2003－MPS－45 (5)
2003／6／24

Reduction and Classification of Input Parameters for Large-scale Simulations

Hitomi Matsuyama[1], Yuki Matsuoka[1], Mika Koganeyama[1],
Chiemi Watanabe[1], Yutaka Ueshima[2], Kazuki Joe[1]

Abstract An integrated management system for simulation cycles has been developed and performed. However, the existing management system often requires complex operation for data management. We investigate an autonomous agent system that advises the simulation users about management operation for large-scale simulations. Specifically, we develop an input parameter customizing agent which is a part of the agent system. As the first prototype of the agent, we analyze input parameter sets for large-scale simulations at JAERI Kansai Research Establishment Advanced Photon Research Center. In this paper, we propose an input parameter analysis method; input parameter sets are reduced to smaller dimensions using Principal Component Analysis (PCA), and classified using Learning Vector Quantization (LVQ) according to simulation types. The validation of the input parameter analysis method is presented by experiments.

1 Introduction

Large-scale simulations are performed in various research fields thanks to the inexpensive simulation costs. An integrated management system for simulation cycles. However, special system procedures are required for the simulation users because they require knowledge about the simulations in detail. An autonomous agent that advises the simulation users on management operation for large-scale simulations. We designed and implemented a prototype agent system for large-scale simulations in JAERI KRE APRC (Japan Atomic Energy Research Institute Kansai Research Establishment Advanced Photon Research Center) as a first step towards addressing this need.

The "progressive parallel plasma" program has been developed and executed on massively parallel computers at JAERI KRE APRC. They have also developed the "P-cube support system" to integrate and manage the simulation cycles. The system is not quite useful because the results of the simulation experiments can not be efficiently stored on data servers, and the choice of input parameters for the system is extremely complicated. Thus we investigated an agent system which learns complex management operations and collects and applies information for large-scale simulation cycles autonomously and automatically [1]. The system includes at least two kinds of agents; an autonomous control agent and an automatic learning agent. The autonomous agent saves data to data servers efficiently. The automatic learning agent chooses custom input parameters. We have started the implementation of an autonomous control agent called the "parallel I/O control agent" [2]. In this paper, we propose an automatic learning method using LVQ (Learning Vector Quantization) and evaluate its capability.

The input parameters for the simulation construct a vector space. The simulation target depends on the values of the vector space. In other words, spatial location in the vector space represents the purpose of the simulation. Therefore, LVQ can be used to classify the vector space. However, it takes a long time to compute them and huge memory to classify the original dimensions. We propose a two-step method; we apply PCA to transform them into smaller dimensions, and then apply LVQ to the reduced dimensions. Using this method, they can be classified by simulation type with reduced cost.

The rest of paper is organized as follows. In Section 2, we explain input parameters for the P-cube support system. In Section 3, parameter reduction using PCA is presented and evaluated. In Section 4, parameter classification by LVQ is proposed and evaluated. Section 5 concludes this paper.

2 Input Parameters for the P-cube Support System

2.1 Overview of input parameters

Simulation users at JAERI KRE APRC have to set input parameters in each of the six categories de-

[1]Nara Women's Univercity
[2]JAERI, Kansai Research Establishment, Advanced Photon Research Center

scribed below. The input parameter set consists of about two hundred fifty different values.

- Parameters for temporary directories and execution conditions
- Parameters for rendering regions of calculation results
- Parameters for visualization
- Parameters for electric charge, number density, and temperature of ions
- Parameters for electric charge, number density, and temperature of electrons
- Parameters for the features of the laser

Finally, we extracted twelve main parameters out of two hundred fifty based on the above six categories. The twelve main parameters depend on simulation types. We expect that LVQ can classify input parameters by simulation type.

2.2 Problems in Input Parameter Settings

As explained in Section 2.1, there are about two hundred fifty kinds of input parameters for the simulations at JAERI KRE APRC. Various simulations are performed with the progressive parallel plasma program. The number of combinations of input parameter sets becomes large because input parameter values are different for each simulation type. It is impossible for users to set input parameters if they do not know the simulation in detail.

The simulation users have to set input parameters for the large-scale simulation with the progressive parallel plasma program using their empirical sense because no rules for the selection have been found. So, most users can not determine appropriate parameter sets for their simulation purpose. As a result, execution error often occurs.

We propose an input parameter customization agent with automatic learning which provides and complements input parameter examples and given parameter values sets, respectively. The input parameter customization agent has to know the simulations in detail. Rule-based learning is difficult because the dependences among parameters are too complicated. The agent should learn from many given patterns of input parameters sets and obtain the resultant knowledge automatically.

We adopt LVQ for the automatic learning capability. By classifying input parameters sets by simulation type with LVQ, it becomes possible to associate unknown input parameter sets with proper simulation types which users want to perform.

2.3 An Agent for Choosing Input Parameters

The computation cost of LVQ execution becomes expensive for learning many input parameter patterns although we have reduced the two hundred fifty input parameters to twelve. We need another technique for the reduction of twelve input parameters to reduce the LVQ execution cost.

Table 1 shows the names and the descriptions of the twelve main parameters. The above twelve parameters may have some dependences between them because we did not extract them from the specification of the progressive parallel plasma program but from the simulation types. To remove the dependences, we apply PCA to the twelve parameters, and obtained the minimum parameters.

3 Applying PCA to Input Parameters

3.1 PCA

PCA [3] is a statistical technique which performs a linear transform of data into an orthogonal non-correlated base to maximize the variance of the original data. PCA transforms data into fewer dimensions for examining relationships between several values while keeping the original information from the reduction.

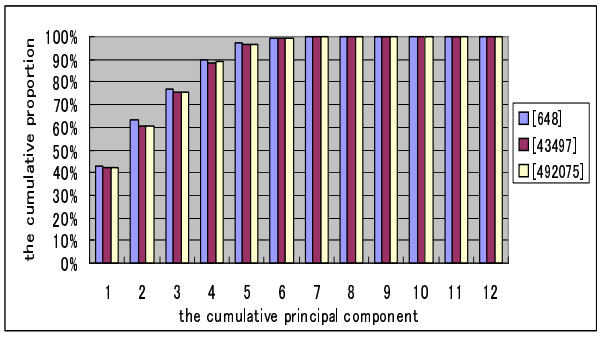

Figure 1: the cumulative proportion

3.2 Applying PCA to Real Input Parameters

We applied PCA to 648, 42,497 and 492,075 input parameter sets to examine the stability of the cumulative proportion. We modified the above parameter values so that the average of the input parameters is set to 0 and the variance is 1 to standardize the input parameters.

Fig.1 shows the cumulative proportion for each element of the input parameter sets. The vertical axis shows the cumulative proportion described in Fig.1. As a result, we find that the cumulative proportions of each parameter set are near to each other regardless of the number of input parameter sets. Thus the results obtained by PCA are stable and reliable.

Fig.1 shows that the first through fifth principal components represent 95% of the original input parameters. Therefore input parameter sets on twelve

name	explanation	name	explanation
dlt_t	normalized time step of numerical simulation	nex1	distribution function of electron momentum
c	light speed normalized by typical electron speed	nex2	distribution function of electron momentum
nix1	distribution function of ion momentum	Te0	temperature of electrons
nix2	distribution function of ion momentum	avrg_dns_e	numerical factor of number density of electrons
Ti0	temperature of ions	E0	intensity of laser
avrg_dns_i	numerical factor of number density of ions	rLwx	laser wavelength in X-axis direction

Table 1: parameter

dimensions can be transformed into five dimensions keeping the original information from the reduction.

4 Parameters Classification by LVQ

4.1 LVQ

Learning Vector Quantization (LVQ) is an arbitrary statistical algorithm with supervised learning [4]. We assume that a number of "codebook vectors" are placed in an input vector space to approximate various domains of input vectors by their quantized values. Values for the codebook vectors that approximately minimize the misclassification errors in the nearest-neighbor classification can be found as asymptotic values in the learning process. This learning process is repeated a specified number of times for training. Repeating the above learning process, the Bayes discriminate boundary can be decided. In this paper, we adopt OLVQ1 for the classification of input parameters.

4.2 Classification Experiments

¿From the experiments in section 3.2, we apply LVQ to input parameters reduced by PCA in this experiment. The data sets we use here provide three kinds of simulations. We call the three simulations A, B and C, respectively.

We compared the recognition accuracy of the following three experiments which are performed varying the size of the training data, codebook vector and the training steps. We chose about 40,000 sets from all the data sets, where some data sets are used for the training data and the others are used for the recognition test.

Experiment 1 LVQ is applied to 2,000, 3,000 and 6,000 data sets, respectively. Each data set consists of the same number of input parameter sets for the three simulation types. Namely, 6,000 data sets are divided into 2,000 data sets by the simulation type. The number of codebook vectors is set to $\frac{1}{10}$ of each data set and the number of training steps is set to forty times the number of codebook vectors.

Experiment 2 LVQ is applied to the same 6,000 data sets as experiment 1, but the number of codebook vectors is 100, 200, 600, 800 and 1,200, respectively. The number of training steps is set to forty times the number of codebook vectors.

Experiment 3 LVQ is applied to the same 6,000 data sets as experiment 1, but the number of training steps is 24,000, 48,000 and 80,000, respectively. The number of codebook vectors is set to 1,200.

4.3 Results of Classification

We analyzed the recognition accuracy of each experiment explained in section 4.2 using the test data. The results of experiment 1 is the low recognition accuracy of simulation C (2,000 data sets) and B (3,000 data sets). On the other hand, recognition accuracy of all the simulations is more than 95% for the 6,000 data sets.

Fig.2 shows the results of experiment 2. The vertical axis shows the recognition accuracy, and the horizontal axis shows the number of codebook vectors.

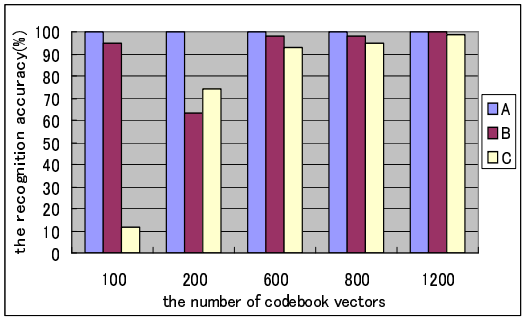

Figure 2: the recognition accuracy of experiment 2

In Fig.2, the recognition accuracy of simulation C is about 10% when the number of codebook vectors is set to 100. Fig.2 shows rapid increase in the recognition accuracy of simulation C of more than 70% when the number of codebook vectors is set to 200.

The more codebook vectors there are, the higher the recognition accuracy is. The recognition accuracy of simulation C is assumed to be highly dependent on the number of the codebook vectors. The recognition accuracy of all the simulations is more than 98% when the number of the codebook vectors is set to 1,200.

Fig.3 shows the results of experiment 3. The vertical axis shows the recognition accuracy, and the horizontal axis shows the number of training steps.

Fig.3 shows the low recognition accuracy of simulation C when the number of training steps is set to 24,000. The recognition accuracy is almost the same

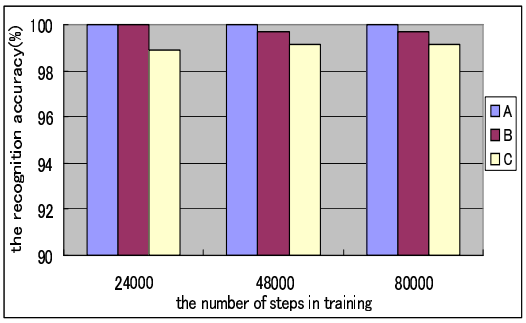

Figure 3: the recognition accuracy of experiment 3

when the number of training steps is set to 48,000 and 80,000.

The training converges when the number of codebook vectors and training steps are set to 1,200 and 48,000, respectively. As a result, it turned out that the recognition accuracy is the highest when LVQ is applied to 6,000 data sets which contain 1,200 sets of code vectors with 48,000 steps in training.

The recognition accuracy of simulation B and C are lower than simulation A. The result of the classification is visualized using a Sammon Mapping [5] to examine the distribution of input parameter categories for simulation A, B and C. The Sammon Mapping is a method which generates a mapping from an n-dimensional data space to the two-dimensional plane. Fig.4 shows the main part of the visualization result of LVQ classification applied to 6,000 data sets which contains 600 sets of codebook vectors with 48,000 training steps. The reason we use 600 codebook vectors is to simplify the presentation of Fig.4.

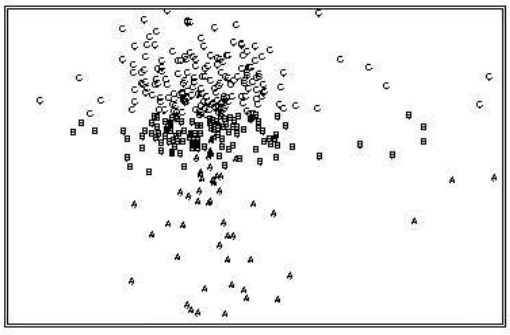

Figure 4: the codebook vectors

In Fig.4, the parameter category of simulation A is distant from the other simulations. On the other hand, the parameter category of simulation B is close to simulation C.

Therefore it is assumed that the reason the recognition accuracy of simulation B and C are lower than simulation A in all the experiments is that some parameter sets can not be classified near the border between parameter categories of simulation B and C.

5 Conclusion

The purpose of this research is to implement an input parameter agent with an automatic learning capability for an autonomous management system for large-scale simulations at JAERI KRE APRC. In this paper, we proposed a two-step method; we applied PCA to transform them into smaller dimensions, and then applied LVQ to the reduced ones.

We applied PCA to the twelve main parameters out of about two hundred fifty. As a result, five principal components obtained using PCA represent more than 95% of the original data. Therefore input parameter sets on twelve dimensions can be transformed into five dimensions.

Input data obtained using PCA was classified into three simulation types using LVQ. The recognition accuracy of all the simulations was more than 98% when LVQ was applied to 6,000 data sets which contained 1,200 sets of codebook vectors with 48,000 training epochs. It turned out that the input parameter sets could be classified into three kinds of simulation types correctly by LVQ.

We will apply this method to input parameters of different simulation types from the three simulations types to examine whether they can be classified correctly. Moreover, adding an automatic generation capability using knowledge obtained using LVQ in the input parameter customization agent, we will be able to complete the implementation of the agent.

References

[1] H.Matsuyama, Y.Matsuoka, M.Koganeyama, Y.Ueshima, K.Joe: "Design and Implementation of an Automatic Learning Agent System for Large-scale Simulation Cycle", IPSJ SIGMPS, 2002-MPS-42, Vol.2002, No.114, pp.17–20 (2002).

[2] Y.Matsuoka, H.Matsuyama, M.Koganeyama, Y.Ueshima and K.Joe: "Parallel I/O Control Agent for Large-scale Simulation Database", IPSJ SIGMPS, 2003-MPS-43, Vol.2003, No.20, pp.11–4 (2003).

[3] M.E.Tipping and C.M.Bishop: "Probabilistic principal component analysis", Journal of Royal Statistical Society, vol.61, No.3, pp.611–622 (1999)

[4] T.Kohonen, J.Kangas, J.Laaksonen and K.Torkkola: "A program package for the correct application of Learning Vector Quantization algorithms", Proceedings of the International Joint Conference on Neural Networks, Vol.I,pp.725–730 (1995).

[5] John W. Sammon Jr.: "A nonlinear mapping for data structure analysis", IEEE Transaction on Computers, Vol.C-18, No.5, pp.401–409 (1969)