International Journal of Control and Automation
Vol.7, No.8 (2014), pp. 93-102
http://dx.doi.org/10.14257/ijca.2014.7.8.09

Comparative Study on Short-term Electric Load Forecasting Techniques

Bongil Koo and Juneho Park

Department of electrical and computer engineering, Pusan National Univ.
koobg@pusan.ac.kr

Abstract

In this paper, the problem of short-term load forecasting is divided into load classification and forecasting. Load classification is needed to obtain meaningful load data as input to train forecasting models. To this end, k-NN and K-mean algorithms are presented. K-mean and k-NN algorithms can handle seasonal load classification and daily load classification, respectively. The classified load data are used to train forecasting models, which are Artificial Neural Networks, Simple Exponential Smoothing, and ARIMA models. As a real case study, we tried to forecast the electric power load of the Republic of Korea. A comparison between the classified and non-classified load forecasts demonstrates the efficiency of the proposed method.

Keywords*: Short-term Electric Load forecasting, ARIMA, k-NN, K-mean, Artificial Neural Network, Simple Exponential Smoothing*

1. Introduction

Because of the rapid growth in the size of power systems and the replacement of traditional central power systems with distributed power systems, grids have become more complex and electric power load forecasting has recently become more important. This is true, not just to ensure the safe operation of a grid without faults, but because of economics. A grid must be able to handle the uncertain output of distributed sources and reduce the losses caused by rapid load variation [1-2].

Because electricity cannot be stored efficiently in large quantities, the amount of generated power must meet the demand, including grid losses. Therefore, short-term electric load forecasting results are used to determine the generation control functions, including interchange evaluation, scheduling, and unit commitment. It also plays an important role in a restructured power system. Forecasting the system demand is necessary for making appropriate market decisions.

Short-term electric load forecasting has many difficulties. Most of all, the seasonal and daily characteristics of loads have considerable complexity. The load at any given hour depends, not only on the previous hour, but also on the load at that hour on the previous day. Electric load patterns reflect social and economic conditions, which are difficult to properly reflect in a model.

When developing a forecasting model, it is first necessary to categorize the load data from the data pool. Normally, most of the load data follow the characteristics for a particular day of the week; for instance, this Saturday will be nearly the same as last Saturday. Thus, forecasting models are trained using load data that has been classified following the days of the week listed on a calendar. However, this can sometimes cause problems, because the electric load is affected by many factors such as social factors, weather, air-temperature, *etc*. These factors are not displayed on the calendar. If

ISSN: 2005-4297 IJCA

forecasting is performed using such improperly categorized load data, this could lead to incorrect forecast results. Consequently, correctly classified load data must be prepared before training forecasting models and forecasting. Therefore, in this paper, we classified the electric load using K-mean and k-NN. The former is applied to seasonal classification, and the later is applied to classification based on the day of the week. Weekly load patterns (from Mon to Sun) are depicted in Figure 1.

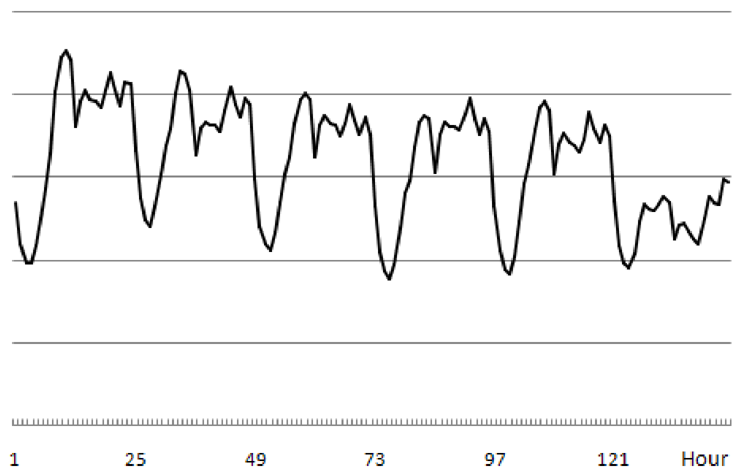

Figure 1. Weekly Load Patterns (Mon. – Sun.)

Several researches have investigated electric load forecasting using regression-based and time series methods [3]–[7], Bayesian estimation model [8] and state space and Kalman filtering technology [9]. In addition to these approaches, artificial intelligence techniques have been adopted [10]–[14]. Although nonlinear functions are very elastic, their poor convergence time and linguistic information processing performance are weak points.

2. Load Classification

2.1. K-mean Clustering

K-mean algorithm was proposed by Cox (1957) and Fisher (1958) and developed by Hartigan (1975) and MacQueen (1967). It takes the input parameter, k, and partitions a set of n objects into n cluster. After clustering, intracluster similarity is high but the intercluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. K-mean algorithm proceeds as follows [15].

Input:
- ▫ k: the number of cluster
- ▫ D: a data set containing n objects.

Output: A set of k clusters.
Method
1) arbitrarily choose k objects from D as the initial cluster centers
2) repeat

3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster
4) update the cluster means, i.e., calculate the mean value of the objects for each cluster
5) It the square-error criterion converges, stop.

$$E = \sum_{i=1}^{k} | p - m_i |^2 \qquad (1)$$

Where, E: the sum of the square error for all objects in the data set
P: the point in space representing a given object
m_i: the mean of cluster C_i

2.2. k-Nearest Neighbor Classification

K-Nearest Neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition.

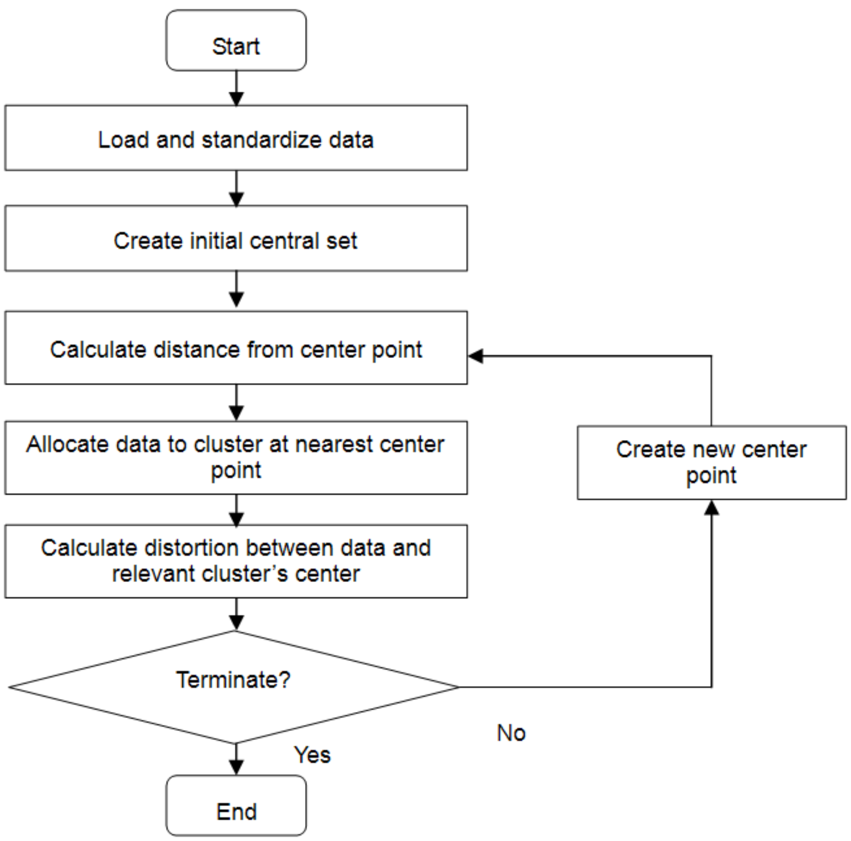

Figure 2. K-mean Clustering Procedure

k-NN is based on learning by analogy, that is, by comparing a given test set with training sets that are similar to it. The training sets are described by n attributes. Each set represents a point in an n dimensional space. In this way, all the training sets are stored in an n-dimensional space. When given an unknown set, a k nearest neighbor searches the pattern space for the k training sets that are closest to the unknown set.

"Closeness" is defined in terms of a distance metric.

In this paper, we used the Euclidean distance between two points or sets,

$$X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$$
$$X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$$

$$\text{Distance}(X_1, X_2) = \sqrt{\sum_{i=1}^{k} (x_{1i} - x_{2i})^2} \quad (2)$$

When we determine a optimal value for k, we can determine experimentally. Starting with k=1, we use a test set to estimate the error rate of the classifier. This process can be repeated each time by incrementing k to allow for one more neighbors. The k value that gives the minimum error rate may be selected.

3. Load Forecasting

3.1. Simple Exponential Smoothing

The simple exponential smoothing model is a moving average method that places a greater weight on the recent demand when calculating the average.

$$F_{t+1} = F_t + \alpha(Z_t - F_t) \quad (3)$$
$$F_t = \alpha Z_{t-1}(1-\alpha)F_{t-1} \quad (4)$$

Where, F_{t+1}: predicted value at time point t+1
Z_t: observed value at time point t
Z – F : error
α: smoothing constant

Larger the smoothing constant α value (0–1), the greater the weight given to the recent data. For data with a particular pattern, more weight should be given to recent data, and for data without a particular pattern or with severe variation, more past data should be included. In addition, α should minimize the mean square error and the mean absolute percentage error (MAPE).

The advantage of this technique is that it needs only three data (the prediction value just obtained, the observed value, and smoothing constants) whereas the weighted moving average method requires the previously observed value and the weight.

3.2. Auto Regression Integrated Moving Average Model(ARIMA Model)

ARIMA model is very useful in describing various homogeneous nonstationary time series. This model involves performing a regression of the time series itself, and the general pth order AR process$\{Z_t\}$ satisfies

$$(1- \varphi_1 B- \varphi^2 B-\ldots-\varphi_p B^p)Z_t= a_t\varphi \quad (5)$$

Where,
a^t: a white noise process with average 0 and dispersion σa

φ: auto regression factor
B : backward operator satisfying

$$BZ_t = Z_{t-1}, B^2Z_t = Zt\text{-}2, .., Z\_, \ldots, B^nZ_t = Z_{t-n} \quad (6)$$

An MA process is a general linear process of Z_t whose limited number of weights is not 0, and a Z_tth order MA process can be written as

$$Z=(1- \theta_1 B- \theta_1 B_2 -...- \theta qBq)a_t \quad (7)$$

Where,
a_t: white-noise process with dispersion $\sigma^2 a$
Z_t: observed value at time point t
θ: parameter.

The AR model and MA model can be combined into a low-order ARMA model when the orders p,q are high, *i.e.*, when there are too many parameters. This will lead to a better approximate value.

The average, dispersion, and autocovariance of time series data are assumed to be independent of temporal changes. However, in actual time series the data are usually abnormal *i.e.*, the average and dispersion depend on the temporal flow. To normalize the data, logarithmic conversion or dispersed stabilization conversion can be applied to the dispersion, and a difference calculation can be applied to the average. The dth order differential time series of the resulting normalized time series data becomes an ARIMA(p,q,r) model whose general equation is

$$\varphi_p(B)(1-B)^d Z_t = \varphi_q(B)a_t \quad (8)$$

3.3. Artificial Neural Network(ANN)

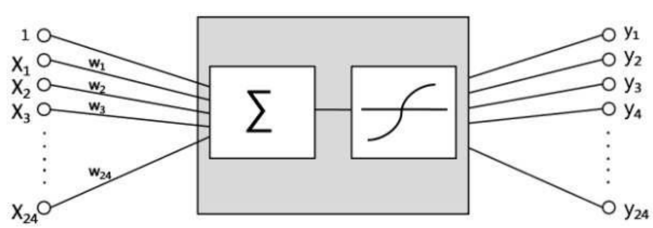

Figure 3. An Artificial Neuron

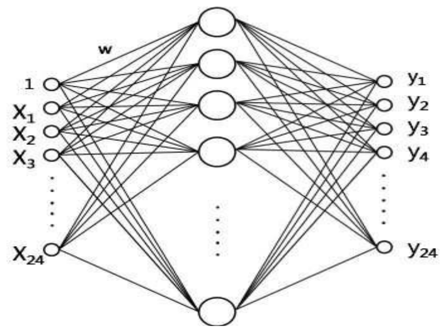

Figure 4. A Two-Layer Feed-Forward Neural Network

Neural networks are mathematical tools originally inspired by the way the human brain processes information. Their basic unit is the artificial neuron, schematically represented in Figure 3. The neuron receives (numerical) information through a number of input nodes, processes it internally, and puts out a response. The processing is usually done in two stages: first, the input values are linearly combined, then the result is used as the argument of a non-linear activation function. The combination uses the weights w_i attributed to each connection. The activation function must be a non-decreasing and differentiable [13].

The neurons are organized in a way that defines the network architecture. The one we shall be most concerned with in this paper is the multilayer perceptron (MLP) type, in which neurons are organized in layers. The neurons in each layer may share the same inputs, but are not connected to each other. If the architecture is feed-forward, the outputs of one layer are used as the inputs to the following layer. The layers between the input nodes and the output layer are called the hidden layers. Figure.4 shows an example of a network with 28 input nodes, 2 layers (one of which is hidden), and 24 output neurons.

4. Case Study

In this paper, we classify the electric load data of Korean national electric load by seasonal(spring, summer, fall, winter) and type of day(Monday, other weekday, Saturday, Sunday) using K-mean and k-NN respectively. After that, using result of the classified load, we trained three load forecasting models (Simple Exponential Smoothing, ARIMA, Artificial Neural Network) and forecasted next day load.

4.1. Load Classification

In this case study, each load has different max-min value according to daily time unit and different according to hourly time unit time-series, two criterions are considered: 1) seasonal criterion, 2) daytype criterion. Seasonal classification was performed by K-means algorithm. This process depends on how choose initial center points. Several ways are exist:1) randomly assign, 2) select some of load data, 3) average values that possibly classified to the same cluster. After experimental test, we choose third way. The test result is shown Table1.

The load data which are classified by K-mean, classify by day type again. k-NN is adopted to day type classification. Before classification, we must decide ‘k’, meaning how many neighbors choose to do measuring similarity between set and neighbor.

Normally, 'k' is adopted odd number and not exceeded 10. 'k' is chosen 1 by experimental test.

Table 1. Results of Seasonal Load Classification for Different Initial Center Allocations

Allocation of center of clusters	Seasonal Classification	Iteration avg.
random	not clear	8.2
select load data	possible	6.64
average values that possibly classified to the same cluster	possible	3.84

4.2. Load Forecasting

Comparative studies are presented to evaluate the efficiency of the proposed method. For this purpose, three forecasting methods are considered: SES, ARIMA, and ANN. For ANN, we chose a multilayer perceptron with 28 inputs, 1 hidden layer, and 24 outputs. The number of hidden neurons is known to affect the forecasting accuracy more or less significantly. A back propagation algorithm is selected or the training algorithm, while the selected iteration tolerance after the error is less than 0.005. The accuracies of the three models were compared after being trained using, non-classified load data and classified load data. Generally, there were only slight differences in the results, as tabulated in Table 2. The forecasting accuracy was improved by 0.041~0.486 in MAPE for three models.

Table 2. Comparison of Forecasting Results Between

Method	Input data	Day type	MAPE	Max	Min
SES	nonclassified load data	Mon.	1.713	2.845	1.008
		Week	1.735	2.912	1.213
		Sat.	2.220	3.003	1.131
		Sun.	2.164	2.321	1.648
	classified load data	Mon.	1.644	2.541	1.022
		Week	1.517	2.117	0.965
		Sat.	2.179	2.586	1.037
		Sun.	1.866	2.632	1.247
ARIMA	nonclassified load data	Mon.	1.335	1.835	1.112
		Week	1.652	1.963	1.397
		Sat.	1.522	2.261	1.121
		Sun.	1.588	1.834	1.362
	classified load data	Mon.	1.267	1.536	1.002
		Week	1.463	1.693	1.033
		Sat.	1.531	2.294	0.902

		Sun.	1.500	1.933	0.115
ANN	nonclassified load data	Mon.	1.432	2.261	0.182
		Week	1.410	2.831	0.844
		Sat.	2.226	2.439	0.303
		Sun.	1.946	2.212	0.097
	classified load data	Mon.	1.366	2.255	0.031
		Week	1.243	2.031	0.009
		Sat.	2.134	2.403	0.126
		Sun.	1.460	2.545	0.083

However, the results for the weekdays were quite different for the three methods. The weekdays data contain many variations caused by the possibility that the day was actually a holiday or was before or after a holiday. For each result, the amounts of the improved accuracy are different, but all of the forecasting results show improved accuracy.

5. Conclusion

In this paper, the electric load forecasting strategy of the classified load using K-mean and k-NN algorithm has been proposed. The problem of short-term load forecasting is divided into load classification and forecasting. Load classification is needed to obtain meaningful load data as an input to train forecasting models. If forecasting is performed such improperly categorized load data, this could lead to incorrect forecast results. However, load classification commonly just depends on calendar, it contains possibilities that classification couldn't be done properly. That is to say, there are load data that existing differences day type between calendar and electrical pattern attribute and we have possibilities to use these data to train forecasting models. Therefore, we did seasonal classification of electric load data using K-mean algorithm and day type classification using k-NN algorithm.

Using a classified load data, we trained three load forecasting models such as, SSE and ARIMA, ANN. Comparing the accuracy of three models between trained by raw load data classified load data, generally, there are slightly different results. However, weekday result shown quiet different result over three methods. The results show possibilities to improve forecasting accuracy when we focus on electric load classification

Acknowledgements

This work was supported by National Research Foundation of Korea(NRF-2013R1A1A2013798).

References

[1] P. Sharma, "Transient Stability Investigations of the Wind-Diesel Hybrid Power Systems", International Journal of Energy, Information, and Communications, vol. 1, no. 1, **(2010)**, pp. 49-63.

[2] D.-J. Kang and S. Park, "A Conceptual Approach to Data Visualization for User Interface Design of Smart Grid Operation Tools", International Journal of Energy, Information, and Communications, vol. 1, no. 1, **(2010)**, pp. 64-76.

[3] H. Mori and K. Kosemura, "Optimal regression tree based rule discovery for short-term load forecasting," in Proc, IEEE Power Eng. Soc.Winter Meeting, Columbus, OH, **(2001)** January, pp. 421–426.
[4] W. Charytoniuk, M. Chen and P. Olinda, "Nonparametric regression based short-term load forecasting," IEEE Trans. Power System, vol. 13, no. 3, **(1998)** August, pp. 725–730.
[5] P. K. Dash, G. Ramakrishna, A. C. Liew and S.Rahman, "Fuzzy neural networks for time-series forecasting of electric load," Proc. Inst. Elect.Eng., Gen., Transm., Distrib., vol. 142, no. 5, **(1995)** September, pp. 535–544.
[6] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," IEEE Trans. Power Syst., vol. 16, no. 4, **(2001)** November, pp. 798–805.
[7] S. J. Huang and K. R. Shih, "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations," IEEE Trans. Power Syst., vol. 18, no. 2, **(2003)** May, pp. 673–678,
[8] K. Liu, S. Subbarayan, R. R.Shoults, M. T.Manry, C. Kwan, F. I. Lewis and J. Naccarino, "Comparison of very short-term load forecasting techniques", IEEE Trans. Power Syst., vol. 11, no. 2, **(1996)**, pp. 877–882.
[9] A. P. Douglas, A. M. Breipohl, F. N. Lee and R.Adapa, "The impact of temperature forecastuncertainty on Bayesian load forecasting", IEEE Transactions on Power Systems, vol. 13, no. 4, **(1998)**, pp. 1507-1513.
[10] D. C. Park, M. A. El-Sharkawi, R. J. Marks II, L. E. Atlas and M. J. Damborg, "Electric load forecasting using an artificial neural network", IEEE Transactions on Power Systems, vol. 6, no. 2, **(1991)**, pp. 442-449.
[11] K. Y. Lee, Y. T. Cha and J. H. Park, "Short-term load forecasting using an artificial neural network," IEEE Trans. Power Syst., vol. 7, **(1992)** February, pp. 124–132.
[12] I. Drezga and D. S. Rahman, "Short-term loadforecasting with local ANN predictors," IEEE Trans. Power Syst., vol. 14, **(1999)** August, pp. 844–850.
[13] H. S. Hippert, C. E. Pedreira and R. C. Souza, "Neural Networks for Short-Term Load Forecasting", A Review and Evaluation, IEEE Trans. Power Syst., vol. 16, no. 1, **(2001)**, pp. 44-55.
[14] C. Kim, I. Yu and Y. H. Song, "Kohonen neural network and wavelet transform based approach to short-term load forecasting", Electric Power Syst. Res., vol. 63, **(2002)**, pp. 169–176.
[15] J. Han and M. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann, San Francisco, **(2006)**, pp. 402-404.

Authors

Bon-Gil Koo, he received his B.S. and M.S. Degrees from Pusan National University, Busan, Korea in 2008 and 2010, respectively. He is currently PH.D candidate at Pusan National University. His research interests include electric load forecasting and intelligent systems applications to power systems.

June Ho Park, he received the B.S., M.S. and Ph.D. degrees from Seoul National University, Seoul, Korea in 1978, 1980, and 1987, respectively, all in electrical engineering. He is currently a Professor at the School of Electrical Engineering, Pusan National University, Busan, Korea. His research interests include intelligent systems applications to power systems. Dr. Park has been a member of the IEEE POWERENGINEERING SOCIETY.