Predicting Fault Numbers via Testing

Marc Roper
Dept. Computer and Information Sciences
University of Strathclyde
Glasgow, U.K.

Motivation

Even after testing there are still going to be faults in software - how do we estimate how many are left?

Useful for:

- Reliability estimation
- Decision making
  - Release?
  - Test further?
- Automated debugging
- etc...

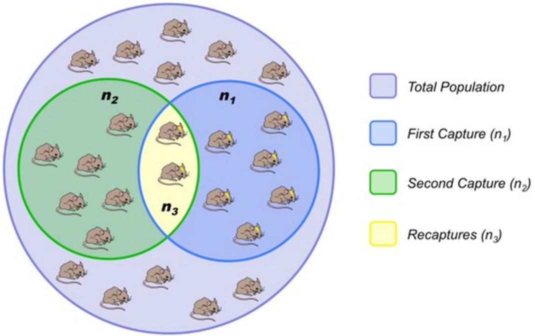
Aim of this talk is to look at the applicaiton of capture-recapture techniques to generate predictions of faults remaining after testing

Capture-Recapture Techniques

Used by population ecologists to estimate number of particular species in an area

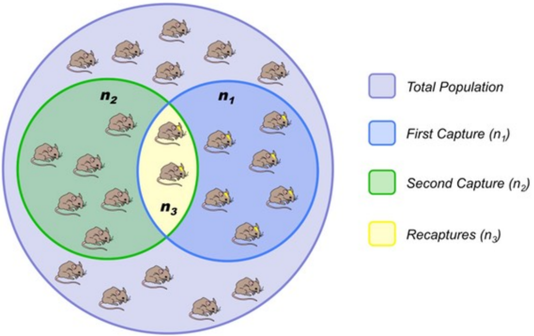
Capture-Recapture Techniques

Used by population ecologists to estimate number of particular species in an area



Capture-Recapture Techniques

Used by population ecologists to estimate number of particular species in an area



- Number of iterations is variable: animals may get caught 0 - many times
- Data fed into capture-recapture models to make predictions about total population in the area

Mapping Capture-Recapture Techniques to Software Testing

- Animal = Fault
- Capture Method = Testing Technique
- Trapping Occasion = Independent Application of a Testing Technique (just 2 considered)

Number of models explored which incorporate different assumptions:

M_t Probability of fault detection differs between techniques/testers

M_h Probability of fault detection differs between faults

M_{th} Combines the above so that both faults and techniques/testers become sources of variation

Mapping Capture-Recapture Techniques to Software Testing

Various estimators for these models, e.g.

Lincoln-Petersen estimator for M_t

$\hat{N} = \frac{n_1 n_2}{m_1}$

Mapping Capture-Recapture Techniques to Software Testing

Various estimators for these models, e.g.

Lincoln-Petersen estimator for M_t

$\hat{N} = \frac{n_1 n_2}{m_1}$

Five considered in total:

- **LP** Lincoln-Peterson estimator for M_t
- **JK** Jackknife estimator for M_h
- **CMT** Chao estimator for M_t
- **CMH** Chao estimator for M_h
- **CMTH** Chao estimator for M_{th}

Experimental Evaluation

Data:

System	Size (loc)	Total Faults	No. Testers	Testing Strategy
Strathclyde1	˜200	8	47	Functional, Structural
Strathclyde2	˜200	9	47	Functional, Structural
Strathclyde3	˜200	8	47	Functional, Structural
Myers	63	15	30*	Functional, Structural
Space	6218	38	30**	Statement coverage +

* split into 2 groups of 15

** simulated, split into 3 groups of 10

Experimental Evaluation

Data:

System	Size (loc)	Total Faults	No. Testers	Testing Strategy
Strathclyde1	˜200	8	47	Functional, Structural
Strathclyde2	˜200	9	47	Functional, Structural
Strathclyde3	˜200	8	47	Functional, Structural
Myers	63	15	30*	Functional, Structural
Space	6218	38	30**	Statement coverage +

* split into 2 groups of 15

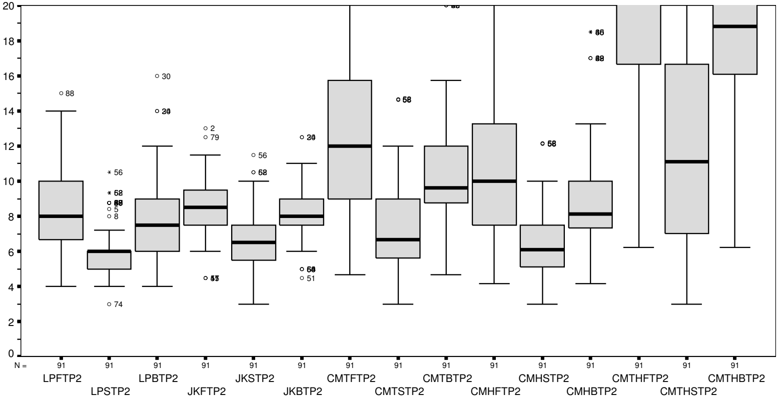
** simulated, split into 3 groups of 10

Method:

- Each of the five estimators evaluated using fault data revealed by each tester
- Every possible pairwise combination considered

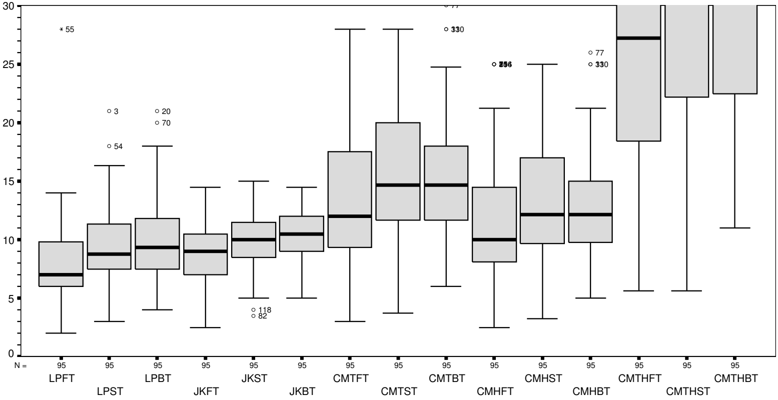
Results (Strathclyde)

Values of Estimators for Program2 (N=9):

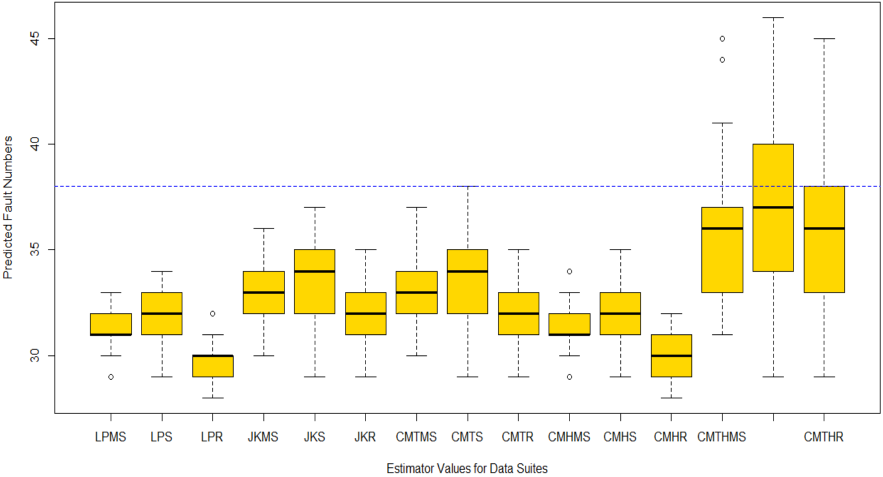


Results (Myers)

Values of Estimators for Myers Data (N=15)

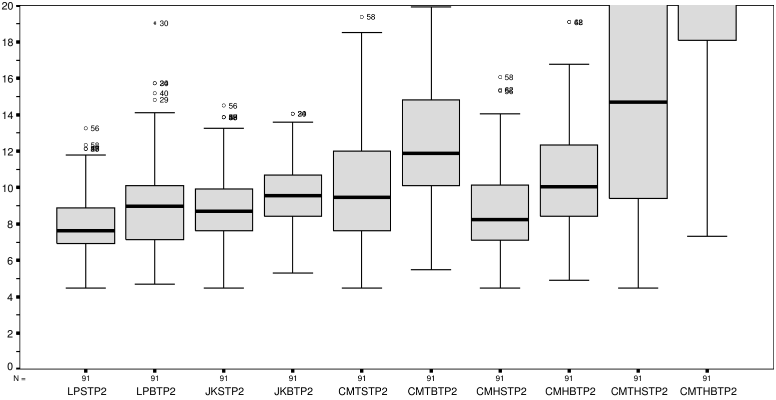


Values of Estimators for Space Data (N=38)



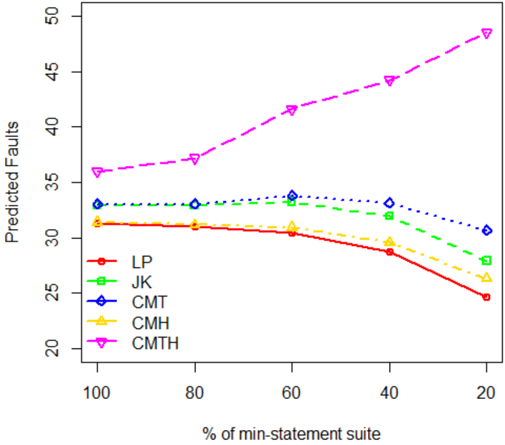
Impact of Coverage Data

Strathclyde data scaled according to coverage (N=9)



Impact of Coverage Data

SPACE data scaled according to coverage (N=38)



Conclusions

Capture-recapture techniques show promise as predictors of faults numbers

- Performance of the models tends to vary between datasets
- Findings illustrate the estimators' susceptibility to the data being used – in particular the pattern of overlap and distinctiveness between faults
- Diversity generated by different testers using different testing approaches are likely to generate the most accurate results
- However, *too much* diversity with respect to the overlap amongst faults may also lead to inaccurate estimates.

Future work will look at more datasets and how random or evolved data can be used to improve the accuracy of estimates