EUNITE Network Competition: Electricity Load Forecasting

Ming-Wei Chang, Bo-Juen Chen, and Chih-Jen Lin

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, 106, Taiwan
cjlin@csie.ntu.edu.tw

Abstract. EUNITE network recently organized a world-wide competition on electricity load forecasting. This paper details our approaches and results where the main machine learning technique used is support vector machine.

1 Introduction

Recently EUNITE network organized a world-wide competition on electricity load forecasting. Given the temperature and the electricity load from 1997 to 1998, the competitors are asked to predict the daily maximal load of January 1999. The main machine learning technique we used on this problem is the support vector machine (SVM).

Load forecasting has been an important topic in power systems research. Some surveys are, for example, [2, 5]. Most of the earlier works aimed at predicting short-term loads such as one-day ahead prediction. However, in this competition, we are asked to predict the loads for a whole month. Comparing to the short-term prediction, this problem is much harder as the result of the long-term prediction might degenerate due to the error propagation.

This paper is organized as follows. In Section 2, we briefly introduce basic concepts of support vector machines (SVM). Then in Section 3 we describe our analysis for the data set and Section 4 presents our methods. In the end, we demonstrate experimental results in Section 5.

2 Support Vector Machine

Support vector machine (SVM) is a new and promising technique for data classification and regression [6]. In this section we briefly introduce support vector regression (SVR) which can be used for time series prediction. Given training data $(x_1, y_1), \ldots, (x_l, y_l)$, where x_i are input vectors and y_i are the associated

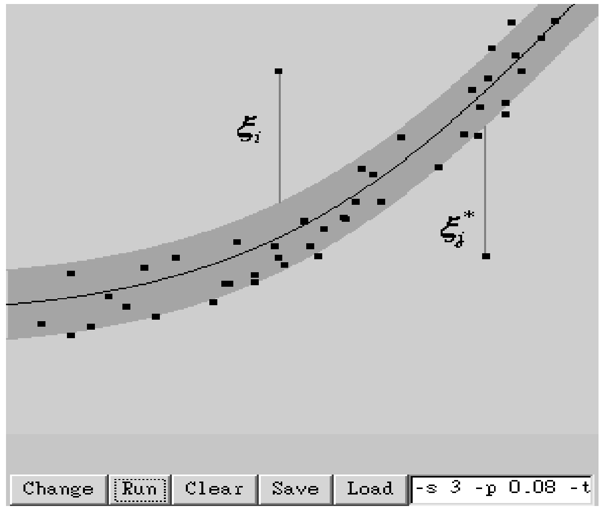

Fig. 1. Support Vector Regression

output value of x_i, the support vector regression is an optimization problem:

$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{1}$$
$$\text{subject to} \quad y_i - (w^T\phi(x_i) + b) \leq \epsilon + \xi_i,$$
$$(w^T\phi(x_i) + b) - y_i \leq \epsilon + \xi_i^*,$$
$$\xi_i, \xi_i^* \geq 0, i = 1, \dots, l,$$

where x_i is mapped to a higher dimensional space, ξ_i is the upper training error (ξ_i^* is the lower) subject to the ϵ-insensitive tube $|y - (w^T\phi(x) + b)| \leq \epsilon$. The parameters which control the regression quality are the cost of error C, the width of tube ϵ, and the mapping function ϕ.

The constraints of (1) imply that we would like to put most data x_i in the tube $|y - (w^T\phi(x) + b)| \leq \epsilon$. This can be clearly seen from Figure 1. If x_i is not in the tube, there is an error ξ_i or ξ_i^* which we would like to minimize in the objective function. For traditional least-square regression ϵ is always zero and data are not mapped into higher dimensional spaces. Hence SVR is a more general and flexible treatment on regression problems.

In this competition, we would like to deploy SVR for time series prediction. An earlier example using SVR is [4]. Given any time series $(\cdots, y_{t-\Delta}, \cdots, y_{t-1}, y_t, \cdots)$, for the training data we consider $(y_{t-\Delta}, \cdots, y_{t-1})$ as attributes of x_i and y_t as the target value. Then using the last Δ elements of the sequence as a test data, we predict the first unknown value. By sequentially adding newly obtained data as attributes (and removing the earliest element), we continue to predict more elements.

3 Data Analysis

In this competition, the data offered include electricity load and temperature. A list of holidays is also provided. The load data set contains the load per half hour of each day from 1997 to 1998 while the temperature data set provides the average daily temperature from 1995 to 1998. Like most data mining tasks here we have to analyze the data first before applying any techniques on them. Some properties observed are as follows:

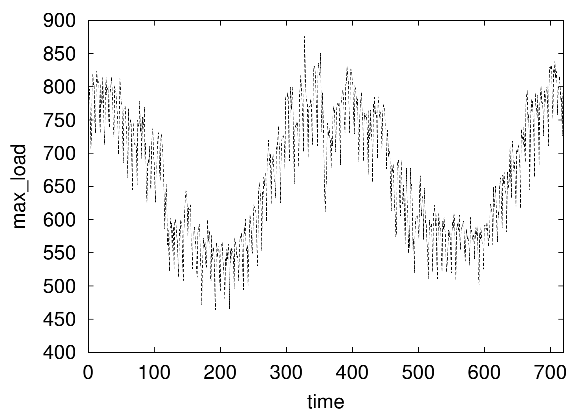

Fig. 2. Maximum Load

3.1 Electricity Load

The given load data are the electricity loads recorded every half hour, from 1997 to 1998. With so many numeric data, we first collected the maximal load of each day as it is the objective of the competition. Figure 2 shows the maximal load in each day of 1997 to 1998. We also analyzed the sum of daily loads. The maximal load and the summed load share similar patterns.

Load Periodicity In Figure 2 clearly there are some periodical patterns for the maximum load data. First, the load is changing with the season: high demand for electricity in the winter while low demand in the summer. Furthermore, the load pattern of weekdays is different from that in the weekend. More precisely, in the weekend the load is usually lower. In addition, the electricity demand on Saturday is a little higher than that on Sunday.

Holiday Effect Earlier work have pointed out that holiday might be a factor which can influence the load. From the two-year load data, it is easily to find out that the load usually lowers down on holidays. With further scrutiny, we found out that the load is not only lower on holidays but also depends on what holiday it is. On some major holidays such as Christmas or New Year, the demand for electricity may be affected more compared with other holidays.

3.2 Weather Influence

As we have pointed out earlier, the load data have some seasonal variation, which indicates a great influence of the temperature. It is clear to see that because of the heating use, in the winter higher temperature causes lower demands. Figure 3 shows the negative correlation between the load demand and the temperature.

There is another interesting observation: the temperature at December 31st, 1998 is the lowest from 1997 to 1998. This observation might imply the high uncertainty of the temperature and load of the incoming January 1999, and thus increase the difficulty of the load prediction.

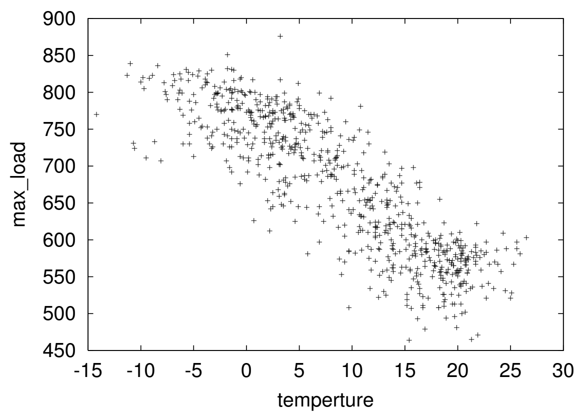

Fig. 3. Maximum Load

4 Methods and Data Preparation

When using SVM for this problem, it is considered a global approach as the learning model is generated by gathering and training all existing data. On the contrary, there are also local modeling approaches which in fact do not have the training stage. We will compare these two approaches and pick the better one for generating our final results.

4.1 Local Modeling

Local models generate predictions by finding segments of the time series that closely resemble the segment of the points immediately proceeding the point to be predicted. Then the prediction is usually the average of elements that occurred immediately after these similar segments of points. A successful example on time series prediction using the local model is in [3].

Practically we have to decide the length of each segment and the number of similar segments. Usually a validation process is conducted in order to decide these parameters. Here we simply consider loads of seven days as a segment and select the closest segment.

4.2 Global Modeling

In Section 2 we have roughly mentioned how to use support vector machines for time series prediction. Here we provide details for solving this particular problem.

If considering only the load information, the date set is a simple time series where past information can be trained for predicting future data. Now in addition to the load information we also know the calendar dates and all holidays. Hence we would like to encode all these information if possible.

Therefore, the training data of one particular day has its load as the target value y_i and the following attributes:

- Seven attributes for maximal loads of the past seven days
- Seven binary attributes indicate which day in a week
- One binary attribute indicates whether this is a holiday or not.
- One attribute for the daily average temperature. However, we may or may not be able to use this attribute due to the lack of temperature in January 1999. We will elaborate more on this later.

We then use SVM to train a model using the above encoding. The newly predicted load will be included as an attribute and used for the next prediction. For example, after obtaining an approximate load of January 1, 1999, it is used with loads of December 26-31, 1998 for predicting that of January 2. We continue this way until finding an approximate load of January 31.

Information such as calendar dates and holidays in January 1999 are known in priori so there are no problems to encode them. However, the temperature of January 1999 is not provided so we may have to approximate it as well. As SVM currently works only for models with a single output, we have to train two SVMs, one for predicting loads and the other for temperature. To be more precise, we train another SVM where each training data has temperature of the past seven days as attributes and the current temperature as the target (output) value.

5 Experiments and Results

5.1 Implementation

We used MATLAB for experiments on local modeling. For the global modeling using SVM, we consider LIBSVM [1], a library for support vector machines.

To evaluate models, we separate the data to two sets, data of January 1998 as the testing set and the rest as the training set. For both methods mentioned in Section 4, we have to choose several parameters. For example, the length of segments for local modeling and the cost of error C in the SVM formulation (1). This is achieved by conducting cross validation on the training set. In other words, the training set is further divided for training and validation. The parameter set which achieves the best validation accuracy will be used for finding the final model for future prediction. Due to the lack of time, for local modeling we restrict to use the most similar segment and try only few segment lengths. Finally we decide to use seven-day information as a segment.

On the other hand, for SVM, there are also quite a few parameters. Some important ones are

1. cost of error C,
2. the width of the ϵ-insensitive tube,
3. the mapping function ϕ, and
4. how many days included for one training data.

As there are too many combinations of the above parameters, for each training data we simply include data of the previous seven days. In addition, we consider only the RBF function where $\phi(x_i)^T\phi(x_j) = e^{-\gamma\|x_i - x_j\|^2}$ and use the default $\epsilon = 0.5$ of LIBSVM. Therefore, parameters left are C and γ which were chosen by a five-fold cross validation.

Furthermore, after some preliminary experiments, we realize that discarding data in the summer leads to better results. Hence we totally do not use information from April to September.

In addition, for all the validation procedures, we evaluate results using the mean square error.

5.2 Results

Beside some miscellaneous tests, we mainly experiment with the following three approaches:

1. Local modeling
2. SVM without temperature information
3. SVM with temperature information

We worry that though temperature is an important factor, it is not clear whether two SVMs together can produce good results. Thus we test the case without using temperature as well.

We find out that it is very difficult to predict the temperature. In particular, if one day the temperature suddenly drops or increases, we cannot correctly predict it so results after that day are erroneous. We conclude that if temperature is used, the variation is higher as sometimes the performance is good but sometimes is very bad.

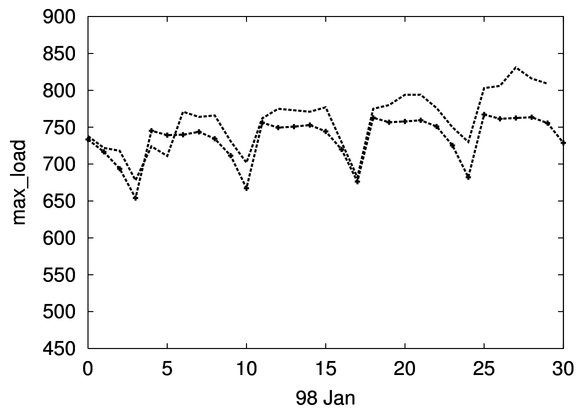

Fig. 4. Prediction for Jan. 1998 (line: real loads; line-points: predicted loads)

Thus we decide to give up using the temperature information. If we calculate the average load for each day in a week, we find out that without using temperature, our predicted values are very close to them. This is reasonable as without temperature, the information about which day in a week becomes the most influential factor. However, we also see the model manages to hold the trend. For instance, in the whole month if the load is slightly increasing, our model reveals this pattern too though in general its changing rate is slower. In Figure 5.2, we can see that the load in January 1998 is creasing. Our prediction shows the same pattern but its increase is not as large as the real load.

We also see that the model really returns smaller values for holidays whose loads are usually lower. However, the gap between the predicted value and the real value is still bigger than that of non-holidays. Therefore, results after encountering a holiday become *more* inaccurate. In particular the first day of January is a holiday so this problem is quite serious. Finally we decide to ignore the holiday information while doing the prediction. In other words, we treat all 31 days in January 1999 as non-holidays. We think that even though the performance on holidays may not be good, the total error is still less. Some earlier work separated holidays and non-holidays and train different models for them. However, now we have data in only two years so information about holidays is not enough.

5.3 Other Considerations

We have also tried other options though they do not show significant improvements and are not included. For example, we tried to give less weight for the holiday attribute while training the model. Originally each attribute has values between 0 and 1 after scaling but we can further reduce the holiday attribute to a smaller ranger like $[0, 0.3]$.

Another modification is as follows: Now we use seven binary attributes for indicating a day in a week. We guess that maybe seven can be reduced to two: weekday and weekend. However, the result does not change much.

6 Conclusion

Based on experiments presented in the previous section, we choose the approach of using SVM without the temperature information for generating our final model.

References

1. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
2. H. S. Hippert, C. E. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting: a review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, 2001.
3. J. McNames, J. A. K. Suykens, and J. Vandewalle. Winning entry of the K. U. leuven time series prediction competition, 1998.
4. K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 243–254, Cambridge, MA, 1999. MIT Press.
5. P. Murto. Neural network models for short-term load forecasting. Master's thesis, Department of Engineering Physics and Mathematics, Helsinki University of Technology, 1998.
6. V. Vapnik. *Statistical Learning Theory.* Wiley, New York, NY, 1998.