Automation in Construction 47 (2014) 92–106

Contents lists available at ScienceDirect

Automation in Construction

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

# Support Vector Machine Regression for project control forecasting

Mathieu Wauters [a], Mario Vanhoucke [a,b,c,*]

[a] Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
[b] Technology and Operations Management Area, Vlerick Business School, Reep 1, 9000 Ghent, Belgium
[c] Department of Management Science and Innovation, University College London, Gower Street, London WC1E 6BT, United Kingdom

ARTICLE INFO

*Article history:*
Received 25 April 2014
Received in revised form 19 June 2014
Accepted 31 July 2014
Available online 30 August 2014

*Keywords:*
Project management
Earned Value Management (EVM)
Support Vector Regression (SVR)
Prediction

ABSTRACT

Support Vector Machines are methods that stem from Artificial Intelligence and attempt to learn the relation between data inputs and one or multiple output values. However, the application of these methods has barely been explored in a project control context. In this paper, a forecasting analysis is presented that compares the proposed Support Vector Regression model with the best performing Earned Value and Earned Schedule methods. The parameters of the SVM are tuned using a cross-validation and grid search procedure, after which a large computational experiment is conducted. The results show that the Support Vector Machine Regression outperforms the currently available forecasting methods. Additionally, a robustness experiment has been set up to investigate the performance of the proposed method when the discrepancy between training and test set becomes larger.

## 1. Introduction

Project scheduling first originated as a subdiscipline of Operations Research with the goal of establishing start and finish times of activities within a project network. These activities are subject to various types of constraints, of which precedence and resource restrictions are the most renowned, while optimizing a certain objective. While the construction of a baseline schedule plays a vital role in the ultimate failure or success of a project, its primary purpose consists of acting as a point of reference. The assessment of a project's risk and the analysis of a project's performance throughout its lifecycle are compared against this predictive plan. Dynamic scheduling [20,25] refers to these three crucial phases in a project's life cycle, namely baseline scheduling, schedule risk analysis and project control. Ever since the inception of the well-known Critical Path Method in the 1950s, the research community focused heavily on project scheduling problems with various extensions. The PERT methodology turned the attention of academics towards the relation between the duration of a project and variability affecting activity durations. The third component of dynamic scheduling is project control. Earned Value Management (EVM) was introduced as a methodology to control a project's time and cost and aids a project manager in keeping track of the execution of a project vis-à-vis the reference point, provided by the baseline schedule. It surfaced in the 1960s thanks to a project of the US Department of Defense. The reader is referred to Fleming and Koppelman [11] for the fundamentals of EVM.

A popular project control topic was the search for accurate and reliable forecasting methods. Forecasting methods that provide a project manager with a reliable estimate of the project's targets are an important asset in the project manager's toolbox. Depending on the allowed deviation, forecasting estimates may serve as early warning signals, triggering actions to bring the project back on track. Even though EVM allows for time and cost monitoring, initial research efforts were mainly directed to cost forecasting. An overview of the different forecasting methods and their accuracy can be found in Christensen [6]. In the early 2000s, the dominance of the cost objective persisted (see e.g. [10] who discuss a project's price tag) until the introduction of the Earned Schedule concept by Lipke [17]. From this point onwards, the time dimension received growing attention, which culminated in publications on time forecasting (see Vandevoorde and Vanhoucke, 2006).

Dynamic scheduling aims at the integration of its three components. The first attempts at integrating schedule risk analysis and project control were executed by Vanhoucke [23] and Vanhoucke [24]. These research studies compare bottom-up (as found in schedule risk analysis) and top-down (as found in EVM) project tracking approaches and study their relation to a project network's topological structure. Furthermore, activity sensitivity was incorporated in a dynamic corrective action framework. In a recent publication, Elshaer [9] proposed an adaptation of one of the Earned Schedule forecasting methods using activity sensitivity metrics. By bridging top-down and bottom-up metrics, he was able to improve the forecasting accuracy of the Earned Schedule method. These publications formed the primary motivation for this paper's research. In order to construct sensitivity measures on the activity level, assumptions need to be made about the range and distribution of the activity durations. Using Monte Carlo simulations, various sensitivity measures can be constructed that provide an idea about the

[*] Corresponding author at: Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium. Tel.: +32 9 264 35 69.
*E-mail addresses:* mathieu.wauters@ugent.be (M. Wauters), mario.vanhoucke@ugent.be (M. Vanhoucke).

http://dx.doi.org/10.1016/j.autcon.2014.07.014
0926-5805/© 2014 Elsevier B.V. All rights reserved.

contribution of an activity to the project's overall sensitivity. However, each simulation run also yields top-down data that can be captured using the EVM performance metrics. This wealth of historical top-down data has great potential value in assisting project managers to make more accurate predictions and will be used by our proposed method. The contribution of this paper is threefold. First of all, we provide a clear framework of how a project manager can use the information from Monte Carlo simulations to improve project forecasting. The field of Artificial Intelligence, a research branch devoted to learning relations between attributes to construct one or multiple outputs, is ideally suited for this purpose. In this paper, we will focus on Support Vector Machines, a well-known technique for classification and prediction. Secondly, this paper intends to improve forecasting estimates using a computational experiment on a large and topologically rich dataset. In order to achieve this purpose, the forecasting accuracy is compared based on a large amount of runs and based on different scenarios. These scenarios provide valuable insights about when the proposed Support Vector Machine approach yields the biggest advantage. Finally, robustness checks are performed to illustrate the pitfalls of using historical data. This is particularly interesting since Artificial Intelligence is susceptible to the well-known “garbage-in, garbage-out” principle.

The outline of this paper is as follows. Section 2 provides a short overview of the underlying principles of Support Vector Regression. In section 3, the research methodology is outlined. The methodology consists of six steps, namely network generation, Monte Carlo simulation, attributes, the division between training and test set, cross-validation and grid search and finally, the testing phase. The settings of the computational experiment are delineated in Section 4 using the six methodological steps. Section 5 presents the main results from the computational experiment and is broken down as follows. First, Section 5.1 provides a thorough discussion of the fine tuning process of the parameters of the Support Vector Regression. A link between the simulation scenario, topological structure and forecast accuracy is established. Next, the relation between accuracy and the project's point of completion is scrutinized. Finally, the limitations of our findings are discussed in Section 5.3 which deals with a robustness check of the computational study. Section 6 draws conclusions and highlights future research avenues.

## 2. Support Vector Machine Regression

### 2.1. General theory

Support Vector Machines (SVMs) in their current form were developed at the AT&T Bell Laboratories and gained momentum with the paper by Cortes and Vapnik [7]. Initial applications focused on binary classification of test instances and pattern recognition. With the rapidly increasing attention for SVMs, a number of introductory articles surfaced and constitute the foundation for this section [2,19,18]. In general, SVMs employ a model to construct a decision surface by mapping the input vectors into a high-dimensional (or infinite-dimensional) feature space. Next, a linear regression is executed in the high-dimensional feature space. This mapping operation is necessary because most of the time, the relation between a multidimensional input vector x and the output y is unknown and very likely to be non-linear. Support Vector Machine Regression (SVR) aims at finding a linear hyperplane, which fits the multidimensional input vectors to output values. The outcome is then used to predict future output values that are contained in a test set. Let us define a set of data points $P = (x_i, a_i)$, $i = 1,\ldots n$ with $x_i$ the input vector of data point $i$, $a_i$ the actual value and $n$ the number of data points. For linear functions $f$, the hyperplane that is constructed by the SVR is determined as follows:

$$f(x) = wx + b. \tag{1}$$

Notation-wise, Eq. (1) displays similarities to a linear regression model. The predicted value, $f(x)$, depends on a slope $w$ and an intercept $b$. In general, one wants to strike a balance between learning the relation between inputs and outputs while maintaining a good generalization behavior. An excessive focus on minimizing training errors may lead to overfitting. A model with low complexity is limited with regard to the decision boundary it can produce but is less likely to overfit. In Cortes and Vapnik [7], it is shown that the probability of a test error depends on two factors, namely the frequency of the training error and a confidence interval, where both factors form a trade-off. The confidence interval is related to the Vapnik–Chervonenkis dimension of the Support Vector Machine, which can be thought of as the complexity of the learning model. Hence, improved generalization may be obtained by improving the confidence interval at the expense of additional training errors. The primary instrument to control this trade-off is C, which explains its importance. The balance between good training and generalization behavior is reflected in Eq. (2), where R denotes the compound risk caused by training errors and model complexity. Naturally, the risk $R$ needs to be kept as low as possible.

$$R = \frac{C}{n}\sum_{i=1}^{n} L_\epsilon(a_i, f(x_i)) + \frac{1}{2}\left|\left|w\right|\right|^2. \tag{2}$$

Eq. (2) yields estimated values for $w$ and $b$ and consists of two main parts. The first part, $\frac{C}{n}\sum_{i=1}^{n} L_\epsilon(a_i, f(x_i))$ consists of the training or empirical risk and is measured by the $\epsilon$-insensitive loss function, $L_\epsilon(a, y)$ (see e.g. [29]). This function implies that the prediction error is ignored if the difference between the predicted value $f(x)$ and the actual value $a$ is smaller than $\epsilon$. The $\epsilon$-insensitive loss function is formally defined in Eq. (3).

$$L_\epsilon(a, y) = \begin{cases} |a - f(x)| - \epsilon \; |a - f(x)| & \geq \epsilon \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The second part of Eq. (2), $\frac{1}{2}||w||^2$, is the regularization term and is related to the complexity of the model (see Cortes and Vapnik [7]). $C$ controls the trade-off between the regularization term and the training accuracy. Large values of $C$ imply that more weight is put on correctly predicting training points, at the cost of a higher generalization error.

The problem of finding an optimal hyperplane is a convex optimization problem. For non-linear relations between input vectors and outputs, it is necessary to define a map, $\phi$, that translates the training points $x_i$ into a higher-dimensional feature space. The consequence is that $w$, after constructing a Lagrangean function from Eq. (1) will no longer be a function of $x_i$ but of $\phi(x_i)$ and that the product $\phi(x_i)\phi(x)$ needs to be calculated. We refer the reader to Smola and Schölkopf [19] for full details about these observations. The function $\phi(x_i)\phi(x)$ is often defined as $K(x_i, x)$ and is referred to as a kernel function. Kernel functions try to achieve linear separability between training points in the higher-dimensional feature space. Many kernel functions exist. In fact, any function that satisfies Mercer's condition [30] can serve as a kernel function. An overview of frequently occurring kernel functions is given in Table 1. $\gamma$, $r$ and $d$ are parameters that are kernel-specific. It is worth noting that the Radial Basis Function kernel (RBF) is sometimes parameterized using $\frac{1}{\delta^2}$ instead of using $\gamma$.

**Table 1**
Overview of common kernel functions.

| Kernel name | Formula |
|---|---|
| Linear | $x_i^T x$ |
| Polynomial | $(\gamma x_i^T x + r)^d$ |
| Radial basis | $e^{-\gamma(\|\|x_i - x\|\|^2)}$ |
| Sigmoidal | $tanh(\gamma x_i^T x + r)$ |