Clustering and Dimensionality Reduction to Determine Important Software Quality Metrics

Metin Turan
Kültür University and Istanbul Technical University, Istanbul, Turkey
Zehra Çataltepe
İstanbul Technical University, Istanbul, Turkey

***Abstract*— During the last two decades research on software engineering is concentrated on quality. The best approach to quality evaluation goes through determining well-defined metrics on software properties. One such property is module complexity, which is a view of the software that is related to how easily it can be modified.**

There has been work on constructing a metrics domain which measures the module complexity. Generally, PCA (Principal Component Analysis) is used for defining principal metrics in the domain. Since there are usually no labels for the software data, an unsupervised dimensionality reduction technique, such as PCA needs to be used for determining the most important metrics.

In this study, we use the clustering similarity obtained when a certain subset of metrics and when the whole set of metrics are used, to determine the most important metrics. We measure the relative difference/similarity between clusterings using three different indices, namely Rand, Jaccard and Fowlkes-Mallow. We use both backward feature selection and PCA for dimensionality reduction. On the publicly available NASA data, we find out that instead of the whole set of 42 metrics, using only 15 dimensions, we get almost the same clustering performance. Therefore, a smaller number of the metrics could be used to evaluate software quality.

I. Introduction

Module complexity is a measure of software quality. It shows how modules are designed and implemented. If complexity is high for a module, the module must be reorganized to meet quality criteria. Module complexity can be identified using internal and external complexity. Internal complexity shows the complexity of the code used within the module. External complexity is related to the interface (parameter list and types) between modules and is measured by the interconnections from one module to the others.

External complexity, also called module coupling, is a well-studied area of module complexity. Statistical techniques are used to build models that relate coupling measures to qualitative aspects of software, such as reliability and maintainability. Previously, Munson and Khoshgoftaar [1] developed a domain model of software attributes to study all aspects of program complexity using PCA (Principal Component Analysis). Briand, et al. [2] also used PCA for a similar purpose while studying a large collection of highly correlated coupling metrics in object-oriented systems. Hall, Tao and Munson [3] built a domain model which is extended to represent the software attribute of module coupling. They again used PCA to find determinant metrics in the domain.

It is hard to decide on the most important metrics, because often the code metric inputs are available, however there are no labels or outputs that define a classification task. Therefore, one can not decide on a better set of metrics based on classification accuracy. When PCA is used, it is assumed that all the inputs are distributed according to a single multivariate normal distribution and the most important dimensions are selected based on that assumption. Normality assumption doesn't need to hold, for example inputs could be distributed according to a number of normals with different parameters. By means of clustering the metric data, we are able to decide on the best metrics for such cases. We get advantage of natural groupings of data, which may correspond to class labels for an application. We compare a reduced dimensional set of metrics to the full set of metrics by comparing the clusterings produced in both cases. If the clusterings are close, in other words if any two data points that fall in the same/separate clusters using the reduced set of dimensions also fall in the same/separate clusters using the full set, we decide that the reduced set of dimensions gives similar performance to the full set data set. We measure the similarity between two clusterings by means of external cluster indices, namely Rand, Jaccard and Fowlkes-Mallow indices. As dimensionality reduction techniques we use both PCA and backward feature selection. When PCA is used, each reduced dimension is a linear combination of the original set of dimensions. When backward feature selection is used, each reduced dimension corresponds to a single original dimension. Backward feature selection produces a subset of selected dimensions, therefore, the dimensions that are not important do not need to measured. This saves both time and energy when features are generated.

We use the publicly available NASA software metrics repository data in our experiments. The software metrics repository consists of thirteen NASA software projects, and is available on the MDP website [5]. This repository provides

project-independent data to the software community to support research in predicting quality of software systems.

The rest of the paper is organized as follows: The metrics names and dimension numbers of the NASA data set are given in at Part A of the II. Work Section. In Part B, the algorithm used for selection of external indices is explained. Definitions of external indices (Rand, Jaccard, Fowlkes-Mallow) and results for different number of clusters (5, 10 and 30) are given in part C. In Part D, we use PCA for dimensionality reduction and compare this with backward feature selection. Finally, we summarize our results and possible future research directions in Part III.

II. WORK

A. Data

The NASA data that we use in our experiments consist of software metrics extracted for different set of software modules. This data set has been used, for example, by Liu et.al. [7] where they tried to discover patterns in software metrics which classify modules as low-quality and high-quality. The JM1 and KC2 data from MDP system have been used by Khoshgoftaar and Raton [8]. They showed that quality of a software measurement is affected by the data set. For this study, we used PC4 and CM1 data sets. The available inputs for these data sets are shown in Table I. Metrics. Please see the NASA web site for more information

B. Algorithm

We give the algorithm that we use to compute the most important features as a pseudocode below:

```
BackwardSelect (dataSet, noOfClusters) {
  Make an initial clustering C using all dimensions
  Fill dimension list with all dimensions in the data
  for NoOfDimensions {
    for each dimension d left in the dimension list {
      Construct a dataset leaving out dimension d
      Cluster this dataset using this data set
      Construct contingency table between
        new clustering and clustering C
      Calculate related external indices
        (Use initial contingency table for comparison)
      if indices value is the maximum {
          Save its dimension
          Save external indices value
      }
      Delete dimension with maximum index
          value from the dimension list
    }
  }
  return dimensions with the maximum index value
  and the external index values saved when
  those dimensions are eliminated
}
```

TABLE I. METRICS

METRIC	DIMENSION#
1-McCabe Metrics	
Cyclomatic Complexity	7
Cyclomatic Density	8
Decision Density	10
Design Complexity	11
Design Density	12
Essential Complexity	16
Essential Density	17
Global Data Complexity	20
Global Data Density	21
Maintenance Severity	30
Normalized Cyclomatic Complexity	34
Pathological Complexity	40
2-Error Metrics	
Error Rate	14
Error Density	15
3-Halstead Metrics	
Program Length	26
Program Volume	29
Program Level	27
Program Difficulty	23
Intelligent Content	22
Programming Effort	24
Error Estimate	25
Programming Time	28
4-Quantitative Metrics	
Branch Count	2
Call Pairs	3
Condition Count	6
Decision Count	9
Edge Count	13
Formal Parameter Count	19
Modified Condition Count	31
Multiple Condition Count	32
Node Count	33
Unique Operators	38
Unique Operands	37
Total Operators	36
Total Operands	35
5-LOC(Line of Code) Metrics	
Total Lines of Code	42
Executable Lines of Code	18
Comments	5
Blank	1
Code and Comment	4
Percent Comments	41
Number of Lines	39

Initially, the BackwardSelect Algorithm uses all dimensions to produce a clustering C. This case is taken as the base case to compare another clustering against, because there are no labels available. At each iteration of the first for loop the data

dimensionality is reduced by one. We leave each available dimension out and compute a clustering. We compare each clustering to the clustering that uses all dimensions, using the contingency table (see Eq. 1) between the clustering and clustering C. External indices are used for comparison. The maximum value index shows the dimension that has a minimum affect on the clustering. Because when it doesn't exist, clustering is closest to the case with all the features. So, this dimension is eliminated, and algorithm goes on until no dimensions are left in the dimension list.

C. *Clustering*

We used the CLUTO Clustering Toolkit [6] in order to produce the clusterings at each step. We developed a MATLAB code for our algorithm and used vcluster.exe of CLUTO with the following parameters:
-clmethod=rb -sim=corr -crfun=i2 -colprune=0.8
Please see the CLUTO manual for more information about these parameters.

Calculation of contingency table and external indices are explained in detail in the Jain and Dubes's book [4]. We will give a brief description here.

Suppose n objects, denoted $\{x_1, x_2, ..., x_n\}$ are to be clustered. Consider two separate clusterings, called U and V, of these n data points into the same number of clusters. According to page 173 of [4] contingency table can be expressed as:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad (1)$$

In this table, a is the number of objects that are in the same cluster in both clusterings, d is the number of objects that are in separate clusters in both clusterings, b is the number of data points that are in the same cluster according to clustering U and in different cluster according to clustering V, c is the number of data points that are in the same cluster according to clustering V and in different cluster according to clustering U. The total number of pairs of objects is:

$$M = a + b + c + d = \frac{n(n-1)}{2} \qquad (2)$$

External indices are used to compare two clusterings U and V. Let m_1 and m_2 denote the number of objects that are in the same cluster according to clustering U and V respectively and let a, b, c, d be defined as above. Then external indices are defined as below. Larger values (close to 1) of these indices show that two clusterings are close to each other.

Rand Index	Jaccard Index	Fowlkes and Mallow Index
$\frac{a+d}{\binom{n}{2}}$	$\frac{a}{a+b+c}$	$\frac{a}{\sqrt{m_1 m_2}}$

D. *Backward Selection Experimental Results*

Figures 1-3 show the three index values computed as we run the BackwardSelect Algorithm for number of clusters of 5, 10 and 30.

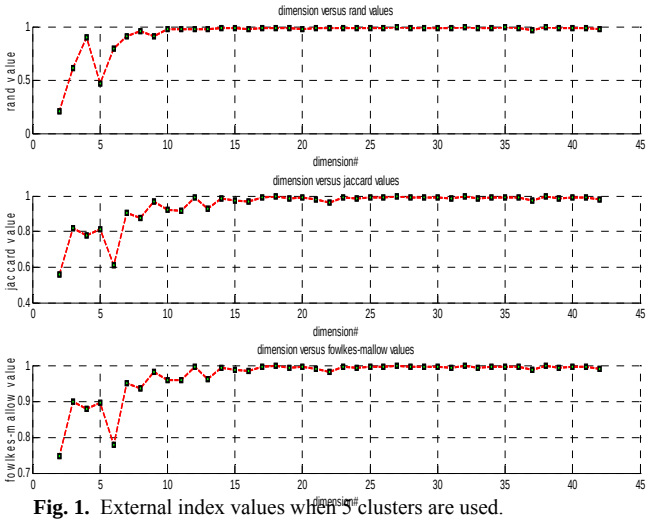

Fig. 1. External index values when 5 clusters are used.

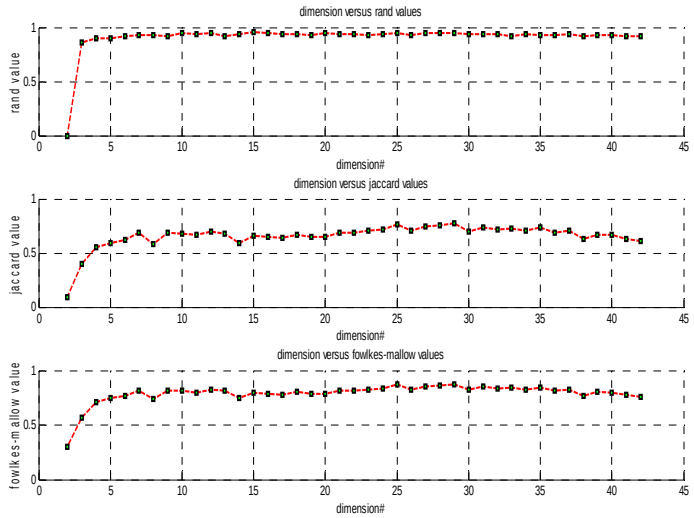

Fig. 2. External index values when 10 clusters are used.

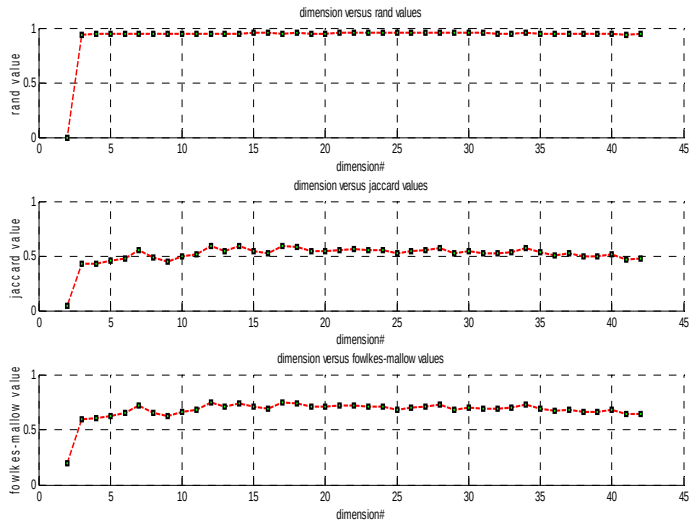

Fig.3. External index values when 30 clusters are used.

As the number of clusters increases indices can not differentiate between useful dimensions, because the probability that any two objects will fall in separate clusters in both clusterings (d) increases and dominates the index value. In our example, the data has five different domain metrics and five clusters seem to give better results. For all clusterings, there seems to be a certain number of features which gives a clustering close to the original feature set.

TABLE II
SELECTED DIMENSIONS FOR 5 CLUSTERS

Rand Index:

18	26	9	17	4	36	30	19	12	33	23
7	25	42	40	28	6	32	37	27	14	31
21	10	39	34	35	38	20	41	13	3	11
1	16	22	5	8	2	15	29	24		

Jaccard Index:

18	26	9	17	4	36	30	19	12	33	23
7	25	42	40	28	6	32	2	16	3	37
5	27	8	21	22	31	34	41	1	13	10
14	38	35	29	15	20	11	39	24		

TABLE III
SELECTED DIMENSIONS FOR 10 CLUSTERS

Rand Index:

2	17	23	10	18	31	9	13	34	21	5
14	32	4	8	11	25	16	6	26	38	28
20	40	3	7	42	15	33	1	39	37	36
35	22	27	19	30	41	12	29	24		

Jaccard Index:

2	17	23	10	36	40	25	14	11	6	37
13	22	26	21	20	30	33	27	38	42	32
3	5	7	1	31	18	12	35	4	16	39
34	8	41	19	28	9	15	29	24		

TABLE IV
SELECTED DIMENSIONS FOR 30 CLUSTERS

Rand Index:

13	34	31	30	4	39	40	16	32	17	2
7	23	3	14	21	9	8	19	36	12	22
1	11	6	35	38	10	24	42	20	26	27
25	37	41	18	5	15	33	29	28		

Jaccard Index:

13	34	31	30	4	39	40	16	32	17	3
27	5	42	10	9	8	11	7	18	25	6
23	26	37	28	1	33	38	35	41	12	14
36	21	19	2	20	22	15	29	24		

Tables II-IV show dimensions of increasing importance (the last eliminated dimensions) for each index. Since Fowlkes-Mallow and Jaccard indices gave the same results, only Jaccard is shown. Rand index gave different solutions. On the other hand, all of the indices result in a different importance order of dimensions for different cluster numbers. The most important 22 dimensions for all cases have been shaded, because as we will see below for 22 dimensions, PCA can explain 98 percent of the variance.

E. PCA(Principal Component Analysis)

We used PCA to select dimensions. We decided on which PCA projected dimension to keep based on the eigen values of the input covariance matrix. Figure 5 shows the proportion of explained variance for each no of dimensions.

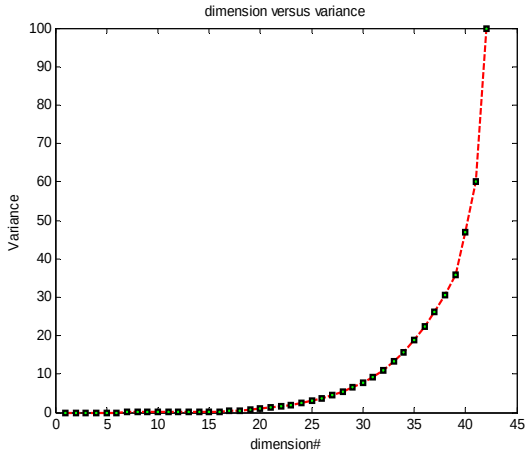

Fig. 4. Proportion of explained variance using PCA.

According to Figure 4, 22 dimensions are enough to explain 98% of the variance. Table V shows the eigen value for each dimension ordered by eigen values. Eight dimensions have zero eigen value, so they are unimportant for variance.

TABLE V
EIGEN VALUES

Dimension #	Eigen Value	Dimension #	Eigen Value
19	0.0001	39	0.2391
8	0.0022	7	0.3271
21	0.0040	1	0.3750
22	0.0055	37	0.4683
23	0.0116	9	0.4931
41	0.0136	33	0.5333
12	0.0234	4	0.6630
14	0.0341	35	0.8763
11	0.0448	15	0.9935
2	0.0587	29	1.1669
40	0.0725	28	1.4323
30	0.0960	18	1.4995
5	0.1174	10	1.7374
24	0.1278	34	2.0080
17	0.1618	42	4.3692
20	0.1824	13	5.1065
6	0.2070	36	15.5486

Table.VI shows the dimensions selected by PCA (they have been shaded).

TABLE VI
SELECTED DIMENSIONS BY PCA

25	26	27	32	16	38	31	3	19	8	21
22	23	41	12	14	11	2	40	30	5	24
17	20	6	39	7	1	37	9	33	4	35
15	29	28	18	10	34	42	13	36		

III. CONCLUSION

In this study, we used clustering and external indices for software metric selection. We also compared the metrics selected by our method to the metrics selected by PCA. TableVII shows the difference in selected dimensions and similarity ratio for both techniques used (backward selection algorithm and PCA). Moreover, the similarities of indices for each experiment (i.e. clustering) have been evaluated.

TABLE VII
DIFFERENCES IN SELECTED DIMENSIONS BY EXTERNAL INDICES AND PCA

Cluster #		Index	Same Dim#	Different Dim#	PCA Sim. (%)
5	Indices Sim.=%91	Jaccard	1, 5, 10, 13, 15, 20, 24, 29, 34, 35, 37, 39	3, 8, 11, 14, 21, 22, 27, 31, 38, 41	34,36
		Rand	1, 5, 10, 13, 15, 20, 24, 29, 34, 35, 39	2, 3, 8, 11, 14, 16, 21, 22, 31, 38, 41	33,16
10	Indices Sim.=%59	Jaccard	1, 4, 5, 7, 9, 15, 18, 24, 28, 29, 34, 35, 39, 42	3, 8, 12, 16, 19, 31, 32, 41	37,66
		Rand	1, 7, 15, 20, 24, 28, 29, 33, 35, 36, 37, 39, 42	3, 12, 19, 22, 27, 30, 38, 40, 41	68,31
30	Indices Sim.=%72	Jaccard	1, 6, 15, 20, 24, 28, 29, 33, 35, 36, 37	2, 12, 14, 19, 21, 22, 23, 25, 26, 38, 41	56,18
		Rand	1, 5, 6, 10, 15, 18, 20, 24, 28, 29, 33, 35, 37, 42	11, 12, 22, 25, 26, 27, 38, 41	36,12

According to Table VII, Rand Index gives better results for small number of clusters. However, when the number of clusters increases, Jaccard and Fowlkes-Mallow is better. But the most similar value (68.31%) is produced for Rand index using 10 clusters. In general, for 10 clusters we get the best agreement between PCA and the backward selection methods. For 10 clusters, the number of same selected dimensions is 13.5 and there is or 53% similarity on average.

On the other hand, PCA shows that weight of dimension 36 (total operator) has larger variance (39.87%). But in the external indices work this dimension barely selected. Dimension 24 (programming effort) is selected most valuable dimension nearly in all the external indices experiments, while PCA gives it less priority. These differences need to be worked on in more detail.

When external indices are compared with respect to each other, they generally select similar dimensions. Using the selected frequency in all experiments, Table.VIII categorizes the dimensions selected by Rand and Jaccard external indices. For four clusterings (5,10,15 and 30) and two indices (Rand and Jaccard), we have performed eight experiments. We divided the frequencies (the frequency for a dimension is the the number of times the dimension is selected among the most important 22 dimensions) into four groups (most selected, generally selected, less selected and not selected). This table shows the important dimensions found by the backward selection algorithm.

TABLE VIII
CATEGORIZATION OF SELECTED DIMENSION BY EXTERNAL INDICES

Degree	Dimensions	Matching PCA Dimensions
Most selected (six or more)	1, 3, 12, 15, 20, 22, 24, 28, 29, 35, 37, 41	1, 15, 20, 24, 28, 29, 35, 37
Generally selected (four and five)	5, 8, 10, 11, 14, 21, 31, 38, 39, 42	5,10,39,42
Less selected (2 to 4)	2, 4, 6, 7, 13, 16, 18, 19, 23, 25, 26, 27, 30, 32, 33, 34, 36, 40	4, 6, 7, 13, 18, 33, 34, 36
Not selected (none or 1)	9, 17	9, 17

When we consider the most and generally selected dimensions found by the backward selection algorithm, Table.IX shows the dimensions which are not selected by the PCA.

This work shows that using external indices or PCA one may get different set of dimensions as the most important software metrics. The dimensions produced by PCA are quite meaningful. However, external indices point to a few other metrics that must also be considered in the module complexity domain. We suggest adding dimensions 3 (call pairs), 14 (error rate) and 21 (global data density) into the 22 dimensional domain model produced by PCA. This new domain will have 25 metrics. These metrics happen to be cited as useful metrics for module complexity domain in the Munson's work also [3].

TABLE IX
DIMENSIONS NOT CONSIDERED IN PCA

Type	Dimension#	Name
Most	3	Call Pairs
Most	12	Design Density
Most	22	Intelligent Content
Most	41	Percent Comments
Generally	8	Cyclomatic Density
Generally	11	Design Complexity
Generally	14	Error Rate
Generally	21	Global Data Density
Generally	31	Modified Condition Count
Generally	38	Unique Operators

IV. FURTHER WORK

In this study, we used clustering together with external cluster similarity indices to compute the most important module coupling metrics. External indices point to new dimensions which may be important for software quality determination. Although we could not find a labeled data set, it would be very interesting to see how useful dimensions selected by PCA and clustering+external indices are. On the other hand, the unsupervised feature selection method we introduced could be used for data sets other than software metric data.

REFERENCES

[1] J. Munson,T. Khoshgoftaar, "The dimensionality of program complexity", In Proc. 11th Annual International Conference on Software Engineering,Pittsburg,Pennsylvania,pp. 245-253

[2] L. Briand, J. Wüst, H. Lounis, 1999, "Using coupling measurement for impact analysis in object-oriented systems", In IEEE International Conference on Software Maintenance,Oxford,England,pp.475

[3] Gregory A. Hall , Wenyau Tao, John C. Munson, "Measurement and Validation of Module Coupling Attributes", Software Quality Journal,13,Netherlands,2005,p.p. 281-296

[4] Anil K. Jain , Richard J. Dubes," Algorithms for Clustering Data", Prentice Hall,New Jersey, ISBN 0-13-022278-X,1988.

[5] Metrics Data Program, NASA IV&V Facility, http://mdp.ivv.nasa.gov/, revised December 16 , 2004.

[6] CLUTO,A Clustering Toolkit Manual Release 2.0, George Karypis University of Minnesota, http://www.cs.umn.edu/~karypis, , revised May 3, 2002.

[7] Yi Liu,Jeng-Foung Yao, Gita Williams,Gerald Atkins, "Studying Software Metrics Based on Real-World Software Systems",Consortium for Computing Sciences in Colleges,Mid-South Conference,2007,p.p. 55-61.

[8] Taghi M. Khoshgoftaar, Boca Raton,"The Necessity of Assuring Quality in Software Measurement Data",10th International Symposium on Software Metrics,2006.

[9] Tim Menzies, Justin S. Di Stefano, Mike Chapman and Ken McGill,,"Metrics That Matter",IEEE Software Engineering Workshop,2003