Expert Systems With Applications 42 (2015) 9412–9425

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Two-layer random forests model for case reuse in case-based reasoning

Shisheng Zhong, Xiaolong Xie, Lin Lin*

School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Keywords:
Two-layer model
Random forests
Case-based reasoning
Case reuse
Ensemble learning

ABSTRACT

Case reuse is important for case-based reasoning (CBR) because without it, a CBR system degrades to a case retrieval system, and a retrieved case generally cannot solve a problem directly. To improve the accuracy of case reuse, a two-layer random forests model is proposed and the framework of the corresponding CBR system is presented. First, clustering analysis is used to organize the cases in the case base, and gravitational self-organizing mapping algorithm is adopted to automatically detect the cluster number on the basis of the structure of the data. Then, a two-layer model scheme is proposed to model the mapping in every cluster. In this scheme, the first layer model obtains the pre-estimate of the output feature value of the query case, and the second layer model models the error of the pre-estimate, which is used to decrease the error generated by the first layer model. Random forests algorithm, which is a popular ensemble learning model, is adopted as a model in the two layers to improve accuracy and stability. Several benchmark datasets are used to validate the proposed two-layer model scheme, and the results demonstrate that it can improve the case reuse accuracy and stability. The proposed two-layer random forests model is applied to hydraulic generator design, and the results confirm that the proposed model is effective for case reuse.

1. Introduction

Case-based reasoning (CBR) is an approach that solves a new problem by using previous cases and experiences similar to the new problem (Kolodner, 2014). The approach assumes that similar problems should have similar solutions; thus, CBR is useful and effective if similar problems often take place. For the complex product design problem, a complex product contains several components and one component is composed of several parts; hence, the design process has massive parameters. Subsequently, plenty of domain knowledge is required if the product is designed in a traditional way, which designs the complex product step by step based on the design manual. Once the design requirements of the customer change, the engineer needs to change the original design alternative and massive repeated works need to be done. Therefore, the traditional design process is time-consuming, and the repeated process takes the majority of time and effort from the designers, which leaves little time for engineers to do creative tasks (Guo, Wen, Shao, & Wang, 2015). If there are enough historical design alternatives, which can be seen as historical cases, CBR can be used to design a new complex product by retrieving and reusing the historical design alternatives, which can avoid repeated designs, save time and guarantee that the design is reasonable.

CBR has several advantages: (i) its computational cost is relatively small (An, Cercone, & Chan, 1997); (ii) it does not rely on statistical assumptions and its justifications are easy to understand (Arshadi & Jurisica, 2005); (iii) it does not need an explicit domain model (Watson & Marir, 1994), making it generic and applicable to a wide range of domains. CBR has been widely used in many fields, such as assembly process design (Chen et al., 2006), injection mold design (Guo, Hu, & Peng, 2012), power transformer design (Hu, Qi, & Peng, 2015), electromotor design (Zhu, Hu, Qi, Ma, & Peng, 2015), fixture design (Wang, Rong, Li, & Shaun, 2010), ship structure design (Yang, Chen, Ma, & Wang, 2012), and so on.

CBR cycle contains four phases: (i) *retrieve* the most similar case or cases; (ii) *reuse* the information and knowledge in similar case or cases to obtain a suggested solution; (iii) *revise* the suggested solution based on its evaluation result; (iv) *retain* this new experience, which will be useful for future problem solving (Aamodt & Plaza, 1994). Many studies on CBR focus on the retrieval phase. (Zhu et al., 2015) used clustering analysis technique to organize cases and used feature selection technique to select important features for case retrieval. (Liu & Chen, 2012) used the Z index method to construct the case base, which can improve efficiency of case retrieval. (Li, Xie, & Goh, 2009) used mutual information technique to select important features. The authors (Xie, Lin, & Zhong, 2013) used edit distance-based measurement to handle the missing values and unmatched features in case

* Corresponding author. Tel.: +86-451-86413847.
E-mail addresses: zhongss@hit.edu.cn (S. Zhong), xiaolong.xie88@gmail.com (X. Xie), waiwaiyl@sina.com, axiexiaolong@163.com (L. Lin).

http://dx.doi.org/10.1016/j.eswa.2015.08.005
0957-4174/

retrieval. (Doğan, Arditi, & Murat Günaydin, 2008) used decision trees to determine feature weights.

Aside from the retrieval phase, the case reuse phase is even more important. By retrieving similar cases, we can only see how to solve the problems under similar conditions, but these similar cases are not exactly the same with the new problem, so generally, we cannot obtain the solution for this new problem in the case retrieval phase. In the case reuse phase, a suggested solution for the new problem can be obtained by adapting the solutions of similar cases. Thus, case reuse is more important, and without this phase, CBR systems act primarily as retrieval systems. Therefore, this paper focuses on the case reuse model. Case reuse can be accomplished in two ways: manual or automatic. Manual reuse approach is executed by experts based on their experience; hence, this approach depends on subjective decisions, lacks reliability, and takes too much time (Jin, Cho, Hyun, & Son, 2012). For automatic case reuse, some researchers focused on knowledge acquisition, e.g., in Lee (2003), associative rules were extracted using data mining techniques, and in Vong and Wong (2010), case-based adaptation scheme was used to adapt cases. Some researchers used statistical approaches to adapt cases, such as *k*-nearest neighbor approach, multiple regression analysis (MRA) based approach (Jin et al., 2012), artificial neural network (ANN) based approach (Lotfy & Mohamed, 2002), genetic algorithm based approach (Mat Jani & Lee, 2008), and grey relational analysis (Hu et al., 2015).

However, these approaches used traditional machine learning algorithms in case reuse, i.e., these approaches just built one global model from the data and then used this model to estimate the solutions. However, the ability of one global machine learning model is generally limited, and sometimes, it may not perform well when dealing with complicated data because the data may contain several different mapping relationships in different subspaces. Thus, using one global model to represent all the mappings contained in the data is difficult. The case reuse accuracy is important, i.e., an accurate case reuse model can generate a reasonable suggested solution, and then in the following case revise phase, where this suggested solution is evaluated and repaired if it fails to satisfy the requirements, the suggested solution does not need too many revisions, which gives engineers time to do creative work. Therefore, an approach to improve case reuse accuracy is required. More importantly, the above mentioned approaches all used one-layer model, which did not consider the error of the model. If the error can be modeled, the accuracy may be improved.

The contributions of this paper are: (i) the proposed two-layer random forests model can improve case reuse accuracy and stability; (ii) the gravitational self-organizing mapping (gSOM) algorithm (Ilc & Dobnikar, 2012) is adopted to organize the cases and improve the case retrieval efficiency. The two-layer model scheme contains two layers, where the first layer model pre-estimates the output and the second layer model is added to model the error of the first layer model. The random forests algorithm, which is a popular ensemble learning algorithm, is adopted in the two-layer model scheme to improve accuracy and stability. In addition, massive cases exist in case base and these cases may belong to different categories, i.e., the mappings from inputs to outputs of these cases may be different. Thus, the first step is organizing all the cases such that cases with the same mapping are in one sub-base; the gSOM clustering algorithm is adopted because it can determine the cluster number automatically based on the structure of the data.

This paper is structured as follows. Section 2 introduces the background, which includes ensemble learning and the gSOM algorithm. Then, Section 3 details the framework of the proposed CBR system and the proposed case reuse model. Next, Section 4 shows the experiments, where the proposed model is evaluated by using several public datasets, and then applied to hydraulic generator design. Finally, Section 5 is the conclusion.

2. Background

This section contains two subsections: ensemble learning and the gSOM clustering algorithm.

2.1. Ensemble learning

Ensemble learning is a machine learning paradigm where multiple learners are trained and aggregated to solve the same problem. The conventional machine learning approaches try to learn one global model from the data, such as ANN and MRA mentioned above. However, the ability of these global models is limited; hence, they are only effective in applications involving simple hypotheses. The ensemble learning models construct multiple learning models and combine them together to obtain better performance than could be obtained by any of the constituent learning models (Rokach, 2010), which are called base learners, i.e., ANN, decision tree, and so on (Polikar, 2006).

To guarantee that the ensemble learning model performs better than its base learners, two principles on base learners should be followed. First, the more accurate the base learners are, the better the performance of the ensemble learner will be (Krogh & Vedelsby, 1995). The second principle is diversity (Polikar, 2006), which means that all the base learners in one ensemble learning model should differ from each other. To achieve diversity, three categories of ensemble strategies have been adopted: (i) using different types of base learners; (ii) manipulating training data (Zhou, 2009); (iii) and manipulating input features. The last two categories can be used when base learners are all of the same type, which are introduced as follows.

All base learners are trained individually; thus, every base learner has its own training sample set. By using the second category of strategies, all base learners will have different training sample sets. Then after the training phase, all base learners will have different parameters even if their types are the same. The most popular strategies in this category are bagging (Breiman, 1996) and boosting (Freund & Schapire, 1997) strategies, which are shown in Fig. 1.

Bagging strategy is the abbreviation of bootstrap aggregating, which uses bootstrap sampling technique to aggregate the base learners. For every base learner, the sampling with replacement technique is adopted to obtain its training samples. Because of the stochastic property of sampling technique, all base learners will have different training sample sets and then all the trained base learners will be different. Boosting strategy also uses the sampling with replacement technique, while all the samples are assigned weights, which denote the probabilities that these samples are selected during the

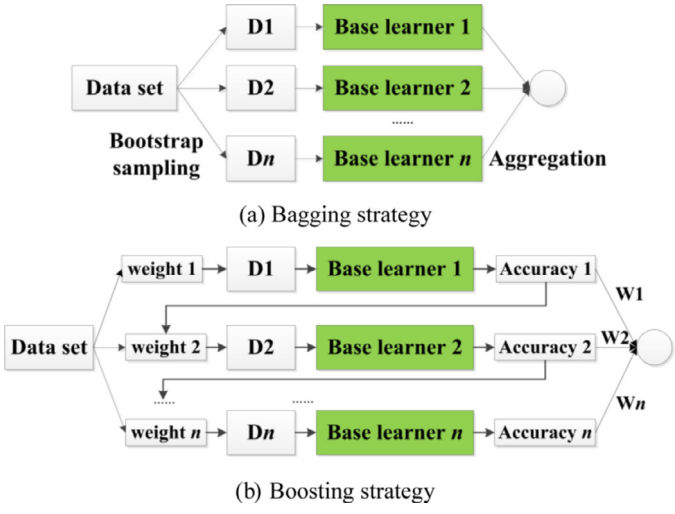

Fig. 1. Bagging and boosting strategies, (a) Bagging strategy and (b) boosting strategy.