Applied Soft Computing 10 (2010) 868–875

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A soft computing system for day-ahead electricity price forecasting

Dongxiao Niu [a], Da Liu [a], Desheng Dash Wu [b,*]

[a] School of Business Administration, North China Electric Power University, Beijing 102206, China
[b] Reykjavik University, Kringlunni 1, IS-103 Reykjavík, Iceland and RiskLab, University of Toronto, 1 Spadina Crescent, Toronto, Canada ON M5S 3G3

ARTICLE INFO

Article history:
Received 23 September 2008
Received in revised form 16 July 2009
Accepted 17 October 2009
Available online 24 October 2009

Keywords:
Electricity price forecasting
Support Vector Machine (SVM)
Particle Swarm Optimization
Self-Organizing Mapping (SOM)

ABSTRACT

Hourly energy prices in a competitive electricity market are volatile. Forecast of energy price is key information to help producers and purchasers involved in electricity market to prepare their corresponding bidding strategies so as to maximize their profits. It is difficult to forecast all the hourly prices with only one model for different behaviors of different hourly prices. Neither will it get excellent results with 24 different models to forecast the 24 hourly prices respectively, for there are always not sufficient data to train the models, especially the peak price in summer. This paper proposes a novel technique to forecast day-ahead electricity prices based on Self-Organizing Map neural network (SOM) and Support Vector Machine (SVM) models. SOM is used to cluster the data automatically according to their similarity to resolve the problem of insufficient training data. SVM models for regression are built on the categories clustered by SOM separately. Parameters of the SVM models are chosen by Particle Swarm Optimization (PSO) algorithm automatically to avoid the arbitrary parameters decision of the tester, improving the forecasting accuracy. The comparison suggests that SOM–SVM–PSO has considerable value in forecasting day-ahead price in Pennsylvania–New Jersey–Maryland (PJM) market, especially for summer peak prices.

1. Introduction

In a competitive power market, it is very important for all participants to forecast the electricity price with high accuracy. Generators, power dealers and customers take actions to maximize their profits or utilities according to the forecasting price. Also regulation authorities monitor the market functioning with the forecasting. The price is affected by the power demand, the grid constraints, the bid strategies of the suppliers and the purchasers, and the price of fuels, etc. There often exist some extreme prices in the power market [1]. Besides, for the periodic variety of hourly load, the hourly price also presents daily, weekly and annual seasonally fluctuations. These factors make the price forecasting more difficult than the load forecasting.

Time series models, such as Auto-Regressive Integrated Moving Average (ARIMA) and General Autoregressive Conditional Heteroscedasticity (GARCH) are used to forecast the price, which present good performance in a stable market [2–6]. However they are not suitable for modeling more complex forecasting, especially for the complex electricity price forecasting. In essential, the time series models fit linear relations better than nonlinear relations. Moreover, most time series modeling simply uses the historical price, and neglects other influential information, whose fluctuation causes the uncertainty of volatile prices [20]. Artificial Neural Network (ANN) and other intelligent algorithms are also applied to forecast electricity price [6–11]. Among the intelligent algorithms, generalized regression neural network (GRNN) with principal components analysis (PCA) shows great potential in electricity price forecasting [7]. ANN can approximate any nonlinear function accurately, where both the historical prices and the influential information are considered as inputs to ANN model. So ANN is more suitable for complex price forecasting. The model is trained and validated by the given data, and then when some new information is put into the trained model, it will generate output. The main principle of training ANN is to find the minimization of the training error, without considering the complexity of the model. It is known that as the model complexity increases, the degree of freedom of the model decreases, which possibly cause either overfitting or underfitting problem. The overfitting or underfitting dilemma of ANN has not been resolved well so far, so the generality of the ANN cannot be guaranteed. It is also a big challenge to choose suitable parameters for ANN modeling.

Recently Support Vector Machines (SVM) based on the theory of Structure Risk Minimization (SRM) have attracted more and more attention in the field of forecasting for their excellent performances. The principle of training SVM is to minimize both the training error and model complexity, which is the milestone of the intelligence algorithm. SVM has more desirable performance than the ANN. SVM models combined with other intelligent algorithms showed their great potential in classification and regression [12–15]. Some literatures showed that with the parameters of SVM models optimized by Genetic Algorithm (GA), the performances are

* Corresponding author.
E-mail address: dash@risklab.ca (D.D. Wu).

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.10.004

significantly improved [13]. As the electricity price is volatile and there exits complex relations between the price and its influential factors, SVM outperforms time series and ANN modeling.

In most of the price forecasting studies, especially in the hourly price forecasting with intelligent algorithms, only one model is built to forecast the 24 hourly prices of the next day. However, it is a hard task to reflect all the characteristics of 24 different hourly prices by that single model. So the model becomes underfiting for some hourly prices; but at the same time, it may become overfitting for some others, which leads to unsatisfactory results. Building 24 different models to forecast the 24 hourly prices may solve the above problem. However, it is not so practical in modeling peak price, because a mass of similar training data for hourly price are often beyond available for intelligent algorithms model to guarantee a good generality. Even within the same hourly price data set, some price may differ greatly from the others for the anomalous weather or other factors. Forecasting of the anomalous prices by the regular models is unsatisfactory.

The prices of different hours may have similar characteristics, so they can be modeled by one model. On the other hand, the anomalous price are apparently alien to prices within the same hour, but they might be similar to other hours, so they can be model together with the prices of the other hours or just be model of their own, which enlarges the training and validation data set; then the data with the similar characteristics in test set are forecasted by this model. Those anomalous prices are also incorporated with the price of the nearest characteristics. Thus it is feasible to forecast the 24 hourly prices with different models.

Fuzzy c-mean (FCM) algorithm is used for daily load pattern clustering, then for each cluster corresponding models are developed for forecasting prices [8]. A technique of ANN model based on similar days method is successfully proposed to forecast day-ahead electricity price in the PJM market [9].

Self-Organizing Map neural network (SOM), an unsupervised neural network, is an excellent method to cluster data according to their similarity. SOM is used here to discriminate the difference between the prices and to cluster the data based on their characters. A multi-layer perceptron (MLP) with SOM to cluster the input has a good performance in forecasting the power load [10]. In [11], SOM is proposed to forecast short-term load. However, it has some defects because some kinds of patterns may not be possible responded properly [11]. In [15], SVM experts with SOM are proposed to cluster the whole input space for time series forecasting. The simulation shows that SVMs experts improve a lot in the generalization performance in comparison with the single SVM model [15].

Then SVM are used to calculate the forecast of the different categories separately, which helps to improve the forecasting significantly. In order to choose the best parameters of SVM, an optimization algorithm, Particle Swarm Optimization (PSO) is adapted in this paper.

The rest of the paper is organized as follows. In Section 2, we describe the fundamental of SOM, SVM and Particle Swarm Optimization (PSO) algorithm; Section 3 demonstrates the approach of building the hybrid models. Experiments of the hybrid models and compared models are showed in Section 4. Finally, the conclusions are presented in Section 5.

2. Fundamental of SOM, SVM and PSO

2.1. SOM

SOM is a widely applied neural network algorithm to cluster data brought forward by professor Kohonen. SOM is very suitable for the analysis and visualization of high dimensional data. It converts complex nonlinear relationships between high dimensional input data into simple geometric relationships. SOM learns to find regularities and correlations of input, and roughly preserves the most important relationships and topological of the original data and inherently clusters future input data accordingly [16].

SOM includes two layers: the input layer and the output layer. The input layer is fully connected to the output layer of map nodes. The learning process is competitive and unsupervised. When an input is presented the output nodes compete to represent the pattern. The node whose vector of weights is closest to the input pattern wins the competition. The winner is then updated by moving its weight vector closer to the input pattern. Units near the winner are also moved, as training progresses units that are neighbors tend to come to represent similar patterns, while nodes far from each other in the map represent dissimilar patterns. The nods within a cluster tend to activate the same output unit, while nods from other clusters will be represented by separate units [16].

The following describes the basic steps of SOM:

Step 1. Initialize the weight vectors with small random values;
Step 2. Present the input vector to the map;
Step 3. Compute Euclidean distance between the input vector and each neuron in the map using;
Step 4. Select the winner neuron (the best matching neuron);
Step 5. Update the weights for the winner neuron and all neurons in the neighborhood.

Repeat until the stopping criterion is reached.

2.2. SVM

It is pointed out in the statistics learning theory that with probability $1-\delta$, the following bound (1) holds:

$$R(f) \le R_{emp}(f) + \sqrt{\frac{h\ln(2l/h)+1-\ln(\delta/4)}{l}} \tag{1}$$

where $R_{emp}(f)$ is called the empirical risk, and $R(f)$ is the expected risk, l is the sample number, h is a non-negative integer called Vapnik Chervonenkis (VC) dimension [17].

From (1) it can be seen that the expected risk not only depends on the empirical risk, but also depends on VC dimension and the sample number. So it must minimize both the empirical risk and the VC dimension to minimize the expected risk, i.e., the test error [18].

The basic idea of SVM is: first to map the input vectors into high dimension space with nonlinear transformation, then to build an optimization model in the high dimension space applying SRM.

Supposing a set of data, $D=\left\{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^l, y^l)\right\}$, $\mathbf{x}\in R^n$, $y\in R$, where $\mathbf{x}$ is the input vector, y is the expected output; l is the number of the data. The data are mapped into high dimension space H by some nonlinear mapping $\varphi(\cdot)$. The optimization regression function is built in the space H:

$$f(\boldsymbol{x}) = \boldsymbol{w}\cdot\varphi(\boldsymbol{x}) + b \tag{2}$$

In (2), $\mathbf{w}$ is the weighted vector, $\mathbf{w}\in R^k$, b is the intercept, $b\in R$.

There are two kinds of SVM for regression problems: ε-SVR and ν-SVR [18]. They are similar basically. This paper explains the fundamental of SVM with ν-SVR.

ν-SVR is supposed to solve the optimization problem (3):

$$\min_{w,b,\xi_i^-,\xi_i^+} \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + C\left(\nu\varepsilon + \frac{1}{l}\sum\nolimits_{1=1}^{l}(\xi_i^- + \xi_i^+)\right)$$
$$\text{subject to}\quad \begin{cases} y^i - f(\boldsymbol{x}^i) \le \varepsilon + \xi_i^- \\ f(\boldsymbol{x}^i) - y^i \le \varepsilon + \xi_i^+ \\ \xi_i^-, \xi_i^+ \ge 0, \quad i = 1, \ldots, l. \end{cases} \tag{3}$$

where C is the upper bound, a pre-specified value, $C>0$; ν is a parameter to control the number of support vectors; and ξ_i^-, ξ_i^+