an important parameter to measure the solar radiation, the information about the amount of global solar radiation (GSR) is essential for the solar energy development and the meteorology research [1].

Different methods for predicting the output photo power of voltaic panels had been proposed due to the increasing need on solar energy and better integration to the existing power systems. A traditional approach is based on autoregressive (AR) model in time series theory. The AR model can be specifically solved using the ordinary least squares (LS). Other variants such as the autoregressive integrated moving average (ARIMA) methods have also been used in solar radiation research [2]. For ARIMA, the processing speed is fast and the algorithm is easy to implement. However, when it comes to the non-stationary time series analysis, the performance may be significantly degraded, which results in large prediction error. Unlike ARIMA models, models with unobserved components may have no concern on non-stationary situation. Some of the most explicit benefits of unobserved component models become obvious only when more complex problems are considered [3]. On the other hand, nonlinear variability in the data has led to people's considerable interest in neural networks [4]. Besides, to combine regressions and neural nets, as in the emerging class of hybrid models is put forward by Gordon Reikard. Hybrid method has potentially higher accuracy than other single forecasting method, but its computational complex is high [5].

## Proposed Method

According to recent research [1], several common behaviors of solar radiation are: 1) strongly seasonal feature of values; 2) adjacent value correlated and 3) to a large extension dependent on meteorological information. It is advantageous to use different models to describe the solar radiations in different days because solar power from the panels in sunny, rainy and cloudy days are considerably different. For example, we can divide the time series values into 24 partitions (hours), i.e. hourly value, where each partition represents a particular hour of all days and can be treated as a separate variable. This concept can be extended to other time scales such as half-hourly data. Since the pattern of the value for adjacent partitions will be highly similar, the application of factor analysis (FA) techniques such as principal component analysis (PCA) is becoming an important research area for such problems. The usefulness of PCA for solar radiation forecasting is its ability to explore and capture the correlation between adjacent periods of interest.

One of the major challenges of solar radiation forecasting is to tackle its time-varying nature. To improve adaptability, online batch processing [6] is usually desirable, where the forecasting is performed by applying the forecasting algorithm to a data block making up of consecutive solar radiation samples. Whenever a new data is available, the existing data block is appended with the new sample and the earliest sample is discarded. This procedure is repeated for each incoming sample or blocks of samples in each update. In this regard, the forecasting algorithm can better adapt to possible changes of trend. However, this may also lead to high arithmetic complexity.

In this paper, to cope with the high arithmetic complexity incurred by such online real-time estimation, we propose a new recursive dynamic factor analysis (RDFA) algorithm for its implementation. It employs efficient recursive subspace tracking and Eigen-decomposition (ED) algorithms to compute the PCs and PC scores in the PCA. Since only the most recent sample is used for the updating, the memory storage required is also reduced [7]. Moreover, we consider a dynamic factor model where the PC scores are modeled as AR processes. By assuming that the innovation is Gaussian distributed, the Kalman filter (KF) algorithm can be used to recursively tracked the PC scores. This also allows the covariance and hence the confident intervals of the forecasted values to be estimated.

An outline and major advantages of the RDFA algorithms [8] are summarized below: RDFA algorithm first employs an efficient subspace tracking algorithm, called the orthonormal projection approximation subspace tracking with rank-1-modification (OPASTr) [8], to track recursively the major subspace spanned by the PCs. This reduces memory requirement and arithmetic complexity.

After the subspace of major components has been tracked, major eigenvectors (PCs) and PC scores in the subspace required are recursively updated by rank-1 modification [9] in the subspace. Consequently, this leads to lower arithmetic complexity because it only works on the major subspace with a much smaller dimension.

It also allows an interval forecast of the solar radiation be derived from the PCs by modeling the PCs scores as AR models. By incorporating a regularization term to the LS estimation of the AR model, the estimation variance can be reduced. The problem can be reformulated as the state estimation problem of a linear state-space model, which can be recursively solved using the Kalman Filter (KF) [10]. A major advantage of the KF is that it can provide a density estimate of the AR coefficients, which allows us to compute the confident interval of the forecast.

Finally, a new multi-clustering model is proposed to classify the solar radiation in the existing data block into different say $k$ groups or clusters, such as rainy, cloudy, sunny etc. Whenever a new sample is available, we can firstly obtain $k$ predicted values respectively from the $k$ models. Then chi-square distribution [11] is used to calculate the probability of each model. Based on Bayesian model averaging [12], a better predicted result is obtained.

Experimental results shows that the proposed approach, which combines the aforementioned subspace tracking algorithms and the KF, is able to achieve better daily ahead forecast accuracy than other conventional approaches for the experimented data set [10]. Moreover, by combining the proposed RDFA algorithm with the proposed multi-clustering model, a real-time updating model based on multi-cluster learning with higher accuracy on solar radiation forecast is obtained.

The thesis is organized as follows: The background of solar radiation forecasting algorithms are introduced in Chapter 2. In Chapter 3, the RDFA algorithm in [8] is discussed. Afterwards, its application to solar radiation forecasting is compared to the existing methods in Chapter 4. Finally, a new multi-cluster model based on RDFA algorithm is presented in Chapter 5. Conclusions are drawn in the Chapter 6.

# Chapter 2 Background

According to recent research [1][5],the fluctuating nature of solar radiation will require reliable forecast information of its availability in various time and spatial scales depending on the application. First attempts in irradiance forecasting have been presented more than twenty years ago [7]. Since then, considerable numbers of research are related to solar radiation forecasting. And many different methods had been approved due to the increasing need on solar research. Chowdhury and Rahman used a statistical autoregressive ARIMA model to forecast sub-hourly irradiance [13]. A two-dimensional (2-D) representation model of the hourly solar radiation data is proposed in [14]. In the last two decades, many other methods, ranging from regressions to unobserved components models (UCM), are proposed aiming at improved forecasting result [3]. In this thesis, we compare the proposed RDFA method with six popular algorithms in [5], which are ARIMA, UCM, Regression Model, ANN, transfer function and hybrid model combining the regressions and neural nets.

## 2.1 Literature review

### 2.1.1 ARIMA

The ARIMA (Auto-Regressive Integrated Moving Average) method is a widely used reference estimator in the prediction of global solar radiation field [13]. It is a parametric stochastic process which includes both the autoregressive (AR) and moving average (MA) components. It is usually applied to auto correlated time series data. It is a valuable tool for understanding and predicting the future value of a specific time series.

Let us denote the auto-regressive integrated moving average model by the notation ARIMA ( $p,d,q$ ), which is given mathematically as follows:

$$\phi_p(B)y_t = \theta_q(B)e_t \tag{1}$$

where $e_t$ is a white noise, $\phi_p(B)$ is an autoregressive operator of order $p$ , $\theta_q(B)$ is a moving average operator of order $q$ , $y_t$ is the output and $B$ is the backward difference operator. The general ARIMA model can be generated from white noise by means of three filtering operations, as indicated in Fig.1. The first filter has input $\varepsilon_t$ , transfer function $\theta_q(B)$, and output

$$x_t = e_t - \beta_1 e_{t-1} - \boxed{?} - \beta_q e_{t-q} = \theta_q(B)\mathrm{e}_t \qquad (2)$$

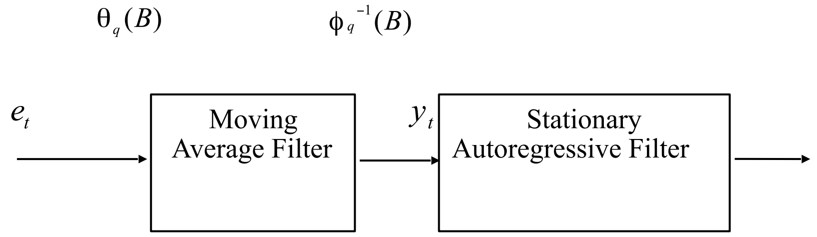


**Fig.1 A general ARIMA model represented by a series of three linear filters**

ARIMA models are, in theory, the most general class of models for forecasting a time series which can be stationary by transformations such as differencing and logging. Reikard applied a regression to the logarithm of the inputs of the ARIMA models to predict solar radiation [13]. He compares ARIMA models with other forecast methods such as Artificial Neural Networks (ANN) [15]. At the 24-h horizon, he states that the ARIMA model captures the sharp transitions in irradiance associated with the diurnal cycle more accurately than other methods. The main advantage of ARIMA is fast processing speed and simple algorithm. However, the main disadvantage of ARIMA compared to other conventional models is that forecasts are unreliable with data series less than 50 data points. Moreover, ARIMA models are more sensitive to presence of outliers in the original series.

### 2.1.2 Unobserved Components models (UCM)

A UCM decomposes the response series into components such as trend, seasons, cycles, and the regression effects due to predictor series. The following model shows a possible scenario:

$$y_t = \mu_t + \gamma_t + \psi_t + \sum_{j=1}^{B} \beta_j \chi_{jt} + \varepsilon_t \qquad (3)$$

where $\varepsilon_t$ is the disturbance term, also called the irregular component, which is usually assumed to be Gaussian white noise, $\beta_i$ is the regression coefficients, and $\chi_{jt}$ is regression variables. The terms $\mu_t$ , $\gamma_t$ and $\psi_t$ represent the trend, seasonal, and cyclical components, respectively. In fact the model can contain multiple seasons and cycles, and the seasons can be of different types. For simplicity of discussion the preceding model contains one term from each of these components. A model can also contain regression variables that have time varying regression coefficients or that have a nonlinear relationship with the dependent series. In some cases disturbance term $\varepsilon_t$ is useful to model the irregular component as a stationary ARMA process. By controlling the presence or absence of various terms and by choosing the proper flavor of the included terms, the UCMs can generate a rich variety of time series patterns [16]. A UCM can be applied to variables after transforming them by transforms such as log and difference.

Some of the benefits of unobserved component models become apparent only when more complex problems are considered [16]. Some multivariate models provide interpretation of the components and insight into the value of, such as, using auxiliary series to improve the effect and of forecasting a target series. The main advantages of this algorithm are linearity, fast computation, and avoidance of some numerical issues.

### 2.1.3 Regression Models

Regression analysis is widely used for prediction and forecasting [2]. It is also used to understand which independent variables are related to the dependent variable, and to explore the forms of these relationships.

A regression model relates the response variable *Y* to a function of exploratory variable *X* and model parameters $\beta$ .

$$Y = f(\mathrm{X}, \beta) + \varepsilon \tag{4}$$

The function $f$ is called the regression curve and describes the overall trend in the scatter plot; that is, it is the function that relates the response variable to the explanatory variables $X$ . $\varepsilon$ is called the random error which may arise from modeling error, etc. It is usually assumed to have zero mean. Since the random variation in the observation $Y$ is modeled by random errors $\varepsilon$ , $Y$ is a random variable with means $f(x, \beta)$ and constant variance (equal to the variance of $\varepsilon$ ).

Many authors have developed empirical regression models to predict the hourly, daily or monthly averaged global solar radiation in different regions [17]. These diverse regression models include linear, logarithmic, quadratic, third order polynomial, exponential and power models and so on.

### 2.1.4 Artificial Neural network

Many researchers estimated global solar radiation by using artificial neural networks (ANN).

The ANN models have been applied to real meteorological data [18]. A neural network is a massively parallel distributed processor made up of simple processing units that aims to learn the natural input and output relationship from training data. ANNs have the ability to model linear and non-linear systems without the need to make assumptions explicitly in most traditional statistical approaches, and it is widely applied to different aspects of science and engineering [18]. Although neural network has good non-linear approximation ability, when training for a long time, the weights and the threshold are sensitive to the initial value, and generalization ability is rather poor.

Mathematically, a neuron's network function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the nonlinear weighted sum, where $f(x) = K(\sum_i \omega_i g_i(\mathrm{x}))$, with $K$ (commonly referred to as the activation function) some predefined function such as the hyperbolic tangent.

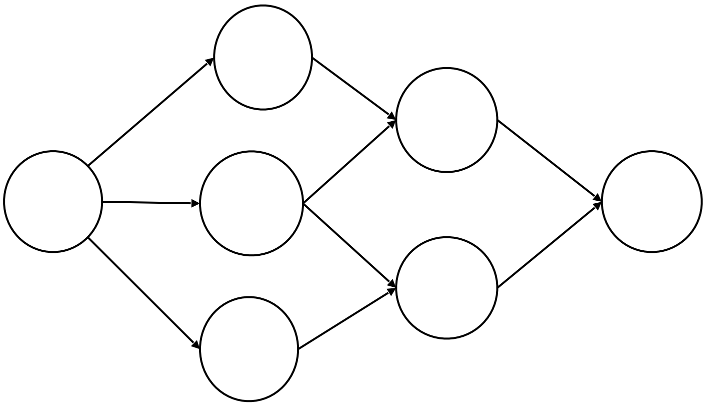


**Fig.2 ANN Dependency Graph**

Fig.2 depicts an example of a decomposition of information in two layers, with dependencies between variables indicated by arrows. Since the components of individual layers are independent of each other, this naturally enables a degree of parallelism in the implementation.

## 5. Hybrid Models

The use of hybrid models has become more popular as it takes advantages of different models [5]. The basic idea of the model combination is to use each model's unique features to capture different patterns in the data. Both theoretical and empirical findings suggest that combining different models can be an efficient way to improve the forecast performance.

## 2.2 The Proposed Method

Although the methods above have different advantages on the forecasting, and every of them are used widely. The main insufficiency is that most methods reported so far cannot realize real-time updating and forecasting. So we propose a new method for online prediction of the solar radiation over a certain period say three hours or one day and update the database every day.

The usefulness of principal component analysis (PCA) for solar radiation forecasting has been reported in [6]. It is able to explore and capture the correlation between adjacent periods of interest. For example, an intraweek seasonal cycle exhibits similarity of the demand from one week to the next.

One of the major challenges of the solar radiation forecast is to tackle the time-varying nature of the solar radiation [2], which can be solved by online batch processing, i.e. prediction is performed by applying the forecasting algorithm to a data block be comprised of consecutive solar radiation data. Whenever a new data sample is available, it will enter into the end of the data block and the earliest sample is abandoned. This procedure is repeated continuously in each update. In this regard, the forecasting algorithm can change of trend according to the change of data. However, this may also lead to high arithmetic complexity.

To cope with the high arithmetic complexity incurred by such online real-time solar radiation estimation, we propose a new recursive dynamic factor analysis (RDFA) algorithm. It employs efficient recursive subspace tracking and eigen-decomposition (ED) algorithms to compute the principal components (PCs) and PC scores in the PCA. Since only the most recent sample is used for the updating, the memory storage required is also reduced. Moreover, we consider a dynamic factor model where the PC scores are modeled as AR processes. By assuming that the innovation is Gaussian distributed, the Kalman filter (KF) algorithm can be used to recursively tracked the PC scores. This also allows the covariance and hence the confident intervals of the forecasted values to be estimated. Finally, the framework is generalized to a multi-clustering model to further improve its performance for large horizon forecasting including weekly or longer correlations.

An outline and major aspects of the proposed algorithms are outlined below:

The proposed RDFA algorithm first employs an efficient subspace tracking algorithm, called the orthonormal projection approximation subspace tracking with rank-1-modification (OPASTr) [9], to track recursively the major subspace spanned by the PCs. This reduces memory requirement and arithmetic complexity.

After the subspace of major components has been tracked, major eigenvectors (PCs) and PC scores in the subspace required are recursively updated by rank-1 modification [9] in the subspace.

Consequently, this leads to lower arithmetic complexity because it only works on the major subspace with a much smaller dimension.

A new interval forecast of the solar radiation is derived from the PCs by modeling the PCs scores as AR models. By incorporating a regularization term to the LS estimation of the AR model, the estimation variance can be reduced. The problem can be reformulated as the state estimation problem of a linear state-space model, which can be solved using the KF recursively. A major advantage of the KF is that it can provide a density estimate of the AR coefficients, which allows us to compute the confident interval of the forecast.

A new multi-cluster model is proposed to separate the solar radiation in the existing data block into different groups ($k$ clusters). Whenever a new sample is available, we can firstly obtain $k$ predicted values respectively from the k models. Then chi-square distribution is used to calculate the probability that each model may occur. Based on Bayesian model averaging, the previous predicted result is corrected.

Though subspace tracking has been reported before for signal array processing communication [18] and recently fault detection [19], the incorporation of interval estimation, and its application to solar radiation forecasting is to our best knowledge new. Experimental results shows that the proposed approach, which combines the aforementioned subspace tracking algorithms and the KF, is able to achieve better daily ahead forecast accuracy than other conventional approaches for the experimented dataset [10]. Moreover, by combining the proposed RDFA algorithm with the proposed multi-clustering model, a real-time updating model using multi-cluster learning with higher accuracy on solar radiation forecast is obtained.

# Chapter 3 The RDFA algorithm

## 3.1 Auto-regression model

According to [1], solar radiation varies over time due to the change of atmosphere condition. Many factors, such as relative humidity and cloud cover cause the solar to fluctuate. The solar radiation $z(n)$ is modeled as an AR process of order $L$ :

$$x_s(n) = \sum_{i=1}^{L} x_s(n-i)\alpha_i + e(n), \qquad (5)$$

Where, $\alpha_i, i = 1,2,\boxed{?}L$ are the AR coefficients and $e(n)$ is the modeling error. The AR coefficient $\alpha_i$ in (2) can be determined by solving the following LS problem [20]:

$$\min_{\boldsymbol{\alpha}}.(\| \boldsymbol{x}_n - \boldsymbol{x}_{n,d} - (\boldsymbol{X}_{n-1} - \boldsymbol{X}_{n-1,d})\boldsymbol{\alpha} \|_2^2), \qquad (6)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \boxed{?}, \alpha_L]^T$, $\boldsymbol{x}_n = [x(L+1), x(L+2), \boxed{?}, x(n)]^T$, and

$$\boldsymbol{X}_{n-1} = \begin{bmatrix} x(L) & x(L-1) & \boxed{?} & x(1) \\ x(L+1) & x(L) & \boxed{?} & x(2) \\ \boxed{?} & \boxed{?} & \boxed{?} & \boxed{?} \\ x(n-1) & x(n-2) & \boxed{?} & x(n-L) \end{bmatrix}$$

. Here, $\boldsymbol{x}_{n,d}$ and $\boldsymbol{X}_{n,d}$ are defined similarly as $\boldsymbol{x}_n$ and $\boldsymbol{X}_n$ respectively. The LS solution to (3) is

$$\boldsymbol{\alpha} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\boldsymbol{x}_n - \boldsymbol{x}_{n,d}), \qquad (7)$$

where $\boldsymbol{A} = \boldsymbol{X}_{n-1} - \boldsymbol{X}_{n-1,d}$ and the superscript $T$ denotes matrix transposition.

Straight forward application of the AR model to solar radiation and other forecasts may ignore their seasonal or daily correlation. Such periodic correlation can be explored using principal component analysis (PCA) and tracking the associated quantities overtime.

## 3.2 Principal Component Analysis

It was shown in [6] that modeling the solar radiation of different periods of a day with different time series models are generally more accurate because the change of solar radiations during each day in the same model is similar. For example, the solar radiation data can be divided into 24 partitions (hours) for hourly solar radiation, where each partition represents a particular hour of all days and can be treated as a separate variable. More precisely, consider $J$ samples of the solar radiation are collected at a regular interval at the $n-th$ day, $n=1,2,\boxed{?},N$ The $J$ samples can be grouped into a vector

$$z(n)=[x((n-1)J+1),x((n-1)J+2),\boxed{?},x(nJ)]^T \qquad (8)$$

Where $x$ is the hourly solar radiation and $J$ is the number of partitions/variables, which is chosen as $J=24$ for hourly solar radiation, then each vector represents the hourly solar radiation for the $n-th$ day. The value of $J$ can also be adjusted for other time scales, such as $J=48$ for half-hourly data. Since the solar radiation for adjacent partitions may be correlated, it is advantageous to apply the PCA reported in [6], which is able to explore and capture the correlation between adjacent partitions.

In the PCA [6], the solar radiation vector $z(n)$ can be approximated by a linear combination of orthogonal basis functions and their associated coefficients, which are referred to as principal component (PC) and PC score respectively. With an appropriate segmentation of the data, correlation between adjacent periods of interest can be explored and captured. For example, the intra-day solar radiation exhibits similarity from one day to the next. Since each PC function is orthogonal, separate time series can be used to model and predict the PC score coefficients and hence the solar radiation across different days. More precisely, suppose we are given the solar radiation vectors of $N$ days, i.e. $z(n)$ $n=1,2,\boxed{?},N$ . $z(n)$ is usually "centered", i.e. with its mean removed, before the PC functions are computed. Hence, the mean of $z(n)$ , $n=1,2,\boxed{?},N$ is first computed and is subtracted from each of the measurement vector to form $\bar{z}(n)$. Let the solar radiation after centering be $\boldsymbol{Z}=[\bar{z}(1),\bar{z}(2),\boxed{?},\bar{z}(N)]^T$ . In PCA, we wish to express the centered solar radiation vector $\bar{z}(n)$ in terms of B PCs: $\bar{z}(n)=\sum_{m=1}^{B}t_m(n)\boldsymbol{p}_m+\boldsymbol{e}(n)$ where B is an appropriately chosen number of PCs to achieve a sufficiently small approximation error $e$, $\boldsymbol{p}_m$ is the m-th PC, and $t_m(n)$ is its associated score for $\bar{z}(n)$. Hence, $\boldsymbol{Z}$ can be written as:

$$Z=\sum_{m=1}^{B}t_m p_m^T+E=\Gamma P^T+E \qquad (9)$$

Where $\boldsymbol{t}_m = [t_m(1),...,t_m(N)]^T$, $\boldsymbol{\Gamma} = [\boldsymbol{t}_1,....,\boldsymbol{t}_B]$ is the score matrix, and $\boldsymbol{P} = [\boldsymbol{p}_1,....,\boldsymbol{p}_M]^T$ is the collection of PCs or loading matrix, and $\boldsymbol{E} = [\boldsymbol{e}(1),\boldsymbol{e}(2),\boxed{?},\boldsymbol{e}(N)]^T$ is the error matrix. A common way to determine the PCs is to compute the ED of the empirical correlation matrix: $\boldsymbol{C}_{zz} = \frac{1}{n-1}\boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$ where the columns of $U$ are the eigenvector and they are also the PCs and $\boldsymbol{\Lambda} = diag\{\lambda_1,...,\lambda_J\}$ contains the eigenvalues in descending order of magnitude $(\lambda_1 \geq \lambda_2 \boxed{?} \geq \lambda_J)$. If the first B largest eigenvalues and their eigenvectors $\boldsymbol{U}_B$ are retained, then one gets $\boldsymbol{P} = \boldsymbol{U}_B$. The subspace spanned by the major PCs $\boldsymbol{P} = \boldsymbol{U}_B$ is usually referred to as the signal subspace. To handle possible system changes, such as the change of the trend, the data collected can be rearranged in a manner such that only the solar radiation of $D$ most recent days is retained, i.e.

$$Z\ (n) = [\mathrm{z(n), z(n-1)}, \boxed{?}, \mathrm{z(n-D+1)}]^{\mathrm{T}} \qquad (10)$$

the correlation matrix can be computed similarly as in (8) , i.e.,

$$C_{zz}(\mathrm{n}) = \frac{1}{\mathrm{D-1}} Z(n)^T Z(\mathrm{n}) = \mathrm{U(n)\Lambda(n)U^T}(n) \qquad (11)$$

for each day $n$ to update the PC.

To perform solar radiation forecasting, time series models can be built for each PC score by AR-based time series models. More precisely, the one day ahead forecast for the solar radiation is

$$\overset{\boxed{?}}{z}(n+1) = \mu(n) + \sum_{m=1}^{B} \overset{\boxed{?}}{t}_m(n+1) p_m(n), \qquad (12)$$

Where $\boldsymbol{\mu}(n) = (1/D)\sum_{i=1}^{D} z(n-D+i)$ is the mean of $\boldsymbol{Z}(n)$ and $\hat{t}_m(N+1)$ is the one step-ahead PC score forecast predicted by time series models, e.g. the AR model in (5). However, the online implementation requires batch eigen-decomposition of the covariance matrix in (11), which requires high arithmetic complexity.

### 3.3 Subspace Tracking

Motivated by the RDFA in [8], the signal subspace can be estimated recursively by the OPASTr algorithm [22] summarized in Table I. Particularly, the signal subspace spanned by the major PCs

$U_B(n)$ is tracked recursively instead of computing the entire ED. In the PAST algorithm [23], the subspace $W(n)$ is recursively computed by minimizing the following objective function:

$$J(\boldsymbol{W}(n)) = \sum_{i=1}^{n} \beta^{n-i} \left\| \bar{\boldsymbol{z}}_s(i) - \boldsymbol{W}(n)\bar{\boldsymbol{y}}(i) \right\|_2^2, \tag{13}$$

Ideally, $\bar{\boldsymbol{y}}(i) = \boldsymbol{W}^T(n)\bar{\boldsymbol{z}}(i)$ and $J(\boldsymbol{W}(n))$ represents the energy in $\bar{\boldsymbol{x}}(i)$ which is outside the subspace $\boldsymbol{W}(n)$. Hence, $\boldsymbol{W}(n)$ is equal to the major PCs $\boldsymbol{U}_B(n)$ up to an orthogonal transformation or rotation, i.e. $span(\boldsymbol{W}(n)) = span(\boldsymbol{U}_B(n))$, and the outer product $\boldsymbol{W}(n)\boldsymbol{W}^T(n)$ is equal to $\boldsymbol{U}_B(n)\boldsymbol{U}_B^T(n)$. In the PAST algorithm, the projection approximation $\bar{\boldsymbol{y}}(i) \approx \boldsymbol{W}^T(i-1)\bar{\boldsymbol{z}}(i)$ is employed so that Eqn. (10) can be relaxed to a quadratic function in $\boldsymbol{W}(n)$. Consequently, conventional recursive least squares (RLS) algorithm can be applied to solve for $\boldsymbol{W}(n)$ with very low arithmetic complexity. In the OPAST algorithm [22], an extra orthonormalization step is added to the PAST algorithm to guarantee the orthonormality of the estimated signal subspace $\boldsymbol{W}(n)$.

**Initialization:**

Initialize $\boldsymbol{\mu}(0)=\mathrm{mean}(\boldsymbol{Z}_s(0))$, where the initial data block is $\boldsymbol{Z}_s(0)=[\boldsymbol{z}_s(1),\boldsymbol{z}_s(2),\boxed{?},\boldsymbol{z}_s(N_0)]$. Obtain $\boldsymbol{U}_B(0)$, $\boldsymbol{\Lambda}_B(0)$ from $\boldsymbol{C}_{zz}(0)=(1/L_d-1)\boldsymbol{Z}_s^{\ T}(0)\boldsymbol{Z}_s(0)$, where $N_0$ is the number of measurements in $\boldsymbol{Z}_s(0)$.

$\boldsymbol{W}(0)=\boldsymbol{U}_B(0)$, $\boldsymbol{\Omega}(0)=\boldsymbol{C}_{yy}^{-1}(0)$, $\boldsymbol{C}_{yy}(0)=\boldsymbol{W}^T(0)\boldsymbol{C}_{zz}(0)\boldsymbol{W}(0)=\boldsymbol{\Phi}(0)\boldsymbol{\Lambda}_B(0)\boldsymbol{\Phi}^T(0)$.

$\beta$, $\beta_\mu$ are the forgetting factors for the OPASTr and the recursive mean estimator respectively. (Typical value 0.99)

**Recursion:**

*For* $n=1,2\boxed{?},N$ *do*

$\boldsymbol{\mu}(n)=\beta_\mu\boldsymbol{\mu}(n-1)+(1-\beta_\mu)\boldsymbol{z}_s(n))$,

$\bar{\boldsymbol{z}}(n)=\boldsymbol{z}_s(n)-\boldsymbol{\mu}(n)$,

$\bar{\boldsymbol{y}}(n)=\boldsymbol{W}^T(n-1)\bar{\boldsymbol{z}}(n)$,

$\boldsymbol{g}(n)=(1/\beta)\boldsymbol{\Omega}(n-1)\bar{\boldsymbol{y}}(n)$,

$\theta(n)=(1/1+\bar{\boldsymbol{y}}^H(n)\boldsymbol{g}(n)),$

$\boldsymbol{p}(n)=\gamma(n)\big(\bar{\boldsymbol{z}}(n)-\boldsymbol{W}(n-1)\bar{\boldsymbol{y}}(n)\big)$,

**Orthonormalization Step for the estimated subspace**

$$r(n)=\frac{1}{\|\boldsymbol{g}(n)\|_2^2}\Big(\frac{1}{\sqrt{1+\|\boldsymbol{p}(n)\|_2^2\|\boldsymbol{g}(n)\|_2^2}}-1\Big),$$

$$\boldsymbol{p}'(n)=\tau(n)\boldsymbol{W}(n-1)\boldsymbol{g}(n)+(1+r(n)\|\boldsymbol{g}(n)\|_2^2)\boldsymbol{p}(n),$$

**Update:**

$\boldsymbol{W}(n)=\boldsymbol{W}(n-1)+\boldsymbol{p}'(n)\boldsymbol{g}^T(n),$

$\boldsymbol{\Xi}(n)=(1/\beta)\boldsymbol{\Xi}(n-1)-\theta(n)\boldsymbol{g}(n)\boldsymbol{g}^T(n).$

**Subspace eigenvector and eigenvalue computation**

$\boldsymbol{y}(n)=\boldsymbol{W}^T(n)\bar{\boldsymbol{z}}(n)$,

Update $\boldsymbol{\Phi}(n)$ and $\boldsymbol{\Lambda}_B(n)$ using the rank-1 modification in Table II.

$[\boldsymbol{\Phi}(n),\boldsymbol{\Lambda}_B(n)]$ = rank-1-modification $(\boldsymbol{\Phi}(n-1),\boldsymbol{\Lambda}_B(n-1),\boldsymbol{y}(n))$.

Compute $\boldsymbol{U}_B(n)=\boldsymbol{W}(n)\boldsymbol{\Phi}(n)$ using (16).

**Initialization:**

Initialize $\boldsymbol{\mu}(0) = \mathrm{mean}(\boldsymbol{Z}_s(0))$, where the initial data block is $\boldsymbol{Z}_s(0) = [\boldsymbol{z}_s(1), \boldsymbol{z}_s(2), \boxed{?}, \boldsymbol{z}_s(N_0)]$.

Obtain $\boldsymbol{U}_B(0)$, $\boldsymbol{\Lambda}_B(0)$ from $\boldsymbol{C}_{zz}(0) = (1/L_d - 1)\boldsymbol{Z}_s^{\ T}(0)\boldsymbol{Z}_s(0)$, where $N_0$ is the number of measurements in $\boldsymbol{Z}_s(0)$.

$\boldsymbol{W}(0) = \boldsymbol{U}_B(0)$, $\boldsymbol{\Omega}(0) = \boldsymbol{C}_{yy}^{-1}(0)$, $\boldsymbol{C}_{yy}(0) = \boldsymbol{W}^T(0)\boldsymbol{C}_{zz}(0)\boldsymbol{W}(0) = \boldsymbol{\Phi}(0)\boldsymbol{\Lambda}_B(0)\boldsymbol{\Phi}^T(0)$.

$\beta$, $\beta_\mu$ are the forgetting factors for the OPASTr and the recursive mean estimator respectively. (Typical value 0.99)

**Recursion:**

```
For $n = 1,2\boxed{?}, N$ do
    $\boldsymbol{\mu}(n) = \beta_\mu \boldsymbol{\mu}(n-1) + (1-\beta_\mu)\boldsymbol{z}_s(n))$,
$\bar{\boldsymbol{z}}(n) = \boldsymbol{z}_s(n) - \boldsymbol{\mu}(n)$,
$\bar{\boldsymbol{y}}(n) = \boldsymbol{W}^T(n-1)\bar{\boldsymbol{z}}(n)$,
$\boldsymbol{g}(n) = (1/\beta)\boldsymbol{\Omega}(n-1)\bar{\boldsymbol{y}}(n)$,
    $\theta(n) = (1/1 + \bar{\boldsymbol{y}}^H(n)\boldsymbol{g}(n)),$
    $\boldsymbol{p}(n) = \gamma(n)\left(\bar{\boldsymbol{z}}(n) - \boldsymbol{W}(n-1)\bar{\boldsymbol{y}}(n)\right)$,
```

**Orthonormalization Step for the estimated subspace**

$$r(n) = \frac{1}{\|\boldsymbol{g}(n)\|_2^2}\left(\frac{1}{\sqrt{1 + \|\boldsymbol{p}(n)\|_2^2\|\boldsymbol{g}(n)\|_2^2}} - 1\right),$$

$$\boldsymbol{p}'(n) = \tau(n)\boldsymbol{W}(n-1)\boldsymbol{g}(n) + (1 + r(n)\|\boldsymbol{g}(n)\|_2^2)\boldsymbol{p}(n),$$

**Update:**

$$\boldsymbol{W}(n) = \boldsymbol{W}(n-1) + \boldsymbol{p}'(n)\boldsymbol{g}^T(n),$$

$$\boldsymbol{\Xi}(n) = (1/\beta)\boldsymbol{\Xi}(n-1) - \theta(n)\boldsymbol{g}(n)\boldsymbol{g}^T(n).$$

**Subspace eigenvector and eigenvalue computation**

$$\boldsymbol{y}(n) = \boldsymbol{W}^T(n)\bar{\boldsymbol{z}}(n),$$

Update $\boldsymbol{\Phi}(n)$ and $\boldsymbol{\Lambda}_B(n)$ using the rank-1 modification in Table II.

$[\boldsymbol{\Phi}(n), \boldsymbol{\Lambda}_B(n)]$ = rank-1-modification $(\boldsymbol{\Phi}(n-1), \boldsymbol{\Lambda}_B(n-1), \boldsymbol{y}(n))$.

Compute $\boldsymbol{U}_B(n) = \boldsymbol{W}(n)\boldsymbol{\Phi}(n)$ using (16).

1.

**Table 1 The OPASTr Algorithm**

To apply the OPAST algorithm, an initial ED is assumed to be available either by performing an ED on an initial data block or pre-determining the eigenvalues offline. The eigenvalues so obtained can be used with the Minimum Description Length (MDL) criterion [24] to estimate the dimension B of the signal subspace. During online application, the OPAST algorithm is invoked to update the signal subspace recursively. Interested readers can refer to [24] - [25] of the PAST and the OPAST algorithms for more details. However, in computing the PCs, the PAST and OPAST algorithms are not directly applicable due to the arbitrary orthogonal rotation mentioned above. In the OPASTr algorithm, the PCs are further extracted from the signal subspace tracked [26]. Given the signal subspace $\boldsymbol{W}(n)$, the covariance matrix $\boldsymbol{C}_{zz}(n) = \boldsymbol{U}(n)\boldsymbol{\Lambda}(n)\boldsymbol{U}^{T}(n)$ is projected onto the signal subspace $\boldsymbol{W}(n)$ to obtain

$$\begin{aligned}\boldsymbol{C}_{yy}(n) &= \boldsymbol{W}^{T}(n)\boldsymbol{U}(n)\boldsymbol{\Lambda}(n)\boldsymbol{U}^{T}(n)\boldsymbol{W}(n)\\ &= \boldsymbol{W}^{T}(n)\boldsymbol{U}_{B}(n)\boldsymbol{\Lambda}_{B}(n)\boldsymbol{U}_{B}^{T}(n)\boldsymbol{W}(n) = \boldsymbol{\Phi}(n)\boldsymbol{\Lambda}_{B}(n)\boldsymbol{\Phi}^{T}(n),\end{aligned} \tag{14}$$

where $\boldsymbol{\Phi}(n)$ is a $B\times B$ orthogonal transformation satisfying $\boldsymbol{\Phi}(n)\boldsymbol{\Phi}^{T}(n) = \boldsymbol{I}$ and

$$\boldsymbol{U}_{B}(n) = \boldsymbol{W}(n)\boldsymbol{\Phi}(n), \tag{15}$$

The covariance matrix $\boldsymbol{C}_{yy}(n) = E[\boldsymbol{y}(n)\boldsymbol{y}^{T}(n))]$ can be recursively updated as

$$\boldsymbol{C}_{yy}(n) = \beta\boldsymbol{C}_{yy}(n-1) + (1-\beta)\boldsymbol{y}(n)\boldsymbol{y}^{T}(n), \tag{16}$$

Where $y(n) = W^{T}(n)\bar{z}(n)$. Here, we remark that $y(n)$ is a projection of $\bar{z}(n)$ on the subspace $W(t)$ and it is different from the projection approximation $\bar{y}(n)$ in (14). $\Phi(n)$ can be recursively computed using the ED of $C_{yy}(n)$.Firstly, let the ED of $C_{yy}(n-1)$ be $\Phi(n-1)\Lambda_{B}(n-1)\Phi^{T}(n-1)$. The expression in (16) can be rewritten as one rank-1 modification given by

$$\boldsymbol{C}_{yy}(n) = \boldsymbol{\Phi}(n-1)[\beta\boldsymbol{\Lambda}_{B}(n-1) + (1-\beta)\boldsymbol{s}(n)\boldsymbol{s}^{T}(n)]\boldsymbol{\Phi}(n-1)^{T}, \tag{17}$$

Where $s(n) = \Phi^{T}(n-1)y(n)$.Let the corresponding ED be

$$\beta\boldsymbol{\Lambda}_{B}(n-1) + (1-\beta)\boldsymbol{s}(n)\boldsymbol{s}^{T}(n) = \boldsymbol{\Phi}(n)\boldsymbol{\Lambda}_{B}(n)\boldsymbol{\Phi}^{T}(n). \tag{18}$$

The ED of the rank-1 update in (18) can be recursively computed using rank-1 modification [9],

[22], which is summarized in Table II. Finally, the eigenvectors of $C_{yy}(n)$ are given by

$$\boldsymbol{\Phi}(n) = \boldsymbol{\Phi}(n-1)\tilde{\boldsymbol{\Phi}}(n), \tag{19}$$

Consequently, the new PCs can be computed according to (15). Then, the PC scores $\boldsymbol{\tau}(n) = [t_1(n), t_2(n), \boxed{?} t_B(n)]^T$ can be computed recursively as

$$\boldsymbol{\tau}(n) = \boldsymbol{U}_B^T(t)\bar{\boldsymbol{z}}_s(n), \tag{20}$$

### 3.4 Kalman Filter for Interval forecasting

After PCs and PC scores are recursively updated, separate time series models such as the AR model can be built for each PC score to perform prediction because the PCs are orthogonal to each other [6], [27] Unlike the PCA in [6], we employed the KF to recursively track the time series so that the solar power output and interval forecast can be computed online in a real-time manner. More precisely, for each PC score $t_m(n)$ of the $m-th$ PC obtained in (20), $m = 1,2,\boxed{?}B$ an AR model of $L-th$ order will be constructed. In contrast to the conventional LS formulation in (6), we incorporate a regularization term $\|\boldsymbol{\alpha}_m(i) - \boldsymbol{\alpha}_m(i-1)\|_2^2$ as follows:

$$\begin{aligned} \min\ \{&\textstyle\sum_{i=1}^{n} \| R_m^{-1/2}(i)(t_m(i) - \Sigma_{j=1}^{L}\alpha_j(i)t_m(i-j))\|_2^2 \\ &+ \textstyle\sum_{i=1}^{n} \| \boldsymbol{Q}_m^{-1/2}(i)(\boldsymbol{\alpha}_m(i) - \boldsymbol{\alpha}_m(i-1))\|_2^2\}, \end{aligned} \tag{21}$$

Where $\boldsymbol{\alpha}_m(i) = [\alpha_{m,1}(i), \alpha_{m,2}(i), \boxed{?}, \alpha_{m,L}(i)]^T$ are the AR coefficients, $R_m(i)$ and $\boldsymbol{Q}_m(i)$ are the co-variances of the loss function $e_m(i) = t_m(i) - \Sigma_{j=1}^{L}\alpha_j(i)t_m(i-j)$ and $\boldsymbol{\varepsilon}_m(i) = \boldsymbol{\alpha}_m(i) - \ \boldsymbol{\alpha}_m(i-1)$ is the regularization term, respectively. The inverses $R_m^{-1/2}(i)$ and $\boldsymbol{Q}_m^{-1/2}(i)$ are used to perform scaling on each variable (whitening) in order to achieve equal variance of the transformed variables. The regularization term requires the estimate to stay close to the previous estimate and hence the variance of the estimator will be reduced. It is shown in [28] that Eqn. (21) can be formulated as the following state space model (SSM):

$$\boldsymbol{\alpha}_m(n) = \boldsymbol{\alpha}_m(n-1) + \boldsymbol{\varepsilon}_m(n), \tag{22a}$$

$$t_m(n) = \boldsymbol{h}_m(n)^T\boldsymbol{\alpha}_m(n) + e_m(n). \tag{22b}$$

Eqn. (22a) is the state equation and it describes the evolution of the AR coefficients over time, as a function of the previous AR coefficients $\boldsymbol{\alpha}_m(n)$ and $\varepsilon_m(n)$ represents the modeling error. Eqn. (22b) is the measurement equation which models the current PC scores $t_m(n)$ with previous scores $\boldsymbol{h}_m(n) = [t_m(n-1), t_m(n-2), \ldots, t_m(n-L)]^T$ , and the AR coefficients $\boldsymbol{\alpha}_m(n)$ represent the weighting of each previous score $t_m(n-i)$ , $i = 1,2,\boxed{?}L$ , and $e_m(n)$ is the measurement noise. We can see that the state equation in (22a) and the measurement equations in (22b) are equivalent to the regularization term and the loss function in (21) respectively. The SSM in (22a) and (22b) can be recursively tracked using the KF as follows:

Predict

$$\begin{aligned} &\boldsymbol{\alpha}_m(n \mid n-1) = \boldsymbol{\alpha}_m(n-1 \mid n-1), \\ &\boldsymbol{\Omega}_m(n \mid n-1) = \boldsymbol{\Omega}_m(n-1 \mid n-1) + \boldsymbol{Q}_m(n), \end{aligned} \tag{23a}$$

Update

$$\begin{aligned} &e_m(n) = t_m(n) - \boldsymbol{h}_m^T(n)\boldsymbol{\alpha}_m(n-1 \mid n-1), \\ &S_m(n) = \boldsymbol{h}_m^T(n)\boldsymbol{\Omega}_m(n \mid n-1)\boldsymbol{h}_m(n) + R_m(n), \\ &\boldsymbol{K}_m(n) = \boldsymbol{\Omega}_m(n \mid n-1)\boldsymbol{h}_m(n)S_m(n)^{-1}, \\ &\boldsymbol{\alpha}_m(n \mid n) = \boldsymbol{\alpha}_m(n \mid n-1) + \boldsymbol{K}_m(n)e_m(n), \\ &\boldsymbol{\Omega}_m(n \mid n) = \boldsymbol{\Omega}_m(n \mid n-1) - \boldsymbol{K}_m(n)\boldsymbol{h}_m^T(n)\boldsymbol{\Omega}_m(n \mid n-1), \end{aligned} \tag{23b}$$

where $\boldsymbol{\Omega}_m(n)$ is the covariance estimate of the AR coefficients. A major difficulty in practical implementation of the KF is that the covariances $\boldsymbol{Q}_m(n)$ and $R_m(n)$ are often unknown. In [19], the following recursive covariance estimators for estimating these covariances from the state error $\boldsymbol{\varepsilon}_m(n)$ and measurement error $e_m^2(n)$ for the KF were proposed:

$$\boldsymbol{Q}_m(n) = \beta_{Q\_m}\boldsymbol{Q}_m(n-1) + (1-\beta_{Q\_m})\boldsymbol{\varepsilon}_m(n)\boldsymbol{\varepsilon}_m^T(n), \tag{24a}$$

$$R_m(n) = \beta_{R\_m}R_m(n-1) + (1-\beta_{R\_m})e_m^2(n), \tag{24b}$$

where $\beta_{Q\_m}$ and $\beta_{R\_m}$ are forgetting factors for the recursive estimators of $\boldsymbol{Q}_m(n)$ and $R_m(n)$ respectively. $\boldsymbol{\varepsilon}_m(n)$ and $e_m(n)$ are the modeling error and observation noise as defined in (23a) and (23b) respectively. Hence, the one-step ahead prediction $\hat{t}_m(n+1)$ is given by

$$\hat{t}_m(n+1) = \boldsymbol{h}_m^T(n+1)\boldsymbol{\alpha}_m(n+1 \mid n), \tag{25a}$$

$$\operatorname{var}(\hat{t}_m(n+1)) = \boldsymbol{h}_m^T(n+1)\boldsymbol{\Omega}_m(n+1 \mid n)\boldsymbol{h}_m(n+1), \tag{25b}$$

From (25a) and (25b) and Appendix A, the one day ahead prediction of the solar radiation vector $\hat{\boldsymbol{z}}(n+1)$ can be determined as follows

**Point Forecast**

$$\hat{\boldsymbol{z}}(n+1) = \boldsymbol{\mu}(n) + \Sigma_{m=1}^{B}\hat{t}_m(n+1)\boldsymbol{u}_m(n), \tag{26a}$$

**Interval Forecast**

$$\operatorname{var}(\hat{\boldsymbol{z}}(n+1)) = \Sigma_{m=1}^{B}\operatorname{var}(\hat{t}_m(n+1))\boldsymbol{u}_m(n)\boldsymbol{u}_m(n)^T, \tag{26b}$$

The point forecast $\hat{\boldsymbol{z}}(n+1)$ is derived similarly as in (9) except that the mean $\boldsymbol{\mu}(n)$ is obtained by the recursive mean estimator in (12) To perform two days ahead prediction, one can append the prediction $\hat{t}_m(n+1)$ into $\boldsymbol{h}_m(n+2) = [\hat{t}_m(n+1), t_m(n), t_m(n-1), \boxed{?}, t_m(n-L+2)]^T$ , and then apply the KF again to predict $\hat{t}_m(n+2)$ and $\operatorname{var}(\hat{t}_m(n+2))$ using Eqn.(26a) and (26b) respectively. This procedure can be repeated for $h$ times to compute a $h$ days ahead prediction.

# Chapter 4 Application to solar radiation prediction

In this chapter, we will apply the RDFA method introduced in Chapter 3 to the solar radiation prediction problem. For comparison purpose, we will consider six conventional methods studied in [5], namely Regression, UCM, ARIMA, transfer function, neural network and Hybrid in paper [5]. Better predicting error is observed and it is summarized in Table 3.

## 4.1 Datasets

In this simulation, we will consider six different datasets obtained from the locations of North American (See the summary in Table 2). The first three datasets consist of hourly data from the National Solar Radiation Database, which is observed by the National Renewable Energy Laboratory in Golden, Colorado: www.nrel.gov/rredc/solar_data.html. These series run from January 1, 1987 to December 31, 1990, comprising 35,040 observations (8760 per year). The locations used were Kansas City, MO, Denver, CO, and Phoenix, AZ. The measure of irradiance is the global horizontal component, in W/m2. On the other hand, the other three datasets are from the Measurements and Instrumentation Data Center baseline measurement system database (http://www.nrel.gov/midc/srrl_bms), and are at a basic resolution of 1 minute. The data from Clark power station in Nevada, and the Solar Radiation Research Laboratory (SRRL), run from January 1, 2007 to February 11, 2008, providing 571,677 observations. The data from the National Wind Technology Laboratory site runs until October 17, 2007, providing 417,434 observations.

**Table 2 The data**

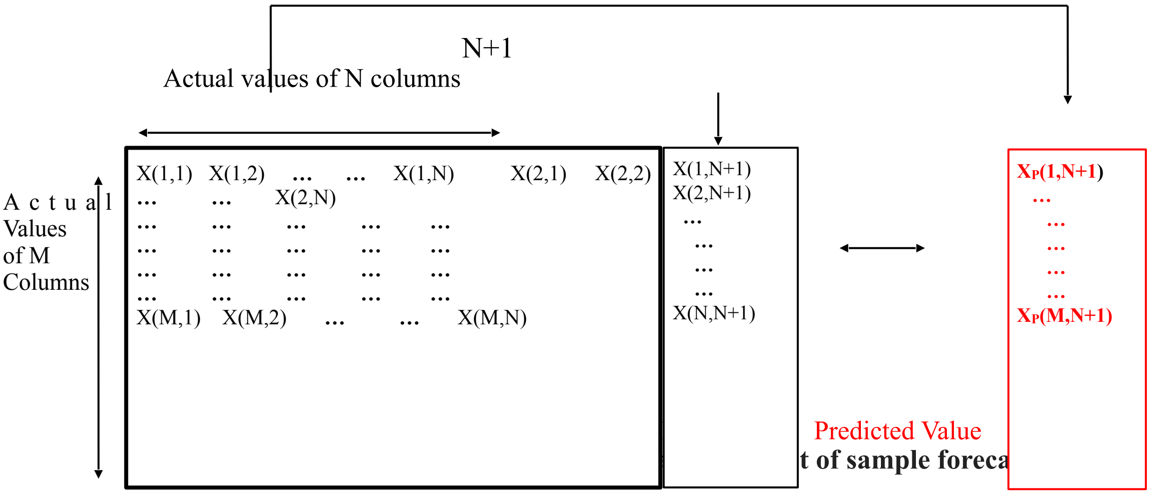
| Database and available series | Time span | Observations |
|---|---|---|
| Kansas City, Missouri | January1, 1987 to December 31, 1990 | 35,040 |
| Denver, Colorado | January 1, 1987 to December 31, 1990 | 35,040 |
| Phoenix, Arizona | January 1, 1987 to December 31, 1990 | 35,040 |
| Clark, Nevada Power Station | January 1, 2007 to February 11, 2008 | 571,677 |
| National Wind Technology Laboratory | January 1, 2007 to October 16, 2007 | 417,434 |
| SRRL | January 1, 2007 to February 11, 2008 | 571,677 |

For fair comparison, we adopt the same data pre-processing approach proposed in [5], which is summarized as follows: (a) For Kansas, Denver and Phoenix datasets, we only focus on daylight hours. Assuming the total number of daylight hours is 12, we set the window width as 480 hours, which is equivalent to 40 days. (b) For the Clark, National Wind and SSRL datasets, we reproduce a hourly data by averaging 60 minutes data within one hour. After that, we partition these hourly data into a block of 480 hours data, which is similar to the data pre-processing approach applied to Kansas, Denver and Phoenix datasets.

### 4.2 Out Of Sample Forecasts

Out of sample is a typical approach used to evaluate the performance of forecasting algorithms. In brief, it means that some testing samples are drawn out from the whole dataset and the remaining samples of the dataset are used to train the forecasting algorithm, i.e. building the predictor. The prediction error is then determined by computing the difference between the predictor and the actual value [5].

Fig.3 below shows the procedure of out of sample forecasts: first, we choose the array with $M$ rows and $N$ columns. Then the array is trained in PCA algorithm and the predicted Column $N+1$ is recorded and is compared with the actual Column $N+1$. After that, the new array is updated with the Column N+1 in and Column 1 out of the queen. The same procedure is repeated for $K$ times.



For fair comparison with [5], we followed the setting in [5] to choose the width of sliding widow as 40. The sliding window adds the latest sample and removes the most-dated sample from the batch for each out of sample loop.

### 4.3 Comparison with Existing Model

We compare the performance of the RDFA and the other six methods reported in [5]. The results for the other six methods are quoted directly from [5].

We employ the commonly used Mean Absolute Percentage Error (MAPE) to evaluate the performance of the algorithms

$$MAPE = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{x^{(k)}(n) - \hat{x}^{(k)}(n)}{x^{(k)}(n)}\right|, \tag{27}$$

Generally, smaller MAPE reflects better accuracy of the algorithms. Table 3 shows the forecast error of the solar intensities for the hourly data. In the three National Solar Radiation Database series, at the 1-h horizon, the MAPE of the RDFA range from as low as 21.49% in Denver, to as high as 39.56% in SRRL.

**Table 3 Comparison of the forecast errors, hourly data**

| **Model** | Forecast horizon | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1h | | 2h | | 3h | | 4h | |
| | Error | Ratio | Error | Ratio | Error | Ratio | Error | Ratio |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Kansas city** | | | | | | | | |
| RDFA | 0.2875 | | 0.2559 | | 0.2497 | | 0.2367 | |
| Regression | 0.3906 | 1.358 | 0.4127 | 1.612 | 0.4138 | 1.657 | 0.4146 | 1.752 |
| UCM | 0.3911 | 1.0013 | 0.4125 | 0.9995 | 0.4142 | 1.0010 | 0.4149 | 1.0007 |
| ARIMA | 0.2642 | 0.6764 | 0.3301 | 0.7999 | 0.3596 | 0.8690 | 0.3823 | 0.9221 |
| Transfer function | 0.2653 | 0.6792 | 0.3324 | 0.8054 | 0.3672 | 0.8874 | 0.3881 | 0.9361 |
| Neural network | 0.3615 | 0.9255 | 0.4015 | 0.9729 | 0.4096 | 0.9899 | 0.4125 | 0.9949 |
| Hybrid | 0.2642 | 0.6764 | 0.3315 | 0.8032 | 0.3602 | 0.8705 | 0.3829 | 0.9235 |
| **Denver** | | | | | | | | |
| RDFA | 0.1849 | | 0.2929 | | 0.2755 | | 0.3085 | |
| Regression | 0.4118 | 1.018 | 0.4362 | 2.2615 | 0.4388 | 2.5002 | 0.4395 | 1.4230 |
| UCM | 0.4113 | 0.9988 | 0.4359 | 0.9993 | 0.4385 | 0.9993 | 0.4391 | 0.9991 |
| ARIMA | 0.2892 | 0.7023 | 0.3467 | 0.7948 | 0.3804 | 0.8669 | 0.3982 | 0.9060 |
| Transfer function | 0.2925 | 0.7103 | 0.3521 | 0.8072 | 0.3208 | 0.7311 | 0.4084 | 0.9292 |
| Neural network | 0.3839 | 0.9322 | 0.4056 | 0.9298 | 0.3031 | 0.6907 | 0.4115 | 0.9363 |
| Hybrid | 0.2901 | 0.7045 | 0.3486 | 0.7992 | 0.3815 | 0.8694 | 0.3997 | 0.9094 |
| **Phoenix** | | | | | | | | |
| RDFA | 0.2449 | | 0.2213 | | 0.2410 | | 0.2807 | |
| Regression | 0.2996 | 0.8329 | 0.3115 | 2.0794 | 0.3127 | 3.8134 | 0.3133 | 4.027 |
| UCM | 0.2992 | 0.9987 | 0.3114 | 0.9997 | 0.3129 | 1.0006 | 0.3138 | 1.0016 |
| ARIMA | 0.2360 | 0.7877 | 0.2711 | 0.8703 | 0.2857 | 0.9137 | 0.2944 | 0.9397 |
| Transfer function | 0.2352 | 0.7850 | 0.2721 | 0.8735 | 0.2862 | 0.9153 | 0.2951 | 0.9419 |
| Neural network | 0.2938 | 0.9806 | 0.3046 | 0.9778 | 0.3078 | 0.9843 | 0.3095 | 0.9879 |
| Hybrid | 0.2367 | 0.7901 | 0.2721 | 0.8735 | 0.2865 | 0.9162 | 0.2952 | 0.9422 |
| **National wind** | | | | | | | | |
| RDFA | 0.3087 | | 0.3023 | | 0.2967 | | 0.2906 | |
| Regression | 0.9051 | 2.9316 | 1.0720 | 3.5451 | 1.0930 | 3.6827 | 1.1020 | 3.7924 |
| UCM | 0.7541 | 0.8332 | 0.9665 | 0.9016 | 0.9766 | 0.8935 | 1.0440 | 0.9474 |
| ARIMA | 0.6447 | 0.7123 | 0.8968 | 0.8366 | 0.9554 | 0.8741 | 1.0080 | 0.9147 |
| Transfer function | 0.6498 | 0.7179 | 0.8979 | 0.8376 | 0.9586 | 0.8770 | 1.0150 | 0.9211 |
| Neural network | 0.6948 | 0.7676 | 0.9435 | 0.8801 | 0.9671 | 0.8848 | 1.0437 | 0.9471 |
| Hybrid | 0.6458 | 0.7135 | 0.8977 | 0.8374 | 0.9566 | 0.8752 | 1.0088 | 0.9154 |
| **Clark, Nevada** | | | | | | | | |
| RDFA | 0.2831 | | 0.265 | 0.289 | 0.2942 | | | |
| Regression | 0.3185 | 0.9818 | 0.383 | 1.6396 | 0.336 | 4.54 | 0.3402 | 7.749 |
| UCM | 0.2421 | 0.7601 | 0.2845 | 0.8410 | 0.2941 | 0.8660 | 0.3170 | 0.9318 |
| ARIMA | 0.1968 | 0.6179 | 0.2569 | 0.7594 | 0.2873 | 0.8460 | 0.3096 | 0.9101 |
| Transfer function | 0.2003 | 0.6289 | 0.2572 | 0.7603 | 0.2876 | 0.8469 | 0.3102 | 0.9118 |
| Neural network | 0.2219 | 0.6967 | 0.2599 | 0.7683 | 0.2988 | 0.8799 | 0.3196 | 0.9394 |
| Hybrid | 0.1987 | 0.6239 | 0.2578 | 0.7620 | 0.2889 | 0.8507 | 0.3102 | 0.9118 |
| **SRRL** | | | | | | | | |

From Table 3, we can find that the RDFA method is generally better than the other six algorithms, except on the data of Clark for the 1-4 hours ahead forecast. However, we remark that the main advantage of RDFA is that it can predict the solar intensity for the whole day, i.e. 1-24 hours ahead, and the prediction error is similarly small as those shown in Table 3. Unlike the other six algorithms, which generally gives larger prediction error when the forecasting period increases, we can see that the RDFA algorithm generally gives consistent forecasting error for 1-4 hours ahead forecast. The better performance of the RDFA is partly contributed by the consideration of the daily data structure. Due to the variation of weather, the solar intensity is usually subjected to fluctuation In this regard; the interval forecast plays an important role on the solar radiation forecast. With the rough range of the solar radiation in the next day, the shortage of fossil energy can be adjusted and supplied, which will make full use of the solar energy and preserve other non-renewable energy. As an illustration, we employed the data from July 1st to October 30th, 1988 in Kansas City.

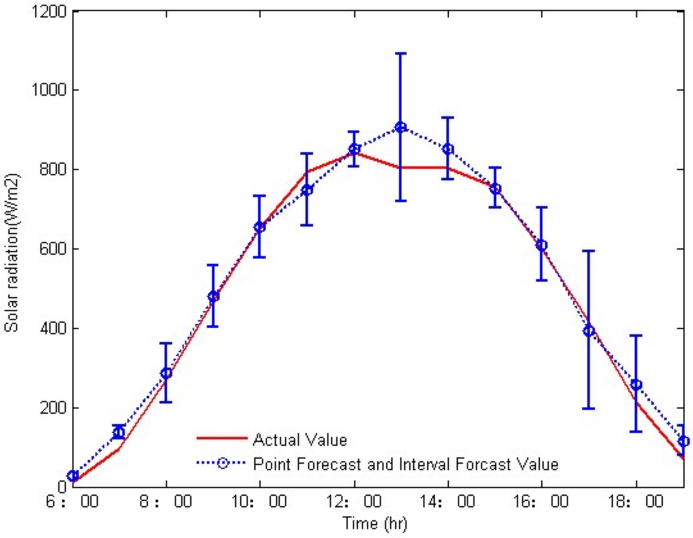


**Fig.4 Point Forecast and Interval Forecast using the Proposed RDFA Algorithm**

Fig. 4 shows the point forecast and interval forecast result employed the solar irradiation in Aug 9, 1988. As an illustration, the predicted solar radiations for Aug 10 and Aug 11 are shown in Fig. 5, and most of the predicted solar intensities lie within the predicted intervals obtained by the proposed method. In general, similar results can be obtained from the remaining data but we have omitted here for simplicity.

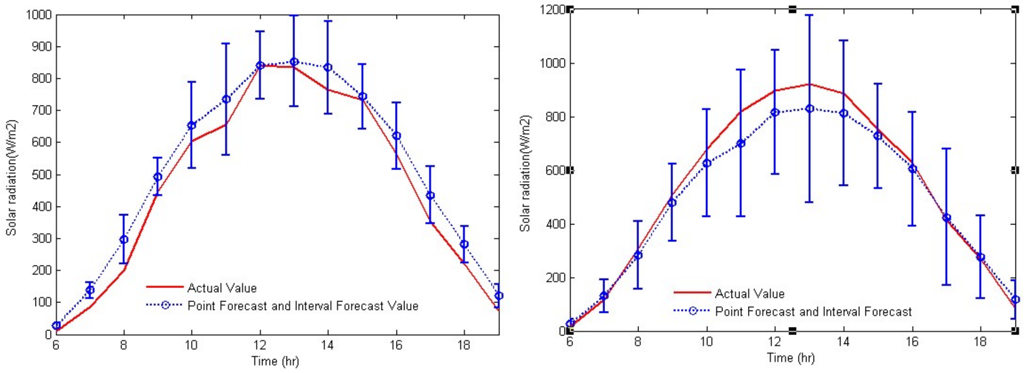


**Fig. 5 Point Forecast and Interval Forecast Result**

# Chapter 5 Recursive Dynamic Factor Analysis with Multi-Clustering and Model Averaging

As mentioned before, the solar radiation varies due to a number of factors, such as relative humility and cloud cover. In general, the solar radiation intensity during sunny and rainy weather could behave distinctively. It may be beneficial to adopt different models to capture the trend of the solar radiation intensities under different weather conditions. In this regard, we further propose a multi-clustering and model averaging extension for the RDFA algorithm. Clustering is a technique that divides all the values into different groups such that similar samples are grouped into the same group and different samples to the different groups. One of the most popular and effective clustering algorithm is the k-mean. In the proposed multi-clustering approach, we employ the K-mean algorithm to divide the discrete samples of the solar intensities into different clusters accordingly. Then, we invoke the RDFA algorithm to build model for each group of data. Afterwards, model averaging is performed on these models to predict solar intensity in future. It will be demonstrated in later sections that the new multi-clustering and model averaging extension will further improve the accuracy on solar radiation forecasting.

## 5.1 Datasets

The data we used in this part is the hourly observations in Phoenix City from January 1, 1988 to August 31, 1990, which is obtained from the National Renewable Energy Laboratory in Golden, Colorado: www.nrel.gov/rredc/solar_data.html.

## 5.2. K-means

K-means is a commonly used clustering algorithm [29].Its main idea is to first randomly choose $k$ points as the centroids of the $k$ cluster. The next step is to assign each of the data sample to the nearest centroid. When no point is pending, the first step is completed and 1st loop is done. Then, the centriods of each group are computed and the resultant centriods will replace the previous ones. The same procedure is repeated until the change of the centriod locations is smaller than a certain threshold.

More specifically, the following objective function is minimized

(28)

$$J = \sum_{j=1}^{N}\sum_{i=1}^{M}\left\|x_i^{(j)} - c_j\right\|^2$$

where $\left\|x_i^{(j)} - c_j\right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre $c_j$, is an indicator of the distance of the $M$ data points from their respective cluster centers.

The steps of k-means algorithm are shown below:

Step 1.Choose a number of clusters.

Step 2.Assign randomly to each point coefficients for being in the clusters.

Step 3.Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than $\varepsilon$ , the given sensitivity threshold).

Step 4.Compute the centroid for each cluster, using the formula above.

Step 5.For each point; compute its coefficients of being in the clusters, using the formula above.

## 5.2.4 Application of K-means clustering

In this essay, we use the K-mean clustering method to find 3 as the optimal number of classification. Then, we use SPSS software [30] to divide the hourly observations in Phoenix City from January 1, 1988 to August 31, 1990. According to the method reported in [5] the optimal clustering number we obtained is 3, which can be supposed as sunny, cloudy and rainy. In other data from different places, the weather condition may be more changeable, the cluster categories can be more diverse.

## 5.3 The proposed multi-cluster analysis

Given an initialization data batch of size $N_{init}^{(k)}$, i.e. $\boldsymbol{Z}^{(k)} = [\boldsymbol{z}(1), \boldsymbol{z}(2), ....]^T$ for each model (note: Due to the clustering, each model might have different init data size), $\boldsymbol{U}^{(k)}(t)$, $\boldsymbol{\Lambda}_B^{(k)}(n)$ and the forecasts $\hat{\boldsymbol{Z}}^{(k)} = [\hat{\boldsymbol{z}}^{(k)}(1), \hat{\boldsymbol{z}}^{(k)}(2), ...]^T$ can be obtained by invoking the OPASTr algorithm on $\boldsymbol{Z}^{(k)}$.

The proposed approach can be mainly divided into four steps:

1. The existing data block is classified into different groups such as k clusters based on k-means clustering algorithm.
2. The data of $k$ groups are put into $k$ sub-blocks, in each block, we can obtain $k$ predicted values $\hat{z}_k(n)$, and the average error $E[e_k(\mathrm{n})] = E[\mathrm{z(n)} - \hat{\mathrm{z}}_\mathrm{k}(\mathrm{n})]$ is calculated by repeating the last step.
3. Then，chi-square distribution is used to predict the likelihood of each model may occur.

$$\hat{g}_k(n+1) = \chi_{P-B}^2(E(\mathbf{e}_k(n)^T \textstyle\sum_{(k)}^{-1}(n)\mathbf{e}_k(n))) \tag{29}$$

where $\boldsymbol{e}_k(n) = \boldsymbol{z}(n) - \hat{\boldsymbol{z}}_k(n)$ is the prediction error, $\tilde{\boldsymbol{e}}(n) = \boldsymbol{\Sigma}_{(k)}^{-1/2}\tilde{\boldsymbol{e}}(n)$ is the standardized error, $\chi_{P-B}^2(.)$ is the chi-square distribution with $P-B$ degrees of freedom and $\boldsymbol{\Sigma}_k^{-1}(n)$ is the inverse covariance matrix of model $k$. $\boldsymbol{\Sigma}_k^{-1}(n)$ can be obtained by maximizing the likelihood $\log(\det(\boldsymbol{C}) - tr(\boldsymbol{C}\sum))$, where $\boldsymbol{C}$ is the estimated inverse covariance matrix. However, computing in such a manner requires batch processing for online implementation. To utilize the use of the FPCA, one may approximate $\boldsymbol{\Sigma}_k^{-1}(n)$ as

$$\boldsymbol{\Sigma}_k^{-1}(n) \approx \boldsymbol{U}_B^{(k)}(n)(\boldsymbol{\Lambda}_B^{(k)}(n))^{-1}\boldsymbol{U}_B^{(k)T}(n) \tag{30}$$

Where $\boldsymbol{U}^{(k)}(t)$ is the principal component (PC) of the k-th model,

$\Lambda_B^{(k)}(n)$ is the eigenvalue of the k-th model, $\Lambda_B^{(k)}(n) = diag(\lambda_1, \lambda_2, ....\lambda_B)$ ,

and $B$ : is the number of selected PCs.

Based on Bayesian model averaging, the expected solar radiation is $E(\hat{z}(n+1)$ where the expectation is taken over the model space. Hence, it is given by

$$E(\hat{z}(n+1)) = \sum_{k=1}^{K} \hat{\omega}_k (n+1)\hat{z}^{(k)}(n+1) \tag{31}$$

where $\hat{z}^{(k)}(n+1)$ is the 1-step ahead forecast of model $k$ , $\hat{\omega}_k(n+1)$ is the predicted probability of the model or weighting and is given by

$$\hat{\omega}_k(n+1) = \frac{\hat{g}_k(n+1)}{\sum_{k=1}^{K} \hat{g}_k(n+1)} \tag{32}$$

One should include the prior of sunny, rainy and cloudy. Such information can be derived from the observatories or past meteorology data. In model selection, after computing the above likelihood $\hat{g}_k(n+1)$ for each model $k$ , one may select the model with the maximum $g_k(n+1)$ and update the corresponding model using $z(n+1)$ . For model averaging, the predicted output is given by (31), but only the best model is updated.

## 5.4 Result Analysis

According to the K-mean clustering results in previous section, we classify the data into three groups for the dataset employed in 5.2. Table 4 presents the prediction error for different groups of samples, and the criterion used to evaluate the error is the MAPE.

**Table 4 The comparison of three models**

| **Daytime** | **Error for Group 1** | **Error for Cluster 2** | **Error for Cluster 3** | **Averaged error** |
|---|---|---|---|---|
| 1h | 0.170838 | 0.290617 | 0.317438 | 0.259631 |
| 2h | 0.090612 | 0.211246 | 0.19579 | 0.165882 |
| 3h | 0.074891 | 0.313934 | 0.18964 | 0.192822 |
| 4h | 0.055477 | 0.199965 | 0.134953 | 0.130132 |
| 5h | 0.061809 | 0.295233 | 0.11562 | 0.157554 |
| 6h | 0.042248 | 0.307228 | 0.098909 | 0.149462 |
| 7h | 0.072621 | 0.334885 | 0.122283 | 0.176597 |
| 8h | 0.073436 | 0.257961 | 0.191541 | 0.174313 |
| 9h | 0.108692 | 0.440563 | 0.285737 | 0.278331 |

| | | | | |
|---|---|---|---|---|
| 10h | 0.151925 | 0.474422 | 0.566723 | 0.39769 |
| 11h | 0.280644 | 0.39724 | 0.33124 | 0.323269 |

Here, the major difficulty of this problem is that we do not know the exact category that the next day belonged to. Since there are three models, we will get three different predictors; In Fig.6, the actual value is May 3, 1988 in Phoenix City, and the predicted value of Model 1, 2 and 3 are the previous 40 days of May 3 in each model dataset. In general, if the model chosen fits the data, the predicted error should be small, e.g. the green line of Model 3 is the most similar to the actual value in Fig. 6. Otherwise, the error rate will be large, such as the blue line and red line of Model 1 and Model 2 in the Fig. 6. In order to compute the forecast from the three models, we can either select one of the models or averaging the forecasts obtained by the three models, and they are referred to model selection and averaging respectively.

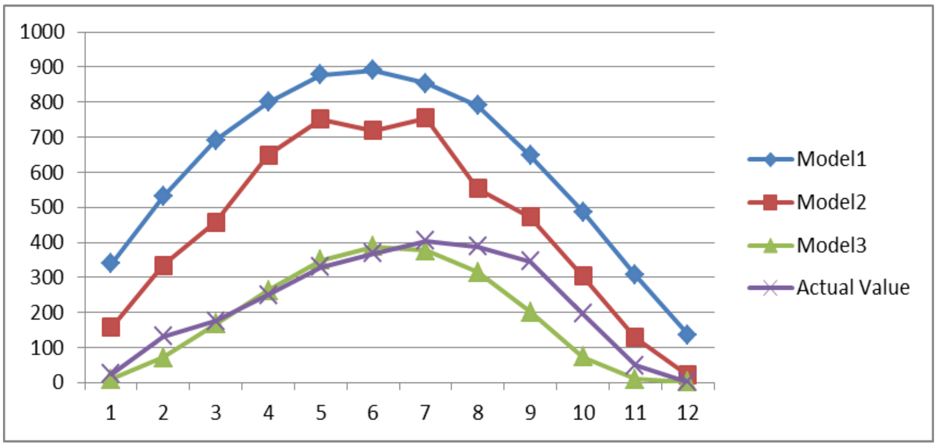


**Fig. 6 The comparison of three models. The X-axis is the daytime hours and Y-axis is the solar radiation.**

## 5.5 Bayesian model selection and Model Averaging

As mentioned above, there are generally two methods to choose the models, i.e. model averaging and Model selection. Model selection is the task of choosing a model from a set of potential models with the best inductive bias, which in practice means selecting parameters in an attempt to create a model of optimal complexity given (finite) training data. In the context of solar intensity prediction, the model that gives the best prediction accuracy is chosen [31]. Bayesian model averaging is a method to make formal and comprehensive inference from multiple models and it can avoid the choice of a particular model [32].

### 5.5.1 Bayesian model selection

Using model selection, we choose predicted values in the model with the largest probability as the final results. The Fig. 7 presents the mean result of model selected forecasting value and actual value in 50 days from January 1, 1988 to July 30, 1988 from Phoenix City. And in Fig. 8, the single day in June 20 from Phoenix City is shown as well.

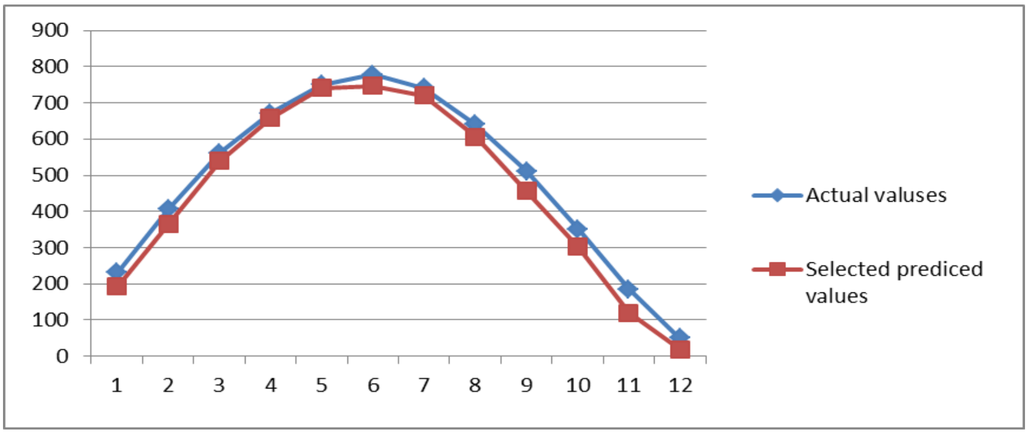


**Fig. 7 The averaged result of model selection. The X-axis is the daytime hour and Y-axis is the solar radiation.**

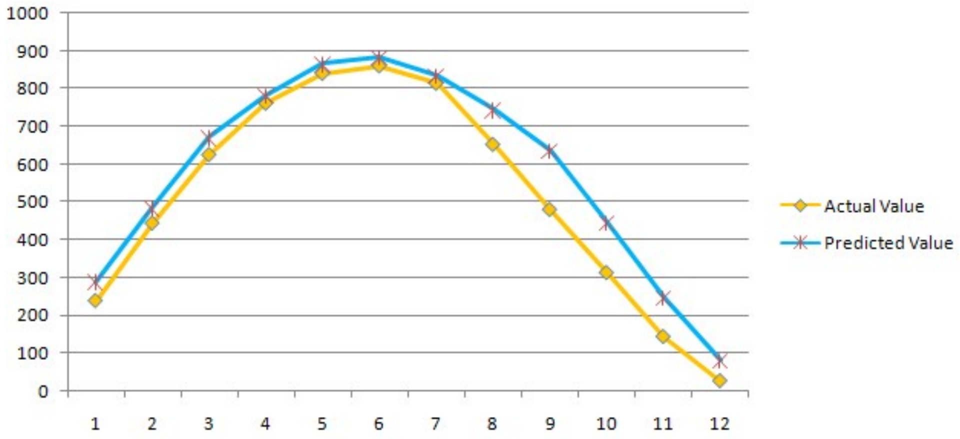


**Fig. 8 The result of model selection. The X-axis is the daytime hour and Y-axis is the solar radiation.**

We find that the predicted values obtained by the selected models generally fit the actual values well.  And in Fig. 8, from 8 hours ahead to 12 hours ahead, the differences between actual value and predicted value become larger. The fact is that model selection totally abandons other models and tries to pick up the "best" model, so it disregards the contribution of other models. Fig.9 below shows the even serious situation using model selection. The Actual value in Fig. 9 is close to Model 2 and Model 3, however, any forecasting result we pick up from Model 2 or Model 3 is biased and unfaithful. The actual value in this figure is the observation in March 16, 1988 from

Phoenix City.

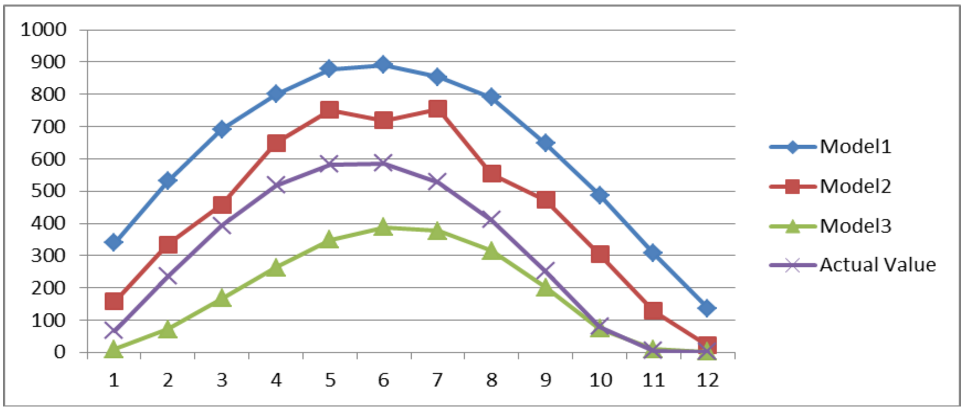


**Fig. 9 The solar intensity forecast obtained by the three models. The X-axis is the daytime hour and Y-axis is the solar radiation.**

It is reported in [33] that model averaging provides much reduced bias in estimates of the parameters unrelated to the response variable. In this regard, we introduce another method other than model selection——Bayesian model averaging, which considers the contribution from different models, and hence it avoids the problem of biasing to a particular model as illustrated in Fig. 9.

### 5.5.2 Bayesian model averaging

Bayesian model averaging [33] provides a way to tackle this problem. Its posterior distribution given data *D* is

$$p_r(\Delta|D) = \sum_{n=1}^{K} p_r(\Delta|M_k, D) p_r(M_k|D) \qquad (33)$$

This is an average of the posterior distributions under each of the models considered, weighted by their posterior model probability. The posterior probability for model $M_k$ is given by

$$p_r(M_k|D) = \frac{p_r(D|M_k) p_r(M_k)}{\sum_{n=1}^{K} p_r(D|M_l) p_r(M_l)} \qquad (34)$$

where

$$p_r(D|M_k) = \int p_r(D|\theta_k, M_k) p_r(\theta_k|M_k) d\theta_k \qquad (35)$$

is the integrated likelihood of model $M_k$, $\theta_k$ is the vector of parameters of model $M_k$ (e.g., for regression $\theta = (\beta, \sigma^2)$, $p_r(\theta_k|M_k)$ is the prior density of $\theta_k$ under model $M_k$, $p_r(D|\theta_k, M_k)$ is the likelihood, and $p_r(M_k)$ is the prior probability that $M_k$ is the true model (given that one of the models considered is true).

The final predicted value is

$$\sum_{n=1}^{L} p_r(n) Z_p(n) \qquad (36)$$

Using Eqn. (36), we calculate the final predicted values. Fig.10 shows the mean result of the predicted values using Bayesian model averaging and the actual values. Since the number of daylight hours is 12 hours for the Phoenix City dataset, we only show the 12 daylight hours averaged from 20 continuous days, i.e. Sep 10 - Sep 30, 1988, and the data samples from Jan 1, 1988 to Sep. 10 are employed as training data.

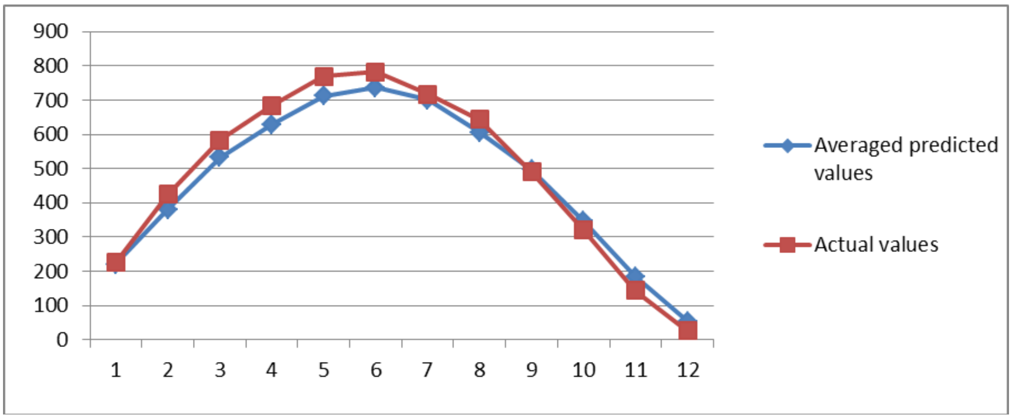


**Fig. 10 The mean result of the Bayesian model averaged predicted values and the actual values. The X-axis is the daytime hour and Y-axis is the solar radiation.**

Apart from that, the result of one day, i.e. in Sep. 10 is shown in Fig. 11.Generally, we find that the predictor obtained from Bayesian model averaging also fits the actual values well.

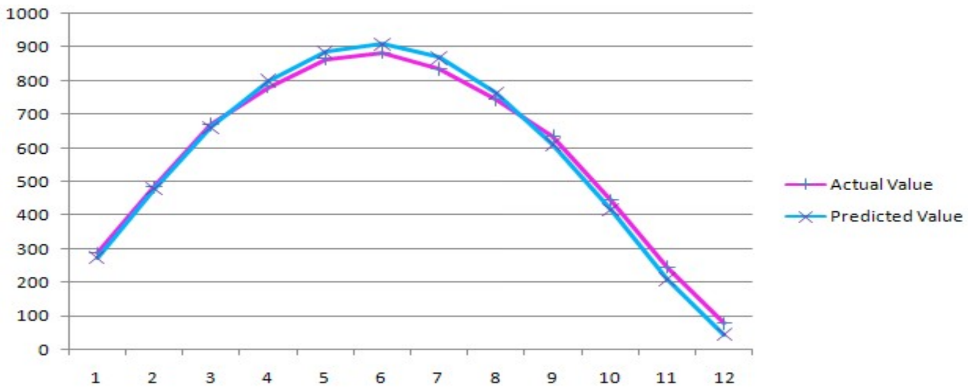


**Fig. 11 The predicted result of the Bayesian model averaging and actual value. The X-axis is the daytime hour and Y-axis is the solar radiation.**

Compared the Fig. 7 and Fig 10, we can in general, the forecasting performances of Bayesian model averaging and Bayesian model selection are comparable. But for more changeable data in a single day, the Bayesian model averaging may produce better results, such as in Fig. 11. Next we shall consider several hours ahead forecast instead of day ahead prediction.

## 5.6 Recursive Dynamic Factor Analysis with multiple partitioning, clustering and Model Averaging

A major limitation of the above scheme is that it only considers the same model for the whole day. Since in some places, such as Singapore, the weather may changes dramatically during different periods of the day, it may be beneficial to adopt different models for different period of the day to better capture the trend of the solar intensity under different weather. In this regard, we can further partition the solar intensities into different partitions, and we employ the proposed multi-clustering and model averaging approach to forecast the solar intensities for each partition.

For illustration purpose, we employ the same Phoenix data. So we further separate the daytime into 4 sections (time slots) according to the change of solar radiation, which means from 7:00 to 10:00 is the first timeslot, 10:00 to 13:00 is the second, 13:00 to 16:00 is the third and 16:00 to 18:00 is the last timeslot. The forecasting values are obtained by averaging the forecasts obtained for 10 days, i.e. from while the previous 250 days (specify the period) are used to provide data from different models and different timeslots. The data used are the same as mentioned in 5.2. The Fig.11 below shows the comparison the result of the predicted solar intensities with the actual values. We find that the new multiple partition scheme further improves the forecasting accuracy of the proposed multi-clustering and model averaging solar forecasting algorithm based on RDFA.

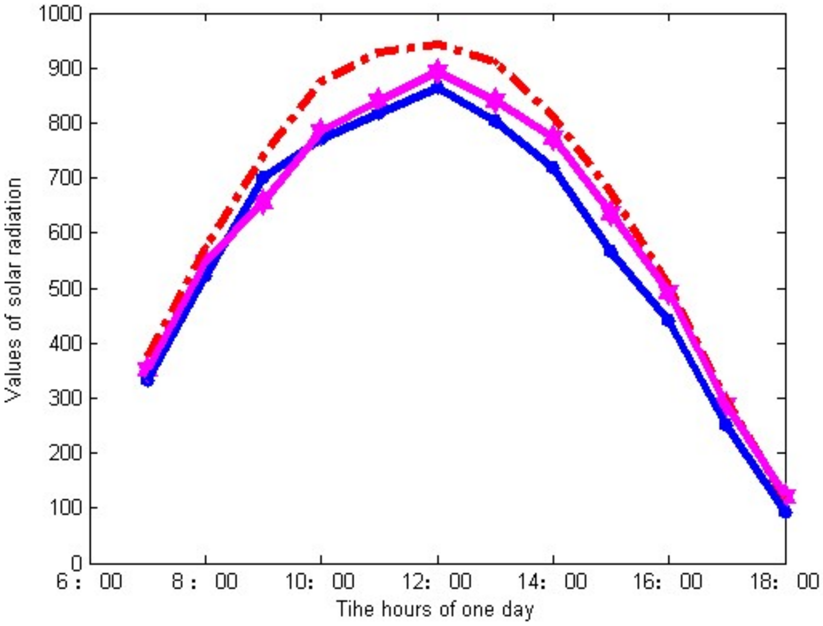


**Fig. 12 The comparison of further clustering result and 3-models result**

In conclusion, the proposed multiple partition, clustering and model averaging solar forecasting algorithm based on the RDFA further improves the forecasting accuracy of the RDFA algorithm and it is more efficient in solar radiation forecast because of its low complexity in real-time updating. It is particularly useful for applications under high fluctuation of weather, the timeslot can be further divided into 1-hour update or even shorter to meet different needs.

# Chapter6 Conclusion

A new multi-cluster and model averaging extension has been proposed based on the RDFA algorithm reported in [9] for solar intensity forecasting. Experimental results using the data set from the Measurements and Instrumentation Data Center baseline measurement system database shows that the RFDA is able to achieve better daily ahead forecast accuracy than other conventional algorithms. Moreover, the interval forecast provides an alternative measure to take account the uncertainty on the solar forecast, which may be useful for generator scheduling in the utilities. Besides, the proposed multi-cluster model further improves the accuracy of the existing RFDA algorithm by. considering the Bayesian model averaging of the models built by the clustered data.

Finally, we further consider a multi-partition extension that allows different models to be employed on different period of a day. This facilitates the RDFA to adopt to changing weather within a day, which may be particularly useful for regions with highly fluctuated weather, such as the tropical area. The efficient recursive implementation, ability on performing interval forecast and real-time update performance of the proposed multi-cluster algorithm based on RFDA make it as an attractive alternative to other conventional approaches to solar radiation forecasting and other possible applications.

# References

1. Detlev Heinemann, Elke Lorenz, Marco Girodo. Forecasting of Solar Radiation. [Online] Available TP: http://www.energiemeteorologie.de/publications/solar/conference/2005/Forcasting_of_Solar_Radiation.pdf pp: 1-6,2005
2. Luis Martín, Luis F. Zarzalejo, Jesús Polo, Ana Navarro, Ruth Marchante, Marco Cony, "Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning" *ISES Advancing Solar Energy Policy*, in Volume 84,pp.3-12, October 2010
3. Siem Jan Koopman Marius Ooms, "Forecasting Daily Time Series Using Periodic Unobserved Components Time Series Models," in Volume 24, pp.2-4, October 2006.
4. S. Cao, W. Weng, J. Chen, W. Liu and G. Yu ,"Forecast of Solar Irradiance Using Chaos Optimization Neural Networks," Asia-Pacific Conference on Power and Energy Engineering, Shanghai, China, Mar. Vol. 53, pp. 3–9.2009
5. Reikard G. “Predicting solar radiation at high resolutions: A comparison of time series forecasts", *Solar Energy,* Vol. 83, pp. 1–7. 2009
6. Rabbette, M., and P. Pilewskie, "Principal component analysis of Arctic solar irradiance spectra", [Online] Available : http://geo.arc.nasa.gov/sgp/radiation/rad5.html
7. Loskutov A.,Istomin I.A., Kotlyarov O.L "Testing and forecasting the time series of the solar activity by singular spectrum,” *Nonlinear Phenomena in Complex System*, pp.1-4, Sept. 2010.
8. H.C. Wu, S. C. Chan, K. M. Tsui and Y. Hou, “A New Recursive Dynamic Factor Analysis for Point and Interval Forecast of Electricity Price,” *IEEE Trans. Power Syst.*, submitted, 2012.
9. T. Gustafsson, “Instrumental variable subspace tracking using projection approximation,” *IEEE Trans. Signal Process.,* vol. 46, no. 3, pp. 669-681, Mar. 1998.
10. B. Liao, Z. *el at.*, “A new robust kalman filter-based subspace tracking algorithm in an impulsive noise Environment,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 9, pp. 740–744, Sep. 2010.
11. J Edward Jackson: "A user's guide to principal components,”. Wiley Series in Probability and Statistics,2003
12. Wasserman, L. (2000) Bayesian model selection and model averaging. *Math. Psychol.* 44, 92–10
13. Reikard: Comparison of ARIMA, neural networks and hybrid models in time series: tourist arrival forecasting, Journal of Statistical Computation and Simulation, Vol. 37, 1986, pp. 31-39,2010
14. Fatih O. Hocaog˘lu :Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks, *Solar Energy,* Vol. 17, pp.107-139,2007
15. Using an Integrated Artificial Neural Networks Model for Predicting Global Radiation: The Case Study of Iran. [Online].Available FTP: http://www.icrepq.com/icrepq07/359-azadeh.pdf
16. Andrew Harvey :Forecasting with Unobserved Components Time Series Models, Vol. 24, pp. 31-39,1999
17. Mechlouch, R.F. and A.B. Brahim, 2008. A global solar radiation model for the design of solar energy systems. Asian J. Sci. Res., 1: 231-238.
18. D. Cano, J. M. Monget, M. Albuisson, H. Guillard, N. Regas,L.Wald: ” A Method for the Determination of the Global Solar Radiation from Meteorological Satellite Data. ” *Solar Energy,* Vol. 37, pp. 31-39,1986,
19. R. D. DeGroat, “Non-iterative subspace tracking,” *IEEE Trans Signal Process*., vol. 40, no. 3, pp. 571-577, Mar. 1992.
20. S. C. Chan *el at*. “Robust recursive eigen-decomposition and subspace-based Algorithms with Application to Fault Detection in Wireless Sensor Networks,” *IEEE Trans. Instrum. Meas., to be appeared,* 2012.
21. K. Maribu, A. Galli and M. Armstrong, “Valuation of spark-spread options with mean reversion and stochastic volatility,” *International Journal of Electronic Business Management*, vol. 5, no. 3, pp. 173–181, 2007.
22. J. Bunch, C. Nielsen, and D. Sorensen, “Rank-one modification of the symmetric eigen problem,” *Numerische Mathematik,* vol. 31, no. 1, pp. 31-48, 1978.
23. K. Abed-Meraim, A. Chkeif, Y. Hua, and T. Paris, “Fast orthonormal PAST algorithm,” *IEEE Signal Process. Lett.*, vol. 7, no. 3, pp. 60-62, Mar. 2000.
24. S. C. Chan, Y. Zhou and W. Y. Lau, “Approximate QR-based algorithms for recursive nonlinear least squares estimation,” *IEEE ISCAS,* pp. 4333-4336, 2005.
25. A. Barron, J. Rissanen and B. Yu, “The minimum description length principle in coding and modeling,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2743-2760, Oct. 1998.
26. G. W. Stewart." An updating algorithm for subspace tracking. " *Computer Science Tech. Report Series*, vol. 40, pp. 71-87,1990
27. T. Winter, S. Mallory, T. Allen and D. Rosenberry, “The Use of Principal Component Analysis for interpreting ground water hydrographs,” *Ground Wate*r, vol. 38, no. 2, pp. 234 - 246, Mar. 2000.
28. G. H. Golub and C. F. Van Loan, Matrix Computations (3rd ed.), Baltimore, MD, USA, *Jonus Hopkins University Press,* 1996.
29. AristidisLikas: The global k-means clustering algorithm, *Pattern Recognition*,page 1-4,2003
30. SPSS17 software (2008). Spss predictive analytics software and solutions. SPSS Inc. [Online].Available FTP: http://www.spss.com.
31. Kass, R.E. &A.E. Raftery Bayesian factors. *Journal of the American Statistical. Association* pp:2-8,1995
32. Martin Swell, Model Selection. [Online]. Available: FTP:http://www.modelselection.org/model-selection.pdf.
33. Jennifer A. Hoeting, David Madigan, Adrian E. Raftery and Chris T. Volinsky, Bayesian Model Averaging: A Tutorial. Available FTP: http://www.iipl.fudan.edu.cn/~zhangjp/literatures/cluster%20analysis/hoeting.pdf