Chapter 17
Machine Learning and Value-Based Software Engineering

Du Zhang
California State University, USA

ABSTRACT

Software engineering research and practice thus far are primarily conducted in a value-neutral setting where each artifact in software development such as requirement, use case, test case, and defect, is treated as equally important during a software system development process. There are a number of shortcomings of such value-neutral software engineering. Value-based software engineering is to integrate value considerations into the full range of existing and emerging software engineering principles and practices. Machine learning has been playing an increasingly important role in helping develop and maintain large and complex software systems. However, machine learning applications to software engineering have been largely confined to the value-neutral software engineering setting. In this paper, the general message to be conveyed is to apply machine learning methods and algorithms to value-based software engineering. The training data or the background knowledge or domain theory or heuristics or bias used by machine learning methods in generating target models or functions should be aligned with stakeholders' value propositions. An initial research agenda is proposed for machine learning in value-based software engineering.

INTRODUCTION

Software engineering research and practice thus far are mainly conducted in a value-neutral setting where each artifact in software development such as a requirement, a use case, a test case, a defect, and so forth, is treated as equally important during a software system development process (Boehm, 2006a). There are a number of shortcomings of such value-neutral software engineering (Biffl et al. 2006): (1) its exclusion of economics, management sciences, cognitive sciences, and humanities from the body of knowledge needed to develop successful software systems; (2) its delimitation

DOI: 10.4018/978-1-4666-0261-8.ch017

of software development by mere technical activities; and (3) its failure to explicitly recognize the fact that software systems continue to satisfy and conform to evolving human and organizational needs is to create value. Value-based software engineering (VBSE) is to integrate value considerations into the full range of existing and emerging software engineering principles and practices so as to increase the return on investment (ROI = (benefits–costs)/costs) for the stakeholders and optimize other relevant value objectives of software projects (Biffl et al. 2006; Boehm, 2006a, Wang, 2007).

Machine learning (ML) has been playing an increasingly important role in helping develop and maintain large and complex software systems. However, machine learning applications to software engineering have been largely confined to the value-neutral software engineering setting (Zhang, 2000; Zhang & Tsai, 2003; Zhang & Tsai, 2005; Zhang & Tsai, 2007, Wang, 2008). In this paper, the general message we attempt to convey is to apply ML methods beyond the value-neutral software engineering setting and to VBSE. The training data or the background knowledge or domain theory or heuristics or bias used by ML methods in generating target models or functions for software development and maintenance should be aligned with stakeholders' value propositions (SVPs) and business objectives. Even though the transition to VBSE from the traditional value-neutral setting is necessarily evolutionary because not all the theories, infrastructures, methodologies and tools for VBSE have been fully developed yet, there are a number of agenda items for VBSE (Boehm, 2006a).

The goal of the road map in VBSE is to make software development and maintenance decisions that are better for value creation (Boehm, 2006a). On the other hand, the hallmark of ML is that it results in an improved ability to make better decisions. VBSE offers a fertile ground where many software development and maintenance tasks can be formulated as ML problems and approached in terms of ML methods. The purpose of this paper is to describe an initial research agenda for ML applications to VBSE with regard to the identified areas in VBSE (value-based requirement engineering, architecting, design and development, verification and validation, planning and control, risk/quality/people managements, and a theory of VBSE (Boehm, 2006a)).

The rest of the paper is organized as follows. Section 2 offers an overview of the related work. Section 3 highlights some important concepts in VBSE. In Section 4, we describe an initial research agenda for ML applications in VBSE. Finally Section 5 concludes the paper with remark on future work.

RELATED WORK

In this section, we provide a brief account for some of the major and emerging software development paradigms which are related to the main theme of this paper. The intent is to highlight the state-of-the-art in the software development landscape and to delineate differences between the existing approaches and the one advocated in this paper.

Besides machine learning in (value-neutral) software engineering (MLSE), there are a number of related and emerging software development paradigms: search-based software engineering (SBSE), evidence-based software engineering (EBSE), model-based software engineering (MBSE), artificial intelligence in software engineering (AISE), and computational intelligence in software engineering (CISE). Figure 1 highlights their similarities and differences.

MLSE

ML falls into the following broad categories: *supervised* learning, *unsupervised* learning, *semi-supervised* learning, *analytical* learning, *reinforcement* learning, and *multi-agent* learning. Each of the categories in turn includes various