A SVM-based Multi-dimension Factor Decision-making Model Framework

Peng NIE*
Jiangxi University of Finance and Economics, Nanchang, Jiangxi, 330013, China
*Corresponding author

Keywords: Cloud-desktop, Software estimation, SVM, Classification.

Abstract. The cloud-desktop based on the virtual desktop infrastructure (VDI) is deployed more often as the advanced mobile office solution. However, it is an assignable challenge to choose a fit VDI product at low cost for too many testing features to be investigated. In this paper, we proposed a SVM-based multi-dimension factor decision-making model framework (SMFDMF) for the VDI-based cloud-desktop application evaluation. SMFDMF is highly data adaptive, applies and is able to account for correlation as well as interactions among features. This makes SMFDMF particularly appealing for high-dimensional cloud-desktop testing feature analysis. The experiment results show that our SMFDMF is workable, easy to implement and result in good estimation accuracies.

Introduction

Nowadays, the cloud-desktop based on virtual desktop infrastructure (VDI) helps the client users simplify provisioning and administration, extending the endpoint hardware investments, and making it easier to support employee-chosen endpoint devices. However, it is still an assignable challenge to make a decision which specific VDI software implementation satisfies the given application environment. Lots of evaluation models have been proposed to evaluate the software system on a certain perspective. Jelinski[1] suggested the Markov chain to calculate the software reliability. Goel[2] used a time-dependent error-detection rate model for software performance measures. Sheldon employed the stochastic petri net models to analyse software safety and reliability. Karunanithdi[4], Adnan[5], Kiran[6] et al. applied Neural Networks in reliability prediction. Afzal, Torkar[7] et al. used the genetic programming for software reliability growth modeling. The works above focused on limited dimensions, mostly only one dimension (such as reliability, safety, availability, survivability, etc) to evaluate a software system. It is quite different from the one dimension evaluation system, the VDI-based cloud-desktop application evaluation is affected by multiple dimensions or a multi-dimension factor matrix, including user experience (UE), tech features (TF), management & control (MC).

In this paper, we proposed a SVM-based multi-dimension factor decision-making model framework (SMFDMF) for the VDI-based cloud-desktop application evaluation.

The rest of this paper is organized as follows. Section II presents the proposed SMFDMF model in detail. Section III gives the experiments and analysis. Finally, Section IV concludes this paper.

SVM-based Multi-dimension Factor Decision-making Model Framework

The main idea of SMFDMF is shown as follows. Construct the cloud-desktop application feature testing results as the SMFDMF input sets. The feature testing results are divided into different dimensions based on their likeness. Then SMFDMF learns the knowledge between the feature values and the corresponding estimation dataset. A final evaluated result is output by SMFDMF based on the learned knowledge.
Notations

- $D$: a multi-dimension feature space
- $D_i$: the i-th sub-class vector of $D$
- $D_{i,j}$: the j-th property of $D_i$
- $C_{i,j}$: the classification of the j-th property in the i-th dimension vector of $D$
- $V$: a multi-dimension decision result vector
- $v_i$: the decision result of $D_i$
- $R$: the final result of SMFDMF

Framework

In our proposal, we employ SVM [8,9] based on the cloud-desktop historical $D$ feature values to construct the system evaluation. Fig. 1 presents the overview of SVMREF. The elements in the SVMREF are introduced in detail as follows.

- The historical $D$ feature values and the correlative estimations are collected and divide into several sub-class $D_i$ which can be identified from each other.
- The SVM classification algorithm is applied to $D_i$, and gives the classification of the j-th property in the i-th dimension vector of $D$.
- SVM classification training uses the reliability input vectors as its input, and trains itself to learn the implicative complex knowledge between $D_{i,j}$ and $D_i$.
- SMFDMF uses the desired multi-dimension feature values and outputs an abecedarian value. The SVM error between the real value and the evaluation is checked, and SVM is trained again until the SVM error is minimized to a given value.
- SVMREF loads the current multi-dimension feature values to evaluate the cloud-desktop system and outputs the evaluation result after the SVM is well-trained.

![Diagram of SMFDMF](image)

Figure 1. SMFDMF overview.

Assume $\{(R_{j,1}^{i}, \cdots, R_{j,n}^{i}), \cdots, (R_{j,1}^{k}, \cdots, R_{j,n}^{k})\}$ be the given training datasets where each $(R_{j,1}^{i}, \cdots, R_{j,n}^{i})$ shows the SVM training data vector in the i-th iteration for $D_j$ properties from 1 to n.
$R_i^j$ is the corresponding target system estimation value in the $i$-th cloud-desktop testing where $i = 1, \cdots, k$. The support vector regression solves an optimization problem as Eq. 1:

\[
\text{minimize: } \frac{1}{2} \| w \|^2 + C \sum_{i=0}^{l} (\xi_i + \xi_i^*) \\
\text{subjected to } \begin{cases} 
R_i^j - \langle w, (R_i^j, \cdots, R_i^{jm}) \rangle - b \leq \varepsilon_i + \xi_i \\
\langle w, (R_i^j, \cdots, R_i^{jm}) \rangle + b - R_i^j \leq \varepsilon_i + \xi_i^* \\
\xi_i, \xi_i^* \geq 0 \quad i = 1, \cdots, l
\end{cases}
\]

where $\xi_i$ is the upper training error ($\xi_i^*$ is the lower) subject to the $\varepsilon_i$ insensitive tube $R_i^j - \langle w, (R_i^j, \cdots, R_i^{jm}) \rangle - b \leq \varepsilon_i$. The parameters which control the regression quality are the cost of error $C$, the width of the tube and the mapping function $\phi$. The constraint simply that we would like to put most data $(R_i^j, \cdots, R_i^{jm})$ in the tube $R_i^j - \langle w, (R_i^j, \cdots, R_i^{jm}) \rangle - b \leq \varepsilon_i$. If $(R_i^j, \cdots, R_i^{jm})$ is not in the tube, there is an error $\xi_i$ or $\xi_i^*$ which we would like to minimize in the objective function. SVM avoids underfitting and overfitting the training data by minimizing the training error $C \sum_{i=0}^{l} (\xi_i + \xi_i^*)$ as well as the regularization term $\frac{1}{2} \| w \|^2$. For traditional least square regression, $\varepsilon_i$ is always zero and data are not mapped into higher dimensional spaces. Hence, SVM is a more general and flexible treatment on regression problems.

The SVM model used herein has three mutually dependent parameters, namely $C, \varepsilon, \gamma$, thus changing the value of one parameter changes the other parameters too. A common way to estimate the SVM parameters $C, \varepsilon, \gamma$ is to separate the data into two sets, namely a training dataset and a validation dataset. The prediction accuracy of this validation dataset reflects the accuracy of the model, and SVM parameters which are able to give minimum prediction error are considered to be the optimal parameters.

Experiments and Analysis

Cloud-desktop System Features

In our testing case, the cloud-desktop system features are divided into 3 dimensions as the tech features, the management & control features, and the user experience features. Each feature dimension has its own $n$-dimension feature space as Fig. 2 shown.

Testing Environment

**Hardware Environment**

Server×1: CPU: Intel Xeon E5-2670 v3, memory: 256G, disk: 2.7T@7200 r/s, no GPU.
PC Client×50: CPU: Intel Core i3-4150, memory: 4G, disk: 1T SATA.
Cloud Desktop VM×50: 2 cores@2 cpus, memory: 4G, disk: 32G

**System Environment**

Server: Xen Server 6.5.
PC Client×50: Windows 7×64.
Cloud Desktop VM: Windows 7×64.
SMFDMF predicting error

Our cloud desktop system results in 3600 testing reports for the tech & user experience feature testing and estimation reports for the management & control requirement. The top 3000 testing reports and the top 3000 estimation reports are used as the training dataset, and the rest datasets are used to provide the predictive accuracy.

![Feature space diagram]

Figure 2. A multi-dimension feature space of the cloud-desktop system.

SMFDMF predicting results are shown as Fig. 3, Fig. 4, Fig. 5 and Fig. 6 for the tech dimension, the user experience dimension, the management & control dimension, and the entire cloud-desktop predictive respectively.

![Tech dimension predictive accuracy]

Figure 3. The tech dimension predictive accuracy.

![User experience dimension predictive accuracy]

Figure 4. The user experience dimension predictive accuracy.
The predictive accuracy is given by Eq. 2.

\[
err = \frac{\sum_{i=1}^{n} F(R_i, Real_i)}{n}
\]

where Real is the observed values, \( F(\cdot) \) is a threshold function which returns 1 when \( R_i = Real_i \), or returns 0 in other cases. The accuracies of the 3 dimension estimations and the entire system estimation are presented in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Tech</th>
<th>User experience</th>
<th>Management &amp; control</th>
<th>Cloud desktop system</th>
</tr>
</thead>
<tbody>
<tr>
<td>err</td>
<td>0.031</td>
<td>0.040</td>
<td>0.003</td>
<td>0.030</td>
</tr>
</tbody>
</table>

**Conclusion**

SVM has proven to be an effective tool for such settings, already having produced numerous successful applications. The complexity and high-dimensionality of cloud-desktop system estimation data require flexible and powerful statistical learning tools for an effective estimation process. The experiment results of the cloud-desktop system estimation based on the SVM show that SMFDMF is workable, easy to implement and result in good estimation accuracies. Additionally, we demonstrated the usability of our framework on the estimation task of high-dimension feature space for the software system.

**References**


