Engineering Applications of Artificial Intelligence 37 (2015) 170–180

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Incorporating adaptability-related knowledge into support vector machine for case-based design adaptation

Jin Qi *, Jie Hu, Yinghong Peng

Institute of Knowledge Based Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 22 March 2014
Received in revised form
13 September 2014
Accepted 15 September 2014
Available online 8 October 2014

Keywords:
Case-based design
Case adaptation
Adaptation knowledge
Support vector machine
Adaptability analysis

ABSTRACT

In case-based design systems, the adaptation operation based on similar cases is a difficult and complex step, and the more adaptable cases usually could make larger contribution for adaptation generation than less ones. Under this ideology, this paper addresses a new case adaptation method which uses support vector machine (SVM) incorporating adaptability-related knowledge provided by the retrieved cases, called adaptability-involving SVM (ASVM). The knowledge of adaptability includes the adaptability characteristic of old cases returned by the adaptability analysis and the guideline that the training data from adaptable case should be given higher weight to build SVM model. So the content of this work presented here consists of two parts. The first one is to explore the adaptable property of old cases by utilizing decision tree technology. The second one is to study the construction of ASVM adaptation model in terms of retrieved cases. We first employ the differences between test and retrieved cases to assemble the adaptation pattern data for ASVM model training. Then the higher adaptability coefficients are given to the training data from more adaptable cases than those from less adaptable cases. We adopt ASVM in actual power transformer design to illustrate its feasibility, and carry out comparison researches with different numbers of retrieved cases in the different data sets to validate its superiority, through comparing the adaptation error results with those provided by other classical methods. Empirical results show that ASVM is feasible and validated for case adaptation.

1. Introduction

Nowadays rapid product development requires methodologies to efficiently explore the design space to build products with reduced costs, improved functionality and quality to compete in global markets. At present, design task has had an enormous improvement thanks to new computer aided methods, among which the most widely used strategies for rapid product development is parametric design (Gane and Haymaker, 2012) which has been known to reduce design time with minimal expense. Traditionally, the parametric values of new product are determined by decision-maker based on his/her expertise and experiences. To improve such qualitative method, the case-based reasoning (CBR) technique has been introduced to design product, i.e., case-based design (CBD). By its nature, the case is deemed to be a representative sample of problem-solving in this domain, and CBD system is a knowledge-based system which solves new design problems by matching the problem feature-values and adapting the solution feature-values of similar old cases (Qi et al., 2012). However, existing commercial CBD systems are generally characterized by a sophisticated case organization and retrieval mechanism, but do not include a well developed case adaptation framework (Finnie and Sun, 2003). This is due to the fact that case adaptation generally needs to be guided by some organized form of domain knowledge, while adaptation knowledge is not always accessible and available.

Early studies employed hand coded adaptation rules for case adaptation, which demand a significant knowledge acquisition effort for case adaptation (Smyth and Keane, 1996). Later, different from manually acquiring adaptation knowledge, several machine learning (ML) methods are applied to perform automatic case adaptation. Recently, support vector machine (SVM), a new way to train polynomial neural networks (NNs) based on the structural risk minimization principle, has been successfully proved to be superior to classical NNs in solving classification and regression problems (Sakr and Elhajj, 2013; Zhang et al., 2013; Gryllias and Antoniadis, 2012; Chuang and Lee, 2011). In general, SVM for regression problem uses some arbitrarily chosen loss functions which equally penalize errors on all training samples, therefore all training examples are considered equally significant. On the other hand, recent papers (Chebel-Morello et al., 2013; Nouaouria and Boukadoum, 2013; Qi et al., 2012; Stéphane et al., 2010) have confirmed that more adaptable case could provide more important

* Corresponding author. Tel.: +86 21 34206076; fax: +86 21 34206313.
E-mail address: jinhuaqj@sjtu.edu.cn (J. Qi).

http://dx.doi.org/10.1016/j.engappai.2014.09.010
0952-1976/

information than those from less adaptable cases for adaptation generation. Therefore, it is advantageous to give high weights on the training data provided by adaptable case to build SVM model.

Inspired by this idea, this paper concentrates on the numerical parameterized adaptation, and proposes a new adaptation method in CBD system. The proposed approach is an improvement of classical SVM-based adaptation, which is the first attempt to integrate SVM model with adaptability-related knowledge from case-base, and provide empirical evidences to prove its validity. The breakdown of this research is divided into six sections. Section 2 is a description on research background. Section 3 builds up the new adaptation framework. Section 4 gives an example to illustrate the procedure of the proposed method. An empirical experiment is provided to make a comparative analysis in Section 5. Section 6 makes conclusions.

2. Research background

2.1. Knowledge-light case adaptation

To overcome the challenge of acquiring sufficient programmable knowledge for case adaptation, the concept of "knowledge-light" adaptation has been proposed (Mitra and Basak, 2005), which aims to reduce the engineering effort needed for the acquisition and organization of adaptation knowledge by employing the knowledge already contained inside the CBR system and its components. Early studies of knowledge-light adaptation acquired the adaptation rules by analyzing the differences between cases and their corresponding solutions, and identifying, if possible, a plausible pattern (Jarmulak et al., 2001). Later, some researchers combined ML methods into CBR to obtain adaptation knowledge. The design of these ML-based knowledge-light algorithms is ideally independent of the domain knowledge, or very little domain knowledge is required for making the adaptation methods. Among them, genetic algorithm (GA) based adaptation (Huang et al., 2009; Saridakis and Dentsoras, 2007; Renner and Ekárt, 2003) and neuro-adaptation (Henriet et al., 2014; Butdee, 2012; Jung et al., 2009; Craw et al., 2006; Lofty and Mohamed, 2003) are two typical knowledge-light methods, where the definition of GA and NN can be guided by the domain knowledge, but the evolution of GA or modelization of NN are not necessarily guided by domain knowledge. Overall, these investigations explore the utilization of inductive learning to acquire adaptation knowledge from examples and apply the acquired knowledge to implement automatic case adaptation. However, such methods have some inherent drawbacks, e.g., the poor performance for high number of attributes, the problem of multiple local convergence, and the danger of overfitting. There are two solutions to overcome these shortcomings, in general, one is to improve the traditional approaches such as improved GA (Liao et al., 2012), the other is to construct a new adaptation model in terms of new algorithm such as SVM (Sharifi et al., 2013; Policastro et al., 2008). Compared to other MLs, SVM could achieve an optimum network structure, and eventually result in better generalization performance for data set with a large number of attributes, as it implements the structural risk minimization principle and tries to minimize an upper bound of generalization error instead of minimizing the misclassification error or deviation from correct solution of the training data (Tay and Cao, 2001). In addition, the solution of SVM may be a global optimum while other ML models may tend to fall into a local optimal solution, so overfitting is unlikely to occur with SVM.

However, Passone et al. (2006) pointed out that insufficient knowledge can badly affect the selection of an appropriate learning algorithm and its performance. To overcome the shortcoming, some researchers proposed new knowledge-light methods which not only get knowledge from the CBR system itself, but also regard that knowledge as the starting point for adaptation processes, and find new knowledge through other ML methods (Minor et al., 2014; Assali et al., 2013; Goh and Chua, 2010).

2.2. Classical support vector machine for case adaptation

According to the study of Policastro et al. (2008), the training data of SVM for case adaptation should be represented as $\{x_i, y_i\}_{i=1}^{n}$, where x_i is a $2p+1$-dimensional input vector of ith training example, including p problem values of retrieved case, p problem values of target case, and one solution value of retrieved case, y_i is the corresponding solution value stored in target case, and n is the total number of training examples. The basic idea of SVM for adaptation is to do regression approximation which addresses the problem of estimating a function to model the associative relations between x_i and y_i. SVM approximates the unknown function with the form $f(x)=\omega^T\varphi(x)+\beta$, where $\{\varphi(x)\}_{i=1}^{n}$ is the high dimensional feature space, nonlinearly mapped from the input space. $\{\omega\}_{i=1}^{n}$ and β are the normal vector and the bias, estimated by minimizing:

$$R(\omega)=C\frac{1}{n}\sum_{i=1}^{n}L_\varepsilon(y_i,f(x_i))+\frac{1}{2}\|\omega\|^2 \tag{1}$$

$$L_\varepsilon(y_i,f(x_i))=\begin{cases}|y_i-f(x_i)|-\varepsilon & \text{if } |y_i-f(x_i)|\geq\varepsilon\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

where $C(1/n)\Sigma_{i=1}^{n}L_\varepsilon(y_i,f(x_i))$ is the empirical error (risk) between expected solution value y_i and calculated output $f(x_i)$, measured by the ε-insensitive loss function given by Eq. (2). This loss function allows us to use sparse data points to represent the function $f(x)$. The second term $1/2||\omega||^2$ is the regularization term. C is referred to as the regularized constant which determines the trade-off between the empirical error and the regularization term. ε is the tube size and it is equivalent to the approximation accuracy placed on the training data. Both C and ε are user-prescribed parameters and are selected empirically.

To obtain the estimations of ω and β, Eq. (1) is transformed to the primal function given by Eq. (3) by introducing positive slack variables ξ_i and ξ_i^* as follows:

$$\text{Minimize}\quad R\left(\omega,\xi,\xi^*\right)=\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{n}\left(\xi_i+\xi_i^*\right)$$
$$\text{s.t.}\quad\begin{cases}y_i-\left(\omega^T\varphi(x_i)+\beta\right)\leq\varepsilon+\xi_i\\ \left(\omega^T\varphi(x_i)+\beta\right)-y_i\leq\varepsilon+\xi_i^*\\ \xi_i,\xi_i^*\geq 0,\quad i=1,2,\ldots,n\end{cases} \tag{3}$$

Although non-linear function φ is usually unknown, all computation related to φ can be reduced to the form $\varphi(x)^{\mathrm{T}}\varphi(y)$, which can be replaced with so-called kernel function $K(x,y)$ that satisfies Mercer's condition. Finally, the function $f(x)$ takes the following explicit form:

$$f(x,\delta_i,\delta_i^*)=\sum_{i=1}^{n}(\delta_i-\delta_i^*)K(x,x_i)+\beta \tag{4}$$

where Lagrange multipliers δ_i and δ_i^* satisfy the equality $\delta_i\delta_i^*=0$, $\delta_i\geq 0$, $\delta_i^*\geq 0$, and they are obtained by maximizing the dual form of Eq. (3) which has the following form:

$$R\left(\xi,\xi^*\right)=\sum_{i=1}^{n}y_i\left(\xi_i-\xi_i^*\right)-\varepsilon\sum_{i=1}^{n}\left(\xi_i+\xi_i^*\right)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\xi_i-\xi_i^*\right)\left(\xi_j-\xi_j^*\right)K(x_i,x_j) \tag{5}$$

with the following constraints:

$$\sum_{i=1}^{n}(\delta_i-\delta_i^*)=0,\quad 0\leq\delta_i\leq C,\quad 0\leq\delta_i^*\leq C,\quad i=1,2,\ldots,n \tag{6}$$

Based on the Karush–Kuhn–Tucker conditions of the quadratic programming, only a number of $(\delta_i-\delta_i^*)$ will assume non-zero values, and the input element associated with them could be