A Classification-Based Demand Trend Prediction Model in Cloud Computing

Qifeng Zhou[1], Bin Xia[2], Yexi Jiang[3], Qianmu Li[2(✉)], and Tao Li[3]

[1] Automation Department, Xiamen University, Xiamen, China
zhouqf@xmu.edu.cn
[2] School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
liqianmu@126.com
[3] School of Computer and Information Sciences,
Florida International University, Miami, USA

Abstract. Cloud computing allows dynamic scaling of resources to users as needed. With the increasing demand for cloud service, a challenging problem is how to minimize cloud resource provisioning costs while meeting the user's needs. This issue has been studied via predicting the resource demand in advance. Existing predicting approaches formulate cloud resource provisioning as a regression problem, and aim to achieve the minimal prediction error. However, the resource demand is often time-variant and highly unstable, the regression-based techniques can not achieve a good performance when the demand changes sharply. To cope with this problem, this paper proposes a framework of predicting the sharply changed demand of cloud resource to reduce the VM provisioning cost. In this framework, we first formulate the cloud resource demands prediction as a classification problem and then propose a robust prediction approach by combining Piecewise Linear Representation and Weighted Support Vector Machine techniques. Our proposed method can capture the sharply changed points in the highly unstable resource demand time series and improves the prediction performance while reducing the provisioning costs. Experimental evaluation on the IBM Smart Cloud Enterprise (SCE) trace data demonstrates the effectiveness of our proposed framework.

Keywords: Cloud computing · Capacity planning · Piecewise Linear Representation · Support Vector Machine

1 Introduction

Computing services have become an increasingly popular computing paradigm which provide different styles of services to the cloud resource users with different flavors. Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) are three primary types of cloud computing for both the applications delivered as services over the Internet and the hardware/software systems in the data centers [1].

J. Wang et al. (Eds.): WISE 2015, Part II, LNCS 9419, pp. 442–457, 2015.
DOI: 10.1007/978-3-319-26187-4_41

IaaS cloud is a provision model in which an organization outsources the equipments used to support operations, including storage, hardware, servers, and networking components [1]. In practical application, Iaas is an elastic and economical choice for business IT support. It enables the cloud customers to dynamically request proper amount of virtual machines (VM) based on their business requirements. With the growth of a gigantic number of computing and business server demand, a key issue of IaaS is how to minimize cloud resource provisioning costs while meeting the clients' demands. This is the problem of *effective cloud capacity planning and instant on-demand VM provisioning.*

In general, resource provisioning is challenging due to the pay-as-you-go flexible charging style in IaaS. The amount of resources demand is rarely static, varying as the changes of application number and time. Inefficiency of resource provisioning leads to either over-provisioning or under-provisioning. Over-provisioning may result in idled resources and unnecessary utility costs, while under-provisioning often causes resource shortage and revenue loss. Moreover, initializing a new virtual machine instantly in a cloud is not possible in practice. Therefore, to accomplish effective cloud capacity planning and instant on-demand VM provisioning, application resource needs must be predicted in advance so that the cloud management system can adjust resource allocations in advance.

Capacity planning and instant on-demand VM provisioning problem can be tackled under a unified framework, generally as both problems can be formulated as a generic time series prediction problem [12]. In cloud capacity management, there are two inherent characteristics: **nonlinearity** and **time variability**. The nonlinearity implies that the relationship between the resource demand and its affecting factors is highly nonlinear while the time variability indicates the relationship changes over time. These two characteristics pose a great challenge on effective cloud resource demand prediction.

The existing studies treat the cloud capacity prediction as a regression problem and leverage the state-of-art time series prediction techniques to predict the future capacity of needed resources [8,11]. The Sliding window method [6], Auto Regression (AR) [18], and other methods based on AR such as ARCH(Auto Regressive Conditional Heteroskedasticity) [7], ARMA (Auto Regressive Moving Average) [17] are commonly used techniques to characterize and model observed time series. However, these models are parametric models they only perform well under stable conditions. Artificial Neural Network(ANN) and Support Vector Machine(SVM) regression have also been used to predict the cloud capacity resource demand. These methods decrease the predictive costs compared with the linear regression [9,10].

However, the existing methods can not achieve good performance on resource demand predicting due to the following reasons: (1) The imbalanced demand distribution, dynamic changing requests, and continuous customer turnover make the resource demand highly non-linear and time-varying. Therefore, it is difficult to predict the exact quantity of demand. (2) In practice, the predicting costs are mainly occurred in the cases of sudden changes. However, it is difficult

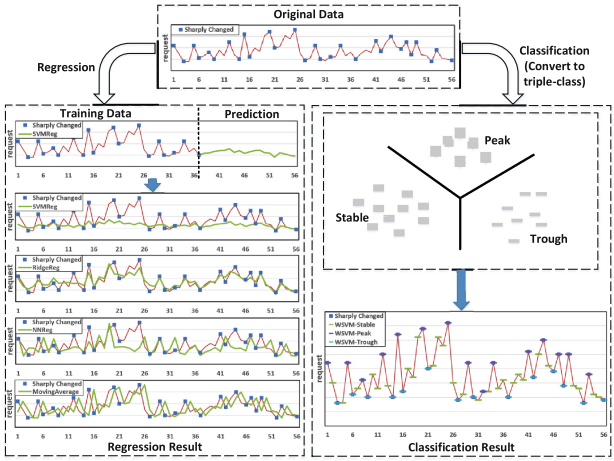


Fig. 1. The illustration of two kinds of cloud resource demand prediction. The left panel includes four commonly used regression-based methods, from top to bottom are Moving average, Nearest neighbour regression, Ridge regression, and SVM regression respectively. The right panel is our proposed PLR-WSVM classification technique. The red line represents real resource demand time series, blue square points are sharply changed demand points, and green line represents the fitting line using different regression methods (Color figure online).

for regression-based methods to capture these changes. (3) Traditional regression cost measures are all symmetric measures [5], but cloud capacity planning is cost sensitive. The estimation for suddenly increasing or decreasing resource demand (noted as peak points and trough points) has different consequences.

In this paper, we propose a framework to address the aforementioned challenges in effective resource provisioning. Our goal is to predict whether the future demand is suddenly changing instead of predicting the actual quantity of demand. First, we formulate the cloud resource demand prediction as a weighted three-class classification problem (peak points, trough points, and stable points). Then, we combine Piecewise Linear Representation (PLR) and Weighted SVM to predict the suddenly changed demand. In addition, we set different weights according to the change rate of the demand, in which the weight reflects the relative importance of each change point.

The main contributions of this paper are describe in Fig. 1. Four commonly used regression-based cloud resource demand prediction methods and their prediction performance on a real world cloud environment are described in the left panel of Fig. 1. Our proposed classification-based method is described in the right panel of Fig. 1. As shown in this figure, commonly used unsupervised regression-based method MA, Nearest Neighbour Regression, Ridge Regression, and SVM Regression cannot capture the suddenly changed points of the resource demand time series. However, our method can identify most of the suddenly changed points and provide a good prediction for all three kinds of points.

The rest of this paper is organized as follows. Section 2 analyses the characteristic of cloud capacity planning problem and briefly introduces PLR and SVM. Section 3 describes the framework of PLR-WSVM. Section 4 presents the experimental results. Finally, Sect. 5 summarizes the paper.

2 Background

To meet the practical demand and reduce the provisioning cost, this paper incorporates PLR and Weighted SVM(WSVM) to predict the change of future cloud resource requirements. PLR is used to extract the peak and trough points, and WSVM is used to model the relationship between the inflection points and the impact factors. We choose these two methods for the following reasons: (1) PLR is simple and the joint points between adjacent segments generated by PLR indicate the change of trends [13–15]. (2) SVM has the excellent generational ability as well as all solutions of SVM model are globally optimal [2,16].

2.1 Cloud Capacity Planning

Highly Unstable. Effective cloud capacity planning aim to prepare the resources properly. However, unstable customer constituents and the freestyle of resource acquisition/releasing make the cloud resource demand highly unstable. Figure 2 shows the change of the overall customer number over time. As is shown, the total number of customers is continually increasing. Therefore, even the request behaviors of old customers keep stable, the overall request still changes over time. Figure 3 illustrates the request history of three frequently requested customers. We can see that three time series share no common property with each other. As a results, the distributions of the resource demands is highly unstable.

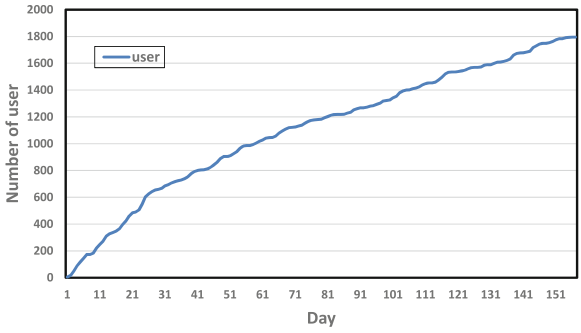


Fig. 2. The change of total cloud service customer over time.

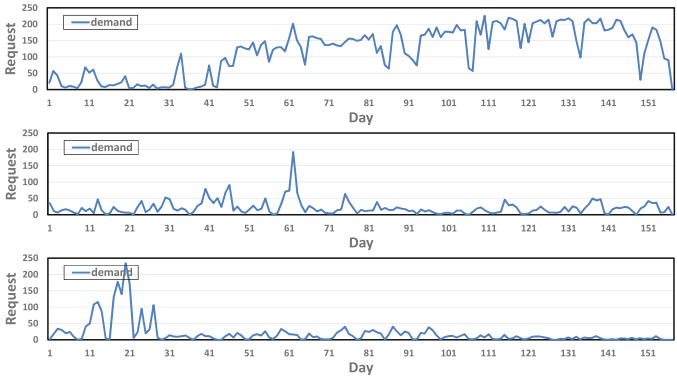


Fig. 3. Time series of resource demands of three frequently requested customers.

Cost Sensitive. Traditional regression-based prediction cost functions such as mean average error (MAE), lease square error (LSE), and mean absolute percentage error (MAPE) are all symmetric measures. In cloud demand prediction, over- and under- prediction will cause different costs, therefore, a symmetric measure is not appropriate for model the asymmetric cost. In this paper, the cloud resource demand prediction is considered as a multi-class classification problem, and we incorporate the different prediction costs as the weights for the samples of different classes. Generally, the weights of peak and trough points should be larger than those of stable points because the predicting costs are increasing when the demand changes suddenly.

2.2 Time Series and PLR

Time series is an ordered set of elements, the element consists of sample values and sample time. Given a time series $T = \{x_1, x_2, ...x_n\}$, the set of segment points is $T_i = (x_{i_1}, x_{i_2}, ..., x_{i_m})$, $(x_{i_1} = x_1, x_{i_m} = x_n, m < n)$, the PLR of T can be described by

$$T_L = \{L_1(x_{i_1}, x_{i_2}), L_2(x_{i_2}, x_{i_3}), ..., L_{m-1}(x_{i_{m-1}}, x_{i_m})\}, \tag{1}$$

where the function $L_{m-1}(x_{i_{m-1}}, x_{i_m})$ represents the linear fitting function at the interval $[x_{i_{m-1}}, x_{i_m}]$. Because the PLR of time series represents a sequence by connecting several linear functions, the value of each point in every interval can be obtained by linear interpolation. Then, the fitting sequence is expressed as $T_i' = (x_1', x_2', ...x_n')$.

Most of the time series segmentation algorithms can be divided into the following three types [13]:

- **Sliding Windows:** A segment is grown until it exceeds some error bound.
- **Top-down:** The time series is recursively partitioned until some stopping criteria are met.

– **Bottom-up:** Starting from the finest possible approximation, segments are merged until some stopping criteria are met. There are two classical ways to find the approximation line [13]:
  - **linear interpolation:** The approximation line for the subsequence $T[a, b]$ is simply the line connecting t_a and t_b.
  - **linear regression:** The approximation line for the subsequence $T[a, b]$ is taken to be the best fitting line in the least squares sense.

2.3 SVM

The main idea of SVM is to generate a classification hyper-plane that separates two classes of data with the maximum margin [2,16,19]. The standard SVM model is as follows:

$$\begin{aligned} &\min_{w,b,\xi_i} \ \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} \xi_i, \\ &s.t. \quad y_i(\langle w, \phi(x_i) \rangle x_i + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \quad i = 1, 2, ...l, \end{aligned} \tag{2}$$

where $x_i \in R^n$ and $y_i \in \{-1, 1\}$ are respectively the training sample and the corresponding class label, ϕ is a nonlinear map from the original space to a high dimensional feature space, w is the normal vector of hyper-plane in the feature space, b is a bias value, ξ_i is the slack variable, $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and C is a penalty coefficient to balance the training accuracy and generalization ability. The dual form of model (2) is:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C, \quad i = 1, 2, ...l, \end{aligned} \tag{3}$$

The model (3) is an linear SVM method, and it can be easily generalized to non-linear decision rules by replacing the inner products $\langle x_i, x_j \rangle$ with a kernel function $k(x_i, x_j)$. When each training sample has a weight, the standard SVM can be extended to weighted SVM (WSVM) [4], the model (3) is transformed to

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C \mu_i, \quad i = 1, 2, ...l, \end{aligned} \tag{4}$$

where $\mu_i (i = 1, ..., l)$ represents the weight of instance x_i. The decision function for WSVM is the same as the standard SVM.

3 The Method

As discussed earlier, a considerable amount of research has been conducted to predict the change of cloud resource demand using regression-based technique. However, due to the characteristics of nonlinearity and time variability, regression-base prediction can only do well in short-term demand prediction. In addition, the predictive errors are usually high when the demand changed suddenly. In this paper, we formulate the cloud demand planning as a classification problem and predict the sharply changed demand.

In this section, we describe the proposed classification-based method named PLR-WSVM. To reduce the time-varying characteristic of resource demand, the whole historic demand dataset is first divided into overlapping training-testing sets. Then, PLR is used to capture the suddenly changed points to form the training dataset, and the weights of the changed points are also assigned according to the changing trend. Finally, WSVM is adopted to build the prediction model.

3.1 The Data Partition

In order to reduce the time-varying feature while maintaining the order of time in time-series data analysis, the whole dataset is often divided into overlapping training-validation-testing sets [3,15]. Suppose the size of whole dataset is m, and the size of each training set and testing set are m_1 and m_2 respectively. Then the whole dataset will be divided into p overlapping training-testing sets:

$$p = \lceil \frac{m - m_1}{m_2} \rceil, \tag{5}$$

where $\lceil x \rceil$ denotes the minimal positive integer that is not less than x.

3.2 Generating the Suddenly Changed Points by PLR

After partitioning the time series dataset into overlapping training-testing sets, PLR is used to automatically generate the suddenly changed demand points. In this work, the top-down algorithm is selected to segment the cloud demand time series and the linear interpolation is adopted to generate the approximation line. The objective segmentation is to produce the best representation such that the maximum error for any segment does not exceed the given threshold δ. The detailed process of PLR is described in Algorithm 1.

Figure 4 presents some examples of using PLR to generate possible suddenly changed points in a period of 120 days. The first subfigure shows the original time series while the rest of the subfigures are generated using different threshold values in PLR. As observed in Fig. 4, the higher the threshold value, the smaller the number of segments generated. For a threshold value of 1.0, there are roughly 65 abrupt changed points while there are only 23 abrupt changed points for a threshold value of 8.0. Each segmentation represents a local peak or trough, and these extremes are transformed into resource demand suddenly changed points.

Algorithm 1. PLR

Input:

δ: the threshold to decide the point is smooth or not;
Reqs: the sequence of requests;

Output:

Label: the type of each point;

1: **for** *index* in *Reqs* (without the first and last point) **do**
2: **if** $Reqs[index] < Reqs[index+1]$ **then**
3: Set $Label[index]$ as pit_{prep}.
4: **else if** $Reqs[index] == Reqs[index+1]$ and $Reqs[index] == 0$ **then**
5: Set $Label[index]$ as $trough_{prep}$.
6: **else**
7: Set $Label[index]$ as $peak_{prep}$.
8: **end if**
9: **end for**
10: Connect the first and last point in *Reqs* with a straight line, and figure out the point P which is farthest from the line. Record the maximum distance as D.
11: **while** $D \geq \delta$ **do**
12: Update the label of P in *Label* as *trough* or *peak* when the label of P is $trough_{prep}$ or $peak_{prep}$.
13: Connect the adjacent unstable points (including the first and last even they are treated as stable points) with straight lines, and figure out the points $P_1, P_2, ..., P_n$ which are farthest from the lines in each segmentation. Record the each maximum distance as $D_1, D_2, ..., D_n$.
14: **end while**
15: Update all $trough_{prep}$ and $peak_{prep}$ points in *Label* as 'smooth'.
16: Return *Label*.

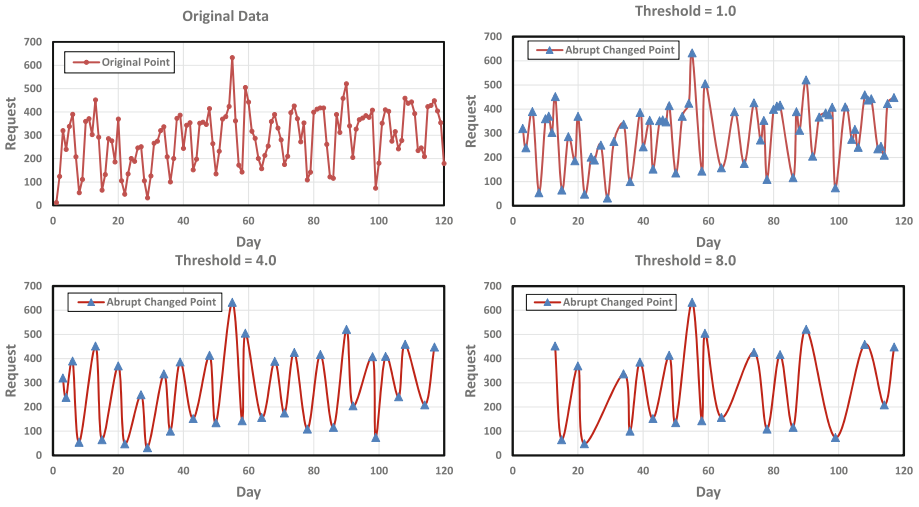


Fig. 4. The possible abrupt changed points generated by PLR

3.3 Constructing Prediction Model by WSVM

We divide all the samples x_i into three classes: peak points(demand suddenly increased points), trough points (demand suddenly decreased points) and stable points(demand changes a little), labeled as 1, 2 and 3, respectively. Furthermore, cloud capacity planning is a cost sensitive problem, the weights of different class instances should be different. According to the cost caused by the error, in model (4), we set

$$\boldsymbol{\mu_i} = \begin{cases} 1+\alpha & if\ y_i = 1 \\ 1+\beta & if\ y_i = 2 \\ 1 & if\ y_i = 3 \end{cases} \tag{6}$$

where y_i is the label of x_i, $\alpha = \lambda \cdot \beta$, $\lambda \geq 1$ is a parameter to adjust the cost between peak and trough points.

Then a three-class weighted classification problem can be constructed for each onerlapping training-testing set:

$$T^{(i)} = T^{(i,tr)} \cup T^{(i,ts)}, i = 1, 2, ...p, \tag{7}$$

where

$$T^{(i,tr)} = \{(x_t^{(i,tr)}, y_t^{(i,tr)}, \mu_t^{(i,tr)}) \mid x_t^{(i,tr)} \in R^n, \tag{8}$$

$$y_t^{(i,tr)} \in \{1, 2, 3\}, \mu_t^{(i,tr)} \geq 1, t = 1, 2, ..., m_1\}, \tag{9}$$

and

$$T^{(i,ts)} = \{(x_t^{(i,ts)}, y_t^{(i,ts)}, \mid x_t^{(i,ts)} \in R^n, y_t^{(i,ts)} \in \{1, 2, 3\}, t = 1, 2, ..., m_2\}, \tag{10}$$

denote the training set and testing set respectively, $x_t^{(i,tr)}$ is training sample, $x_t^{(i,ts)}$ is testing sample, $y_t^{(i,tr)}$ and $y_t^{(i,ts)}$ are corresponding class label. $\mu_t^{(i,tr)}$ is the weight of the training sample computed according to Eq. (6).

WSVM is used to model this three-class classification problem. The overall framework of PLR-WSVM is illustrated in Algorithm 2.

4 Experiment Design and Evaluation

We use the real VM trace log of IBM Smart Colud Enterprise to evaluate the effectiveness of our method. The trace data we obtained records the VM requests for more than 4 months (from March 2011 to July 2011), and it contains tens of thousands of request records with more than 100 different VM types. The original trace data include 21 features such as Customer ID, VM type, Requset Start Time, Requset End Time, and etc [12].

Algorithm 2. PLR-WSVM

Input:

X: Cloud resource demand time series;

$\delta, r_1, r_2, \alpha, \beta$: the modeling parameters;

Output:

The testing accuracy, the decision of next day's request (the type of each point);

1: Normalizing the dataset X by $\tilde{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$;
2: Computing p, the number of partitions according to (5);
3: set i=1;
4: **while** $i \leq p$ **do**
5: Selecting the ith training set and testing set from X;
6: Generating the three-class sample points by Algorithm 1;
7: Setting the weights of each instance in the ith training set according to (6);
8: Training a three-class WSVM model from the ith training set according to (4);
9: Predicting the labels on ith test set.
10: Set i=i+1;
11: **end while**
12: Computing the test accuracy;
13: Return *Label*;

4.1 Data Preprocessing

There are two data preprocessing steps before the raw data recorded by SCE are used in modeling and prediction:

- **Feature Selection:** The raw data contain some request fields that are used during prediction and also contain some noise during temporal pattern mining. Therefore, not all these twenty-one features of a request record are useful. In this work, we only selected two original features, VM Type (which illustrates the type of VM the customer requests), Company (which include the information of the customer send the request) and four statistics features obtained from history data as the feature subset. The details of features involved in our experiments are described in Table 1.
- **Time Series Aggregation Granularity Selection:** The raw trace data are recorded per second. Aggregate these time series by different granularities would have different levels of information and difficulty for prediction. A too fine granularity would make the value on each timestamp lack statistical significance, however, too large granularity would loss some useful information. Figure 5 shows the different cloud capacity provisioning time series aggregated by hour, day and week, respectively. We can see that the coarser the granularity, the larger the provisioning amount in each time slot. Since the lifetime of a VM is usually longer than hours, aggregate the records by hour is not suitable in practice. On the contrary, if we aggregate the time series by week, the cloud required to prepare the most VMs for each time slot. In this case, the small prediction deviation will result in a large cost [12]. In order to produce the enough modeling data while maintaining the statistical significance of raw time series, in this work, we use the daily time series in our system.

Table 1. The description of features

Field	Description
request1Part	The number of request in current time period
requestAvg	The average of request counts in fixed period recently
requestVar	The variance of request in fixed period recently
requestLastWeek	The number of request in the same time period last week
requestSubject	The subjects of request currently (e.g., types of VM)
requestCompany	The companies of request currently

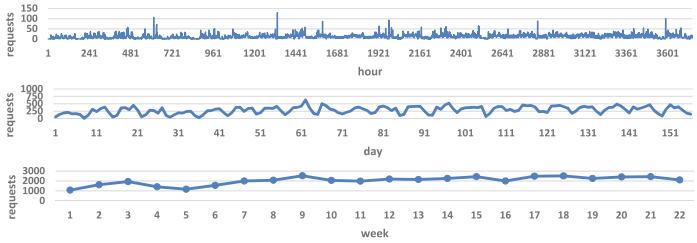


Fig. 5. Time series aggregation granularity selection.

4.2 Experiment Results

We compare the proposed PLR-WSVM method with several commonly used regression-based methods. The detail of methods are described in Table 2.

Regression-Based Methods vs. Real Time Series. Figure 6 displays the original capacity change time series in three different periods and the fitting results using different regression-based methods. From top to bottom, the time series are: (1) Time series predicted by Moving Average; (2) Time series predicted by Nearest Neighbour regression; (3) Time series predicted by Ridge regression; (4) Time series predicted by SVM regression. From left to right, the unstability of three parts of time series is generally increased (i.e., from low to high).

In Fig. 6, we can see that all the regression-based methods can predict well only when the real time series are smooth. With the increasing of unstability

Table 2. The description of methods

Method name	Description
Moving Average	Naive Predictor
Nearest Neighbour Regression	Linear Regression
Ridge Regression	Non-linear Regression
SVM Regression	Non-linear Regression with RBF kernel

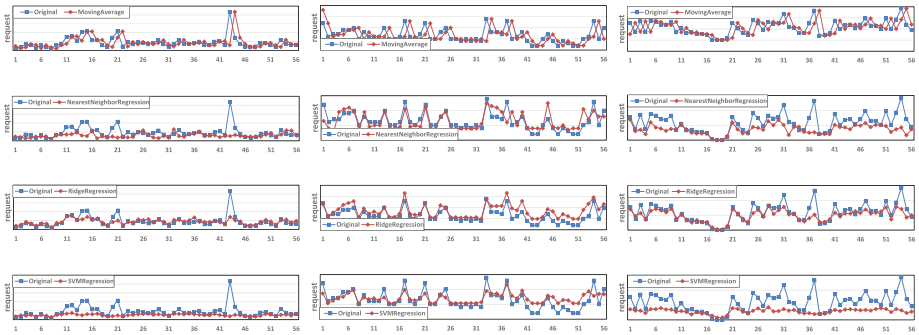


Fig. 6. Regression-based prediction results of three parts of time series.

(sharply changed demand), the fitting error is also increasing. These regression-based methods can give a good prediction in the average sense. For those sharply changed points, regression-based methods cannot predicting well. Among these regression techniques, Moving Average shows a most similar changing tendency with original time series, but its predicting curves have a time delay, limiting its applicability in practice. Ridge Regression has the better performance than Nearest Neighbour regression and SVM regression techniques, but it also cannot predict the sharply changed points well in unstable time series.

PLR-WSVM vs. Regression-Based Methods. The goal of this paper is to study a model predicting the suddenly changed points of cloud resource demand. PLR-WSVM as a classification-based technique can give the results directly. However, regression-based methods must do the following two steps: fitting the original data and setting a threshold to decide whether the next demand is a suddenly changed point or not. It is difficult to set the decision threshold because the resource demand varying from one moment to another.

Table 3 shows the performance of PLR-WSVM compared with other regression-based methods. The experimental setup is using one week time series to predict the demand of next day. To ensure the comparability, the predicting results of regression-based methods are transformed into three classes by comparing the relative change of resource demand with a proper threshold. The transform rule is defined as:

$$\boldsymbol{label}(\boldsymbol{x_t}) = \begin{cases} 1 & if\ c(x_t) \geq\ \theta \\ 2 & if\ c(x_t) \leq -\theta \\ 3 & if\ |c(x_t)| < \theta \end{cases} \tag{11}$$

where $c(x_t)$ represents change rate of resource demand between the current regression value and the previous one, defined as

$$\boldsymbol{c}(\boldsymbol{x_t}) = \frac{R(x_t) - R(x_{t-1})}{R(x_{t-1})}, \tag{12}$$

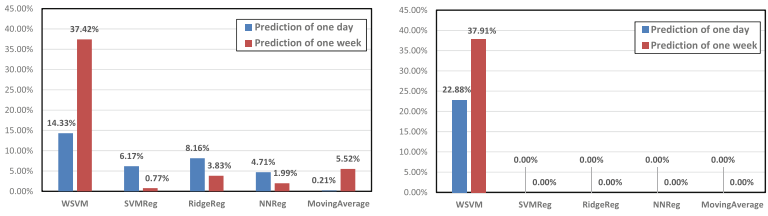
where $R(x_t)$ and $R(x_{t-1})$ indicate the current and previous regression value. θ is the threshold to transform the relative change rate into a label.

Table 3. The comparison of predicted changed point between PLR-WSVM and other regression-based methods.

Method	Threshold	Recall accuracy of Peak(%)	Recall accuracy of Trough(%)
SVMReg	0.5	7.2	4.5
	1.0	0.8	0.0
RidgeReg	0.5	20.7	5.6
	1.0	3.8	0.0
NNReg	0.5	11.2	7.1
	1.0	2.0	0.0
MovingAverage	0.5	0.0	0.0
	1.0	0.2	0.0
PLR-WSVM	-	37.42	37.91

We compared the range of threshold θ from 0.1 to 1.5, Table 3 shows the experiment results. From this table, we can see that PLR-WSVM has the best performance in predicting the troughs and peaks of cloud resource demand while maintain a comparable overall accuracy. Regression-based techniques are very sensitive to the thresholds and have poor performance in predicting suddenly changed points.

We also compare the long term prediction performance of different methods (e.g., using one month time series to predict resource demand of next week). Figure 7 illustrates the prediction results. Compared with short term predicting results, we can see that the predict performance of PLR-WSVM is increased when the period has been extended from one day to one week. The performance of other regression-based methods decreased greatly, especially on trough points. The experimental results indicate that PLR-WSVM has more robust long term prediction ability.



(a) Prediction results for peaks (b) Prediction results for troughs

Fig. 7. The performance compare on different time span. The left figure is the prediction results of peaks and the right figure is the prediction results of troughs.

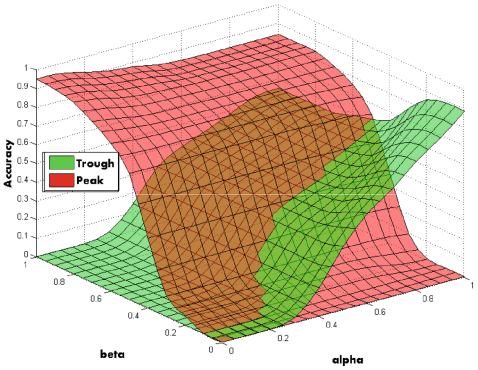


Fig. 8. The effect of turning α and β for capacity prediction.

In addition, PLR-WSVM can balance the cost among different classes of points in cloud resource demand. According to Eq. (6), we can set different weights to trough and peak points using $\alpha = \lambda\beta$. In cloud predicting, since the cost of under-prediction is large than over-prediction, we set $\lambda \geq 1$.

Figure 8 shows the trough and peak points prediction with varying α and β. We can see that the prediction accuracies of trough and peak points have different variation trends. In practice, cloud providers can tradeoff these two costs according the real demand to minimize the total cost.

5 Conclusion

The prediction of cloud resource demands is a very challenging task due to the time-variant and highly unstable characteristics. Traditional regression-based techniques cannot achieve good prediction performance, especially when resource demand changed sharply. In this paper, we discuss the cloud capacity planning problem from a new perspective: predicting the sharply increased and decreased resource demand. Thus the service vendors can cope with the abrupt changed cloud resource demand in advance and improve the quality of cloud service.

We transform the cloud capacity planning problem into a classification problem and use PLR-WSVM to predict the trough and peak points. In particular, PLR is used to generate the training samples from the original resource demand time series, and then WSVM is used to model the prediction of sharply changed demand. Unlike regression-based techniques, our method formulates the cloud resource demand into a three-class classification problem and it does not need to determine the threshold of trough and peak points. Furthermore, WSVM can assign the different weights for peak and trough points to minimize the total provisioning costs.

Experimental results on the trace data of IBM Smart Cloud Enterprise demonstrate the effectiveness of our proposed method. Compared with

regression-based techniques, our proposed method achieves more accurate and robust prediction performance on suddenly changed cloud resource demand.

Acknowledgement. This work is supported by Natural Science Foundation of China under Grant No. 61503313 and the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology), Grant No. 30920140122007.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
2. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
3. Cao, L.J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans. Neural Netw. **14**(6), 1506–1518 (2003)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 27 (2011)
5. Chatfield, C.: The Analysis of Time Series: An Introduction. CRC Press, Boca Raton (2013)
6. Dietterich, T.G.: Machine learning for sequential data: a review. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 15–30. Springer, Heidelberg (2002)
7. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica J. Econometric Soc. **50**(4), 987–1007 (1982)
8. Hamilton, J.D.: Time Series Analysis, vol. 2. Princeton University Press, Princeton (1994)
9. Iqbal, W., Dailey, M.N., Carrera, D.: Black-box approach to capacity identification for multi-tier applications hosted on virtualized platforms. In: 2011 International Conference on Cloud and Service Computing (CSC), pp. 111–117. IEEE (2011)
10. Islam, S., Keung, J., Lee, K., Liu, A.: Empirical prediction models for adaptive resource provisioning in the cloud. Future Gener. Comput. Syst. **28**(1), 155–162 (2012)
11. Jiang, Y., Perng, C.S., Li, T., Chang, R.: ASAP: a self-adaptive prediction system for instant cloud resource demand provisioning. In: 2011 IEEE 11th International Conference on Data Mining (ICDM), pp. 1104–1109. IEEE (2011)
12. Jiang, Y., Perng, C.S., Li, T., Chang, R.N.: Cloud analytics for capacity planning and instant vm provisioning. IEEE Trans. Netw. Serv. Manage. **10**(3), 312–325 (2013)
13. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings IEEE International Conference on Data Mining, ICDM 2001, pp. 289–296. IEEE (2001)
14. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: a survey and novel approach. Data Min. Time Ser. Databases **57**, 1–22 (2004)
15. Luo, L., Chen, X.: Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. Appl. Soft. Comput. **13**(2), 806–816 (2013)

16. Vapnik, V.N., Vapnik, V.: Statistical Learning Theory, vol. 1. Wiley, New York (1998)
17. Whitle, P.: Hypothesis Testing in Time Series Analysis. Almqvist & Wiksells, Uppsala (1951)
18. Yule, G.U.: On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers. Philos. Trans. Roy. Soc. Lond. Ser. A **226**, 267–298 (1927). Containing Papers of a Mathematical or Physical Character
19. Zhou, Q., Hong, W., Shao, G., Cai, W.: A new SVM-RFE approach towards ranking problem. In: IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009, vol. 4, pp. 270–273. IEEE (2009)