Grouping of technological features using a clustering algorithm

Gruppierung von technologischen Merkmalen unter Verwendung eines Clustering-Algorithmus

Galina Nikolcheva, Julieta Kaleicheva

Faculty of Industrial Technology, Technical University of Sofia, Sofia, Bulgaria, e-mail: ginic@tu-sofia.bg, jkaleich@tu-sofia.bg

Abstract — The work presents a clustering algorithm that groups technological features for the generation of setups. The features processed in each setup have a similar technological process. The clustering algorithm - COBWEB is used for their grouping. It is an incremental conceptual clustering algorithm that groups clusters into interacting features at each setup. A variant method is used to create the technological process.

***Zusammenfassung* —** Die Arbeit präsentiert einen Clustering-Algorithmus, der technologische Features für die Generierung von Setups gruppiert. Die in jedem Setup verarbeiteten Funktionen haben einen ähnlichen technologischen Prozess. Der Clustering- Algorithmus - COBWEB wird für ihre Gruppierung verwendet. Es ist ein inkrementeller konzeptioneller Clustering-Algorithmus, der Cluster in interagierende Features bei jedem Setup gruppiert. Eine Variantenmethode wird verwendet, um den technologischen Prozess zu schaffen.

I. Introduction

In the past decades manufacturing industries are undergoing a transition from traditional methods to advanced manufacturing technologies, a many of which are computer based. The manufacturing research community has focused on developing and improving technologies such as CAD/CAM and Computer- aided process planning (CAPP). The features are the link between design and manufacturing in a CIMS environment. In recent years have seen many works on automated process planning and connected with that determining the number of setups, the sequence of making features in each setup [8,11, 14,17].

The sequence of making the features depends on feature interacting such as the geometric relationships between them are the reason for theirs interacting. The feature types are also used to identify the type of interaction and thus the applicable rules that govern their removal. Technological features are two types: non-intersect (simple) and intersect (complicated) that are interacted to one another.

The goal of the paper is group the interacting technological features based on their geometrical relationships by using clustering algorithm so that the groups obtained to have similar process plan.

II. Grouping the Features

The geometrical relationships between features influences the process planning. They are studied and defined by different approaches: fuzzy, neural network and expert systems [14, 15,16]. Most researchers have used the idea of feature interactions to express the problems that occur in the planning process when some elements are arranged in a given geometric orientation relative to another [3].The feature interactions express Hayes [7] looks for feature interactions when generating the fixturing, and the feature sequencing portions of the process plan. It uses special rules aimed at particular interactions, and avoids those interactions by reordering sequence of features, or putting features into different setups. The geometric relationships between features are used [10] to identify intersected features and the sequences in making them. To determine the sequence of making the features it is necessary to identify intersected features and to organize them into regular groups of patterns. These groups of features have a similar process plan. Chang [2] grouped feature clusters based on tool approach direction. He defined approach and feed directions used for setup generation. An approach direction is a straight path that gives a tool an unobstructed access to the feature in the workpiece. The features may have more than one approach direction. Based on approach direction of the tools, similar features are grouped for a setup. Different methods are used for grouping of various objects and the results are best by using clustering algorithms [2,3].

A. Clustering Algorithms to Grouping of Features

Traditional clustering algorithms adopt one of two primary approaches: agglomerative and divisive. Agglomerative methods repeatedly group objects and clusters of objects together, based on a changing similarity requirement, to form larger and larger clusters. To achieve this a distance metric which allows comparisons between objects and clusters is needed. Divisive methods do the reverse, subdividing large clusters into smaller and smaller ones. The results of both methods can be viewed as a hierarchical tree.

Most methods produce an exclusive partitioning, simultaneously taking a set of instances and placing those instances into disjoint clusters (solving the clustering problem). A separate task is the formulation of a description for each cluster (the categorization problem) [1]. This is the assignment of category labels to the clusters. A large body of older research solves the clustering problem using statistical methods, but leaves categorization unaddressed.

Concept of nodes representing descriptions of concepts and instances is constructed in this approach. The learning system typically adds one instance at a time, following a path deeper down the hierarchy in a decision tree manner until the new instance has been classified under an existing concept or a new clustering methods provide a natural approach to solving both clustering and categorization problems. These methods include the formation of a concept hierarchy using an incremental top down classification scheme and thus perform unsupervised learning. A knowledge based structure consisting concept cluster has been formed. This approach is divisive as instances classified under previously formed concepts are further segregated when sub-generalizations form. These sub-generalizations may include instances previously classified under a more general concept.

In this work we have considered incremental conceptual clustering algorithms such as UNIMEM [9], ERAM [4], COBWEB[5], STAGGER[12], ID4 [13].

B. APPLICATIONS TO FEATURE GROUPING

The COBWEB system is an incremental conceptual clustering algorithm. The program is written by Raymond Joseph Mooney in 1991. The system forms classification trees that are intended to yield "good" prediction among many attributes and can be used for a wide variety of purposes.

An important difference between COBWEB and earlier conceptual clustering systems is that it is incremental – COBWEB integrates an observation into an existing classification tree by classifying the observation along a path of "best" matching nodes. Like ID4, probabilistic summaries of previous observations are stored at each node, but the matching functions and the criteria used for subtree revision differ considerably. COBWEB uses the category utility function [6] to guide classification and tree formation.

Category utility bases its evaluation on all of the observation's attribute – values rather than a single one, making COBWEB a polythetic classifier as opposed to a monothetic classifier e.g. ID4. Subtree revisions in COBWEB are triggered by considering prediction ability over all attributes, but concern for multiple attributes complicates subtree revision. In ID4 a subtree is simply deleted, but in COBWEB a deletion that benefits one attribute may be inappropriate for others. In response, the system identifies points in the tree for cost-effective prediction of individual attributes. These points are marked by default values that COBWEB dynamically maintains during incremental clustering.

C. COBWEB

We use COBWEB – to group interacting features of each setup in clusters, so that they all have a similar technological process achieved by variant machining process.

The following lists of structures of data for the use of COBWEB must be create:

- A list of attributes-names;
- A list of domains – value corresponding to attributes;
- A list of raw – examples, appropriate with upper structures.

The attributes-names are included in the first structure (Figure 1). These names are obtained in different way. The type of the model (model) and the type of the base (type_of_the_base_solid) are known. The type of interacting (nested and tangent), the geometry (type_of_the_second_feature) and the identical geometry form of the features (type) are obtained by analysis. The approach directions were described in the previous sections. Two additional attributes have been introduced showing the passing (type_of_the_interaction_1) in a direction perpendicular to the plane, where the approach direction lies and the passing (type_of_the_interaction_2) in a direction parallel to or coincidental with the plane where the approach direction lies.

Some values of attributes in second list are obtained from the feature classification. They are:

- For the attribute model – protrusion or depression;

```
(setf *feature-names* '(
      model
      nested
     tagent
     type_of_the_second_feature
     approach_direction
     type_of_the_base_solid
     type_of_the_interaction_1
     type_of_the_interaction_2
     type ))
```

Fig. 1. First structure: attributes-names

```
(setf *domains*
  '(  (protrusion depression )
  (no_nested  nested_simple  nested_complex)
  (no_tangent tangent_one_side
     tangent_two_sides tangent_three_sides)
  (cylindrical_hole prism_hole conical_hole)
  (on_one_side on_different_sides )
  (cylindric prismatic sphere )
  (through_horizontal blind_horizontal)
  (through_vertical   blind_vertical)
  (identical          different))
```

Fig. 2. Second structure: domains

- For the attribute base – geometrical type of the base – cylindric, prismatic and sphere;
- For the attribute geometrical type of feature - cylindrical_hole (cylindrical hole), prism_hole (pocket), conical_hole (hole with a cone-shaped bottom).
- The values of the remaining attributes such as nested, tangent,type_of_the_interaction_1,type_of_the_interaction_2 and type - Figure 2 are obvious.

Graphic figures of the interacting features have been created for carrying out an experiment - Figure 3. These are the most commonly used features in machining the details. Feature interactions are used to reason about geometric relations between features [3]. The interacting features are grouped in clusters after implementation of COBWEB. Each cluster contains features with similar technological process. The procedure starts by function train, as its argument is the third list – Figure 4 (train raw-examples).

The third list is a result of the first two, as its structure includes the feature names and their values.

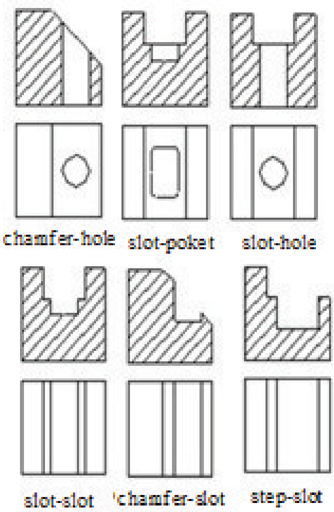

Fig. 3. Graphic figures of the interacting features

```
        setf *raw-examples* '((chamfer_hole  (depression   nested_simple  no_tangent
cylindrical_hole on_one_side  prismatic  blind_horizontal    through_vertical different   ))
                             (slot_pocket (depression  nested_simple     no_tangent
prismatic_hole on_one_side prismatic blind_horizontal   blind_vertical identical    ))
                          (slot_hole  (depression   nested_simple     no_tangent
  cylindrical_hole on_one_side prismatic  blind_horizontal    through_vertical different  ))
                          (slot_slot  (depression   nested_simple     no_tangent
prism_hole on_one_side prismatic through_horisontal  blind_vertical identical   ))
                          (chamfer_slot (depression  nested_simple  no_tangent  prism_hole
on_one_side prismatic through_horisontal  blind_vertical identical   ))

                            (step_slot (depression nested_simple no_tangent prism_hole
           on_one_side prismatic through_horisontal  blind_vertical identical    ))
```

Fig. 4. Train raw-examples

The result of COBWEB is two clusters, containing features with a similar technological process – the first cluster is

- C-37{STEP_SLOT CHAMFER_SLOT SLOT_SLOT SLOT_POCKET}

and the second one is

- C-38{SLOT_HOLE CHAMFER_HOLE}.

III. CONCLUSIONS

The utilized COBWEB system gives very good results for grouping of details that contain interacting features with a similar technological process. An increase in the number of instances leads to better precision in cluster forming. It may serve The classification and hierarchy structures may serve to generate setups based on machining operations.

REFERENCES

[1] J.R. Anderson, M. Matessa, "Exploration of an incremental Bayesian algorithm for categorization", *Machine Learning*; (9):1992, pp.275-308.

[2] T.C. Chang, *Expert Process Planning for Manufacturing, Addison-Welsey*, Reading, MA,1990.

[3] C.L.P. Chen, "Setup Generation and Feature Sequencing using an Unsupervised learning Algorithm", Proceedings 199, *NSF Design and Manufacturing Systems* 1993; 1,pp. 981-986.

[4] E. Feigenbaum, H. Simon,"EPAM-like models of recognition and learning", *Cognitive Science* 1984; 8, 305-336.

[5] D.H. Fisher, "Knowledge acquisition via incremental conceptual clustering", *Machine Lerning*, 1987; (2): 139- 172

[6] M.A Gluck, J.E. Corter, "Information, uncertainty, and the utility of categories*". Proceeding of the Seventh Annual Conference of Cognitive Science Society Irvine CA:Lawrence Elbaum* 1985; pp.283-287

[7] C.Hayes, "*Machineng planning: A Model of an Expert Level Planning Process*", Ph.D. Dissertation, Carnegie Mellon University, 1990.

[8] M.Hazarika, U. Dixit, "*Setup Planning for Machining*", Springer Chain Heidelberg, 2015

[9] M.Lebowitz, "Experiments with incremental concept formation: UNIMEM", *Machine Lerning* 1987; (2), pp.103-1138.

[10] S.K.Ong, A.Y.C. Nee, "Application of Fuzzy set Theory to setup planning", *Annals of the CIRP*, 1994, 43,1, pp.137-143.

[11] Y.H. Pao, K. Komeyli, D. Shei, S. LeClair, A.Winn, "The Episodal Associative Memory: Managing Manufacturing Information on the Basis of Similarity and Associativity", *Journal of Intelligent Manifacturing*,1993.

[12] J.C.Schlimmer, "*Concept acquisition through representational adjustment*", Unpublished doctoral dissertation. Irvine: University of California, Department of Information and Computer Science, 1987.

[13] J.C. Schlimmer, D. Fisher, "A case study of incremental adjustment of representations of learning". *Proceeding of the Fifth National Conference on Artificial Intelligence* 1986; Philadelphia, PA, 496-501.

[14] T.E. Westhoven, C.L.P.Chen,Y.H. Pao, S.R. LeClair, "The Episodal Associative Memory Approach for Sequencing Interactive Features in Process Planning", *AI for Eng., Design, Analy. And manufacturing*, 1992, 6, 3, pp.177-197.

[15] W. Wang, B.Subagdja, Ah-Hwee Tan, "A Self- Organizing Approach to Episodic Memory Modeling", 2010 IEEE *World Congress on Computational Intelligence* July, pp.18-23, CCIB, Barcelona, Spain, 2010.

[16] N. Xu, S.Huang, Y.Rong, "Automatic setup planning: current state-of-the-art and future perspective". *Int J Manuf Technol Manag* 2007,11(2), pp.193–208.

[17] Zhang G. Z, J. Han, Y. Y. Liu, "Approach to Machining Setup Process Planning of NC Machine Tools Based on Graph Theory", *Advanced Materials Research*, 2012,Vols. 542-543, pp. 1190-1195.