Accelerated Continuous Conditional Random Fields For Load Forecasting

Hongyu Guo

Abstract—Increasingly, aiming to contain their rapidly growing energy expenditures, commercial buildings are equipped to respond to utility's demand and price signals. Such smart energy consumption, however, heavily relies on accurate short-term energy load forecasting, such as hourly predictions for the next n ($n \geq 2$) hours. To attain sufficient accuracy for these predictions, it is important to exploit the relationships among the n estimated outputs. This paper treats such multi-steps ahead regression task as a sequence labeling (regression) problem, and adopts the continuous conditional random fields (CCRF) to explicitly model these interconnected outputs. In particular, we improve the CCRF's computation complexity and predictive accuracy with two novel strategies. First, we employ two tridiagonal matrix computation techniques to significantly speed up the CCRF's training and inference. These techniques tackle the cubic computational cost required by the matrix inversion calculations in the training and inference of the CCRF, resulting in linear complexity for these matrix operations. Second, we address the CCRF's weak feature constraint problem with a novel multi-target edge function, thus boosting the CCRF's predictive performance. The proposed multi-target feature is able to convert the relationship of related outputs with continuous values into a set of "sub-relationships", each providing more specific feature constraints for the interplays of the related outputs. We applied the proposed approach to two real-world energy load prediction systems: one for electricity demand and another for gas usage. Our experimental results show that the proposed strategy can meaningfully reduce the predictive error for the two systems, in terms of mean absolute percentage error and root mean square error, when compared with three benchmarking methods. Promisingly, the relative error reduction achieved by our CCRF model was up to 50 percent.

Index Terms—Continuous conditional random fields, energy demand forecast, multi-target decision trees, tridiagonal matrix

1 Introduction

Enabling commercial buildings to use energy in a flexible way, such as reducing usage during consumption peak hours, is of crucial importance, not only for preventing the disruptions of the utility grid, but also for containing buildings' rapidly growing energy cost. Such smart energy consumption, however, requires accurate short-term load predictions.

To attain sufficient accuracy for these load predictions, it is important to exploit the relationships among the multiple estimated outputs. This is because these predicted outcomes are typically correlated. For instance, knowing the current hour's overall energy usage will help estimate the next hour's energy demand. Aiming at making good use of these interrelationships, this paper deploys the conditional random fields (CRF) [1], a sequential labeling method. More specifically, we adopt the continuous conditional random fields (CCRF) [2] to explicitly model the interplays between the multiple outputs. As depicted on the left subfigure in Fig. 1, the CCRF approach intuitively integrates two layers. The first layer consists of variable (node) features (filled squares in Fig. 1), and aims at the prior knowledge for the multiple outputs. The second layer employs edge potential features (unfilled squares in Fig. 1) to implicitly model the interplays of the interconnected outputs, aiming at improving the predictions from the first layer. Consequently, the proposed method makes predictions not only basing on observed features, but also considering the estimated values of related outputs, thus improving the overall predictive accuracy.

Importantly, we addressed two challenge problems in the CCRF: the weak feature constraint for continuous features and the expensive computation cost in the training and inference. The weak feature constraint limits the CCRF's capability to mediate the relationship between two target variables differently, and the expensive computation is caused by a repeated matrix inversion during CCRF's training and inference. In detail, first, we cope with the weak feature constraint problem in the CCRF with a novel edge function, thus boosting its predictive performance. Such weak constraint issue arises because CCRF takes aim at target outputs with continuous values. Specifically, CRF's function constraints are weak for edge features with continuous values, compared to that of binary features, because of CRF's linear parameterization characteristics [3], [4]. That is, for a binary feature, knowing the mean is equivalent to knowing its full probability distribution. On the contrary, knowing the mean may not tell too much about the distribution of a continuous variable. As illustrated on the top of Fig. 2, the mean of the distribution depicted by the red curve does not distinguish the three subsumed sub-distributions, depicted by the blue, brown, and green curves, respectively. Since CRF strategies are devised to form models satisfying certain feature constraints [4], such weak feature constraints will limit the resultant CRF's predictive performance. Moreover, typical approaches of dividing continuous values into

• *The author is with the National Research Council of Canada, 1200 Montreal Road, Ottawa, ON, K1A 0R6, Canada. E-mail: hongyu.guo@nrc-cnrc.gc.ca.*

Manuscript received 22 July 2013; revised 26 June 2014; accepted 27 Jan. 2015. Date of publication 10 Feb. 2015; date of current version 2 July 2015. Recommended for acceptance by K. Chakrabarti.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2399311

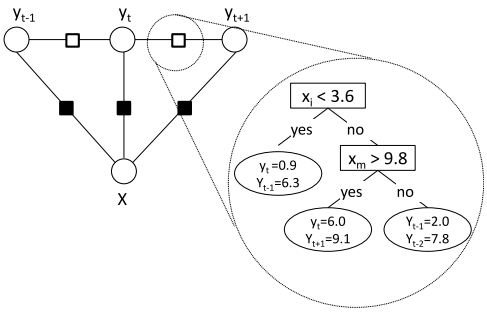


Fig. 1. A chain-structured continuous CRF with PCTs trees (right subfigure) as edge potential functions (unfilled squares). Here, the filled squares are the node features.

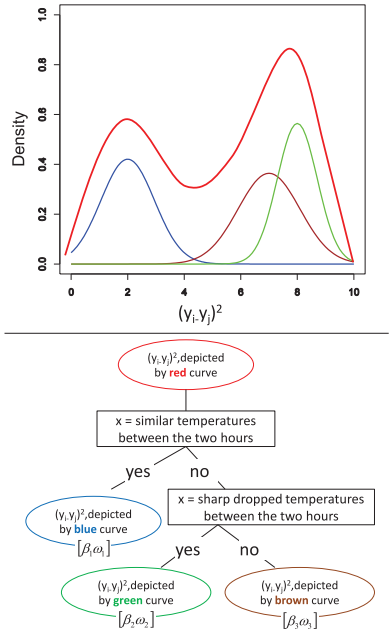


Fig. 2. Top subfigure: distribution of $(y_i - y_j)^2$ (red curve), and the subsumed three sub-distributions (blue, brown, and green curves); bottom subfigure: the PCT tree that shows what X values were used to convert the red curve into the three sub regions.

"bins" cannot be applied to our CCRF method here because for multi-steps ahead tasks, one has to be able to simultaneously "bin" multiple *correlated* target variables that are unknown in inference time. That is, we are interested in "binning" the relationship of the target variables. To address the above concern for the CCRF model, we employ a multi-target function, namely the Predictive Clustering Trees (PCTs) strategy [5], as the CCRF's edge feature. The PCTs method first partitions instances with similar values for multiple related target variables, only based on their shared observation features, into disjoint regions. Next, it models a separate relationship among these target variables in each smaller region. In other words, the PCTs convert the relationship of the related target variables into a set of subrelationships, each containing more specific constraints for the related target variables. As a result, it enables the CCRF to better capture the correlations between related outputs, thus boosting the CCRF's predictive performance.

Second, we address the cubic computation complexity needed for the matrix calculations in the training and inference of the CCRF. Through deploying two tridiagonal matrix techniques, we enable linear computational cost for these matrix computations. The CCRF's expensive training and inference cost is caused by a matrix inversion operation. As will be discussed in detail in Section 4.2, aiming at constructing a computationally tractable CRF, the CCRF model implements its training and inference through matrix computation, of which a matrix inversion operation is involved. Consequently, for an application with n target variables, finding the inverse of the matrix needs computation complexity of $\mathcal{O}(n^3)$. This expensive matrix calculation will result in poor scalability for the CCRF when the n is large. To address this issue, we introduce two fast computation techniques. First, we adopt a linear computational cost strategy presented by Rybicki and Hummer [6] for our matrix computation in the training. This method enables computing the non-zero entries in the inverse of a *tridiagonal* matrix quickly, without the need to construct the full inverse of the matrix. Second, we apply the Thomas algorithm [7] to implement the inference calculation for the CCRF, again, avoiding the expensive matrix inversion operation. As a result, the cubic computational cost matrix calculations in the training and inference of the CCRF can be executed in linear time $\mathcal{O}(n)$.

We applied the proposed method to two real-world energy load forecasting systems: one for gas which is used to warm buildings in winter, and another for electricity for building cooling in summer. Also, we compared our approach with three benchmarking strategies. Our experimental results show that the proposed method can significantly reduce the predictive error, in terms of mean absolute percentage error (MAPE) and root mean square error (RMSE), for the two energy systems, when compared with the three baseline algorithms.

2 Mathematical Background

2.1 Conditional Random Fields

Conditional random fields are undirected graphical models that define the conditional probability of the label sequence $Y = (y_1, y_2, \ldots, y_n)$, given a sequence of observations $X = (x_1, x_2, \ldots, x_r)$. That is, the discriminative strategy aims to model $P(Y|X)$. Specifically, benefiting from the Hammersley-Clifford theorem, the conditional probability can be formally written as:

$$P(Y|X) = \frac{1}{Z(X)} \prod_{c \in C} \Phi(y_c, x_c), \tag{1}$$

where C is the set of cliques[1] in the graph, Φ is a potential function defined on the cliques, and $Z(x)$ is the normalizing partition function which guarantees that the distribution sums to one.

One of the most popular CRF is the linear chain CRF (depicted on the left of Fig. 1), which imposes a first-order Markov assumption between labels Y. This assumption allows the CRF to be computed efficiently via dynamic programming. In addition, the clique potentials Φ in the linear chain CRF are often expressed in an exponential form, so that the formula results in a maximum entropy model.

1. A clique is a fully connected subgraph.

Formally, the linear-chain CRF is defined as a convenient log-linear form:

$$P(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{n} exp(\upsilon^T \cdot f(t, y_{t-1}, y_t, X)), \quad (2)$$

$$\text{where,} \quad Z(X) = \sum_{Y} \prod_{t=1}^{n} exp(\upsilon^T \cdot f(t, y_{t-1}, y_t, X)).$$

Here, $f(t, y_{t-1}, y_t, X)$ is a set of potential feature functions which aim to capture useful domain information; υ is a set of weights, which are parameters to be determined when learning the model; and y_{t-1} and y_t are the label assignments of a pair of adjacent nodes in the graph.

2.2 Continuous Conditional Random Fields

The CRF strategy is originally introduced to cope with discrete outputs in labeling sequence data. To deal with regression problems, continuous conditional random fields has recently been presented by Qin et al. [2], aiming at document ranking. In CCRF, Equation (2) has the following form:

$$P(Y|X) = \frac{1}{Z(X, \alpha, \beta)} \cdot exp\left(\sum_{1}^{n} H(\alpha, y_i, X) + \sum_{i \sim j} G(\beta, y_i, y_j, X) \right), \quad (3)$$

where $i \sim j$ means y_i and y_j are related, and

$$Z(X, \alpha, \beta) = \int_y exp\left(\sum_{1}^{n} H(\alpha, y_i, X) + \sum_{i \sim j} G(\beta, y_i, y_j, X) \right) dy. \quad (4)$$

Here, potential feature functions $H(y_i, X)$ and $G(y_i, y_j, X)$ intend to capture the interplays between inputs and outputs, and the relationships among related outputs, respectively. For descriptive purpose, we denote these potential functions as *variable (node) feature* and *edge feature*, respectively. Here, α and β represent the weights for these feature functions. Typically, the learning of the CCRF is to find weights α and β such that conditional log-likelihood of the training data, i.e., $L(\alpha, \beta)$, is maximized, given training data $D = \{(X, Y)\}_1^L$ (L is the number of sample points in D):

$$(\widehat{\alpha}, \widehat{\beta}) = \underset{\alpha, \beta}{\text{argmax}}(L(\alpha, \beta)),$$
$$\text{where} \quad \mathrm{L}(\alpha, \beta) = \sum_{l=1}^{L} log\ P(Y_l|X_l). \quad (5)$$

After learning, the inference is commonly carried out through finding the most likely values for the $P(Y_l)$ vector, provided observation X_l:

$$\widehat{Y_l} = \underset{Y_l}{\text{argmax}}(P(Y_l|X_l)). \quad (6)$$

Promisingly, as shown by Radosavljevic et al. [8], if the potential feature functions in Equation (3) are quadratic functions of output variables Y, the CCRF will then have the form of a multivariate Gaussian distribution, resulting in a computationally tractable CCRF model. Our approach deploys such a Gaussian form CCRF model with newly designed edge and variable features. We will discuss our model and feature design in detail next.

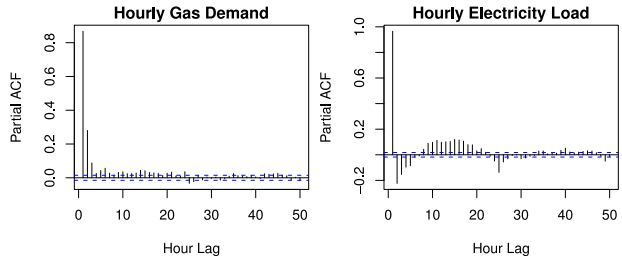


Fig. 3. Partial autocorrelation graphs of the gas demand and electricity load data.

3 CCRF for Energy Load

One of the core developments for a CCRF model is its edge and variable features.

3.1 Model Design

Our edge features are designed to capture the relationships between *two adjacent* target variables, and to ensure that the resultant CCRF has a multivariate Gaussian form. Our motivations are as follows. Our analysis on real-world short-term energy load data indicates that, for these data the adjacent target variables are highly correlated. As an example, Fig. 3 pictures the partial autocorrelation graphs [9] of two years' hourly gas demand and electricity load (we will discuss these two data sets in detail in Section 5) in the left and right subfigures, respectively. These two subfigures indicate that adjacent target variables show significant correlation, compared to the other related target variables. For example, as shown in Fig. 3, for both the gas demand and electricity load data, the first lag bears a correlation value of over 0.8. In contrast, other lags have a correlation of less than 0.3. These results suggest that the energy loads of a pair of adjacent hours are highly correlated.

Aiming to capture the above mentioned correlation, we consider two target variables y_i and y_j to be related (denoted by $i \sim j$) if they are adjacent, and deploy $G(\beta, y_i, y_j, X) = \sum_S \beta_s \omega_s (y_i - y_j)^2$ as our edge function form. Here y_i and y_j are the ith and jth target outputs, respectively. The ω_s is the sth of a set of S indicator functions with values of either zero or one, indicating if the correlation between y_i and y_j should be measured or not. For each y_i and y_j pair, we have multiple edge features (denoted by S), each responsible for a different type of relationship between y_i and y_j. This is pictured on the bottom panel of Fig. 2, where each leaf in the tree defines one edge feature (Section 4.1 will discuss this in detail). In other words, one can have S different edge features for the same y_i and y_j pair, each having its own β_s and $\omega_s (s \in S)$ parameters and mediating the relationship between the two target variables differently. The β here represents the weights for these feature functions, and these weights will be learned by the CCRF during the training. In particular, the quadratic

function forms here are specially designed to ensure that the CCRF results in a multivariate Gaussian form with tractable computation cost for the learning and inference, as will be further discussed later in this section.

In contrast to the edge potential feature which takes into account the interactions between predicted target variables, the variable potential feature of the CCRF, as described in Equation (3), aims at making good use of many efficient and accurate regression predictors. To this end, we consider variable features of the form $H(\alpha, y_i, X) = \sum_{k=1}^{m} \alpha_k (y_i - f_k(X))^2$. Here, y_i indicates the ith target output, $f_k(X)$ is the kth of m predictors for the target output y_i. This specific variable feature form is motivated by the following two reasons. First, with this particular form, the resultant CCRF strategy is able to include many efficient and accurate single-target regression models, such as regression trees or support vector machines, or existing state-of-the-art energy load predictors as its features. One may include a large number of such predictors, namely with a large m, and the CCRF will automatically determine their relevance levels during training. For example, for target output y_i, we can have the output from a single-target regression trees and the prediction from a SVM as its two features; during the learning, the CCRF will determine their contribution to the final prediction of the y_i through their weights. Second, the quadratic form here ensures that the final model results in a computationally tractable CCRF, as will be discussed next.

With the above edge and variable features, our CCRF strategy results in the graph structure depicted in Fig. 1, bearing the following formula:

$$\begin{aligned} P(Y|X) &= \frac{1}{Z(X,\alpha,\beta)} \\ &\quad \cdot exp\left(\sum_{1}^{n} H(\alpha, y_i, X) + \sum_{i\sim j} G(\beta, y_i, y_j, X) \right) \\ &= \frac{1}{Z(X,\alpha,\beta)} \cdot exp(- \sum_{i=1}^{n} \sum_{k=1}^{m} \alpha_k (y_i - f_k(X))^2 \\ &\quad - \sum_{i\sim j} \sum_{s=1}^{S} \beta_s \omega_s (y_i - y_j)^2). \end{aligned} \tag{7}$$

In this equation, we have n target outputs (i.e., $\{y_i\}_1^n$), m variable features (i.e., $\{f_k(X)\}_1^m$) for each target y_i, and S edge features (with s as index) for modeling the correlation between two outputs y_i and y_j (where indicator function ω_s indicates if the correlation between the ith and jth outputs will be taken into account or not). In our case, we use edge features to constrain the square of the distance between two outputs when the two outputs are adjacent. Note that we here assume that the neighboring information between two target outputs will be given.

Intuitively, the integration of the variable and edge feature, as described in Equation (7), forms a model with two layers. The variable features $\alpha_k (y_i - f_k(X))^2$ are predictors for individual target variables. That is, these variable features depend only on the inputs. Hypothetically, if the edge functions are disabled, the predictions of the CCRF model will be the outputs of these individual predictors. In this sense, we can consider the variable features as the *prior knowledge* for the multiple outputs. On the other hand, the edge potential functions $\beta_s \omega_s (y_i - y_j)^2$ involve multiple related target variables, constraining the relationships between related outputs. In fact, we can think of the edge features as representing a separate set of weights for each multi-targets output configuration. In other words, these weights serve as a second layer on top of the variable features. This second layer aims to fine-tune the predictions from the first layer, namely the prior knowledge provided by the variable features.

Promisingly, following the idea presented by Radosavljevic et al. [8], the above CCRF, namely Equation (7) can be further mapped to a multivariate Gaussian because of their quadratic forms for the edge and variable potential features:

$$\begin{aligned} P(Y|X) &= \frac{1}{(2\pi)^{n/2} |\sum|^{1/2}} \\ &\quad \cdot exp\left(-\frac{1}{2} (Y - \mu(X) \right)^T \sum^{-1} (Y - \mu(X))). \end{aligned} \tag{8}$$

In this Gaussian mapping, the inverse of the covariance matrix Σ is the sum of two $n \times n$ matrices, namely $\Sigma^{-1} = 2(Q^1 + Q^2)$ with

$$\begin{aligned} Q^1_{ij} &= \begin{cases} \sum_{k=1}^{m} \alpha_k & \text{if } i = j \\ 0 & otherwise \end{cases} \\ Q^2_{ij} &= \begin{cases} \sum_{j=1}^{n} \sum_{s=1}^{S} \beta_s \omega_s & \text{if } i = j \\ -\sum_{s=1}^{S} \beta_s \omega_s & \text{if } i \neq j. \end{cases} \end{aligned} \tag{9}$$

Also, the mean $\mu(X)$ is computed as $\Sigma\vec{\theta}$. Here, θ is a n dimensional vector with values of

$$\theta_i = 2 \sum_{k=1}^{m} \alpha_k f_k(X). \tag{10}$$

Practically, this multivariate Gaussian form results in tractable computation cost for the learning and inference of the CCRF model, which is discussed next.

3.2 Training CCRF

In the training of a CRF model, feature function constraints require the expected value of each feature with respect to the model be the same as that with respect to the training data [4]. Following this line of research, with a multivariate Gaussian distribution that aims at maximizing log-likelihood, the learning of a CCRF as depicted in Equation (5) becomes a convex optimization problem. As a result, stochastic gradient ascent can be applied to learn the parameters. In detail, for each instance in data set D, with learning rate η and initial values for $\log \alpha_k$ and $\log \beta_k$, the updating rules of the gradient ascent are as follows:

$$\log \alpha_k \leftarrow \log \alpha_k + \eta \times \nabla_{\log \alpha_k}, 1 \leq k \leq m, \tag{11}$$

$$\log \beta_s \leftarrow \log \beta_s + \eta \times \nabla_{\log \beta_s}, 1 \leq s \leq S, \tag{12}$$

where[2]

$$\begin{aligned}\nabla_{\log \alpha_k} &= \frac{\partial L(\alpha,\beta)}{\partial \log \alpha_k} = \frac{\partial \log \sum_{l=1}^{L} P(Y_l|X_l;\alpha,\beta)}{\partial \log \alpha_k} \\ &= \alpha_k \sum_{l=1}^{L}\left[-\frac{\partial \log Z}{\partial \alpha_k} - \sum_{i=1}^{n}(y_i - f_k(X))^2\right] \\ &= -\alpha_k \sum_{l=1}^{L}\left[\frac{\partial \log Z}{\partial \alpha_k} + \sum_{i=1}^{n}(y_i - f_k(X))^2\right], \end{aligned} \tag{13}$$

and

$$\begin{aligned}\frac{\partial \log Z}{\partial \alpha_k} &= \frac{\partial \log Z(X,\alpha,\beta)}{\partial \alpha_k} \\ &= \frac{1}{2}\frac{\partial(\log|\Sigma|)}{\partial \alpha_k} = \frac{1}{2}\mathrm{Tr}\left[\Sigma^{-1} \times \frac{\partial \Sigma}{\partial \alpha_k}\right] \\ &= -\frac{1}{2}\mathrm{Tr}\left[\Sigma \times \frac{\partial(\Sigma^{-1})}{\partial \alpha_k}\right]. \end{aligned} \tag{14}$$

Here, Tr is the Trace operation, which is defined to be the sum of the elements on the main diagonal of a matrix; $\times$ denotes matrix multiplication.

3.3 Inference in CCRF

In inference, finding the most likely predictions Y, given observation X as depicted in Equation (6), boils down to finding the mean of the multivariate Gaussian distribution. Specifically, it is computed as discussed earlier:

$$\widehat{Y} = \underset{Y}{\mathrm{argmax}}(P(Y|X)) = \mu(X) = \Sigma\vec{\theta}. \tag{15}$$

Furthermore, the 95 percent-confidence intervals of the estimated outputs can be obtained by $\widehat{Y} \pm 1.96 \times diag(\Sigma)$, due to the Gaussian distribution.

In the following section, we tackle two challenge issues in the CCRF, namely the weak feature constraint for continuous features and the expensive computational cost for the training and inference.

4 Improved CCRF

4.1 Cope with Weak Feature Constraint in CCRF

Recall from Section 3.1 that the edge features in the CCRF have the form of $(y_i - y_j)^2$. This particular function form aims to ensure that not only the correction between adjacent outputs are taken into account, but also the resultant CCRF has a multivariate Gaussian form with tractable computation cost for the learning and inference. This design, however, results in a weak feature constraint problem for the CCRF. This is due to the fact that now each edge function depends on multiple, *continuous* target variables. As a result, the mean of $(y_i - y_j)^2$, for example, may not be able to capture complex relationships between y_i and y_j well. We detail this challenge as follows.

In a nutshell, CRF is a maximum entropy model with feature constraints that capture relevant aspects of the training data. That is, training a CRF amounts to forcing the expected value of each feature with respect to the model to be the same as that with respect to the training data. Consequently, the constraints with binary feature, for example, contain essential information about the data because knowing the mean of the binary feature is equivalent to knowing its full distribution. On the other hand, knowing the mean may not tell too much about the distribution of continuous variables because of CCRF's linear parameterization characteristics [4], [10]. As an example, the mean value of the red curve distribution on the top subfigure in Fig. 2 does not tell us too much about the distribution of the curve. As a result, the CCRF may learn less than it should from the training data.

To tackle this constraint weakness, one typically introduces the "Binning" technique. That is, one can divide the real value into a number of bins, and then each bin is represented by a binary value. However, in the CCRF, typical "Binning" techniques are difficult to apply to the edge functions because all the values for these features are predicted values of the target variables, and we do not know these values beforehand. That is, we do not know, for example, the values of y_i and y_j in inference time. To cope with unknown target variables, one may have to "Bin" these features using only the known input variables. Nevertheless, relying on only the observed inputs may not be enough to distinguish the interactions between the pair of unknown outputs. For example, a large y_i value and a small y_j value may have the same result, as computed by $(y_i - y_j)^2$, as that of a small y_i and a large y_j value pair. These observations suggest that it will be beneficial to have a "Binning" technique that is able to *simultaneously* take the interactions of a pair of outputs and the observed inputs into account.

Following this line of thought, we propose to use the predictive clustering trees [5]. The aim here is to use the PCTs to divide the relationships of related outputs into a set of "sub-relationships", each providing more specific feature constraints for the interplays of the related outputs. The PCTs strategy considers a decision tree as a hierarchy of clusters. The root node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. When dealing with multiple target attributes, the PCTs approach can be viewed as a tree where each leaf has multiple targets, compared to that of traditional decision tree which learns a scalar target. The PCTs method extends the notion of class variance towards the multi-dimensional regression case. That is, given a distance function, such as the sum of the variances of the target variables, for the multi-dimensional target space, the PCTs algorithm partitions the input space, namely X, into different disjoint regions, where each is a leaf and each groups instances with similar values for the target variables Ys. When deployed for CCRF, each PCTs tree can be used to model the interactions between the related Ys through its *leaves*. The graph structure in Fig. 1 pictures our CCRF model, where each *unfilled square* describes an edge feature, and each is represented by a PCTs tree on the right of the figure.

We illustrate the above weak edge feature constraint and the proposed PCTs solution with Fig. 2. In this figure, the red curve shows the distribution of the edge potential feature of

2. We here present the computations for $\nabla_{\log \alpha_k}$; $\nabla_{\log \beta_k}$ can be computed using the same equations.

$(y_i - y_j)^2$ in the gas demand (used for heating) data set. Here, y_i and y_j represent the energy loads of two neighboring hours, namely hours i and j, respectively. This distribution subsumes three sub-distributions, depicted by the blue, brown, and green curves, respectively. In detail, the blue curve pictures the distribution of $(y_i - y_j)^2$ where the hours i and j have similar temperature; the brown curve presents the same two hours with a dramatically increasing temperature; and the green curve shows the distribution of the same two hours where the temperature drops sharply. Intuitively, one can consider the red curve pictures the marginal density of a joint probability $P((y_i - y_j)^2, X)$, and the other three curves show the densities of conditional probability $P((y_i - y_j)^2|X)$ when X takes one of the three weather scenarios, namely, similar, sharply increasing, and dramatically dropping temperatures between two neighboring outputs.

As can be seen from this example, the edge feature[3] of $(y_i - y_j)^2$, as shown by the red curve, is not able to distinguish the three sub-relationships clustered by the blue, brown, and green curves. That is, the edge feature constraints represented by the red curve cannot distinguish between a similar, increasing, or reducing energy consumption trends. Such weak edge feature will limit the constraining power of the edge potential functions in the CCRF. It is worth to further noting that, if we do not *simultaneously* consider the input variables and the interplays between the target variables, as what the PCTs do, we may not be able to distinguish the brown and green curves since these two curves represent similar $(y_i - y_j)^2$ values.

Let us continue with the above example. Tackled by the PCTs (as shown at the bottom of Fig. 2), the original edge feature of $(y_i - y_j)^2$, as depicted by the red curve, will be replaced by three sub features, namely the distributions shown in the blue, brown, and green curves. In other words, *three* edge feature constraints, instead of *only one*, will be used by the $G(\beta, y_i, y_j, X)$ function, representing three different types of interplays between the (y_i, y_j) pair: one constraining a small change between y_i and y_j, another defining a sharp increase of energy consumption, and the other confining a quick drop in term of energy consumption.

Let us sum up the above example. The edge function with PCTs here can naturally model the multi-steps ahead energy consumptions: 1) if the temperature (which can be observed or forecasted) is sharply dropping, the constraint of a small y_i and a large y_j will have a high probability; 2) if the temperature is dramatically increasing, the constraint of a large y_i and a small y_j will have a high probability; 3) if the temperature is similar, similar values for y_i and y_j will then have a high probability.

4.2 Accelerated Training and Inference in CCRF

Recall from Section 3.1 that, aiming at constructing a computationally tractable CRF model, the training and inference of the CCRF is implemented through matrix computation. Consequently, the computation cost of the CCRF model mainly comes from these matrix operations. In detail, as depicted in Equations (14) and (15), both the training and inference involve finding the inverse of the given matrix Σ^{-1}. Such inversion operation typically requires cubic computation complexity, namely $\mathcal{O}(n^3)$, with respective to the number of target variables n. As a result, the CCRF may not scale well when the number of target variables n is large. Below, we provide solutions to enable the CCRF to conduct the training and inference without constructing the matrix Σ, i.e., the full inverse of the given Σ^{-1} matrix. By doing so, the CCRF results in linear computation complexity for the matrix calculations in its training and inference. Next, we discuss these two effective computation methods in detail.

4.2.1 Fast Training

Since the main computation cost for training the CCRF comes from solving Equation (14), let us further target this equation. In fact, the $\mathrm{Tr}[\Sigma \times \frac{\partial(\Sigma^{-1})}{\partial \alpha_k}]$ in Equation (14) can be rewritten as the Frobenius scalar product of the two matrices, namely Σ and $\frac{\partial(\Sigma^{-1})}{\partial \alpha_k}$. In other words, Equation (14) can be rewritten as following:

$$\begin{aligned}\frac{\partial \log Z}{\partial \alpha_k} &= -\frac{1}{2}\mathrm{Tr}\left[\Sigma \times \frac{\partial(\Sigma^{-1})}{\partial \alpha_k}\right] \\ &= -\frac{1}{2}\sum_i \sum_j \Sigma_{ij}\left[\frac{\partial(\Sigma^{-1})}{\partial \alpha_k}\right]_{ij}.\end{aligned} \tag{16}$$

This rewritten equation implies that $\frac{\partial \log Z}{\partial \alpha_k}$ may be computed without constructing Σ if some of the elements in the matrix $\frac{\partial(\Sigma^{-1})}{\partial \alpha_k}$ are zero. In our case, as will be discussed further, the $\frac{\partial(\Sigma^{-1})}{\partial \alpha_k}$ matrix is, indeed, a diagonal matrix, and the $\frac{\partial(\Sigma^{-1})}{\partial \beta_k}$ matrix is a tridiagonal one. We take advantage of these observations, and are able to significantly lower the calculation cost for solving $\frac{\partial \log Z}{\partial \alpha_k}$ and $\frac{\partial(\Sigma^{-1})}{\partial \beta_k}$, as follows.

Recall from Section 3.1 that, our CCRF only needs to take into account the correlation between two adjacent outputs. As a result, the resulting matrix Σ^{-1} is a sparse one. Specifically, the matrix Σ^{-1} is, indeed, a n x n *tridiagonal* matrix, where nonzero entries lie along the main diagonal, the immediate sub-diagonal, and the super-diagonal of the matrix.[4] That is, the Σ^{-1} tridiagonal matrix takes the following form:

$$\Sigma^{-1} = \begin{bmatrix} \vartheta_{11} & \vartheta_{12} & & 0 \\ \vartheta_{21} & \ddots & \ddots & \\ & \ddots & \ddots & \vartheta_{n-1n} \\ 0 & & \vartheta_{nn-1} & \vartheta_{nn} \end{bmatrix}, \tag{17}$$

3. Note that, as discussed in Section 3.1, the quadratic function forms here are specially designed to ensure that the CCRF results in a multivariate Gaussian form with tractable computation cost for the learning and inference.

4. For descriptive purpose, we denote the elements lie along the main diagonal, the immediate sub-diagonal, and the super-diagonal of a matrix as the "non-zero entries" of the matrix. Note that, they may not be the only non-zero elements in a matrix. For example, when we talk about the "non-zero entries" in the inverse of the matrix Σ^{-1}, we do not mean that all the other elements in the inverse have the value of zero.

where

$$\vartheta_{ij} = \begin{cases} \sum_{k=1}^{m} \alpha_k + \sum_{j=1}^{n} \sum_{s=1}^{S} \beta_s \omega_s & \text{if } i = j \\ -\sum_{s=1}^{S} \beta_s \omega_s & |i-j| = 1 \\ 0 & otherwise. \end{cases}$$

Because the matrix Σ^{-1} has a tridiagonal form as depicted in Equation (17), one can conclude that the $\frac{\partial(\Sigma^{-1})}{\partial \alpha_k}$ is a diagonal matrix and the $\frac{\partial(\Sigma^{-1})}{\partial \beta_k}$ is a tridiagonal one. Consequently, Equation (16) can be computed by using only all the non-zero entries in the two matrices. For conciseness purpose, we here only present the computations for $\frac{\partial \log Z}{\partial \beta_k}$ because diagonal matrix can be considered as a special case of tridiagonal matrix, so $\frac{\partial \log Z}{\partial \alpha_k}$ can be computed with the same equations. It is worth to noting that Σ is often a dense matrix, rather than a tridiagonal one. Since finding the non-zero entries for the matrix $\frac{\partial(\Sigma^{-1})}{\partial \beta_k}$ needs $3n$ operations due to its tridiagonal form, the computation complexity here lies in computing the non-zero entries in the matrix Σ, namely the inverse of the matrix Σ^{-1}.

As mentioned earlier, finding the inverse of a $n \times n$ matrix typically requires $\mathcal{O}(n^3)$ operations. Fortunately, Rybicki and Hummer [6] have proven that, if one does not need to construct the full inverse of a tridiagonal matrix, all the non-zero entries in the inverse of the tridiagonal matrix can be found quickly with only $3n$ operations. Following the idea presented in [6], for our $n \times n$ matrix Σ^{-1}, we first compute the diagonal elements (denoted as λ_{ii}) of its inverse Σ. This process needs n operations. Second, making use of the previously computed λ_{ii}, we then calculate the off-diagonal elements using a recursive procedure. In this way, calculating the two off-diagonal, non-zero entries (denoted as $\lambda_{ij}, |i-j| = 1$) of the Σ needs $2n$ operations (see Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2399311, for the details).

By doing so, we need $3n$ operations to calculate the non-zero entries in the matrix Σ, and $3n$ operations to compute all the non-zero elements in the tridiagonal matrix $\frac{\partial(\Sigma^{-1})}{\partial \beta_k}$. Consequently, the computation of the Frobenius scalar product of the two matrices, as outlined in Equation (16), needs a computation complexity of only $\mathcal{O}(n)$. In other words, through taking advantage of the fact that one of the matrices in the Equation (16) is a tridiagonal matrix, the Frobenius scalar product formulated in Equation (16) can be computed with a linear cost, compared to that of complexity of $\mathcal{O}(n^3)$ required by a straightforward computation in which a matrix inversion operation is needed.

4.2.2 Linear Time Inference

Recall from Section 3.3 that, a matrix operation, namely the product between the matrix Σ and the vector θ as shown in Equation (15), accounts for the computation cost of the inference procedure in the CCRF. Specifically, this inference process needs to construct the inverse of the matrix Σ^{-1}. This matrix inversion operation typically has complexity of $\mathcal{O}(n^3)$, if one computes it directly. To speed up the inference of the CCRF, we introduce the Thomas algorithm [7]. As a result, the inference of the CCRF can be conducted with cost of $\mathcal{O}(n)$. Next, we discuss the application of the Thomas algorithm in detail.

Recall from Section 4.2.1 that, our CCRF only needs to take into account the correlation between two adjacent outputs. Hence, the resulting matrix Σ^{-1} is, indeed, a $n \times n$ tridiagonal matrix. This tridiagonal matrix can be stored using only three n-dimensional vectors since there is no need to store the zero elements. In particular, Equation (15), namely $\widehat{Y} = \underset{Y}{\operatorname{argmax}}(P(Y|X)) = \Sigma\vec{\theta}$, can be rewritten as:

$$\Sigma^{-1}\widehat{Y} = \vec{\theta}. \tag{18}$$

In matrix computation form, this looks like:

$$\begin{bmatrix} \vartheta_{11} & \vartheta_{12} & & 0 \\ \vartheta_{21} & \ddots & \ddots & \\ & \ddots & \ddots & \vartheta_{n-1n} \\ 0 & & \vartheta_{nn-1} & \vartheta_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n-1} \\ \theta_n \end{bmatrix}, \tag{19}$$

With this rewritten form, finding the product of $\widehat{Y} = \Sigma\vec{\theta}$ reduces to solving the linear system $\Sigma^{-1}\widehat{Y} = \vec{\theta}$. Consequently, this equation can be resolved using the Thomas algorithm directly, in linear computational cost.

The Thomas algorithm is a special case of Gaussian elimination, and consists of three main steps. First, the LU decomposition of the matrix Σ^{-1} is determined. This factorization results in a lower triangular matrix L and an upper triangular matrix U, where both matrices L and U are bi-diagonals. Second, a forward substitution procedure is proceeded to calculate $L\vec{v} = \vec{\theta}$. Here $\vec{v}$ is an intermediate vector. Finally, a backward substitution, where the matrix U is involved, is conducted to compute $U\widehat{Y} = \vec{v}$, finding the solution for $\widehat{Y}$. In this way, the Thomas algorithm makes the calculation of $\widehat{Y}$ with a linear computational time $\mathcal{O}(n)$ (see Appendix B, available in the online supplemental material, for the details).

In summary, through introducing two fast tridiagonal matrix computation techniques, we can significantly speed up the training and inference in the CCRF.

5 Experimental Studies

5.1 Data Sets

Two real world data sets were collected from a typical commercial building in Ontario: one aims to predict the hourly electricity loads for the next 24 hours, and another for the next 24 hours' gas demands. For the electricity, one year of hourly energy consumption data in 2011 and three months of summer data, from March 1st to May 31st in 2012, were collected; for gas, we have the whole year's data in 2011 and winter data from January to March in the year of 2012.

In our experiments, for both the electricity and gas, we trained the model with the 2011 data and then tested the model using the data from 2012. In addition, during the testing, we shifted the target window in a daily basis. That is, we generated 24 predictions at mid night for each

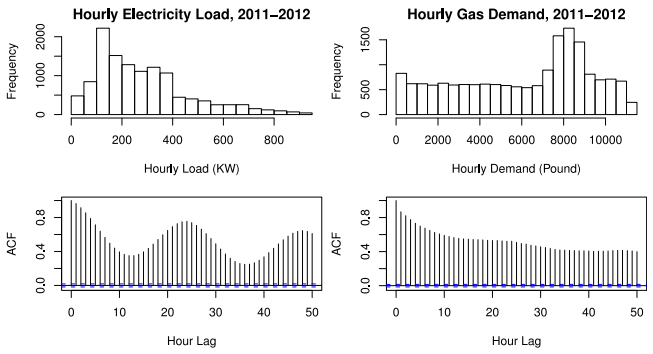


Fig. 4. Histogram and Autocorrelation graphs of the hourly Gas Demand (right subfigure) and Electricity Load (left subfigure) for the collected 2011 and 2012 data.

day. Also, due to the seasonal effects, monthly energy consumption are often very different as well as the interaction among the target variables. Consequently, we considered each month as a separate test case, and presented the results of all months.

We chose the gas and electricity usage data to evaluate our proposed method because these two data sets bear very different energy load characteristics. Fig. 4 shows the histogram and autocorrelation graphs of the two data sets.

5.2 Features and Settings

In these two energy load forecasting systems, the proposed CCRF method deployed 23 edge features, as discussed in Section 3.1. Each such feature aims to capture the interplays of an adjacent pair of target variables, namely two consecutive hours of the 24 hours. The number of sub-regions generated for each of these edge features were controlled by the search depth of the PCTs trees. The larger this number, potentially more sub-regions or clusters will be created to group a pair of related target variables. In our experiments, we set this number to 3. In fact, we compared with different settings and the model was insensitive to this parameter.

Also, 24 variable features were used, each focusing on one target variable, namely an individual hour of the 24 output hours. To this end, we deploy Friedman's additive gradient boosted trees [11], [12] as our CCRF model's variable features. Friedman's additive boosted trees can be considered as a regression version of the well-known Boosting methodology for classification problems. Promising results of applying this additive approach have been observed, in terms of improving the predictive accuracy for regression problems [11]. In our studies here, each such variable feature, namely each target y_i, is modeled using an additive gradient boosted strategy with the following parameters: a learning rate of 0.05, 100 iterations, and a regression tree as the base learner. The input features for the Friedman machine include past energy usages, temperatures, the day of the week, and the hour of the day.

In addition, to avoid overfitting in the training of the CCRF, penalized regularization terms $0.5\alpha^2$ and $0.5\beta^2$ were subtracted from the log-likelihood function depicted in Equation (5). Also, the number of iterations and learning rate for the gradient ascent in the CCRF learning were set to 100 and 0.0001, respectively.

5.3 Methodology

We compared our method with three benchmarking approaches. The first comparison algorithm is a state-of-the-art multi-target system, namely the ensembles of Multi-Objective Decision Trees (MODTs) [13]. We obtained the settings of the ensembles of MODTs from their authors. That is, in our experiments, a random forest strategy was applied to combine 100 individual multi-objective decision trees. The second benchmarking algorithm is a strategy that trains independent regression models for each target attribute and then combines the results [14], [15]. In our studies, a collection of regression trees were used where each tree models a target variable. The last comparison approach we compared with is a CCRF model with basic features. That is, this CCRF strategy used 24 single-target regression trees as its variable features. Also, each of the 23 edge features captures the square of the distance between two adjacent target variables. The comparison here aims to evaluate the impact of the newly designed features, namely the predictive clustering approach, to the CCRF strategy.

We implemented the CCRF models in Java on a 2.93 GHz PC with 64 bit Windows Vista installed. We measured the performance of the tested algorithms with the mean absolute percentage error and the root mean square error. For descriptive purpose, we referred to the random forests approach with multi-objective decision trees, the method of learning a collection of regression trees, the basic CCRF algorithm, and the proposed CCRF strategy as MODTs, RTs, CCRFs_BASE, and CCRFs_EP, respectively.

5.4 Experimental Results

In this section we examine the predictive performance of the proposed method against both the electricity and gas data, in terms of MAPE and RMSE.

5.4.1 *Electricity Usage*

Our first experiment studies the performance of the tested methods on the electricity load data. We present the MAPE and RMSE obtained by the four tested approaches for each of the three months, namely March, April, and May, in Fig. 5. In this figure, we depicted the MAPE and RMSE obtained on the top and bottom subfigures, respectively.

The MAPE results, as presented on the top subfigure of Fig. 5 show that the CCRF method appears to consistently reduce the error rate for each of the three months, when compared to all the other three tested strategies, namely the collection of regression trees, the random forests with multi-objective decision trees, and the CCRF model with basic features. For example, when compared with the collection of regression trees method, namely the RTs approach, the CCRFs_EP model decreases the absolute MAPE for months March, April, and May with 0.51, 0.53, and 1.0, respectively. The relative average error reduced for these three tested months was 17.9 percent (drop from 3.80 to 3.12 as shown on the top of Fig. 5). In terms of RMSE, for each of the three months, the error was reduced by the CCRFs_EP method from 139.63, 136.91, and 165.86 to 112.11, 114.76, and 118.17, respectively. As depicted at the bottom of Fig. 5, a relative average reduction was 22.0 percent (drop from 147.47 to 115.01).

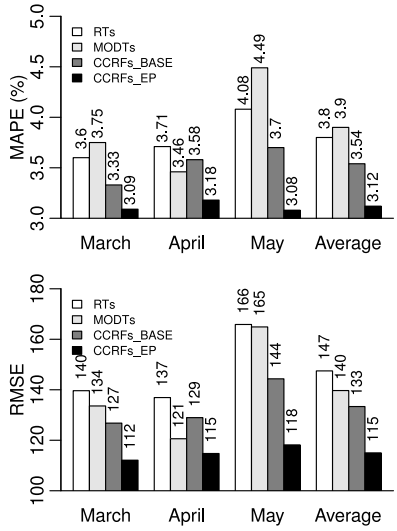


Fig. 5. MAPE and RMSE obtained by the four methods, against the electricity data in the months of March, April, and May in 2012.

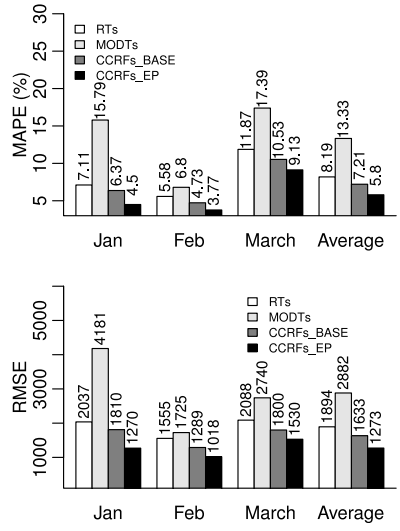


Fig. 6. MAPE and RMSE obtained by the four methods, against the gas data in the months of January, February, and March in 2012.

When considering the comparison with the random forests of multi-objective decision trees, namely the MODTs method, the results as depicted in Fig. 5 indicate that the CCRFs_EP model was also able to meaningfully reduce the error. As shown in Fig. 5, for both MAPE and RMSE, the CCRFs_EP strategy was able to reduce the error for all the three months. On average, relative error reductions of 20.08 and 17.66 percent were achieved by the CCRFs_EP model over the MODTs strategy, in terms MAPE and RMSE, respectively.

Comparing to the CCRFs_BASE algorithm, the CCRFs_EP method also appears to consistently outperform the CCRFs_BASE strategy for each of the three months regardless the evaluation metrics used, namely no matter if the MAPE or RMSE was applied as the predictive performance metrics. As depicted in Fig. 5, average relative error reductions of 11.87 and 13.75 percent were achieved by the CCRFs_EP model over the CCRFs_BASE approach, in terms MAPE and RMSE, respectively. These results suggest that the advanced potential feature functions as introduced in Section 4.1 enhanced the proposed CCRF model's predictive performance.

5.4.2 Gas Consumption

Our second experiment investigates the performance of the tested methods on the gas demand data. We present the MAPE and RMSE obtained by the four tested methods for each of the three months, namely January, February, and March, in Fig. 6. In this figure we depicted the MAPE and RMSE obtained on the top and bottom subfigures, respectively.

The MAPE results, as presented on the top subfigure of Fig. 6 show that the proposed CCRFs_EP method appears to consistently reduce the error for all the three months, when compared to the RTs, MODTs, and CCRFs_BASE methods. For instance, when compared with the RTs algorithm, the results on the top subfigure of Fig. 6 show that the CCRFs_EP model decreases the absolute MAPE for months January, February, and March with 2.61, 1.81, and 2.74, respectively. The relative average error reduced for these three months was 29.15 percent (drop from 8.19 to 5.80 as indicated in Fig. 6). In terms of RMSE, results at the bottom of Fig. 6 demonstrate that, the CCRF_EP strategy outperformed the RTs algorithm for all the three tested months. A relative average reduction was 32.77 percent (drop from 1,894 to 1,272 as shown at the bottom of Fig. 6).

When considering the comparison with the MODTs method, the results in Fig. 6 indicate that the proposed CCRF model meaningfully reduce the error rate. For example, for both MAPE and RMSE, the CCRF_EP strategy was able to reduce the error for all the three months. As shown at the bottom of Fig. 6, average relative error reductions of 56.47 and 55.82 percent were achieved by the CCRF_EP model over the random forests ensemble.

Comparing to the CCRFs_BASE, the CCRFs_EP method again appears to consistently outperform the CCRFs_BASE strategy for all the three months in terms of both the MAPE or RMSE. As depicted in Fig. 6, average relative error reductions of 19.55 and 22.05 percent were achieved by the CCRFs_EP model over the CCRFs_BASE strategy, in terms MAPE and RMSE, respectively.

In summary, the experimental results on the six data sets indicate that, the proposed CCRF model consistently outperformed the other three tested methods in terms of MAPE and RMSE. Promisingly, the relative error reduction achieved by the proposed CCRF algorithm was at least 11.87 percent, and up to 56.47 percent.

5.4.3 Confidence Intervals, Steps-Ahead Error, and Speed Up Evaluation

In addition to its superior accuracy, the proposed CCRF has the form of a multivariate Gaussian. Therefore, it can provide projects with probability distributions rather than only the forecasted numbers. In Fig. 7, we depicted a sample of the 24 predictions with their 95 percent confidence intervals from our gas forecasting system. The 24 hours ahead predictions, along with their confidence intervals, were generated for the date of April 1st, 2012, at midnight. In this figure, the dark and gray curves in

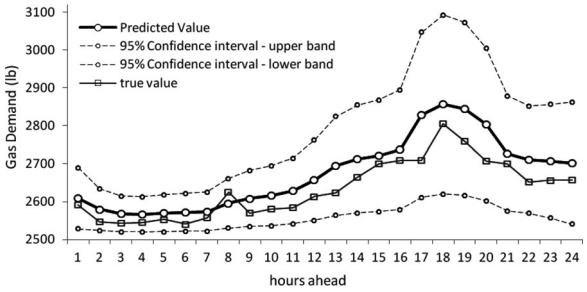


Fig. 7. Outputs with 95 percent confidence bands for the gas consumptions of April 1st, 2012.

the middle show the 24 predictions and their corresponding true labels, respectively. Also, the two dot curves depict the two confidence interval bands. These smooth, uncertainty information could be beneficial for better decision makings in energy load management.

To further understand the CCRFs_EP's predictive performance over the prediction horizon, we also provided, in Figs. 8 and 9, the MAPE obtained by the CCRFs_BASE and the CCRFs_EP approaches against the electricity and gas data sets from the first hour up to the 24th hour into the future. Results as shown in Figs. 8 and 9 suggest that in all the future time stamps, the CCRFs_EP outperformed the CCRFs_BASE. Promisingly, Fig. 8 indicates that, for the electricity data, although smaller predictive improvements were obtained for the first several hours ahead by the CCRFs_EP, larger accuracy improvements were achieved in the further time stamps in the prediction horizon. Similar improvement can also be observed from Fig. 9 when against the gas data. These observations imply that the edge features introduced by the CCRF_EP was able to better constrain the relationship between the target variables.

To investigate the computational speed up without the matrix inversion for the CCRF, we compared the average training time needed for the CCRF_EP against two settings: 1) a straightforward computation in which a matrix inversion operation is involved, and 2) the proposed fast training method as discussed in Section 4.2. Our experimental results indicated that the former was twice slower than the latter: the average training times for the two methods were 191.46 seconds and 80.97 seconds, respectively. In our studies here, the matrices were of size 24×24. It is worth noting that the presented linear computational speed up would be particularly relevant when a CCRF model has a large number of target variables, such as hundreds of structural targets as required by predicting Ontario's every five-minutes provincial electricity demand for the next 24 hours.

6 Related Work

Short-term energy load forecasting has been an active research area for decades, and a variety of machine

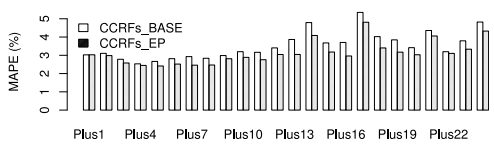


Fig. 8. MAPE by hours into the future on electricity.

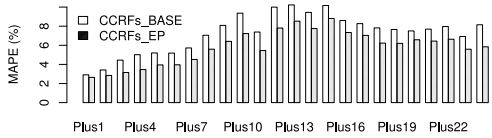


Fig. 9. MAPE by hours into the future on gas.

learning techniques have been proposed to cope with this challenge, including regression algorithms [16], time series analysis strategies [17], Neural networks [18], and Support Vector Machines [19], amongst others. An informative review has been reported by Feinberg and Genethliou [20]. Comparing with the CCRF methods, many existing approaches either have difficulties to make use of different types of features (such as dependent features, categorical features etc.), to generate statistical information of the estimated values (e.g., the confidence intervals), or to explore the interrelationships among the multiple outputs (e.g., structured outputs).

Recent years, conditional random fields has been devised to provide a probabilistic model, within the undirected graphical framework, to represent the conditional probability of a particular label sequence. This discriminative framework has been very successfully applied to many classification tasks, including text labeling [1], [21], [22], activity recognition [4], [23], [24], social recommendation [25], and image recognition [26], amongst others. On the other hand, only a few applications of applying this framework on regression tasks have been reported. These applications include document ranking [2], Aerosol optical depth estimation [8], and travel speed prediction [27]. To our best knowledge, this paper is the first to report an application of conditional random fields on short-term energy load forecasting. Additionally, we address the weak feature constraint problem in the CCRF strategy. Furthermore, we employ two tridiagonal matrix computation techniques to significantly speed up the training and inference of the CCRF.

Within the CRF research community, issues related to the powerful and flexible CRF model have also been actively studied. For instance, Dietterich et al. [28] have proposed to train a CRF via additive gradient boosted trees. Nevertheless, this approach cannot be applied to our proposed CCRF method. This is because our CCRF strategy requires all its potential feature functions in quadratic forms, so that the CCRF model has the formula of a multivariate Gaussian distribution. Furthermore, their CRF method deploys single-target trees and takes aim at classification tasks.

More recently, Yu et al. [3] has proposed a spline-based method to cope with continuous features. This "binning" technique cannot be directly deployed in our CCRF strategy. In our proposed CCRF approach, we need to simultaneously consider *correlated* target variables based on only their shared feature space. That is, our multi-target trees algorithm allows partition multiple target variables in each divided region. In particular, such partitions rely on only the observed features, instead of the target variables themselves.

This paper builds on our earlier work as presented in [29]. In comparison with the earlier paper, this manuscript presents two fast tridiagonal computation techniques to

avoid the matrix inversion operations in CCRF's training and inference. The newly introduced techniques enable these matrix calculations to be conducted in linear time, instead of cubic computation complexity as needed for naive matrix computations. Also, this paper contains additional material, including a detailed description of the learning, additional experimental results, and extended related work.

7 Conclusions and Future Work

Embracing "smart energy consumption" to optimize energy usage in commercial buildings has provided a unique demand for modeling short-term energy load. We have devised a continuous conditional random Fields strategy to cope with these structured outputs tasks. In particular, we improve the CCRF's predictive performance with a novel edge feature and speed up its training and inference with two tridiagonal matrix computation techniques. Our experimental studies show that the proposed approach can meaningfully reduce the predictive error for two energy systems, in terms of mean absolute percentage errors and root mean square errors.

In our future studies, we will devise an incremental learning schema for our proposed algorithm, and further evaluate its scalability against high speed data streams. We also plan to employ similar decision trees techniques to enable CCRF to further model different subregions of each variable feature's input space.

Acknowledgments

The author thanks the reviewers for their insightful comments on their submission, which helped improve its quality. H. Guo is the corresponding author.

References

[1] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

[2] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, and H. Li, "Global ranking using continuous conditional random fields," in *Proc. Adv. Neural Inform. Process. Syst. 21*, 2008, pp. 1281–1288.

[3] D. Yu, L. Deng, and A. Acero, "Using continuous features in the maximum entropy model," *Pattern Recogn. Lett.*, vol. 30, no. 14, pp. 1295–1300, Oct. 2009.

[4] D. L. Vail, M. M. Veloso, and J. D. Lafferty, "Conditional random fields for activity recognition," in *Proc. 6th Int. Joint Conf. Autonomous Agents Multiagent Syst.*, 2007, pp. 235:1–235:8.

[5] H. Blockeel, L. D. Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 55–63.

[6] G. B. Rybicki and D. G. Hummer, "An accelerated lambda iteration method for multilevel radiative transfer. i. non-overlapping lines with background continuum," *Astron. Astrophys.*, vol. 245, pp. 171–181, 1990.

[7] R. L. Burden and J. D. Faires, *Numerical Analysis*, 7th ed. Pacific Grove, CA, USA: Books/Cole, 2001.

[8] V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous conditional random fields for regression in remote sensing," in *Proc. 19th Eur. Conf. Artif. Intell.*, 2010, pp. 809–814.

[9] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco, 1970.

[10] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction Statistical Relational Learning*. Cambridge, MA, USA: MIT Press, 2007.

[11] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, pp. 367–378, 1999.

[12] J. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, 2000.

[13] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Ensembles of multi-objective decision trees," in *Proc. 18th Eur. Conf. Mach. Learn.*, 2007, pp. 624–631.

[14] M. Petrovskiy, "Paired comparisons method for solving multi-label learning problem," in *Proc. 6th Int. Conf. Hybrid Intell. Syst.*, Dec. 2006, p. 42.

[15] B. Ženko and S. Džeroski, "Learning classification rules for multiple target attributes," in *Proc. Adv. Knowl. Discovery Data Mining*, 2008, pp. 454–465.

[16] R. Engle, C. Mustafa, and J. Rice, "Statistical comparisons of classifiers over multiple data sets," *J. Forecasting*, vol. 11, p. 241251, Dec. 1992.

[17] J. Fan and J. McDonald, "A real-time implementation of short-term load forecasting for distribution power systems," *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 988–994, May 1994.

[18] T. Peng, N. Hubele, and G. Karady, "Advancement in the application of neural networks for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 7, no. 1, pp. 250–257, Feb. 1992.

[19] B.-J. Chen, M.-W. Chang, and C.-J. lin, "Load forecasting using support vector machines: a study on eunite competition," *Trans. Power Syst.*, vol. 19, pp. 1821–1830, 2004.

[20] E. Feinberg and D. Genethliou, *Load Forecasting*. New York, NY, USA: Springer, 2005.

[21] C. Yang, Y. Cao, Z. Nie, J. Zhou, and J.-R. Wen, "Closing the loop in webpage understanding," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 639–650, May 2010.

[22] J. Tang, M. Hong, J. Li, and B. Liang, "Tree-structured conditional random fields for semantic annotation," in *Proc. 5th Int. Semantic Web Conf.*, 2006, pp. 640–653.

[23] J. Yin, D. H. Hu, and Q. Yang, "Spatio-temporal event detection using dynamic conditional random fields," in *Proc. 21st Int. Jont Conf. Artif. Intell.*, 2009, pp. 1321–1326.

[24] H. Wang, C. Wang, C. Zhai, and J. Han, "Learning online discussion structures by conditional random fields," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2011, pp. 435–444.

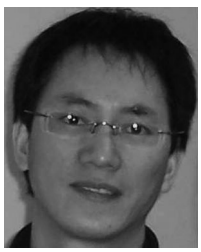
[25] X. Xin, I. King, H. Deng, and M. R. Lyu, "A social recommendation framework based on multi-scale continuous conditional random fields," in *Proc. 18th ACM Conf. Inform. Knowl. Manage.*, 2009, pp. 1247–1256.

[26] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *Proc. Neural Inform. Process. Syst.*, 2004, pp. 1097–1104.

[27] N. Djuric, V. Radosavljevic, V. Coric, and S. Vucetic, "Travel speed forecasting by means of continuous conditional random fields," *Transport. Res. Record: J. Transport. Res. Board*, vol. 2263, pp. 131–139, 2011.

[28] T. G. Dietterich, A. Ashenfelter, and Y. Bulatov, "Training conditional random fields via gradient tree boosting," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 28–36.

[29] H. Guo, "Modeling short-term energy load with continuous conditional random fields," in *Proc. Eur. Conf. Mach. Learn. Principles Pract. Knowl. Discovery Databases*, 2013, pp.pp. 433–448.



Hongyu GUO received the BEng and PhD degrees from the Shanghai Jiao Tong University and University of Ottawa, respectively, both in Computer Science. He is a research officer at the National Research Council of Canada. His research interests include the fields of machine learning and data mining. He was a co-winner of the Best Paper Award at the 2012 MLDM conference and a co-winner of the Distinguished Papers Award at the 2006 ECML-PKDD conference.