IET Cyber-Physical Systems: Theory & Applications

Research Article

IET Journals
The Institution of Engineering and Technology

Smart building uncertainty analysis via adaptive Lasso

ISSN 2398-3396
Received on 24th January 2017
Revised on 1st February 2017
Accepted on 10th February 2017
doi: 10.1049/iet-cps.2017.0011
www.ietdl.org

Bingqing Lin[1], *Bei Yu*[2] ✉

[1]*College of Mathematics and Statistics, Shenzhen University, Shenzhen 518060, Guangdong, People's Republic of China*
[2]*CSE Department, The Chinese University of Hong Kong, Shatin, NT, Hong Kong*
✉ *E-mail: byu@cse.cuhk.edu.hk*

Abstract: Uncertainty analysis plays a pivotal role in identifying the important parameters affecting building energy consumption and estimate their effects at the early design stages. In this work, we consider the adaptive Lasso for uncertainty analysis in building performance simulation. This procedure has several appealing features: (1) We can introduce a large number of possible physical and environmental parameters at the initial stage to obtain a more complete picture of the building energy consumption. (2) The procedure could automatically select parameters and estimate influences simultaneously and no prior knowledge is required. (3) Due to computational efficiency of the procedure, non-linear relationship between the building performance and the input parameters could be accommodated. (4) The proposed adaptive Lasso can use a small number of samples to achieve high modeling accuracy and further reduce the huge computational cost of running building energy simulation programs. Furthermore, we propose a stable algorithm to rank input parameters to better identify important input parameters that affect energy consumption. A case study shows the superior performance of the procedure compared with LS and OMP in terms of modeling accuracy and computational cost.

1 Introduction

Smart buildings are able to offer numerous opportunities to improve the energy efficiency and comfort level by efficient temperature and energy control systems, thus currently are playing an important role in the everyday lives of people providing a more comfortable and smart environment. As people's demands such as the comfort level and energy efficiency increase speedily, advanced technologies greatly promote the development of smart buildings. On the other hand, building stock, including commercial and residential buildings, is a major energy consumption entity in electricity market. As estimated by the Department of Energy, buildings account for 40% of the total energy consumption and 70% of electricity usage, and also contribute to 40% of greenhouse gas emission in the United States [1]. The global energy demand of buildings keeps upward in recent years, mainly due to the growth in the population and the increasing demand for building services and conform levels [2].

To achieve a high energy efficiency and provide a high energy comfort level, an accuracy temperature and energy model predicting in-building temperature and energy consumption is imperative. With an accuracy modelling, sensitivity and uncertainty analysis is able to conduct identifying the most important factors that influence building performance most. Nowadays, building simulation is widely used at different stages to predict the thermal performance and energy at the building design stage [3]. Typically, the inputs of a building simulation program include the building system and components, the climate, internal gains from lighting, equipments and occupants, heating and cooling systems, schedules of occupants, equipments and lighting and so on. The simulator runs the heat balance algorithm and then predicts the energy and temperature of the building [3]. Through building simulation, we can achieve many advantages for designing energy-efficient smart buildings.

In addition, it is of importance to quantify the impact of these uncertainties on the energy and thermal performance by a probabilistic approach. Such uncertainty analysis is usually accompanied with a sensitivity analysis, which aims to identify the most important parameters contributing to the uncertainties of the building performance. In this way, we can obtain the key parameters that influence the building performance most.

Many building uncertainty and sensitivity analyses are proposed, which can be broadly divided into two categories [4]: the first one is local analysis that focuses on the effects of uncertain parameters around nominal values; and the second one is global analysis that focuses on the influences of uncertain parameters over the entire variation space. Compared with local analysis, global analysis usually is with higher reliability due to the global view searching variation space. Linear regression methods are widely utilised in global analysis [5–8]. Zhao *et al.* [9] applied recurrent neural network (RNN) to build non-linear compact thermal model of the building. Liu *et al.* [10] employed cross-validation (CV) technique to search for appropriate model order. However, since a building performance can be impacted by thousands of parameters, conventional regression may result in a high-dimensional regression model, which may be expensive in terms of computational cost.

To reduce building model complexity, Rahni *et al.* [11] proposed cluster-based method, which groups uncertainty parameters with similar sensitivity magnitude or sharing the same sign of sensitivity, and thus depends on the prior knowledge. Recently, Chen *et al.* [12] proposed sparse regression method based on orthogonal matching pursuit (OMP) [13], which aims to solve a large number of model coefficients from a small set of simulation samples without over-fitting. It is a two-step procedure: (i) apply OMP to select a set of parameters; (ii) estimate coefficients of selected parameters by least-squares (LS). However, OMP-based method may suffer from runtime overhead problem thus cannot be applied in practical large scale building modelling cases.

To overcome the limitation of previous sparse regression methods, in this work we consider the adaptive Lasso [14] for uncertainty analysis in building performance simulation. Our key contributions are as follows:

- Our method is able to automatically select parameters and estimate coefficients simultaneously and no prior knowledge is required.

- Due to computational efficiency of the procedure, non-linear relationship between the building performance and the input parameters could be accommodated by including quadratic or higher order polynomial terms in the uncertainty analysis.
- The number of samples is allowed to be less than the number of input parameters, the adaptive Lasso can use a small number of samples to achieve high modelling accuracy and further reduce the huge computational cost of running building energy simulation programs.
- We further propose a stable algorithm to rank input parameters to better identify important input parameters that affect energy consumption.

The remaining of this paper is organised as follows. In Section 2, we provide some preliminaries and the problem formulation. In Section 3, we propose an adaptive Lasso based uncertainty analysis. In Section 4, we report the experimental results, followed by the conclusion in Section 5.

2 Preliminary

In general, an uncertainty analysis can be divided into three steps: pre-processing, simulation and post-processing [5]. In the pre-processing, input parameters (e.g. building geometry, fenestration and building materials amongst others) that may affect the energy consumption of a building and the associated probability distributions are determined. Let $\boldsymbol{x} = (x_1, \ldots, x_p)$ be a vector of input parameters of a building. Random samples $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$ are generated by Latin hypercube sampling method [15], where N is the number of samples. Each sample is independently ran on building energy simulation programs (e.g. EnergyPlus [16], ESP-r [17] etc.) to quantify the building performance. In this paper, we consider the annual energy consumption. In the post-processing, uncertainty analysis is implemented to identify the key parameters that affect the building energy consumption and gain insight the relationship between these input parameters and the building performance.

Let $\boldsymbol{y} = (y_1, \ldots, y_N)$ be the vector of building performance generated by building energy simulation programs and $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)})$ be the design matrix, i.e. $X_{ij} = x_j^{(i)}$. We consider the following linear regression model

$$y_i = \sum_{j=0}^{p} \beta_j X_{ij}, \tag{1}$$

where X_{i0} equals to 1. The linear regression model is widely used as a global uncertainty analysis tool in the literature [4, 6]. To make the coefficients comparable, and without loss of generality, we can assume that the building performance is centred and the parameters are standardised

$$\sum_{i=1}^{N} y_i = 0, \quad \sum_{i=1}^{N} X_{ij} = 0 \quad \text{and} \quad \sum_{i=1}^{N} X_{ij}^2 = N,$$

for $j = 1, 2, \ldots, p$. The linear regression model (1) is equivalent to

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ contains the model coefficients. The effects due to the scale of the input parameters are eliminated and these coefficients are directly comparable, i.e. magnitudes of the coefficients representing the influences of parameters to the response. The method of LS provides estimates of the coefficients by minimising the LS objective function

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2,$$

where $\|\cdot\|_2$ denotes the L_2-norm. However, LS method has its drawbacks. First, if the number of samples N is less than the number of parameters p, the LS estimates are not unique. In fact, there is an infinite number of solutions that make the objective function equal to zero and almost surely overfit the dataset [18]. Thus, when we consider a large number of input parameters, a large number of samples should be generated. However, it is often time consuming to run the building energy simulation programs. This makes the whole uncertainty analysis process computationally intensive. Second, due to low bias, but the large variance of the LS estimate and the accumulation of noise, prediction accuracy of the final model is often compromised in practice. Finally, all of the LS estimates are typically not zero, which makes it more challenging to identify the key input parameters and characterise the precise contribution of each input parameter to the building energy performance.

To make the uncertainty analysis process computationally efficient and more reliable in analysing the influence of input parameters, it is essentially important to reduce the dimension of the parameters. The methods for reducing dimensionality in the uncertainty analysis can be divided into two categories: cluster-based methods and variable selection methods. Cluster-based methods focus on grouping uncertainty parameters with similar sensitivity magnitude or sharing the same sign of sensitivity. The cluster-based methods are straightforward and easy to implement. However, grouping the parameters depends on the prior knowledge that is usually unavailable at the early stage of planning. On the other hand, variable selection typically does not need prior knowledge. In practice, there are only a small number of uncertainty parameters that affect the building energy performance. In such cases, the p-dimensional parameters are assumed to be sparse with many coefficients being zero. Non-zero coefficients indicate the importance of the corresponding parameters. Under the assumption of sparsity, variable selection can improve the prediction accuracy by reducing the variance and enhance the ability of uncertainty analysis by automatically identifying the important parameters. Besides, variable selection methods allow $N < p$. This alleviates the need of large samples as LS estimate, and thus, reduces the computational cost in the uncertainty analysis.

3 Uncertainty analysis by adaptive Lasso

In this section, we consider the adaptive Lasso [14] that minimises the penalised LS objective function taking the form

$$\arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=j}^{p} w_j |\boldsymbol{\beta}_j|, \tag{3}$$

where λ is a non-negative regularisation parameter and $\boldsymbol{w} = (w_1, \ldots, w_p)$ is a known weight vector. When $\boldsymbol{w} = 1$, the second term in (3) is the so-called 'ℓ_1 penalty'. The penalised LS objective function (3) becomes the Lasso [19]. Both methods have been widely used for simultaneously choosing important parameters and estimating their effects in high-dimensional statistical inference.

In practice, weight w_j is usually set to be absolute of the inverse of an initial estimator, i.e. $w_j = 1/|\hat{\beta}_{j,\text{init}}|$. We propose to use the estimator of Lasso $\hat{\beta}_{\text{Lasso}}$ as the initial estimator. If $\hat{\beta}_{j,\text{Lasso}} \neq 0$, we set $w_j = 1/|\hat{\beta}_{j,\text{Lasso}}|$; if $\hat{\beta}_{j,\text{Lasso}} = 0$, we set $\hat{\beta}_{j,\text{adapt}} = 0$. Furthermore, if $|\hat{\beta}_{j,\text{asso}}|$ is large, the adaptive Lasso employs a small penalty for the coefficient β_j, this means the coefficient β_j could enjoy less bias compared with Lasso. Conversely, if $|\hat{\beta}_{j,\text{Lasso}}|$ is small, the adaptive Lasso tends to shrink the coefficient β_j to zero, this yields a more sparse solution and reduces the number of unimportant parameters compared with Lasso. Figs. 1 and 2 show the estimates of regression coefficients $\hat{\beta}_j, j = 1, \ldots, 4$, for a simple simulated example by Lasso and the

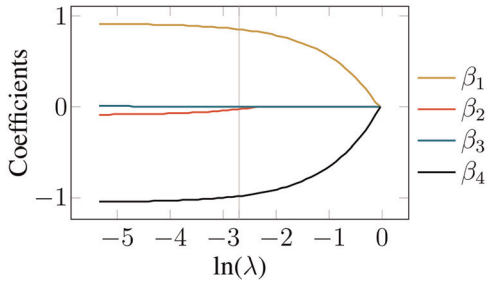

Fig. 1 *Estimates of regression coefficients $\hat{\beta}_j, j = 1, \ldots, 4$, for a simple simulated example by Lasso and the adaptive Lasso. Lasso estimates as a function of ln(λ). Coefficients shrink as λ increases*

adaptive Lasso. At the first stage, β_3 is identified as 0 and the estimates of coefficients for the remainder three parameters are $\hat{\beta}_1 = 0.87$, $\hat{\beta}_2 = -0.04$ and $\hat{\beta}_4 = -0.99$, respectively. Therefore, we set $\hat{\beta}_{3,\text{adapt}} = 0$ and $w_1 = 1/0.87$, $w_2 = 1/0.04$ and $w_4 = 1/0.99$. At the second stage, it is easier to shrink β_2 to 0, since w_2 is very big, while the adaptive Lasso only shrink β_1 and β_4 slightly since the weights for these two parameters are relatively small.

The simple dataset is generated from linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where true coefficients of this simulated example is $\boldsymbol{\beta} = (1, 0, 0, -1)$, the design matrix is generated from standard normal distribution independently and the error $\boldsymbol{\epsilon}$ is also generated from standard normal distribution. We let $N = 100$.

The regularisation parameter λ controls the tradeoff between the LS loss function and penalty. When $\lambda = 0$, all coefficients are typically non-zero and the model is unidentifiable when $p > N$. When $\lambda = \infty$, all coefficients are set to zero, which indicates that none of the input parameters is important. This indicates that λ governs the complexity of the model: the smaller values of λ tends to free up more parameters and the larger values of λ tends to restrict the parameters more and lead to a sparser model. An optimal choice of λ could balance the tradeoff between biases and variances and yield a small generalised mean squared error. CV is often adopted to select the optimal λ. The CV procedure will be discussed in detail later.

3.1 Computations: a two-stage procedure

Given the weights, the adaptive Lasso is essentially a convex optimisation with a weighted ℓ_1 constraint and it has the same order of computational complexity as Lasso [20]. We will describe a two-stage procedure for minimising (3): firstly, we set $\boldsymbol{w} = 1$ and optimise (3) by pathwise coordinate optimisation algorithm [21]; and secondly, given the initiate estimator, we transform the adaptive Lasso to a Lasso problem to simultaneously select the important uncertainty parameters and estimate coefficients [14].

If $p < N$, one could define the weights vector as the inverse of the absolute of LS solutions. However, in uncertainty analysis, it is time-consuming to generate samples and run the build energy simulation programs. To reduce the computational cost, we propose to define the weights vector by the Lasso with penalty parameter chosen by ten-fold CV, when the number of uncertainty parameters is large. Moreover, it is computationally efficient to solve Lasso problem by pathwise coordinate optimisation. The computational cost is the same order of computation of a single LS fit.

Given $\hat{\boldsymbol{w}}$, define $\beta_j^* = \hat{w}_j|\beta_j|$. We can rewrite the objective function (3)

$$\arg\min_{\boldsymbol{\beta}^*} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} X_{ij}^* \beta_j^* \right)^2 + \lambda \sum_{i=j}^{p} |\beta_j^*|, \tag{4}$$

where $X_{ij}^* = X_{ij}/\hat{w}_j$. Pathwise coordinate optimisation can be adopted to solve the objective function (4). The estimates of adaptive Lasso are defined by $\hat{\boldsymbol{\beta}}_{\text{adapt}} = \hat{\boldsymbol{\beta}}^*/\hat{\boldsymbol{w}}$. The details of the two-stage procedure are illustrated in Algorithm 1.

Algorithm 1: A two-stage procedure for the adaptive Lasso
Require: $\boldsymbol{X}$, input matrix; $\boldsymbol{y}$, building performance; λ, regularisation parameter.
Ensure: An estimate of $\boldsymbol{\beta}$ for given λ.

1: Solve the Lasso problem

$$\hat{\boldsymbol{\beta}}_{\text{init}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=j}^{p} |\beta_j|; \tag{5}$$

2: If $\hat{\beta}_{j,\text{init}} \neq 0$, define $\hat{w}_j = 1/|\hat{\beta}_{j,\text{init}}|$ and $X_j^* = X_j/\hat{w}_j$; otherwise, $\beta_{j,\text{adapt}}^* = 0$ and $X_j^* = 0$;
3: Solve the Lasso problem

$$\hat{\beta}_{\text{adapt}}^* = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}^*\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=j}^{p} |\beta_j|; \tag{6}$$

4: Set $\hat{\boldsymbol{\beta}}_{j,\text{adapt}} = \hat{\boldsymbol{\beta}}_{\text{adapt}}^*/\hat{w}_j$, if $\hat{\beta}_{j,\text{init}} \neq 0$; otherwise $\hat{\boldsymbol{\beta}}_{j,\text{adapt}} = 0$, if $\hat{\beta}_{j,\text{init}} = 0$.

3.2 Choice of regularisation parameter

The regularisation parameter λ controls the tradeoff between the data fitting and model complexity. A larger value of λ tends to select fewer uncertainty parameters and the model may underfit the data, whereas a smaller value of λ tends to select more uncertainty parameters and the model may overfit the data. In our case study, we adopt ten-fold CV method to select λ for both Lasso and adaptive Lasso. The flows of CV method are presented in Algorithm 2 (see Fig. 3).

Another practical issue about selecting the regularisation parameter λ is how to general candidates of λ for CV. We first find the range of candidates for λ, $[\lambda_{\min}, \lambda_{\max}]$. The value of $\lambda_{\max}$ is set to be $\max_j |(1/N)\boldsymbol{x}_j\boldsymbol{y}|$ that would let all coefficients be zero, while the value of $\lambda_{\min}$ be $\delta \cdot \lambda_{\max}$. Let δ be 0.0001 or 0.01 for the situation $p < N$ or $p > N$, respectively. We then generate 100 candidates within $[\lambda_{\min}, \lambda_{\max}]$ such that they are evenly distributed in the log scale.

3.3 Stable ranking of input parameters

In addition to build an improved prediction model, it is also critical to know important input parameters that affect the energy consumption, so that building designer is able to create energy-efficient smart building by optimising energy consumption at the early stages.

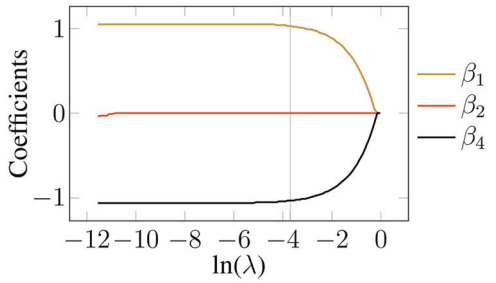

Fig. 2 *Estimates of regression coefficients $\hat{\beta}_j, j = 1, \ldots, 4$, for a simple simulated example by Lasso and the adaptive Lasso. The same plot for the adaptive Lasso. The vertical dotted line indicates the logarithm of λ selected by CV*

Algorithm 2

Require: **X**, input matrix; **y**, building performance; K, number of folds for cross-validation; $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_M)$, a set of regularisation parameters.

Ensure: an optimal λ for the adaptive Lasso.

1: Divide samples into K roughly equal-sized parts denoted as $C_1, C_2, \ldots, C_k$, where C_k denotes the indices of the samples in part k. Denote $M = |\Lambda|$, i.e. the number of regularization parameters.
2: **for** $k = 1$ to K **do**
3: **for** $m = 1$ to M **do**
4: Find the estimate $\hat{\boldsymbol{\beta}}^{k,\lambda_m}_{\text{adapt}}$ by Algorithm 1 for λ_m from samples with part k removed.
5: Compute the model error

$$\text{ME}_{k,\lambda_m} = \frac{1}{|C_k|}\sum_{i\in C_k}(y_i - \sum_{j=1}^{p} X_{ij}\hat{\beta}^{(k,\lambda_m)}_{j,\text{adapt}})^2.$$

6: **end for**
7: **end for**
8: Average the computed modeling error for each λ, i.e., $\text{ME}_{\lambda_m} = \frac{1}{K}\sum_{k=1}^{K}\text{ME}_{k,\lambda_m}$
9: Select λ with the smallest modeling error.

Fig. 3 *Select optimal λ by CV*

If there are B training datasets generated from the same building simulated setting, different models may be identified by the same variable selection algorithm for different datasets. It is natural to assume that important input parameters can be included in most models. On the other hand, unimportant input parameters could only be randomly included in few models. Thus, unimportant input parameters are included in the B models with very low probability. However, it is time-consuming to generate luxury B training sets in reality. In practice, we only have one data set at hand. Moreover, in order to reduce the total computational time, the sample size of the data set is usually small.

Here, we propose a stable ranking algorithm based on bootstrap and the adaptive Lasso (Sadapt). We draw B bootstrap datasets with replacement from the original training set. Let S_b denotes the model selected by the adaptive Lasso in bth bootstrap dataset. Let indicator function $I(\boldsymbol{x}_j \in S_b)$ be 1 if input parameter $\boldsymbol{x}_j$ contained in S_b and 0 otherwise. For every input parameter $j = 1, \ldots, p$, the relative frequency is given by

$$\frac{\sum_{b=1}^{B} I(\boldsymbol{x}_j \in S_b)}{B}.$$

The details of Sadapt are illustrated in Algorithm 3.

Algorithm 3: Sadapt: Ranking the input parameters
Require: $\boldsymbol{X}$, input matrix; $\boldsymbol{y}$, building performance; B, number of bootstrap datasets.

1: **for** $k = 1$ to B **do**
2: Generate a bootstrap dataset with replacement.
3: Select a set of variables, S_b, via the adaptive Lasso with CV method.
4: **end for**
5: Compute the relative frequency for every input parameter $j = 1, \ldots, p$

$$r_j = \frac{\sum_{b=1}^{B} I(\boldsymbol{x}_j \in S_b)}{B}.$$

6: Sort the r_js in a decreasing order.

4 Case study

4.1 Simulation setup

In this section, the empirical efficiency for uncertainty analysis of the adaptive Lasso algorithm is assessed by a building simulation data. The simulation setup was used in [12]. A ten-storey building is created and each storey is divided into nine zones. We assume the temperature is the same within each zone and different zones can have different temperatures. We consider $p = 1106$ input parameters as candidate uncertainties that may affect the energy consumption and the temperature of the building. Then, 8000 independent samples are generated by Latin hypercube sampling method [15]. The building is modelled using the energy calculation software EnergyPlus [16] performed on an Intel core i7 cpu 3.6 GHz with 16 GB memory. The data are split randomly into two disjoint groups with 6000 samples as training set $D_t = \{\boldsymbol{X}_t, \boldsymbol{y}_t\}$ and the remaining 2000 samples as validation set $D_v = \{\boldsymbol{X}_v, \boldsymbol{y}_v\}$.

To test the performance of the adaptive Lasso (adapt) procedure, we compare it with the following three global uncertainty analysis algorithms:

- Classic linear regression (LS). The coefficients of parameters are provided by the LS equation function

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y}.$$

- OMP, a greedy algorithm for sparse approximation of the ℓ_0-norm regularisation method

$$\text{minimise} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|, \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \le \lambda,$$

where $\|\boldsymbol{\beta}\|_0$ denotes the ℓ_0 norm of a vector and is defined as $\|\boldsymbol{\beta}\|_0 = \sum_{i=1}^{p} I_{\{\beta_i=0\}}$. OMP does a forward search; in each step, it adds the most correlated parameter with the residual of the current model and the residual is updated by LS after each step. As in [12], five-fold CV is used to find the regularisation parameter λ. Under the assumption that only a small number of parameters significantly affect the building performance, we often choose a maximum value of λ to reduce the computational cost, especially when $N < p$. In our experiments, we let $\lambda_{\max} = \min(N, p)/2$.
- The least absolute shrinkage and selection operator (Lasso) is a regularisation technique by setting $\boldsymbol{w} = 1$ in the objective function (3). Lasso was implemented by R package 'glmnet' [21]. Ten-fold CV is used to find the regularisation parameter λ in Lasso.

4.2 Comparisons of the algorithms on modelling error and computational cost

In what following, we examine different aspects of these algorithms, including computational cost, number of selected parameters and

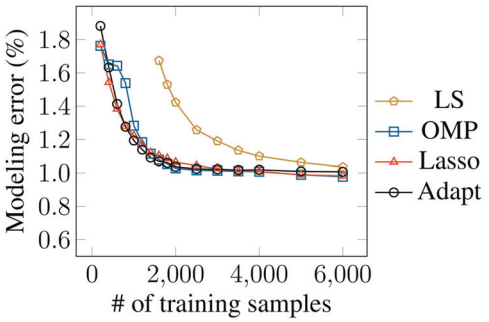

Fig. 4 *MAPE of the energy assumption by the four methods. The number of samples, n, varies from 200 to 6000. MAPEs by the four methods without quadratic terms*

modelling error, on the response that measures the annual energy consumption. To accommodate possible non-linear relationship between the response and the input parameters, we also apply all four methods to the extended data that include all quadratic terms of the original uncertain parameters. This expands the scope of the model and allows a non-linear fit. These procedures are denoted as LS2, OMP2, Lasso2 and adapt2, respectively. There are 2212 parameters in the extended data. The out-of-sample performance was evaluated using mean absolute percentage error (MAPE)

$$\frac{1}{2000}\sum_{i\in D_v}\left|\frac{y_i - x_i\hat{\boldsymbol{\beta}}}{y_i}\right|.$$

To test whether or not these algorithms successfully fit the simulated dataset, we studied the performance of the models chosen by the four algorithms (LS considers the full model) given N samples varying from 200 to 6000. As shown in Fig. 4, sparse regression methods, OMP, Lasso and the adaptive Lasso, consistently outperform LS, especially when the number of samples is small. Besides, sparse regression methods with quadratic terms perform significantly better in terms of modelling error (Fig. 5), because they allow to capture the non-linear relationship between the input parameters and the energy consumption. For example, when $N = 6000$, MAPE of adapt is about 1.00%, while MAPE of adapt is 0.74% with quadratic terms considered. When the number of samples is relatively small, the modelling error of Lasso and the adaptive Lasso is often substantial smaller than that of OMP. When the number of samples is large, the Lasso and the adaptive Lasso do not improve. The adaptive Lasso yields comparable results with Lasso. As the increasing of the number of samples, the adaptive Lasso improves slightly.

The number of selected parameters for each method is shown in Figs. 6 and 7. In most settings, the adaptive lasso selects around 130 parameters for the original dataset and 120 parameters when quadratic terms are included. The number of parameters selected

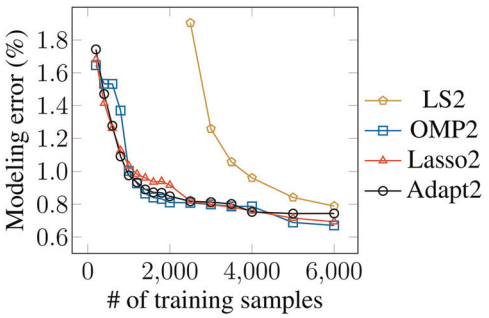

Fig. 5 *MAPE of the energy assumption by the four methods. The number of samples, n, varies from 200 to 6000. MAPEs by the four methods with quadratic terms*

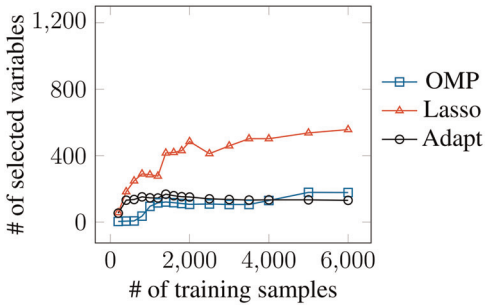

Fig. 6 *Number of selected variables by OMP, Lasso and adapt for number of samples from 200 to 6000*

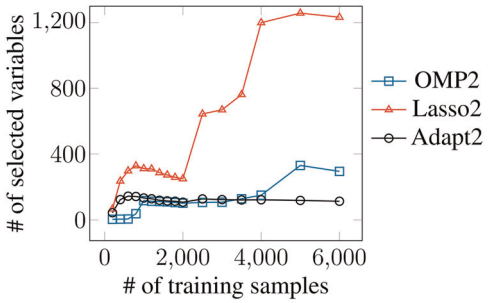

Fig. 7 *Number of selected variables by OMP2, Lasso2, and adapt2 for number of samples from 200 to 6000*

by Lasso increases as the number of samples increases. The number of parameters selected by OMP is very small when the number of samples is small and it stabilises at around 110 when the number of samples is larger than 1000.

Fig. 8 shows the elapsed running time for the four methods with and without quadratic terms. Due to the fact that it adds one input

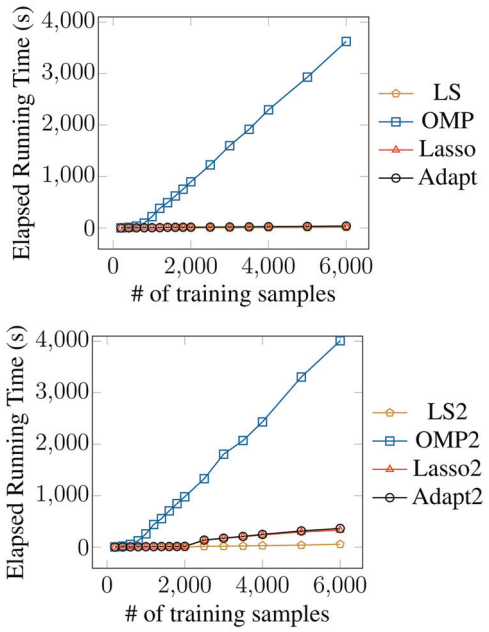

Fig. 8 *Running time (in seconds) of LS, LS2, OMP, OMP2, Lasso, Lasso2, adapt and adapt2. Algorithms were asked to search for a model given p = 1106 or 2212. The number of training samples n varies from 200 to 6000. The zoom-in figure shows a close-up plot for number of samples varying from 200 to 800*

a Running time for *n* varying from 200 to 6000
b Running time for *n* varying from 200 to 800

Table 1 Comparison of modelling error, number of selected parameters and computational cost by the four algorithms without quadratic terms given the number of samples, $N = 1600$

	LS	OMP	Lasso	Adapt
MAPE, %	1.67	1.07	1.10	0.87
number of parameters	1106	117	419	159
simulation time, h	14.6	14.6	14.6	14.6
fitting time, s	2.87	622.0	10.1	13.3
total cost, h	14.6	14.8	14.6	14.6

Table 2 Comparison of modelling error, number of selected parameters and computational cost by the four algorithms with quadratic terms given the number of samples, $N = 2500$

	LS2	OMP2	Lasso2	Adapt2
MAPE, %	1.90	0.81	0.82	0.82
number of parameters	2212	107	645	128
simulation time, h	22.8	22.8	22.8	22.8
fitting time, s	15.4	1330	135.5	137.4
total cost, h	22.8	23.2	22.8	22.8

parameter at a time, the OMP algorithm is very slow when the number of training samples or the number of input parameters becomes large. Compared with Lasso, the adaptive Lasso only added slightly computational burden. This is expected since the first round of adaptive Lasso has already reduced the dimensionality of the dataset significantly, the second round can be implemented very quickly. It is interesting to note that sparse regression methods with quadratic terms do not add much computational cost.

Tables 1 and 2 summarise the modelling error, number of selected parameters and computational cost for simulation and fitting when the number of samples is 1600 or 2500. First, we can see that simulation time dominates the total computational cost for all algorithms. Second, the adaptive Lasso seems advantageous, selecting proper number of parameters and achieving small modelling error under relatively small computational cost. For OMP the computational cost increases very fast with the increasing of either the number of parameters or the number of samples. For Lasso, the number of selected parameters seems unreasonably large.

4.3 Comparisons of the algorithms on ranking

For this task, we adopt the ROC curves to assess the performance of Sadapt and OMP algorithm. Two evaluation metrics are defined as follows: true positive rate (tpr) = TP/(TP + FN) and false positive rate (fpr) = FP/(FP + TN), where TP, TN, FN and FP represent the number of truly related input parameters correctly included, the number of truly unrelated input parameters correctly rejected, the number of truly related input parameters incorrectly rejected, and the number of truly unrelated input parameters incorrectly included.

Although we do not know the truly related input parameters in reality, in theory, the accuracy of identifying truly related input parameters improves with the increase of training samples. Besides, the R^2 of linear regression model is 97% when $n = 8000$. We therefor can evaluate the performance of an algorithm for a small number of training samples against a proxy for the truth in which 8000 training samples are used. Specifically, we denote the intersection of input parameters identified by OMP and adapt as the truth for 8000 training samples.

As the number of training samples increases, both algorithms perform better, as shown in Figs. 9 and 10. However, Sadapt has the best overall performance, i.e. Sadapt is able to achieve substantially higher tpr than the OMP algorithm at the same fpr. For instance, when the number of samples is 200, the tpr of Sadapt is around 0.21 with fpr fixed at 0.1, while the tpr of OMP is 0.18.

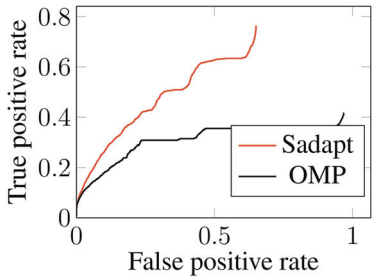

Fig. 9 *ROC-curves for Sadapt and OMP with n = 200*

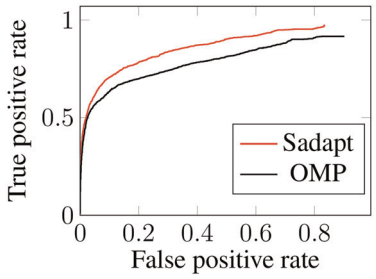

Fig. 10 *ROC-curves for Sadapt and OMP with n = 800*

5 Conclusion

Efficiently finding important parameters from a large number of candidates using a small number of samples can improve the modelling accuracy and greatly alleviate the huge computational task of running the building energy simulation programs. In this work, we considered the adaptive Lasso for uncertainty analysis in building performance simulation. The adaptive Lasso can identify which input parameters are significantly affected the building performance and estimate the influences of these parameters to it simultaneously. Besides, we proposed a stable ranking algorithm that is able to rank important input parameters ahead of unimportant ones. The experimental results show that the advantage of the adaptive Lasso over the recently proposed method [12] that adopted OMP lies in both computational cost and the modelling accuracy. Furthermore, due to its computational efficiency, quadratic terms and even higher polynomial terms are allowed to be included for adaptive Lasso to accommodate possible non-linear relationship between the building performance and the input parameters. Based on these results, we believe that the adaptive Lasso can be fruitfully applied to the uncertainty analysis in building energy analysis.

6 Acknowledgment

The authors thank Prof. Xin Li and Dr. Xiaoming Chen for providing benchmark and helpful comments. Lin's research was supported by the National Natural Science Foundation of China (NSFC), Tianyuan fund for Mathematics (grant no. 11626159) and the Natural Science Foundation of Shenzhen University (grant no. 201542).

7 References

1 'Buildings Energy Data Book', Available at http://buildingsdatabook.eren.doe.gov
2 Pérez-Lombard, L., Ortiz, J., Pout, C.: 'A review on buildings energy consumption information', *Energy and Buildings*, 2008, **40**, (3), pp. 394–398
3 Augenbroe, G.: 'Trends in building simulation', *Building and Environment*, 2002, **37**, (8), pp. 891–902
4 Tian, W.: 'A review of sensitivity analysis methods in building energy analysis', *Renewable and Sustainable Energy Reviews*, 2013, **20**, pp. 411–419
5 Hopfe, C.J., Hensen, J.L.M.: 'Uncertainty analysis in building performance simulation for design support', *Energy and Buildings*, 2011, **43**, pp. 2798–2805

6 Hygh, J.S., DeCarolis, J.F., Hill, D.B., *et al*.: 'Multivariate regression as an energy assessment tool in early building design', *Building and Environment*, 2012, **57**, pp. 165–175
7 Silva, A.S., Ghisi, E.: 'Uncertainty analysis of user behaviour and physical parameters in residential building performance simulation', *Energy Build.*, 2014, **76**, pp. 381–391
8 Sanchez, D.G., Lacarrière, B., Musy, M., *et al*.: 'Application of sensitivity analysis in building energy simulations: combining first- and second-order elementary effects methods', *Energy and Buildings*, 2014, **68**, pp. 741–750
9 Zhao, H., Quach, D., Wang, S., *et al*.: 'Learning based compact thermal modeling for energy-efficient smart building management', IEEE/ACM International Conference Computer-Aided Design (ICCAD), 2–6 Nov 2015, pp. 450–456
10 Liu, W., Wang, H., Zhao, H., *et al*.: 'Thermal modeling for energy-efficient smart building with advanced overfitting mitigation technique', 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), 25–28 Jan 2016, pp. 417–422
11 Rahni, N., Ramdani, N., Candau, Y., *et al*.: 'Application of group screening to dynamic building energy simulation models', *Journal of Statistical Computation and Simulation*, 1997, **57**, (1–4), pp. 285–304
12 Chen, X., Li, X., Tan, S.X.-D.: 'From robust chip to smart building: CAD algorithms and methodologies for uncertainty analysis of building performance', IEEE/ACM International Conference Computer-Aided Design (ICCAD), 2–6 Nov 2015, pp. 457–464
13 Tropp, J.A., Gilbert, A.C.: 'Signal recovery from random measurements via orthogonal matching pursuit', *IEEE Transactions on Information Theory*, 2007, **53**, (12), pp. 4655–4666
14 Zou, H.: 'The adaptive Lasso and its oracle properties', *Journal of the American Statistical Association*, 2006, **101**, pp. 1418–1429
15 McKay, M.D.: 'Latin hypercube sampling as a tool in uncertainty analysis of computer models'. Proc. of Winter Simulation Conf., 1992, pp. 557–564
16 Crawley, D.B., Lawrie, L.K., Winkelmann, F.C., *et al*.: 'EnergyPlus: creating a new-generalization building energy simulation program', 2001, **33**, pp. 319–331
17 Strachan, P.A., Kokogiannakis, G., Macdonald, I.A.: 'History and development of validation with the ESP-r simulation program', *Building and Environment*, 2008, **43**, (4), pp. 601–609
18 Hastie, T., Tibshirani, R., Wainwright, M.: 'Statistical learning with sparsity: the Lasso and generalization' (CRC press, 2015)
19 Tibshirani, R.: 'Regression shrinkage and selection via the Lasso', *Statistical Methodology*, 1996, **58**, pp. 267–288
20 Huang, J., Ma, S., Zhang, C.-H.: 'Adaptive Lasso for sparse high-dimensional regression models', *Statistica Sinica*, 2008, **18**, pp. 1603–1618
21 Friedman, J., Hastie, T., Höfling, H., *et al*.: 'Pathwise coordinate optimization', *The Annals of Applied Statistics*, 2007, **1**, pp. 302–332