Journal of Geoscience and Environment Protection, 2017, 5, 39-48
http://www.scirp.org/journal/gep
ISSN Online: 2327-4344
ISSN Print: 2327-4336

Polarimetric Meteorological Satellite Data Processing Software Classification Based on Principal Component Analysis and Improved K-Means Algorithm

Manyun Lin, Xiangang Zhao, Cunqun Fan*, Lizi Xie, Lan Wei, Peng Guo

National Satellite Meteorological Centre, Beijing, China
Email: *fancq@cma.gov.cn

How to cite this paper: Lin, M.Y., Zhao, X.G., Fan, C.Q., Xie, L.Z., Wei, L. and Guo, P. (2017) Polarimetric Meteorological Satellite Data Processing Software Classification Based on Principal Component Analysis and Improved K-Means Algorithm. *Journal of Geoscience and Environment Protection*, **5**, 39-48.
https://doi.org/10.4236/gep.2017.57005

Received: March 20, 2017
Accepted: July 10, 2017
Published: July 13, 2017

Abstract

With the increasing variety of application software of meteorological satellite ground system, how to provide reasonable hardware resources and improve the efficiency of software is paid more and more attention. In this paper, a set of software classification method based on software operating characteristics is proposed. The method uses software run-time resource consumption to describe the software running characteristics. Firstly, principal component analysis (PCA) is used to reduce the dimension of software running feature data and to interpret software characteristic information. Then the modified K-means algorithm was used to classify the meteorological data processing software. Finally, it combined with the results of principal component analysis to explain the significance of various types of integrated software operating characteristics. And it is used as the basis for optimizing the allocation of software hardware resources and improving the efficiency of software operation.

Keywords

Principal Component Analysis, Improved K-Mean Algorithm, Meteorological Data Processing, Feature Analysis, Similarity Algorithm

1. Introduction

With the increase of meteorological satellite observation level and the rich variety of remote sensing products, meteorological satellite remote sensing products are more and more widely used. National Satellite Meteorological Center Fengyun meteorological satellite ground application system deals with a large

number of satellite observations in real time every day. It has put forward high requirements on the timeliness and reliability for ground application system data processing. At the same time, it challenges the design and operation of these applications to support the work of IT platform. How to fully understand the resource requirements of various types of data processing software and make effective use of IT resources has become an urgent problem in the field of meteorological satellite ground application system [1].

Various types of meteorological satellite data processing prototype software in National Satellite Meteorological Center are the crystallization of countless scientists' hard work for several years. With the development of remote sensing instruments and the development of remote sensing technology, the ground meteorological software is constantly enriched and renewed. In engineering construction, these prototyping softwares become an important component of Fengyun meteorological satellite ground application system after engineering. It should be necessary to establish the detection and evaluation methods, after the engineering data processing software and the use of hardware resources to assess the rationality. Fengyun meteorological satellite ground application system has a large number of data processing software, so classification of software resources and the use of the characteristics of its operation are the basis for carrying out evaluation work.

Experimental data used in this paper are from the collected data on the operation of Fengyun-3C data processing software. First of all, we collected the original software running feature data, processing feature extraction, to better express the characteristics of the software. Secondly, principal component analysis (PCA) was used to analyze the operational characteristics of the collected data, and the principal components were extracted and their features were described. Then the clustering analysis is carried out by using the processed software characteristic data to realize the classification of meteorological software, such as computing-intensive, memory-intensive, I/O-intensive and network-intensive. Finally, based on the results of PCA, the characteristics of each type of software are described, which provides basic data and basis for further work, such as software resource consumption rationality analysis, software operation rationality evaluation, optimization of hardware and software systems, and provides scientific decision data support for future hardware and software platform planning and configuration of new projects. Therefore, the classification of software based on software operating characteristics, so as to further optimize the software hardware resource allocation and improve software operating efficiency.

2. Extraction and Processing of Software Running Feature

2.1. Software and Hardware Environment Overview

The object of this paper are the 182 sets of polar orbiting meteorological satellite data processing software of the 12 categories of instruments for the Fengyun-3C satellite ground application system. Hardware resources, including 6 IBM minicomputers, detailed configuration in Table 1.

Table 1. Hardware configuration.

serv_name	cpu_hz	cpu_type	cpu_core	real_mem	vir_mem
coss1-3c	4.4 Ghz	POWER7	32	186 G	62 G
coss2-3c	4.4 Ghz	POWER7	32	186 G	62 G
pgs1-3c	4.4 Ghz	POWER7	22	134 G	52 G
pgs2-3c	4.4 Ghz	POWER7	22	134 G	52 G
pgs6-3c	4.2 Ghz	POWER7	12	250 G	78 G
pgs7-3c	4.2 Ghz	POWER7	12	250 G	78 G

2.2. Maintaining the Integrity of the Specifications

Software operating characteristic data acquisition range contained 182 sets of polar orbit meteorological satellite data processing software for 12 kinds of instruments. Polar orbit meteorological satellites carried remote sensing instrument. Its mode of operation is to collect data on a regular basis and download the collected data to the ground station. The software needs to run multiple times per day (each run is called a track). Data acquisition environment is the simulation environment and acquisition time is 4 days. The collection method for operating characteristics of the software is to force each weather processing software running in serial (the actual environment running is in parallel), so that each software can get sufficient hardware resources and give full play to software performance. Software operating characteristics data acquisition types included CPU, system, process, and job level data, with CPU-level and system-level acquisition cycles of 1 second. Job-level data acquisition fields are the main software start time, end time and the located server. System-level data acquisition fields are CPU system and disk wait for usage, CPU idle usage, memory usage, virtual memory usage, disk read and write rates, network receive and send rate. CPU-level data acquisition field has the core CPU system utilization and idle utilization.

2.3. Characterization of Operational Characteristics

Software feature analysis needs to express the operating characteristics of the software as much as possible, and ultimately to express the operating characteristics of each software through a vector. Characterization of the software running characteristics need to consider from two aspects: 1) time-series characteristics of software operation; 2) to eliminate differences between the platforms and the resource consumption of the system (only consider the resources consumed by the software itself).

The time-series features of the software running are represented by peak, mean and summation of resource consumption. Eliminating platform differences requires the conversion of resource usage to usage. The consumption of resources of the software itself needs to throw away the occupied resources of the system. To this end, we carried out based on the parameters of the collected information and software running on the server information, synthesis of new

feature parameters. Specific treatment is as follows:

Software running time:

$$t = t_{\text{start}} - t_{\text{end}} \tag{1}$$

CPU user calculation:

$$\text{cpu\_am} = \left(100 - \text{cpu\_rat}_{\text{syswa}} - \text{cpu\_rat}_{\text{idle}}\right) * \text{cpu}_{\text{core}} * \text{cpu\_hz} \tag{2}$$

CPU calculation total:

$$\text{cpu\_sum} = \sum_{i=0}^{t} \text{cpu\_am}_i \tag{3}$$

CPU Calculated Peak:

$$\text{cpu\_max} = \max\left(\text{cpu\_am}_i\right) \tag{4}$$

Memory usage:

$$\text{mem} = \text{mem\_rat} * \text{real\_mem} \tag{5}$$

Virtual memory usage:

$$\text{swap} = \text{swap\_rat} * \text{vir\_mem} \tag{6}$$

Disk Read:

$$\text{disk\_read\_sum} = \sum_{i=0}^{t} \text{disk\_read}_i \tag{7}$$

Disk write:

$$\text{disk\_write\_sum} = \sum_{i=0}^{t} \text{disk\_write}_i \tag{8}$$

Network receiving:

$$\text{net\_rec\_sum} = \sum_{i=0}^{t} \text{net\_rec}_i \tag{9}$$

Network sending:

$$\text{net\_send\_sum} = \sum_{i=0}^{t} \text{net\_send}_i \tag{10}$$

Through the above conversion, the software's each track operating characteristic data is transferred into a vector, and then we calculate the average value of the software multi-track running characteristics, finally formatted a 182 × 14 data matrix of the original operating characteristics.

2.4. Characteristic Data Normalization

In the original data, the unit of each characteristic parameter value is not the same, and the difference between the data is very big. In order to facilitate the analysis, the data are normalized. In this paper, the Min-max normalization method is used to transform the original data linearly. Let min$A$ and max$A$ be the minimum and maximum values of attribute A, and normalize the original value x of A by Min-max to the value in interval [0, 1]. The formula is:

$$x' = \left(x - \min A\right) / \left(\max A - \min A\right) \tag{11}$$

3. Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method. Through ortho-

gonal transformation to a group of variables may be related to the conversion of a group of linearly unrelated variables, the group of variables after transformation is called the principal component. The results of principal component analysis are mainly dependent on the correlation between indicators. If the correlation is very strong, the results of principal component analysis will be very good, otherwise it is poor [2]. Principal Component Analysis method can reduce the software operation characteristic data dimension and explain the software characteristic information.

In this paper, SPSS is used to analyze the original running characteristic data matrix. The correlation between 14 features was calculated firstly, and the results are shown in Table 2. The total variance is then explained. Finally, the principal components are selected and their features are extracted. The calculation method and steps of the characteristic analysis method are as follows.

3.1. Compute the Correlation Matrix from the Original Data Matrix

The raw data matrix represents the operating characteristics of each software, and each column represents the value of one operating characteristic of the software. SPSS software analysis results are shown in Table 2. The matrix reflects the correlation between the running characteristics of any two software programs.

3.2. The Principal Component Is Extracted by Total Variance

According to the Ref. [3], when ρ (cumulative%) ≥ 0.8 - 0.9, we can use the first five principal components instead of the original 14 operating characteristics, and retain the original 14 operating characteristics contain the main information, The first five principal components are called public influence factors.

3.3. Calculation of the Main Components of the Software

According to the analysis in Table 3, the cumulative values of the four principal components of 1, 2, 3, 4 are 78.471%, which can represent the main factors of the original matrix. In the process of running, the expression of the variable is not the original variable, but the standardized variable, such as the first principal component, for example, can be other standardized variables:

$$F_1 = 0.732 * Zx_1 + 0.547 * Zx_2 + \cdots + 0.254 * Zx_{14} \quad (12)$$

By analyzing the four principal component coefficients in Table 4, the operational characteristics with high correlation coefficient are selected as the analysis factors. In Table 5, it can be found that the main components in the first category are mainly related to run time and disk read and write resources. The second category is mainly related to network resources and CPI. The third category is mainly related to computing resources. And the forth category are related to memory and cache.

From Table 4 and Table 5, the new principal component formula is extracted as follows:

Table 2. Hardware configuration.

	Running time	The maximum cache size	The maximum memory	Computational complexity	Computational peak value	CPI average	Disk read total	Disk read peak	Disk write total	Disk write peak	Network Sending Total	Network Sending Peak	Network Receiving Total	Network Receiving Peak
Running time	1	0.326	0.069	0.399	0.354	−0.617	0.534	0.54	0.48	0.546	0.52	0.023	0.273	−0.012
The maximum cache size	0.326	1	0.774	0.193	0.212	−0.364	0.331	0.445	0.261	0.276	0.159	0.269	0.044	−0.002
The maximum memory	0.069	0.774	1	0.127	0.124	0.159	0.166	0.258	0.135	0.188	0.268	0.586	0.358	0.354
Computational complexity	0.399	0.193	0.127	1	0.892	−0.307	0.084	0.179	0.066	0.263	0.271	0.081	0.265	0.041
Computational peak value	0.354	0.212	0.124	0.892	1	−0.354	0.118	0.272	0.079	0.396	0.249	0.098	0.228	0
CPI average	−0.617	−0.364	0.159	−0.307	−0.354	1	−0.255	−0.42	−0.183	−0.292	−0.115	0.228	0.15	0.412
Disk read total	0.534	0.331	0.166	0.084	0.118	−0.255	1	0.586	0.938	0.546	0.438	0.192	0.208	0.009
Disk read peak	0.54	0.445	0.258	0.179	0.272	−0.42	0.586	1	0.406	0.721	0.426	0.161	0.22	−0.004
Disk write total	0.48	0.261	0.135	0.066	0.079	−0.183	0.938	0.406	1	0.569	0.374	0.174	0.164	0.011
Disk write peak	0.546	0.276	0.188	0.263	0.396	−0.292	0.546	0.721	0.569	1	0.482	0.174	0.327	0.079
Network Sending Total	0.52	0.159	0.268	0.271	0.249	−0.115	0.438	0.426	0.374	0.482	1	0.66	0.761	0.39
Network Sending Peak	0.023	0.269	0.586	0.081	0.098	0.228	0.192	0.161	0.174	0.174	0.66	1	0.555	0.452
Network Receiving Total	0.273	0.044	0.358	0.265	0.228	0.15	0.208	0.22	0.164	0.327	0.761	0.555	1	0.706
Network Receiving Peak	−0.012	−0.002	0.354	0.041	0	0.412	0.009	−0.004	0.011	0.079	0.39	0.452	0.706	1

Table 3. Explain the total variance.

Ingredients	Initial eigenvalue			Extract the square load		
	Total	Variance %	Cumulative %	Total	Variance %	Cumulative %
1	4.948	35.346	35.346	4.948	35.346	35.346
2	2.724	19.457	54.803	2.724	19.457	54.803
3	1.815	12.965	67.768	1.815	12.965	67.768
4	1.498	10.702	78.471	1.498	10.702	78.471
5	0.813	5.808	84.278			
6	0.666	4.759	89.037			

Table 4. Component matrix.

Parameter	Original indicators	Ingredients			
		1	2	3	4
Zx_1	Running time	**0.732**	−0.375	0.044	−0.168
Zx_2	The maximum cache size	0.547	−0.062	−0.054	**0.798**
Zx_3	The maximum memory	0.469	0.48	−0.021	**0.706**
Zx_4	Computational complexity	0.473	−0.165	**0.779**	−0.03
Zx_5	Computational peak value	0.506	−0.214	**0.744**	−0.006
Zx_6	CPI average	−0.386	**0.715**	−0.165	−0.09
Zx_7	Disk read total	**0.722**	−0.212	−0.512	−0.128
Zx_8	Disk read peak	**0.736**	−0.25	−0.175	0.082
Zx_9	Disk write total	**0.652**	−0.192	−0.531	−0.17
Zx_{10}	Disk write peak	**0.766**	−0.184	−0.105	−0.154
Zx_{11}	Network Sending Total	**0.755**	0.356	0.024	−0.312
Zx_{12}	Network Sending Peak	0.474	**0.681**	−0.025	0.135
Zx_{13}	Network Receiving Total	0.573	**0.63**	0.173	−0.32
Zx_{14}	Network Receiving Peak	0.254	**0.775**	0.071	−0.182

Table 5. Main ingredient.

1	2	3	4
Running time	CPI average	Computational complexity	The maximum cache size
Disk readtotal	Network Sending Peak	Computational peak value	The maximum memory
Disk readpeak	Network Receiving Total		
Disk writetotal	Network Receiving Peak		
Disk writepeak			
Network SendingTotal			

$$F_1 = 0.732 * Zx_1 + 0.722 * Zx_7 + 0.736 * Zx_8 + 0.652 * Zx_9 + 0.766 * Zx_{10} + 0.755 * Zx_{11} \quad (13)$$

$$F_2 = 0.715 * Zx_6 + 0.681 * Zx_{12} + 0.63 * Zx_{13} + 0.775 * Zx_{14} \quad (14)$$

$$F_3 = 0.779 * Zx_4 + 0.774 * Zx_5 \quad (15)$$

$$F_4 = 0.798 * Zx_2 + 0.706 * Zx_3 \quad (16)$$

4. Improved K-Means Algorithm

4.1. Typical K-Means Algorithm

K-means algorithm is a typical distance-based clustering algorithm. Distance was used as the evaluation index of similarity. That is, the closer the distance between two objects, the greater the similarity. The clustering results of the traditional K-means algorithm are susceptible to the number of clusters [4] [5] [6]. The choice of the initial cluster center depends on relatively large. The clustering results of different initial clustering centers are usually different. The results are highly uncertain. The clustering index tends to converge to the local optimum. K-means algorithm can be used to classify polar orbit meteorological data processing software.

Nowadays, the research on k-means algorithm is mainly focused on two directions: firstly, how to obtain better initial clustering center; the second is how to get the best clustering number. For the selection of the initial clustering center point of k-means algorithm, Ref. [7] [8] proposed that k points in high density distribution are chosen as the initial clustering center algorithm. In this paper, the first k initial center points are chosen and the k values are combined according to the clustering results [9] [10].

4.2. Cluster Analysis Results

Through the improved K-means algorithm, the clustering results are shown in the following **Figure 1**, and three types are obtained.

Combining the principal component analysis results and the clustering results, the following results are easily obtained: In the first category, the third principal component value of the individual is higher, the first principal component is medium, the requirements of memory and cache are relatively high, the disk and network resource demand are moderate, and the CPU resource requirement is low; In the second category, the three main components are very high, the software is an integrated intensive, the disk, network and memory requirements are relatively large, especially for CPU requirements are particularly large; In the third category, the three main components are relatively low, which are small-scale resource-intensive. The running time of this kind of software is relatively short, and the demand for various resources is low. The overall demand for network and memory is relatively high.

4.3. Clustering Analysis

Through **Figure 1** and **Figure 2** it can be found, the overall CPU utilization is low. We can reduce the CPU configuration or add applications on this server software.

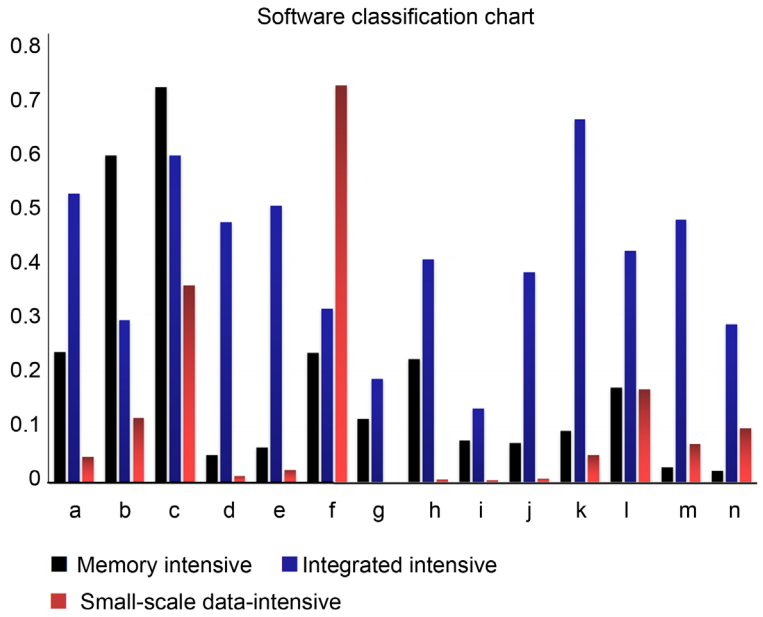

Figure 1. Software classification feature map. (Notes: a. Running time; b. The maximum cache size; c. The maximum memory; d. Computational complexity; e. Computational peak value; f. CPI average; g. Disk read total; h. Disk read peak; i. Disk write total; j. Disk write peak; k. Network Sending Total; l. Network Sending Peak; m. Network Receiving Total; n. Network Receiving Peak).

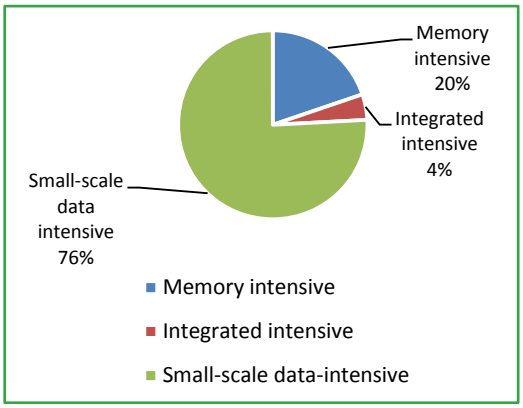

Figure 2. Software classification results.

More intensive and centralized memory, we can increase the memory capacity to enhance the speed. Software on the disk read and write speed is relatively high, sudden traffic increases, the proposed disk is equipped with high speed or increase the multilevel cache resources to reduce the disk read and write on the software calculation.

The results of the research and experiment prove that the above results are in accordance with the actual situation and can be used as the basis for optimization of hardware and software resources.

5. Summary

In this paper, principal component analysis and K-means clustering algorithm

are used to classify the software in the meteorological field, and the software classification in the meteorological field is solved. At the same time, the characteristics of each kind of software are analyzed. Using the results of the classification, the software scheduling algorithm is further analyzed to improve the hardware utilization and reduce the software waiting time.

Acknowledgements

The work presented in this study is supported by National High-tech R&D Program (2011AA12A104).

References

[1] Chen, Z. and Luo, C.C. (2015) Application of an Improved K-Means Algorithm in Anomaly Detection. *Journal of Chongqing University of Technology*: *Natural Science*, No. 5, 66-70.

[2] Fang, C., Yang, Y. and Wu, S.J. (2009) Application of Principal Component Analysis and Cluster Analysis in Software Reconstruction. *Computer Engineering and Design*, **30**, 365-369.

[3] Li, Z.-Y., Ding, J. and Peng, L.-H. (2004) Principles and Methods of Environmental Quality Assessment. Chemical Industry Press, Beijing.

[4] Jia, R.-Y. and Song, J.-L. (2016) K-Means Optimal Cluster Number Determination Method Based on Clustering Center Optimization. *Microelectronics & Computer*, **33**, 62-66.

[5] Yin, C.-X., Zhang, H.-J., Zhang, R., Qi, X.-L. and Wang, B. (2014) An Improved K-Means Algorithm. *Computer Technology and Development*, **24**, 30-33.

[6] Li, Y.-S., Yang, S.-L. and Ma, X.-J. (2006) Study on K-Value Optimization in Spatial Clustering Algorithm. *Journal of System Simulation*, **18**, 573-576.

[7] Mac Queen, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the* 5*th Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 281-297.

[8] Huang, F., Da, P., Lin, Q.H., Zhou, J.M., *et al.* (2009) An Improved Similarity Algorithm for Personalized Recommendation. *International Forum on Computer Science-Technology and Applications*, 25-27 December 2009, 54-57. https://doi.org/10.1109/ifcsta.2009.20

[9] Abraham, M.H., Grellier, P.L., Prior, D.V., *et al.* (1990) Hydrogen Bonding. Part 10. A Scale of Solute Hydrogen-Bond Basicity Using Log K Values for Complexation in Tetrachloromethane. *Journal of the Chemical Society*, *Perkin Transactions*, 2, 521-529. https://doi.org/10.1039/p29900000521

[10] Söylev, T.A. (2016) Comparison of Measured and Prescribed K-Values for the Equivalent Performance of Fly Ash Concrete. Service Life of Cement-Based Materials and Structures, 187.