JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
ISSN 1392-3730 print/ISSN 1822-3605 online
2014 Volume 19(Supplement 1): S202–S211
doi:10.3846/13923730.2013.801919

PREDICTING PROJECT SUCCESS IN CONSTRUCTION USING AN EVOLUTIONARY GAUSSIAN PROCESS INFERENCE MODEL

Min-Yuan CHENG, Chin-Chi HUANG, Andreas Franskie Van ROY
The National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
Received 16 Mar 2012; 01 Jun 2012

Abstract. There are many factors that affect the success of the implementation process of a project. The importance of each of these factors varies according to the different phases of the project lifecycle, which makes it very difficult to predict the final result of a project. In practice, foreseeing the result of a project is based on the judgment of those in management, which is grounded in their experience. This study aimed to build an Evolutionary Gaussian Process Inference Model (EGPIM), using a Gaussian process, along with Bayesian inference and particle swarm optimization, which helps to optimize the hyper-parameters required for making Gaussian process predictions. With this model at its core, this study can efficiently extract expert knowledge and experience from case studies and historical data to determine relationships between factors which significantly influence the outcome of a project so that its success may be predicted. Historical cases were ordered as a time series based on the Continuous Assessment of Project Performance (CAPP) research results. The model was trained using the EGPIM and these cases to predict the success of a project. This model proved quite accurate at predicting the success of a project and had outstanding performance in time-series applications.

Keywords: Gaussian process, particle swarm optimization, Bayesian inference, project success, CAPP, EGPIM.

Reference to this paper should be made as follows: Cheng, M.-Y.; Huang, C.-C.; Roy, A. F. V. 2013. Predicting project success in construction using an evolutionary Gaussian process inference model, *Journal of Civil Engineering and Management* 19(Supplement 1): S202–S211. http://dx.doi.org/10.3846/13923730.2013.801919

Introduction

The primary task of performance control is to ensure that project goals are achieved and to provide feedback on the status of each phase of construction. However, post-implementation performance evaluation is resource-intensive, time consuming and is impotent in its influence on the success of the project's implementation. It also does not provide the benefits of real-time monitoring of the current construction status.

Traditional methods of project control are commonly based on the experience and habits of those in management. The subjectivity of the choice of these methods often leads to error. This is especially prominent in the management of larger construction projects as predicting a number of possible issues from a huge set of data become more difficult. In recent years, there have been many studies dedicated to improving project success. Khosravi and Afshari (2011) proposed a success measurement model for construction projects to determine how successful projects were after their closing phase. There have also been many academic assessments of Critical Success Factors within construction projects (Chan *et al.* 2004; Griffith *et al.* 1999; Sanvido *et al.* 1992).

The time series method is widely used in construction to make predictions based on historical data. In order to preserve past experience and to resolve the issue of huge datasets in project control, the "Continuous Assessment of Project Performance" (CAPP) system was developed by the Construction Industry Institute (CII) and was used to collect and compile project information and analyse the differences between successful and unsuccessful project progress s-curves (Russell *et al.* 1997). Statistical analyses using this system were undertaken by various studies to confirm the significance levels of known factors that influence project performance and to investigate whether there are other key factors that may influence the success of a project. Even though CAPP is useful in analysing these factors, it is not able to accurately predict the end result of a project. Ko and Cheng (2007) proposed to build prediction models using an Evolutionary Fuzzy Neural Inference Model (EFNIM), but in practice the required calculations are time and system resource consuming, making it difficult to update prediction models. For this reason, this study adopted the Evolutionary Gaussian Process Inference Model (EGPIM) to solve this issue.

The EGPIM features a short training time and precise predictions, making it suitable for application as a dynamic prediction model to provide construction managers with information about the project in real time to aid their decision making. The dynamic prediction model that

Corresponding author: Andreas Franskie Van Roy
E-mail: d9505106@gmail.com

this study used to calculate the success of a project is based on information that was collected from CII's database of historical information. The CAPP was first used to perform a statistical analysis of the influential factors, thus confirming the key factors that influence project success. A time series was then applied to organize the cases from the database. With that done, the EGPIM was applied to these cases for training before going on to predict the success of new projects. The resultant prediction is able to assist those in project management to efficiently control project performance, expedite the discovery of potential problems in the field as well as remedy these problems during construction.

With these benefits in mind, a database was created using the CAPP research results. A time series was then applied to this data for sorting and the EGPIM was applied to build a dynamic prediction model for the success of a project. It was verified that the time series predictions of the EGPIM were very precise and the current project performance was monitored in real time so that management personnel can handle the project more efficiently.

1. Review of approaches

1.1. Gaussian process regression

Gaussian process (GP), an artificial technique actively developed in recent years, has been applied in the fields of chemistry, construction, and medicine, among others (Brahim-Belhouari, Bermak 2004). In the field of construction, GP has primarily been applied in regression and classification prediction. Yan *et al.* (2011) proposed a GP machine learning-based model to classifying surrounding rocks. Su and Xiao (2011) combined the Gaussian process (GP) and importance sampling method (ISM) in a new method to analyse slope reliability that obtained highly accurate results.

Along with other AI techniques, GP gives a statistical advantage and is easy to learn (Chu, Ghahramani 2005; Kocijan *et al.* 2004); thus, based on probability theorem, Gaussian Process can not only make predictions on unknown input data, but can also provide prediction accuracy based on the predictions (estimation variances), which highly elevates the statistical significance in prediction (Bonilla *et al.* 2009). GP can be regarded as a combination of random variances, of which capricious and limited numbers of random variances all obey Gaussian distribution:

$$F(\boldsymbol{X}) = \{f(X_1), f(X_2), \ldots, f(X_N)\} \sim N(\mu, K), \quad (1)$$

where: μ is the mean of variances; and K is covariance matrix. $\boldsymbol{X}$ is the collection of data input factors of N dimensions $X_1, X_2, \ldots, X_N$, GP can be described via mean function $m(X)$ in $f(X_i)$ and covariance function $k(X,X')$ in a random process.

$$f(X) \sim GP\left(m(X), k(X, X')\right). \quad (2)$$

In real situations, however, data prediction is often accompanied by noise, and therefore, when the value $\boldsymbol{Y}$ is calculated by the estimation of the function, an error parameter ϵ should be considered. Likewise, ϵ also coincides with the Gaussian distribution. $\boldsymbol{Y}$ is calculated as follows:

$$\boldsymbol{Y} = F(\boldsymbol{X}) + \epsilon . \quad (3)$$

Denoting the training set as $\{\boldsymbol{X}, \boldsymbol{Y}\}$, new input data is X_*, and desire output is Y_*.

Joint distribution calculated under Gaussian distribution; θ represents the parameters in the joint distribution:

$$\begin{bmatrix} \boldsymbol{Y} \\ Y_* \end{bmatrix} \Big| \boldsymbol{X}, \theta \sim N\left(0, \begin{bmatrix} K + \sigma^2 I & k \\ k^T & \kappa + \sigma^2 \end{bmatrix}\right), \quad (4)$$

where: $k = \left[k(X_*, X_1) \ldots k(X_*, X_N)\right]^T$ is the $n \times 1$ vector formed from the covariance between X_* and the training input $\boldsymbol{X}$. The scalar $\kappa = k(X_*, X_*)$, σ^2 is variance.

Hence, the conditional of probability distribution can also be calculated with expected value together with noise:

$$Y_* \big| \boldsymbol{Y}, \boldsymbol{X}, \theta, \sigma^2 \sim N(m(X_*), v(X_*)) . \quad (5)$$

In the end, based on conditional probability distribution, the mean $m(X_*)$ and variance $v(X_*)$ of expected value Y_* can be calculated.

$$m(X_*) = k^T (K + \sigma^2 I)^{-1} \boldsymbol{Y} ; \quad (6)$$

$$v(X_*) = \kappa + \sigma^2 - k^T (K + \sigma^2 I)^{-1} k . \quad (7)$$

1.2. Bayesian inference

Apart from model information and data information, Bayesian inference also utilizes the distribution information of unknown parameters (Markvardsen 2004). This kind of information existed prior to the experiment, and is expressed with the probability distribution of unknown parameters, so it is generally called "prior".

The general model is: prior + sample information => posterior

Bayesian theorem aims to use known information to construct the posterior probability density of system status variances, which means utilizing the model to predict the prior estimated density of the status, and then using the latest observation information to rectify and thus get probability density. Using observation information to calculate status variances, we can trust in the accuracy of different values, and receive the best estimation of the model (Chamberlain, Imbens 2003; Seng 2008). The Bayesian inference commonly used in probability reasoning (Mahdavi Adeli *et al.* 2011) and engineering is also often used in reliability analysis (Der Kiureghian 2008; Maes 2007) and Bayesian networks (Perelman, Ostfeld 2012).

1.3. Particle Swarm Optimization algorithm (PSO)

The Particle Swarm Optimization (PSO) algorithm is a relatively new algorithm derived by Kennedy and Eberhart (1995) from a simplified social model simulation. PSO algorithms mimic mechanisms used by birds to share information in flight. The particle concept requires members in groups without mass and volume and with designated speed and acceleration. The first version of PSO added neighboring speed values and considered multi-dimensional search and distance-based acceleration. Inertia weight, introduced later, enhanced the algorithm's exploitation and exploration and paved the way to form a standard version of the algorithm (Clerc, Kennedy 2002). PSO is often applied in engineering to solve multi-objective decision-making (Azadnia, Zahraie 2010) and optimization (Li *et al.* 2010) tasks. In recent years, PSO has been increasingly associated with other AI tools to develop numerous new optimization methods (Yan, Zhang 2011; Zhao *et al.* 2006).

2. Evolutionary Gaussian process inference model

This model is founded on historical data and formed with Gaussian process, in combination with Particle Swarm Optimization (PSO) and Bayesian inference. In this model, GP is used to reveal the intricate relationship between variance input and output. Bayesian inference structure gives the posterior probability of the entire function, and serves as the reference for parameter optimization. PSO is used to search the best hyper-parameter GP and required Bayesian analysis; the structure is shown in Fig. 1. The model includes three parts.

A. Data input

Collecting and arranging input data $\boldsymbol{X}$ and data $\boldsymbol{Y}$, X is the collection of data input factor of N dimensions X_1, X_2, …, X_N; and $\boldsymbol{Y}$ is the collection of m pieces of desire Y_1, Y_2, …, Y_m. Thus, any Y_i is the reflection of the desire value of case input value $\{X_{1i}, X_{2i}, \ldots, X_{Ni}\}$ (Money *et al.* 2012).

The corresponding function value of any input factor X_j is $f(X_j)$: $F(\boldsymbol{X}) = \{f(X_1), f(X_2), \ldots, f(X_N)\}$; $F(\boldsymbol{X})$ is the function congregation to demonstrate the relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and here the Gaussian process is used to describe function distribution. Assuming function $F(\boldsymbol{X})$ coincides with Gaussian distribution, and to make the work easier, the expected value $m(X)$ is 0, the probability is shown as:

$$P\left(F\right)=\frac{1}{(2\pi)^{\frac{N}{2}}\left|K\right|^{\frac{1}{2}}}\exp\left[-\frac{1}{2}F^{T}K^{-1}F\right]\sim N(0,K),\ (8)$$

where: K is the matrix constructed from the covariance function $k = (X, X')$; and the equation above the probability of the set function F is regarded to be controlled by the covariance matrix K.

B. Gaussian process and Bayesian inference

(1) Covariance matrix and parameter.

After determining the stationary pattern, covariance function is chosen to construct the covariance matrix. The parameter model and quantity vary according to the differences of functions, and this study adopts the most common Squared Exponential covariance function.

$$k_{SE}\left(X_i,\ X_j\right)=\sigma_f^2\exp\left[-\frac{1}{2}\left(\frac{X_i-X_j}{r_i}\right)^2\right]+\sigma_n^2\delta_{ij},\ (9)$$

where: σ_f (signal variance) – controls the volatility of the entire function; σ_n (noise) – indicates the errors of the entire function; r_i (length-scale) – shows the relationship between variances X_i and X_j in function space; σ_f, σ_n, r_1, r_2, …, r_n represent the hyper-parameters in the matrix.

In this paper, we use θ to represent the aggregation of hyper-parameters (Fig. 1).

(2) Bayesian inference and posterior probability.

According to chosen covariance function, and utilizing Bayesian theorem, the posterior probability of the entire function $P\left(F|\boldsymbol{X},\boldsymbol{Y}\right)$ is inferred.

$$P\left(F\left|\boldsymbol{X},\boldsymbol{Y}\right.\right)=\frac{P(Y\left|F,\boldsymbol{X}\right.)P(F)}{P(\boldsymbol{Y}\left|\boldsymbol{X}\right.)}. \quad (10)$$

To maximize the posterior probability $P\left(F|\boldsymbol{X},\boldsymbol{Y}\right)$ minimizing the Negative Log-Marginal Likelihood (NLML) and combining PSO are approaches employed with the goal of having the most likely hyper-parameter during the minimization process.

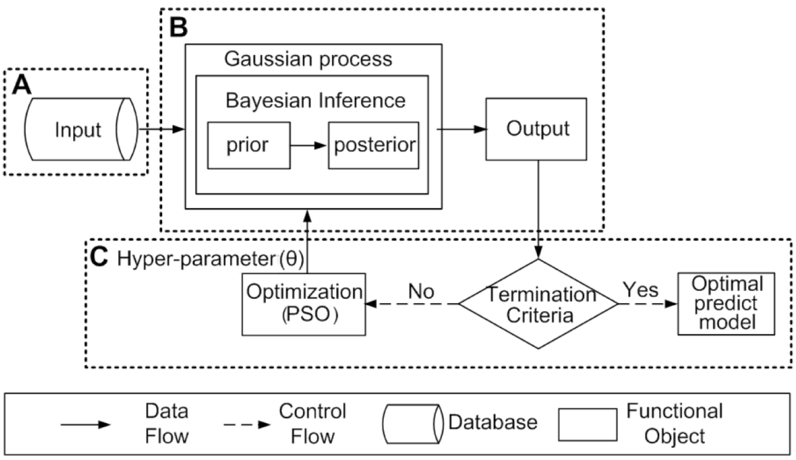


Fig. 1. EGPIM structure

C The optimization of hyper-parameter

PSO is applied to EGPIM to optimize the hyper-parameter in function space, and comprises the best function in the model

(1) Initial stage.

PSO parameter was set up, and the particle groups, particle speed and positions were then randomly started to initiate and proceed with iteration:

- group scale $\boldsymbol{m}$;
- maximum speed $V_{\max}$;
- acceleration constant c_1 and c_2;
- maximum inertia weight $W_{\max}$;
- minimum inertia weight $W_{\min}$;
- maximum iteration times $Iter_{\max}$;
- terminate accuracy requirement NLML (Negative Log Marginal Likelihood),

where: group scale $\boldsymbol{m}$ represents number of particles; $V_{\max}$ is the maximum particle velocity; c_1 and c_2 are acceleration constants that are also called learning factors. Usually, $c_1 = c_2 = 2$; $W_{\max}$ is the final inertia weight and $W_{\min}$ is initial inertia weight, used to calculate inertia weight; $Iter_{\max}$ sets the maximum number of particle swarm optimization times; NLML is the fitness value of the PSO. In general, iterative termination is defined as when either the maximum number of iterative times and/or some minimum fitness value is reached.

(2) Optimization stage.

We used a fitness calculation of particles to discriminate between good and bad particles. The adaptation value depended on NLML. In practice, prior knowledge is insufficient to fix appropriate values for the hyper-parameters that define the covariance. We therefore gave prior distributions to the hyper-parameters and based predictions on a sample of values from their posterior distribution. Sampling from the posterior distribution requires computation of log likelihood based on the datasets, which is:

$$-\log \mathrm{P}\left(\boldsymbol{Y}\middle|\boldsymbol{X}\right) = \frac{1}{2}Y^T\left(K\left(X,X\right)+\sigma^2 I\right)^{-1}Y) + \frac{1}{2}\log\left|K\left(X,X\right)+\sigma^2 I\right| + \frac{N}{2}\log 2\pi \,. \tag{11}$$

The calculation of particle search speed and direction is conducted as follows:

Particle speed calculation:

$$V_{id}^{t+1} = w^{t+1}\times V_{id}^t + c_1\times rand()\times\left(pbest_{id} - S_{id}^t\right) + \\ +c_2\times rand\left(\right)\times\left(gbest_{id} - S_{id}^t\right). \tag{12}$$

Particle weight:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}}\times iter \,. \tag{13}$$

New search direction calculation:

$$S_{id}^{t+1} = S_{id}^t + V_{id}^{t+1}\,, \tag{14}$$

where: V_{id}^t is the velocity of particle i at iteration t in dimension d; V_{id}^{t+1} is the new updated particle velocity; position of $S_{id}^t\, i$ is the current location; S_{id}^{t+1} is the new updated particle location; $pbest_{id}$ is the optimization found by the particle itself, which are the extrema of body; $gbest_{id}$ is the optimization of the whole swarm, which is the global extrema; Rand ( ) are the random numbers within (0, 1); and c_1 and c_2 are called learning factors.

w is the weighting efficient, with a value between 0.1 to 0.9. Through constant learning and renewing of location and speed, particles gradually fly into the optimum location of space until the searching process ends. The final output, *gbest*, is the best optimization.

(3) Termination stage.

After a continuous search in function space, the best global solution is *gbest*. If the fitness value > global solution, then the search will continue. The conditions for search ending are:

- Coincides with the requirement accuracy (NLML);
- Reaches search $Iter_{\max}$.

Otherwise, the search is continued.

3. Prediction of project success using EGPIM

The EGPIM proposed herein adopts a proactive approach that utilizes time series data to predict a single ongoing project outcome at different stages of completion, given by percentages. The implementation process follows Roy's (2009) methods, as shown in Figure 2.

3.1. The implementation process

This seven-step process is divided into two parts, the first being steps 1 through to 6 and the second being step seven, which applies the EGPIM to make predictions on project success. The following details the method of each step:

(1) Assign project type as the project parameter.

Fifty four historical projects from the CAPP system database with diverse data characteristics were used for this study. The process project type was chosen as the project parameter for this study in order to gain a more complete understanding of the factors that influence projects. This type of project typically covers about 64% of project data in the CAPP database, with the best factors identified by CAPP for predictive ability.

(2) Identify influencing factors.

This study adopted the CAPP software's recommendation that the variable level of significance should be set below 0.10. This significance level represents the statistical difference between project outcomes and factors considered to have a predictive ability for project success. CAPP software analysed 76 factors from the project data set with 11 factors being identified as significant (as shown in Table 1).

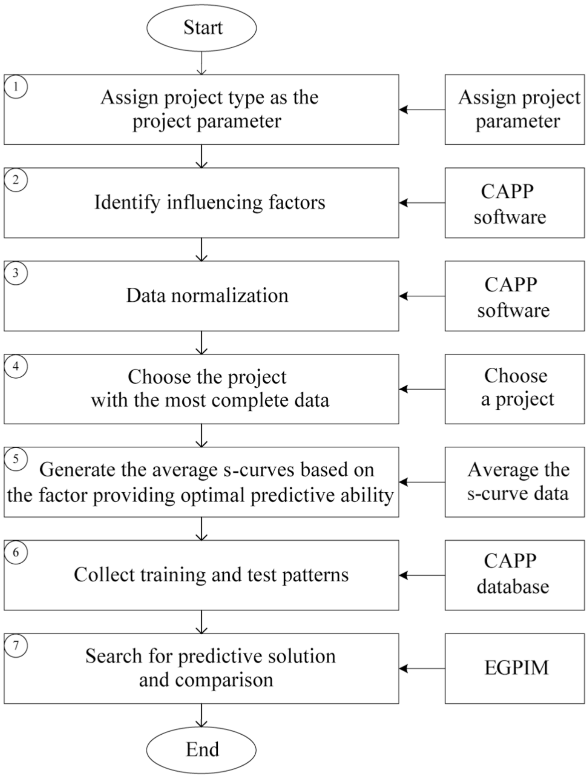


Fig. 2. Using EGPIM implementation steps to predict project success

(3) Data normalization.

Based on data analysis, CAPP normalized the project data from 0–100 percent completion into 30 reporting periods. It also identified that actual owner expenditure factors have the greatest impact on predicting project outcome. As per our study objectives, owner expenditure factors were chosen as the factor to be normalized for all process projects. Corresponding with 30 reporting periods, the normalized data for owner expenditures provided the basic data to generate s-curve graphs.

(4) Choose the project with the most complete data.

A proactive approach was used by this study to predict the outcome of a single ongoing project. To distinguish the project from other process projects in the database, only one project was chosen as the 'assessment project'. The study required the chosen project to have complete data for all 11 of the time-dependent factors for success identified by CAPP. Of the 34 process projects, Project 233 fulfilled these requirements.

(5) Generate the average s-curves based on the factors to gain optimal predictive ability.

There are four project outcome categories in the CAPP system, namely "successful", "on time or on budget", "less than successful", and "disastrous". All project outcomes were recorded within the CAPP database upon project completion. The outcomes of the projects that were examined in this study are listed in Table 2. Average s-curves were then generated based on these four project outcomes using generated normalized data. Since the three projects in the 'disastrous' category did not have data on actual owner expenditure factors, we were unable to plot an average s-curve for this category. Four different zones representing each of the project outcome ranges were then created proportionally within those three average s-curve lines (Fig. 3). As an example, zone 0.667 (for on time or on budget) was formed by two limit lines (upper and lower). For the lower limit, the line can be drawn based on average values for the actual owner expenditure percentage between the average of all successful projects and the average of all on-time or on-budget projects. The same approach also applies to the upper-limit line, as well as to the rest of the limit lines. This zone apportionment may later be used to determine the project outcome degree as it relates to the assessment of ongoing projects at every completion interval up until total project completion.

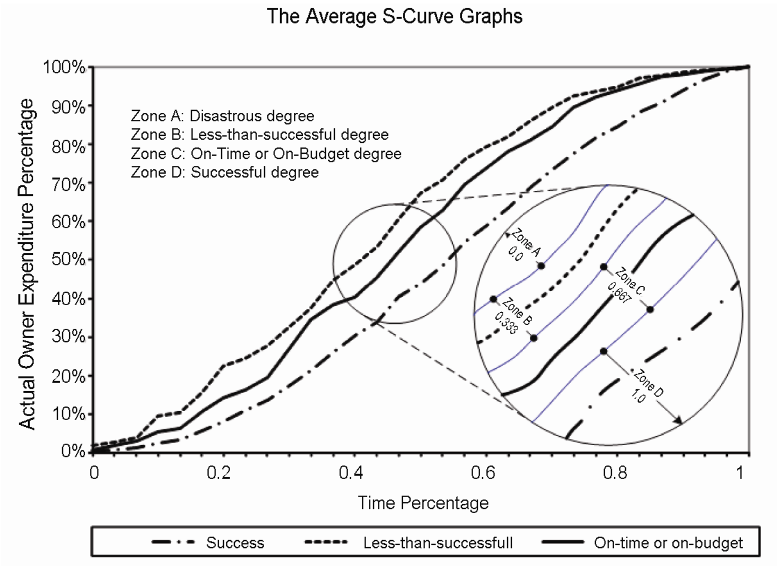
(6) Collect training and testing patterns.

Each of the 11 factors identified by CAPP software as significant was employed as input patterns. Output data was derived from the project outcome at every completion interval that tracks along the zone path of the average s-curve graphs for Project 233. To replicate a proactive approach, three different sets of training patterns were collected at 50%, 67%, and 90% completions, with the two adjacent completion percentage data increments for every training pattern data set used as testing data. In Table 3, testing data extracted for the 50% completion training pattern were at 53% and 57% completion.

Table 1. Description of 11 time-dependent factors with levels of significance

No	Factors	Column I.D. in CAPP	Significance level
1	Actual design % complete	C5_16	0.01
2	Actual owner expenditure	C3_10	0.01
3	Invoiced construction costs	C2_14	0.02
4	Designer planned effort hours	C2_13	0.01
5	Actual invoices for material and equipment	C3_28	0.01
6	Paid construction costs	C3_14	0.01
7	Cost of owner project commitments	C2_24	0.01
8	Recordable incident rate (by period)	C2_38	0.01
9	Cost of change orders	C2_17	0.02
10	Quantity of change orders	C3_17	0.01
11	Actual overtime work	C3_41	0.02

(Cheng *et al*. 2010)



(Roy 2009)

Fig. 3. The average s-curve graphs for actual owner expenditures and zone apportionment of degree of project outcome

Table 2. The four quantitative values associated with project outcomes

Degree of project outcome	Value
Successful	1
On time or on budget	0.6667
Less-than-successful	0.3333
Disastrous	0

Similar arrangements were applied to the 67% and 90% completions.

(7) Search for predictive solution and comparison.

The proposed AI system, EGPIM, was applied to predict project outcome based on factors identified in the three different learning sets (i.e. 50%, 67%, and 90% completion). The performance of the proposed system was evaluated using RMSE and an average error percentage.

3.2. Results

In order to highlight the potential and effectiveness of the proposed system, EGPIM was compared against Evolutionary Fuzzy Support Vector Machine Inference Model (ESFIM), support vector machines (SVM) and against the original Gaussian process (GP). In this study, as suggested parameter settings for SVMs by (Hsu, Lin 2002) and the GP were established by conjugate gradients to find good hyper-parameter settings. Table 4 shows the average RMSEs achieved by EGPIM, SVMs, and GP. The accuracy obtained by EGPIM was significantly better than that obtained by either SVM or GP; Although EFSIM obtained slightly better results at the 50% and 67% completion stages, EGPIM earned significantly better results than EFSIM at the 90% completion stage. Table 5 shows a detailed error percentage for the three percentage completions.

Conclusion

This paper presented an implementation of an EGPIM to predict a project outcome path and to determine the likely project outcome based on identified time-dependent factors. CII's proprietary CAPP software and database were employed to extract time-dependent factors identified to be significantly associated with predicting a project's outcome.

This study used historical case studies to examine EGPIM's ability to predict a project's outcome. The results showed that EGPIM has an excellent predictive capability. EGPIM's performance was also demonstrated to be better than both SVMs and the GP in practical applications.

These results highlight its suitability for construction projects, as well as displaying its potential benefits to project managers. Since decisions must be made for many events throughout a construction project, project managers can use our model to compile the data and use its predictions as a reference to help them make such important and complex decisions.

This model holds great potential as a predictive tool when used proactively to assess project outcome, giving project managers a better chance to take actions necessary to ensure projects are accomplished successfully.

Acknowledgement

The authors would like to thank the Construction Industry Institute for their kind permission to use, analyse, extract, and publish data from their CAPP database.

References

Azadnia, A.; Zahraie, B. 2010. Application of multi-objective particle swarm optimization in operation management of reservoirs with sedimentation problems, in *Providence, RI,* 2260–2268. May 16–20, 2010.

Bonilla, E. V.; Chai, K. M. A.; Williams, C. K. I. 2009. Multi-task Gaussian process prediction, in *21st Annual Conference on Neural Information Processing Systems, NIPS 2007*, December 3–6, 2007, Vancouver, BC, Canada.

Brahim-Belhouari, S.; Bermak, A. 2004. Gaussian process for nonstationary time series prediction, *Computational Statistics and Data Analysis* 47(4): 705–712. http://dx.doi.org/10.1016/j.csda.2004.02.006

Table 3. Example learning and testing data applied to 50% completion data for Project 233 (11 time-dependent variables and 1 output)

	% Completion	Quantitative project status per period	Input patterns										
			1	2	3	4	5	6	7	8	9	10	11
		(Output)	C5_16	C3_10	C2_14	C2_13	C3_28	C3_14	C2_24	C2_38	C2_17	C3_17	C3_41
Training set	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0.0714	0.0579	0	0.0824	0	0	0	0	0	0	0
	7	0	0.1224	0.0831	0	0.1418	0	0	0	0	0	0	0
	10	0	0.1837	0.0864	0	0.2172	0	0	0	0	0	0	0
	…	…	…	…	…	…	…	…	…	…	…	…	…
	37	0.6667	0.8061	0.2872	0.0526	0.8396	0.0478	0.0345	0	0	0.2977	0.339	0
	40	0.3333	0.9082	0.353	0.0897	0.8846	0.0954	0.0345	0	0	0.3065	0.3559	0
	43	0.6667	0.949	0.3923	0.1629	0.9224	0.2108	0.0897	0	0	0.312	0.3729	0
	47	0.6667	0.9694	0.4293	0.1943	0.9843	0.2119	0.0897	0	0	0.7582	0.7458	0
	50	0.6667	0.9745	0.4598	0.2358	0.9879	0.2564	0.1263	0	0.5	0.7582	0.7458	0
Testing set	53	0.66667	0.9796	0.4903	0.2772	0.9915	0.3008	0.1629	0	1	0.7582	0.7458	0
	57	0.66667	0.9898	0.5593	0.3991	0.9957	0.4388	0.1943	0	1	0.7711	0.7627	0

Notes: Quantitative project statuses are assigned based on the four project outcome categories of: successful (1), on-time or on-budget (0.6667), less-than-successful (0.3333), and disastrous (0).

Table 4. RMSE and average error percentage comparisons between EGPIM, SVMs, and GP

		% Completion											
		50%				67%				90%			
		1	2	3	4	1	2	3	4	1	2	3	4
RMSE		0.0121	0.0081	0.1083	0.0687	0.0047	0.0039	0.3874	0.1402	1.86E-09	0.0001	0.016	0.0229
Average Error (%)		1.21	0.8	10.81	6.84	6.56	0.3	38.74	13.83	1.87e-07	0.01%	1.56	2.29
C parameter		–	31	1	–	–	31	1	–	–	31	1	–
g parameter		–	0.0109	0.0909	–	–	0.5739	0.0909	–	–	0.0734	0.0909	–
Hyper-parameter	σ_f:	0.5778			0.987	2.2979			1.1025	1.8150			0.8386
	r_1:	1.3205			0.3126	2.2110			0.8124	1.6023			0.8848
	r_2:	2.5328			0.7454	1.8518			1.4358	7.0094			0.484
	r_3:	1.6658			0.372	2.1463			1.0954	6.2906			0.8107
	r_4:	2.8168			1.0099	1.3067			1.5725	1.3197			0.3407
	r_5:	2.7209			0.708	1.8320			0.9787	8.0565			0.9121
	r_6:	4.0550		–	0.1338	1.0766		–	0.2114	3.7408		–	0.5418
	r_7:	1.8488			0.4473	0.6040			0.2721	2.2483			0.0299
	r_8:	3.6237			1.0095	5.3495			2.769	1.5403			1.7214
	r_9:	1.7727			0.7697	1.9695			0.7211	3.0087			0.258
	r_{10}:	0.4882			0.1729	3.4762			0.2961	6.4939			0.3811
	r_{11}:	1.5168			0.6029	2.2946			0.9199	0.9605			0.9836
	σ_n:	2.3069			2.1071	1.1164			2.4139	11.5585			2.26

Notes: 1. EGPIM; 2. EFSIM (Quadratic time function); 3.SVM; 4. GP.

Table 5. Detailed error percentage comparisons for 50%, 67%, and 90% completion

% Completion	Predicted % Completion	Desire	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Error Percentage* 1	Error Percentage* 2	Error Percentage* 3	Error Percentage* 4
50%	53%	0.6667	0.6562	0.6595	0.5522	0.6055	1.05	0.72	11.45	6.12
	57%	0.6667	0.6531	0.65773	0.5650	0.5911	1.36	0.90	10.17	7.56
					Average Error %		1.21	0.81	10.81	6.84
67%	70%	0	0.0654	0.0034	0.3938	0.1149	6.54	0.34	39.38	11.49
	73%	0	0.0658	0.0043	0.3809	0.1616	6.58	0.43	38.09	16.16
					Average Error %		6.56	0.39	38.74	13.83
90%	93%	0	1.81E-09	0.0001	0.0189	0.0218	1.81E-07	0.01	1.89	2.18
	97%	0	1.92E-09	0.0001	0.0123	0.024	1.92E-07	0.01	1.23	2.4
					Average Error %		1.87E-07	0.01	1.56	2.29

*Note: Error Percentage = $\left|\text{Predicted} - \text{Desire}\right| \times 00\%$

1 EGPIM

2 EFSIM with quadratic time function

3 SVM

4 GP

Chamberlain, G.; Imbens, G. W. 2003. Nonparametric applications of Bayesian inference, *Journal of Business & Economic Statistics* 21(1): 12–18. http://dx.doi.org/10.1198/073500102288618711

Chan, A. P. C.; Scott, D.; Chan, A. P. L. 2004. Factors affecting the success of a construction project, *Journal of Construction Engineering and Management* 130(1): 153–155. http://dx.doi.org/10.1061/(ASCE)0733-9364 (2004) 130:1(153)

Cheng, M.-Y.; Wu, Y.-W.; Wu, C.-F. 2010. Project success prediction using an evolutionary support vector machine inference model, *Automation in Construction* 19(3): 302–307. http://dx.doi.org/10.1016/j.autcon.2009.12.003

Chu, W.; Ghahramani, Z. 2005. Gaussian processes for ordinal regression, *Journal of Machine Learning Research* 6: 1019–1041.

Clerc, M.; Kennedy, J. 2002. The particle swarm – explosion, stability, and convergence in a multidimensional complex space, *Evolutionary Computation, IEEE Transactions* 6(1): 58–73.

Der Kiureghian, A. 2008. Analysis of structural reliability under parameter uncertainties, *Probabilistic Engineering Mechanics* 23(4): 351–358. http://dx.doi.org/10.1016/j.probengmech.2007.10.011

Griffith, A. F.; Gibson, G. E. Jr; Hamilton, M. R.; Tortora A. L.; Wilson, C. T. 1999. Project success index for capital facility construction projects, *Journal of Performance of Constructed Facilities* 13(1): 39–45. http://dx.doi.org/10.1061/(ASCE)0887-3828(1999)13:1(39)

Hsu, C. W.; Lin, C. J. 2002. A simple decomposition method for support vector machines, *Machine Learning* 46(1–3): 291–314. http://dx.doi.org/10.1023/A:1012427100071

Kennedy, J.; Eberhart, R. 1995. Particle swarm optimization, in *Perth, Aust, 1942–1948: IEEE.*

Khosravi, S.; Afshari, H. 2011. A success measurement model for construction projects, in *International Conference on Financial Management and Economics Singapore*, IACSIT Press.

Ko, Ch.-H.; Cheng, M.-Y. 2007. Dynamic prediction of project success using artificial intelligence, J*ournal of Construction Engineering and Management* 133(4): 316–324. http://dx.doi.org/10.1061/(ASCE)0733-9364(2007) 133:4(316)

Kocijan, J.; Murray-Smith, R.; Rasmussen, C. E.; Girard, A. 2004. Gaussian process model based predictive control, in *Proceedings of the 2004 American Control Conference (AAC)*, June 30–July 2, 2004, Boston, MA, United states, 2214–2219.

Li, T.; Fu, Q.; Meng, F. 2010. Research on partial least-squares regression model based on particle swarm optimization and its application, in *2nd International Workshop on Intelligent Systems and Applications (ISA),* May 22–23, 2010, Wuhan, 1–4. http://dx.doi.org/10.1109/IWISA.2010.5473428

Maes, M. A. 2006, Exchangeable condition states and Bayesian reliability updating, keynote address, in *Proceedings, 13th IFIP WG7.5 Working Conference on Reliability and Optimization of Structural Systems*, October, Kobe, Japan, Taylor and Francis, 27–42.

Mahdavi Adeli, M.; Deylami, A.; Banazadeh, M.; Alinia, M. M. 2011. A Bayesian approach to construction of probabilistic seismic demand models for steel moment-resisting frames, *Scientia Iranica* 18(4 A): 885–894.

Markvardsen, A. J. 2004. Bayesian probability theory applied to the space group problem in powder diffraction, in *AIP Conference Proceedings* 735(1): 219–226. http://dx.doi.org/10.1063/1.1835216

Money, E. S.; Reckhow, K. H.; Wiesner, M. R. 2012. The use of Bayesian networks for nanoparticle risk forecasting: model formulation and baseline evaluation, *Science of the Total Environment* 426: 436–445. http://dx.doi.org/10.1016/j.scitotenv.2012.03.064

Perelman, L., Ostfeld, A. 2012. Bayesian networks for estimating contaminant source and propagation in a water distribution system using cluster structure, in *Proceedings of the 12th International Conference, Water Distribution System Analysis 2010*, 426–435.

Roy, A. F. V. 2009. *Evolutionary fuzzy decision model for construction management using weighted support vector machine*: PhD Thesis, Department of Construction Engineering, National Taiwan University of Science and Technology.

Russell, J. S.; Jaselskis, E. J.; Lawrence S, P. 1997. Continuous assessment of project performance, *Journal of Construction Engineering and Management* 123(1): 64–71. http://dx.doi.org/10.1061/(ASCE)0733-9364(1997) 123:1(64)

Sanvido, V.; Grobler, F.; Parfitt, K.; Guvenis, M.; Coyle, M. 1992. Critical success factors for construction projects, *Journal of Construction Engineering and Management* 118(1): 94–111. http://dx.doi.org/10.1061/(ASCE)0733-9364(1992)118:1(94)

Seng, K. N. K.. 2008. *Non-linear dynamics identification using Gaussian process prior models within a Bayesian context*: PhD Thesis, Department of Electronic Engineering, National University of Ireland Maynooth.

Su, G. S.; Xiao, Y. L. 2011. Gaussian process method for slope reliability analysis, *Yantu Gongcheng Xuebao*/*Chinese Journal of Geotechnical Engineering* 33(6): 916–920.

Yan, K.-Z.; Zhang, Z. 2011. Research in analysis of asphalt pavement performance evaluation based on PSO-SVM, in *2011 International Conference on Civil Engineering and Transportation, ICCET 2011*, October 14–16, 2011, Jinan, China, 203–207.

Yan, Z.; Su, G.; Yan, L. 2011. Classification of surrounding rocks in tunnel based on Gaussian process machine learning, in *2011 International Conference on Electric Technology and Civil Engineering (ICETCE),* April 22–24, 2011, Lushan, 3971–3974.

Zhao, F.; Zhang, Q.; Yang, Y. 2006. A scheduling holon modeling method with petri net and its optimization with a novel PSO-GA algorithm, in *2006 10th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2006*, May 3–5, 2006, Nanjing, China, 1302–1307.

Min-Yuan CHENG. Professor of Construction Engineering, at the National Taiwan University of Science and Technology, Taiwan. His research interests include geographic information system, construction automation, management information system, and construction management process reengineering.

Chin-Chi HUANG. PhD, candidate of Construction Engineering, at the National Taiwan University of Science and Technology, Taiwan. Her research interests include construction management, project management and decision analysis.

Andreas Franskie Van ROY. Professor of Civil Engineering, at the Parahyangan Catholic University, Indonesia. His research interests include construction automation, management information system, and applications of artificial intelligence.