Information Systems 53 (2015) 161–169

Contents lists available at ScienceDirect

Information Systems

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

Short term power load prediction with knowledge transfer

Yulai Zhang*, Guiming Luo

School of Software, Tsinghua University, Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China

ARTICLE INFO

Available online 28 January 2015

Keywords:
Transfer learning
Gaussian process
Power load prediction

ABSTRACT

A novel transfer learning method is proposed in this paper to solve the power load forecast problems in the smart grid. Prediction errors of the target tasks can be greatly reduced by utilizing the knowledge transferred from the source tasks. In this work, a source task selection algorithm is developed and the transfer learning model based on Gaussian process is constructed. Negative knowledge transfers are avoided compared with the previous works, and therefore the prediction accuracies are greatly improved. In addition, a fast inference algorithm is developed to accelerate the prediction steps. The results of the experiments with real world data are illustrated.

1. Introduction

Power load forecast is a crucial issue in the management of the smart grids. Accurate forecast can greatly cut down the operational cost of power systems [1]. The scheduling and rescheduling of the generation plan and maintenance plan are both relied heavily on the short term power load forecast. The results of the short term power load forecast are also important to the other basic functions such as interchange evaluation and security assessment in the smart grid [2].

In the early researches, classic statistical models such as the AR (auto-regressive) model [3,4], the ARMAX (auto-regressive moving average with external input) model [5], the BJ (Box–Jenkins) model [6,7] and the SS (state-space) model with Kalman filter [8,9] are used in the short-term power load forecast. In the past decade, along with the fast development of artificial intelligence and machine learning technic, mainstream AI based methods and models, such as ANN (artificial neural networks) [10,11], SVM (support vector machine) [12,13], evolutionary algorithm [14] and GP (Gaussian Process) model [15] have been adopted to solve this problem. Many hybrid models and methods are also proposed in the last few years.

The neural network model with harmony search is developed in [16]. The PSO (particle swarm optimization) is implemented with SVR (support vector regression) in [17] and ANN in [18].

From the latest development mentioned above, we can see that parameter optimization is a critical issue to the improvement of the prediction results. The parameter optimization is a time consuming task for many models and unlikely to be performed in every prediction steps particularly when the time interval for prediction is very short. Most of the above models are parametric, except Gaussian process. One advantage of the GP model is that, as a non-parametric model, the hyper-parameter optimization step in the Gaussian process model is equivalent to the model structure selection step of the parametric models such as ANN and SVM and the prediction step of GP models is equivalent to both the parameter optimization step and the prediction step of the parametric models [19]. So the latent parameters of the GP model will be adapted to the new data automatically. This is the initial reason for us to consider the GP model in our work.

The power load values can be affected by many hidden variables of both natural environments (such as wind speed and sunlight) and human activities (such as holidays

* Corresponding author.
E-mail addresses: zhangyl08@mails.tsinghua.edu.cn (Y. Zhang), gluo@tsinghua.edu.cn (G. Luo).

http://dx.doi.org/10.1016/j.is.2015.01.005
0306-4379/

and emergency events). Most of these variables are difficult to obtain and some of them are uneasy to quantify. As a result, most of the existing works use the history load values as the elements of the feature vectors to obtain the predictions. Yet there are still some works try to use the weather conditions to forecast the power load, such as [20,21]. However, for the specific problem of short term load forecast, when the weather information is concerned, the short term weather forecast itself is a nontrivial task with great uncertainty. On the other hand, the hidden variables always have similar values for the neighboring cities that are located in the same region. Therefore, in the task of power load forecast, the prediction accuracies can be increased significantly by utilizing the knowledge and information from the data of these nearby cities.

Transfer learning [22] is a research hot spot in the recent years. In a transfer learning process, the performance of the target task will be improved by using the transferred knowledge from the selected source tasks as well as that of the target task itself. Transfer learning methods have been successfully used in the problems such as natural language processing [23] and wireless localization [24]. In these problems, common knowledge and information such as language grammars and wireless communication environments are shared among the tasks. So it is reasonable to believe that the transfer learning methods can also give promising results in the filed of power load prediction.

The first problem in developing a transfer learning method is how to find the source tasks for a given target task. In the multi-task learning method proposed in [25], tasks are clustered manually. Therefore, negative knowledge transfers [26], which are destructive to the prediction accuracies, are observed due to incorrect human choices. Another important reason for the negative transfer is that the knowledge transfer between tasks may not be symmetric. Even if the knowledge of one task improves the performance of another task, the improvement on the opposite direction cannot be guaranteed. In order to solve these problems, a transfer learning method with automatic source task selection is developed for the power load prediction problems in this paper. Transfer learning has also been named as the asymmetric multi-task learning in some literatures. The proposed method chooses different source tasks for different target tasks to improve the predictions.

The second problem in this research is the computational time. The concept of the covariance function in the Gaussian process (GP) model can describe the relationships of the data points from different tasks easily. This is the second and the most important reason for us to use the GP model in our work. But one of the deficiencies of the Gaussian process model is that the computational time of its inference algorithm is $o(n^3)$. Moreover, larger numbers of data points will be involved due to the knowledge transfer, so the computation amount will increase significantly. A fast inference algorithm based on the proposed transfer learning GP model will be developed in this paper to accelerate the inference calculation. The time complexity will be reduced to $o(n^2)$.

In the following sections, the basic concepts of the Gaussian process model and the covariance function for power load sequences will be introduced in Section 2. The transfer learning method for power load prediction will be presented in Section 3. And finally, the experiments will be given in Section 4.

2. Preliminaries

2.1. Gaussian process model for prediction

For the prediction problem with data set $\{(x_t, y_t)| t = 1, \dots, n\}$, a regression model can be presented as

$$y_t = f(x_t) + e_t \tag{1}$$

In Eq. (1), y_t is the scalar output of the process y at time t and x_t is the corresponding feature vector with length d. e_t is the independent white Gaussian noise with mean value $\mu_e = 0$ and variance σ_n^2. For time series prediction problems, the x_t can be chosen as: $x_t = [y_{t-1}, y_{t-2}, \dots, y_{t-d}]^T$, and d is called the number of dimensions of the feature space. Let $k(\cdot,\cdot)$ denote the covariance function of the Gaussian process model with $k(x_i, x_j) = k(x_j, x_i)$. The explicit forms of the covariance function will be discussed in the next subsection.

Let y_t be zero mean for simplicity. The joint distribution of the series $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ can be written as

$$\mathbf{y} \sim \mathcal{N}(0, K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I) \tag{2}$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, and the elements of the Gram matrix K are $K_{ij} = k(x_i, x_j)$. The identity matrix I in (2) can be substituted by an auto-covariance matrix if non-white Gaussian noises are concerned. See [27] for more details.

The predictor of the output sequence at time $n+1$ is denoted by $\hat{y}_{n+1}$, and the joint distribution of the predictor of y_{n+1} and the history data $\mathbf{y}$ can be written as

$$\begin{bmatrix} \mathbf{y} \\ \hat{y}_{n+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I & \overline{k}(x_{n+1}, \mathbf{x}) \\ \overline{k}^T (x_{n+1}, \mathbf{x}) & k(x_{n+1}, x_{n+1}) \end{bmatrix}\right) \tag{3}$$

where the covariance vector $\overline{k}$ is defined as

$$\overline{k}(x_{n+1}, \mathbf{x}) = [k(x_{n+1}, x_1), k(x_{n+1}, x_2), \dots, k(x_{n+1}, x_n)]^T$$

The expectation and the variance of the prediction can be obtained by the properties of the joint distribution as

$$\overline{y}_{n+1} = E(\hat{y}_{n+1}) = \overline{k}^T (x_{n+1}, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} \mathbf{y} \tag{4}$$

$$\begin{aligned} \mathrm{Var}(\hat{y}_{n+1}) = {} & k(x_{n+1}, x_{n+1}) \\ & - \overline{k}^T (x_*, \mathbf{x})(K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I)^{-1} \overline{k}(x_*, \mathbf{x}) \end{aligned} \tag{5}$$

In the GP model, the power load prediction $\hat{y}_{n+1}$ is taken as a random variable which is correlated with the previous load values. And the magnitude of the correlations is decided by the covariance function $k(\cdot,\cdot)$.

In this work, we will focus on the one-step-ahead prediction problems. For the k-steps-ahead problems with $k > 1$, one solution is to write the feature vector as

$$x_t = [y_{t-k}, \dots, y_{t-d-k+1}].$$

Another solution is to use the predicted values in the feature vector x_t. The uncertainty propagation problem [28] should be discussed when the predicted values are used.