

# The Effects of Duration-Based Moving Windows with Estimation by Analogy

Sousuke Amasaki<sup>1</sup>(✉) and Chris Lokan<sup>2</sup>

<sup>1</sup> Okayama Prefectural University, Soja, Okayama, Japan  
amasaki@cse.oka-pu.ac.jp

<sup>2</sup> School of Engineering and Information Technology, UNSW Canberra,  
Canberra, ACT 2600, Australia  
c.lokan@adfa.edu.au

**Abstract. Context:** Recent studies have revealed that estimation accuracy can be affected by only using a window of recent projects as training data for building an effort estimation model. The studies also showed that the effect and its extent could be affected by effort estimation methods and windowing policies (fixed size or fixed duration). However, a study of perhaps the most common situation — using Estimation by Analogy (EbA) for effort estimation, and only considering as training data projects completed recently in windows defined by duration — is lacking.

**Objective:** To investigate the effects on estimation accuracy of using the fixed-duration windowing policy, particularly in comparison to fixed-size windows, when using EbA.

**Method:** Using a single-company ISBSG data set studied previously in similar research, we examine the effects of using a fixed-duration windowing policy on the accuracy of estimates using EbA. As a preliminary step, we evaluate the effect of some changes to how we apply EbA itself.

**Results:** Fixed-duration windows can improve the accuracy of estimates with EbA. Some window sizes lead to statistically significant improvements. Reinforcing previous research, the effect is smaller and is seen in a narrower range of window sizes than when fixed-size windows are used.

**Conclusions:** Fixed-duration windows are helpful with this data set when using EbA. Variations in the settings for EbA can change the sizes at which windows are helpful. This suggests the need for reviewing optimal window sizes when adopting a new setting of EbA.

## 1 Introduction

Accurate effort estimation is an essential key to software project success. Many studies have sought to improve the accuracy of methods and models for estimating software development effort.

A software effort estimation model is developed from past project data. Most studies evaluate the accuracy of software effort estimation models using cross-validation. This approach uses data from all other projects to estimate the effort of a given project. For all but the last project, this means that data from projects that are still in the future are used when estimating the effort for the new project.

This makes no sense. Another evaluation approach exploits the reality that software projects can be arranged chronologically. It uses data from past projects as training data to predict new projects. Intuitively, it may also make sense to use only recent projects as a basis for effort estimation: older projects might be less representative of an organization’s current practices. Lokan and Mendes [1] examined whether using only recent projects improves estimation accuracy. They used a window to limit the size of training data so that an effort estimation model used only recently finished projects. As projects finish, they replace old projects in the window. The results supported the advantage of the windowing approach.

Recent studies also showed the effect and its extent could be affected by windowing policies [2, 3] and effort estimation models [4]. Lokan and Mendes [2, 3] compared two types of window policies: fixed-size and fixed-duration. A fixed-size window policy determines the window size by the number of projects: the training set is the last  $N$  projects to finish before the target project starts. A fixed-duration policy determines the window size by calendar months: the training set is projects whose whole life cycle occurred during the last  $w$  months before the target project starts. They found that estimation accuracy could improve by using either window policy, but the policies affected the accuracy differently.

Amasaki and Lokan [4] examined the applicability of the windowing approach (using fixed-size windows) to Estimation by Analogy (EbA). The previous studies only used linear regression (LR), EbA and LR are both common approaches for estimating effort. The results showed difference in accuracy between using and not using the windowing approach. However, the effect of using a window was weaker with EbA than with LR.

This paper continues research into the use of windows with EbA. It focuses primarily on the effect of changing the windowing policy, from fixed size to fixed duration. This is relevant because arguably fixed-duration windows make more intuitive sense than fixed-size windows. In practice, we believe that people considering “windows of recent projects” think naturally in terms of window duration, not the number of training projects in the window. The use of windows of different durations with EbA has not previously been studied, but we believe it is commonly in estimators’ minds.

First we must investigate the effect of changing some details of how we apply EbA in this paper, to improve its realism compared to [4].

We address the following questions:

- RQ1. Is there a difference in the accuracy of estimates between EbA as used in [4] and EbA based on a more realistic situation, still using fixed-size windows?
- RQ2. Is there a difference in the accuracy of estimates with and without windows, using the revised EbA, and using fixed-duration windows?
- RQ3. How do these results compare with results based on fixed-size windows?

## 2 Related Work

Research in software effort estimation models has a long history. However, few studies evaluated software effort estimation models with consideration of the chronological order of projects.

Mendes and Lokan [5] compared estimates based on a growing portfolio with estimates based on leave-one-out cross-validation, using two different data sets. In both cases, the cross-validation estimates showed significantly superior accuracy. With cross-validation, all other projects in the data set — even some that were still in the future — are used as training data for a given project. Thus, estimates using cross-validation are based on unrealistic information. If estimates based on unrealistic information are significantly more accurate than estimates considering chronology (based on realistic information), the implication is that the apparent accuracy achieved when ignoring chronology does not reflect what an estimator would achieve in practice.

To the best of our knowledge, Kitchenham et al. [6] first mentioned the use of moving windows. As a result of an experiment, they argued that old projects should be removed from the data set as new ones came in so that the size of the dataset remained constant. MacDonell and Shepperd [7] investigated moving windows as part of a study of how well data from prior phases in a project could be used to estimate later phases. They found that accuracy was better when a moving window of the five most recent projects was used as training data, rather than using all completed projects as training data.

Lokan and Mendes [1] studied the use of moving windows with linear regression models (LR) and a single-company dataset from the ISBSG repository. Training sets were defined to be the  $N$  most recently completed projects. They found that the use of a window could affect accuracy significantly; predictive accuracy was better with larger windows; some window sizes were ‘sweet spots’. Later they also investigated the effect on accuracy when using moving windows of various durations to form training sets on which to base effort estimates [2, 3]. They showed that the use of windows based on duration can affect the accuracy of estimates, but to a lesser extent than windows based on a fixed number of projects.

Amasaki and Lokan [4] examined the applicability of the windowing approach to Estimation by Analogy (EbA) in addition to LR. They found ranges of window sizes for which it was significantly better to use a window, with both regression and estimation by analogy. The effect of using a window was stronger with regression. They focused on the effects of the fixed-size windowing approach and left as future work an investigation for the fixed-duration window approach.

This study builds on both [4] and [3]. It extends [4] by changing details of EbA to improve the realism in practical use. It also differs from [4] in using duration as the basis for defining window size. This study also extends [3] by adopting EbA instead of LR to explore the effects of moving windows.

### 3 Research Method

#### 3.1 Dataset Description

The data set used in this paper is the same one analyzed in [1–4]. This data set is sourced from Release 10 of the ISBSG Repository. Release 10 contains data for 4106 projects; however, not all projects provided the chronological data we needed (i.e. known duration and completion date, from which we could calculate start date), and those that did varied in data quality and definitions. To form a data set in which all projects provided the necessary data for size, effort and chronology, defined size and effort similarly, and had high quality data, we removed projects according to the following criteria:

- The projects are rated by ISBSG as having high data quality (A or B).
- Implementation date and overall project elapsed time are known.
- Size is measured in IFPUG 4.0 or later (because size measured with an older version is not directly comparable with size measured with IFPUG version 4.0 or later). We also removed projects that measured size with an unspecified version of function points, and whose completion pre-dated IFPUG version 4.0.
- The size in unadjusted function points is known.
- Development team effort (resource level 1) is known. Our analysis used only the development team’s effort.
- Normalized effort and recorded effort are equivalent. This should mean that the reported effort is the actual effort across the whole life cycle.
- The projects are not web projects.

In the remaining set of 909 projects, 231 were all from the same organization and 678 were from other organizations. We only selected the 231 projects from the single organization, as we considered that the use of single-company data was more suitable to answer our research questions than using cross-company data. Preliminary analysis showed that three projects were extremely influential and invariably removed from model building, so they were removed from the set. The final set contained 228 projects.

We do not know the identity of the organization that developed these projects.

Release 10 of the ISBSG database provides data on numerous variables; however, this number was reduced to a small set that we have found in past analyses with this dataset to have an impact on effort, and which did not suffer from a large number of missing data values. The remaining variables were size (measured in unadjusted function points), effort (hours), and four categorical variables: development type (new development, re-development, enhancement), primary language type (3GL, 4GL), platform (mainframe, midrange, PC, multi-platform), and industry sector (banking, insurance, manufacturing, other).

Table 1 shows summary statistics for size (measured in unadjusted function points), effort, and project delivery rate (PDR). PDR is calculated as effort divided by size; high project delivery rates indicate low productivity. In [1], the

**Table 1.** Summary statistics for ratio-scaled variables

Variable	Mean	Median	StDev	Min	Max
Size	496	266	699	10	6294
Effort	4553	2408	6212	62	57749
PDR	16.47	8.75	31.42	0.53	387.10

authors examined the project delivery rate and found it changes across time. This finding supports the use of a window.

The projects were developed for a variety of industry sectors, where insurance, banking and manufacturing were the most common. Start dates range from 1994 to 2002, although only 9 started before 1998. 3GLs are used by 86 % of projects; mainframes account for 40 %, and multi-platform for 55 %; these percentages for language and platform vary little from year to year. There is a trend over time towards more enhancement projects and fewer new developments. Enhancement projects tend to be smaller than new development, so there is a corresponding trend towards lower size and effort.

This study adopted the same range of window sizes as [3]. The smallest window size was based on the statistical significance of linear regression with windowed project data. The largest window size was based on the necessary number of testing projects for evaluation. The window ranges for the fixed-size policy is from 20 to 120 projects; those for the fixed-duration policy is 12 to 84 months.

### 3.2 Modeling Techniques

This study used Estimation by Analogy (EbA) to estimate efforts. EbA is a model-free method [8] and does not construct a model. Instead, EbA has several options to be optimized for a specific dataset [9].

In [4], the settings for EbA were as follows:

- Effort and Size were transformed to a natural logarithmic scale.
- The similarity between projects was based on Euclidean distance.
- An estimate was obtained from the arithmetic mean of logarithmic efforts of similar projects.
- Independent variables were selected with the wrapper approach [10], minimizing median MRE, on the basis of the whole dataset.

The last setting is unrealistic, in that only for the last project is the whole data set available. In practice, variables should be selected for each new estimation based on the past project data available at that time. The reason for doing a single variable selection based on the whole data set in [4] was that the wrapper approach was computationally expensive. A light weight variable selection method can resolve this problem. Furthermore, the application of EbA with these settings can be improved to improve the estimation accuracy.

This study mitigated these problems as follows:

- Select independent variables separately for every new project. This improves realism.
- Select independent variables with Lasso [11], minimizing the mean squared error. This involves less computation than using the wrapper approach.
- Adopt inverse rank weighted mean (IRWM) [12] to obtain estimates. This method was a simple method for better estimation.

The number of neighbors  $k$  we considered was  $k = 1, 2, 3, 5$ , as in [4].

### 3.3 Effort Estimation on Chronologically-Ordered Projects

This study evaluated the effects of moving windows of several sizes along with a timeline of projects' history. The effects were measured by performance comparisons between moving windows and a growing portfolio. A growing portfolio uses all past projects as the training set.

For a window of size  $w$ , this evaluation was performed as follows:

1. Sort all projects by start date
2. Find the earliest project  $p_0$  for which using that window size could make a difference to the training set: that is, at least one project that had finished by the start of  $p_0$  was "too old" to be included in the window.
3. For every project  $p_i$  in chronological sequence (ordered by start date), starting from  $p_0$ , form estimates using moving windows and using a growing portfolio.
  - For fixed-duration moving windows, the training set is the finished projects whose whole life cycle had fallen within a window of size  $w$  months prior to the start of  $p_i$ .
  - For fixed-size moving windows, the training set is the  $w$  projects that finished most recently prior to the start of  $p_i$ .
  - For the growing portfolio, the training set is all of the projects that had finished before the start of  $p_i$ .
4. Evaluate estimation results.

### 3.4 Performance Measures

Performance measures for effort estimation models are based on the difference between estimated effort and actual effort. As in previous studies, this study used MMRE and MMAE [13] for performance evaluation.

To test for statistically significant differences between accuracy measures, we used the Wilcoxon ranked sign test and set statistical significance level at  $\alpha = 0.05$ . We used the test as is because we focused on the significance of each window size, not all sizes.

**Table 2.** Accuracy with the modified EbA with  $k = 5$  (growing and fixed-size moving windows)

Window size(N)	Testing projects	Growing MAE	Window MAE	p-val.	Growing MRE	Window MRE	p-val.
20	201	2936	2838	0.36	1.53	1.45	0.06
30	178	2656	2759	0.50	1.46	1.50	0.80
40	165	2582	2785	0.34	1.43	1.54	0.41
50	153	2572	2684	0.89	1.46	1.53	0.83
60	136	2486	2353	0.06	1.54	1.43	0.08
70	126	2341	2142	0.01	1.54	1.39	0.01
80	126	2341	2298	0.29	1.54	1.56	0.32
90	111	2449	2302	0.08	1.56	1.39	0.04
100	88	2501	2504	0.21	1.49	1.62	0.21
110	75	2243	2200	0.06	1.53	1.51	0.11
120	71	2251	2274	0.36	1.52	1.52	0.76

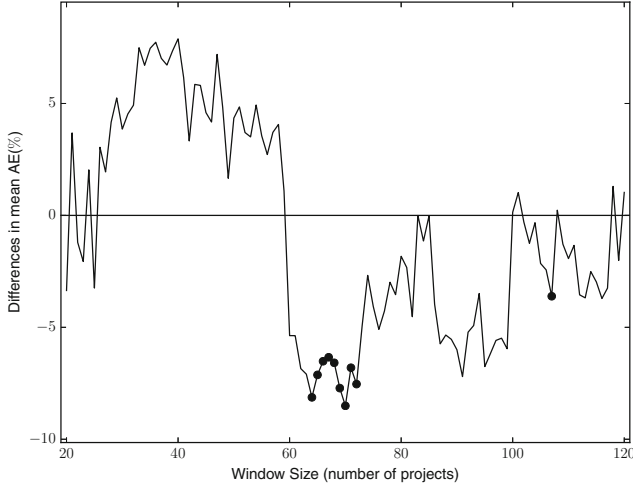
## 4 Results

### 4.1 The Effects of Changes in EbA

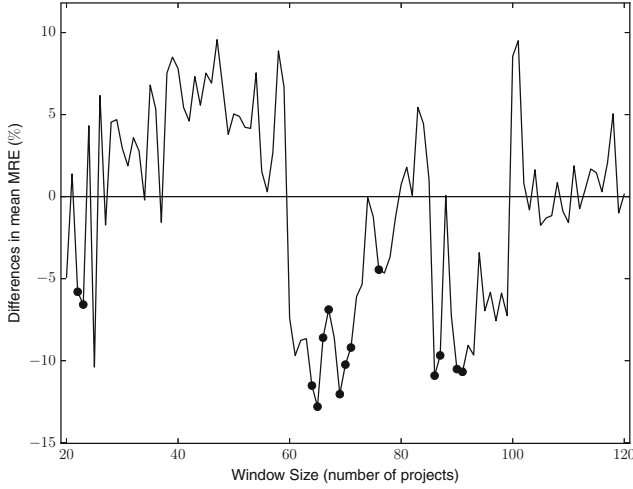
We begin by comparing estimation accuracy between EbA as used in [4] and EbA as adopted in this paper. The difference between them is the realism in practical use.

Table 2 shows the effect of fixed-size windowing with EbA as adopted in this paper, on mean absolute residuals and mean MRE. Here the number of neighbors was  $k = 5$ , which showed better performance than  $k = 2$ , the number used in [4]. The first column shows window sizes. The 2nd column shows the total number of projects used as a target project with the corresponding window size. The 3rd and 4th columns show accuracy measures of the growing portfolio and the moving windows based on MAE. The 5th column shows the p-value from statistical tests on accuracy measures based on MAE between the growing portfolio and the moving windows. The 6th and 7th columns show accuracy measures of the growing portfolio and the moving windows based on MRE. The 8th column shows the p-value from statistical tests on accuracy measures based MRE between the growing portfolio and the moving windows. The results were computed for every size; the tables only show every 10 sizes, due to space limitations. This is sufficient to show the essential trends.

Figure 1 shows the difference in mean MAE and mean MRE between the growing portfolio and moving windows with the modified EbA with  $k = 5$ . The x-axis is the number of projects in the window, and the y-axis is the subtraction of the accuracy measure value with a growing portfolio from that with moving windows at the given x-value (expressed in relative percentage terms). Smaller values of MAE and MRE are better, so the window is advantageous where the



(a) Differences in mean MAE



(b) Differences in mean MRE

**Fig. 1.** Results with Fixed-size Window, modified EbA with  $k = 5$ 

line is below 0. Circle points mean a statistically significant difference, in favor of moving windows.

Figure 1 and Table 2 revealed characteristics of moving windows compared to the growing portfolio:

- With windows of up to 60 projects, MAE showed no significant preference for any approach. The line starts below zero and quickly goes above zero (favoring the growing portfolio), but the difference was not significant as shown in



**Table 3.** Accuracy with EbA as used in [4] (repeated from [4])

Window size(N)	Testing projects	Growing MAE	Window MAE	p-val.	Growing MRE	Window MRE	p-val.
20	201	2943	3162	0.19	1.42	2.24	0.32
30	178	2711	2976	0.44	1.41	1.95	0.65
40	165	2623	2923	0.59	1.36	1.83	0.55
50	153	2575	2675	0.76	1.35	2.17	0.88
60	136	2479	2436	0.09	1.48	1.54	0.11
70	126	2305	2243	0.19	1.46	1.37	0.21
80	126	2305	2304	0.60	1.46	1.52	0.58
90	111	2662	2362	0.05	1.66	1.47	0.04
100	88	2735	2584	0.60	1.59	1.46	0.51
110	75	2467	2407	0.52	1.64	1.54	0.39
120	71	2465	2351	0.28	1.61	1.51	0.29

**Table 4.** Accuracy with the modified EbA with  $k = 2$  (growing and fixed-size moving windows)

Window size(N)	Testing projects	Growing MAE	Window MAE	p-val.	Growing MRE	Window MRE	p-val.
20	201	2891	2918	0.82	1.57	1.48	0.55
30	178	2769	2926	0.66	1.53	1.51	0.57
40	165	2718	2950	0.96	1.51	1.80	0.62
50	153	2682	2872	0.43	1.49	1.86	0.31
60	136	2541	2505	0.18	1.56	1.48	0.26
70	126	2364	2362	0.41	1.58	1.70	0.47
80	126	2364	2479	0.83	1.58	1.65	0.51
90	111	2461	2382	0.33	1.53	1.41	0.19
100	88	2459	2878	0.73	1.37	1.81	0.86
110	75	2216	2702	0.73	1.41	1.92	0.91
120	71	2199	2805	0.17	1.40	1.87	0.16

Fig. 1(a). MRE showed a similar trend, except that moving windows were sometimes significantly advantageous around small window sizes, as shown in Fig. 1(b).

- For windows of 60 to 100 projects, moving windows are advantageous in MAE. There were several window sizes around 60 to 75 projects where the difference is significant, as shown in Fig. 1(a). The difference in MRE showed a similar trend, again with a significant advantage around 60 to 75 projects but also at several sizes around 90 projects.

- With windows of 100 projects or more, both measures showed no clear preference for windows or growing.

In summary, in this data set, moving windows improved estimation accuracy significantly with windows in the middle of the range of sizes investigated.

Comparing these results to [4], in the previous paper the effects of fixed-size moving windows were as follows:

- With a window of 20 to 55 projects, all measures were always better using the growing portfolio though the difference was not statistically significant.
- With a window of 90 or 91 projects, all measures were better using the moving windows and the difference is statistically significant. Although there were the only sizes where the difference was statistically significant, these were not just “lucky” window sizes: at nearly all window sizes from 61 to 120 projects, average values of all of the accuracy statistics were better with the moving windows.

Two things have changed between [4] and here: how EbA was applied, and the choice of the best value for  $k$ . To separate the effect of the two changes, we present two tables. Table 3 repeats the results from [4], for convenience. Table 4 presents an intermediate stage: it shows the accuracy with the modified EbA but with  $k$  held at 2. Thus the difference between Tables 3 and 4 shows the effect of modifying EbA, and the difference between Tables 4 and 2 shows the subsequent effect of changing  $k$ .

Most of the values in Table 4 are similar to or worse than the corresponding values in Table 3. This implies that the modification to EbA reduces the accuracy of the estimates. This may be because variable selection was done once in [4], using the entire data set; hence insights drawn from the whole data set were used in variable selection for every project, even early ones in the sequence. Less information is available for most projects in the modified approach, which could make the estimates less accurate.

However, most of the values in Table 2 are better than the corresponding values in Table 3. Increasing  $k$  from 2 to 5 more than overcomes the loss of accuracy in modifying the EbA approach.

Overall, the change in EbA, which is aimed at improving the realism of the estimation procedure and reducing computation effort, has also improved the estimation accuracy when combined with a change in  $k$ . Estimates are more accurate on average, and need fewer comparison projects for windows to be valuable: using the modified approach, windows were significantly better than the growing approach at windows of around 60 to 75 projects, according to MAE, and around 60 to 90 according to MRE, instead of 90 projects with the original method.

Table 2 and Fig. 1 present the best results for this data set, using windows defined as containing a fixed number of projects. In the next section we perform a similar experiment, using the same estimation method, but defining windows as covering fixed numbers of months.

**Table 5.** Accuracy with modified EbA with  $k = 5$  (growing and fixed-duration moving windows)

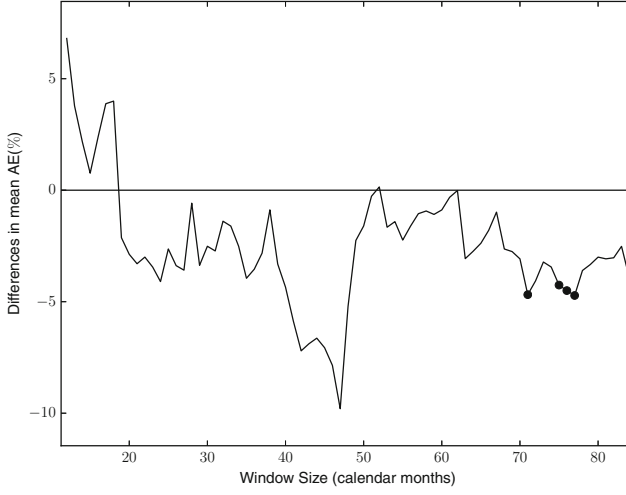
Window size(N)	Testing projects	Growing MAE	Window MAE	p-val.	Growing MRE	Window MRE	p-val.
12	165	2582	2757	0.56	1.43	1.49	0.88
18	193	2834	2947	0.83	1.47	1.54	0.80
24	201	2936	2816	0.29	1.53	1.45	0.55
30	202	2940	2866	0.53	1.52	1.39	0.63
36	206	2940	2836	0.81	1.50	1.41	0.63
42	206	2940	2728	0.34	1.50	1.39	0.45
48	206	2940	2787	0.43	1.50	1.42	0.68
54	206	2940	2898	0.44	1.50	1.39	0.52
60	198	2951	2925	0.72	1.54	1.46	0.57
66	184	2776	2726	0.76	1.46	1.41	0.57
72	153	2572	2468	0.09	1.46	1.36	0.05
78	126	2341	2257	0.07	1.54	1.45	0.04
84	80	2461	2364	0.16	1.61	1.55	0.26

## 4.2 The Effects of Moving Windows of Fixed Duration

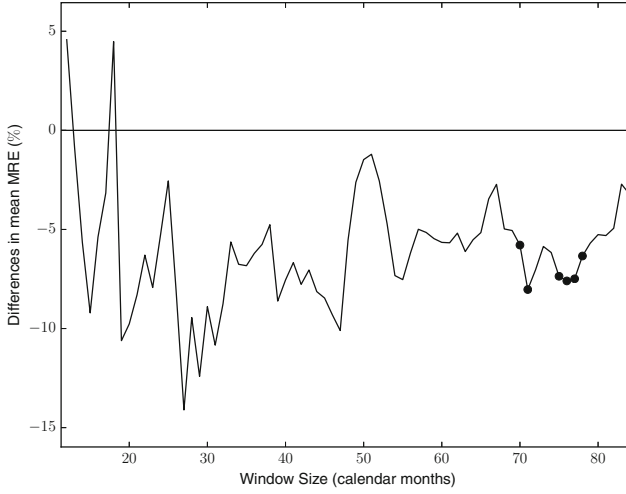
Table 5 shows the effect of the fixed-duration windowing, with the modified EbA with  $k = 5$ , on mean absolute residuals and mean MRE. Figure 2 shows the difference in mean MAE and mean MRE between the growing portfolio and moving windows. The notation is as same as in Fig. 1, except that the x-axis is now the window duration in months. The table and the figure reveal the following:

- With windows of up to 20 months, the growing portfolio was advantageous in terms of MAE. No difference was statistically significant. The advantage was not clear in MRE for that range.
- With windows of 20 to 50 months, the lines go down under the zero line and support the moving windows in terms of average differences in MAE and MRE. However, statistical tests showed no statistically significant differences. The lines then go back to close to zero.
- With windows of more than 55 months, moving windows are advantageous again. There were significant differences between 70 to 80 months, supporting the moving windows.

In [3], the authors used the same dataset, the same range of window durations, and linear regression to examine the effects of fixed-duration windows. Thus the difference between this work and [3] is the use of EbA instead of linear regression. The observations in [3] were:



(a) Differences in mean MAE



(b) Differences in mean MRE

**Fig. 2.** Results with Fixed-duration Windows, EbA with  $k = 5$ 

- With windows up to 24 months, the growing portfolio was advantageous. Statistical tests sometimes supported the growing portfolio.
- With windows between 24 to 50 months, moving windows were advantageous. There were some window sizes where the difference was statistically significant.
- With larger windows, the difference got smaller, and there was no statistical difference between the growing portfolio and moving windows.

The observations in [3] and the results in this paper show different trends. The window durations at which windows are advantageous compared to the growing portfolio are larger with EbA than with LR, and the range of durations

for which windows are advantageous is narrower with EbA than with LR. The difference in advantageous window sizes and their number between EbA and LR were reported in [4]. These observations were common between this study and [4].

## 5 Discussion

### 5.1 Answer to RQ1

The first part of this research differs from [4] in that changes were made in settings for EbA, with the aim of improving realism and reducing computation effort. Our first research question is whether the change in settings makes a difference to the estimation accuracy, while still adopting fixed-size windows.

The results are different in three respects. The first difference is a change in the optimal setting for the best number of neighbors,  $k$ . Previously  $k = 2$  was best. The change in estimation method brought a new best setting  $k = 5$ . The second difference is an improvement in estimation accuracy. Comparison between Tables 2 and 3 show that the modified EbA with  $k = 5$  has better estimation accuracy on average. The third difference is a change in the window sizes at which moving windows are advantageous for estimation accuracy. With the changes to EbA and the optimal number of neighbors, we see a change of advantageous window sizes. The result shows a wider range of advantageous window sizes, and smaller advantageous window sizes.

We thus conclude that the change in estimation method made a difference, improving the accuracy of estimates.

This result updates [4]. It repeated the same underlying experiment, in which the key is the use of fixed-size windows and EbA, but with a better method for applying EbA.

### 5.2 Answer to RQ2

The second research question is whether the use of fixed-duration windows, instead of a growing portfolio, makes a difference to estimation accuracy when the new EbA is adopted.

Figure 2 showed the general trend that when using fixed-duration moving windows instead of a growing portfolio, the estimation accuracy improved as the window size increased. The differences are statistically significant at several durations between 70 to 80 months. The general trend looked similar to that with LR, as shown in [3], although the window sizes where the moving windows were significantly advantageous are different.

We thus conclude that fixed-duration windows can make a difference, and are effective to improve estimation accuracy.

### 5.3 Answer to RQ3

Table 2 and Fig. 1 present the results for this data set, using the modified EbA method and using windows defined as containing a fixed number of projects. Table 5 and Fig. 2 present the corresponding results when windows are defined as having fixed duration instead of containing a fixed number of projects.

From RQ1 and RQ2 we see that both windowing approaches can lead to significantly better estimation accuracy.

Figure 1 shows that at the window sizes where fixed-size windows lead to significantly better estimates than the growing portfolio, the improvement in MAE is around 7–9 % and the improvement in MRE is mostly around 12 %. With fixed-duration windows, as seen in Fig. 2, significant improvements in MAE are around 5 % and significant improvements in MRE are around 7–9 %. Thus the gains are smaller with fixed-duration windows.

With fixed-duration windows, the number of advantageous window sizes is smaller with EbA than with LR. This was also observed in [4]; this property was maintained in this study despite the changes to EbA and window policies. The degree of the improvement was weaker than that obtained with fixed-size windows. This characteristic was also observed in [3].

These observations imply that the use of fixed-size windows has more impact on estimation accuracy than the use of fixed-duration windows, at least with this dataset. The difference of datasets caused the difference of the effects of the moving windows as shown in [3]. Further study with other datasets is an area for future work.

### 5.4 What are the Practical Implications of this Study?

The implications of this study are as follows:

First, moving windows are suggested as an alternative approach to effort estimation for companies instead of using the whole history of past data. They have been shown now to be effective with the two most common estimation methods, LR and EbA. Research is still needed on the use of moving windows with other estimation methods.

Second, although it is more natural to think in terms of durations of windows rather than the number of projects in windows, in this data set the fixed-size window policy is more effective than the fixed-duration window policy. This has been shown using both LR and EbA. Practitioners may need to change their thinking, such that how many projects are available from which to learn might be more important than how recent the projects are.

Third, effective window sizes might be different even among practitioners. EbA resembles practitioners' thinking. Changes to how they arrive at an estimate may change the number of projects they should consider. This can result in a change to advantageous window sizes. This may partly explain why practitioners can make different estimates while drawing on the same repository of data about past projects.

## 6 Threats to Validity

This study has some threats to validity in common with previous studies.

First, we used only one dataset. The dataset is a convenience sample and may not be representative of software projects in general. Thus, the results may not be generalized beyond this dataset; this is true of all studies based on convenience samples. We trust that some potential sources of variation are avoided by the selection of a single-company dataset. Since the dataset is large and covers several years, we assume it is a fair representation of this organization’s projects. The inclusion of the industry sector as an independent variable helps to allow for variations among sectors in the dataset. Experiments with other datasets are our major future work.

Second, this study applied EbA in a specific way. EbA has several options to be optimized for a specific dataset, as shown in [9], and high-quality models are dataset-driven in nature. Our choice of method might have missed more accurate or more realistic methods. Based on our past experience building models manually, we believe that the approach used here is acceptable, and the variable selection approach is more realistic than previously studied in [4].

## 7 Conclusions

This paper investigated the effect on the accuracy of effort estimation using EbA, when moving windows are used to retain only “recent” training data and the windows are of fixed durations.

The use of fixed-duration windows was able to improve the accuracy of estimates, in terms of both MAE and MRE, compared to the growing portfolio in which the entire history of training data is retained.

The advantage over a growing portfolio from using fixed-duration windows was smaller than the advantage from using fixed-size windows. The same was found in [3], in which LR was used rather than EbA as the estimation approach.

The paper has made these contributions:

- Changes were proposed and evaluated to how EbA was applied in [4], to improve its realism in practice and to reduce the computational effort. The changes improved the accuracy of estimates, and the useful window sizes were smaller so less data needed to be retained.
- Windows based on duration can improve the accuracy of estimation by analogy. This is useful because estimation by analogy is very common, and anecdotally filtering of projects based on recency is also very common. Past research has shown that fixed-duration windows help less than fixed-size windows, and windows help less with EbA than with LR. Evidence that duration-based windows can be effective with EbA is valuable.

The above observations were obtained using one specific approach to EbA, with one dataset. Our future work involves generalization with other settings: other companies’ datasets and perhaps other options for EbA such as using recency as part of the distance metric [14] and greedy search for feature selection [15].

**Acknowledgment.** The authors would like to thank the anonymous reviewers for their thoughtful comments and helpful suggestions on the first version of this paper. This work was partially supported by JSPS KAKENHI Grant #25330083 and #15K15975.

## References

1. Lokan, C., Mendes, E.: Applying moving windows to software effort estimation. In: Proceedings of ESEM 2009, pp. 111–122 (2009)
2. Lokan, C., Mendes, E.: Investigating the use of duration-based moving windows to improve software effort prediction. In: Proceedings of APSEC 2012, pp. 818–827 (2012)
3. Lokan, C., Mendes, E.: Investigating the use of duration-based moving windows to improve software effort prediction: a replicated study. *Inf. Softw. Technol.* **56**(9), 1063–1075 (2014)
4. Amasaki, S., Lokan, C.: The effects of moving windows to software estimation: comparative study on linear regression and estimation by analogy. In: 2012 Joint Conference of 22nd International Workshop on Software Measurement and the 7th International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 23–32. IEEE, October 2012
5. Mendes, E., Lokan, C.: Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions: a replicated study. In: Proceedings of EASE 2009 (2009)
6. Kitchenham, B., Pfleeger, S.L., McColl, B., Eagan, S.: An empirical study of maintenance and development estimation accuracy. *J. Syst. Softw.* **64**(1), 57–77 (2002)
7. MacDonell, S.G., Shepperd, M.: Data accumulation and software effort prediction. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM (2010)
8. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining Inference and Prediction*. Springer, New York (2009)
9. Kocaguneli, E., Menzies, T., Bener, A., Keung, J.W.: Exploiting the essential assumptions of analogy-based effort estimation. *IEEE Trans. Softw. Eng.* **38**(2), 425–438 (2012)
10. Dejaeger, K., Verbeke, W., Martens, D., Baesens, B.: Data mining techniques for software effort estimation: a comparative study. *IEEE Trans. Softw. Eng.* **38**, 2354–2364 (2011)
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. Ser. B* **58**, 267–288 (1996)
12. Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S.: A comparative study of cost estimation models for web hypermedia applications. *Empirical Softw. Eng.* **8**(2), 163–196 (2003)
13. Port, D., Korte, M.: Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. In: Proceedings of ESEM 2008. ACM (2008)
14. Kolodner, J.: *Case-Based Reasoning*. Morgan-Kaufmann, San Mateo (1993)
15. Kirsopp, C., Shepperd, M., Hart, J.: Search heuristics, case-based reasoning and software project effort prediction. In: GECCO 2002: Genetic and Evolutionary Computation Conference. AAAI (2002)



Software Measurement

25th International Workshop on Software Measurement  
and 10th International Conference on Software Process  
and Product Measurement, IWSM-Mensura 2015,  
Kraków, Poland, October 5-7, 2015, Proceedings  
Kobyliński, A.; Czarnacka-Chrobot, B.; Świerczek, J. (Eds.)  
2015, XII, 209 p. 77 illus. in color., Softcover  
ISBN: 978-3-319-24284-2