PROCEEDINGS OF SCIENCE

Assembly Data Mining Platform based on Python

Kai Zhang[1]

College of Mechatronics Engineering and Automation, National University of Defense Technology
Changsha, 410073, China
*E-mail:*kai277063148@163.com

Guoxi Li[2]

College of Mechatronics Engineering and Automation, National University of Defense Technology
Changsha, 410073, China
*E-mail:*lgx2020@sina.com

Meng Zhang

College of Mechatronics Engineering and Automation, National University of Defense Technology
Changsha, 410073, China
*E-mail:*z.mengdr@gmail.com

There are rich data in the manufacturing information systems, but they are not utilized in an effective way. The establishement of assembly data mining platform is to take advantage of the data to improve assembly process. Assembly factors and assembly data mining templates are the core tables in the assembly database. The methods of dealing with data vacancies and noise points are pre-set in the data mining templates. Python is used as the data mining engine by script customization and algorithm library encapsulation. Firstly, Python algorithm scripts are customized when programing the platform. Then the platform generates the execution script according to the user operation. Finally, the main program of Python executes the generated script and returns the results. Also, Mlpy is applied to make corresponding algorithm processing module. The functions are pre-compiled so that the assembly technicians without knowledge of data mining can utilize the assembly data to predict the assembly performance and analyse the assembly process.

ISCC2017
16-17 December 2017
Guangzhou, China

[1] Speaker

[2] This study is supported by National Natural Science Foundation of China (Grant No. 51775552)

1.Introduction

As the last link in the manufacture of mechanical systems, Assembly has important influence on the dynamic performance[1]. There are many factors such as component properties, assembly process, assembly method and assembly environment which may affect the performance. Some of them are non-linear and hard to control. Furthermore, each part has different geometric errors of each assembly. Thus, it is difficult to make theoretical analysis on the key assembly processes of complicated assembly. At present, the assembly shop takes long-term continuous trial, repeated adjustment and accumulation of experience to guarantee the mechanical product performance. This costs a large amount of cost and time[2].

On the other hand, with the rapid development of data mining technology and the gradual popularization of information systems such as PDM, CAPP, ERP and MES in recent years, data mining on the assembly process data becomes a feasible solution to achieve assembly performance prediction[3]. Kretschmer introduces the implementation for knowledge-based design for assembly in agile manufacturing by using data mining methods in the field of series production with high variance[4]. Wallis presents an approach for data mining-supported generation of assembly process plans to enhance planning efficiency based on the classification and clustering of product and process data[5]. Schmitt explores the application of data mining to a precision assembly task to maximize the yield and machine uptime at the full scope of available data[6]. Cheng reviews the development and makes discussion of data mining techniques on the applications production management[7].

Python is an object-oriented scripting language and widely used in data mining nowadays[8]. It has many advantages such as flexibility, open source, good scalability and rich library of methods. Machine Learning Python(mlpy) is a high-performance Python library for predictive modeling. In light of the previous research[9], an assembly data mining platform is built based on Python by managing assembly process data, setting up assembly database and establishing assembly data mining template. Mlpy is applied to make corresponding algorithm processing moduleby which the assembly technician can learn the knowledge of the assembly process data and boost product accuracy.

2.Platform Structure

According to the analysis of the assembly data mining process[10], the assembly data mining platform structure is divided into support layer, integration layer, functional layer and user layer(Figure 1). The support layer mainly includes assembly data sources, assembly mining templates and data mining function modules. The data come from the database of enterprise information system. The manufacturing system database provides product assembly BOM, material parameters, dimension tolerances, and product performance testing results, etc. The integration layer provides access to data exchange among the platform, algorithm and database. The interfaces include database, Web services, algorithm libraries and visualization tools. Function layer is the core of assembly data mining system including assembly information management, assembly process management and data mining process control. The user layer is widely used and maintained by assembly process designers, assembly technicians, quality inspectors and data mining technicians. The user layer provides interfaces for different users to simplify functions and the operation process.

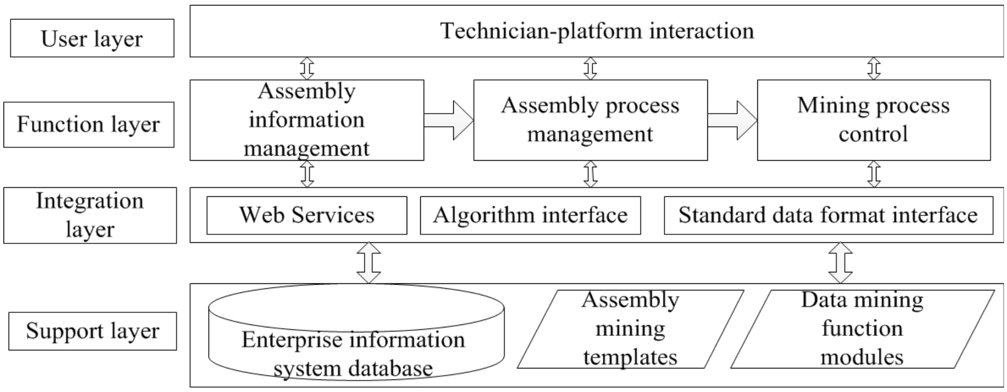

User layer
Technician-platform interaction
Function layer
Assembly information management
Assembly process management
Mining process control
Integration layer
Web Services
Algorithm interface
Standard data format interface
Support layer
Enterprise information system database
Assembly mining templates
Data mining function modules

Figure 1: Assembly Data Mining Platform Structure

3.Assembly Data Integration

The assembly data mining needs the whole process data, since the different assembly data may come from different manufacturing departments and the data may be saved in different databases. Besides, the database and data format may be programmed by different software companies. As a result, the data cannot be effectively shared and compatible. Therefore, assembly data integration is necessary.

3.1Assembly Database Information Model

IDEF1X method is used to establish database information model. The information models of assembly factors and assembly data mining templates are introduced. Figure 2(a) is the assembly associated relationship information model.

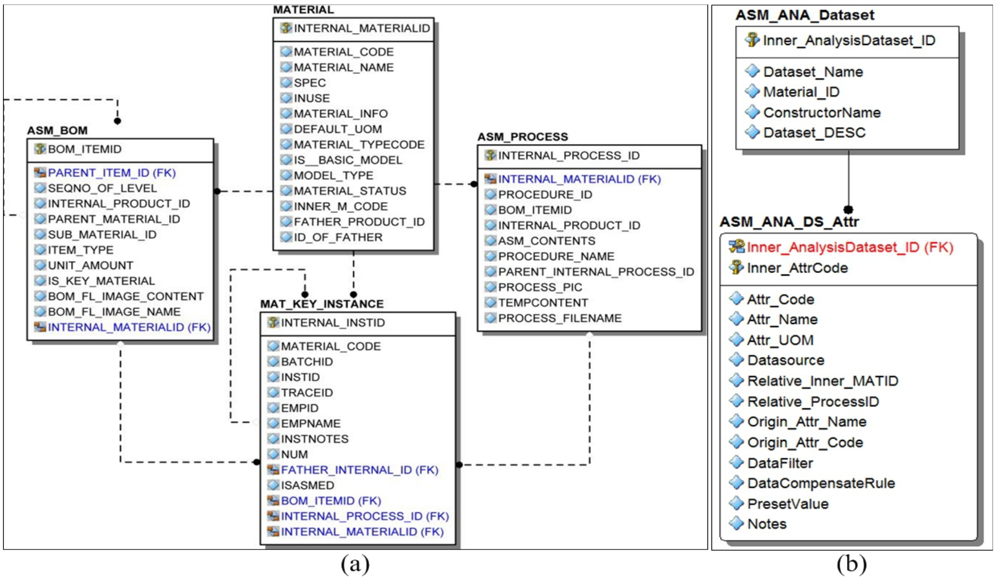


Figure 2: Information Models of Assembly Factors and Assembly Data Mining Templates

The material is the root node and the internal code is the foreign key relationship among BOM, process and instance. Each table is attached to the sub-tables, such as parts, parameters, assembly relationship, assembly process parameters table, etc. The instance table refers to each assembly product and inherits all the basic assembly information. Figure 2(b) is the information model of assembly data mining template. Each template includes the source of the data column for the current dataset, preprocessing algorithm and missing data completion rules.

3.2 Assembly Data Preprocessing

Data preprocessing is the key step of data mining and directly affects the accuracy of results. The data have noise points and vacancies. As each assembly process parameter has their own tolerance range, it only needs to identify the instance with noise points. Then the assembly technician can analyze whether the parameter is a data entry error and identify the eligibility of the instance. The methods of dealing with data vacancies include deleting the current data instance, completing the preset value, the completing the distribution rules value and manual inputting. Distribution rules include uniform distribution, normal distribution and Weibull distribution.

4. Python Algorithm Integration

The platform achieves the algorithm integration by calling the Python script. Firstly, Python algorithm scripts are customized when programing the platform. Then the platform generates the execution script according to the user operation. Finally, Python main program executes the generated script and returns the results.

4.1 Algorithm Execution Flow

Figure 3 shows the algorithm execution flow chart, which mainly refers to the process of operation when user is operating the platform. The main features are the algorithm preprocessing, target data setting and test data setting. Algorithm preprocessing is inevitable according to algorithm characteristics. For example, k-Nearest-Neighbor for classification requires the assembly technician to pre-classify and specify the data column classification label.

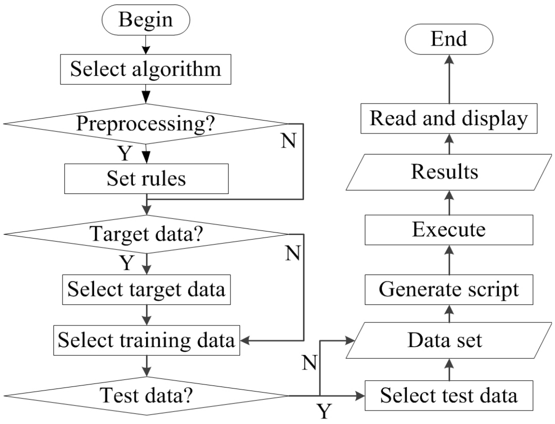


Figure 3: The Algorithm Execution Flow

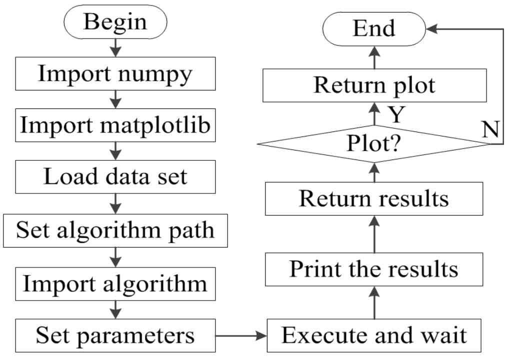


Figure 4: Python Script Customization Flow

4.2 Python Script Customization

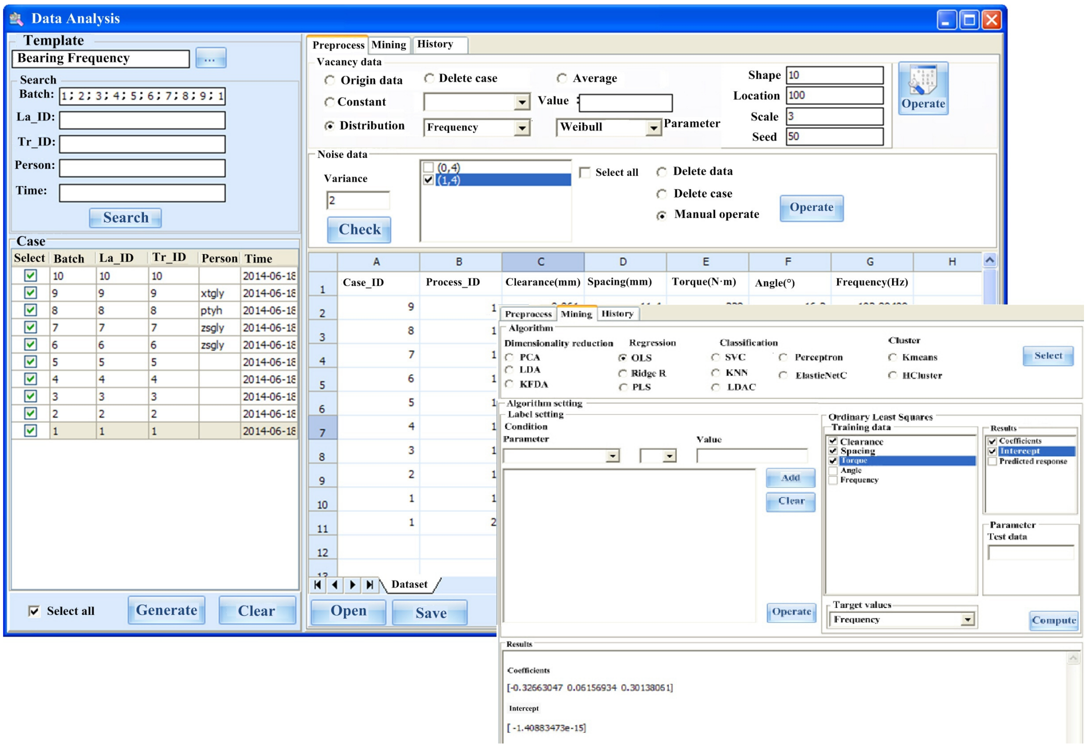
Python script is generated dynamically and runs automatically by the platform after the user performs data mining operations. The main content is loading the algorithm library, setting the data input and output paths and executing the algorithm. Numpy component is used to calculate the N-dimensional array of the same element. Matplotlib component is used to draw the data mining result map. The specific process is displayed in Figure 4.

4.3 Algorithm Library Encapsulation

The function of each algorithm in mlpy has its own class, parameters and return values. Algorithm library encapsulation means programming the algorithm script in accordance with the characteristics of the respective functions with the data mining process. Thus when the platform calls the algorithm library, the execution script only need to set the input and output data with standard format. For example, Ridge regression is known as regularized least squares , and itsmain function invloves training data, targeting values and regulating parameter. And training data must be centered by columns. After computing the regression coefficients and intercept, the function can predict the response of the test data.

4.4 Platform Application and Case Study

Figure 5 is the data mining interface of the platform. Here is a case about the frequency study of bearing assembly so as to find out the relationship of axial bearing clearance, spacing, driving torque, bearing initial contact angle and first order frequency. The assembly technicians choose the data mining template of bearing-frequency and then select ten assembly instances in the database. The platform extracts the process parameters in the assembly instance according to the preset parameters in the template and automatically generates the training data set.



Template: Bearing Frequency

Search
Batch: 1; 2; 3; 4; 5; 6; 7; 8; 9; 1
La_ID:
Tr_ID:
Person:
Time:

Case

Select	Batch	La_ID	Tr_ID	Person	Time
☑	10	10	10		2014-06-18
☑	9	9	9	xtgly	2014-06-18
☑	8	8	8	ptyh	2014-06-18
☑	7	7	7	zsgly	2014-06-18
☑	6	6	6	zsgly	2014-06-18
☑	5	5	5		2014-06-18
☑	4	4	4		2014-06-18
☑	3	3	3		2014-06-18
☑	2	2	2		2014-06-18
☑	1	1	1		2014-06-18

☑ Select all

Vacancy data
○ Origin data ○ Delete case ○ Average
○ Constant Value
● Distribution: Frequency, Weibull; Parameter: Shape 10, Location 100, Scale 3, Seed 50

Noise data
Variance: 2
□ (0,4)
☑ (1,4)
□ Select all ○ Delete data ○ Delete case ● Manual operate

	A	B	C	D	E	F	G	H
1	Case_ID	Process_ID	Clearance(mm)	Spacing(mm)	Torque(N·m)	Angle(°)	Frequency(Hz)	
2	9	1						
3	8	1						
4	7	1						
5	6	1						
6	5	1						
7	4	1						
8	3	1						
9	2	1						
10	1	1						
11	1	2						
12								

Algorithm
Dimensionality reduction: ○ PCA ○ LDA ○ KFDA
Regression: ● OLS ○ Ridge R ○ PLS
Classification: ○ SVC ○ KNN ○ LDAC ○ Perceptron ○ ElastieNetC
Cluster: ○ Kmeans ○ HCluster

Algorithm setting
Label setting
Condition
Parameter Value

Ordinary Least Squares
Training data
☑ Clearance
☑ Spacing
☑ Torque
□ Angle
□ Frequency

Results
☑ Coefficients
☑ Intercept
□ Predicted response

Parameter
Test data

Target values: Frequency

Results
Coefficients
[-0.32663047 0.06156934 0.30138061]
Intercept
[-1.40883473e-15]

Figure 5: Data Mining Interface of the Platform

Ordinary least squares regression is used to analyze the problem and angle is not selected as the training data. This method returns the coefficients and intercept and the expression can be written as Frequency=-0.32663047Clearance+0.06156934Spacing+0.30138061Torque. From the coefficients as it shows, clearance and torque have exerted greater influences on the first order frequency compared with spacing. And the first order frequency increases with the increment of torque and diminution of clearance.

5. Conclusion

Assembly data mining platform is introduced and Python is proposed as the data mining algorithm library. The information models of assembly factors and assembly data mining templates are built according to the assembly process. To simplify the data mining process, the methods of dealing with data vacancies and noise points are preset in the data mining templates. To reduce the difficulty of operating the platform, Python script was customized and encapsulated in the platform. In this way, the assembly technicians without knowledge of data mining can utilize the assembly data to predict the assembly performance and analyze the assembly process. With the proposed platform, data mining technology can be applied to manufacturing data in the enterprise information systems and databases.

References

[1] F. Wan, G. Li, J. Gong, B. Wu. *An improved algorithm for the normal contact stiffness and damping of a mechanical joint surface* [J]. Proc IMechE Part B: J Engineering Manufacture, 2014, 228(5):751-765

[2] F. Wan, G. Li, J. Gong, B. Wu. *Dynamic study of anti-backlash gears and angle-contact ball bearing based on a simplification of the Hertz contact theory* [J]. Proc IMechE Part B: J Engineering Manufacture, 2016, 230(1):66-82

[3] P. Vazan, D. Janikova, P. Tanuska, M. Kebisek, Z. Cervenanska. *Using data mining methods for manufacturing process control* [J]. IFAC PapersOnLine, 2017, 50-1:6178-6183

[4] R. Kretschmer, A. Pfouga, S. Rulhoff, J. Stjepandic. *Knowledge-based design for assembly in agile manufacturing by using Data Mining methods* [J]. Advanced Engineering Informatics, 2017, 33:285-299

[5] R. Wallis, O. Erohin, R. Klinkenberg, J. Deuse, F. Stromberger. *Data Mining-supported Generation of Assembly Process Plans* [J]. Procedia CIRP, 2014, 23:178-183

[6] R. Schmitt, F. Dietrich, K. Dröder. *Big Data Methods for Precision Assembly* [J]. Procedia CIRP, 2016, 44:91-96

[7] Y. Cheng, K. Chen, H. Sun, Y. Zhang, F. Tao. *Data and knowledge mining with big data towards smart production* [J]. Journal of Industrial Information Integration, 2017, http://dx.doi.org/10.1016/j.jii.2017.08.001

[8] M. Wagner, G. Llort, E. Mercadal, J. Gimenez, J. Labarta. *Performance Analysis of Parallel Python Applications* [J]. Procedia Computer Science, 2017, 108C:2171-2179

[9] K. Zhang, G. Li, B. Wu, M. Zhang. *Python-based experimental analysis of machinery assembly data mining* [C]. 2nd International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SMTA 2015) Paris, France Aug 09-10, 2015:90-93

[10] K. Zhang, G. Li, J. Gong, M. Zhang. *Research on Assembly Data Mining Technology of Complex Mechanical System* [J]. Procedia CIRP, 2016, 44:97-101