H. Petersson and C. Wohlin, "An Empirical Study of Experience-based Software Defect Content Estimation Methods", Proceedings International Symposium on Software Reliability Engineering, pp. 126-135, Boca Raton, Florida, USA, 1999.

An Empirical Study of Experience-Based Software Defect Content Estimation Methods

Håkan Petersson
Department of Communication Systems
Lund University
hakan.petersson@tts.lth.se

Claes Wohlin
Department of Communication Systems
Lund University
claes.wohlin@tts.lth.se

Abstract

Capture-recapture models and curve-fitting models have been proposed to estimate the remaining number of defects after a review. This estimation gives valuable information to monitor and control software reliability. However, the different models provide different estimates making it difficult to know which estimate is the most accurate. One possible solution is to, as in this paper, focus on different opportunities to estimate intervals. The study is based on thirty capture-recapture data sets from software reviews. Twenty of the data sets are used to create different models to perform estimation. The models are then evaluated on the remaining ten data sets. The study shows that the use of historical data in model building is one way to overcome some of the problems experienced with both capture-recapture and curve-fitting models, to estimate the defect content after a review.

1. Introduction

Estimation of the number of remaining defects after software reviews is an important issue from both a project management and a software quality perspective. An estimate of the remaining number of defects could help project managers to plan, control and take informed decisions regarding resource allocation and process control. From a quality perspective, an estimate is important since it would help software engineers to track, control and improve the handling of software defects. Thus, early control of software defects will have a direct effect on the reliability of the software when it is released.

Capture-recapture models and curve fitting models have been proposed to estimate the remaining number of defects after a review in [6][18]. However, most of the times the different models produce different estimates, which makes it hard to know which estimate is the most accurate. One possible solution is to identify a selection procedure, which task is to select the method most likely to give the best estimate. The problem, however, is to find a procedure working for different data sets.

If it is infeasible to find a single model which is superior, it may be possible to find a method which provides an interval. Either that it estimates an interval or that it provides boundaries of when estimates can be trusted. This paper discusses both the opportunity to provide a good estimate, and the possibility to provide an interval which can help in creating trustworthy estimates.

Two different approaches to the problems are presented in this paper. The first approach is based on subjective judgement of which models that ought to be the best or provide a certain estimate, for example, over- or underestimate for a given data set. The second approach is based on using historical data to determine how the estimation models normally behave, e.g. determine if the model has a tendency to overestimate or underestimate. The results from the analysis are then used for future predictions. The study uses most of the available defect content estimation models. Moreover, the study is based on 30 data sets from both industry and academia.

The parameters in the methods discussed are derived from 20 data sets, and are evaluated on the remaining data sets. This paper shows that the interval approach is feasible, and it also indicates that as good estimates as from the more statistically advanced models can be obtained by simply multiplying the found defects with an experience-based factor. The latter approach requires some sort of experience base. This is a disadvantage compared to the other estimation methods, which are based only on the current data set. The results are, however, promising and it is concluded that reuse of data from previous reviews provides an important input to defect content estimation. Experience-based estimations should be further studied, although some attempts along these lines already exist [15][17].

The paper is organized as follows. In Section 2, the existing defect content estimation models are introduced and the differences between them are highlighted. Section 3 discusses the idea of using subjective judgements to select suitable estimation models. The experience-based approach, i.e. the use of historical data is presented in Section 4. Finally, a discussion is provided in Section 5 and some conclusions and future directions are presented in Section 6.

2. Capture-Recapture

This paper is focused upon estimation models from which the number of defects can be estimated directly from the defect data. In other words, models based on software metrics are not discussed here, see for example [9]. The primary objective of the models is to estimate the number of defects remaining after a completed review. Two types of models have been identified for this purpose:

- Capture-recapture models, i.e. models using the overlap and non-overlap between reviewers defect detection to estimate the remaining defect content. The models have their origin in biology where it is used for population estimations [10].
- Curve fitting models, i.e. models that plot the review data from the reviewers in a predetermined way, and based on the plot fit a mathematical function. It is then used to estimate the remaining defect content [18].

These two types of models are discussed in more detail in the subsequent sections.

2.1. Capture-Recapture Models

Different capture-recapture models use different assumptions regarding reviewers and defects. Reviewers may have the same or different ability of finding defects, and the defects themselves may be equally difficult to find or not. Thus, capture-recapture models can be divided into four different types. The four types are:

1. Reviewers are assumed to have the same ability to find defects, and the different defects are found with the same probability. This type of models is denoted M0, since it neither takes the reviewers' ability nor the detection probabilities into account.
2. Reviewers are assumed to have the same ability to find defects, though different defects are found with different probabilities. This type of models is denoted Mh (variation by heterogeneity), since it takes the detection probabilities into account but not the reviewers' ability.
3. Reviewers are assumed different, i.e. they have different ability to detect defects, and all defects are found with the same probability. This type of models is denoted Mt (variation by time), since it takes the reviewers' ability into account but not the detection probabilities.
4. Reviewers are assumed different, i.e. they have different profiles for detecting defects, and different defects are found with different probabilities. This type of models is denoted Mth (variation by time and heterogeneity), since it takes both the reviewers' ability and the detection probabilities into account.

The use of the words heterogeneity and time has its origin in biology.

A figure that illustrates the assumptions for the four types of models can be found in [3]. Of the four types of models, it is quite clear that model of type four is the most realistic model. It should however be noted that the realism leads to more complicated statistical models, leaving models of type four as the most complicated ones. This also implies that it is more difficult to get stable estimates from models of type four.

Statistical estimators can be applied to the different types of models. One statistical estimator for each type of model is presented in Table 1.

Table 1: Statistical models in relation to the different types of capture-recapture models.

Reviewer ability	Detection probabilities	
	Equal	Different
Equal	M0: Maximum-Likelihood [10]	Mh: Jackknife [10]
Different	Mt: Maximum-Likelihood [10]	Mth: Chao [5]

It is assumed that the reviewers work independently of each other. For more details regarding the models refer to the references depicted in Table 1.

2.2. Curve Fitting Models

The basic principle behind the curve fitting models is to use a graphical representation of the data in order to estimate the remaining defect content. Two different types of models have been proposed [18]:

1. Decreasing model type: Models based on plotting the detected defects versus the number of reviewers that found the defects. The defects are sorted in decreasing order with respect to the number of reviewers that found a defect. This means that the plot can be approximated with a decreasing function. Both exponentially and linearly decreasing functions have been evaluated. The exponential model is introduced in [18], and the linear model is proposed in [4] as a way of coping with data sets where the exponential model failed.
2. Increasing model type: Models based on plotting the cumulative number of defects found versus the total number of detection events. For example, if the first defect is detected by five reviewers and the second by four reviewers, then the first bar is five units high and the second bar nine units high. The defects are sorted in the same order as for the model of type 1, however, plotting the cumulative number leads to that this type of model may be approximated with an increasing func-

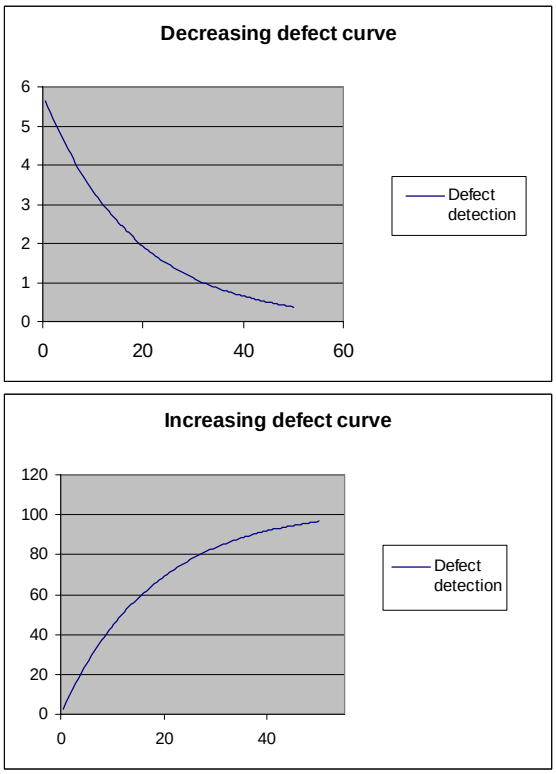

Figure 1. An illustration of curve fitting models. The exponential model is used to illustrate both a decreasing and increasing function.

tion. An increasing exponential model is proposed in [18].

The two model types are illustrated in Figure 1.

In [18], it is suggested that the estimates from the curves should be obtained as follows:

- Decreasing: The estimated defect content is equal to the defect number when the curve for the last time is above 0.5.
- Increasing: The remaining defect content is estimated to the value that the increasing curve is approaching asymptotically, minus the cumulative number of defects found so far.

3. Subjective Judgement

Curve fitting models do not need a graph but would function with only a numerical representation of the review data. The only reason for the graphs is to facilitate for a human observer. As an observer, it is easier to understand and grasp certain things if data is represented with a graph instead of with numbers. Additionally, it can be easier to discover how various things relate when studying a graph. The human eye has a great ability of identifying patterns.

A way to make more direct use of the graphical representations of the review data is to include some degree of subjective judgement. By combining subjective judgements, the human eye's ability of finding patterns and the clarifying characteristics of graphs, it may be possible to anticipate valuable characteristics of how estimators will behave when applied to the data.

To through a subjective judgement, make an estimate of how many defects that remains after a review is difficult. However, a number of other questions are valuable to know the answer of when presented with a review data set and a couple of estimators:

a) Which of the available estimators gives the most accurate value?
b) Which of the estimators gives an estimate that lies maximum a certain percentage off from the true value? This provides information about which estimate that can be trusted.
c) Which estimators underestimate and which overestimate? It is valuable to know whether the estimate represents an upper limit or a lower limit. This information can be used to create new estimates by interpolating.

An attempt was made to use the concept of subjective judgement to answer the three questions above. The idea was to let a group of test subjects compare several graphs representing review data with graphs created from data sets where the estimates had certain characteristics. For example, data sets where a certain estimator gave the most accurate value.

The experiment was designed but never run. An evaluation of the graphs that were to be compared showed that they were too similar. There was nothing left for the human eye to find patterns in. This leads to the conclusion that the choice of how to construct the graphs and what data to present are of great importance. Even if our first attempt of introducing subjective judgement into defect content estimation did not succeed, the idea is promising enough to justify further studies.

Subjective judgement can be seen as a way of introducing the use of experience. When allowing for subjective judgement, the experience and ability of the test subjects are utilised in the estimations. Another way of introducing experience is to create an experience base by collecting historical data. In the next section, this approach is used to examine how the use of historical behaviour of the estimators can be used to improve the estimates.

4. History-Based Interval Creation

4.1. Introduction

The approach of utilising subjective judgement failed. A variant of the third case is to instead of telling when an estimate overestimates or underestimates, try to find values that could confine the correct estimate between them. How should such values be determined? One possibility is to find an estimator, which always delivers overestimates and another estimator, which always delivers underestimates. This idea was mentioned in [11]. If such estimators could be found, they can be used as limits in an interval providing several interesting possibilities:

a) The interval created by the two estimators represents a 100 percent confidence interval.
b) An improved estimate could be created by interpolating between the limit values of the interval.
c) The interval could be used to cut off extreme outliers and thereby improve other existing estimators. Especially it could be used to improve the estimator MthChao that has shown good result when it comes to bias but not on variance and extreme outliers [2].

In search of estimators to be used for the limits of the interval, an investigation of the behaviour of the existing estimators is performed in Section 4.2. However, this only resulted in candidates for the lower limit of the interval. To find a candidate for the upper limit the use of experience-based multiplicative bias correction is used and presented in Section 4.3. The bias correction approach resulted in two possible models for creating the interval. These two models are then evaluated and the results are presented in Section 4.4.

4.2. Limit Estimator Candidates

To construct estimators that always overestimates or always underestimates are difficult, but estimators in use today might have the desired characteristics. They do probably not work to 100 percent but they may produce overestimates or underestimates often enough to be useful. The higher limit of the interval is the most difficult. In the case of the lower interval limit, there is already a candidate in using the number of unique defects found. The unique number of defects found always works as an underestimate (If all faults have been found, the unique number of faults is the true value and not an underestimate).

Table 2: Limit estimator candidates.

Abbrev.	Name	Ref.
M_0ML	M_0 Maximum Likelihood	[10]
M_tML	M_t Maximum Likelihood	[10]
M_hJK	M_h Jackknife	[10]
DPM	Detection Profile Method	[18]
CDPM	Cumulative Detection Profile Method	[18]

Not all the estimators presented in Section 2 are included in the investigation. The estimators that are included are listed in Table 2.

MthChao is not included because it has shown to have large variance, especially in the case of few reviewers. The MthChao is also the estimator that is to be improved by applying the interval and therefore excludes it as a candidate to be a limit estimator. The linear fit method mentioned in Section 2.2 is not included because it was only constructed as taking care of special cases in a selection algorithm presented in [4].

The CDPM has not been used or evaluated except for the work made in [18]. However, since the estimator was designed to give a high estimation it is a valuable candidate for the upper limit of the interval. Both the DPM and the CDPM have been slightly modified compared to the ones presented in [18]. Direct numerical least square curve fitting is used here. For the calculation of the MhJK the program CAPTURE [14] is used (version of 16th May 1995). Searching the literature, 30 suitable review data sets were found, see Table 3.

These data sets are collected from reviews in different contexts and also reviews conducted with different review and reading techniques such as ad hoc, checklists and perspective based reading, (PBR) [1]. It could be questioned whether capture-recapture estimators should be used with data from reviews conducted with PBR, since the assumptions of the estimators are not fulfilled. However, it has been shown that capture-recapture estimators are robust in this aspect [3][16]

The data sets were divided into two groups. Twenty data sets were randomly selected to be used for model construction, denoted with 'Fit' in Table 3, and the rest was saved to be used for evaluation of the model, denoted 'Test'. All possible combinations of three and four reviewers were created forming all possible reviews that could have taken place. These simulated reviews are called *virtual reviews*.

All the reviews originally included between five and eight reviewers except for data set No. 6, 11 and 12. To ensure that the results from these data sets did not affect the result too much, seven reviewers were randomly selected out of these data sets to represent the review.

The number of reviewers in the virtual reviews was chosen to 3 and 4. This was made mainly for three reasons:

1. Capture-recapture estimators produce less accurate values with few reviewers. The biggest need for improvements is in the case with few reviewers.

2. The size of four reviewers is recommended by [7] and mentioned in [12] as one common team size for reviews. Our own observations of reviews in industry are that they are often performed by less than four reviewers.
3. Because of the data sets originally were created by teams of between five to eight reviewers, the size of review teams with three and four reviewers maximized the number of cases created when making combinations. This gives a more reliable evaluation.

To all the created virtual reviews, the five estimators from Table 2 were applied. In a few cases, one or more of the estimators failed to estimate because of lack of overlap. This happened in 5 percent of the virtual reviews with three reviewers and 2 percent of the virtual reviews with four reviewers. These cases were removed from the investigation to ensure a fair comparison.

For each virtual review, the numbers of times the estimators overestimated and underestimated were counted and a total percentage was calculated. The results can be seen in Table 4.

Boxplots showing the estimates from the different estimators are presented in Figure 2.

The boxplots show relative error (RE) defined as:

$$RE = \frac{\text{Estimated number of defects} - \text{Actual number of defects}}{\text{Actual number of defects}}$$

The box has lines at the lower quartile, median, and upper quartile values. The whiskers have the length of 1.5 times the inter-quartile range.

Table 4: Percent overestimations and underestimations.

	% Overestimations				
No. Reviewers	M0ML	MtML	MhJK	CDPM	DPM
3	17	10	19	26	16
4	10	6	28	34	20
Total	13	8	24	30	18
	% Underestimations				
	M0ML	MtML	MhJK	CDPM	DPM
3	80	86	72	68	82
4	86	90	64	60	76
Total	83	88	68	64	79

From the boxplots, it is obvious that all the estimators tend to underestimate. Only 30 percent of the CDPM, which was designed to create high estimates, are overestimations. As for the lower limit estimator, there are several candidates The lower limit should have as many underestimations as possible and not have too many outliers. The MtML was selected because of having most underestimations, see Table 4. However, an estimator suitable for the higher limit of the interval cannot be found among these estimators.

A direct approach of using estimators as an upper limit of an interval does not work. Another possibility is to modify one of the estimators based on experience of how the estimator usually behaves and force it to estimate higher, i.e. make some kind of *bias correction*.

Table 3: Data sets.

No.	Name	Nbr. of reviewers	Used for	Ref.	No.	Name	Nbr. of reviewers	Used for	Ref.
1	AdhAtmJun	8	Fit	[8]	16	PbrNANov	6	Fit	[8]
2	AdhAtmNov	6	Fit	[8]	17	PbrNBJun	7	Fit	[8]
3	AdhPgJun	6	Fit	[8]	18	PbrNBNov	6	Test	[8]
4	AdhPgNov	6	Fit	[8]	19	PbrPgJun	8	Fit	[8]
5	ChklATM	6	Test	Unpubl.[a]	20	PbrPgNov	6	Test	[8]
6	EngDMod	7 (22)	Test	[17]	21	PbrStatA	8	Fit	[8]
7	NasaAJun	7	Fit	[8]	22	PbrStatB	7	Fit	[8]
8	NasaANov	6	Fit	[8]	23	PbrTextA	8	Test	[8]
9	NasaBJun	6	Test	[8]	24	PbrTextB	7	Fit	[8]
10	NasaBNov	6	Test	[8]	25	PbrZinsA	8	Fit	[8]
11	PBRAtmMod	7 (15)	Fit	[13]	26	PbrZinsB	7	Fit	[8]
12	PBRPgMod	7 (15)	Test	[13]	27	Cdata3A	5	Fit	[15]
13	PbrAtmJun	6	Fit	[8]	28	Cdata4A	5	Test	[15]
14	PbrAtmNov	6	Fit	[8]	29	Cdata5A	5	Test	[15]
15	PbrNAJun	6	Fit	[8]	30	Cdata6A	5	Fit	[15]

a. Used in [13] though the data set is not published.

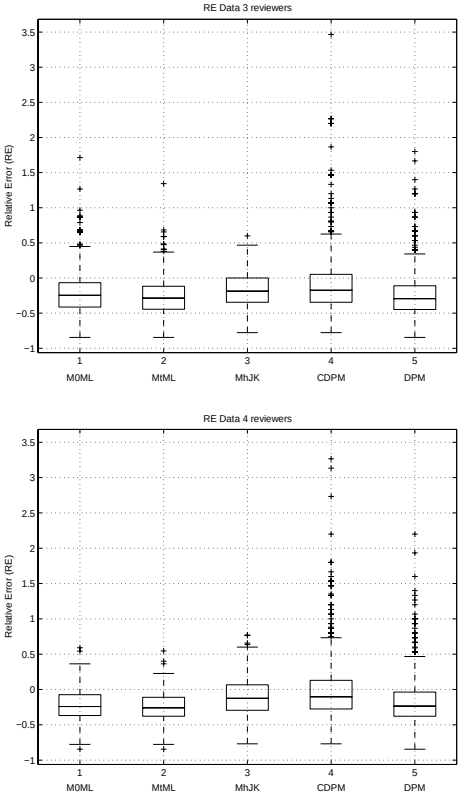

Figure 2. Boxplots showing relative error of the estimator candidates.

4.3. Experience-Based Multiplicative Bias Correction

The most direct approach of bias correction is to multiply the estimate with a factor x. However, this leads to a direct increase in variance (var(cX) = c^2 var(X)). An alternative, to use direct multiplication, would be to use the number of unique faults found, denoted D, and add x*D to the estimate. This also leads to an increase in variance since D is stochastic too. An evaluation of which of these two approaches is most suitable to use for bias correction revealed that using $\hat{N}$+x*D, where $\hat{N}$ denotes the estimation, gives slightly lower variance and was used for the higher limit. As for the estimator to use for the calculation of $\hat{N}$, MhJK is chosen. The main reasons for choosing MhJK are that it has shown promising results in other studies, for example [2], and it shows a good combination of mean value and variance for the fit data sets.

A simpler approach of creating an upper limit that does not rely on any estimator at all is to multiply D with a factor x. This approach could be used for creation of the upper limit as well as the lower limit.

Table 5 shows the result of the two different approaches when using different values as the constant x. Different values of the constants have to be used for 3 and 4 reviewers. To avoid forcing the mean value of the limits too high but still keep as many overestimations respective underestimations as possible the value of 80 percent is chosen as a threshold for when to accept the estimator as a limit. Based on this there are two models for the interval to evaluate:

1. 3 reviewers: Using $\hat{N}$+0.85*D as upper limit and MtML as lower limit

   4 reviewers: Using $\hat{N}$+0.65*D as upper limit and MtML as lower limit
2. 3 reviewers: Using 2.30*D as the upper limit and 1.40*D as the lower limit

   4 reviewers: Using 1.95*D as the upper limit and 1.30*D as the lower limit

These two models are evaluated according to the three, in Section 4.1, listed possibilities, a-c.

Instead of using bias correction to create upper and lower limits, the bias correction parameters could instead be tuned to place the mean value at zero, e.g. remove the bias. This can be made for model 1 as well as model 2. The parameter values to achieve this have also been marked in Table 5, and are evaluated in the next subsection.

4.4. Evaluation Results

To evaluate the two methods, they are applied to the review cases created out of the ten remaining data sets, marked with 'Eval' in Table 3. There are two main models that are evaluated, model 1, $\hat{N}$+x*D, and model 2, x*D. There two models is examined in four different aspects:

1. How well does the created interval cover the correct value?
2. If we make an interpolation of the limits, (high+low)/2, how good estimates does this *interpolation model* produce?
3. If we change the parameters in model 1 and 2 to instead remove the bias, how good estimates does this *bias removal model* produce?
4. If we apply the interval on MthChao and uses the limits value instead of MthChao's estimate when MthChao falls outside the interval, how good estimates does this *limiting model* produce?

4.4.1. Interval. Table 6 shows the mean width of the interval expressed in terms of relative error. Increasing the number of reviewers makes the interval width smaller.

Table 5: Parameter calibration for bias correction.

3 reviewers Nhat+x*D

x	*%Over*	*% Under*	*Mean*	*Var*
0.25	43,7	55,5	-0,045	0,084
0.30	47,6	51,4	-0,017	0,089
0.35	50,9	49,1	0,012	0,094
0.40	53,9	43,9	0,041	0,099
0.45	58,5	41,5	0,069	0,104
0.75	73,9	23,7	0,242	0,137
0.80	77,6	22,0	0,270	0,144
0.85	79,9	20,1	0,299	0,150
0.90	81,4	18,6	0,328	0,156
0.95	82,6	17,4	0,356	0,163

4 reviewers Nhat+x*D

x	*%Over*	*% Under*	*Mean*	*Var*
0.10	42,9	56,4	-0,050	0,079
0.15	45,4	54,1	-0,018	0,083
0.20	51,0	48,5	0,014	0,086
0.25	54,3	43,6	0,047	0,090
0.30	62,2	37,5	0,079	0,094
0.55	78,0	22,0	0,240	0,117
0.60	78,9	19,2	0,272	0,122
0.65	82,6	17,4	0,304	0,127
0.70	83,6	15,5	0,336	0,132
0.75	85,2	14,6	0,368	0,138

3 reviewers x*D

x	*%Over*	*% Under*	*Mean*	*Var*
1.30	14,3	85,7	-0,254	0,052
1.35	15,5	83,3	-0,225	0,056
1.40	18,6	81,4	-0,197	0,060
1.45	20,8	78,2	-0,168	0,065
1.50	24,4	72,5	-0,139	0,069
1.65	39,9	60,1	-0,053	0,084
1.70	42,3	54,9	-0,024	0,089
1.75	49,7	50,3	0,004	0,094
1.80	50,3	47,6	0,033	0,100
1.85	55,1	44,9	0,062	0,106
2.20	75,4	24,6	0,263	0,149
2.25	77,6	21,2	0,291	0,156
2.30	78,8	21,2	0,320	0,163
2.35	78,8	21,2	0,349	0,170
2.40	82,1	17,9	0,377	0,178

4 reviewers x*D

x	*%Over*	*% Under*	*Mean*	*Var*
1.20	13,8	86,2	-0,228	0,041
1.25	15,8	83,1	-0,196	0,045
1.30	19,4	80,6	-0,163	0,048
1.35	24,8	72,5	-0,131	0,052
1.40	30,8	69,2	-0,099	0,056
1.45	37,2	61,2	-0,067	0,060
1.50	44,6	53,3	-0,035	0,064
1.55	51,3	48,7	-0,003	0,069
1.60	55,3	43,6	0,030	0,073
1.65	58,9	41,1	0,062	0,078
1.85	74,7	25,3	0,191	0,098
1.90	74,8	21,7	0,223	0,103
1.95	83,4	16,6	0,255	0,109
2.00	83,4	16,1	0,287	0,114
2.05	83,9	16,1	0,319	0,120

Model 2 has slightly narrower intervals, which affects the number of times its interval covers the correct value.

The percentage of how many times the interval manages to cover the correct value is shown in Table 7. Model 1 is the more successful of the two. This is mainly due to the high bias of model 2's lower limit.

4.4.2. Interpolation and bias removal. Combining model 1 and 2 with the interpolation (int) and bias removal (rem) model creates four variants. Boxplots over these variants estimates are shown in Figure 3. The other estimators are included for comparison.

Table 6: Mean width of the interval (RE).

	Model 1		Model 2	
	3 rev.	4 rev.	3 rev.	4 rev.
Width	0.72	0.55	0.66	0.45

Table 7: Percentage of intervals that contains correct value.

	Model 1		Model 2	
	3 rev.	4 rev.	3 rev.	4 rev.
Covering	73 %	66 %	34 %	31 %

Table 8 shows the variance and bias (mean value), of our four models' estimations and Table 9 shows the variance and bias for the other estimators in the boxplots.

There is an improvement of bias level for model 1 compared to the unmodified regular models. The improvement is made to the cost of increased variance. Model 2's variance is lower but its bias higher. The only model of the five regular estimators that has a bias as good as model 1 is CDPM. However, its variances very large.

4.4.3. Using Interval to find reliable estimates. The final possibility, proposed in Section 4.1, is to use the interval to cut off MthChao. The idea is to get rid of the extreme outliers that MthChao produced though still keep its good bias. The limits are applied in such a way that if MthChao's estimate is higher or lower than respective limit, the limit's value is used instead.

A boxplot showing the estimate of MthChao and the two limiting models based on model 1 and 2, are shown in Figure 4. The approach of cutting of MthChao seems to succeed. Both limiting models 1 and 2 (Marked as Enh M1 and Enh M2 in Figure 4), manage to keep MthChao's bias level and cuts off most of the outliers. Table 10, shows to what

Table 8: Variance and bias for the four models.

		Model 1		Model 2	
		3 rev.	4 rev.	3 rev.	4 rev.
Interpolation	Variance	0.0944	0.0710	0.0852	0.0691
	Bias	0.0499	-0.0518	0.3639	0.1142
Bias Removal	Variance	0.0979	0.0813	0.0777	0.0652
	Bias	0.0428	-0.0891	0.2858	0.0617

Table 9: Variance and bias for the estimators.

	3 reviewers				
	M0ML	MtML	MhJK	CDPM	DPM
Variance	0.0614	0.0583	0.0709	0.2852	0.1645
Bias	-0.2561	-0.3075	-0.2144	-0.0718	-0.2143
	4 reviewers				
	M0ML	MtML	MhJK	CDPM	DPM
Variance	0.0412	0.0409	0.0674	0.1996	0.1300
Bias	-0.2790	-0.3297	-0.2260	-0.0699	-0.2108

degree MthChao lies outside the interval and how often MthChao produces a better estimate itself.

4.5. Summary of Results

Of the two models evaluated, model 1 ($\hat{N}$ +x*D) shows the most promising results in the evaluations conducted in Section 4.4.1 to Section 4.4.3. It manages to confine the true value in approximate 70 percent of the cases. It also shows improved bias levels for both the interpolation and bias removal approach with only a small increase in variance, and it manages to cut off most of MthChao's outliers and still keep MthChao's good bias level. These results illustrate that the use of historical data in the estimations process is a feasible way. This is further elaborated in Section 5. However, it must be remembered that these improvements are at the cost of using experience in the form of data. The estimators described in Section 2, all have the advantage of being able to be used without knowledge of earlier estimations.

It should also be noted that there is a threat to this evaluation of the models. The models are built and evaluated on 30 reviews. However, it is not based on 30 totally different documents. Some of the reviews were performed on the same or slightly modified documents. This would lead to a restriction on the possible variance in the estimates as well as in the number of found faults D. The effect of this threat is increased by using virtual reviews to create many combinations because the way of combining reviewers also narrows the possible variation. However, this threat is equal to all the estimators and the comparisons between estimators are made relative each other. Therefore should this threat not affect the overall result.

Table 10: Percentage of how often MthChao lies outside the interval.

	Model 1		Model 2	
	3 rev.	4 rev.	3 rev.	4 rev.
Outside Interval	42 %	22 %	71 %	68 %
Outside and better	0 %	5 %	18 %	20 %

5. Discussion

In Section 3, we started with introducing the idea of utilising the concept of subjective judgement in estimations of remaining defect content. The idea was to see whether valuable information about how the estimators should behave could be found when studying a graphical representation of the review data set. The approach we tried failed, however the idea of selective judgement should be further explored as a new approach to defect content estimations. To be able to improve the estimations, especially when only few reviewers are available, it seems as if we have to try other approaches than the traditional capture-recapture methods. In [18], one such new approach was introduced with the introduction of the Detection Profile Method. The aim of defect content estimation research should be to improve capture-recapture methods and curve fitting methods, but also to introduce new approaches to give us a wider range of estimation tools. By getting a wider range of tools, we can increase the amount of information used when producing the estimate.

One such tool, however not new, is to use historic information to calibrate the estimates. However, with historical information we loose the advantage of being able to make an estimate only relying on the current review information.

In Section 4.2, our initial goal was to find estimators which always underestimate or always overestimates. However, these kind of estimators was not to be found among the ones we evaluated. All the estimators had a general tendency to underestimate. However, if such estimators could be found we could use them to improve our estimations without any need for historical information. When this failed, we applied historical information to build an experience base that could aid the estimators in creating the intervals. We also used historical data without applying it to an estimator but using it to create a new estimator by multiplying the number of found defect with a factor. Both of the approaches show promising results, as shown in Section 4.4. It is important to remember that it is not the specific values

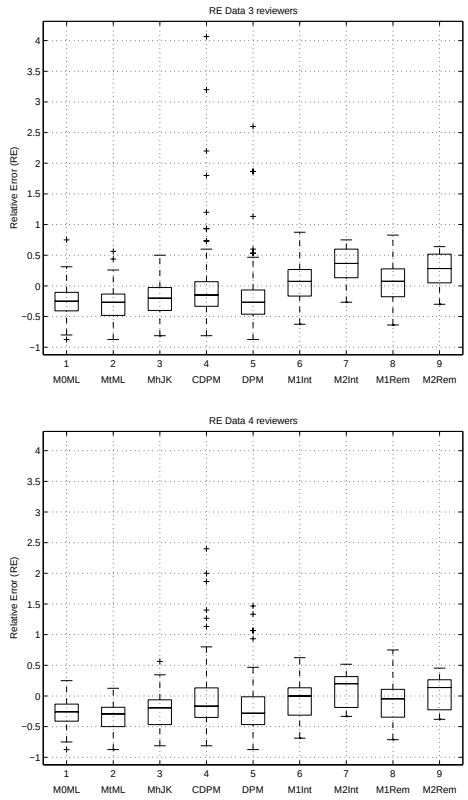

Figure 3. Boxplots for interpolation models and bias removal models[a].

a. The boxplots show the median value of the estimates and not the mean value.

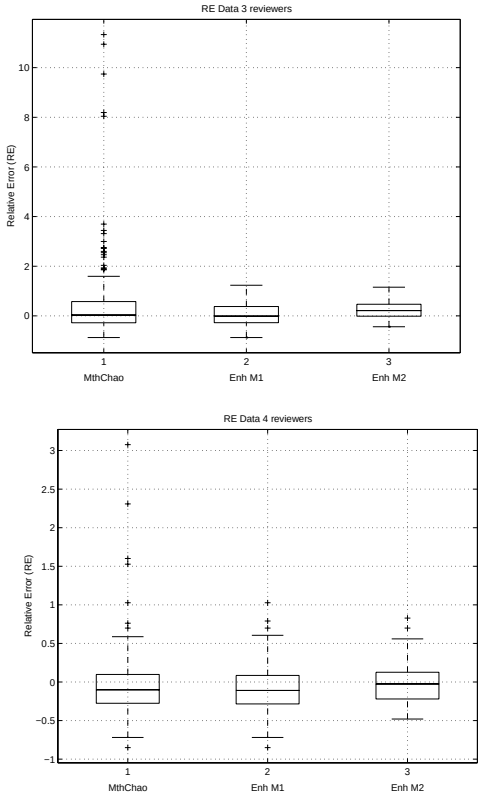

Figure 4. Boxplots showing MthChao and limiting models.

of the parameters in our models that are important, but the approach of using old data to aid in new estimations.

This paper shows that in general, both the capture recapture estimators and the curve fitting approaches tend to underestimate with one exception, MthChao. When performing a review it is most likely that both the reviewers' ability to find defects and how difficult the defects are to find, are different. This leaves Mth as the model to best describe the real world case which this paper indicates when it comes to bias level, see Section 4.4.3. However, MthChao has in earlier studies and in this study shown too large variance and shown the characteristic of producing many extreme outliers. The model becomes more stable with increased number of reviewers but if the capture-recapture models are to be used in industry, the methods must work in the cases with few reviewers too. As shown in Section 4.4.3 it is possible to improve MthChao with the use of historical data too.

6. Conclusion

To provide an estimation of the remaining number of defects after a software review would provide support to software development. Thus the types of methods discussed in this paper should be included in the software review process to allow for improved quality control. The estimation should be used as one of the inputs to facilitate an informed decison regarding the appropriate action after a review, for example, adding a reviewer performing a review or approve the review. The inclusion of these types of estimation methods in the review process calls for a number of practical conciderations, for example, treatment of different defect types and of the number of reviewers that is cost-effective to use. Both of these issues are directly related to the ability of the estimation methods to provide reliable estimates. For example, reviews conducted with few reviewers often lead to unreliable estimations. We have shown in this paper that the estimators in general tend to underestimate for reviews conducted with three or four reviewers. This leads to the conclusion that the estimators need more information in or-

der to provide estimations that are more accurate. This paper has in particular highlighted the following:

- New approaches to aid with the estimations should be created and explored with subjective judgement as one possibility,
- The estimators in general tend to underestimate except for MthChao,
- It is possible to improve the current estimators including MthChao with the utilisation of historical data.

Historical data is one way to overcome some of the problems experienced with the estimations as indicated by the results presented in this paper.

Acknowledgement

We would like to thank Per Runeson and Thomas Thelin at the Department of Communication Systems, for their valuable comments on this paper. This work was partly funded by The Swedish National Board for Industrial and Technical Development (NUTEK), grant 1K1P-97-09673.

References

[1] V. R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørumgård and M. V. Zelkowitz: “The Empirical Investigation of Perspective-Based Reading”, *Empirical Software Engineering: An International Journal*, Vol. 1, No. 2, pp. 133-164, 1996.

[2] L. Briand, K. El Emam, B. Freimut and O. Laitenberger: ”Quantitative Evaluation of Capture-Recapture Models to Control Software Inspections”, In *Proc. of the 8:th International Symposium on Software Reliability Engineering*, pp. 234-244, 1997.

[3] L. Briand, K. El Emam B. G. Freimut and O. Laitenberger: “A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content”. Fraunhofer Institute for Experimental Software Engineering, International Software Engineering Research Network Technical Report, ISERN-98-31,1998.

[4] L. Briand, K. El Emam and B. Freimut: “A Comparison and Integration of Capture-Recapture Models and the Detection Profile Method”. In *Proc. of the 9:th International Symposium on Software Reliability Engineering*, 1998.

[5] A. Chao, S. M. Lee and S. L. Jeng: “Estimating population size for capture-recapture data when capture probabilities vary by time and individual animal”. *Biometrics*, vol 48 pp. 201-216, march 1992

[6] S. Eick, C. Loader, D. Long, L. Votta and S. Vander Wiel: “Estimating Software Fault Content Before Coding”. In *Proc. of the 14th International Conference on Software Engineering*, pp. 59-65, 1992.

[7] M. E. Fagan: “Design and Code Inspections to Reduce Errors in Program Development”, *IBM System Journal*, Vol. 15, No. 3, pp 182-211, 1976.

[8] B. Freimut, “Capture-Recapture Models to Estimate Software Fault Content”. Diploma Thesis, University of Kaiserslautern, Germany, June 1997

[9] N. Ohlsson and H. Alberg: “Predicting fault-prone software modules in telephone switches”, In *IEEE Transactions on Software Engineering*, 22(12), pp. 886-894, 1996.

[10] D. Otis, K. Burnham, G. White and D. Anderson: “Statistical Inference from Capture Data on Closed Animal Populations”. *Wildlife Monographs*, No. 62, October 1978.

[11] H. Petersson, and C. Wohlin: “Evaluation of using Capture-Recapture Methods on Software Review Data”. In *Proc. Third International Workshop on Empirical Assessment and Evaluation in Software Engineering*, Keele University, 1999

[12] A. A. Porter, H. P. Siy, C. A. Toman and L. G. Votta: “An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development”, In *IEEE Transactions on Software Engineering*, Vol. 23, No. 6, pp 329-345, June 1997.

[13] B. Regnell, P. Runeson, and T. Thelin “Are the Perspectives Really Different? - Further Experimentation on Scenario-Based Reading of Requirements”, Technical Report CODEN: LUTEDX(TETS-7172) / 1-38 / 1999 & local 4, Dept. of Communication Systems, Lund University, 1999.

[14] E. Rexstad and K. P. Burnham, *User’s guide for interactive program CAPTURE*, Colorado Cooperative Fish and Wildlife Research Unit, Colorado State University, Fort Collins, CO 80523, USA, 1991.

[15] P. Runeson and C. Wohlin: “An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections”, *Empirical Software Engineering: An International Journal*, Vol. 3, No. 4, pp. 381-406, 1998.

[16] T. Thelin and P. Runeson, "Capture-Recapture Estimations for Perspective-Based Reading – A Simulated Experiment", In *Proc. of the Conference on Product Focused Software Process Improvement*, Oulu, Finland, pp. 182-200, 1999.

[17] C. Wohlin, P. Runeson and J. Brantestam: “An Experimental Evaluation of Capture-Recapture in Software Inspections”, In *Journal of Software Testing, Verification and Reliability*, Vol. 5, No. 4, pp. 213-232, 1995.

[18] C. Wohlin and P. Runeson: “Defect Content Estimations from Review Data”, In *Proc. of the 20th International Conference on Software Engineering*. pp. 400-409, 1998.