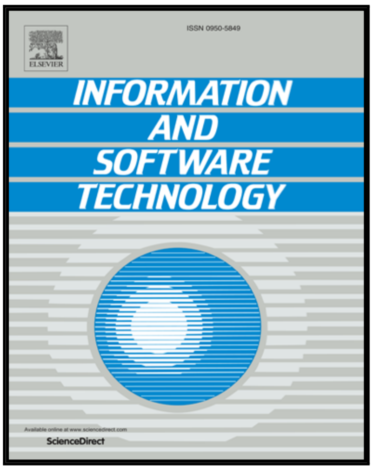
Accepted Manuscript




# Support Vector Regression for Predicting Software Enhancement Effort

Andrés García-Floriano , Cuauhtémoc López-Martín ,
Cornelio Yáñez-Márquez , Alain Abran

PII: S0950-5849(17)30290-2
DOI: 10.1016/j.infsof.2018.01.003
Reference: INFSOF 5941

To appear in: *Information and Software Technology*

Received date: 1 April 2017
Revised date: 16 November 2017
Accepted date: 6 January 2018

Please cite this article as: Andrés García-Floriano , Cuauhtémoc López-Martín , Cornelio Yáñez-Márquez , Alain Abran , Support Vector Regression for Predicting Software Enhancement Effort, *Information and Software Technology* (2018), doi: 10.1016/j.infsof.2018.01.003

# Support Vector Regression for Predicting Software Enhancement Effort

Andrés García-Floriano
*Unidad de Estudios Superiores Tepotzotlán*
*Universidad Mexiquense del Bicentenario*
*Tepotzotlán, México, México*
*andres.garcia@umb.mx*

Cuauhtémoc López-Martín
*Department of Information Systems*
*Universidad de Guadalajara*
*México*
*cuauhtemoc@cucea.udg.mx*

Cornelio Yáñez-Márquez
*Laboratorio de Cómputo Inteligente*
*Instituto Politécnico Nacional*
*Centro de Investigación en Computación*
*México City, México*
*cyanez@cic.ipn.mx*

Alain Abran
*Department of Software Engineering and Information Technology*
*École de technologie supérieure*
*Université du Québec, Canada*
*alain.abran@etsmtl.ca*

**Abstract**
*Context*: Software maintenance (SM) has to be planned, which involves SM effort prediction. One type of SM is enhancement, where new functionality is added or existing functionality changed or deleted.
*Objective*: Analyze the prediction accuracy of two types of support vector regression (ε-SVR and υ-SVR) when applied to predict software enhancement effort.
*Method*: Both types of support vector regression used linear, polynomial, radial basis function, and sigmoid kernels. Prediction accuracies for ε-SVR and υ-SVR were compared with those of statistical regressions, neural networks, association rules, and decision trees. The models were trained and tested with five data sets of enhancement projects from Release 11 of the International Software Benchmarking Standards Group (ISBSG). Each data set was selected on the basis of data quality, development platform, programming language generation, and levels of effort recording.
*Results*: The polynomial kernel ε-SVR (PKε-SVR) was statistically better than statistical regression, neural networks, association rules and decision trees, with 95% confidence.
*Conclusions*: A PKε-SVR could be used for predicting software enhancement effort in mainframe platforms and coded in a third-generation programming languages, and when enhancement effort recording includes the efforts of the development team, its support personnel, the computer operations involvement, and end users.

Keywords: Software enhancement effort prediction, support vector machine, support vector regression, statistical regression, neural networks, association rules, decision trees, ISBSG.

## 1. Introduction

The software development life cycle includes a number of phases (i.e., requirement analysis, design, construction, testing, deployment, and maintenance), usually supported by other activities such as configuration management, engineering management, and quality assurance [1]. The longest phase, and in most cases also the most expensive, is software maintenance (SM) [2]. SM has been defined as "the modification of a software product after its delivery to correct faults, improve performance or other attributes, or adapt the product to a modified environment" [3]. SM has been classified according to its intention, activity and evidence [4]. There are four categories of intention: corrective, adaptive, perfective and preventive. The activity classification expresses the reason for the change, and has two categories: corrections and enhancements. Evidence, from observations and comparisons, measures the software before and after modification. Evidence is classified into four clusters: support interface (training, consultative, evaluative), documentation (reformative, updative), software properties (groomative, preventive, performance, adaptive), and business rules (reductive, corrective, enhancive). The type of maintenance considered for our study is enhancement, defined as "changes made to an existing application where new functionality has been added, or existing functionality has been changed or deleted. This would include adding a module to an existing application, irrespective of whether any of the existing functionality is changed or deleted" [11].

SM is one of the ten areas of the Software Engineering Body of Knowledge (SWEBOK). One objective of SWEBOK is to promote a consistent view of software engineering worldwide [5]. The first release of SWEBOK, published in 2004, included approximately 500 reviewers from 42 countries. In regard to SM, they concluded: