A novel approach to describing and detecting performance anti-patterns

Jinfang Sheng[1], Yihan Wang[1], Peipei Hu[1] and Bin Wang[1,*]

[1] School of Information Science and Engineering, Central South University, Changsha, 410083, China

* Corresponding author

Abstract. Anti-pattern, as an extension to pattern, describes a widely used poor solution which can bring negative influence to application systems. Aiming at the shortcomings of the existing anti-pattern descriptions, an anti-pattern description method based on first order predicate is proposed. This method synthesizes anti-pattern forms and symptoms, which makes the description more accurate and has good scalability and versatility as well. In order to improve the accuracy of anti-pattern detection, a Bayesian classification method is applied in validation for detection results, which can reduce false negatives and false positives of anti-pattern detection. Finally, the proposed approach in this paper is applied to a small e-commerce system, the feasibility and effectiveness of the approach is demonstrated further through experiments.

1. Introduction

As an extension of the concept of the pattern, the anti-pattern describes a commonly used bad solution, which will have a negative impact on the application system [1]. Anti-pattern detection as a bridge between design patterns and inexperienced personnel emphasizes the identification of harmful software structures that repeatedly appear in multiple projects. Anti-patterns are likely to be at all stages of the software life-cycle. As a software defect, anti-patterns can cause serious damages, especially in the software project management [2].

The presence of anti-patterns in the system can lead to many issues such as poor maintainability, scalability, reusability, and low performance. As a kind of anti-pattern, the existence of performance anti-pattern may have a serious impact on system performance. Therefore, it is very important to study the effective method of anti-pattern detection.

2. Related Work

Through the analysis of the current situation of anti-pattern detection, this paper holds that the current anti-pattern detection methods have the following problems.

1) The research on the detection of performance anti-pattern is relatively less.
2) The description capability of anti-pattern is insufficient.
3) No result process of performance anti-pattern detection.
4) No evaluation on the impact of the performance anti-pattern on the system.

In view of the shortcomings of these methods, this paper summarizes the 14 performance anti-patterns in literature [3-8], and selects the more common performance anti-patterns which have relatively high performance impact: Empty Semi Trucks [3] as an example for detailed analysis and discussion. This paper adopts first order predict method to describe anti-patterns based on the overall consideration of anti-patterns' manifestations and symptoms against deficiencies. In addition, this paper proposes a verification method of detection results based on Bayesian classifier, which can effectively reduce underreporting and misinformation. Finally, we apply this method to a small e-commerce system to further validate the feasibility and effectiveness of it.

3. Anti-Pattern Description Method Based on First-Order Predicate

Description1 Determine the measurement used to describe anti-patternsM_i, $M_i \in \sigma, \sigma$ is a set of all measurements, that is $\sigma = \{M_1, M_2, \cdots M_n\}$;

Description2 Measurement conditions:

$$P_i = f(M_i), R$$

Among them,P_i is a proposition which represents the comparison between $f(M_i)$ and R, $f(M_i)$ is a function to measure M, R represents the threshold of $f(M_i)$,a number of P_i constitute the conditions used to describe anti-patterns.

Description 3 AP is the name of anti-pattern, C_i represents the composition relationships between measurements, they are expressed as below:

$$AP = C_1 \vee C_2 \vee \cdots \vee C_n$$

$$C_{i=}P_1 \wedge P_2 \wedge \cdots \wedge P_n$$

When AP is true, it means the conditions of anti-pattern detection is met.

4. The Anti-Pattern Detection Method Based on Bayesian

4.1. The Probability Calculation of Candidate Anti-Patterns

4.1.1. Extremum Setting and Interval Division. Extremum is when measurement reaches it we can clearly determine if this pattern is anti-pattern. As shown in figure 1, the horizontal axis shows measurement and the vertical axis shows the probability of candidate anti-pattern whose range is from 0 to 1. L is the pre-set threshold, which is used to determine some rules shaped like $t_{value} \leq L$. The probability of L is set by specialists.

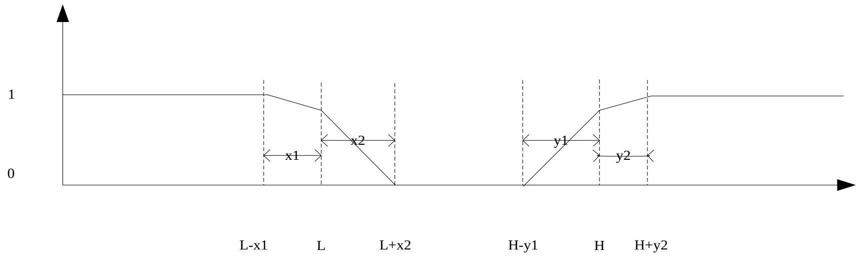

Figure 1. The probability setting interval of candidate anti-patterns

H is used to determine rules shaped like $t_{value} \geq H$. The probability of H is set by specialists.

4.1.2. The Probability Calculation of Intervals. The probability calculation of intervals can be divided into the following three types:

(1) The probability is calculated according to different intervals. As shown in equation 1.

$$p = \begin{cases} 1, \ t_{value} \leq L - x_1; \\ p_i + \frac{L - t_{value}}{x_1}(1 - p_i), \ L - x_1 < t_{value} \leq L; \\ p_i - \frac{t_{value} - L}{x_1}(1 - p_i), \ L < t_{value} \leq L + x_2; \\ 0, \ t_{value} > L + x_2; \end{cases} \tag{1}$$

(2) The probability is calculated according to different intervals. As shown in equation 2.

$$p = \begin{cases} 0,\ t_{value} < H - y_1; \\ p_{j-}\frac{H - t_{value}}{y_1}\left(1 - p_j\right),\ H - y_1 \le t_{value} < H; \\ p_j + \frac{t_{value} - H}{y_2}\left(1 - p_j\right),\ H \le t_{value} \le H + y_2; \\ 0,\ t_{value} \ge H + y_2; \end{cases} \tag{2}$$

(3) When we need to determine rules shaped like L<BP<H, we can represent it as BP>L and BP<H and calculate its probability by combining equation2 and equationr3.

4.1.3. The Probability Calculation of Synthesis Rules. (4)For those synthesized with some conditions, shaped like A^B, the probability can be expressed as the multiplication of all these conditions' probability. As shown in equation 3 in which p_i is the probability of a single condition.

$$p = \prod_{i=1}^{n} p_i \tag{3}$$

4.1.4. The Probability of Combined Rules. For those combined with some conditions, shaped like$A \vee B$, the probability can be calculated by equation4

$$f(x) = \begin{cases} 1,\ \forall p_i, \exists p_i = 1 \\ 1 - \sum_{i=1}^{n}(1 - p_i),\ \forall p_i x \neq 1 \end{cases} \tag{4}$$

p_i is the probability of the single rule which is anti-pattern. If $p_i = 1$, the probability of this condition being anti-pattern is 1. In other case, the probability can be calculted by equations.

4.2. The Verification of Candidate Anti-Patterns

Combining the Bayesian classification process, the process of anti-pattern detection based on Bayesian is defined as follows.

1) Suppose an anti-pattern can be represented by an n-dimension eigenvector $X = \{x_1, x_2, \cdots x_n\}$ and x_i describes n attributes of it.

2) The detection results of anti-pattern can be divided into two kinds:$c = \{c_1, c_2\}$. c_1 means the detection result is anti-pattern , on the contrary, c_2 means it is not.

3) The calculation of prior probability.

4) According to equation7:

$$p(C_1/x) = (p(x/C_1\ )p(C_1))/(p(x))$$
$$p(C_2/x) = (p(x/C_2\ )p(C_2))/(p(x)) \tag{5}$$

If considering system contexts (distributed applications? &expert knowledge) during anti-pattern detection, then $p(x/c_i)$ can be expressed as equation8

$$P\left(\frac{x}{c_i}\right) = \prod_{j=1}^{n} p\left(\frac{x_j}{c_i}\right) \tag{6}$$

Combining equation7 and8, we can know:

$$p(x) = \sum_{k=1}^{m} \prod_{j=1}^{n} p\left(\frac{x_j}{c_m}\right) p(c_m) \tag{7}$$

So according to equation7 and 9, we can know the probability of belonging to c_i is $p(c_i/x)$, which can be expressed as equation 10.

$$p\left(\frac{c_i}{x}\right) = \frac{\prod_{j=1}^{n} p\left(\frac{x_j}{c_i}\right)}{\sum_{k=1}^{m} \prod_{j=1}^{n} p\left(\frac{x_j}{c_m}\right) p(c_m)} p(c_i) \tag{8}$$

5) Respectively calculate $p(c_1/x)$ and$p(c_2/x)$. If $p(c_1/x) > p(c_2/x)$, this sample is anti-pattern, otherwise not.

5. Experiment

5.1. Sample Scenario Description

This paper applies the method mentioned earlier to the research object Java Pet Store1.3.2 [9] for anti-pattern detection. Java Pet Store1.3.2 is a simulated pet shop which simulates the whole process of buying pets by users. By analyzing the source code of Java Pet Store1.3.2, we find the anti-pattern Empty Semi Trucks appears during the period of viewing orders by users.

5.2. Results and Analysis

In order to complete the detection of Empty Semi Trucks, we firstly set the threshold of this anti-pattern' parameters based on experience. The sequence diagrams of system calls at runtime can be acquired according to the analysis of source code, so that the observed value corresponding to these parameters can be obtained. The anti-pattern detection starts after getting required information and completing threshold setting. Anti-pattern detection is divided into the following four steps:

1) Extremum setting and interval partition

Set the extremum and interval of each parameter.

2) The probability calculation of candidate anti-patterns at intervals

Combine equation 2 and 3 to calculate the probability of observed value being anti-pattern.

3) The probability calculation of synthetic rules and combined rules

Combine equation 4 and 5 to calculate the probability of candidate anti-patterns under various combined conditions.

4) The verification of detection results

In order to explain the application of Bayesian to anti-pattern detection, we simulate the following results aimed at Empty Semi Trucks. The experiment is divided into 3 groups, the data of group a meet the threshold conditions, that of group B is near the threshold and that of group C dissatisfy the threshold. Experimental data is illustrated in table 1.

Table 1. Simulated data

No.	A	B	C
1	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=18 $\mathrm{Th}_{\mathrm{maxRemInst}}$=6	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=12 $\mathrm{Th}_{\mathrm{maxRemInst}}$=5	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=11 $\mathrm{Th}_{\mathrm{maxRemInst}}$=4
2	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=18 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.28	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=13 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.28	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=10 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.28
3	$\mathrm{Th}_{\mathrm{maxGetMethNum}}$=6	$\mathrm{Th}_{\mathrm{maxGetMethNum}}$=4	$\mathrm{Th}_{\mathrm{maxGetMethNum}}$=2
4	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=15 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.28 $\mathrm{Th}_{\mathrm{maxRemInst}}$=6	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=13 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.31 $\mathrm{Th}_{\mathrm{maxRemInst}}$=4	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=10 $\mathrm{Th}_{\mathrm{minNetUtils}}$=0.35 $\mathrm{Th}_{\mathrm{maxRemInst}}$=3
5	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=15 $\mathrm{Th}_{\mathrm{maxRemInst}}$=6 $\mathrm{Th}_{\mathrm{maxGetMethNum}} = 4$	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=11 $\mathrm{Th}_{\mathrm{maxRemInst}}$=4 $\mathrm{Th}_{\mathrm{maxGetMethNum}} = 3$	$\mathrm{Th}_{\mathrm{maxRemMsgs}}$=10 $\mathrm{Th}_{\mathrm{maxRemInst}}$=4 $\mathrm{Th}_{\mathrm{maxGetMethNum}} = 2$

Conduct the anti-pattern detection by using the method mentioned in this paper and an ordinary method respectively.

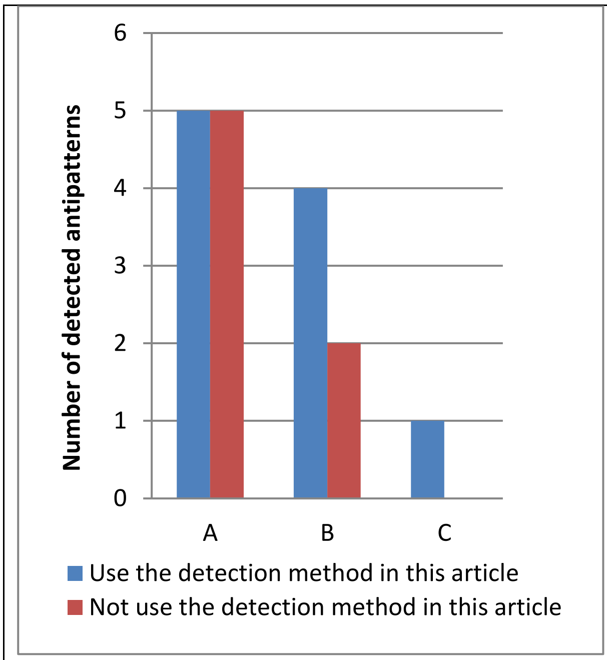

Figure 2. Results after verification based on Bayesian classifier

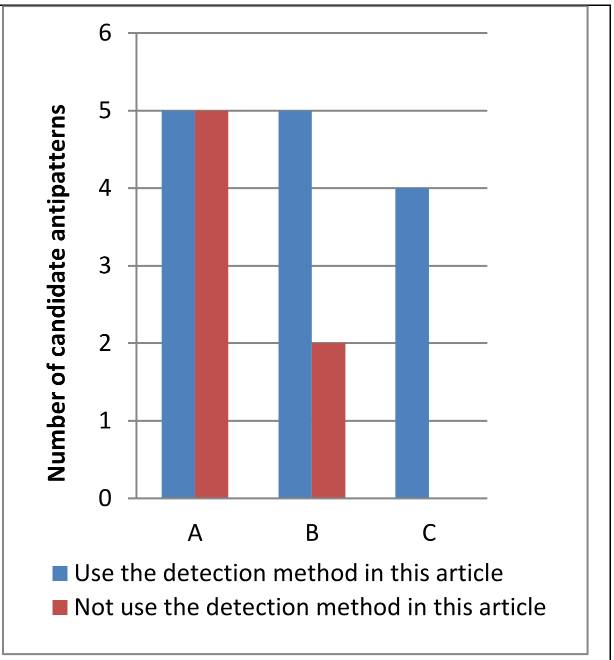

Figure3. Calculations of candidate anti-pattern

From the above-mentioned results, for conditions meeting the threshold, anti-patterns that are detected by an ordinary method can also be detected by the method in this paper; for conditions fluctuating across the threshold, the underreporting of anti-patterns can be effectively reduced; for those dissatisfying the threshold, the anti-patterns that are not detected by an ordinary method can be detected partly by the method in this paper, which can highly reduce the misinformation of anti-pattern detection.

6. Conclusion

The description method of anti-pattern based on first order predicate combines the manifestation with symptoms of anti-patterns to improve the reliability of description. This method is independent of specific rules templates and analytical models, so that it can modify and expand at any time. That is why the scalability of this method is superior to those with specific templates and models. Furthermore, this method is easy to implement without limitation to specific analytic language or inference engine. Meanwhile, adopting the Bayesian classifier with the existing testing information can effectively reduce the misinformation of detection results when introducing the concept of candidate anti-patterns. In addition, introducing the automated verification method based on Bayesian classifier before manual verification can also reduce the workload of artificial work so that improve the efficiency of manual verification. The results show that the description and detection method in this paper has obvious advantages.

7. References

[1] Keck P, Van Hoorn A, Okanović D, et al. Antipattern-Based Problem Injection for Assessing Performance and Reliability Evaluation Techniques[C]//Software Reliability Engineering Workshops (ISSREW), 2016 IEEE International Symposium on. IEEE, 2016: 64-70.

[2] Khomh F, Vaucher S, Sahraoui H. BDTEX: A GQM-based Bayesian approach for the detection of antipatterns [J]. Journal of Systems & Software, 2011, 84(4):559-572.

[3] Smith C U, Williams L G. More new software performance antipatterns: Even more ways to shoot yourself in the foot[C]//Computer Measurement Group Conference. 2003: 717-725.

[4] Crasso M, Zunino A, Moreno L, et al. JEETuningExpert: A software assistant for improving Java Enterprise Edition application performance [J]. Expert Systems with Applications, 2009, 36(9): 11718-11729.

[5] Smith C U, Williams L G. New software performance antipatterns: More ways to shoot yourself in the Foot[C]//Int. CMG Conference. 2002: 667-674.

[6] Williams L G, Smith C U. PASA (SM): An Architectural Approach to Fixing Software Performance Problems[C]//Int. CMG Conference. 2002: 307-320.

[7] Dugan Jr R F, Glinert E P, Shokoufandeh A. The Sisyphus database retrieval software performance antipattern[C]//Proceedings of the 3rd international workshop on Software and performance. ACM, 2002: 10-16.

[8] Marin M, Van Deursen A, Moonen L. Identifying aspects using fan-in analysis[C]//Reverse Engineering, 2004. Proceedings. 11th Working Conference on. IEEE, 2004: 132-141.