Chapter 4

Auto-Associative Neural Networks to Improve the Accuracy of Estimation Models

Salvatore A. Sarcia
Università di Roma Tor Vergata, Italy

Giovanni Cantone
University of Maryland, USA

Victor R. Basili
University of Maryland, USA

ABSTRACT

Prediction of software engineering variables with high accuracy is still an open problem. The primary reason for the lack of high accuracy in prediction might be because most models are linear in the parameters and so are not sufficiently flexible and suffer from redundancy. In this chapter, we focus on improving regression models by decreasing their redundancy and increasing their parsimony, i.e., we turn the model into a model with fewer variables than the former. We present an empirical auto-associative neural network-based strategy for model improvement, which implements a reduction technique called Curvilinear component analysis. The contribution of this chapter is to show how multi-layer feedforward neural networks can be a useful and practical mechanism for improving software engineering estimation models.

INTRODUCTION

Prediction of software engineering variables such as project cost, fault proneness, and number of defects is a critical issue for software organizations. It is important to get the best estimate possible when planning a new project, activity, or task. For instance, we may have to predict the software cost, the effort in carrying out an activity (e.g. coding a module), or the expected number of defects arising from a module or sub-system. Improving the prediction capability of software organizations is one way of improving their competitive advantage. Better predictions can improve the development process in terms of planning resources, setting and achieving quality goals, and making more informed decisions

DOI: 10.4018/978-1-60566-758-4.ch004

about the schedule. The point is that, prediction is a major problem when trying to better manage resources, mitigate project risk, and deliver products on time, on budget and with the required features and functions (quality) (Boehm, 1981).

Despite the fact that prediction of software variables with high accuracy is a very important issue for competing software organizations, it is still an unsolved problem (Shepperd, 2007). A large number of different prediction models have been proposed over the last three decades (Shepperd, 2007). There are predefined mathematical functions (e.g., FP-based functions, COCOMO-I, and COCOMO-II), calibrated functions (e.g., function based on a regression analysis on local data), machines learning (e.g., estimation by analogy, classification and regression trees, artificial neural networks), and human-based judgment models. There exist a large number of empirical studies aimed at predicting some software engineering variable (Myrtveit, 2005). Most often, the variable is software project cost, which we will use as the main exemplar of the approach described in this chapter.

In this chapter, we present a computational intelligence technique to improve the accuracy of parametric estimation models based on regression functions. The improvement technique is defined, tested, and verified through software cost estimation data (Sarcia, Cantone, & Basili, 2008). In particular, this chapter offers a technique that is able to improve the accuracy of log-linear regression models in the context of the COCOMO-81 projects (Boehm, 1981; Shirabad, & Menzies, 2005). From a mathematical point of view, even if we change the model variables, e.g., we predict effort instead of predicting the number of defects, or vice versa, the strategy to improve the estimation model does not change at all. This means that we can apply the improvement strategy to any set of variables in any kind of development context without change. Of course, it does not mean that the improvement technique will succeed in every context.

It is important to note that, even if we were making predictions through human-based judgment, to increase estimation accuracy we would have to build parametric models calibrated to the local data and use them to estimate variables of interest for the specific context of the organization (McConnell, 2006). For instance, productivity (lines of code/time) may change according to the context of the organization (e.g., capability of analysts and developers, complexity of the software system, environment of the development process). The point is that, even experts should use regression functions for gathering suitable information from the context in which they are operating. Therefore, apart from the kind of estimation model used, when improving estimation accuracy, calibrating regression functions is a required activity. We believe that, enhancing accuracy of regression functions is the core of any estimation activity. That is why we focus upon improving regression functions and do not deal with other approaches.

We begin with an introduction that provides the general perspective. Then, we provide some background notes, required terminology, and define the problem. We present, from a practical point of view, the main issues concerning parametric estimation models, multi-layer feed-forward neural networks, and auto-associative neural networks. We illustrate the empirical strategy for dealing with the problem (the solution). We conclude with a discussion of the benefits and drawbacks of the methodology and provide some ideas on future research directions.

PARAMETRIC ESTIMATION MODELS

The estimation models that we refer to are based on parametric models as illustrated in Figure 1.

Mathematically, the estimation model (EM) in Figure 1 can be represented by a regression function f_R such that $y = f_R(x,\beta) + \varepsilon$, where x is a set of independent variables, y is the dependent