Replication study on "Reducing False Alarm in Software Defect Prediction by Decision Threshold Optimization" Ayse Tosun, Ayse Bener

Parisa Lak

DSL-April 6^{th}, 2018

Data Science Lab, Ryerson University

Ryerson University
Data Science Institute

Description of the Original Study

- Authors: Dr.Ayse Tosun, Dr.Ayse Bener
- Presented: Third International Symposium on Empirical Software Engineering and Measurement
- Published: 2009

Outline

- Introduction
- Background
- Information about the original study
- Information about replication
  - Motivation
  - Level of interaction with the original authors
  - Change of original study
- Comparison of replication results with the original results
- Conclusion across studies

Introduction

- Imbalanced Nature of Software data
  - Mis-classification costs are unequal and class distributions are highly skewed
  - False Alarm: marking safe modules as defective
    - Waste developer's time
  - False Negative: fail to predict the defective files
    - Expensive and hard to fix failures
- Propose a simple and powerful analysis on ROC curves to improve the prediction performance of Naive Bayes classifier

Background

- Imbalance data mitigation techniques
  - Sampling Strategies [2]
    - Does not always improve prediction
  - nearest neighbor sampling on cross-company data [4]
    - outperforms simple sampling techniques
  - ROC curve based on over sampling or under sampling are similar to adjusting decision thresholds [3]
    - ROC shows the true positive rate (TPR) vs. false positive rate(FPR) by adjusting the threshold [1]

Information about the Original Study

- 13 publicly available datasets from Promise Repository
- 9 experiments using different thresholds for Naive Bayes Decision
- Result shows the reduction in false alarm from 34% to 23%

Information about the Replication

- Motivation
  - The optimization in BPR is based on AUC that is the area under the ROC curve
  - Naive Bayes classification is simple and powerful classification algorithm using Bayes theory
    - $p(c_i|x) = \frac{p(x|c_i)p(c_i)}{p(x)}$
    - In the case of binary classification

$$\begin{cases} c_1 & if \quad p(c_1|x) \geq 0.5 \\ c_2 & Otherwise \end{cases}$$

Information about the Replication

- Level of Interaction with the original author
  - Emails to clarify discrepancies observed in the data
  - Clarification of the approach
  - Failed to retrieve the original code

Information about the Replication

- Changes from Original Study
  - Some different datasets in terms of attributes, number of observations, ...
  - Replicated codes based on the Pseudo-code available at github

Data from the Original Study

- 13 publicly available datasets from Promise Repository
- ar1,3,4,5,6 from white-goods manufacturer
- kc1,2,3 , pc1,3, mc2 and jm1 from NASA MDP
- Defect rates between 7% to 35 % $\rightarrow$ imbalance nature of the datasets
- Number of static code attributes: 21 to 39
- Class label 1: defective, 0: defect-free

Data for Replication Study

Table 1: Datasets used in this study

	# defectives	defect-free	defect rate	# attributes
ar1	9	112	7.4	29
ar3	8	55	12.7	29
ar4	20	87	18.7	29
ar5	8	28	22.2	29
ar6	15	86	14.9	29
jm1	2106	8779	19	21
kc1	326	1783	15	21
kc2	107	415	20	21
kc3	36	158	19	39
mc2	44	81	35	39
pc1	61	644	9	37
pc3	134	943	12	37

- ar, jm1, kc2, mc2, pc1, pc3 datasets are updated on Feb2, 2016
- kc1 updated on November 15,2015
- kc3 updated on November 2011

Data Comparisons

	# defectives	#defect-free	defect rate	# attributes
ar1	2103	8777	19.33	21
ar3	326	1783	15.46	21
ar4	107	415	20.50	21
ar5	43	415	9.39	39
ar6	52	109	32.30	39
jm1	9	112	7.44	29
kc1	8	55	12.70	29
kc2	20	87	18.69	29
kc3	8	28	22.22	29
mc2	15	86	14.85	29
pc1	77	1032	6.94	37
pc3	160	1403	10.24	37

Figure 1: Data from Original Study

	# defectives	defect-free	defect rate	# attributes	Original Paper
ar1	9	112	7	29	jm1
ar3	8	55	13	29	kc1
ar4	20	87	19	29	kc2
ar5	8	28	22	29	kc3
ar6	15	86	15	29	mc2
jm1	2106	8779	19	21	?ar1
kc1	326	1783	15	21	ar3
kc2	107	415	20	21	ar4
kc3	36	158	19	39	?
mc2	44	81	35	39	?ar5
pc1	61	644	9	37	?ar6
pc3	134	943	12	37	?

ROC Analysis

Table 2: Confusion Matrix

Actual	**Predicted**	
	Defective	**Defect-free**
Defective	TP	FN
Defect-free	FP	TN

- Probability of detection(pd) or TP Rate $= \frac{TP}{TP+FN}$ (3)
- Probability of false alarm(pf) or FP Rate $= \frac{FP}{FP+TN}$ (4)

Finding Optimal Threshold

- In a ROC curve
  - point (0,0) $\rightarrow$ all instances as defect-free
  - point (1,1) $\rightarrow$ all instances as defective
  - point (0,1) $\rightarrow$ perfect classification
- $distance(pd, pf) = \sqrt{(1 - pd)^2 + (0 - pf)^2}$ (6)
- $balance = 1 - distance$
- Optimal Decision Point $t_{opt} = argmin_t distance(pd, pf)$ (7)

Experiment

```
1: Dataset = {ar1, ar3, ar4, ar5, ar6, jm1, kc1, kc2, kc3, mc2, pc1, pc3, pc4}
2: Pre-processing = {None, Log Filtering}
3: Threshold = [0:0.1:1]
4: for all D in Dataset do
5:         Apply pre-processing on D
6:         for iteration=1 to 10 do
7:                 D ← Shuffle data D
8:                 for bin=1 to 10 do
9:                         Train ← take the first nine bins from D
10:                        Test ← take the last bin from D
11:                        Predict ←NaiveBayes (Train, Test)
12:                end for
13:                Predictions ← Predict
14:        end for
15:        for all P in Predictions do
16:                for all t in Threshold do
17:                        if p(C1|x) ≥ t then
18:                        %change the prediction in Predictions array
19:                                P ← C1
20:                        else (p(C1|x < t)
21:                                P ← C2
22:                        end if
23:                end for
24:        end for
25:        Calculate TP, FP, distance using Eq. (3), (4), (6)
26:        Find optimum threshold that gives the minimum distance using Eq. (7)
27: end for
```

Figure 2: Pseudocode for the experiment.

ROC for kc1 dataset

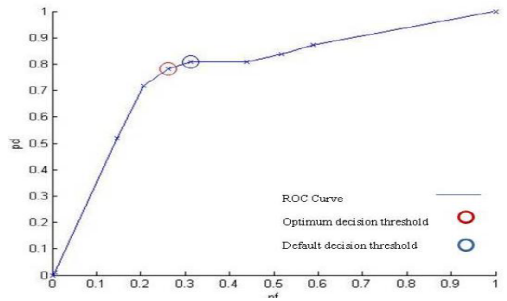

Figure 3: ROC curve for kc1 - Original

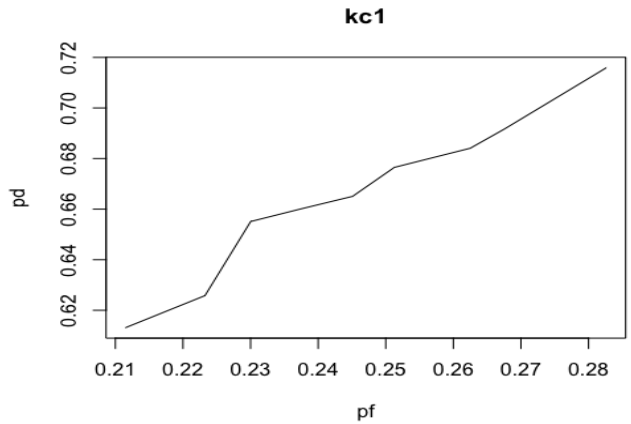

Figure 4: ROC curve for kc1 - Replicated

Replicated Results

	NB with default threshold (t_0 =0.5)			NB with Optimal threshold (t_{opt})			
	pd	**pf**	**bal**	**t_{opt}**	**pd**	**pf**	**bal**
ar1	0.66	0.40	0.47	0.1	0.89	0.42	0.57
ar3	0.62	0.35	0.49	0.8	0.62	0.33	0.50
ar4	0.70	0.27	0.59	0.9	0.70	0.23	0.62
ar5	0.62	0.14	0.60	0.5	0.62	0.14	0.60
ar6	0.27	0.12	0.25	0.9	0.27	0.12	0.26
jm1	0.21	0.06	0.21	0.1	0.24	0.06	0.23
kc1	0.67	0.25	0.57	0.1	0.72	0.28	0.59
kc2	0.80	0.22	0.69	0.6	0.80	0.22	0.70
kc3	0.48	0.28	0.40	0.8	0.48	0.26	0.41
mc2	0.65	0.30	0.48	0.6	0.65	0.28	0.50
pc1	0.60	0.23	0.52	0.1	0.65	0.24	0.54
pc3	0.70	0.31	0.56	0.8	0.70	0.29	0.57
pc4	0.73	0.34	0.56	0.2	0.78	0.35	0.58
avg	**0.58**	**0.25**	**0.49**	**-**	**0.62**	**0.25**	**0.51**

Figure 5: Results for Replicated Study

Replicated Results

	NB with default threshold (t_0 =0.5)			NB with Optimal threshold (t_{opt})			
	pd	pf	bal	t_{opt}	pd	pf	bal
ar1	0.66	0.40	0.47	0.1	0.89	0.42	0.57
ar3	0.62	0.35	0.49	0.8	0.62	0.33	0.50
ar4	0.70	0.27	0.59	0.9	0.70	0.23	0.62
ar5	0.62	0.14	0.60	0.5	0.62	0.14	0.60
ar6	0.27	0.12	0.25	0.9	0.27	0.12	0.26
jm1	0.21	0.06	0.21	0.1	0.24	0.06	0.23
kc1	0.67	0.25	0.57	0.1	0.72	0.28	0.59
kc2	0.80	0.22	0.69	0.6	0.80	0.22	0.70
kc3	0.48	0.28	0.40	0.8	0.48	0.26	0.41
mc2	0.65	0.30	0.48	0.6	0.65	0.28	0.50
pc1	0.60	0.23	0.52	0.1	0.65	0.24	0.54
pc3	0.70	0.31	0.56	0.8	0.70	0.29	0.57
pc4	0.73	0.34	0.56	0.2	0.78	0.35	0.58
avg	**0.58**	**0.25**	**0.49**	-	**0.62**	**0.25**	**0.51**

Figure 6: Results for Replicated Study

	NB with default threshold (t_0=0.5)			NB with optimum threshold (t_{opt})			
	pd	pf	bal	t_{opt}	pd	pf	bal
ar1	0.70	0.33	0.56	0.5	0.70	0.33	0.56
ar3	1.00	0.35	0.65	0.8	0.80	0.13	0.76
ar4	0.80	0.39	0.56	0.6	0.70	0.31	0.57
ar5	0.90	0.23	0.75	0.6	0.80	0.10	0.78
ar6	0.65	0.42	0.45	0.6	0.55	0.31	0.45
jm1	0.50	0.30	0.42	0.5	0.50	0.30	0.42
kc1	0.82	0.37	0.59	0.7	0.65	0.20	0.60
kc2	0.85	0.26	0.70	0.6	0.84	0.21	0.74
kc3	0.90	0.33	0.66	0.6	0.83	0.25	0.70
mc2	0.60	0.40	0.43	0.6	0.50	0.18	0.47
pc1	0.73	0.32	0.58	0.6	0.70	0.22	0.63
pc3	0.85	0.37	0.60	0.6	0.76	0.27	0.64
pc4	0.91	0.30	0.69	0.7	0.82	0.17	0.75
avg	**0.79**	**0.34**	**0.59**	-	**0.70**	**0.23**	**0.62**

Figure 7: Results for Original Study

Conclusion Across Studies

- Changes in prediction performance by optimizing the decision threshold for Naive Bayes classifier
  - **Original:** *pf* changes from 34% to 23%
  - **Replicated:** No changes for *pf*
  - **Original:** *bal* changes from 59% to 62%
  - **Replicated:** *bal* changes from 49% to 51%
  - **Original:** *pd* decreases from 79% to 70%
  - **Replicated:** *pd* increases from 58% to 62%

Conclusion Across Studies

- **Original:** Decision threshold optimization helps to decrease false alarms, while keeping high pd rates
- **Replicated:** Decision threshold optimization helps to increase pd rate, while keeping the pf rates with no change
- **Original& Replicated:** a simple technique applied on a simple and robust classifier is easier to be understood and interpreted by the practitioners, instead of more complex models

Questions? Comments?