A Stratification and Sampling Model for Bellwether Moving Window

Solomon Mensah[1], Jacky Keung[1], Michael Bosu[2], Kwabena Bennin[1] and Patrick Kwaku Kudjo[3]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]Centre for Business, Information Technology and Enterprise, Wintec, Hamilton, New Zealand
[3]Department of Computer Science and Communication Engineering, Jiangsu University, China
{smensah2-c, kebennin2-c}@my.cityu.edu.hk, Jacky.Keung@cityu.edu.hk,
michael.bosu@wintec.ac.nz, kudjo@ujs.edu.cn

***Abstract*—An effective method for finding the relevant number (window size) and the elapsed time (window age) of recently completed projects has proven elusive in software effort estimation. Although these two parameters significantly affect the prediction accuracy, there is no effective method to stratify and sample chronological projects to improve prediction performance of software effort estimation models. Exemplary projects (*Bellwether*) representing the training set have been empirically validated to improve the prediction accuracy in the domain of software defect prediction. However, the concept of *Bellwether* and its effect have not been empirically proven in software effort estimation as a method of selecting exemplary/relevant projects with defined window size and age. In view of this, we introduce a novel method for selecting relevant and recently completed projects referred to as *Bellwether moving window* for improving the software effort prediction accuracy. We first sort and cluster a pool of *N* projects and apply statistical stratification based on Markov chain modeling to select the *Bellwether moving window*. We evaluate the proposed approach using the baseline Automatically Transformed Linear Model on the ISBSG dataset. Results show that (1) *Bellwether effect* exist in software effort estimation dataset, (2) the *Bellwether moving window* with a window size of 82 to 84 projects and window age of 1.5 to 2 years resulted in an improved prediction accuracy than the traditional approach.**

Keywords—Bellwether effect; Markov chains; Window age; Window size; Chronological dataset.

1. INTRODUCTION

In the construction of a robust and accurate prediction model, there is the need to consider the best selection of training and validation sets prior to modeling. According to Lokan and Mendes [5][12], selection of training and validation sets from a subset of chronologically arranged projects (referred to as *moving window*) can further improve the prediction accuracy of effort estimation models. The moving window theory is based on the assumption that recent projects are likely to share similar characteristics with *new* projects. Recent studies [10][2][7][17] support the theory by Lokan and Mendes that, *the use of moving window as the training set improves the estimation accuracy of effort estimation models*. The moving window approach has recently been supplemented with *weighting functions which further improve the estimation accuracy of effort estimation models.* This theory was postulated by Amasaki and Lokan [10][2][18][17] and was empirically confirmed to improve the estimation accuracy when using large-sized moving windows. These previous studies found software effort estimation (SEE) models built using recently completed projects (weighted/unweighted moving window) superior to models that use all available historical projects (referred to as the *growing portfolio*).

Irrespective of the significant improvement of using weighted/unweighted moving windows, challenges still exist in determining how large and how old the moving window should be prior to SEE modeling. The *window size* simply refers to the number of most recently completed historical projects appropriate for building the SEE model. The *window age* refers to the elapsed time of projects (within the moving window) that has existed for not more than *t* calendar years or calendar months [7]. Previous studies [2][5][10][18] randomly used different moving window sizes such as 20, 30, 40…120 projects. Similarly, different moving window ages (such as 1 year, 1.5 years, 2 years…) were used by previous studies [7][12][17]. These arbitrary window sizing and window aging parameters significantly affect the estimation accuracy irrespective of the weighting functions used [2][10]. This is mainly due to the variations in the sample sizes drawn from the chronological datasets. Based on previous moving window studies [10][5][2][12], this study aims at addressing the unstable window sizing and window aging constraint to further improve the estimation accuracy of predictive models. The existence of exemplary projects referred to as *Bellwether* has successfully been used to improve the relative estimation accuracy in software defect prediction by Krishna et al. [1].

We therefore seek to address this problem by introducing a novel model to sort, stratify and statistically sample exemplary and recently completed historical projects to be considered as the *Bellwether moving window*. The model operates by applying the concept of *Bellwether* [1] and a probabilistic modeling approach namely Markov chain Monte Carlo [6] to determine an effective size and age of the moving window used in modeling.

Based on Markov chain modeling, the outcome of an event (*new* project) in an experiment depends only on the previous experiment (recently completed projects with defined transition probabilities). The concept of *Bellwether* (specifically the *Bellwether effect*) and the Markov chains theory [6] are empirically proven to determine the best window age and window size of moving windows in this study.

In this study, we make the following contributions: To the best of our knowledge, this is the first study to 1) empirically prove the existence of *Bellwether* in software effort estimation, and 2) compute the best window size and window age of the

DOI reference number: 10.18293/SEKE2017-126

sampled *Bellwether moving window* based on Markov chain Monte Carlo.

This study is unique since the *Bellwether method* (as elaborated in Section 4) addresses the window aging and sizing constraint by developing a step-by-step sampling method as well as computing the window size and window age of the *Bellwether moving window* from a set of chronological projects.

The remaining sections of the paper are organized as follows. Section 2 presents the *Bellwether* concept and the Markov chain Monte Carlo approach. Section 3 presents the postulations with their respective proofs. The proposed approach is presented in Section 4. Section 5 details the methodological procedure. Section 6 presents the experimental results. Section 7 presents the threat to validity and Section 8 concludes the study.

2. BACKGROUND

A. The Concept of Bellwether

According to Krishna et al. [1], the concept of *Bellwether* which is a simple transfer learning technique is defined in 2-folds namely the *Bellwether effect* and the *Bellwether method*:

1) The Bellwether effect states that, given a set of N projects, there exist exemplary project(s) that can form the Bellwether and can provide the best prediction accuracy for the remaining N-1 projects.

2) The Bellwether method uses a heuristic approach to search for that particular Bellwether from a pool of completed projects and applies it to a new project whose target is to be estimated.

This concept of *Bellwether* was considered by Krishna et al. [1] in the domain of defect prediction whereby given a set of non-chronological projects, each project was used as a potential *Bellwether* to successfully make predictions on the remaining projects. Results from their study show that *Bellwethers* could obtain the best training sets with relative improved prediction accuracy for building estimation models.

This paper therefore investigates the feasibility of using *Bellwether* as a *moving window* for building software effort estimation models. We first, sort the projects chronologically and apply a statistical stratification technique to obtain only the recently completed projects. We then check for the existence of *Bellwether* based on a defined *Bellwether method* as elaborated in Section 4. We define the moving window obtained from the Bellwether method as the *Bellwether moving window.* The *Bellwether moving window*, w_i is used to estimate each of the remaining windows, w_j in $N \; \forall \; i \neq j$.

B. Markov chain Monte Carlo

The Markov chain Monte Carlo (MCMC) is a probabilistic technique that seeks to solve the problem of sampling by exploring a given sample space through the construction of an ergodic Markov chain (EMC) whose limiting distribution is the target distribution [6]. MCMC utilizes Markov chains to effectively simulate a random variable *X* whose future states are independent of past states given the present state. MCMC has been applied and proven successful in different software engineering domains such as software testing [8], image processing [11] as well as applied mathematics, statistical and biomedical engineering.

We provide a statistical investigation on the use of MCMC to obtain a potential subset of relevant projects to be considered as the *Bellwether*. Let *D* denotes a set of projects from a given population (industry) with *d* degree of dimensions (features). Then, $X = (x_1, \ldots, x_d) \epsilon D$ can represent a sample whose limiting distribution is known and can form the *Bellwether*. Each of the sample points $x_1, \ldots, x_d$ of the sample is allowed to take a discrete or continuous time value which denotes the project ages already known from the given repository *(D)*. The Markov chain is then executed a number of times until the chain converges to its limiting distribution to obtain the target sample subset [6]. The selected sample with its respective *t* states (or ages) and constructed limiting distribution forms the *Bellwether* which can be used as the moving window. We define this type of moving window as the *Bellwether moving window.*

3. POSTULATIONS

In this study, we prove the following three postulations based on empirical analysis that:

1) Given a set of N chronological projects with a finite mean μ and variance σ^2, then there exist a potential Bellwether which can be used as a moving window.

Proof: Let $\{X^{(1)}, \ldots, X^{(t)}\}$ be independent and identically distributed (*iid*) random samples chronologically drawn from a given population, *N*. If a random sample $X^{(i)} > 30$ is drawn from *N*, then (a) according to the central limit theorem [13] and the law of large numbers [14], $X^{(i)}$ will follow the normal distribution with mean, μ and variance, σ^2 that is, $X^{(i)} \sim N(\mu, \sigma^2)$ and (b) the $X^{(i)}$ with the minimum accuracy measure to be considered as a *Bellwether* can then be used to successively predict the remaining samples, $X^{(j)} \; \forall \; j \neq i$. That is the i^{th} *Bellwether* sample predicts the remaining j^{th} samples.

In order to confirm the existence of a potential *Bellwether* from a given population set *N*, empirical analysis was performed based on statistical stratification of *N* using the *X-means* clustering [15], and samples were drawn from each stratum. We realized that, the best population stratum in which potential samples can be drawn from was of size not less than 100 projects. This was empirically validated by applying the *Bellwether method* (in Section 4) to the population set.

2) If P denotes a regular transition probability matrix (TPM) of a Markov chain, then there exist an ergodic Markov chain (EMC) whose respective window can be used as the Bellwether.

Proof: We define the Markov chain as a set of random variables $\{X^{(1)}, \ldots, X^{(t)}\}$ or a collection of stochastic events $\{X(t) | t \geq 0\}$ whereby given the present event of a state at time *t*, the prediction of a future event at *t+1* is independent of past events but the present event. The *iid* sample space for all states at their respective times can be described as a Markov chain if

$$P(X_{t+1} = i_{t+1} | X_t = i_t, X_{t-1} = i_{t-1} \ldots X_1 = i_1, X_0 = i_0) = P(X_{t+1} = i_{t+1} | X_t = i_t) = p_{ij}^{(k)} \quad (1)$$

Assume there exist a variable '*P*' that can be formulated as a matrix of transition probabilities of a Markov chain [6] from

the sample space $\{X^{(1)},\ldots,X^{(t)}\}$ where $i \epsilon T$ denotes the transition states (project ages), then the ij^{th} element, $p^{(k)}_{ij} \epsilon P$ is the probability that the Markov chain starting from a particular state, t_i will transition to t_j after k steps. If $p^{(k)}_{ij}$ is homogeneous, then (2) holds and there exist a unique probability matrix, $\boldsymbol{\theta}_u$ that can form the ergodic Markov chain (EMC) such that for any $\boldsymbol{\theta}_o$ and for large values of u, (3) can be defined as follows:

$$P(\mathrm{X}^{(\mathrm{k})} = j \,|\, \mathrm{X}^{(\mathrm{k}-1)} = i) = \mathrm{p}_{ij}^{(k)} \quad (2)$$

$$\lim_{u\to\infty} \theta_{u+1} = P^u \theta_1 \quad (3)$$

Thus, the EMC can be obtained given a non-negative power of P by making all the entries of the probability matrix non-zero and irreducible. If the limiting state probability matrix (EMC) exists from the TPM constructed with the moving window sample, then that particular sample can be used as the *Bellwether moving window*. We consider such moving window sample as *Bellwether* since its limiting or stationary distribution has been reached and hence can form a potential training set for modeling given it has the best prediction accuracy performance.

3) Given a Bellwether whose ergodic Markov chain is known, then its size and age can be defined.

Proof: Let a sample X with projects $\{p_1,\ldots,p_n\}$ be classified into t states based on their respective ages. Assume that $p_i \epsilon X$ are sorted in non-decreasing order based on the project ages and the EMC of X is known. Then, the age of the *Bellwether* sample, X can be found as the difference between the maximum and minimum states (ages) of the sorted p_i whose EMC is known. Similarly, the size of the *Bellwether* can be found as the total number of projects in X.

A. Assumptions

The following assumptions are made for the effective functioning of the proposed *Bellwether method*:

- Each project used in modeling has an age. Thus, we pruned off irrelevant projects without start and completion dates prior to modeling.
- All selected projects from N have the same features.
- All selected projects are from a single industry.
- All selected projects share common development policies or characteristics.
- Each p_{ij} element in the TPM lies within 0 and 1.
- The sum of each i^{th} row of TPM should be approximately 1.
- All entries of the ergodic Markov chain are non-zero.

4. Bellwether Moving Window: A New Approach

In order to support the replication of this study, we describe the *Bellwether method* for obtaining the moving window in this section. Thus, we introduce a stratification and sampling approach that will assist a software practitioner to select the best moving window from a repository, D of chronological projects to aid in estimating the software effort of a *new* project. We illustrate the general architecture of the *Bellwether method* in Figure 1. Using historical data from D, the following three operators (*SORT+CLUSTER*, *GENERATE TPM* and *APPLY*) can be applied by the software practitioner:

SORT + CLUSTER

Given a set of N completed projects from D, sort based on the project completion dates and stratify the data into q clusters.

1. For all N projects from D, sort in an increasing order using their respective project completion dates.
2. Subject sorted data to *X-means*[1] clustering algorithm [15] to obtain q clusters. Perform data stratification based on the q obtained. That is, stratify N into q clusters whereby each cluster (or window) has a set of chronologically arranged projects.
3. For each window, compute the weighted moving window by applying the respective weighting functions (Triangular, Epanechnikov, Gaussian and Rectangular [10]) on the q windows. We define the resulting q weighted windows in an increasing order as $w_1, w_2 \ldots w_q$.
4. Use w_q as a baseline. w_q contains recently completed projects in a chronological order and can form a *baseline weighted moving window*.

GENERATE TPM

Generate the transition probability matrix (TPM) for the resulting weighted moving window and find the respective ergodic Markov chain (EMC).

5. For the resulting baseline window (w_q), generate the Transition Probability Matrix (TPM) of the Markov chain. Here, use the project ages in w_q as the transition states with their respective effort to generate the TPM.
6. Compute the EMC for the generated TPM in order to validate the limiting (or stationary) distribution of w_q. Thus, using the TPM with s states, successfully perform the squaring of TPM until TPM reaches its stationary distribution (ergodic). That is, when individual probability elements, p_{ij} of TPM cannot be reduced further, we say that TPM is regular or ergodic [6].
7. Report w_q* as a *Bellwether moving window* if the weighted moving window, w_q is stationary and provides the best prediction accuracy for majority of the remaining windows, $w_1, w_2 \ldots w_{q-1}$. Else go to step 4 to update w_q with additional project(s) from w_{q-1} and repeat steps 5 and 6. Conversely, update w_q by sequentially removing project(s) from w_q and adding to w_{q-1} and repeating steps 5 and 6.

APPLY

Apply the Bellwether moving window, w_q to the new project data (set as a hold-out) whose software effort is to be estimated.*

8. Once w_q* is obtained as a *Bellwether moving window* in step 7 prior to its application to the *new* project data (set as a hold-out), its size and age can be defined from its dimensions. The size of the moving window is computed as the total number of projects within the

[1] *X-means* automatically estimate the number of optimal clusters with respect to the Bayesian Information Criterion [15]

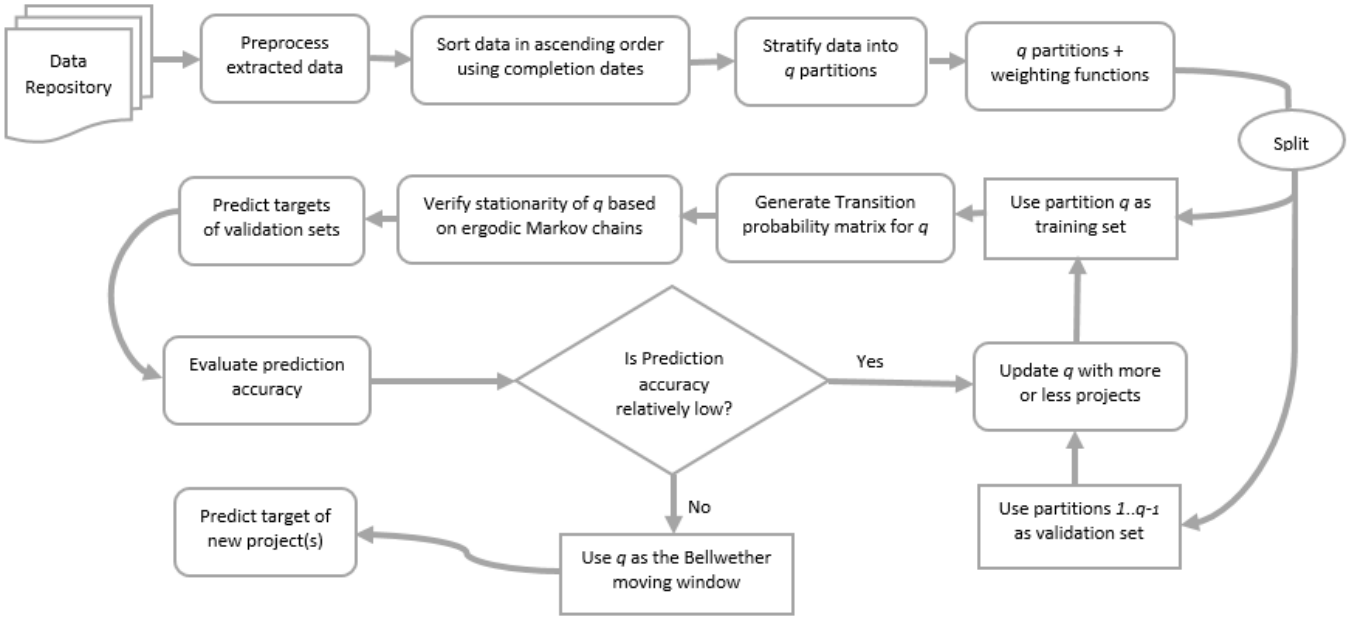

Figure 1. Overview of Bellwether Method for sampling the Bellwether Moving Window

Bellwether moving window whiles the age of the moving window is computed as the difference between the maximum and minimum project ages.

9. Predict the software effort of the *new* project(s) using the resulting *Bellwether moving window* model.

5. METHODOLOGY

A. Dataset

In this study, we use the International Software Benchmarking Standards Group (ISBSG) dataset release 10 which is sourced from the ISBSG dataset repository[2]. It is a chronological cross company dataset with a total of 4106 projects. All historical projects span approximately 20 years between May, 1988 and November, 2007. For easy comparison of our approach with previous moving window studies, we use the same ISBSG dataset that has previously been considered for moving windows [2][5][7]. Although the dataset seems old, the focus is not on how old the dataset is but to prove that *new* projects can be predicted from previously completed projects based on the *Bellwether effect* and Markov chain modeling.

We preprocess the ISBSG dataset following a similar approach by previous studies [2][7]. The preprocessing resulted in a total of 1097 projects (26.7%) selected from the population set (4106 projects). The ISBSG dataset contains variants of projects from different industries with different policies and development methodologies. Hence, such cross projects are heterogeneous in nature and do not share common development policies. We therefore extracted projects from a single industry and hence selected 237 projects (21.6%) out of the total 1097 preprocessed projects. Ten projects (4.2%) with recent completion dates are set aside as hold-out for testing the prediction accuracy of the *Bellwether method*. The selected features from the ISBSG dataset are the size of the projects measured in Unadjusted Function Points (UFP), primary language type (3GL, 4GL), development type (new development, re-development and enhancement), platform (PC, mainframe, midrange and multi-platform), industry sector (manufacturing, banking, insurance and other) and the effort measured in person-hours. It should be noted that, we used the same features considered in previous moving window studies [2][7] to support the replication of previous studies.

B. Experimental Setup and Evaluation Measures

This study employs a proposed *Bellwether method* described in Section 4 to select the *Bellwether moving window* with the respective window size and window age for estimating the software effort of *new* projects. Comparison is made across the use of weighted moving windows (Triangular, Epanechnikov and Gaussian), unweighted moving windows (Rectangular) and growing portfolio (use of all preprocessed projects as training set). These weighting functions have been considered in previous studies [2][10][18]. We evaluate the prediction accuracy performance of the *Bellwether moving window* by using the selected window sample to perform successive predictions on the remaining unselected windows (see detailed explanation in Section 4). Thus, whiles the selected moving window is used as the training set, the unselected windows are used as the validation sets. Thus, after obtaining the *Bellwether* (partition sample with the best prediction accuracy on the remaining partitions), it is then used for estimating the *new* projects. On the other hand, we use the *growing portfolio* as the training set as done in previous studies [2][7][10] for estimation purposes. We went a step further to apply the weighting functions considered in [10] on the *growing portfolio* to empirically investigate their performance in the estimation process. Lastly, we compare our *Bellwether moving window* approach with the *growing portfolio* approach.

[2] http://www.isbsg.org

The estimation model considered in this study is the Automatically Transformed Linear Model (ATLM) proposed by Whigham et al. [3]. ATLM was developed using multiple linear regression with a minimal threshold to act as a baseline for software effort estimation. The prediction accuracy of ATLM proved superior to ensemble methods and a hybrid model (particle swarm optimization, analogy-based estimation and clustering).

We employed the Mean Absolute Error (MAE) which has been proven reliable by Foss et al. [9] and considered in previous studies [3][10][2]. MAE is a risk function that measures the average absolute deviation of the estimated effort values from the true effort values. It provides an unbiased measure that favors model generalization. A robust statistical test called the Welch *t-test* statistic as recommended by Kitchenham et al. [19] is also used to determine the statistical pairwise differences among the weighted (Triangular, Epanechnikov and Gaussian) and unweighted (Rectangular) windows [10] and the growing portfolio. We considered the Kruskal-Wallis *H-test* statistic [4] to find the existence of statistical difference among the four weighting functions applied for both the moving window and the growing portfolio. The Glass's Δ effect size [16] is used to find the practical significance among the *Bellwether moving window* and the *growing portfolio*. We considered these performance measures due to their robustness to outliers.

6. RESULTS

In this section, we present the empirical analysis of the *Bellwether effect* in the chronological dataset. We compare the evaluation performance of using the *Bellwether moving window* and the *growing portfolio* for prediction purposes.

A. Bellwether Effect in Chronological dataset

To show that there exist exemplary projects to be considered as the *Bellwether* sample (postulation 1 in Section 3), we randomly sampled projects from size 30 to the total number of utilized projects (237). Each size ($N \geq 30$) acted as the sample space and the *Bellwether method* (Section 4) was applied on each sample space until a desired threshold size was obtained for the *Bellwether*. Results show that *Bellwether effect* exist in a sample space of $N > 100$ projects to obtain a potential *Bellwether moving window*.

After obtaining the threshold for the existence of *Bellwether*, we report the empirical analysis for this study using the maximum threshold of $N = 237$ (setting 10 projects as hold-out). Results show that, after subjecting the sorted chronological dataset to the *X-means* clustering algorithm, 3 clusters (with initial approximate size of 76 projects) were recorded. The transition probability matrix (TPM) from the Markov chain modeling show that, there exist an ergodic Markov chain (EMC) from the 3 clusters or windows whereby its respective window can be considered as the *baseline window*. We empirically validated this initial *baseline* with MAE and updated the sizes (as shown in step 7 of Section 4) until the *Bellwether moving window* was obtained.

We present the prediction evaluation performances of different potential *Bellwether moving windows* in Figure 2. Results from Figure 2 depict the MAE evaluation of using relevant windows weighted with the four functions (Gaussian, Triangular, Epanechnikov and Rectangular). The selected

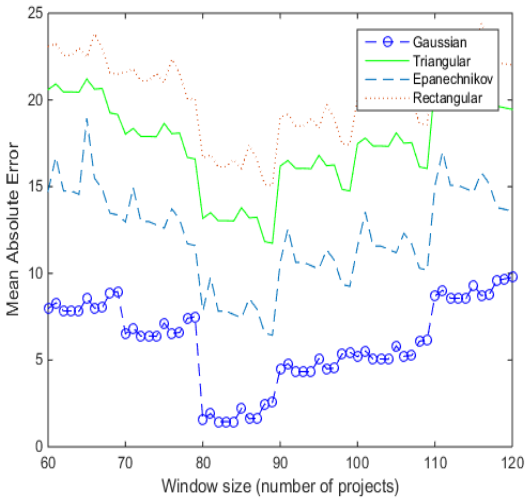

Figure 2. MAE prediction evaluation of different window sizes subjected to the four weighting functions

potential *Bellwether moving windows* were used to predict the projects that were set as hold-out as shown in Section 4. We present results of potential *Bellwether moving windows* of sizes ranging from 60 to 120 since they yielded relatively minimal error evaluation measures on average. On average the best number of projects to be considered as the *Bellwether moving window* ranges from 80 to 90 projects (35.2% to 39.7% of the training data) irrespective of the weighting function considered. These projects have project ages of not more than 3 years. Results also show that the Gaussian weighting function yielded the best relative predication accuracy as compared to the remaining weighting functions (Figure 2).

B. Best Sizing and Aging for Bellwether moving window

Results from Table 1 show the window sizes and ages of the potential *Bellwether moving windows* with their respective performance measures. As a result of establishing the baseline that the Gaussian weighting function yields a relative prediction performance, we investigated which window size and age yielded the best prediction accuracy. Out of the range of 80 to 90 exemplary projects, we realized from their respective MAE that a subset of 82 to 84 projects (36.2% to 37.0% of the training data i.e. 227 projects) with their respective ages of 1.5 to 2 years yielded a relative prediction accuracy (Table 1).

Table 1: Prediction Accuracy Performance of Bellwether moving windows (exemplary projects) with respect to MAE

Window age (years)	Window size	MAE
1	80	1.5868
1	81	1.9144
1.5	82	**1.4523***
1.5	83	**1.4546***
2	84	**1.4377***
2	85	2.2069
2	86	1.6152
2	87	1.6517
2.5	88	2.4752
3	89	2.5686

'' used to indicate best prediction performance results (comparison made for each evaluation measure along each column). Thus, the minimum MAE signifies the best prediction accuracy.*

C. Statistical Comparison of the Bellwether moving window versus the Growing portfolio

To investigate the significance of the estimation accuracy performance of the *Bellwether moving window* models, we performed a pairwise statistical test to benchmark the *Bellwether moving window* models with the *growing portfolio* models. The Welch *t-test* result show that at 5% asymptotic significance level, there is a significant difference between the *Bellwether moving window* and the *growing portfolio*. This statistical significance is confirmed by the large Glass's Δ effect size value of 0.627 (>0.5) implying practical significance in the use of the *Bellwether moving window* over the *growing portfolio*.

D. Effect of weighting functions on the Bellwether Moving windows

The Kruskal-Wallis *H-test* resulted in a Chi-square value of 71.17 (*p-value<0.05*) showing the existence of statistical significant differences at 5% asymptotic significance level across the weighting functions for the *Bellwether moving windows* (Triangular, Epanechnikov and Gaussian). Similarly, incorporating the unweighted moving window (Rectangular) [10] with the weighted *Bellwether moving windows* resulted in a Chi-square value of 97.52 (*p-value<0.05*) showing the existence of statistical significant differences. This confirms result by a previous study [2] that the use of a weighting function can affect the estimation performance of a prediction model. We realized that the use of the Gaussian weighting function on the *Bellwether moving window* improves the relative prediction accuracy performance.

7. Threat to Validity

We used a single release of the ISBSG dataset (release 10) as considered in previous studies [2][7]. This dataset is a convenience sample but cannot be a general representation of all chronological datasets. Therefore, our results might not be confidently generalized beyond this dataset.

8. Conclusion and Future work

The findings show that, *Bellwether effect* exist in software effort estimation and its respective *Bellwether moving window* has a relative effective window size of 82 to 84 projects (36.2% to 37.0%) and window age of 1.5 to 2 years for improved prediction accuracy. Incorporating weighting functions on the *Bellwether moving window* affect the prediction accuracy when the window size and window age of the *Bellwether moving window* differs from the aforementioned threshold. We observed that, the Gaussian weighting function was more advantageous for software effort estimation when using moving windows. The *Bellwether moving window* will continue to be useful for prediction purposes provided the phenomenon being measured share similar characteristics with the *Bellwether* projects.

The introduced *Bellwether method* can be used in other software engineering domains for selecting the *Bellwether moving window* to be considered for the training and validation needs of predictive models. We intend to extend the *Bellwether method* to empirically analyze the Finnish dataset and other industrial projects to further validate the existence of *Bellwether moving windows.*

Acknowledgment

This work is supported in part by the General Research Fund of the Research Grants Council of Hong Kong (No. 125113, 11200015 and 11214116), and the research funds of City University of Hong Kong (No. 7004683 and 7004474).

References

[1] R. Krishna, T. Menzies, and W. Fu. "Too much automation? the Bellwether effect and its implications for transfer learning." *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*. ACM, 2016.

[2] S., Amasaki, and C. Lokan. "On the effectiveness of weighted moving windows: Experiment on linear regression based software effort estimation."*Journal of Software: Evolution and Process* 27.7 (2015): 488-507.

[3] P. A. Whigham, C. A. Owen, and S. G. Macdonell. "A baseline model for software effort estimation." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 24.3 (2015): 20.

[4] T. W. MacFarland, and J. M. Yates. "Kruskal–Wallis H-Test for Oneway Analysis of Variance (ANOVA) by Ranks." *Introduction to Nonparametric Statistics for the Biological Sciences Using R*. Springer International Publishing, 2016. 177-211.

[5] C. Lokan, and E. Mendes. "Applying moving windows to software effort estimation." *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009.

[6] R. P. Dobrow. "Markov Chain Monte Carlo." *Introduction to Stochastic Processes With R* . Wiley Online Library (2016): 181-222.

[7] C. Lokan, and E. Mendes. "Investigating the use of duration-based moving windows to improve software effort prediction: A replicated study."*Information and Software Technology* 56.9 (2014): 1063-1075.

[8] B. Zhou, H. Okamura, and T. Dohi. "Enhancing performance of random testing through Markov chain Monte Carlo methods." *IEEE Transactions on Computers* 62.1 (2013): 186-192.

[9] T. Foss, E. Stensrud and B. Kitchenham. "A simulation study of the model evaluation criterion MMRE." *IEEE Transactions on Software Engineering* 29.11 (2003): 985-995.

[10] S. Amasaki, and C. Lokan. "A replication study on the effects of weighted moving windows for software effort estimation." *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2016.

[11] Z. Tu, and S. C. Zhu. "Image segmentation by data-driven Markov chain Monte Carlo." *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002): 657-673.

[12] C. Lokan, and E. Mendes. "Investigating the use of duration-based moving windows to improve software effort prediction." *2012 19th Asia-Pacific Software Engineering Conference*. Vol. 1. IEEE, 2012.

[13] H. Fischer. *A history of the central limit theorem: From classical to modern probability theory*. Springer Science & Business Media, 2010.

[14] K. Yao, and J. Gao. "Law of large numbers for uncertain random variables." *IEEE Transactions on Fuzzy Systems* 24.3 (2016): 615-621.

[15] D. Pelleg, and A. W. Moore. "X-means: Extending K-means with Efficient Estimation of the Number of Clusters." *ICML*. Vol. 1. 2000.

[16] M. Shepperd, and S. MacDonell. "Evaluating prediction systems in software project estimation." *Information and Software Technology* 54.8 (2012): 820-827.

[17] S. Amasaki, and C. Lokan. "The effects of gradual weighting on duration-based moving windows for software effort estimation." *International Conference on Product-Focused Software Process Improvement*. Springer International Publishing, 2014.

[18] S. Amasaki, and C. Lokan. "The evaluation of weighted moving windows for software effort estimation." *International Conf. on Product Focused Software Process Improvement*.Springer Berlin Heidelberg, 2013.

[19] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, et al. "Robust statistical methods for empirical software engineering." *Empirical Software Engineering* (2016): 1-52.