University of Pennsylvania
ScholarlyCommons

Real-Time and Embedded Systems Lab (mLAB) School of Engineering and Applied Science

11-14-2015

Sometimes, Money Does Grow On Trees: Data-Driven Demand Response with DR-Advisor

Madhur Behl
University of Pennsylvania, mbehl@seas.upenn.edu

Rahul Mangharam
University of Pennsylvania, rahulm@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/mlab_papers

Part of the Computer Engineering Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

Madhur Behl and Rahul Mangharam, "Sometimes, Money Does Grow On Trees: Data-Driven Demand Response with DR-Advisor", *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* , 137-146. November 2015. http://dx.doi.org/10.1145/2821650.2821664

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/mlab_papers/82
For more information, please contact repository@pobox.upenn.edu.

Sometimes, Money Does Grow On Trees: Data-Driven Demand Response with DR-Advisor

Abstract

Real-time electricity pricing and demand response has become a clean, reliable and cost-effective way of mitigating peak demand on the electricity grid. We consider the problem of end-user demand response (DR) for large commercial buildings which involves predicting the demand response baseline, evaluating fixed DR strategies and synthesizing DR control actions for load curtailment in return for a financial reward. Using historical data from the building, we build a family of regression trees and learn data-driven models for predicting the power consumption of the building in real-time. We present a method called DR-Advisor called DR-Advisor, which acts as a recommender system for the building's facilities manager and provides suitable control actions to meet the desired load curtailment while maintaining operations and maximizing the economic reward. We evaluate the performance of DR-Advisor for demand response using data from a real office building and a virtual test-bed.

Disciplines

Computer Engineering | Electrical and Computer Engineering

This conference paper is available at ScholarlyCommons: http://repository.upenn.edu/mlab_papers/82

Sometimes, Money Does Grow On Trees:

Data-Driven Demand Response With DR-Advisor

Madhur Behl and Rahul Mangharam
Department of Electrical and Systems Engineering
University of Pennsylvania
PA, USA
{mbehl, rahulm}@seas.upenn.edu

ABSTRACT

Real-time electricity pricing and demand response has become a clean, reliable and cost-effective way of mitigating peak demand on the electricity grid. We consider the problem of end-user demand response (DR) for large commercial buildings which involves predicting the demand response baseline, evaluating fixed DR strategies and synthesizing DR control actions for load curtailment in return for a financial reward. Using historical data from the building, we build a family of regression trees and learn data-driven models for predicting the power consumption of the building in real-time. We present a method called DR-Advisor called DR-Advisor, which acts as a recommender system for the building's facilities manager and provides suitable control actions to meet the desired load curtailment while maintaining operations and maximizing the economic reward. We evaluate the performance of DR-Advisor for demand response using data from a real office building and a virtual test-bed.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]

General Terms

Algorithms; Design

Keywords

Cyber Physical Systems; Demand Response; Machine Learning

1. INTRODUCTION

In 2013, the National Climate Assessment report provided evidence that the most recent decade was the nation's warmest on record [16] and it is expected that temperatures are only going to rise. Heat waves in summer and polar vortexes in winter are growing longer in duration and pose increasing challenges to an already over-stressed electric grid.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
BuildSys'15, November 4–5, 2015, Seoul, South Korea..
© 2015 ACM. ISBN 978-1-4503-3981-0/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2821650.2821664.

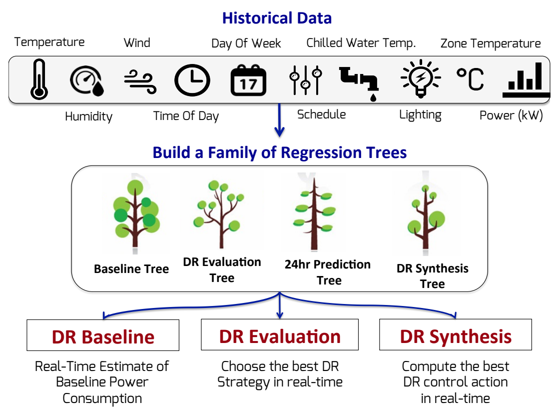

Figure 1: DR-Advisor Architecture

With the increasing penetration of renewable generation, the grid is experiencing a shift from predictable and dispatchable electricity generation to variable generation. This adds another level of uncertainty and volatility to the electricity grid. For e.g., in July 2013, the nominal price of electricity was \$27.34 per MW h but increased to \$672.41 per MW h in the New-England ISO [18]. Such steep inclines of 22-28 times in electricity prices are common during summer months. Demand response and real-time electricity pricing are considered as an agreed upon means of mitigating the uncertainty and volatility of renewable generation and improving the grid's efficiency and reliability.

Across the United States, electric utilities and independent system operators (ISOs) are devoting increasing attention and resources to demand response (DR) [12]. The potential demand response resource contribution from all U.S. demand response programs is estimated to be nearly 72,000 megawatts (MW), or about 9.2 percent of U.S. peak demand [5] making DR the largest virtual generator in the U.S. national grid. The annual revenue to end-users from DR markets with PJM ISO alone is more than \$700 million [15]. Global DR revenue is expected to reach nearly \$40 billion from 2014 through 2023 [21]. The organized electricity markets across the world all use some variant of real-time locational marginal price for wholesale electricity. Locational marginal pricing is a way for wholesale electric energy prices to reflect the value of electric energy at different locations, accounting for the patterns of load, generation, and the physical limits of the transmission system. For e.g., PJM ISO's real-time market is a spot market where electricity prices are calculated at five-minute intervals based on the grid operating conditions.

Electricity costs are the one of the largest components of a large commercial and industrial (C&I) building's operating budget. This is because, such customers are often subject to peak-demand based electricity pricing. In this pricing policy, a customer is charged not only for the amount of electricity it has consumed but also for its peak demand over the billing cycle. High peak loads also lead to a higher cost of production and distribution of electricity. Therefore, these peaks are not only operationally inefficient but also extremely expensive for both the utilities and the end-users. Furthermore, the volatility and variance in real-time electricity rates poses a risk for large buildings [1]. They need the capability to respond to the price volatility in a fast and reliable manner. Such customers are increasingly looking to demand response programs to help manage their electricity costs. DR programs involve a voluntary response of a building to real-time price signals. In such programs, end-users reduce their electricity load during periods of high prices or upon receiving a DR request from the utility and receive a financial reward for their load curtailment.

There are four barriers to successfully enabling real-time building electricity prediction and demand response: (a) Each building is designed and used in a different way and therefore, it has to be uniquely modeled. Learning high fidelity predictive models of buildings using first principles based approaches is very cost and time prohibitive and requires retrofitting the building with several sensors [22]; (b) Secondly, the building's operating conditions, internal thermal disturbances and environmental conditions must be taken into account to make appropriate DR decisions, which is not possible with rule-based and pre-determined demand response strategies since they do not account for the state of the building but are instead based on best practicies and rules of thumbs. (c) Thirdly, upon receiving a notification for a DR event, the building's facilities manager must determine an appropriate DR strategy to achieve the required load curtailment. These control strategies can include adjusting zone temperature set-points, supply air temperature and chilled water temperature set-point, dimming or turning off lights, decreasing duct static pressure set-points and restricting the supply fan operation etc.. In a large building, it is difficult to asses the effect of one control action on other sub-systems and on the building's overall power consumption because the building sub-systems are tightly coupled. (d) Lastly, predictive models for buildings, regardless how sophisticated, can effectively be rendered powerless unless they can be interpreted by human experts. For e.g., artificial neural networks (ANN) obscure physical control knobs and hence, are difficult to interpret by building facilities managers. Therefore, the required solution must be transparent, human centric and highly interpretable.

We present a method called DR-Advisor (Demand Response-Advisor), which acts as a recommender system for the building's facilities manager and provides the power consumption prediction and control actions for meeting the required load curtailment and maximizing the economic reward. Using historical meter and weather data along with set-point and schedule information, DR-Advisor builds a family of interpretable regression trees to learn non-parametric data-driven models for predicting the power consumption of the building (Figure 1). DR-Advisor can be used for real-time

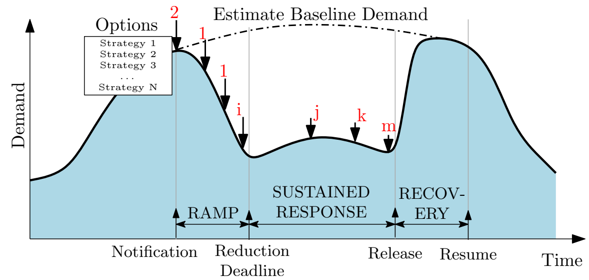

Figure 2: Example of a demand response timeline.

demand response baseline prediction, strategy evaluation and control synthesis, without having to learn first principles based models of the building. This work has the following contributions:

1. We demonstrate the benefit of using regression trees based approaches for demand response: specifically, for estimating the DR baseline power consumption. Using regression tree-based algorithms eliminates the cost of time and effort required to build and tune first principles based high fidelity models of buildings for DR.

2. We present an approach for demand response policy evaluation, which takes into account the state of the building and weather forecasts to help choose the best DR strategy among several pre-determined strategies.

3. We introduce a model based control with regression trees (mbCRT) algorithm to enable control with regression trees use it for real-time DR synthesis.

We evaluate the performance of DR-Advisor on two buildings: (a) a Department of Energy's (DoE) large commercial reference building and (b) a real office building, using actual meteorological data. We have also evaluated the performance of DR-Advisor against other data-driven methods on the data-set from AHRAE's great energy predictor shootout challenge (tech report available at `http://repository.upenn.edu/mlab_papers/75/`). While regression trees are a popular choice for prediction, this is the first time they are used in the context of demand response. This is also the first time regression tree based methods are used for controller synthesis.

This paper is organized as follows: Section 2 describes the challenges with demand response. In Section 3, an overview on learning regression trees is presented. Section 4, presents a new algorithm to perform control with regression trees for DR synthesis. Section 6 presents a comprehensive case study with DR-Advisor using data from a real building. In Section 7, a survey of related work has been presented. We conclude this paper in Section 8 with a summary of our results and a discussion about future directions.

2. PROBLEM DEFINITION

The timeline of a DR event is shown in Figure 2. The main period during which the demand needs to be curtailed is the *sustained response period*. The start of this period, i.e., the time by which the reduction must be achieved, is the *reduction deadline*. Prior to that deadline, an *event notification* is issued, at the notification time. The end of the response period is when the main curtailment is released. The normal operation is resumed during the *recovery period*. The DR event ends at the end of the recovery period.

We focus on three challenging problems of end-user demand response, which are described next.

2.1 DR baseline prediction

The DR baseline is an estimate of the electricity that would have been consumed by a customer in the absence of a demand response event. The measurement and verification of the demand response baseline is the most critical component of any DR program. The baseline is the primary tool for measuring curtailment during a DR event, and determining financial paybacks. DR-Advisor utilizes historical power meter and weather data to estimate the baseline power consumption in real-time during a DR event.

2.2 DR strategy evaluation

Upon receiving a notification for a DR event, the building's facilities manager must choose a control strategy among several pre-determined strategies to achieve the required power curtailment level. Each strategy includes adjusting temperature set-points, lighting levels and temporarily switching off equipment, such as escalators, and plug loads to different levels across different time intervals. As only one strategy can be use at a time, the question then is, *how to choose the DR strategy from a pre-determined set of strategies which leads to the largest load curtailment ?*

In Fig. 2, there are N different strategies available to choose from. DR-Advisor predicts the power consumption of the building due to each strategy at every time-step and chooses the DR strategy which leads to the largest load curtailment.

2.3 DR strategy synthesis

Instead of choosing a DR strategy from a pre-determined set of strategies, one may ask *how to synthesize new DR strategies ?* For example, in the traditional rule-based approaches, determined by prior curtailment experiments and operator experience, the zone temperature set-points of the building should be increased to pre-determined levels to reduce the cooling load. However, based on the state of the building and environmental conditions of the current day, it is unclear by how much and for how long the particular rule-based curtailment will comply to the curtailment requirements? This is the problem of demand response synthesis because we want to synthesize optimal control actions which are suitable for the DR event based on the current state of the building, outside weather and real-time electricity prices.

2.4 Rule-based and model-based DR

The two most popular approaches to respond to DR include rule based and model based DR strategies. In a rule based DR strategy, different levels of curtailment are achieved by following a pre-programmed strategy. Fixed DR strategies have the advantage of being simple but they do not account for the state of the building and weather conditions during a DR event. Despite this lack of predictability, rule-based DR strategies account for the majority of DR approaches.

Model based DR involves mathematically modeling the building in order to predict the overall power consumption and take actions based on the predicted response. Creating and learning such high fidelity models (e.g., with EnergyPlus [7]) is extremely cost and time prohibitive [22]. The user expertise, time, and associated costs required to develop a model of a single building is very high. This is because usually a building modeling domain expert will use a software tool to create the geometry of a building from the building design, add detailed information about material properties, about equipment and operational schedules. There is always a gap between the modeled and the real building and the domain expert must manually tune the model to match measured data [17].

The goal with data-driven methods, such as with DR-Advisor, is to make the best of both worlds; i.e. simplicity of rule based approaches and the predictive capability of model based strategies, but without the expense of first principle or grey-box model development.

3. LEARNING REGRESSION TREES

Data driven modeling consists of obtaining a functional model that relates the value of the response variable Y with the values of the predictor variables $X_1, X_2, \cdots, X_m$. For example in linear regression, a linear form is assumed for the unknown function and the parameters of the model are estimated using a least squares criterion. Predictors like linear or polynomial regression are global models, where a single predictive formula is assumed to hold over the entire data space. When the data has lots of features which interact in complicated, nonlinear ways, assembling a single global model can be difficult, lead to poor response predictions and hopelessly confusing when you do succeed.

An approach to non-linear regression is to partition the data space into smaller regions, where the interactions are more manageable. We then partition the partitions again; this is called recursive partitioning, until finally we get to chunks of the data space which are so tame that we can fit simple models to them. Therefore, the global model has two parts: the recursive partition, and a simple model for each cell of the partition. Regression trees belong to the class of recursive partitioning algorithms. The seminal algorithm for learning regression trees is CART as described in [4]. For more details on how regression trees are built, we direct the reader to ***Appendix A***.

3.1 Boosting and random forests

The problem with regression trees is that they can have high variance and can sometimes overfit the data. It is the price to be paid for estimating a simple model. While pruning and cross validation can help reduce over fitting, we can also use ensemble methods for growing more stable trees. DR-Advisor uses two ensemble methods for building power prediction: boosted regression trees and random forests.

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator. Random forests or tree-bagging are a type of ensemble method which makes predictions by averaging over the predictions of several independent base models. The essential idea is to average many noisy but approximately unbiased trees, and hence reduce the variance. Injecting randomness into the tree construction can happen in many ways. The choice of which dimensions to use as split candidates at each leaf can be randomized, as well as the choice of coefficients for random combinations of features. For a more comprehensive review of random forests we refer the reader to [3]. In boosting, trees are fitted iteratively

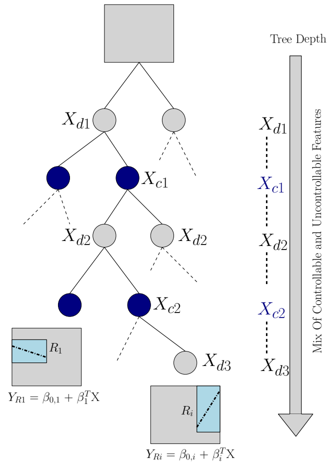

Figure 3: Example of a regression tree with linear regression model in leaves. Not suitable for control due to the mixed order of the controllable X_c (solid blue) and uncontrollable X_d features.

to the training data, gradually improving on the observations modeled poorly by the existing collection of trees. A boosted regression tree (BRT) model can be understood as an additive regression model in which individual terms are simple trees, fitted in a forward, stage-wise fashion [10].

3.2 Model-based regression trees

In model based regression trees, the definition of the leaf of a tree is extended to allow for simple functions, other than averaging, in the leaves which predict the response. The use of linear regression functions in the leaves of the tree, or local linear regression, has been presented in [20] in an algorithm called M5. [11] describes another variant of this idea.

4. DR SYNTHESIS

In this section, we extend the theory of regression trees to incorporate making control decisions rather than just making predictions. This is then used to solve the demand response synthesis problem described earlier.

Recall that the objective of learning a regression tree is to learn a model for predicting the response Y with the values of the predictor variables or features $X_1, X_2, \cdots, X_m$. Given a forecast of the features $\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m$ we can predict the response $\hat{Y}$. Now consider the case where a subset, $\mathbb{X}_c \subset \mathbb{X}$ of the set of features $\mathbb{X}$'s are controllable i.e., we can change their values in order to drive the response $(\hat{Y})$ towards a certain value. In the case of buildings, the set of features can be separated into disturbances (or uncontrollable) variables like outside air temperature, humidity, wind etc. while the controllable variables would be the temperature and lighting set-points within the building.

4.1 Model-based control with trees

The key idea in enabling control is with the separation of features/variables into controllable and uncontrollable features. Let $\mathbb{X}_c \subset \mathbb{X}$ denote the set of controllable variables and $\mathbb{X}_d \subset \mathbb{X}$ denote the set of disturbances such that $\mathbb{X}_c \cup \mathbb{X}_d \equiv \mathbb{X}$. Using this separation of variables we build upon the idea of simple model based regression trees Section 3.2 to model based control with regression trees (mbCRT).

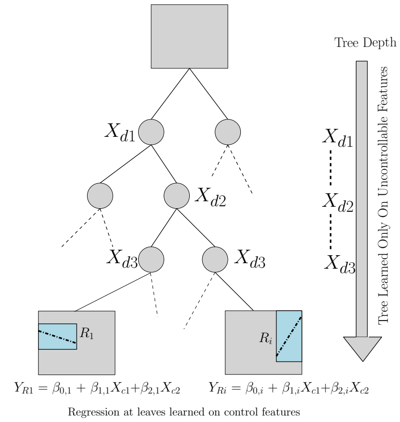

Figure 4: Example of a tree structure obtained using the mbCRT algorithm. The separation of variables allows using the linear model in the leaf to use only control variables.

Figure 3 shows an example of how controllable and uncontrollable features can get distributed at different depths of model based regression tree which uses the following linear regression function in the leaves of the tree:

$$Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X} \tag{1}$$

Where Y_{Ri} is the predicted response in region R_i of the tree using all the features $\mathbb{X}$. Since the controllable and uncontrollable variables appear in a mixed order in the tree depth, we cannot use this tree for control synthesis. In such a tree the prediction can only be obtained if the values of all the features X's is known, including the values of the control variables X_{ci}'s. Since the value of the control variables X_{ci}'s is unknown, one cannot navigate to any single region using the forecasts of disturbances alone.

The mbCRT algorithm avoids this problem using a simple but clever idea. We still partition the entire data space into regions using CART algorithm (Appendix A), but the tree is learned only on the uncontrollable features $\mathbb{X}_d$ as opposed to all the features $\mathbb{X}$ (Figure 4) In every region at the leaves of the *uncontrollable* tree a linear model is fit but only on the control variables $\mathbb{X}_c$:

$$Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c \tag{2}$$

Separation of variables allows us to use the forecast of the disturbances $\hat{\mathbb{X}}_d$ to navigate to the appropriate region R_i and use the linear regression model with only the control features in it. The linear model between the response variable Y_{Ri} and the control features $\mathbb{X}_c$ is assumed for computational simplicity. Other models could also be used at the leaves as long as they adhere to the separation of variables principle. Since the leaf model only has control variables, one can solve the following linear program in real-time to determine the optimal values of the control variables to minimize an objective function of the response variable:

$$\begin{aligned} \underset{\mathbb{X}_c}{\text{minimize}} \quad & f(Y_{Ri}) \\ \text{subject to} \quad & Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c \\ & \mathbb{X}_c \in \mathbb{X}_{safe} \end{aligned} \tag{3}$$

Where $\mathbb{X}_{safe}$ is the user specified safe set of values for the control input $\mathbb{X}_c$ and the objective $f(Y_{Ri})$ is a function of the response variable. For buildings, where the response variable is power consumption, the objective function can denote the financial reward of minimizing the power consumption.

Algorithm 1 mbCRT: Model Based Control With Regression Trees

DESIGN TIME
procedure MODEL TRAINING
  Separation of Variables
  Set $\mathbb{X}_c \leftarrow$ Controllable Features
  Set $\mathbb{X}_d \leftarrow$ Uncontrollable Features
  Build the *uncontrollable* tree T_{mrt} with $\mathbb{X}_d$
  for all Regions R_i at the leaves of T_{mrt} **do**
    Fit linear model $Y_{Ri} = \beta_{0,i} + \beta_i^T \mathbb{X}_c$
  end for
end procedure
RUN TIME
procedure CONTROL SYNTHESIS
  At time t obtain forecast $\hat{\mathbb{X}_d}(t+1)$ of disturbances $\hat{X_{d1}}(t+1), \hat{X_{d2}}(t+1), \cdots$
  Using $\hat{\mathbb{X}_d}(t+1)$ determine the leaf and region R_{rt}
  for Region R_{rt} **do**
    Solve optimization in Eq3 for optimal control action $\mathbb{X}_c^*(t)$
  end for
end procedure

The intuition behind the mbCRT Algorithm 1 is that at run time t, we use the forecast $\hat{\mathbb{X}_d}(t+1)$ of the disturbance features to determine the region of the *uncontrollable* tree and hence, the linear model to be used for the control. We then solve the simple linear program corresponding to that region to obtain the optimal values of the control variables.

5. DATA DESCRIPTION

In order to build a regression tree which can predict the power consumption of the building, we need to train on time-stamped historical data. The data that we use can be divided into three different categories as described below:

5.1 Weather data

Weather data includes measurements of the outside temperature, relative humidity, wind characteristics and solar irradiation at the building site.

5.2 Schedule data

Using time-stamp information in the building power consumption data, we create *proxy* variables which correlate with repeated patterns of electricity consumption e.g., due to occupancy or equipment schedules.

1. **Day of Week:** This is a categorical predictor which takes values from $1-7$ depending on the day of the week. This variable can partition the data space on patterns which occur on specific days of the week. For instance, there could a big auditorium in an office building which is only used on certain days.

2. **Weekends and Holidays:** For most buildings the equipment schedule and occupancy patterns change significantly over weekends and holidays. Weekends, special days and holidays are represented by a single binary predictor which takes the values $\{1, -1\}$.

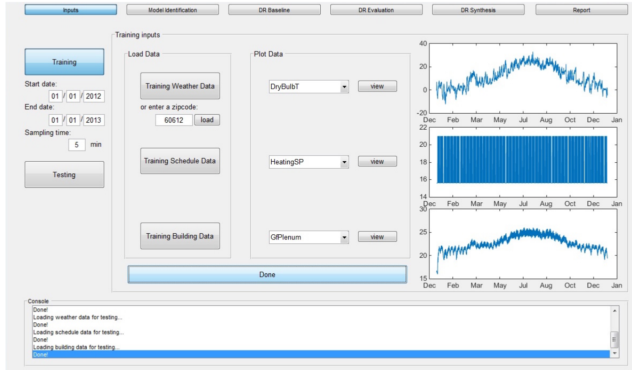

Figure 5: MATLAB GUI for DR-Advisor

3. **Time of Day:** This is quite an important predictor as it can adequately capture daily patterns in power consumption due to occupancy, lighting and appliance use without directly measuring any one of them.

Besides using proxy schedule predictors, actual building schedules can also be used as training data for building the trees. The prime candidate for obtaining actual schedules are temperature set-points schedules of chilled water supply, supply air temperature and zone air temperature on the HVAC side and lighting schedules. Using actual schedule information can greatly improve the accuracy of the power consumption prediction as we shall see later.

5.3 Building data

Lastly, since we are trying to predict the power consumption of the building, we require historical time-stamped power consumption (or meter) data. The power consumption is the response variable of the regression tree i.e., the variable which we want to predict. The state of the building is required for DR strategy evaluation and synthesis. This includes (i) Chilled Water Supply Temperature (ii) Hot Water Supply Temperature (iii) Zone Air Temperature (iv) Supply Air Temperature (v) Lighting levels

6. CASE STUDY

DR-Advisor (Figure 5) is being developed into a toolbox (`http://mlab.seas.upenn.edu/dr-advisor/`). DR-Advisor is also compared against other data-driven methods on the data-set from AHRAE's great energy predictor shootout challenge (tech report available at `http://repository.upenn.edu/mlab_papers/75/`). In this section, we present a comprehensive case study to show how DR-Advisor can be used to address the three demand response challenges (Section 2).

6.1 Building description

Figure 6: Left: Building 101 in Philadelphia; Right: 3D rendering of the DoE commercial reference building in EnergyPlus.

We evaluate the performance of DR-Advisor using data from two different buildings.

Building 101, located in Philadelphia, PA is the temporary headquarters of the U.S. Department of Energy's consortium for buildings energy innovation. It is a three floored building with a gross building floor area of 75,156 sq-ft (Figure 6(Left)). There are a total of 27 conditioned zones in the building served by 3 air handling units. We use actual power meter and weather data for the Building 101 site.

The second building is the DoE Commercial Reference Building (DoE CRB) simulated in EnergyPlus [8]. This virtual test-bed is a large 12 story office building consisting of 73 zones with a total area of $500,000$ sq ft (Figure 6 (Right)). There are $2,397$ people in the building during peak occupancy. During peak load conditions the building can consume up to 1.6 MW of power. For the simulation of the DoE CRB building we use `Actual Meteorological Year` (AMY) data from Chicago for the years 2012 and 2013.

6.2 Model validation

For Building 101, multiple regression trees were trained on weather and power consumption data from the year 2014. Only the weather forecasts (W) (Section 5.1) and proxy variables (P) (Section 5.2) were used to train the models. We then use the DR-Advisor to predict the power consumption in the test period i.e., Feburary 2015. The predictions are obtained in time-steps of 5 minutes. The predictions on the test set are compared to the actual power consumption of the building during the test-set period. This comparison is shown in Figure 7. The following algorithms were evaluated: single regression tree, k-fold cross validated (CV) trees, boosted regression trees (BRT), random forests (RF) and model based regression trees (M5) (Section 3). Our chosen metric of prediction accuracy is the normalized root mean square error (NRMSE). NRMSE is the RMSE divided by the mean of the data. The NRMSE values for Building 101 are listed in Table 1. For this data set the boosted regression tree (NRMSE 3.16%) algorithm and random forests (NRMSE 3.41%) can predict the power consumption with very high accuracy ($\sim 97\%$).

For the DoE CRB building, the tree-based models were trained on data from the year 2012 and the prediction accuracy was evaluated for the entire 2013 year. The values of

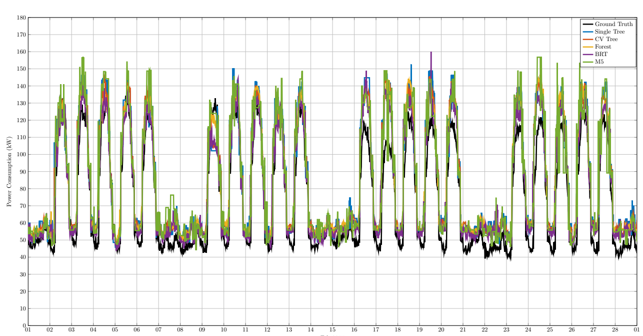

Figure 7: Model validation for Building 101. Comparison between the the actual power consumption of the building (ground truth) and the power consumption prediction obtained from DR-Advisor

Method	NRMSE % (W,P)
Single Tree	4.82
Cross-Validated Tree	3.69
Boosted Regression Tree	**3.16**
Random Forest	3.41
M5 Model Based RT	4.88

Table 1: Comparison of methods on Building 101 data

Method	NRMSE % (W,P)	NRMSE % (W,P,SP)
Single Tree	7.25	4.21
Cross-Validated Tree	6.80	3.98
Boosted Regression Tree	6.71	3.96
Random Forest	**6.04**	**3.86**
M5 Model Based RT	8.12	4.40

Table 2: DoE CRB Model Validation. Training on 2012 data and testing on 2013 data.

the NRMSE are listed in Table 2. Yet again the ensemble methods (BRT and random forest) can predict the baseline power with high accuracy (94%). For k-fold cross validation, instead of training one single tree, we build k different trees, each trained on non-overlapping $k-1$ folds of the training data. During prediction, we obtain k predictions for power consumption, one from each tree and average them to obtain a single prediction. The value of k for the cross validation was chosen to be 20. For the ensemble methods we grow 500 trees each for boosted regression trees and random forests. By growing a large number of trees for the ensemble methods, we can get a good estimate of the importance of the predictor variables as shown in Figure 8.

For DoE CRB, we compared the prediction accuracy of trees trained only on weather (W) and proxy features (P) with trees trained on weather (W), proxy features (P) and set-point schedule (SP) data. Using the results listed in Tables 1 & 2, we can draw the following observations about the use of tree-based methods for building electricity prediction.

Observation 1 *The proxy variables i.e. Time of Day, Day of Week, Day of Month are important predictors of building power consumption. This is because they capture repeated patters of occupancy and building operation.*

Observation 2 *Using set-point schedule information significantly improves the accuracy of the power consumption estimate for regression trees or all types.*

Both observations indicate that it is possible to improve power consumption prediction accuracy without investing in any additional sensors.

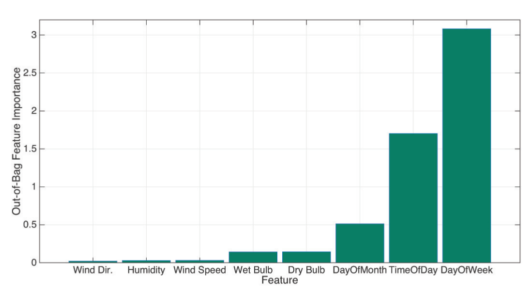

Figure 8: Predictor variables (feature) importance in the random forest ensemble.

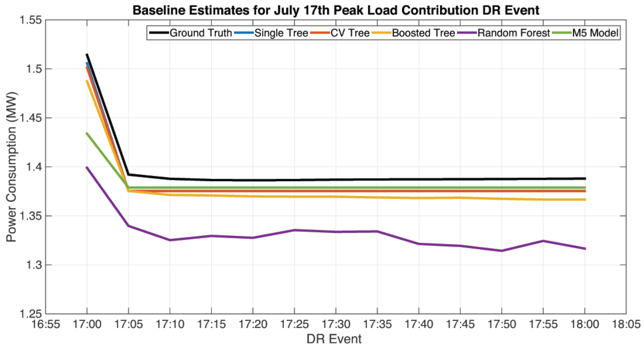

Figure 9: Comparison between the actual power consumption and the baseline prediction. There was a DR event from 1700-1800 hrs.

Method	NRMSE % (W,P)
Single Tree	2.01
Cross-Validated Tree	**1.96**
Boosted Regression Tree	3.11
Random Forest	10.91
M5 Model Based RT	3.88

Table 3: NRMSE for DR Baselining

Building the family of trees is the most computationally intensive step, although it must be performed only once. Training a single regression tree on a year's worth of 5 min interval data with 13 predictor variables take only 4s; a 20 fold cross validated tree takes 48.4s; 723.7s for boosted regression trees; 626.7s for a random forest with 500 trees and 916.3s for the m5 algorithm. However, once the trees are built offline, it only take less than 1s to obtain the predicted power consumption at each time-step. These times were obtained on a 2.6GHz Intel Core i5 machine running MATLAB 2014b.

6.3 DR baseline prediction

The problem of real-time DR baseline prediction is to estimate the power consumption the building would have consumed, if it had not curtailed during a DR event.

In model validation we predict the power consumption of the building using only weather data (W) and proxy variables (P) (and set-point schedules (SP), if available). Estimating the baseline is very similar to model validation.

We estimate the baseline power consumption of the DoE CRB building for the DR event which occurred on July 17, 2013 from 1700 − 1800 hrs at PJM. Since, the building is simulated in EnergyPlus, we can compare our estimate with the ground truth. This allows us to evaluate DR-Advisor's prediction with the ground truth power consumption. The result of this comparison is shown in Figure 9 and summarized in Table 3. Even without any set-point schedule information, the error for most of the trees is quite low ($< 7\%$)

6.4 DR strategy evaluation

DR strategy evaluation involves choosing good DR strategies from several fixed strategies, in real time (Section 2.2). Unlike DR baseline prediction estimates, here we need additional training data in addition to just power consumption, weather (W), proxy variables (P) and set-point schedules (SP). Specifically, data about the state of the building (ST), described in Section 5.3, is also required.

Method	NRMSE % (W,P,SP,ST)
Single Tree	7.14
Cross-Validated Tree	6.84
Boosted Regression Tree	17.10
Random Forest	**6.23**
M5 Model Based RT	11.19

Table 4: NRMSE for DR Strategy Evaluation

Upon receiving the notification of the DR event at 1600 hrs, there are several pre-determined strategies that can be executed by the facilities manager. For simplicity, we only consider strategies in which two set-points are changed, the zone air temperature and the chilled water supply temperature set-point. 4 different pre-determined strategies were considered. In each of the strategy the zone temperature set-point was varied between 24°C and 26°C and the chilled water set-point was varied between 6.5°C and 9°C. DR-Advisor evaluates the predicted response from each fixed strategy and chooses the one which leads to the lowest power consumption as shown in Figure 10. The best rule-based strategy chosen by DR-Advisor is shown in Figure 11. Table 4, shows the NRMSE between the predicted power consumption and the actual building power consumption due to the above strategy. Yet again, random forests provides the lowest NRMSE of only 6.23% for predicting the power consumption of the building.

This shows how DR-Advisor could be used for predicting the power consumption profile of a large commercial building in real-time due to several rule-based pre-determined DR strategies and then choose and recommend the best response to the buildings facilities manager.

6.5 DR strategy synthesis

We now evaluate the performance of the mbCRT (Section 4.1) algorithm for real-time DR synthesis. Similar to DR evaluation, the regression tree is trained on weather (W), proxy (P), set-point schedules (SP) and building state (ST) features. We first partition the set of features into controllable features (or control inputs) and uncontrollable features (or disturbances). There are three control inputs to the system: the chilled water set-point, zone air temperature set-point and lighting levels. At design time, the model based tree built (Algorithm 1) has 138 leaves and each of them has a linear regression model fitted over the control inputs with the response variable being the power consumption of

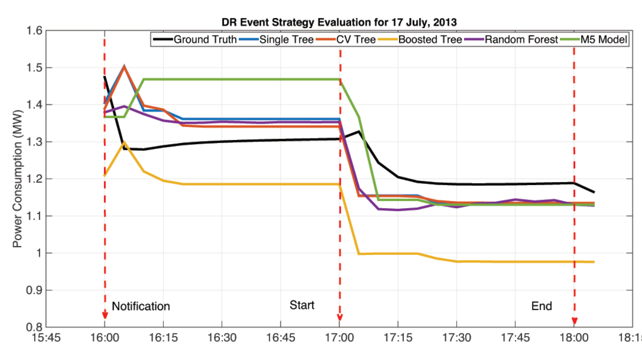

Figure 10: Comparison between the actual power consumption and the predicted power for July 17th, 2013; There is a DR event from 1700-1800 hrs.

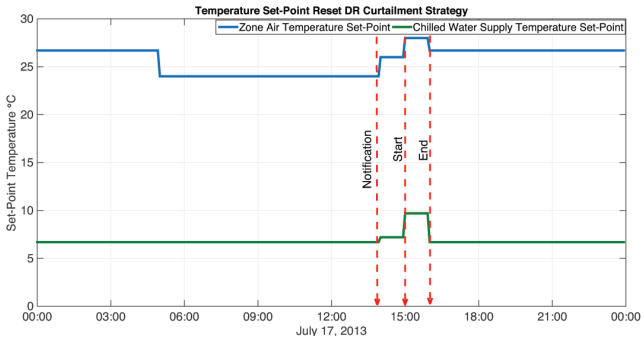

Figure 11: Rule based demand response temperature set-point reset strategy executed for July 17, 2013.

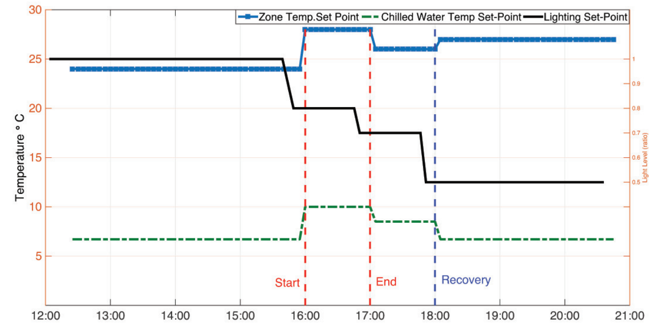

Figure 13: Optimal DR strategy as determined by the mbCRT algorithm.

the building. When the DR event commences, at every time-step (every 5 mins), DR-Advisor uses the mbCRT algorithm to determine which leaf, and therefore, which linear regression model will be used for that time-step to solve the linear program (Eq 3) and determine the optimal values of the control inputs. Figure 12 shows the power consumption profile of the building using DR-Advisor for the DR event. We can see that using the mbCRT algorithm we are able to achieve a sustained curtailed response of $\sim$ 300kW over a period of 1 hour as compared to the baseline power consumption estimate. The DR strategy synthesized by DR-Advisor is shown in Figure 13. Figure 14 shows a close of view of the curtailed response. The index of the linear model used during each time-step by the mbCRT algorithm is also shown. We can see that the model switches several times during the event, based on the forecast of disturbances.

These results show the effectiveness of the mbCRT algorithm to synthesize DR actions in real-time while utilizing a simple data-driven tree-based model.

6.5.1 Revenue from DR

We use Con Edison utility company's commercial demand response tariff structure [6] to estimate the financial reward obtained due to the curtailment achieved by the DR-Advisor for our Chicago based DoE commercial reference building. The utility provides a \$25/kW per month as a reservation incentive to participate in the real-time DR program for summer. In addition to that, a payment of \$1 per kWh of energy curtailed is also paid. For our test-bed, the peak load curtailed is 331kW and a total of 327.4kW h of energy was saved. If we consider $\sim$ 5 such events per month for 4 months, this amounts to a revenue of $\sim$ \$39,700 for participating in DR only for the summer. This is a significant amount, especially since using DR-Advisor does not require an investment in building complex modeling or installing sensor retrofits to a building.

6.5.2 Meeting thermal comfort

In the objective function of the mbCRT algorithm, we only consider minimizing the electricity load during the duration of the DR event while meeting user defined constraints on the control inputs (i.e., set-points). It is assumed that the constraints provided are indicative of the thermal comfort tolerance for the building occupants. However, this might not always be the case, and the facilities manager would want recommendations which ensure a certain thermal comfort level. In this case, the effect of the control strategy on thermal comfort must be accounted for directly in the optimization carried out at the leaves of the tree in the mbCRT algorithm. Instead of fitting a single linear model at each leaf of the tree, we can learn another model for how control variables affect thermal comfort. Such models of thermal comfort have been widely studied [23, 26] in literature. Integrating thermal comfort constraints in order to improve the mbCRT algorithm is left as a task for future work.

7. RELATED WORK

There is a vast amount of literature ([2, 19, 25]) which addresses the problem of demand response strategies under different pricing schemes. The majority of approaches are using either rule-based approaches for curtailment or white/grey box model-based approaches. These usually as-

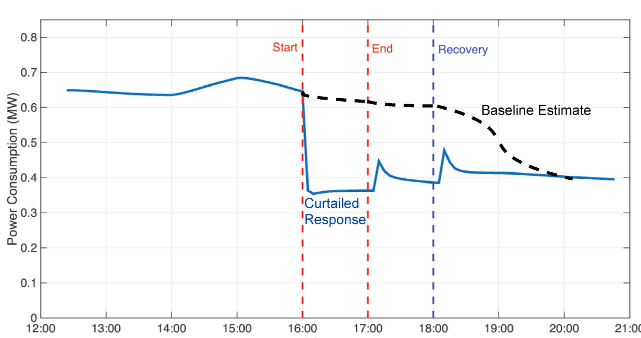

Figure 12: Real-Time DR synthesis using the mbCRT algorithm for July 17, 2013. A curtailemnt of 300kW is sustained during the DR event period.

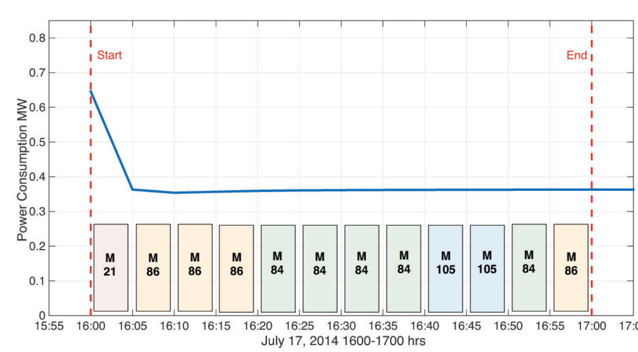

Figure 14: Close up view of the DR synthesis showing how the mbCRT algorithm selects the appropriate linear model for each time-step based on the forecast of the disturbances.

sume that the model of the system is either perfectly known or found in literature, whereas the task is much more complicated and time consuming in case of a real building and sometimes, it can be even more complex and involved than the controller design itself. After several years of work on using first principles based models for demand response, multiple authors [22, 28] have concluded that the biggest hurdle to mass adoption of intelligent building control is the cost and effort required to capture accurate dynamical models of the buildings. Since DR-Advisor only learns an aggregate building level model and combined with the fact that weather forecasts are expected to become cheaper; there is little to no additional sensor cost of implementing the DR-Advisor recommendation system in large buildings. There are ongoing efforts to make tuning and identifying white box models of buildings more autonomous [17]. Figuring out the correct response on a fast time scales (1-5 mins) using just data-driven methods has'nt been adequately addressed before and makes the DR-Advisor approach and tool novel.

Several machine learning approaches [9, 24, 27, 14] have been utilized before for forecasting electricity load. However, there are three significant shortcomings of the work in this area: (a) First, the time-scales at which the load forecasts are generated range from $15-20$ min upto an hour; which is too coarse grained for DR events which only last for at most a couple of hours and for real-time electricity prices which exhibit frequent changes. (b) Secondly, these approaches are not aimed at solving demand response problems but are restricted to long term load forecasting with applications in evaluating building retrofits savings and building energy ratings. (c) Lastly, in these methods, there is no focus on control synthesis or addressing the suitability of the model to be used in control design; whereas the mbCRT algorithm enables the use of regression trees for control synthesis with applications in demand response.

8. CONCLUSION

DR-Advisor, a data-driven method for demand response has been presented. It is being developed into a MATLAB based toolbox (`http://mlab.seas.upenn.edu/dr-advisor/`) We show how regression tree based methods provide an excellent way to predict the power consumption response of a large commercial building while being simple and interpretable. The use of regression trees based methods for demand response evaluation and synthesis based challenges for large scale commercial buildings is novel. The performance of DR-Advisor is evaluated using data from a DOE commercial reference building and from a real office building. DR-Advisor achieves a prediction accuracy of **94-97%** for DR baseline and DR strategy evaluation. We present a model based control with regression trees (mbCRT) algorithm which enables control synthesis using interpretable tree based structures. Using our algorithm DR-Advisor can achieve a sustained curtailment during a DR event. Using a DR pricing structure from Con Edison utility, we estimate a potential revenue of $\sim$ $**40,000** for the DoE reference building over one summer. The primary advantage of DR-Advisor is that it bypasses cost and time prohibitive process of building high fidelity grey/white box models of buildings. Our ongoing work involves extending the mbCRT algorithm to account for thermal comfort and use DR-Advisor from the perspective of the utility company for optimal DR dispatch. We are also extending the algorithm to operate in a finite receding horizon manner as opposed to a one-step look-ahead optimization.

9. ACKNOWLEDGMENTS

This work was supported in part by STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA and by NSF MRI-0923518 grants.

10. REFERENCES

[1] Energy price risk and the sustainability of demand side supply chains. *Applied Energy*, 123(0):327 – 334, 2014.

[2] D. Auslander, D. Caramagno, D. Culler, T. Jones, A. Krioukov, M. Sankur, J. Taneja, J. Trager, S. Kiliccote, R. Yin, et al. Deep demand response: The case study of the citris building at the university of california-berkeley.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[5] F. E. R. Commission et al. Assessment of demand response and advanced metering. 2012.

[6] Con Edison. Demand response programs details.

[7] D. B. Crawley, L. K. Lawrie, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):319 – 331, 2001.

[8] M. Deru, K. Field, D. Studer, et al. U.s. department of energy commercial reference building models of the national building stock. 2010.

[9] R. E. Edwards, J. New, and L. E. Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49:591–603, 2012.

[10] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

[11] J. H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

[12] C. Goldman. Coordination of energy efficiency and demand response. *Lawrence Berkeley National Laboratory*, 2010.

[13] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[14] T. Hong, W.-K. Chang, and H.-W. Lin. A fresh look at weather impact on peak electricity demand and energy use of buildings using 30-year actual weather data. *Applied Energy*, 111:333–350, 2013.

[15] P. Interconnection. 2014 deamand response operations markets activity report. 2014.

[16] J. M. Melillo, T. Richmond, and G. W. Yohe. Climate change impacts in the united states: the third national climate assessment. *US Global change research program*, 841, 2014.

[17] J. R. New, J. Sanyal, M. Bhandari, and S. Shrestha. Autotune e+ building energy models. *Proceedings of the 5th National SimBuild of IBPSA-USA*, 2012.

[18] New-England ISO. Real-time maps and charts, 2013.

[19] F. Oldewurtel, D. Sturzenegger, G. Andersson, M. Morari, and R. S. Smith. Towards a standardized building assessment for demand response. In *Decision and Control (CDC), IEEE 52nd Annual Conference on.* IEEE, 2013.

[20] J. R. Quinlan et al. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, pages 343–348. Singapore, 1992.

[21] N. Research. Demand response for commercial & industrial markets market players and dynamics, key technologies, competitive overview, and global market forecasts. 2015.

[22] D. Sturzenegger, D. Gyalistras, M. Morari, and R. Smith. Model predictive climate control of a swiss office building: Implementation, results, and cost-benefit analysis. *Control Systems Technology, IEEE Transactions on*, 2015.

[23] M. Taleghani, M. Tenpierik, S. Kurvers, and A. van den Dobbelsteen. A review into thermal comfort in buildings. *Renewable and Sustainable Energy Reviews*, 26:201–215, 2013.

[24] A. Vaghefi, M. Jafari, E. Bisse, Y. Lu, and J. Brouwer. Modeling and forecasting of cooling and electricity load demand. *Applied Energy*, 136:186–196, 2014.

[25] P. Xu, P. Haves, M. A. Piette, and J. Braun. Peak demand reduction from pre-cooling with zone temperature reset in an office building. *Berkeley National Laboratory*, 2004.

[26] L. Yang, H. Yan, and J. C. Lam. Thermal comfort and building energy consumption implications–a review. *Applied Energy*, 115:164–173, 2014.

[27] W. Yin, Y. Simmhan, and V. K. Prasanna. Scalable regression tree learning on hadoop using openplanet. In *Proceedings of third international workshop on MapReduce and its Applications Date*, pages 57–64. ACM, 2012.

[28] E. Žáčeková, Z. Váňa, and J. Cigler. Towards the real-life implementation of mpc for an office building: Identification issues. *Applied Energy*, 135:53–62, 2014.

APPENDIX

A. BUILDING REGRESSION TREES

We explain how regression trees are built using an example adapted from [13]. Tree-based methods partition the feature space into a set of rectangles (more formally, hyper-rectangles) and then fit a simple model in each one. They are conceptually simple yet powerful. Let us consider a regression problem with continuous response Y and inputs X_1 and X_2, each taking values in the unit interval. The top left plot of Figure 15 shows a partition of the feature space by lines that are parallel to the coordinate axes. In each partition element we can model Y with a different constant. However, there is a problem: although each partitioning line has a simple description like $X_1 = k$, some of the resulting regions are complicated to describe. To simplify things, we can restrict ourselves to only consider recursive binary partitions, like the ones shown in the top right plot of Figure 15. We first split the space into two regions, and model the response by the mean of Y in each region. We choose the variable and split-point to achieve the best prediction of Y. Then one or both of these regions are split into two more regions, and this process is continued, until some stopping rule is applied. This is the recursive partitioning part of the algorithm. For example, in the top right plot of Figure 15, we first split at $X_1 = t_1$. Then the region $X_1 \leq t_1$ is split at $X_2 = t_2$ and the region $X_1 > t_1$ is split at $X_1 = t_3$. Finally, the region $X_1 > t_3$ is split at $X_2 = t_4$. The result of this process is a partition of the data-space into the five regions $R_1, R_2, \cdots, R_5$. The corresponding regression tree

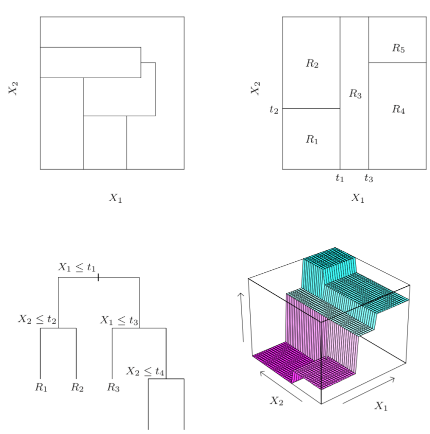

Figure 15: Top right: 2D feature space by recursive binary splitting. Top left: partition that cannot be obtained from recursive binary splitting. Bottom left: tree corresponding to the partition. Bottom right: perspective plot of the prediction surface.

model predicts Y with a constant c_i in region R_i i.e.,,

$$\hat{T}(X) = \sum_{i=1}^{5} c_i I\left\{(X_1, X_2) \in R_i\right\} \tag{4}$$

This same model can be represented by the binary tree shown in the bottom left of Figure 15. The full data-set sits at the top or the root of the tree. Observations satisfying the condition at each node are assigned to the left branch, and the others to the right branch. The terminal nodes or leaves of the tree correspond to the regions $R_1, R_2, ..., R_5$.

A.1 Node splitting criteria

For regression trees we adopt the sum of squares as our splitting criteria i.e a variable at a node will be split if it minimizes the following sum of squares between the predicted response and the actual output variable.

$$\sum (y_i - \hat{T}(x_i))^2 \tag{5}$$

It is easy to see that the best response c_i (from equation 4 for y_i from partition R_i is just the average of output samples in the region R_i i.e

$$c_i = avg(y_i | x_i \in R_i) \tag{6}$$

Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible. A greedy algorithm is used instead. Starting with all of the data, consider a splitting variable j and split point s, and define the following pair of left (R_L) and right (R_R) half-planes

$$\begin{aligned} R_L(j,s) &= \left\{X | X_j \leq s\right\}, \\ R_R(j,s) &= \left\{X | X_j > s\right\} \end{aligned} \tag{7}$$

The splitting variable j and the split point s is obtained by solving the following minimization:

$$\min_{j,s} \left[\min_{c_L} \sum_{x_i \in R_L(j,s)} (y_i - c_L)^2 + \min_{c_R} \sum_{x_i \in R_R(j,s)} (y_i - c_R)^2 \right] \tag{8}$$

where, for any choice of j and s, the inner minimization in equation 8 is solved using

$$\begin{aligned} c_L &= \mathrm{avg}(y_i | x_i \in R_L(j,s)) \\ c_R &= \mathrm{avg}(y_i | x_i \in R_R(j,s)) \end{aligned} \tag{9}$$

For each splitting variable X_j, the determination of the split point s can be done very quickly and hence by scanning through all of the inputs (X_i's), the determination of the best pair (j, s) is feasible. Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions.

A.2 Stopping criteria and pruning

Every recursive algorithm needs to know when it's done. For regression trees this means when to stop splitting the nodes. A very large tree might over fit the data, while a small tree might not capture the important structure. A preferred, strategy is to grow a large tree, stopping the splitting process only when some minimum number of data points at a node (`MinLeaf`) is reached. Then this large tree is pruned using cost-complexity pruning methods.