CITS5502 Software Processes

Semester 2, 2019

Assignment 2

Nicole Low

21151969

Table of Contents

Introduction

It is universally understood that most processes become more efficient as the people completing said processes acquire more knowledge about the process. This concept can be applied to software development; as an individual learns more about a problem and programming language, the more efficient they become at solving subsequent similar problems. Many models have been created attempting to model this ‘learning curve’, such that with reasonable data on similar prior projects, predictions for effort required could be made on new projects. This is also known as knowledge acquisition, and knowledge encoding, and there are many different techniques such as Protocol-generation techniques, Protocol analysis techniques, Hierarchy-generation techniques, Matrix-based techniques, Sorting techniques and Diagram based techniques. Fitting models to represent effort required in encoding a program are explored in the context of data collected from CITS8820 students. Using this data set, two models of ‘learning’ and the change in effort from this will be examined.

Source Data

	Problem 1 Language A				**Problem 2 Language A**				**Problem 1 Language B**			
t	0	1	2	3	0	1	2	3	0	1	2	3
P1	270	170	117	114	50	59	45	40	105	80	60	60
P2	55	20	19	18	83	15	13	13	115	35	29	41
P3	205	165	58	64	150	115	107	102	159	113	87	82
P4	169	82	47	27	73	38	91	58	106	22	28	28
P5	150	75	63	45	75	78	27	26	73	135	28	66
P6	210	71	63	26	60	31	21	18	38	32	71	24
P7	212	253	54	47	46	34	25	17	73	32	26	19
μ	181.571	119.429	60.1429	48.7143	76.7143	52.8571	47	39.1429	95.5714	64.1429	47	45.7143

Table 1: CITS8220 data, seven random students selected

Effort Models

The four models proposed are functions of time (*t*), and they predict effort (in person-minutes), and have parameters *a, b* and *c* which need to be fit. They are as follows:

$$(A)\ Effort = \frac{a + bct}{bt + 1}$$

$$(B)\ Effort = (a - c)\,(t + 1)^{-b} + c$$

$$(C)\ Effort = (a - c)e^{-bt} + c$$

$$(D)\ Effort = a + bt + ct^2$$

Each model has a slightly different curve to represent the combination of criteria;

- Model A represents these criteria with the positive part of a hyperbolic equation.
- Model B also represents these criteria with a hyperbolic equation, but with a greater effect of *b*.
- Model C represents these criteria with the basic exponential equation $y = ae^{bx}$, except it is negatively exponential and includes the *c* parameter.
- Model D represents these criteria with a quadratic parabolic equation.

Model Suitability and Interpretation

These effort models are mapping a "learning curve", where there should realistically be a reduction in effort as *t* increases. The model must therefore possess certain criteria;

- The model must show reduced effort as *t* increases
- The model must show that all tasks have a non-zero minimum effort (parameter *c*)
- The model must show that initial knowledge of a task affects the initial effort required, where the curve intercepts the *y*-axis at $t = 0$ (parameter *a*)
- The model must reflect the learning rate of individuals (parameter *b*)

Models A, B and C reflect these criteria. All three models show an effort reduction and include all three parameters to appropriately model a learning curve. Model D, however, does not show an effort reduction as it is parabolic. As it is parabolic, the model will show an increased effort after *t* reaches a certain point. Model D would thus be unsuited for modelling effort.

For the purposes of this report, Models A and B were chosen to be fit against the selected CITS8220 data.

Model Fitting

Seven students' results from the CITS8220 dataset were chosen randomly for model fitting. For each of the three problem sets from the data (Problem 1 Language A (P1LA), Problem 2 Language A (P2LA), and Problem 1 Language B (P1LB)), the four datapoints of each of the chosen students were averaged to produce three sets of four datapoints (see Table 1). These datapoints were fitted to Models A and B, using a non-linear least-squares regression method in Python using the SciPi package (See Appendix: Figures 1-6). The parameter values for each model fit were then entered into Microsoft Excel, where the models were graphed and R^2 values were generated. This R^2 value can be used as an indication for the quality of the models fit.

Both models fit very well to the data as shown by the high R^2 value (see Table 2). The best fitting model was Model B on P2LA, with an R^2 of 0.992. Overall, Model A held a higher average R^2 value.

	Model		
	A	**B**	μ
P1LA	0.98277101	0.98072393	0.98174747
P2LA	0.99197125	0.99241287	0.99219206
P1LB	0.9897074	0.98927626	0.98949183
μ	0.98814989	0.98747102	

Table 2: CITS8220 R^2 values

Model Discussion

	Problem				
Model	**P1LA**	**P2LA**	**P1LB**	% Change 1	% Change 2
A	182.834378	76.6132113	95.7225885	-58.096933	-47.645191
B	182.910889	76.5772906	95.7817802	-58.1341	-47.63473
μ	182.872633	76.5952509	95.7521843	-58.115516	-47.639961

% Change 1 = (P2LA-P1LA)/P1LA x 100
% Change 2 = (P1LB-P1LA)/P1LA x 100

Table 3: CITS8220 a parameter comparison

Prior Learning, Similar Type and Different Language

Using the model fitted data from CITS8220, the extent that individuals carry learning from previous projects over to another project of a similar type, but in a different language can be explored. Within the CITS8220 data, P1LA and P1LB provide a point of comparison. Parameter *a* indicates knowledge encoding and will be used to investigate. On average, there was a reduction of initial effort (parameter *a*) of 47.6% when switching from programming language A to programming language B on problem 1 (see Table 3, % Change 2). This reduction in effort is significant, representing almost a half reduction in effort, which indicates learning is carried over to similar projects regardless of language used.

Prior Learning, Different Project and Same Language

Similar to the section above, using the CITS8220 data the extent that individuals carry learning from previous projects over to solving another project in the same language can be explored. Within the data, P1LA and P2LA provide a comparison. Parameter *a* shows the initial effort required in projects, and will be used to investigate. On average, there was a reduction of initial effort (parameter *a*) of 58.1% when switching between the first and second problems (see Table 3, % Change 1). This shows that there is over half the initial effort is required on the second project, indicating learning is carried over to subsequent projects of the same language.

Effect of Practice on Estimation

The CITS8820 data is not large enough to make strong predictions about the effect of practice on estimation. It is additionally hard to draw conclusions as the idea of estimation is not isolated in the data. As problem familiarity increases, actual data points are decreasing. To properly determine the extent which practice enables prediction of effort, experimental variables would need to be created and controlled to record this.

Learning Patterns

	Problem 1 Language A			Problem 2 Language A			Problem 1 Language B		
t	0	3	% change	0	3	% change	0	3	% change
P1	270	114	-57.77778	50	40	-20	105	60	-42.85714
P2	55	18	-67.27273	83	13	-84.33735	115	41	-64.34783
P3	205	64	-68.78049	150	102	-32	159	82	-48.42767
P4	169	27	-84.02367	73	58	-20.54795	106	28	-73.58491
P5	150	45	-70	75	26	-65.33333	73	66	-9.589041
P6	210	26	-87.61905	60	18	-70	38	24	-36.84211
P7	212	47	-77.83019	46	17	-63.04348	73	19	-73.9726
μ	181.57143	48.714286	-73.32913	76.71429	39.14286	-50.75173	95.571429	45.714286	-49.9459

Table 4: % Change ((*t* =3) - (*t* =1)) in Effort (minutes)

The learning patterns of individuals are represented by parameter *b* in both models. Parameter *b* changes the rate of decrease of both models by a factor, which represents the difference in rate of learning between each individual. The downwards sloping 'learning curve' in all models shows this reduction in effort as *t* increases, which suggests an increase in knowledge allows for a faster completion time, and thus effort. Theoretically, with knowledge acquisition each subsequent program should be completed quicker. Additionally, as seen in Table 4, the percentage in effort is largely reduced for each program from *t = 0* to *t = 3*.

Minimum Time

Parameter *c* represents a 'limiting factor' which signifies that even with prior knowledge a problem must have *Effort* > 0 when *t* = 0 (i.e. parameter *c* must be non-zero). Parameter *c* is added to each model, shifting it by its specific amount. In the CITS8220 data as shown in Table 5, we can examine the *c* values for each of the problems to determine minimum time. For P2LA and P1LB, this minimum time is around 22 minutes. For P1LA, the model fit did not produce a *c* parameter of understandable size.

	Problem		
Model	**P1LA**	**P2LA**	**P1LB**
A	-91.235489	24.9342335	23.3126331
B	-2017.4958	19.0151801	20.9789065
μ	-1054.3656	21.9747068	22.1457698

Table 5: Parameter *c* comparison

Limitations

The goal of model fitting is to generalise patterns in data so that the fitted curve can correctly predict new data that has not been presented to the model (i.e. generalisation). In this process, a small CITS8820 data set was presented, where results were requested to be averaged into 12 points of data. This small data set, with no clear training and test data is arguably not fit to accurately estimate new data. The size of this data may discern issues relating to generalisation, overfitting and data imbalance.

The CITS8820 data is being used to examine the idea of a 'learning curve', or knowledge acquisition and how it affects subsequent programming problems. What is not considered is the complexity of each programs requirements and how that may affect the effort required. An additional metric, such as one that represents the complexity of code and how difficult the program was to encode. Comparing different programs of different complexities to measure effort is not reasonable for accurate prediction without accounting for these differences.

There is additionally some ambiguity surrounding the data, such as the accuracy of the recorded timings for program encoding. Do the times reported by different individuals all include the same processes of software development, such as development, writing, test and development phases? Using this data for real estimation would require these questions to be addressed to maintain data accuracy.

Conclusion

Two effort models were fitted to examine how they could model real-world data, and if any inferences could be derived from the parameters that were produced from those models. Models A and B were selected as the two to be fit, as Model D was unsuitable. Parameters *a, b* and *c* that were generated from the CITS8820 data and were analysed to show changes between each problem and model. These parameters provided insights into how knowledge acquisition affects effort required to develop a software program. These models were fit to the data using a code written in Python, with the parameter results transferred to Microsoft Excel in order to visually plot the results. Both Models A and B were good fits to the data according to the R^2 values calculated, with Model B being slightly higher. However, due to the small data set and untrained models, these values should be interpreted with caution. This modelling provides an entry to understanding the relationship between knowledge acquisition and effort, however a larger data set and additional variables would be needed in order to make a more reliable prediction model.

Appendix

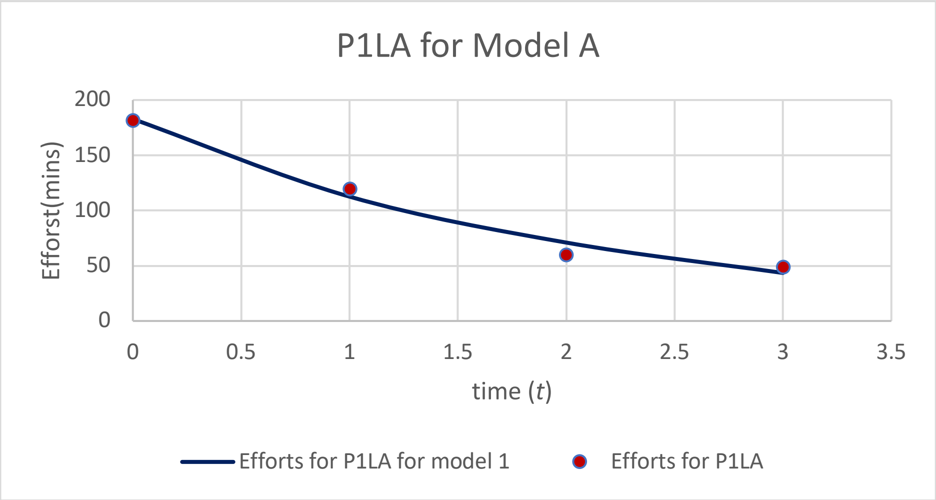

Figure 1: P1LA fit to Model A

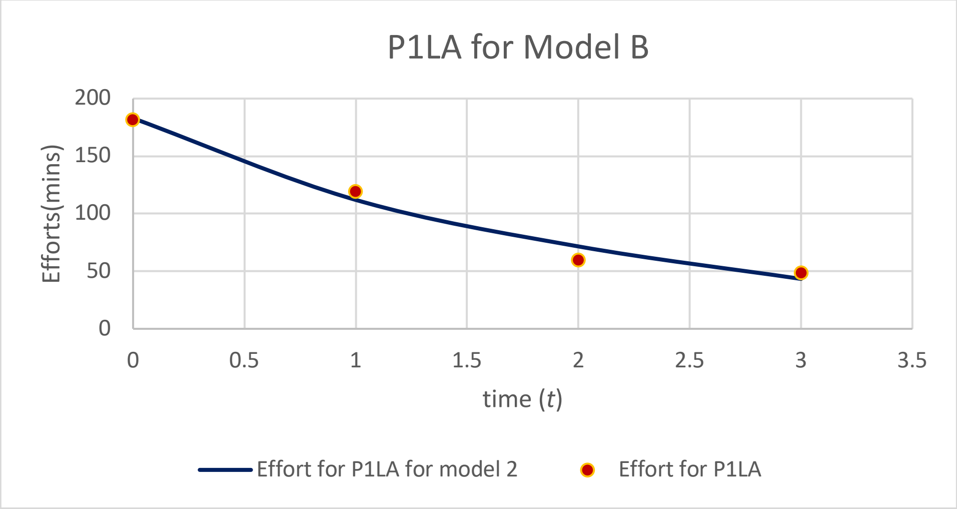

Figure 2: P1LA fit to Model B

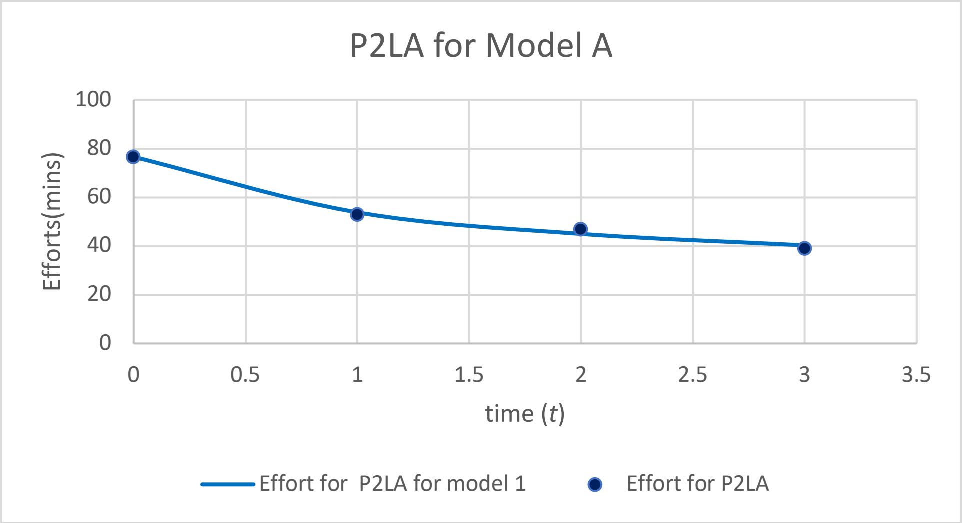

Figure 3: P2LA fit to Model A

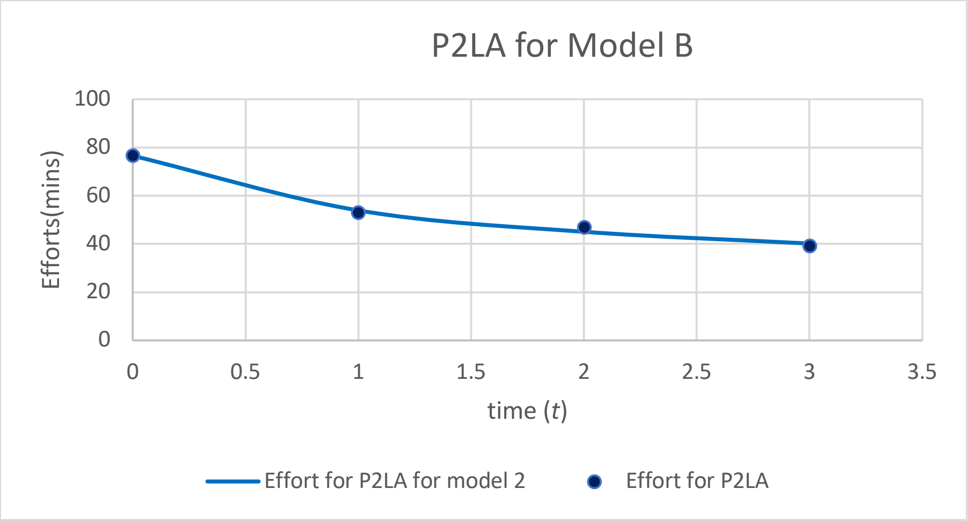

Figure 4: P2LA fit to Model B

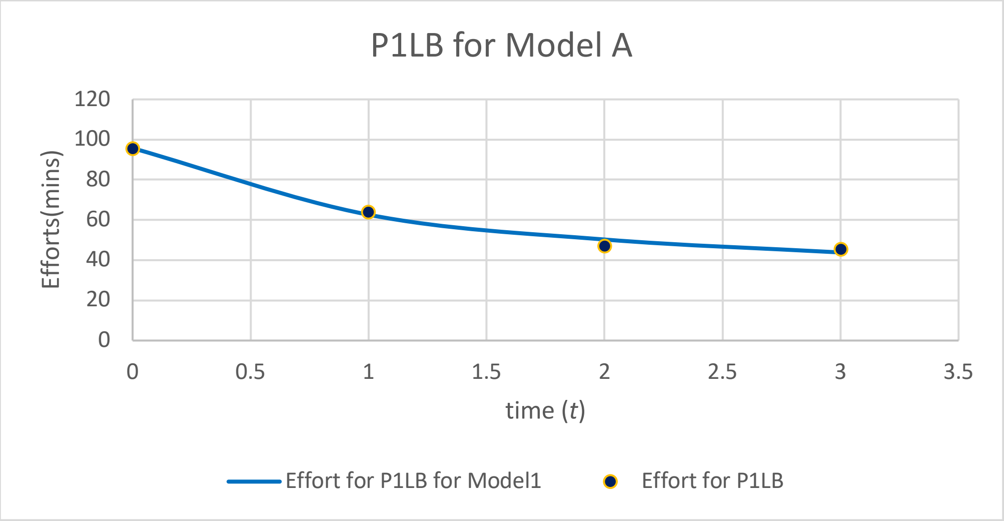

Figure 5: P1LB fit to Model A

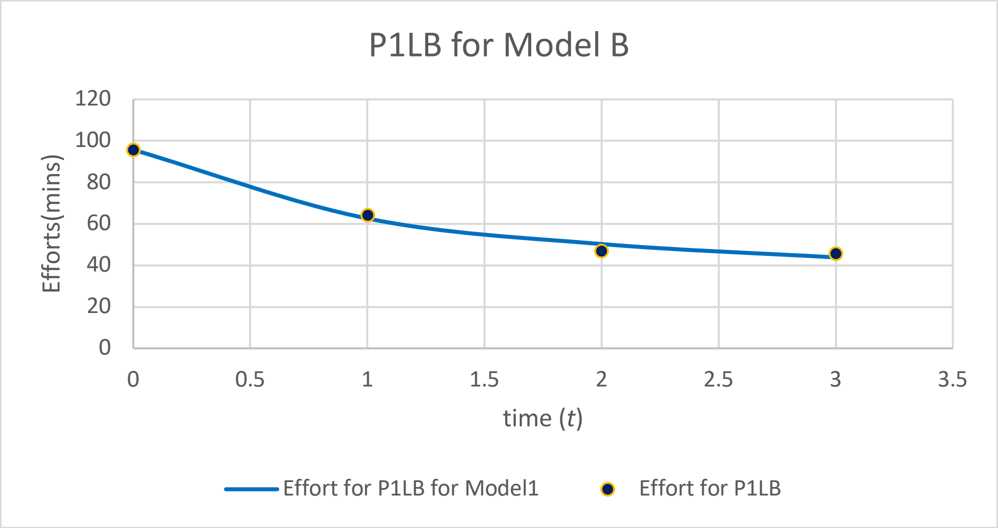

Figure 6: P1LB fit to Model B