Load Forecasting in a Smart Grid through Customer Behaviour Learning Using L1-Regularized Continuous Conditional Random Fields

Xishun Wang
School of Computing and Information Technology
University of Wollongong
xw357@uowmail.edu.au

Minjie Zhang
School of Computing and Information Technology
University of Wollongong
minjie@uow.edu.au

Fenghui Ren
School of Computing and Information Technology
University of Wollongong
fren@uow.edu.au

ABSTRACT

Load forecasting plays a critical role in Smart Grid. As there have been various types of customers with different behaviours in a Smart Grid, it would benefit load forecasting if customer behaviours were taken into consideration. This paper proposes a novel load forecasting method that efficiently explores customers' power consumption behaviours through learning. Our method uses L_1-CCRF to initially learn the behaviour of each customer, followed by a hierarchical clustering process to cluster all the customers according to their different behaviour patterns, and then fine-tunes a corresponding L_1-CCRF to predict the load for each customer cluster, and finally, sums all the predicted loads of customer clusters to obtain the load for the whole Smart Grid. The proposed method utilizes L_1-CCRFs to effectively capture the relationships between various customers' loads and a range of outside influential factors. Experiments from different perspectives demonstrate the advantages of our load forecasting method through customer behaviour learning.

Keywords

Load Forecasting; Customer Behaviour; Continuous Conditional Random Fields; Regularization

1. INTRODUCTION

Load forecasting benefits a power grid in supply-demand balance and efficient energy distributions. As it plays a critical role, load forecasting has been widely studied [22, 10]. Representative methods include time series models [7], ARIMA [19] and neural networks [6, 8]. Recently, some novel learning-based methods have been proposed. Srinivasan [21] introduces a group method of data handling neural network for load forecasting. In his method, six categories of consumers are predicted respectively, yet the customer groups are stipulated manually. Amjady et al. [3] uses a bilevel method, which is composed of a feature selection technique and a forecasting engine, to predict the power load of a single micro-grid. Their method has been tested on the load forecasting for a campus. Motamedi et al. [15] combine a multi-input multi-output forecasting engine for joint price and load prediction with data association mining algorithms, through which the relationship of load and price is extracted. This method is applied to a macro scope, regardless of the types of customers.

In Smart Grid, the concept of "customer" has extended to not only general energy consumers, but also interruptible consumers, consumers with storage capacity and even small renewable energy producers. Due to these potential customers' various power consumption behaviours, traditional load forecasting methods, which orient to the whole grid or a specific customer, face challenges to effectively forecast the load of a Smart Grid with such varieties of customers. Even though some researchers [21] manually distinguish different customers' power consumption behaviours, there has been no effective load forecasting method which can automatically take various customers' behaviours into consideration, to the best of our knowledge.

This paper proposes an innovative load forecasting method by learning to explore different behaviours of various customers. The proposed LF-CBL (Load Forecasting through Customer Behaviour Learning) focuses on short-term load forecasting [10], i.e. forecasting the hourly power usages in the future 24 hours. The customer behaviour here specially refers to the customer's power consumption behaviour. In our work, a novel learning method, L_1 regularized Continuous Conditional Random Fields (L_1-CCRF), is proposed and used in LF-CBL. The pipeline of LF-CBL is briefly described as follows. First, a range of features, which may facilitate the load forecasting, are extracted. Based on the rich features, an L_1-CCRF is learned for each customer. L_1-CCRF analyzes customers' behaviours from feature selection and feature weighting. Feature selection determines whether the customer's behaviour is influenced by a certain feature, while feature weighting analyzes how much the power consumption is related to a selected feature. According to the different behaviour patterns in power consumption, all customers are clustered. For each customer cluster, a corresponding L_1-CCRF is fine-tuned and used to predict the load of the cluster. Finally, the load for a Smart Grid is obtained by the sum of the loads of all customer clusters.

The motivation of using L_1-CCRF for LF-CBL is briefly explained on two successive reasons, which are: 1) The reason to introducing CCRF to model the load forecasting prob-

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

lem, and 2) The reason to introducing L_1 norm for CCRF regularization. **To the first reason:** In short-term load forecasting, a sequence of load variables is influenced by the outside observations (time, weather conditions etc.). Besides, partial autocorrelation [9] reveals that there are strong correlations of adjacent target variables. CCRF gains advantages to model the load sequence simultaneously taking the above two factors into account. Moreover, CCRF optimally outputs the load sequence based on the whole observations. However, to our best knowledge, the only work used CCRF for load forecasting was Guo's study [9]. He used CCRF to predict the short-term load of a building, and demonstrated superior performance to state-of-the-art methods. To take the advantages of CCRF and to further explore CCRF in load forecasting, we adopt CCRF to model the load forecasting problem. **To the second reason:** L_1 norm is capable of feature selection and avoiding model over-fitting. Its feature selection property can be utilized in customer behaviour analysis. Therefore, we introduce L_1 norm as the regularization term for CCRF. To fit our problem and to provide the potential for broad applications, in the proposed L_1-CCRF, we change the traditional CCRF in two aspects. First, we extend the definition domain of weights of CCRF from a practical perspective. Second, we provide a new learning method for L_1-CCRF.

The proposed method for load forecasting in a Smart Grid has two major contributions. **1)** Our method provides a new solution to handle the challenges of various customers' different behaviours in load forecasting. To our knowledge, it is the first attempt to introduce learning method to explore customers' power consumption behaviours for load forecasting. Experiment evaluations demonstrate the advantages of utilizing learned customer behaviours in load forecasting for a Smart Grid. **2)** In our method, the proposed L_1-CCRF can be a suggested method for feature selection and prediction to apply to related research domains, because the first exploration of L_1-CCRF in feature selection and load forecasting has produced convincing results demonstrated from our experiments.

The rest of the paper is organized as follows. Section 2 makes a brief introduction to CCRF. Section 3 describes the proposed LF-CBL in details. Section 4 demonstrates and analyzes the proposed LF-CBL through experiments in different perspectives. Section 5 presents the related work. Finally, conclusions are drawn in Section 6.

2. AN INTRODUCTION TO CCRF

In this section, the concept of CCRF is introduced. As C-CRF is originated from Conditional Random Fields (CRF), we first introduce CRF, and then extend CRF to CCRF.

2.1 Conditional Random Fields

The Conditional Random Fields (CRF) [12] was initially proposed for labeling sequence data. The chain-structured CRF, as illustrated in Figure 1, is widely used.

Assume $X = \{x_1, x_2, \cdots, x_m\}$ is the given sequence of observations, and $Y = \{y_1, y_2, \cdots, y_n\}$ is the label sequence to be predicted. CRF defines the conditional probability $P(Y|X)$ in Equation 1.

$$P(Y|X) = \frac{1}{Z(X)} exp(\Psi), \tag{1}$$

where Ψ is the energy function, and $Z(X)$ is the partition function that normalizes $P(Y|X)$. The energy function Ψ is further defined as

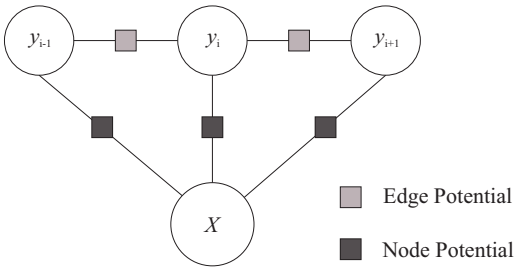

Figure 1: An illustration of a CRF with a chain structure.

$$\Psi = \sum_i \sum_{k=1}^{K_1} \alpha_k f_k(y_i, X) + \sum_{i,j} \sum_{k=1}^{K_2} \beta_k g_k(y_i, y_j, X), \tag{2}$$

where function $f_k(y_i, X)$ is called node potential and function $g_k(y_i, y_j, X)$ is called edge potential, and α_k and β_k are corresponding weight parameters. In the energy function, the node potential captures the associations between inputs and outputs, and the edge potential captures the interactions between related outputs. The partition function $Z(X)$ is defined in Equation 3.

$$Z(X) = \sum_Y exp(\Psi) \tag{3}$$

CRF explicitly defines $P(Y|X)$, which means Y is determined by the whole observation X. Therefore, CRF gains the advantage of considering the whole observed sequence for the output.

2.2 Continuous Conditional Random Fields

The CRF model outputs discrete values, while CCRF extends CRF to be capable of outputing real values. The definition of CCRF differs from CRF in three aspects [17]. 1) The output $Y = \{y_1, y_2, \cdots, y_n\}$ can be a real value sequence. 2) The partition function $Z(X)$ is alternatively defined as:

$$Z(X) = \int_Y exp(\Psi) \tag{4}$$

3) The weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are required to be positive to ensure the partition function is integrable.

To learn a CCRF model, maximum log-likelihood is used to find the fittest weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Given training data $D = \{(X, Y)\}_1^Q$, where Q is the total number of training samples, the log-likelihood $L(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is maximized:

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \text{argmax}_{(\boldsymbol{\alpha}, \boldsymbol{\beta})}(L(\boldsymbol{\alpha}, \boldsymbol{\beta})), \tag{5}$$

where

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{l=1}^{Q} logP(Y_q|X_q) \tag{6}$$

After the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are obtained, inference for a C-CRF is to find the most likely value for Y_k, provided an observed sequence X_k:

$$\hat{Y_k} = \text{argmax}_{Y_k}(P(Y_k|X_k)) \tag{7}$$

In machine learning, regularization has been commonly used in learning process to avoid over-fitting. L_2 norm regularization has been used in [5, 18, 9] in learning CCRF. L_1 norm regularizer is theoretically studied in [16] by Ng, and in practice, the L_1 norm regularizer has gained roughly the same accuracy as the L_2 norm regularizer [13]. Besides, L_1 norm has a favorable property of selecting effective features, which can be utilized to analyze customer behaviours in our research. Therefore, we introduce L_1 norm to regularize the CCRF. However, L_1 norm is not differentiable at zero. Some special methods have been proposed to tackle the learning with L_1 norm regularizer [4, 26]. In our work, we introduce the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) [4] to learn an L_1-CCRF.

3. LOAD FORECASTING THROUGH CUSTOMER BEHAVIOUR LEARNING

LF-CBL focuses on the short-term load forecasting, i.e. to predict the hourly power usages in the next 24 hours. In the following subsections, we first illustrate how to use CCRF to model the load forecasting problem, and then the learning of L_1-CCRFs with the consideration of customer behaviours is described, and finally, the load forecasting using learned L_1-CCRFs is presented.

3.1 Model Design

CCRF discriminatively models the conditional probability $P(Y|X)$. In our research, we use a vector $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ to denote the hourly power usages to be predicted. The observations X are specified with a matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)^T$, where each row $\mathbf{x}_i$ represents the observed D-dimension feature vector for hour i.

In our model, both node potential and edge potential bear a quadratic form. In this scenario, CCRF can be derived into a multivariate Gaussian [18], resulting in convenience in learning and inference. The node potential f_k and edge potential g_k are further discussed below.

The node potential is defined as follows.

$$f_k(y_i, \mathbf{X}) = -(y_i - \mathbf{X}_{i,k})^2 \tag{8}$$

A node potential associates the input feature with the output. For an output y_i, we associate it with the current observed feature $\mathbf{x}_i$. As the number of features in $\mathbf{x}_i$ is D, D node potentials are generated for the current observation. After preprocessing, the feature, without any transformation, is used in the node potential, so we can analyze how much the feature relates to certain customer's load in the learning process.

The edge potential, which captures the interactions between outputs, is defined as follows.

$$g_k(y_i, y_j, \mathbf{X}) = -\delta_k^{(p)}(y_i - y_j)^2 \tag{9}$$

Edge potentials are generated in between every two adjacent outputs. The quadratic edge potential contributes to convenient learning and inference, but results in a weak feature constraint problem [9], which is briefly explained as follows. When CRF works on a binary case, the edge potentials can reflect the true distribution of y_i. In contrast, CCRF copes with continuous problems, and it is not easy to capture the true distribution of y. Thus, it is reasonable to divide the distribution of y into sub-distributions to approximate the true distribution. We follow the work by Guo [9] and introduce the Predictive Clustering Trees (PCTs) [25] to tackle this problem. Δx is introduced to denote the change of adjacent features, and Δy is to denote the change between y_i and y_j. The PCTs provide more sophisticated relationships between Δy and Δx through the indicator function $\delta_k^{(p)}$. The value of an indicator function is determined by its corresponding assertion. When the assertion holds, it takes the value 1, otherwise, it is 0. Figure 2 uses the temperature feature as an instance to illustrate how PCTs work. When the assertion that Δx is small (similar temperatures in the two adjacent hours) holds, $\delta_k^{(1)}$ is true, otherwise, it is 0. Similar processes repeat for $\delta_k^{(2)}$ and $\delta_k^{(3)}$. That is how PCTs determines the indicator function $\delta_k^{(p)}$, and thus provides more interactions between Δy and Δx. In our situation, $P = 3$ is adequate to supply sufficient relationship information between Δy and Δx.

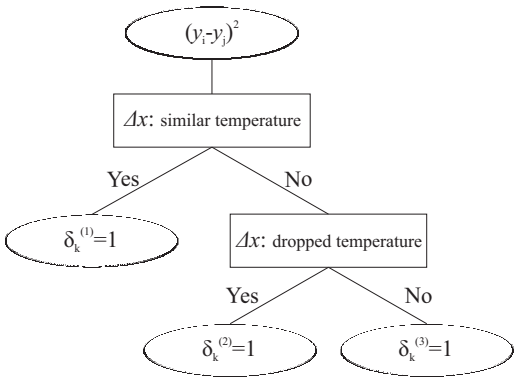

Figure 2: An illustration of how PCTs work with respect to the temperature feature

With the node potential in Equation 8 and edge potential in Equation 9, our CCRF model finally results in the following formula.

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \cdot \\ exp(-\sum_{i=1}^{n}\sum_{k=1}^{D}\alpha_k(y_i - \mathbf{X}_{i,k})^2 - \\ \sum_{i,j}\sum_{k=1}^{D-1}\sum_{p=1}^{P}\delta_k^{(p)}\beta_k^{(p)}(y_i - y_j)^2) \end{aligned} \tag{10}$$

Following Radosavljevic's work [18], the CCRF in Equation 10 can be derived into the following multivariate Gaussian form to facilitate learning and inference.

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \cdot \\ exp(-\frac{1}{2}(\mathbf{y} - \mu(\mathbf{X}))^T\mathbf{\Sigma}^{-1}(\mathbf{y} - \mu(\mathbf{X}))) \end{aligned} \tag{11}$$

In Equation 11, the inverse of the covariance matrix $\mathbf{\Sigma}^{-1}$, is

the sum of two $n \times n$ matrices, further expressed as follows.

$$\begin{aligned}
&\mathbf{\Sigma}^{-1} = 2(\mathbf{M}^1 + \mathbf{M}^2), where \\
&M^1_{i,j} = \begin{cases} \sum_{k=1}^{D} \alpha_k & if \quad i = j \\ 0 & otherwise \end{cases} \\
&M^2_{i,j} = \begin{cases} \sum_{j=1}^{n} \sum_{k=1}^{D-1} \sum_{p=1}^{P} \delta_k^{(p)} \beta_k^{(p)} - \\ \sum_{k=1}^{D-1} \sum_{p=1}^{P} \delta_k^{(p)} \beta_k^{(p)} & if \quad i = j \\ -\sum_{k=1}^{D-1} \sum_{p=1}^{P} \delta_k^{(p)} \beta_k^{(p)} & if \quad i \neq j \end{cases}
\end{aligned} \tag{12}$$

Moreover, the mean $\mu(\mathbf{X})$ is computed by

$$\mu(\mathbf{X}) = \mathbf{\Sigma}\boldsymbol{\theta} \tag{13}$$

Here, $\boldsymbol{\theta}$ is an n-dimension vector, where each element is calculated by

$$\theta_i = 2 \sum_{k=1}^{D} \alpha_k \mathbf{X}_{i,k} \tag{14}$$

Practically, the multivariate Gaussian form, shown in Equation 11, brings convenience to learning and inference in C-CRF, which is introduced in details in the following subsections.

3.2 Learning L_1-CCRFs

Learning L_1-CCRFs for load forecasting in a Smart Grid is introduced in this subsection. A L_1-CCRF is first learned for each customer to analyze the customer's behaviours, then all the customers are clustered based on their behaviour patterns, and finally, for each customer cluster, a corresponding L_1-CCRF is fine-tuned.

3.2.1 Learning a L_1-CCRF for Each Customer

From a practical perspective, we first extend the definition domain of the weights of CCRF, and then introduce OWL-QN to optimize the weights for a CCRF.

In previous CCRF [17, 18], the partition function takes the following form:

$$\begin{aligned}
Z(\mathbf{X}) = \int_{\mathbf{y}} exp(\sum_{i} \sum_{k=1}^{K_1} -\alpha_k (y_i - \mathbf{X}_{i,k})^2 + \\
\sum_{i,j} \sum_{k=1}^{K_2} -\delta_k \beta_k (y_i - y_j)^2)
\end{aligned} \tag{15}$$

For Equation 15, we do not pay much attention to any specific parameters, but focus on the quadratic terms. When the variables $\mathbf{X}$ and $\mathbf{y}$ are defined in infinite domains, both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are required to be positive to ensure that the partition function is integrable. However, in practical use, the observed feature $\mathbf{X}$ is preprocessed to be within a certain domain, and Y is also targeted in a finite range. Thus, the partition function is integrable regardless of the domains of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Therefore, we do not have to constrain $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

To perform feature selection and avoid over-fitting, we introduce L_1 norm to regularize the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. $\boldsymbol{\lambda} =< \boldsymbol{\alpha}, \boldsymbol{\beta} >$ is introduced to compactly represent the weights. The cost function for L_1-CCRF is shown in Equation 16.

$$F(\boldsymbol{\lambda}) = -L(\boldsymbol{\lambda}) + \rho \|\boldsymbol{\lambda}\|_1 \tag{16}$$

In the cost function, the first term is the loss function, which is a negative of log-likelihood of the training set (see Equation 6), while the second term is the L_1 norm of $\boldsymbol{\lambda}$, used as a regularization term. The parameter ρ compromises the loss and the regularization term.

As the L_1 norm is non-differentiable at zero, we seek to OWL-QN algorithm to minimize the cost function. We first derive the gradient of α_k and $\beta_k^{(p)}$ in the loss function. With the gradients of α_k and $\beta_k^{(p)}$, the corresponding pseudo-gradients [4] of Equation 16 is obtained. Then we use OWL-QN algorithm to minimize $F(\boldsymbol{\lambda})$, resulting in the optimal weights for L_1-CCRF.

3.2.2 Clustering Customers

As a L_1-CCRF has been learned for each customer, we can utilize the obtained weight vector $\boldsymbol{\lambda}$ to analyze each customer's behaviours and accordingly cluster customers. In the learned weights $\boldsymbol{\lambda} =< \boldsymbol{\alpha}, \boldsymbol{\beta} >$ for a certain customer, each α_k reveals how much a feature influences the power usage of the customer, and each β_k reveals how much the adjacent feature change influences the hourly usage change. As we use L_1 norm to regularize CCRF, the weights of unrelated features have been pushed to zero, and the rest features with non-zero weights reflect how much the load is influenced by the related features. As the learned weights meaningfully relate to the customer power consumption behaviours, we use them to cluster the various customers.

We propose a hierarchically clustering method to cluster the customers with respect to the weights of L_1-CCRF, as illustrated in Figure 3. First, each weight in $\boldsymbol{\lambda}$ is binarized.

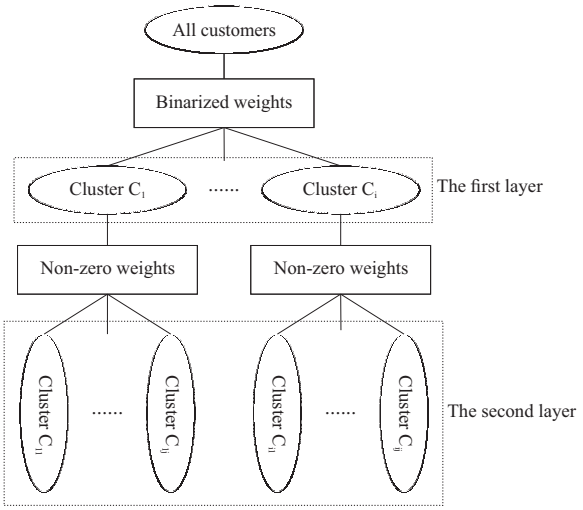

Figure 3: An illustration of clustering customers

To be specific, all the non-zero values are converted into 1, and the zero values remain. In the first layer, all customers are clustered according to the binarized weights. Customers who share the same binarized weights fall into the same cluster. In the second layer, for the each cluster $C_k, k = 1, \cdots, i$, the corresponding non-zero weights are used to further cluster the customers. K-means, with an Euclidean distance criterion Δ, is utilized in the second layer clustering. Δ is a critical parameter, which determines the number of final clusters and influences the final precision of load forecasting. This parameter is further analyzed in the Experiment part (see Subsection 4.4).

A two-layer clustering tree is obtained (see Figure 3), and clusters in each layer indicate clear physical meanings. In

the first layer clustered by the binarized weights, the features with weights "1" are related to the customers' power usages. Thus, customers in the same cluster are influenced by the same range of features. In practice, customers in this layer can be certain customer genres such as wind producers, householders influenced by temperatures, householders regardless of temperatures. In the second layer, each cluster C_k is further divided into smaller clusters according to the non-zero weights, which indicate how much each feature influences the customers' power usages. After a second clustering, customers in each smaller cluster C_{kl} share similar sensitivity to the range of features. Take the office buildings for an example. Office buildings in one cluster may adaptively adjust their power usages with respect to temperatures, while buildings in another cluster are less sensitive to the influence of temperatures.

3.2.3 *Fine-tuning L_1-CCRFs*

After obtained the clustering tree, we fine-tune a L_1-CCRF for each cluster. In Subsubsection 3.2.1, a L_1-CCRF has been learned for each customer. For a customer cluster, one learned L_1-CCRF is randomly selected and fine-tuned. Fine-tuning a L_1-CCRF for each cluster gains two advantages. 1) Increasing prediction precision: for an individual customer, his/her behaviour is chaotic, thus the short-term load is hard to predict. On the contrary, the customers' usage data seem to be "smoothed" in a customer cluster. 2) Reducing computation cost: for a cluster with N customers, only one fine-tuned L_1-CCRF is needed in the end.

Fine-tuning a L_1-CCRF for a cluster is quite straightforward. For the selected L_1-CCRF, the input feature $\mathbf{X}_q$ remains, while for the ground-truth, each element y_i in $\mathbf{y}_q$ becomes the sum of all the customers' power usages in each hour. Then training process repeats with the input features and the new ground-truth. Fine-tuning process results in a quick convergence, because the weights in the selected L_1-CCRF are close to the optimal values of the final L_1-CCRF.

3.3 Load Forecasting

With the learned L_1-CCRF for each customer cluster, the hourly load can be predicted. Aggregating the predicted load for each cluster, the final load for the whole Smart Grid can be obtained.

To predict the load for each customer cluster, we find the most likely $\mathbf{y}$ given the observed feature $\mathbf{X}$, as formulated in Equation 7. Benefiting from the multivariate Gaussian form, the inference becomes quite tractable. To maximize $P(\mathbf{y}|\mathbf{X})$ in the multivariate Gaussian (see Equation 11), we simply make $\mathbf{y}$ equal to $\mu(\mathbf{X})$,

$$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}}(P(\mathbf{y}|\mathbf{X})) = \mu(\mathbf{X}) = \mathbf{\Sigma}\boldsymbol{\theta} \tag{17}$$

The above equation shows how the load for each cluster is predicted. Assuming there are N customer clusters formed in a Smart Grid, adding up each cluster's predicted load $\hat{\mathbf{y}}_i$ in element wise, the final load $\mathbf{y}_{SG}$ of the whole grid is obtained by the following equation.

$$\mathbf{y}_{SG} = \sum_{1}^{N} \hat{\mathbf{y}}_i \tag{18}$$

4. EXPERIMENT AND ANALYSIS

Three experiments were conducted from different perspectives to evaluate LF-CBL. **Experiment 1:** Evaluation of customer behaviour learning. Two other prediction methods based on L_1-CCRF but without the consideration of customer behaviours were constructed. We compared LF-CBL with the two methods to demonstrate the advantage of customer behaviour learning. **Experiment 2:** Comparison with Linear Regression with LASSO. Linear Regression with LASSO was introduced to learn customer behaviours to predict the load in a Smart Grid. we compared the performance of LF-CBL with the new method on customer behaviour learning and load forecasting precision. **Experiment 3:** Analysis of the clustering criterion Δ. As the clustering criterion Δ determined the granularity of final customer clusters and affected the final load forecasting result, we tried different values for Δ to find better one for practical use of LF-CBL.

Our experiments were conducted on the platform of Power Trading Agent Competition (Power TAC) [11]. Power TAC has drawn wide attentions and has become a benchmark in the Smart Grid research community. Power TAC simulates a variety of customers with various behaviours in a Smart Grid. Moreover, there are rich features, including real-world weather conditions and real-time market status. Besides, the Power TAC server supplies rich logs of customers' hourly power usages, which are regarded as the ground-truth to evaluate the proposed LF-CBL.

4.1 Experiment Settings

Representative features, which may relate to the customer behaviours, are studied in our work. These features include temporal features, weather features and market features. Table 1 lists the contents of the three features, and the indexes provide convenience for further discussions.

Table 1: Features used in LF-CBL

Feature	Content	Index
Temporal feature	hour of a day	t_1
	day of a week	t_2
Weather feature	temperature	w_1
	wind strength	w_2
	wind direction	w_3
	cloudiness	w_4
Market feature	lowest price	m_1
	average price	m_2

We configured the Power TAC server and weather data server, and utilized Power TAC games to generate training and test data. Training data were generated by six games for the year 2009 (with real-world weather data). Test data were from six games for the year 2010. The logged customers' usages were regarded as the ground-truths of loads to be predicted. To induce rich features, three broker models, TacTex [24], cwiBroker [14] and our own broker model. were introduced to compete in the games. In Power TAC, 40 customers, each customer with a certain population, were simulated. For two customers, each had population up to tens of thousands. We split these two customers into customers with population of 100 and rendered them with some

random behaviours in the log data, such as power usage decreasing for going out at night, or load increasing for having a party at home. In the end, 538 customers were obtained. These customers were manageable for experiments and sufficient to analyze the proposed LF-CBL.

The configurations in LF-CBL are described in details. L_1-CCRF modeled 24 hours power usages under the influences of 24 hourly features. For the features in each hour, 8 node potentials were generated. There were 23 intervals between every two adjacent hours in a day. Edge potentials were generated in every intervals, and PCTs were applied to weather and market features, but not temporal features. Thus, in each interval, 20 edge potentials were generated. To ensure the L_1-CCRF for each customer was converged, we set 50 iterations for OWL-QN. For the fine-tuned L_1-CCRF for each customer cluster, 10 iterations were sufficient to ensure convergence. L_1-CCRF was implemented in Matlab, and OWL-QN was implemented based on minFunc [20].

In the proposed LF-CBL, there are only two parameters. One parameter is the ρ in the cost function of L_1-CCRF in Equation 16, and the other is the criterion Δ in customer clustering. In learning a L_1-CCRF for each customer, ρ was determined by cross-validation process. For the final L_1-CCRF for each customer cluster, similar process repeated. Our cross-validation process extracted the daily data of every four days. The extracted data were used for validation, and the remained data were for training. The parameter Δ determines the granularity of the customer clusters, which has a strong influence to the performance of LF-CBL. Therefore, we will further analyze this parameter in Experiment 3. For Experiment 1 and 2, we set $\Delta = 0.05$. In the following three subsections, we report and analyze the results of the three experiments, respectively.

4.2 Experiment 1: Evaluation of Customer Behaviour Learning

We evaluated the contribution of customer behaviour learning in load forecasting by comparing LF-CBL with other two methods, which both used L_1-CCRF but without the considerations of customer behaviours. In method 1, one L_1-CCRF was trained for each customer. The load of a Smart Grid was a sum of all individual customer's loads predicted by L_1-CCRFs. We named this method as LF-S. In method 2, one L_1-CCRF was trained towards the whole Smart Grid, regardless of any individual customer's behaviours. We used LF-W to denote this method. Mean Absolute Percentage Error (MAPE) for each hour of the three methods are illustrated in Figure 4. We can see that LF-S performs slightly better than LF-W, and LF-CBL gains patent advantages over the other two methods.

LF-S used L_1-CCRF to predict the load for each customer. However, some behaviours of an individual customer were random and impossible to predict. For instance, some household customers may occasionally go out for parties on any weekday. Thus, the weakness of LF-S came from many accumulated errors resulted from the random customer behaviours. For LF-W, it utilized one L_1-CCRF to predict the load for all the customers, but a single L_1-CCRF failed to handle the various customers with different behaviours. In the end, the final load prediction result of LF-W was not satisfactory.

In contrast, LF-CBL performed well because it overcame the disadvantages in the above two methods. LF-CBL used

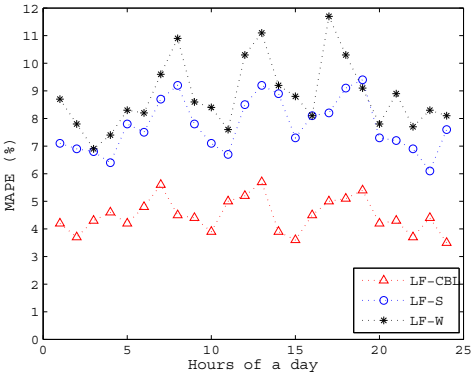

Figure 4: Performance comparison of LF-S, LF-W and LF-CBL

L_1-CCRF to first analyze customer behaviours, then customers were clustered and customers shared the similar behaviours gathered in one cluster. In a cluster of customers, the chaotic random behaviours were averaged, resulting in "smooth" power usage data. With a fine-tuned L_1-CCRF for each customer cluster, the final prediction result was much better than that of the other two methods.

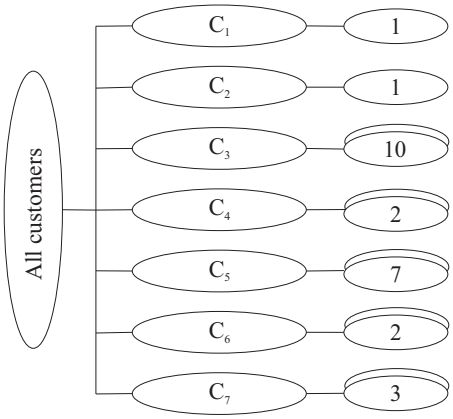

Figure 5: Clustering tree of LF-CBL

Figure 5 shows the clustering tree of LF-CBL. In the clustering tree, 7 clusters $(C_1, \cdots, C_7)$ are obtained in the first layer. In the second layer, the digit on each ellipse indicates the number of clusters. 26 clusters are formed in the end.

In the first layer of the clustering tree, based on the binarized feature weights, we can determine if customer behaviours in one cluster is influenced by a certain feature. Table 2 uses a binary matrix to show the relationships between the clusters and features. In Table 2, "1" indicates that the cluster is influenced by this feature, while "0" means the feature is not related to this cluster. The clusters in the first layer show clear physical meanings. For instance, customers in C_1 are wind power producers, and their behaviours are influenced by the related wind features. Customers in C_2 are solar energy producers whose power usages are affected by time, temperature and cloudiness. Observing the customers in each cluster, we can see that they generally belong to the same category of customers. For example, customers in

C_4 are thermal storage customers, and customers in C_5 are householders and office users.

Table 2: Cluster and feature relation matrix

	t_1	t_2	w_1	w_2	w_3	w_4	m_1	m_2
C_1	0	0	0	1	1	0	0	0
C_2	1	0	1	0	0	1	0	0
C_3	1	1	0	0	0	0	0	1
C_4	1	0	0	0	0	0	1	1
C_5	1	1	1	0	0	0	0	0
C_6	1	1	1	0	0	0	1	0
C_7	1	0	1	0	0	0	1	0

In the second layer, customers are further clustered, resulting in 26 smaller clusters. Take C_5 for example. C_5 is further clustered into 7 smaller clusters based on the non-zero weights. In each customer cluster C_{5j}, customers show similar responses to the influences of outside features. For the 26 customer clusters, accordingly, 26 corresponding L_1-CCRFs are fine-tuned. In the end, 26 L_1-CCRFs are maintained for load forecasting for a whole Smart Grid with many different types of customers. Therefore, our LF-CBL results in a reasonable computation cost.

4.3 Experiment 2: Comparison with Linear Regression with LASSO

We constructed Linear Regressions with LASSO [23] to forecast load with the consideration of customer behaviours. This method followed the framework of LF-CBL, but a Linear Regression with LASSO was used to instead a L_1-CCRF. We named the Load Forecasting using Linear Regressions with LASSO as LF-LRL. The same criterion $\Delta = 0.05$ in customer clustering was applied to both methods.

Linear Regression with LASSO is briefly described as follows. A Linear Regression uses a linear form to estimate load sequence from the input features, which is shown in Equation 19 in a vectorized form.

$$\hbar(\mathbf{X}) = \mathbf{aX} + \mathbf{b} \tag{19}$$

$\boldsymbol{\lambda} =< \mathbf{a}, \mathbf{b} >$ is used to denote the weight vector. Linear Regression with LASSO minimizes the following cost function to learn the weights.

$$F(\boldsymbol{\lambda}) = \frac{1}{2}\sum_{q=1}^{Q}(\mathbf{y}_q - \hbar(\mathbf{X}_q))^2 + \rho\|\boldsymbol{\lambda}\|_1 \tag{20}$$

In the above equation, the first term is a mean square error loss function, and the second term is L_1 norm regularizer. Weights in $\boldsymbol{\lambda}$ are learned through minimizing the Equation 20 using LASSO algorithm [23].

LF-LRL and LF-CBL were compared in customer behaviour analysis and final prediction precisions. We compared the behaviour analysis from the clustering trees formed in LF-CBL (refer to Figure 5) and LF-LRL (similar to the clustering tree of LF-CBL, not shown due to page limit). In the first layer, there were 7 clusters in LF-LRL, same as that in LF-CBL. We also constructed relationship matrix between clusters and features for LF-LRL, and there was only one row in LF-LRL different from that in LF-CBL. Besides, the distributions of customers in customer clusters in the first layer were also similar in both methods. In the second layer, the non-zero weight vectors were quite different, resulting in different final customer clusters and customer distributions in clusters. Through the comparisons in customer behaviour analysis of the two methods, LF-LRL and LF-CBL performed similar in feature selection, but the feature weights were different, as a result, the result of load forecasting would be different, as shown in Figure 6.

The MAPE for each hours of LF-LRL and LF-CBL is depicted in Figure 6. We can see that the MAPE of LF-CBL is

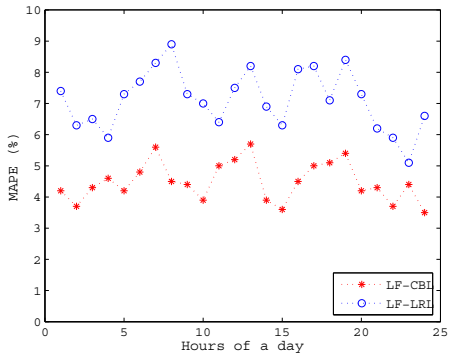

Figure 6: Performance comparison of LF-LRL and LF-CBL

better than that of LF-LRL. The total MAPE of LF-CBL is 4.51%, while that of LF-LRL is 7.26%. That LF-CBL gains better performance than LF-LRL in load forecasting indicates that the L_1-CCRF has advantage over Linear Regression. L_1-CCRF not only models the relationship between the feature and the load using node potentials, but also models the interactions of load changes and feature changes using edge potentials. In contrast, Linear Regression can only associate the features with the load.

With respect to the clustering tree of LF-CBL, we further analyzed LF-CBL and LF-LRL on load forecasting in two typical customer clusters. One customer cluster contained householders sensitive to temperatures. The overall MAPE of this cluster using LF-CBL was 4.62%, and that of LF-LRL was 8.57%. LF-CBL showed patent advantage over LF-LRL. For the householders sensitive to temperature, their loads had strong interactions between adjacent hours, and their adjacent hourly load changes were influenced by the changes of temperatures. In LF-CBL, L_1-CCRF successfully modeled the above factors, while in LF-LRL, the Linear Regression could not consider those factors. Therefore, LF-CBL got better prediction precision than LF-LRL. Another customer cluster was a group of wind power producers. The overall MAPE using LF-CBL and LF-LRL were 4.79% and 4.82%, respectively. These customers were only affected by the weather conditions. Thus it was not necessary to model the interactions of power usages in adjacent hours. As expected, LF-CBL and LF-LRL achieved similar performances in load forecasting for wind power producers. When there were no interactions of power usages in adjacent hours, L_1-CCRF degenerated to regression.

4.4 Experiment 3: Analysis of the Clustering Criterion Δ

In this experiment, an important parameter Δ was set to different values, resulting in different granularity of the final customer clusters. We observed the influence of the granularity of customer cluster to the final load forecasting precision, and thus suggest a reasonable range of Δ for practical use.

Four different values were set for Δ, which were: 0.025, 0.05, 0.75, 0.10. In the learning process, after a L_1-CCRF was learned for each customer, different values of Δ were applied to the clustering process and four clustering trees were formed, respectively. For each clustering tree, fine-tuning process was applied to the L_1-CCRF for each customer cluster. In the end, the final load was predicted using Equation 17 and 18. Table 3 summarizes the number of clusters and the overall MAPE under different Δ values.

Table 3: The influence of Δ on the MAPE and the number of clusters

Δ	Number of clusters	overall MAPE(%)
0.025	65	5.96
0.05	26	4.51
0.075	24	4.72
0.10	17	5.73

In Table 3, when Δ was set to 0.025, 65 customer clusters were formed, and the total MAPE was 5.96%. When we set $\Delta = 0.05$, 26 customer clusters were obtained, and the total MAPE became better. Comparing the above two settings, we could see that a small Δ resulted in fine granularity of customer clusters. When the customer cluster was too small, the "smoothness" of load data was compromised. That is why small Δ led to a less satisfactory prediction result. When Δ was set to 0.075, 24 customer clusters were formed, and the total MAPE was 4.72%. This indicated that when Δ changed from 0.05 to 0.075, the performance of LF-CBL did not change much. Besides, as the number of clusters also determined the required final L_1-CCRFs, the range from 0.05 to 0.075 resulted in an acceptable computation cost. When Δ was 0.10, the prediction precision declined. From the above analysis, we suggest that the reasonable range of Δ is $[0.05, 0.075]$. In this range, LF-CBL demonstrated a reasonable computation cost and competitive performance.

5. RELATED WORK

Load forecasting towards a whole grid or a specific customer has been deeply studied [22, 10], but there is little work considering different customer behaviours. Srinivasan [21] manually divided different customers in a power grid into six groups, and he introduced a group method of data handling (GMDH) neural network for load forecasting. Our method first introduces a learning method to explore different customer behaviours. Then customers are clustered based on the learned customer behaviour patterns. As we take the learned customer behaviours into account, our method is more advantageous than the existing methods in load forecasting in Smart Grid.

Some novel learning methods have also been introduced for load forecasting. Amin-Naseri and Soroush [2] used supervised and unsupervised learning to predict the dailiy peak load. Ali et al. [1] combined neural network, time series models and ANOVA for load forecasting. Recently, Guo [9] used Continuous Conditional Random Fields (CCRF) to forecast the short-term power and gas usages in a building. His work demonstrated the advantages of CCRF and achieved superior performances in load forecasting. Our work took a further step on the study of CCRF in load forecasting in a whole Smart Grid. We extended the definition domain of weights for CCRF and introduced L_1 norm as a regularization term. We creatively used L_1-CCRF to perform customer behaviour analysis and short-term load forecasting, resulting in a satisfactory result.

6. CONCLUSIONS

This paper proposed a load forecasting method through customer behaviour learning (LF-CBL). Our method introduced L_1-CCRF to analyze the customers' behaviours by using the weights of features in L_1-CCRF to reflect customer behaviours. The proposed method was evaluated and analyzed by experiments from three different perspectives, and experimental results demonstrated the advantages of the proposed LF-CBL against other baseline methods. With the consideration of learned customer behaviours, LF-CBL achieved a good performance in load forecasting. In addition, evaluation results also indicated that the proposed L_1-CCRF is effective in feature selection. Thus, L_1-CCRF can also be used in other related research domains.

Acknowledgments

The authors would like to thank Tao Qin for his useful discussions on CCRF. Moreover, we thank the organizers of Power TAC, which supplies a simulation of real-world Smart Grid market. Our experiments are done based on that platform. This work is supported by a Discovery Project (DP140100974) from Australian Research Council.

REFERENCES

[1] A. Ali, S. Ghaderi, and S. Sohrabkhani. Forecasting electrical consumption by integration of neural network, time series and anova. *Applied Mathematics and Computation*, 186(2):1753–1761, 2007.

[2] M. Amin-Naseri and A. Soroush. Combined use of unsupervised and supervised learning for daily peak load forecasting. *Energy Conversion and Management*, 49(6):1302–1308, 2008.

[3] N. Amjady, Keynia F, and H. Zareipour. Short-term load forecast of microgrids by a new bilevel prediction strategy. *IEEE Transactions on Smart Grid*, 1(3):286–294, 2010.

[4] G. Andrew and J. Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[5] T. Baltrusaitis, N. Banda, and P. Robinson. Dimensional affect recognition using continuous conditional random fields. In *Automatic Face and Gesture Recognition*, pages 1–8. IEEE, 2013.

[6] TWS Chow and C. Leung. Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, 11(4):1736–1742, 1996.

[7] J. Fan and J. McDonald. A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Transactions on Power Systems*, 9(2):988–994, 1994.

[8] R. Gareta, L. M Romeo, and A. Gil. Forecasting of electricity prices with neural networks. *Energy Conversion and Management*, 47(13):1770–1778, 2006.

[9] H. Guo. Accelerated continuous conditional random fields for load forecasting. *to appear in IEEE Transactions on Knowledge & Data Engineering*, 2015.

[10] L. Hernandez, C. Baladron, J. Aguiar, B. Carro, Antonio J Sanchez-Esguevillas, J. Lloret, and J. Massana. a survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys & Tutorials*, 16(3):1460–1495, 2014.

[11] W. Ketter, J. Collins, P. Reddy, and M. Weerdt. The 2014 power trading agent competition. *ERIM Report Series Reference No. ERS-2014-004-LIS*, 2014.

[12] J. Lafferty. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*. Morgan Kaufmann, 2001.

[13] T. Lavergne, O. Cappé, and F. Yvon. Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513. Association for Computational Linguistics, 2010.

[14] B. Liefers, J. Hoogland, and P. La. A successful broker agent for power tac. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 99–113. Springer, 2014.

[15] A. Motamedi, H. Zareipour, and W. Rosehart. Electricity price and demand forecasting in smart grids. *IEEE Transactions on Smart Grid*, 3(2):664–674, 2012.

[16] A. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[17] T. Qin, T. Liu, X. Zhang, D. Wang, and H. Li. Global ranking using continuous conditional random fields. In *Advances in neural information processing systems*, pages 1281–1288, 2009.

[18] V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous conditional random fields for regression in remote sensing. In *ECAI*, pages 809–814, 2010.

[19] S. Saab, E. Badr, and G. Nasr. Univariate modeling and forecasting of energy consumption: the case of electricity in lebanon. *Energy*, 26(1):1–14, 2001.

[20] M. Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. http://www.cs.ubc.ca/ schmidt-m/Software/minFunc.html, 2005.

[21] D. Srinivasan. Energy demand prediction using gmdh networks. *Neurocomputing*, 72(1):625–629, 2008.

[22] L. Suganthi and A. Samuel. Energy models for demand forecastin–a review. *Renewable and Sustainable Energy Reviews*, 16(2):1223–1240, 2012.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[24] D. Urieli and P. Stone. Tactex'13: a champion adaptive power trading agent. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1447–1448, 2014.

[25] D. Vail, M. M. Veloso, and J. Lafferty. Conditional random fields for activity recognition. In *Proceedings of international conference on Autonomous agents and multiagent systems*, page 235. ACM, 2007.

[26] J. Yu, S. Vishwanathan, S. Günter, and N. Schraudolph. A quasi-newton approach to nonsmooth convex optimization problems in machine learning. *The Journal of Machine Learning Research*, 11:1145–1200, 2010.