*ICIC Express Letters*
Volume **11**, Number **4**, April **2017**

ICIC International ©2017 ISSN 1881-803X
pp. 829–834

# IMPROVEMENT OF HIERARCHICAL CLUSTERING ALGORITHM AND ITS APPLICATION IN SOFTWARE RE-ARCHITECTING

Sheng Gao, Shiqun Yin*, Mengmeng Sun and Zhanhao Han

Faculty of Computer and Information Science
Southwest University
No. 2, Tiansheng Road, Beibei Dist., Chongqing 400715, P. R. China
gslypswu@163.com; *Corresponding author: qqqq-qiong@163.com

Received September 2016; accepted December 2016

Abstract. *In the process of software evolution, software architecture gradually becomes perplexing, which seriously affects the maintenance work in the later period. Although the traditional software re-architecting method is based on model driven that makes software architecture achieve a nested form, the software architecture of hierarchy is not clear. To solve these problems, a hierarchical clustering algorithm based on information loss (HCBIL) is proposed in this paper. Because the information loss is used as the criterion of similarity measure, the low information loss of HCBIL algorithm makes the high quality of clustering in the later stage of clustering. And we refine the character weights, so as to avoid the strong coupling phenomenon between clusters in the later stage of clustering. In this paper, we apply the HCBIL algorithm in software re-architecting, and further put forward the model of software re-architecting. Through the model, software architecture with a clear nesting level is implemented. By the contrast experiment, it is proved the clustering quality of HCBIL algorithm is higher than that of traditional hierarchical clustering algorithms, whose similarity computation is based on distance calculation.*

**Keywords:** Software re-architecting, Information loss, Hierarchical clustering, Mutual information

## 1. Introduction.

In the middle of the 1960s, with the appearance of large capacity, high speed computer and advanced language, the amount of software development increased rapidly, which led to the outbreak of the software crisis. Subsequently expose some of the problems, for example, the progress difficult to predict, the cost difficult to control, and the product difficult to maintain, which are closely concerned by the software developer. In the legacy system, the messiness and bloat of the software architecture have a serious impact on the late maintenance work. In order to overcome the problem, and make existing software architecture clearer and more readable, that try to seek for an automatic and effective method to achieve the re-construction, is particularly important.

With the development of data mining technology in the end of 1980s, the clustering algorithm in data mining has become an important technology in software re-architecting. Andritsos and Tzerpos [1] proposed an algorithm for comprehension-driven clustering (ACDC) based on hierarchical clustering method, realizing the software re-architecting, and basically restoring the structure of nested hierarchical structure. However, ACDC does not use the method of hierarchical clustering completely, which makes the structure of the legacy system a simple segmentation, when the software re-architects. Maqbool and Babri [2] proposed weight combined algorithm (WCA), which adopts the method of weight assignment to the entity's characters. However, in the iterative process, the weak competitive entities are discarded, which leads to the low coupling property of clusters. So the clustering quality is unsatisfactory. The architecture of the legacy system is simple to partition and clusters have low coupling property, which become problems in software re-architecting. Considering the above problems, we propose an improved hierarchical clustering algorithm which was named hierarchical clustering algorithm based on information loss (HCBIL). Through applying this algorithm to software re-architecting, we get a clear hierarchy view, which shows that the clustering quality is higher.

## 2. Improved Hierarchical Clustering Algorithm.

Hierarchical clustering algorithm based on information loss (HCBIL) improves the hierarchical clustering algorithm in two aspects. (1) In construction of character vectors, we divide character weights into global weight and local weight. (2) HCBIL algorithm uses information loss as the standard of similarity measure. We will introduce the specific content from the following sections.

### 2.1. Construction of character vectors.

Cluster entities and characters constitute the two-dimensional matrix, where each entity $E_i$ corresponds to different character vectors $C_i$. $C_i$ is defined as follows:

$$C_i = (C_{i1}, C_{i2}, \ldots, C_{ij})$$

In the definition above, $j$ is the category of entity characters.

In HCBIL algorithm, the character weights are divided into global weight $W_j$ and local weight $w_{ij}$. The global weight represents the degree of the impact of character on all the entities, and the local weight indicates the degree of influence on the single entity. The character vector $C_i$ of the entity $C_i$ which is processed by the weight is expressed as follows:

$$C_i = \left(W_i^1 * C_{i1}, W_i^2 * C_{i2}, \ldots, W_i^j * C_{ij}\right);$$
$$W_j = \lambda, \quad \lambda \in (0,1);$$
$$w_{ij} = \frac{N_{ij}}{\sum_{j=1}^{k} N_{ij}};$$
$$W_i^j = W_j * w_{ij} = \frac{\lambda N_{ij}}{\sum_{j=1}^{k} N_{ij}} \tag{1}$$

In Formula (1), the value of $\lambda$ is given by experience. $N_{ij}$ represents the number of the $j$ characters of the entity $E_i$. $k$ is total number of the category of entity characters.

### 2.2. Similarity computation.

The similarity computation of HCBIL algorithm proposed in this article is based on the information bottleneck theory. As the base of information bottleneck, mutual information describes the influence degree of an event on the appearance of another event. The formula is as follows:

$$I(X;Y) = \sum\nolimits_{m\in[0,1]} \sum\nolimits_{n\in[0,1]} P(X=m, Y=n) * \log \frac{P(X=m, Y=n)}{P(X=m) * P(Y=n)} \tag{2}$$

Among them, $P(X=m)$ stands for the occurring probability of $X$. $P(X=m, Y=n)$ stands for the jointly occurring probability of $X$ and $Y$.

According to Formula (2), the information loss formula is deduced as follows:

$$\begin{aligned}\sigma(E_i, E_j) = {} & \sum\nolimits_{m\in[1,k]} P(E_i) * D_{KL}\left(P(C_m|E_i) \| P\left(C_m|\tilde{E}\right)\right) \\ & + \sum\nolimits_{m\in[1,k]} P(E_i) * D_{KL}\left(P(C_m|E_j) \| P\left(C_m|\tilde{E}\right)\right)\end{aligned}$$

$$D_{KL}\left(P(C_m|E_j) \| P\left(C_m|\tilde{E}\right)\right) = \sum\nolimits_{m\in[1,k]} P(C_m|E_i) * \log \frac{P(C_m|E_i)}{P\left(C_m|\tilde{E}\right)} \tag{3}$$

In Formula (3), $P(C_m|E_i)$ stands for the probability of the $m$ characteristic of entity $E_i$. $\tilde{E}$ stands for the merged entity cluster. The smaller the loss of information is, the greater the degree of similarity between entities is. A hierarchical clustering algorithm based on information loss will merge the two entities with the smallest information loss.

2.3. **HCBIL algorithm's pseudo codes.** The steps of hierarchical clustering algorithm based on information loss are as follows. (1) Data processing: The data obtained from the legacy system is normalized and weighted. (2) Entity merging: Calculating the information loss among the entities, then select two entities with the smallest information loss to cluster, forming the cluster. (3) Updating the character vectors: The calculation formula of the character vectors of entity clusters $\tilde{C}$ is $\tilde{C} = \alpha * C_i + \beta C_j$. Among them, $C_i$ stands for the character vectors of entity $E_i$. $C_j$ stands for the character vectors of entity $E_j$. (4) Iterative merging: Through re-computing the similarity between entity clusters, merge the entity clusters with the largest similarity, until the termination conditions are satisfied. (5) Cluster termination: Calculate the number of layers and the number of clusters. If they reach the given threshold, the clustering is terminated. The pseudo codes of HCBIL algorithm are as follows.

```
The pseudo codes of HCBIL algorithm
Input   The two-dimensional matrix data of entity-character
Data normalization
// Construction of entity character vectors
for Number of entities $i = 1, 2, \ldots, n$
      Character weight $W_i^j = \frac{\lambda N_{ij}}{\sum_{j=1}^{6} N_{ij}}$   // $j$ represents the characters category
      The character vector $C_i = \left(W_i^1 * C_{i1}, W_i^2 * C_{i2}, \ldots, W_i^j * C_{ij}\right)$
end
// Similarity computation and clustering
if Number of entities or clusters $n > n_0$&&Layer number of clustering $N < N_0$
      repeat
            for Number of entities $i, j = 1, 2, \ldots, n$
                  $P(E_i) = P(E_j) = \frac{1}{n}$
                  According to Formula (3), the information loss between the two entities
is calculated of $\sigma(E_i, E_j)$
                  Selecting the smallest $\sigma(E_{i_0}, E_{j_0}) = \mathrm{Min}\{\sigma(E_i, E_j)\}$
                  Merging entities $E_{i_0}$ and $E_{j_0}$, thereby forming entity clusters $\tilde{E}$
                  Number of entities or clusters $n--$
                  Layer number of clustering $N--$
            end
      Updating character vectors of clusters $\tilde{C} = \alpha * C_i + \beta C_j$
      Until $n \le n_0$&&$N \ge N_0$
End
```

3. **Software Re-architecting Model Using HCBIL Algorithm.** HCBIL algorithm is applied in the software re-architecting, which makes the structure of the bloated legacy system back into a clear structure. The steps are as follows. (1) Select legacy system. We select the software system with long evolution time. (2) Generate UML class diagram and XML documents. The legacy system source codes input the reverse engineering tool to generate UML class diagram and XML documents. (3) Data preprocessing. The two-dimensional matrix data of entity-character are extracted from the XML documents to deal with noise. (4) Generate the tree structure of clusters. The tree structure of clusters is finally formed after using HCBIL algorithm for clustering. Software re-architecting model is shown in Figure 1.

According to the software system which is programmed in different languages, the selection of cluster entities is also different. We choose the legacy system, which is programmed by object-oriented language as the object of refactoring. Class is the basis of

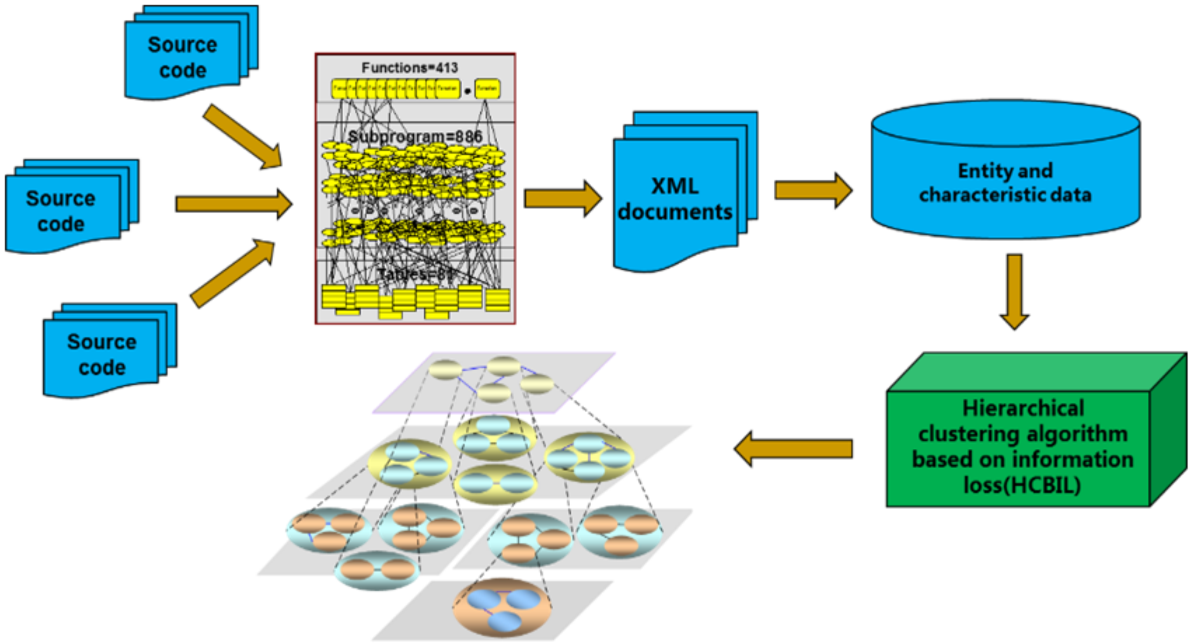
Source code
Source code
Source code
Functions=413
Subprogram=886
Tables=81
XML documents
Entity and characteristic data
Hierarchical clustering algorithm based on information loss(HCBIL)

FIGURE 1. The software re-architecting model

object-oriented language, we take the class as the cluster entity. In order to improve the quality of the cluster, it is necessary to select the appropriate information as entity characters. On the one hand, the data is referenced by class; on the other hand, it is the relationship between the two classes. Based on the above introduction, we select the following six aspects as the entity characters: (1) the data referenced by class; (2) global variables referenced by class; (3) local variables referenced by class; (4) association relationship between classes; (5) generalization relationship between classes; (6) dependency relationship between classes.

4. **Experimental Results and Analysis.** We do the experiment to prove that the clustering quality of HCBIL algorithm proposed in this article is higher than the traditional algorithms, such as CURE algorithm and SBAC algorithm. Among these algorithms, the similarity calculation of CURE algorithm and SBAC algorithm is based on the distance. Both CURE algorithm and SBAC algorithm take account of the impact of the character on entity. In the experiment, we choose *Open laboratory management system* programed by Java as the legacy system. Through reverse engineering tool Enterprise Architect, the source codes of legacy system are generated of UML class diagrams and XML documents.

In the HCBIL algorithm and SBAC algorithm applied in the software re-architecting experiment, the global weight distribution is according to 5:5:5:3:1:1. Because the HCBIL algorithm and the SBAC algorithm are different in the similarity computation in the experiment, the same global weight distribution can be given to show whether the similarity calculation based on information loss is superior to other similarity computation. In the CURE algorithm applied in the software re-architecting experiment, the global weight distribution is according to 1:1:1:1:1:1. Because the similarity computation of the CURE algorithm and the SBAC algorithm is based on the distance method, setting different global weight distribution in SBAC algorithm and CURE algorithm can reflect the importance of the global weights in the experimental results. Further reflect whether it is necessary to set the global weight in HCBIL algorithm. The results of the experiment are shown in Figure 2.

For CURE algorithm in clustering, because the set of global weights is equal, in the early stage of clustering, the number of clusters is less, and the coupling of clusters is

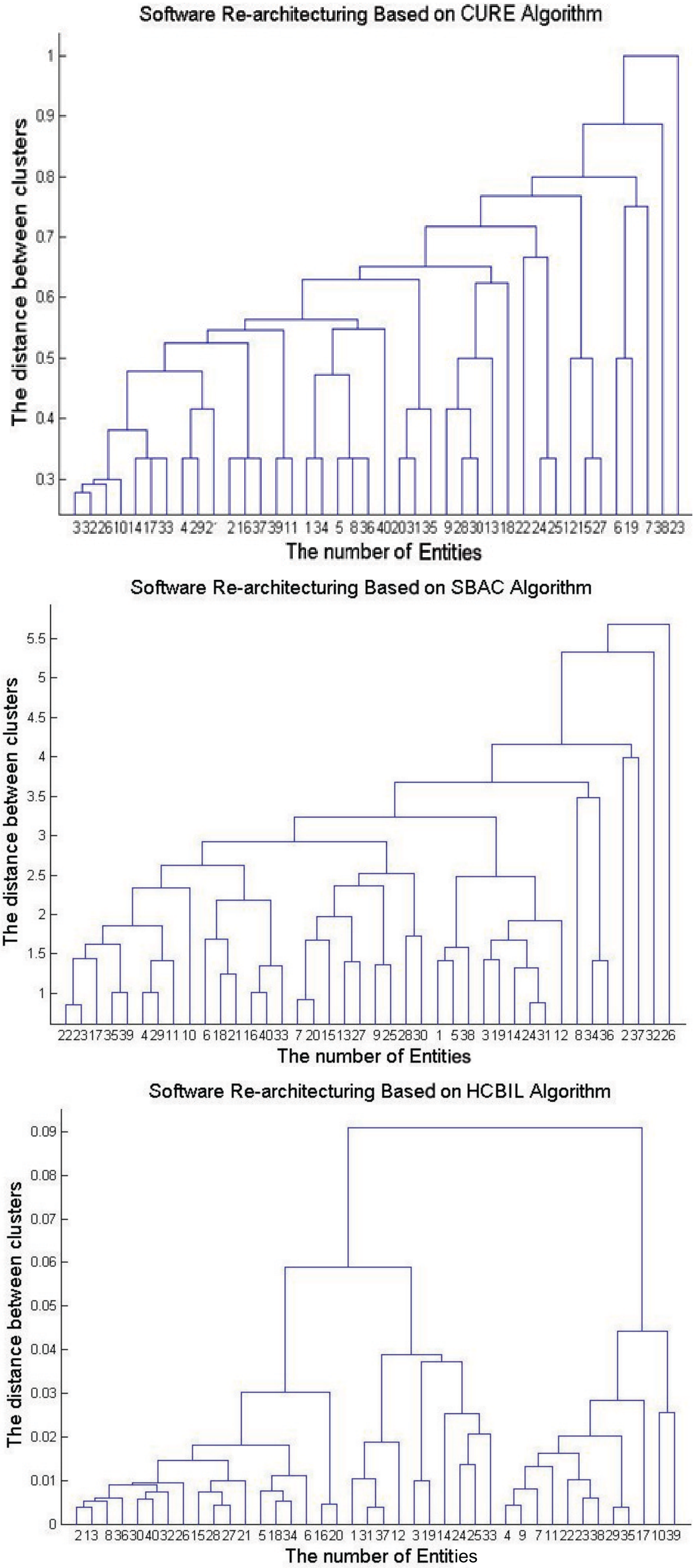

Figure 2. Comparison of experimental results of different algorithms

relatively strong. SBAC algorithm takes account of the influence of global weight, so the number of clusters is more in the prophase of clustering. However, the distance between clusters is larger, and the similarity between clusters is low. So the quality of clustering is not high when clustering again. CURE algorithm and SBAC algorithm similarity computation is based on the distance method, therefore, their clustering results show a ladder distribution, and the quality of clustering is not high. Because the HCBIL algorithm takes account of the influence of the global weight and the local weight, it not only avoids the strong coupling between the clusters, but also avoids the ladder distribution. The similarity of HCBIL algorithm is based on information loss, so the low loss of information between clusters makes high quality of clustering in the later stage of clustering.

5. **Conclusion.** By applying the clustering idea to the software re-architecting, we propose HCBIL algorithm. In the algorithm, the information loss is used as the criterion of similarity measure. Compared with the traditional similarity measuring method based on distance, the clustering of outliers sensitive and strong coupling between the clusters are overcome. Fully considering the influence degree of characteristics to the entity, the algorithm divides the character weight into global weight and local weight. Experiments show that the HCBIL algorithm is applied in the software re-architecting not only implements the hierarchical nesting of the software architecture, but also achieves a high quality of clustering. In the traditional hierarchical clustering algorithms and HCBIL algorithm, the clustering between entities or clusters is "either this or that". So the entity after clustering cannot be used as an independent individual to participate in the new clustering. In further study, we will introduce the fuzzy set theory on the basis of HCBIL, so as to solve the "either this or that" problem.

**Acknowledgment.** This work is supported by the Science & Technology project (20130 01287, 41008114, and 41011215).

## REFERENCES

[1] P. Andritsos and V. Tzerpos, Information-theoretic software clustering, *IEEE Trans. Software Engineering*, vol.31, no.2, pp.150-165, 2005.

[2] O. Maqbool and H. A. Babri, The weighted combined algorithm: A linkage algorithm for software clustering, *Proc. of the 8th European Conf. Software Maintenance and Reeng.*, pp.15-24, 2004.

[3] W. F. Opdyke, *Refactoring: A Program Restructuring Aid in Designing Object-Oriented Application Frameworks*, Urbana-Champaign: Univ. of Illionis, 1992.

[4] M. Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.

[5] M. Pinzger and G. Antoniol, Guest editorial: Reverse engineering, *Empirical Software Engineering*, vol.18, no.5, pp.857-858, 2013.

[6] Durak. Umut, Pragmatic model transformations for refactoring in Scilab/Xcos, *International Journal of Modeling Simulation and Scientific Computing*, vol.7, no.1, 2016.

[7] W. G. Griswold and W. F. Opdyke, The birth of refactoring a retrospective on the nature of high-impact software engineering research, *IEEE Software*, vol.32, no.6, pp.30-38, 2015.

[8] H. Brunelière, J. Cabot, G. Dupé and F. Madiot, MoDisco: A model driven reverse engineering framework, *Information and Software Technology*, vol.56, no.8, 2014.

[9] A. Corazza, S. D. Martino, V. Maggio and G. Scanniello, Weighing lexical information for software clustering in the context of architecture recovery, *Empirical Software Engineering*, vol.21, no.1, pp.72-103, 2016.

[10] C. Liu, H. Fan, W. Zhang and D. Xiao, Software reconstruction technology research and application, *Information Technology*, vol.10, pp.4-6, 2012.