2013 Joint Conference of the 23nd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura)

The Effects of Variable Selection Methods on Linear Regression-based Effort Estimation Models

Sousuke Amasaki
Department of Systems Engineering
Okayama Prefectural University
Soja, Okayama 719–1197 JAPAN
Email: amasaki@cse.oka-pu.ac.jp

Tomoyuki Yokogawa
Department of Systems Engineering
Okayama Prefectural University
Soja, Okayama 719–1197 JAPAN
Email: t-yokoga@cse.oka-pu.ac.jp

***Abstract*—Stepwise regression has often been used for variable selection of effort estimation models. However it has been criticized for inappropriate selection, and another method is recommended. We thus examined the effects of Lasso, which is one of such variable selection methods. An experiment with datasets from PROMISE repository revealed that Lasso-based selection stably selected better variables than stepwise in predictive performance. We thus concluded Lasso-based selection is preferable to stepwise regression.**

I. Introduction

Software effort estimation is one of the important activities in software development. Its accuracy has a significant effect on project success. A systematic review revealed that model-based software effort estimation has been popular [1].

Making a quality model requires careful consideration for dataset cleansing, formulation, and so on. Manual model construction is the best way to obtain a quality model. However, an automatic model construction procedure has been popular in those studies. One reason is that those studies usually evaluated the predictive performance of models with cross-validation and chronological splitting [2]. The evaluation requires many repetitions of modeling and estimation. Furthermore, public datasets used for evaluation bring little information, and their characteristics are often unknown. Thus, it is difficult to perform manual model construction.

Linear regression is one of the well-known and high performance methods for effort estimation [3]. It often takes the following form:

$$\log(\text{Effort}_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}.$$

where, x_{ij} represents jth features of a project i in a historical record. β_js represent a coefficient for the features. What features (variables) are included affects predictive performance and interpretability for a manager.

Stepwise regression has been often used for feature selection. However, it violates the principle of statistical estimation and hypothesis testing. Harrell [4] criticized stepwise regression as follows:

1) It yields R^2 values that are biased high.
2) The F and χ^2 test statistics do not have the claimed distribution.
3) Standard errors of regression coefficient estimates are biased low, and confidence intervals are falsely narrow.
4) P-values are too small and difficult to correct.
5) Regression coefficients are biased high in absolute value.
6) Not think about the problem (ignore domain knowledge).

It was also reported that stepwise regression may not identify sets of variables that fit well, even when such sets exist.

Lasso [5] is an extension of linear regression and can also be used as variable selection method. Lasso is one of the methods which overcomes the shortcomings of stepwise regression. The past study [6] used Lasso as an effort estimation model. However, it has been unveiled how adopting Lasso as a variable selection method affects on predictive performance and variable selection results.

This paper thus investigated the following questions:

1) What is different in selected models between stepwise regression and Lasso-based selection?
2) Does the difference of variable selection models affect on predictive performance?
3) Which is to be used with linear regression?

II. Variable Selection Methods

A. Stepwise Regression

Stepwise regression is a popular variable selection method in software engineering research area. It has been used for finding crucial factors and improving effort estimation models.

There are two strategies for stepwise: forward- and backward-stepwise regressions [4]. Forward stepwise regression starts with the intercept, and then sequentially add into the model the variable that most improves the fit. Backward stepwise regression starts with the full model, and sequentially deletes the predictor that has the least impact on the fit. These stepwise strategies are mixed in some software package such as `step` function in statistical software R.

Experts often use Bivariate screening. Bivariate screening examines relationship between Effort and a feature with univariate regression model or correlation coefficients. The experts consider only significant features as predictors. Bivariate

978-0-7695-5078-7/13 $26.00 © 2013 IEEE
DOI 10.1109/IWSM-Mensura.2013.24

screening also belongs to stepwise regression and has the same problem as stepwise regression.

B. Lasso-based Selection

Lasso is a shrinkage method proposed by Tibshirani [5]. Lasso shrinks the regression coefficients by imposing a penalty on their size. The lasso coefficients minimize a penalized residual sum of squares,

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \|\beta_j\| \right\}.$$

Here, $\lambda > 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage. The coefficients are shrunk toward zero. The penalty alleviates the problem that regression coefficients become large in absolute value where a linear regression model holds many variables.

In this study, we first determined λ with leave-one-out cross-validation so as to minimize residual error. Lasso implementation is fast enough to perform leave-one-out cross-validation with the datasets we used. We applied Lasso with the determined λ and extracted a selected variable set. Then, we fitted linear regression model with the variable set.

III. Datasets

A. COCOMO81 Dataset

COCOMO81 dataset was collected by Boehm [7]. COCOMO81 dataset contains 63 projects. COCOMO81 dataset holds size, COCOMO I model's effort multipliers, and development mode. The effort multipliers EM_j and size form the following equation:

$$\text{Effort} = A \cdot Size^b \cdot \prod_{j=1}^{J} EM_j.$$

The effort multipliers are originally ordinal scale. COCOMO I model definition provided corresponding continuous values for the levels. We used these values in the above formulation. Linear regression was thus used for calibrating these multipliers with log-transformed formula:

$$\log(\text{Effort}) = A' + b \log(Size) + \sum_{j=1}^{J} \beta_j \log(EM_j).$$

We adopted this formulation. In addition we added a development mode variable to the formula. Development mode is a categorical variable having three levels. We converted it into two binary variables.

B. Desharnais Dataset

Desharnais dataset was collected at Canadian software house [8]. We removed projects having missing values and used 77 projects remained. Desharnais dataset contains five size-related variables: Transactions, Entities, Unadjusted FP, Adjusted FP, and the adjustment factor. All variables except for the adjustment factor were log-transformed. Desharnais dataset also recorded years of team experience and manager experience. We added them without transformation. Desharnais dataset also contains a categorical variable indicating Language used. We converted it into binary variables.

C. Maxwell Dataset

This dataset was shown in the book by Maxwell [9]. Maxwell dataset was collected at one of the biggest commercial banks in Finland from 1985 to 1994. Maxwell dataset contains 63 projects with 22 ordinal or categorical characteristics such as application type and staff skill levels. Product size was measured in function points, and effort was measured in hours. Maxwell dataset contains 15 ordinal variables recording project characteristics such as the degree of requirements volatility. We treated them as continuous variables as well as the past study. We transformed the other nominal variables into multiple binary variables.

The rule of thumb [4] said one predictor requires 10 instances in linear regression modeling. The rule would be violated if backward stepwise regression is used. Lasso-based selection is advantageous for the situation.

IV. Experiment Settings

A. Experiment Procedure

We conducted three types of model evaluation: goodness of fit, cross-validation, and chronological splitting. Goodness of fit is a basic evaluation method for linear regression models. We applied a variable selection method to a whole dataset and evaluated the goodness of fit of linear regression model with a resultant variable set.

Cross-validation is a common procedure for evaluation of software effort estimation models. Cross-validation divides a dataset into a training set and a testing set. The variable selection methods were applied to the training set, and a linear regression models were constructed with the training set and the selected variables. The trained model estimated efforts of the testing project. The above procedure was repeated for all folds.

This study adopted leave-one-out cross-validation because of its deterministic property and suitability for small dataset. We compared predictive performance, selected variable sets, and stability of the selected variable sets. Stability is an important factor for a manager. He/She will get confused if a small addition/deletion of projects causes a large change in selected variables.

It is more appropriate to evaluate a software effort estimation model with consideration for chronological order of projects [2]. Chronological splits reflected the idea. This study conducted an experiment following [9]:

1) Split a dataset into two subsets according to project date
2) Train a software effort estimation model with the older subsets
3) Estimate efforts of the newer subsets with the model

We separated Maxwell dataset where project start year is 1991 as same as [9]. The size of testing set is 12. We also separated

TABLE I. RESULTS OF VARIABLE SELECTION ON WHOLE DATASET (COCOMO81)

	Stepwise	Lasso
(Intercept)	$^{***}1.030$	$^{***}1.059$
lKSLOC	$^{***}1.113$	$^{***}1.110$
lRELY	$^{*}1.237$	$^{*}1.284$
lDATA	1.336	1.378
lCPLX	$^{*}1.070$	$^{\cdot}0.852$
lTIME	$^{**}2.000$	$^{*}1.847$
lSTOR	–	0.294
lVIRT	$^{\cdot}1.292$	0.850
lTURN	–	0.325
lACAP	$^{**}1.814$	$^{*}1.582$
lAEXP	–	0.594
lPCAP	$^{**}1.282$	$^{**}1.416$
lVEXP	$^{**}2.932$	$^{\cdot}2.313$
lLEXP	–	1.227
lMODP	$^{*}1.319$	0.862
lTOOL	–	1.088
lSCED	$^{*}2.214$	$^{\cdot}1.876$
dev_embedded	0.252	0.270
Adjusted R^2	0.95	0.95
AIC	81.8	88.4

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{\cdot}p < 0.1$

TABLE II. PREDICTIVE PERFORMANCE (COCOMO81)

	leave-one-out		chronological	
	Stepwise	Lasso-based	Stepwise	Lasso-based
MMRE	0.461	0.438	0.775	0.723
MdMRE	0.298	0.319	0.419	0.594
PRED(25)	0.413	0.365	0.348	0.348
MMAE	250	233	318	106
MdMAE	34	35	23	15
MBRE	0.606	0.585	0.857	0.858
SA	77.6	79.1	70.0	90.0

Desharnais dataset where project end year is 1986. The size of testing set is 19. The separation year for COCOMO81 was 1977. The testing set had 23 projects.

B. Modeling Procedure

In evaluations with cross-validation and chronological splitting, we constructed linear regression models following the steps below:

1) Select variables with a training set
2) Fit a linear regression model with the selected variables and the training set
3) Remove outliers
4) If the model becomes rank deficient after outlier removal, go to 1) with the reduced training dataset with the reduced variable set.

We identified influential data points (outliers) with Cook's distance [9]. Cook's distance values were calculated for all projects. Any projects with distances higher than $3 \times 4/N$, where N represents the total number of projects, were immediately removed.

C. Evaluation Measures

We evaluated goodness of fit with adjusted R^2 and AIC. The both measures are common for evaluating the degree of fitting with correction for overfitting.

To report predictive performance, we used the following performance measures; MMRE, MdMRE, PRED(25), MBRE, MMAE, MdMAE, and SA. MMRE and MdMRE are an average and a median of *Magnitude of Relative Error*(MRE) values. MRE defines a relative error between actual and estimated efforts as follows:

$$\mathrm{MRE} = \frac{\|\mathrm{Actual}_i - \mathrm{Estimated}_i\|}{\mathrm{Actual}_i}.$$

PRED(25) represents the percentage of estimations which satisfy $\mathrm{MRE}_i < 0.25$.

MBRE is an average of *Balanced Relative Error* BRE values. BRE [10] alleviates the bias of MRE definition. BRE revises a denominator definition of MRE:

$$\mathrm{BRE} = \frac{\|\mathrm{Act}_i - \mathrm{Est}_i\|}{min(\mathrm{Act}_i, \mathrm{Est}_i)}.$$

MMAE and MdMAE are an average and a median of *Magnitude of Absolute Error* (MAE) values. MAE is the difference between actual and estimated efforts:

$$\mathrm{MAE}_{P_i} = \|\mathrm{Act}_i - \mathrm{Est}_i\|.$$

SA [11] is also defined on MAE. MAE uses MAE of random guessing P_0. The definition of accuracy measure SA_{P_i} is as follows:

$$\mathrm{SA}_{P_i} = \left(1 - \frac{\mathrm{MMAE}_{P_i}}{\overline{\mathrm{MMAE}}_{P_0}}\right) \times 100.$$

where $\overline{\mathrm{MAE}}_{P_0}$ is the mean value of MAEs of a large number runs of random guessing. MAE_{P_i} is MAE of an effort estimation method P_i. The interpretation of SA is that the ratio presents how much better P_i is than random guessing.

To compare model performance, we adopted effective size Δ, which is also proposed in [11]:

$$\Delta = \frac{\mathrm{MMAE}_{P_i} - \overline{\mathrm{MMAE}}_{P_j}}{s_{P_j}}. \quad (1)$$

where s_{P_i} is the sample standard deviation of a prediction system i. The effect size is interpreted as follows: small ($\Delta \approx 0.2$), medium ($\Delta \approx 0.5$), and large ($\Delta \approx 0.8$).

We also examined the number and types of variables. The stability of selected variable sets were examined in case of leave-one-out and chronological splitting.

V. RESULTS

A. COCOMO81 dataset

Table I shows fitted models with the selected variables. The last variable represents development mode (embedded). Note that the effort multipliers were log-transformed and prefixed by 'l'.

Stepwise regression dropped 6 variables. Most of the remained variables were statistically significant at least $\alpha = 0.05$. Contrastingly, Lasso selected all effort multipliers. Lasso-based selection only dropped one development mode (dev_semidetached). Parameters were inferred with linear regression, and quite a few of them were statistically insignificant. The significant variables selected by Lasso-based selection were also significant in the result of stepwise regression. Adjusted R^2 was equivalent, but AIC supported stepwise

TABLE III. SELECTED MODELS BY LEAVE-ONE-OUT (COCOMO81)

Methods	Formula	Frequency
Stepwise	lksloc + lrely + ldata + lcplx + ltime + lvirt + lacap + lpcap + lvexp + lmodp + lsced + dev_embedded	53
	lksloc + lrely + lcplx + ltime + lvirt + lacap + lpcap + lvexp + lmodp + lsced + dev_embedded	6
	(formulae appeared only once)	4
Lasso	lksloc + lrely + ldata + lcplx + ltime + lstor + lvirt + lturn + lacap + laexp + lpcap + lvexp + llex + lmodp + ltool + lsced + dev_embedded	58
	(formulae appeared only once)	4

TABLE IV. SELECTED MODELS IN CHRONOLOGICAL SPLITTING (COCOMO81)

Methods	Formula
Stepwise	lksloc + ltime + lstor + lturn + laexp + lpcap + lvexp + lsced + dev_semidetached
Lasso	lksloc + ldata + ltime + lstor + lturn + laexp + lpcap + llexp + lmodp + lsced + dev_embedded

TABLE V. RESULTS OF VARIABLE SELECTION ON WHOLE DATASET (DESHARNAIS)

	Stepwise	Lasso
(Intercept)	$^{**}1.482$	$^{***}1.842$
ManagerExp	–	0.041
lEntities	–	0.019
lPointsNonAdjust	$^{***}0.912$	–
lPointsAjust	–	$^{***}0.866$
Adjustment	$^{**}0.017$	0.008
Language1	$^{***}1.346$	$^{***}1.363$
Language2	$^{***}1.336$	$^{***}1.330$
Adjusted R^2	0.78	0.78
AIC	78.4	80.9

TABLE VI. PREDICTIVE PERFORMANCE (DESHARNAIS)

	leave-one-out		chronological	
	Stepwise	Lasso	Stepwise	Lasso
MMRE	0.381	0.378	0.301	0.280
MdMRE	0.312	0.325	0.301	0.280
PRED(25)	0.429	0.416	0.579	0.421
MMAE	1754	1744	1569	1367
MdMAE	1038	1045	378	651
MBRE	0.468	0.470	0.385	0.361
SA	57.3	57.6	61.9	66.9

regression. Because Lasso-based selection retained more variables than stepwise regression.

Table II shows predictive performance. With respect to leave-one-out CV, there was no clear difference between Lasso-based selection and stepwise regression in performance. However, their selections were different. Table III shows selected variable patterns during the leave-one-out evaluation. The most frequent models held the same variables as the fitted models shown in Table I. Although training sets of leave-one-out CV are almost equivalent each other, the variable selection methods produced a variety of variable sets. Stepwise regression preferred simple models while Lasso-based selection preferred complex models. Lasso-based selection was slightly more stable than stepwise selection. Lasso-based selection produced the same pattern 58 times among 77 folds. Stepwise regression produced the same pattern 53 times. Stepwise regression was less stable or sensitive for small changes.

With respect to chronological splitting, MMRE, PRED(25), and MBRE showed no clear difference. Contrastingly, MMAE and SA supported Lasso-based selection. The effect size is $\Delta = (318 - 106)/1169 \approx 0.18$, where the standard deviation of MMAE of Stepwise regression was 1169. Thus, the difference was trivial. Table IV shows selected variables by stepwise regression and Lasso-based selection. Stepwise regression selected different variables from those in leave-one-out CV. Contrastingly, the all selected variables by Lasso-based selection also appeared in the most frequent model. No new variable was added. These results implied the instability of stepwise regression.

B. Desharnais Dataset

Table V shows results of fitted models with the selected variables. Stepwise regression selected 5 variables, and the variables were statistically significant. Lasso selected 7 variables. Three out of the seven were insignificant. Both methods selected different FP-related measures. Stepwise regression took Unadjusted FP and Lasso-based selection took Adjusted FP. However, adjusted R^2 and AIC were equivalent between them.

Table VI shows predictive performance. With respect to leave-one-out CV, there was no clear difference. Table VII shows selected variable patterns during the leave-one-out CV. The most frequent models held the same variables as the fitted models shown in Table V. Regardless of variable selection methods, the top two models were equivalent except for the absence of MangerExp. Therefore, both methods selected equivalent models at most time. Stepwise regression selected slightly parsimonious sets.

With respect to chronological splitting, PRED(25) only supported stepwise regression. The other measures supported Lasso-based selection. However, the difference was trivial. Table VIII shows selected variables. The all selected variables by Lasso-based selection appeared in the most frequent model shown in Table VII. Some of the selected variables by stepwise did not appear in the most frequent model shown in Table VII. However, the difference was small.

C. Maxwell dataset

Table IX shows results of fitted models with the selected variables. Stepwise regression selected 12 variables while Lasso-based selection retained 9 variables. Lasso-based selection resulted in worse AIC. However, the number of selected variables were smaller. This might imply that the selected model was too simple. However, adjusted R^2 showed no clear difference.

Table X shows predictive performance. With respect to leave-one-out CV, Lasso-based selection was preferable. The effect size was $\Delta = (3827 - 2700)/4997 \approx 0.23$, where the standard deviation of stepwise regression was 4997. The

TABLE VII. SELECTED MODELS BY LEAVE-ONE-OUT (DESHARNAIS)

Methods	Formula	Frequency
Stepwise	lPointsNonAdjust + Adjustment + Language1 + Language2	52
	ManagerExp + lPointsNonAdjust + Adjustment + Language1 + Language2	22
	lPointsAjust + lPointsNonAdjust + Adjustment + Language1 + Language2	2
	ManagerExp + lPointsAjust + lPointsNonAdjust + Adjustment + Language1 + Language2	1
Lasso	ManagerExp + lEntities + lPointsAjust + Adjustment + Language1 + Language2	57
	lEntities + lPointsAjust + Adjustment + Language1 + Language2	19
	lTransactions + ManagerExp + lEntities + lPointsAjust + Adjustment + Language1 + Language2	1

TABLE VIII. SELECTED MODELS IN CHRONOLOGICAL SPLITTING (DESHARNAIS)

Methods	Formula
Stepwise	lPointsNonAdjust + Adjustment + lPointsAjust + Language1 + Language2
Lasso	ManagerExp + Adjustment + lPointsAjust + Language1 + Language2

TABLE IX. RESULTS OF VARIABLE SELECTION ON WHOLE DATASET (MAXWELL)

	Stepwise	Lasso
(Intercept)	$^{**}2.434$	$^{***}2.443$
Size	$^{***}0.746$	$^{***}0.768$
App1	–	-0.154
Dba1	0.335	–
Ifc	$^{***}-1.009$	$^{**}-0.838$
T02	$^{\cdot}-0.150$	–
T03	$^{\cdot}0.127$	–
T06	0.168	–
T07	0.111	0.100
T08	0.100	$^{\cdot}0.130$
T09	$^{**}0.286$	$^{*}0.219$
T10	0.125	0.115
T12	$^{*}-0.239$	–
T14	–	-0.094
Time	$^{**}-0.100$	-0.057
Adjusted R^2	0.83	0.81
AIC	81.7	87.0

TABLE X. PREDICTIVE PERFORMANCE (MAXWELL)

	leave-one-out		chronological	
	Stepwise	Lasso	Stepwise	Lasso
MMRE	0.646	0.469	0.328	0.335
MdMRE	0.429	0.317	0.372	0.216
PRED(25)	0.242	0.339	0.417	0.583
MMAE	3827	2700	1429	1145
MdMAE	2639	2005	1101	584
MBRE	0.867	0.558	0.670	0.428
SA	57.0	69.6	79.8	83.8

difference was non-trivial. Table XI shows selected variables patterns during the leave-one-out CV. The most frequent models held the same variables as the fitted models shown in Table IX. There was a clear difference in the number of patterns obtained. Stepwise regression produced 28 patterns and the most frequent models were produced only 16 times. Contrastingly, Lasso-based selection produced 3 patterns with fewer variables. This stability was preferred for interpretation. Thus, Lasso-based selection was better than stepwise regression.

With respect to chronological splitting, all measures except for MMRE supported Lasso-based selection. $\Delta = (1429 - 1145)/1396 \approx 0.20$, where the standard deviation of MMAE of stepwise regression was 1396. The effect of Lasso was small but non-trivial. Table XII shows selected models. Stepwise regression selected many variables. Many of them were not in the most frequent model shown in Table XI. Contrastingly, Lasso-based selection produced only three variables. They were all found in the most frequent model. Lasso-based selection was more stable than stepwise regression.

VI. DISCUSSION

A. What is different in selected models between stepwise regression and Lasso-based selection?

Both methods selected different sets of variables. It is generally difficult to specify a true variable selection and to evaluate which result is more appropriate than the another. With respect to COCOMO81 dataset, it was known that all effort multipliers were specified and quantified on COCOMO81 dataset. This background supports Lasso-based selection because all effort multipliers were retained in full-fit and leave-one-out CV. For Desharnais dataset, they produced almost the same variable sets. It is difficult to rank the methods. For Maxwell dataset, Lasso-based selection left fewer variables than stepwise regression. The selected variables were left in the variable set that was selected with manual stepwise regression in [9]. Lasso-based selection was also stable for small changes of a training dataset.

B. Does the difference of variable selection models affect on predictive performance?

The goodness of fit showed no clear difference though AIC preferred stepwise regression. Leave-one-out CV results showed Lasso-based selection was slightly better than stepwise regression. However, clear difference was found for Maxwell dataset only. Chronological splitting results also supported Lasso-based selection. MAE-based measures were especially improved. The improvement was subtle but firm. We thus concluded the difference affected on predictive performance.

C. Which is to be used with linear regression?

We concluded Lasso-based selection was preferable to stepwise regression because of its stability, appropriateness, and small but firm improvement in predictive performance.

VII. THREATS TO VALIDITY

This study has the same threats to validity as other software effort estimation research. First, the datasets we used are a convenience sample and may not be representative of software projects in general. Thus, the results may not be generalized beyond the dataset. We selected various types of datasets found in the past studies. Therefore, we believe the results can be useful for datasets with similar property.

TABLE XI. SELECTED MODELS BY LEAVE-ONE-OUT (MAXWELL)

Methods	Formula	Frequency
Stepwise	lSize + Dba1 + Ifc + T02 + T03 + T06 + T07 + T08 + T09 + T10 + T12 + Time	16
	lSize + Ifc + T02 + T03 + T06 + T08 + T09 + T10 + T12 + Time	7
	lSize + Har2 + Dba1 + Ifc + T02 + T03 + T06 + T08 + T09 + T10 + T12 + Time	6
	lSize + Har2 + Dba1 + Ifc + T02 + T03 + T06 + T07 + T08 + T09 + T10 + T12 + Time	5
	lSize + Ifc + T02 + T03 + T06 + T07 + T09 + T10 + T12 + Time	3
	lSize + App3 + Har2 + Dba1 + Ifc + Source + Telonuse + Nlan + T02 + T03 + T06 + T08 + T09 + T12 + T13 + T14 + Time	2
	lSize + Dba1 + Ifc + Nlan + T02 + T03 + T06 + T08 + T09 + T12 + Time	2
	(formulae appeared only once)	4
Lasso	lSize + App1 + Ifc + T07 + T08 + T09 + T10 + T14 + Time	34
	lSize + Ifc + T07 + T08 + T09 + T10 + T14 + Time	26
	lSize + App1 + Ifc + T01 + T07 + T08 + T09 + T10 + T14 + Time	2

TABLE XII. SELECTED MODELS IN CHRONOLOGICAL SPLITTING (MAXWELL)

Methods	Formula
Stepwise	lSize + App3 + Har2 + Ifc + Source + T02 + T03 + T06 + T09 + T12 + T13 + T15 + Time
Lasso	lSize + T08 + T14

Second, all the models employed in this study were built automatically. Automating the process necessarily involved making some assumptions, and the validity of our results depends on those assumptions being reasonable. However, based on our experience building models manually, we believe that these assumptions are acceptable.

VIII. RELATED WORK

Research in software effort estimation models has a long history. Jørgensen et al. conducted a systematic survey and revealed linear regression model was one of the most popular methods [1]. Furthermore, Dejaeger et al. revealed that linear regression is one of the most accurate methods [3].

The effectiveness of feature subset selection was confirmed [3], [12]. These methods are complicated but flexible. For instance, they can optimize performance measures such as MMRE directly. However, stepwise regression is easy to apply, and more popular than those methods in case of linear regression model.

Lasso has been rarely used in software effort estimation research. Nguyen et al. examined its performance as a software effort estimation model [6]. However, Lasso was not used as variable selection, and thus the authors did not concern what variables were selected, and what was different from those of stepwise regression.

IX. CONCLUSION

This study compared Lasso-based selection with stepwise regression. As a result, we found that Lasso-based selection and stepwise showed different preferences. There was a clear difference in predictive performance for some datasets. They also selected different variable sets. Lasso-based selection tended to select the same variable set with the stability. Finally, we concluded that Lasso-based selection was preferable to stepwise regression.

As future work, experiments with other datasets are needed for generality of the results. ISBSG dataset is one of candidates because it has different characteristics from the datasets we used. We also need to perform an experiment with chronological growth of a repository [2] for practical sense.

ACKNOWLEDGEMENT

This work has been conducted as a part of "Research Initiative on Advanced Software Engineering in 2013" supported by Software Reliability Enhancement Center (SEC), Information Technology Promotion Agency Japan (IPA).

REFERENCES

[1] M. Jørgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," *IEEE Trans Softw Eng*, vol. 33, no. 1, pp. 33–53, 2007.

[2] E. Mendes and C. Lokan, "Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions: a replicated study," in *Proc. of EASE 2009*, 2009.

[3] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, "Data Mining Techniques for Software Effort Estimation: A Comparative Study," *IEEE Trans. on Softw. Eng.*, vol. 38, no. 2, pp. 375–397, Mar. 2012.

[4] F. E. Harrell, *Regression Modeling Strategies*. Springer, Jan. 2001.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. Ser. B*, pp. 267–288, 1996.

[6] V. Nguyen, B. Steece, and B. Boehm, "A constrained regression technique for cocomo calibration," in *Proc. of ESEM '08*, 2008.

[7] B. W. Boehm, *Software Engineering Economics*. Prentice-Hall, 1981.

[8] J. M. Desharnais, "Analyse statistique de la productive des projets informatique a partie de la technique des point des fonction," Master's thesis, Univ. of Montreal, 1989.

[9] K. D. Maxwell, *Applied Statistics for Software Managers*. Prentice Hall, Jun. 2002.

[10] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," *Journal of Systems and Software*, vol. 27, no. 1, Oct. 1994.

[11] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology*, vol. 54, no. 8, pp. 820–827, Aug. 2012.

[12] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Information and Software Technology*, vol. 52, no. 11, pp. 1155–1166, Nov. 2010.