Electric Power Load Forecasting

Michael Roberts, Stacey Andreadakis, and Prasoon Karmacharya

Contents

Problem Statement **2**

Background **2**
- Why forecast load? . . . 2
- Time Series Data Collection . . . 2
- Time Granularity . . . 2
- Dimensionality Reduction & Similarity Matching . . . 2

Our Data Collection Methodology **3**
- Energy Market & Operational Data: NY ISO Load Data . . . 3
- Weather Data . . . 3

Time Series Modeling **3**
- What is time series modeling? . . . 4
- Types of time series forecasting models . . . 4
- Model Selection - Univariate Time Series . . . 4
  - What is ARIMA model? . . . 4
  - Requirements for ARIMA model . . . 5
  - Stationarity Check using Augmented Dickey-Fuller test (ADF Test) . . . 5
  - Hyperparameters for ARIMA model . . . 6
  - Model Evaluation Metrics: . . . 6

Introduction to Multivariate Time Series Analysis with $VAR(p)$ **6**
- Let's start by answering the following question: . . . 6
  - Is our system stationary? . . . 6
  - What does it mean for a variables to interact with each other in the context of time series data? 7
  - Endongenous versus Exogenous . . . 7
  - Now, let's get back to answering the question of stationarity . . . 7
  - Underpinnings of Stationarity . . . 8
  - But what about seasonality and trends, you ask? . . . 9

Model Selection: ARIMA & VAR **9**

Model Evaluation **9**

Conclusion **9**

Acknowledgments **10**

References **10**
- Data Sources: . . . 10
- Research Papers and Text Books: . . . 10

Problem Statement

Using time series analysis, can we forecast power load demands and detect system disruptions based on historical load data in conjunction with weather data early enough to minimize economic impact?

Background

Why forecast load?

Load forecasting is the process of predicting the future load demands in electrical systems using historical data. It is an essential component for planning in the electricity industry as it plays a vital role in power management and electric capacity scheduling. Electric power load forecasts can also serve as a key indicator in anomaly detection which could signal potential disaster or major power outage in the area. This could be a valuable information for electrical distribution companies.

In our project, we set out to engineer a model that will help us forecast short-term forecasting with reasonable accuracy.

Time Series Data Collection

When working with time series analysis forecasts there are several critical components that must be taken into consideration during the data collection process, especially when mining big data:

- Time Granularity
- Dimension Reduction
- Similarity Matching

Time Granularity

Information granularity can impact the complexity of the model as well as modeling accuracy. In other words, as information granularity decreases, accuracy of predictions increase, but the computational complexity increases significantly. In the context of time series analysis, time series information granularity is defined as each time stamp collected and the delta between said time stamps. There are many mathematical methods in determining the optimal information granularity for the time series problem one is trying to solve. For the purposes of this project, we leave this research to the reader. However, we list a few methods to guide each reader in their pursuit and provide links to some research papers as a starting point.

A couple of methods to aid in determining time granularity are as follows:

1. Granular Computing Theory
2. The Wavelet Analysis Method

Dimensionality Reduction & Similarity Matching

Although data mining has significantly improved, when dealing with raw data, specifically time series data, structuring time series data either in a general way or specialized manner can significantly impact the framework of solving a time series problem. More specifically, time series data generally is met with the "curse of dimensionality" since it can easily grow exponentially when collected at a high frequency, thus requiring high computational overhead. Consequently, when reducing dimensionality to minimize computational overhead, there are two objectives that are equally as important to consider:

1. The reduction of dimensionality by either reducing the number of data points or the number of records and looking at a smaller time period.

2. The transformation and representation of the raw data into a condensed form that preserves sufficient information in order to solve the problem at hand.

Methods for reducing dimensionality vary. The simplest method is sampling or resampling where time series data of length M can be reduced to length, K, such that K<M. Other methods include using the average value of each segment to represent each new data point in the formation of a compressed version of the raw data. One of these methods is known as the Piecewise Aggregate Approximation. However, the method one chooses to use is dependent upon the context of the problem. For instance, one may approximate time series data points by linear interpolation or linear regression. Moreover, if one desired to only preserve features that are intuitively indicative of how to model the series such as features that exhibit trends, seasonality or are cyclical in nature, then the PIP (Perceptually Important Points) method could be employed.

When working with multiple time series data sets, similarity matching is necessary to search for trends between the sets that could possibly be used during the modeling process. Popular methods for similarity search mechanisms include the Discrete Fourier Transform and Discrete Wavelet Transform.

All of this is to say that there are many methods available to the keen data scientist who wishes to minimize computational overhead while maintaining the integrity of the time series data. For the stalwart reader, further details can be found here and here.

Our Data Collection Methodology

Given the time constraints of our project, and the objective, we collected data from two primary sources listed below, where an explanation and link to each data set can be found. Data was resampled so that each data set would match accordingly. Since one of our objectives was to gain a comprehensive understanding of working with time series data, the dimensionality of our data was intentionally limited. However, we were aware that additional variables such as the day of the week, holidays, and trends in technology (for instance, the sales and use of LED lightbulbs) could impact load forecasting for future energy consumption requirements.

Energy Market & Operational Data: NY ISO Load Data

Source: https://www.nyiso.com/load-data

NY ISO (Independent System Operator) are an independent, not-for-profit corporation responsible for operating the state's bulk electricity grid, administering New York's competitive wholesale electricity markets, conducting comprehensive long-term planning for the New York state's electric power system.

Weather Data

World Weather Online covers approximately 3 million cities/towns worldwide. Their weather forecasts are trusted and used by a wide variety of companies and organisations from SME's to large corporate clients. That includes Intelligent Planet, Sonitus Systems, Skydive Mag, G&D Creations, Surfcast, Weathercare and many more.

Time Series Modeling

If you can relate to having a very limited knowledge of time series forecasting before the start of a project, then you are in the right place. In this work, we not only modeled our load data to make short-term forecast but also attempted to elucidate the complexity behind time-series analysis in easily understood, step-by-step process. Please note that more advanced neural network techniques can be employed but this work does not cover these topics.

What is time series modeling?

Time series involves temporal datasets that change over a period of time. A common approach to model time series is to regard the label at the current time step X_t as a variable dependent on previous time steps X_{t-k}. Many natural phenomenon that one might be interested in predicting have an inherent temporal nature to them as previously defined. The load data we obtained from NYISO, which measures the electric power load (MW) usage every 5 minutes, was no exception.

Types of time series forecasting models

The type of time series model you can use to forecast mostly depends on the nature of the time series data at hand. Does the data contain a single variable or multiple variables? Are the variables endogenous or exogenous? What are the characteristics of your data – does it demonstrate stationarity, seasonality or is there a trend? All these factors are deemed critical when performing time series forecasting. As such, there are two major classes of time series forecasting each requiring careful consideration that will be discussed throughout this work.

- **Univariate**: Univariate time series forecasting refers to the case in which the set of observations over time of a single variable is considered. Fundamentally, such a model is forecasting time series of the target value on nothing more than the time series itself.
- **Multivariate**: Multivariate time series forecasting refers to the case in which the set of observations overtime of multiple variables are considered.

In our project, we considered both the univariate and multivariate cases.

Model Selection - Univariate Time Series

One of the models we used to forecast the power load data is a simple univariate, **ARIMA** model. The model selection and hyperparameter search process for the univariate model described in this paper are what is commonly referred to as Box-Jenkins method. To help those new to time-series analysis, this paper attempts to describe the core components of said framework. A more comprehensive treatment of this methodology can be found here.

What is ARIMA model?

ARIMA, stands for Auto Regressive Integrated Moving Average. The ARIMA model is comprised of three main components.

1. Autoregressive
2. Integrated
3. Moving Average

As the name suggests, **Autoregressive (AR)** means we regress a variable on itself–i.e., the future values are regressed on the values preceding it. AR model of order p has a general form of:

$$\begin{aligned} Y_t &= \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} \\ &= \beta_0 + \sum_{i=1}^{p} \beta_p Y_{t-p} \end{aligned}$$

Here, p is the hyperparameter that dictates the number of previous values of Y to put into our model.

Moving Average (MA) Model is another important component of the ARIMA model that takes into consideration the previous error terms as inputs. It incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations. In essence, the MA captures

the white noise of the previous steps. Thus, accounting for sudden shocks in our data. MR model of order q has a general form of:

$$\begin{aligned} Y_t &= \mu + w_1\epsilon_{t-1} + w_2\epsilon_{t-2} + \cdots + w_p\epsilon_{t-q} \\ &= \beta_0 + \sum_{k=1}^{q} w_q\epsilon_{t-q} \end{aligned}$$

Here q is the hyperparameter that dictates the number of previous errors ϵ to put into our model.

Integrated process: In the case of univariate ARMA model one of the key requirements is stationarity in our data. What is stationarity? Simply put, stationary data has a constant mean and a constant variance over time. However, if the data is not stationary, we use differencing to obtain stationarity. This is exactly what **Integrated process** in ARIMA allows us to do, if and when necessary. In other words, when we have data that shows a trend we can remove the trend by differencing (d) the time steps y_t with y_{t-1}.

Both the AR(p) and MA(q) components of the ARIMA model explain the stochastic nature of our time series data. While the main function of the AR model is to explain the long-term trend in our data, MA allows us to capture the white noise in our time series data. By adding the integrated process into the mix, the ARIMA model can be succinctly represented as:

$$Y_t^{(d)} = \beta_0 + \sum_{k=1}^{p} \beta_k Y_{t-k}^{(d)} + \sum_{i=1}^{q} w_i\epsilon_{t-1} + \epsilon_t$$

Requirements for ARIMA model

We learned that ARMA model requires the data to be stationary. Since the integrated process in the ARIMA model corrects for non-stationarity by applying differencing between previous time steps, we need not worry about stationarity. However, we do need to pick appropriate lag order (d). Common approaches to identifying the lag order (d) for univariate time series are to use plots of Autocorrelation function (ACF), Partial Autocorrelation function (PACF), and/or Augmented Dickey Fuller (ADF) Test. While ACF plot is merely a bar chart of the coefficients of correlation between a time series and lags of itself, the PACF plot is a plot of the partial correlation coefficients between the series and lags of itself. For our time series, we decided to use PACF to determine the lag because ACF fails to remove the relationships of intervening observations in the time series data.

Stationarity Check using Augmented Dickey-Fuller test (ADF Test)

We checked for stationarity of our data using the Augmented Dickey-Fuller test. ADF test is a commonly used statistical significance test to determine if the time series is stationary or not. It is a subcategory of a Unit Root Test. The presence of a Unit Root implies that the time-series is non-stationary.

Null hypothesis in ADF Test is, H_0: $\alpha = 1$

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} + \delta_2 \Delta Y_{t-2} + \cdots$$

where, Y_t, is the value of the time-series at time t Y_{t-1}, is the first lag term of time-series

Essentially, the ADF test uses a series of linear regressions of Δy_t to regress against t and y_{t-1} and check the value of γ. If $\gamma = 0$, then we have a random walk process. However, if $-1 < 1 + \gamma < 1$, then we have a stationary series. In doing so we are checking the p-value for the series. When the p-value is less than 0.05, we reject the null hypothesis. In addition to determining the stationarity of the time series data, the ADF test also helps us determine the number of differencing operations we need to apply to obtain stationarity in non-stationary time-series data.

Using this ADF test we were able to statistically reject the null hypothesis, and prove that our load data was stationary. Hence, no differencing was required, i.e., differencing order (d) = 0.

Hyperparameters for ARIMA model

As discussed in the previous section, the ARIMA model has three hyperparameters:

- p is the order of lag observations included in the AR model, also known as lag order.
- d is the differencing order (how often we difference the data), also known as the degree of difference
- q is the number of previous errors to put into our MA model.

1. Determining optimal hyperparameters For our univariate model, we used an exhaustive grid-search method to determine our optimal hyperparameters. However, this may not always be feasible since it can get computationally expensive if you have a large dataset. Upon determining the hyperparameters we fit our univariate ARIMA model and predicted on the split of the train set and then compared our predictions to the validation set respectively. Despite the simplicty of the model it performed fairly well on the very short term prediction but as expected the forecast for the long term load converged towards the mean.

Model Evaluation Metrics:

Commonly used evaluation metric in time-series analysis are AIC and BIC. Rather than a concrete metric, both AIC and BIC are heuristics information criterion that are commonly used in evaluating the model parsimoniously and informing hyperparameter choices. AIC and BIC works by evaluating the model's fit on the training data, and adding a penalty term for the complexity of the model. In essence, these metrics share similar fundamentals to regularization in linear models like Lasso and Ridge regularization.

1. AIC (Akaike Information Criterion) The desired result is to optimize over the AIC, in order to find an optimal upper bound of the lag order. This serves the eventual goal of maximizing fit on out-of-sample data by means of grid searching.
$$AIC = -2ln(\hat{L}) + 2k$$
where, $\hat{L}$ = Likelihood function k = number of parameters estimated by the model
2. BIC (Bayesian Information Criterion) Similarly, one optimizes over the BIC by minimizing the maximum likelihood function in the following equation.
$$BIC = kln(n) - 2ln(\hat{L})$$
where, $\hat{L}$ = Likelihood function n = number of data observation (sample size) k = number of parameters estimated by the model.

Introduction to Multivariate Time Series Analysis with $VAR(p)$

The majority of data preprocessing was conducted during inital phases of methodology. However, multivariate time series analysis modeling requires different–albeit only slightly–processes for verifying systemic structures such as seasonality, trends and stationarity. Methods such as the Johansen Test, measuring multivariate model performance, and the basic underpinnings of lag order selection will be discussed.

Let's start by answering the following question:

Is our system stationary?

Forecasting, in general, throws a curveball at even the most adept data scientists: time. To be more specific, unlike classification and regression models, time series models are temporally dependent between observations.

In the univariate case, this may seem intuitive enough but for multivariate case there are few nuances that one has to take into consideration.

Event A_0 occurs at time t_0, event A_1 occurs at time t_1 and so forth. Event A_1 depends on event A_0 at the prior time step t_0, so if you were to predict A_2, then looking at the relationship between A_0 and A_1, at the prior time steps, seems like a fair place to start.

Now, imagine a world where event type A_i does not exist in a vacuum. It almost feels unnatural for A_i to exist in complete isolation without any interactions. Afterall, we exist in a dynamic system. That is what multivariate time series models attempt to capture: unlike univariate models such as ARIMA, which typically assume autoregressive relationships, multivariate models allow for a more complex relationship between variables. Thereby allowing the model to leverage the use of multiple variables and each variable's respective sytemic structure.

What does it mean for a variables to interact with each other in the context of time series data?

The value of one variable is not only related to its predecessors in time, but additionally depends on past values of other variables. For example, household consumption expenditures could potentially depend on income, and interest rates. If this is the case, then income and interest rates could be used as additional information to predict household consumption expenditures. This brings us to the concept of endogenous and exogenous variables.

Endongenous versus Exogenous

1. Endogenous Variables

   Input variables that are influenced by other variables in the system and on which the output i.e. the forecast variable depends. In the case of predicting househould consumption expenditures, income and interest rates would be endogenous.

2. Exogenous Variables

   Input variables that are not influenced by other variables within the system and on which output depends. Continuing with the example of predicting household consumption expenditures, an exogenous variable could be the rate of unemployment.

Now, let's get back to answering the question of stationarity

Once endogenous and exogenous variables have been identified, checking stationarity is necessary to determine model selection. However, how to determine if a multivariate problem is stationary deviates from the univariate case.

We define a $VAR(p)$ model (VAR model of order p) as follows:

$$y_t = v + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t$$

where $t \in \mathbb{Z}$, $y_t = (y_{1t}, ..., y_{Kt})'$ is a $(K \times 1)$ random vector, the A_i are fixed $(K \times K)$ coefficient matrices, $v = (v_1, ..., v_t)'$ is a fixed $(K \times 1)$ vector of intercepts such that $\mathbb{E}(y_t)$ may be nonzero. Furthermore, $u_t = (u_{1t}, ..., u_{Kt})'$ is a $K-$ dimensional white noise process such that $\mathbb{E}(u_t) = 0$,$\mathbb{E}(u_t u_t') = \Sigma_u$ and $\mathbb{E}(u_t u_s') = 0$

Suppose the generation of each event y_i starts at some time t, then we see

$$
\begin{aligned}
y_1 &= v + A_1 y_0 + u_1, \\
y_2 &= v + A_1 y_1 + u_2 \\
&= v + A_1(v + A_1 y_0 + u_1) + u_2 \\
&= (I_K + A_1)v + A_1^2 y_0 + A_1 u_1 + u_2, \\
&\vdots \\
y_t &= (I_K + A_1 + \ldots + A_1^{t-1})v + A_1^t y_0 + \sum_{t=0}^{t-1} A_1^{-i} u_{t-i}
\end{aligned}
$$

Thus, the vectors $y_1, \ldots, y_t$ are uniquely determined by $y_0, u_1, \ldots, u_t$. Additionally, the joint distribution of $y_1, \ldots, y_t$ is determined by the joint distribution of $y_0, u_1, \ldots, u_t$.

It is important to note that while in application we assume that a process has started at a specified period in time, the definition of $VAR(p)$ assumes that it has been started in the infinite past. In consideration of this assumption, the question follows. What kind of process adheres to such a property?

Underpinnings of Stationarity

To better understand this, let's take another look at the $VAR(1)$ process:

$$
\begin{aligned}
y_t &= v + A_1 y_{t-1} + u_t \\
&= (I_K + A_1 + \ldots + A_1^j)v + A_1^{j+1} y_{t-j-1} + \sum_{i=0}^{j} A_1^i u_{t-i}
\end{aligned}
$$

Absolutely summable If all eigenvalues of A_1 have modulus less than 1, the sequence $A_1^i, i = 0, 1, \ldots$, is absolutely summable. Recall from matrix and vector algebra for A_1^i to be absolutely summable the vectors obtained as a result of the transformation corresponding to the given summation method must be absolutely convergent.

A system is said to be bounded input-bounded output stable (BIBO stable) if the output signal is bounded for all input signals that are bounded. In the multivariate time series case, the bounds are defined such that the eigenvalues are greater than zero and less than one.

Below, we provide a refresher for understanding the process of finding eigenvalues and what that means in the context of our $VAR(p)$ model:

$$
y_t = v + A_1 y_{t-1} + ... + A_p y_{t-p} + u_t
$$

Let A be an $(m \times m)$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. Then there exists a nonsingular matrix P such that:

$$
P^{-1} A P = \begin{bmatrix} \Lambda_i & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \Lambda_n \end{bmatrix} =: \Lambda
$$

where,

$$
\Lambda_i = \begin{bmatrix} \lambda_i & 1 & 0 & \ldots & 0 \\ 0 & \lambda_i & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & \ddots & 1 \\ 0 & 0 & \ldots & \ldots & \lambda_i \end{bmatrix}
$$

such that $A = P\Lambda P^{-1}$. This decomposition of A is also known as the Jordan canonical form.

Finally, if multiple roots of the characterstic polynomial, obtained by solving $det(A - \lambda I_n) = 0$, such that $0< \lambda <1$ exist, then the system is **stationary**.

Furthermore, the sequence $\sum_{j=0}^{\infty} \alpha_{kl,j}$ is finite for all \$k,l=1,. . . ,m,\$ such that $\alpha_{kl,j}$ is an elment of coefficient matrix A^j, as defined above and in $VAR(p)$.

Thankfully, the creators of statsmodels have provided us with a function that does all of the above in the few clicks of a button and a single import, called the **Johansen Test**. However, please be aware that the underlying concept here is imperative to make a sound decision in your model selection process. If the Johansen Test fails and, therefore, data is nonstationary, then $VAR(p)$ is not an appropriate choice. It is also important to note that the Johansen Test can only be used for up to 12 variables.

But what about seasonality and trends, you ask?

Well, it turns out that the same applications used (ACF, PACF plots & seasonal decompose) for univariate models must also be applied to each endongenuos variable. Methods for dealing with seasonality and trends include but are not limited to using other models: VARMA which accounts for seasonality and VECMs which account for cointegration i.e. a nonstationary system. Unfortunatley, due to time constraints, we were unable to use VARMA, even though our data indicated there were seasonal trends.

Model Selection: ARIMA & VAR

While our data satisfied conditions for stationarity and did not demonstrate any trends, it did exhibit seasonality. As per the aforementioned, timing limited our abilities to implement models that would correct for seasonlity.

Model Evaluation

Despite not having accounted for seasonality, our univariate model was successful in providing short term forecasts of load impact. Addtionally, the same can be said of the VAR model with endogenous and exogenous variables. To evaluate both model performances, we optimized for more precise short-term forecasting versus consistency in forecasting. Therefore, our objective was to minimize the RMSE in both cases. While the RMSE does not provide a comprehensive picture of model performance, the metric proved to suffice for modeling purposes as a heuristic.

Note: While not comprehensively discussed in this project, something to consider when selecting a model and, in turn, evaluating its performance is the lag. There are a few common criterion used during this process, namely BIC, AIC and HQ. Without getting into the details, as mentioned earlier, one may compare the function of said criterion to the lasso and ridge regularization techniques used in linear models. For instance, similar to lasso's degree of penalty, BIC will typically yield a lower upper bound of lag order. Whereas, AIC tends to be more lenient in this regard.

Conclusion

Given that our objective was to engineer a model that prioritized precision over consistency, we can confidently say that we achieved our goal in both the univariate and multivariate case. However, we acknowledge that in both cases our models were overly simplified in the interest of understanding the complexity inherent in forecasting with time series analysis. That said, we included ample research and further readings to not only continue our studies with time series analysis methods but to also provide a resource for any data scientist in search of a comprehensive guide for forecasting with time series analysis.

Acknowledgments

We'd like to thank the following people for aiding us in furthering our understanding of time series analysis. Travis Whalen, Noelle Brown, Dan Wilhelm, Niloofar Bayat, Kunal Mahajan, Vishal Misra, and Dan Rubenstein.

References

Data Sources:

- NYISO Load Data
- NYISO Power Grid Data
- Electric Disturbance Events (OE-417) Annual Summaries

Research Papers and Text Books:

- **New Introduction to Multiple Time Series Analysis** by Helmut Lütkepohl
- **Deep Learning for Time Series Forecasting** by Jason Brownlee
- **Introduction to Time Series Forecasting with Python** by Jason Brownlee
- Granular Computing Theory
- The Wavelet Analysis Method
- An Approach to Dimensionality Reduction in Time Series
- A comparison of DFT and DWT based similarity search in time-series databases
- ARIMA Models and the Box–Jenkins Methodology