

Predicting Defect-Prone Software Module Using GA-SVM

Young-Ok Kim[†] · Ki-Tae Kwon^{**}

ABSTRACT

For predicting defect-prone module in software, SVM classifier showed good performance in a previous research. But there are disadvantages that SVM parameter should be chosen differently for every kernel, and algorithm should be performed iteratively for predict results of changed parameter. Therefore, we find these parameters using Genetic Algorithm and compare with result of classification by Backpropagation Algorithm. As a result, the performance of GA-SVM model is better.

Keywords : Defect-Prone Module, SVM, GA, Classification, Prediction Model, Reliability

GA-SVM을 이용한 결함 경향이 있는 소프트웨어 모듈 예측

김 영 옥[†] · 권 기 태^{**}

요 약

소프트웨어의 결함 경향 모듈 예측을 위해 SVM 분류기가 우수한 성능을 보인다는 연구들이 많지만, SVM에서 필요한 파라미터 선정 시 매 커널마다 다르게 선정해야 하고, 파라미터의 변경에 따른 결과예측을 위해 알고리즘을 반복적으로 수행해야 하는 불편함이 있다. 따라서 본 논문에서는 SVM의 파라미터 선정 시 유전알고리즘을 이용하여 스스로 찾게 하는 GA-SVM 모델을 구현하였다. 그리고 분류 성능 비교를 위해 신경망의 역전파알고리즘을 이용하여 분류했던 기존 논문과 비교 분석한 결과, GA-SVM 모델의 성능이 더 우수함을 확인하였다.

키워드 : 결함경향모듈, 서포트벡터머신, 유전알고리즘, 분류, 예측모델, 신뢰도

1. 서 론

현대 사회는 소프트웨어가 강조된 제품과 시스템들, 그리고 다양한 운영 환경에 의존하고 있다. 또 시스템들의 기능이 매우 많고 복잡하며 높은 신뢰성을 요구하지만 시스템의 시험 기간이 매우 길고 시험에 대한 충분한 정보도 부족하다. 과거에 개발한 시스템의 정보를 이용하여 현재 시스템의 결함 경향을 분석하면 현재 시스템의 시험을 수행하는데 많은 중요한 정보를 줄 수 있다. 즉, 어떤 모듈에서 결함이 발생할 경향이 높은 것으로 예측되면 이 모듈을 재설계하여 재개발하거나 시스템의 시험을 주의 깊게 할 수 있는 정량적 기준을 제시할 수 있다[1][2].

이러한 결함 발생 경향이 높은 모듈예측을 위해 회귀모델(regression model)과 분류모델(classification model)을 사용한다. 결함경향성 여부를 판단하는 많은 연구들에서 제안한 모델들은 대부분 메트릭 벡터들로 설계 개체들을 정량화한

후 이들을 위험 그룹과 비 위험 그룹으로 분류하는 분류모델들이었다[3]. 의사결정트리 분류기, 규칙기반 분류기, 신경망, 지도도백터기계, 나이브베이저안 분류기 등의 분류모델들이 결함유발 소프트웨어를 증명하는데 사용되고 있고 혼란 알고리즘으로는 복잡한 통계기법들이나 인공지능기법들이 사용되고 있다[1].

본 논문에서는 분류문제에 있어 일반화 성능이 높아 많은 분야에서 응용되고 있는 SVM을 이용하여 결함 경향 모듈을 예측하는 실험을 하였다. 그러나 SVM은 데이터 경향에 맞는 커널 함수와 파라미터들을 매번 조절하면서 더 좋은 결과를 찾아내야 하고, 파라미터의 변경에 따른 결과예측을 위해 알고리즘을 반복적으로 수행해야 하는 불편함이 있다.

따라서 본 논문에서는 유전알고리즘을 이용하여 SVM에서 사용될 최적의 파라미터들을 스스로 찾게 하였다. 또, 인공신경망의 역전파알고리즘을 이용하여 모듈의 결함 경향을 예측하였던 SungBack Hong et al.[2]의 연구 결과와 분류 성능 면에서 비교 분석해 보았다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구를, 3장에서 GA-SVM 모델의 실험과정을, 4장에서 실험 결과와 분석을, 그리고 5장에서 결론을 맺는다.

[†] 준 회원: 강릉원주대학교 컴퓨터공학과 박사수로

^{**} 종신회원: 강릉원주대학교 컴퓨터공학과 교수

논문접수: 2012년 8월 10일

수정일: 1차 2012년 9월 13일

심사완료: 2012년 9월 21일

* Corresponding Author: Ki-Tae Kwon(ktkwon@gwnu.ac.kr)

2. 관련 연구

2.1 유전알고리즘(Genetic Algorithm)

유전알고리즘은 자연 선택의 원리와 자연계의 생물 유전학을 바탕으로 한 병렬적이고 전역적(Global)인 탐색 알고리즘으로, 모든 생물은 주어진 다양한 환경 속에 적응함으로써 살아남는다는 Darwin의 적자생존(Survival of the fittest)의 이론을 기본 개념으로 한다[4]. 1970년대 John Holland에 의해 개발되었고, 생물이 갖는 환경에 대한 적응능력을 공학적으로 이용하려는 것으로 탐색, 최적화 및 기계학습을 위한 도구로 널리 사용되고 있다[5].

유전알고리즘의 과정은 다음과 같다[6][7]. 적합도가 높은 개체의 집합이 선택(Selection)되어 다음 세대의 자손을 생성하는 부모가 되며, 돌연변이(Mutation) 또는 교배(Crossover)를 통해 자손을 생성한다. 이렇게 생성된 자손을 이용하여 다시 적합도 검사를 실시하고 적합도가 가장 좋은 개체가 다음 세대를 위한 부모가 된다. 연구에서 원하는 평가 기준에 도달할 때까지 위와 같은 진화를 계속 반복하게 된다.

본 논문에서는 이 유전알고리즘을 이용하여 SVM에서 사용할 파라미터의 최적값들을 찾게 하였다.

2.2 SVM(Support Vector Machine)

SVM은 1995년 러시아 통계학자인 Vladimir Vapnik에 의해 제안된 커널 기반의 지도 학습 알고리즘으로[8], 분류문제에 있어 일반화 성능이 높기 때문에 많은 분야에서 응용되고 있으며, 다른 학습 알고리즘에 비해 조정해야 할 파라미터의 수가 많지 않아 비교적 간단하게 학습에 영향을 미치는 요소들을 규명할 수 있다.

1) 분리 가능한 선형 SVM

선형 SVM은 두 집합 사이의 분리간격(Margin)을 최대로 하는 초평면(hyperplane)을 찾는 분류기로서 최대 마진 분류기(maximal margin classifier)라 불린다[9].

Fig. 1에서 모든 검은 원을 클래스 +1, 모든 흰 원을 클래스 -1에 속한다고 하면, 시험 사례 x_i 에 대한 클래스 y 는 다음과 같이 나타낼 수 있다.

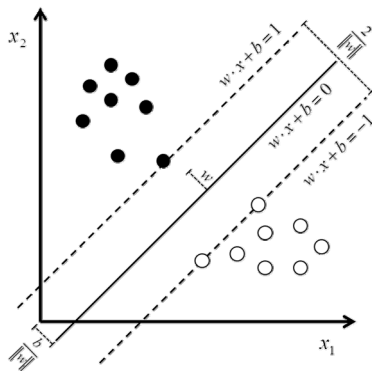


Fig. 1. Linear classifier

$$y = \begin{cases} 1, & w \cdot z + b > 0 \text{ 일 때} \\ -1, & w \cdot z + b < 0 \text{ 일 때} \end{cases} \quad (1)$$

두 경계 마진을 최대화하기 위한 목표 함수는 다음과 같은 최적화 문제로 나타낼 수 있다.

$$\text{Min} \left(\frac{\|w\|^2}{2} \right) \text{ Subject to } y_i (w \cdot x_i + b) \geq +1 \quad (2)$$

최적화 문제를 풀기 위해 라그랑주 승수(Lagrange multiplier) α_i 를 이용하여 라그랑지안(Lagrangian) 식을 유도하면 다음과 같다.

$$L_P = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i (y_i (w \cdot x_i + b) - 1) \quad (3)$$

$$\alpha_i \geq 0, \quad i = 1, \dots, N$$

식 (3)을 이용하여 매개변수 w , b 에 관한 최소값 L_P 를 찾아야 한다. 목표 함수 식 (2)에서 함수는 $w = 0$ 일 때 최소가 되므로, L_P 는 다음과 같은 최소화 조건식을 따르게 된다.

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^N \alpha_i y_i x_i = 0, \quad \frac{\partial L_P}{\partial b} = \sum_{i=1}^N \alpha_i y_i = 0 \quad (4)$$

식 (3)은 식 (4)의 조건을 따르는 쌍대문제(Dual formulation)로 변환하여 해결할 수 있다.

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (5)$$

식 (4), 식 (5)를 이용하면 분리간격(Margin)을 최대로 하는 초평면(hyperplane)을 결정할 수 있다.

2) 분리 가능하지 않은 선형 SVM

실제 입력 데이터를 적용할 경우 분리 불가능한 데이터가 존재하게 되는데, 이러한 오분류 데이터를 제거하기 위한 방법으로 슬랙 변수 ξ 와 패널티 값 C 를 사용한다[9]. 동일 입력데이터에 대해 패널티 상수 C 를 변화시킴에 따라 분리경계면 사이의 오분류 데이터와 분리간격의 변화를 아래 Fig. 2에서 확인할 수 있다.

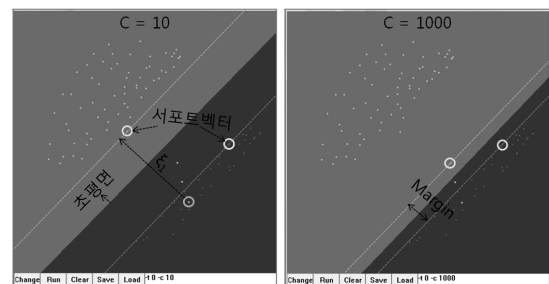


Fig. 2. Changes of separation boundary according to the C

C 값을 적용한 후 변경된 목표 함수는 다음 식과 같다.

$$Min(\frac{\|w\|^2}{2} + C \sum_{i=0}^n \xi_i) \quad (6)$$

쌍대문제로 변환하면 α_i 는 $0 \leq \alpha_i \leq C$ 의 범위를 갖는다.

3) 비선형 SVM

비선형 경계를 갖는 데이터를 SVM으로 분류하기 위해서는 아래 Fig. 3과 같이 본래의 좌표 공간에 있는 데이터 x 를 선형 분류를 가능하게 하는 새로운 차원의 좌표 공간 $\Phi(x)$ 로 맵핑하여 초평면을 구하게 된다[9].

$$f(X) = \sum_{i=1}^N \alpha_i y_i \Phi(x_i) \cdot \Phi(X) + b \quad (7)$$

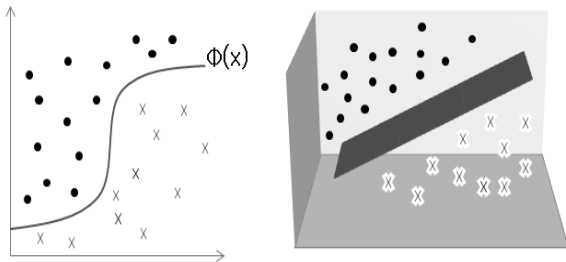


Fig. 3. Mapping for linear separation

$\Phi(x_i) \cdot \Phi(X)$ 에 의한 고차원 맵핑 문제를 해결하기 위해 커널 트릭 방법을 사용하는데, 가장 많이 사용되고 있는 커널 함수로는 Linear, Polynomial, RBF, Sigmoid가 있다.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (8)$$

커널 함수들은 고유한 비선형 그래프를 가지고 있으므로 실제 입력 데이터의 분포와 관련하여 가장 유사한 그래프를 선택해 맵핑 함수로 사용하게 된다.

3. GA-SVM을 이용한 실험

SVM은 분류 최적화에 대한 일반화 능력이 뛰어나지만, 예측 데이터의 경향에 따라 최적의 커널 함수와 파라미터들을 찾아야 하는 문제가 있어, GA-SVM 모델에서는 유전알고리즘을 이용하여 주요 파라미터들의 최적값을 예측하도록 하였다. SVM 관련 파라미터들은 아래와 같다.

- γ : sigmoid 커널함수에서 사용되는 파라미터 (sigmoid 커널함수 : $\tanh(\gamma * u' * v + coef0)$)
- C : SVM에서 사용하는 페널티 상수
- ϵ : SVM에서 측정값의 허용오차 설정 변수

GA-SVM 모델의 예측 과정은 Fig. 4와 같다. ① SVM에서 사용될 파라미터들을(γ, C, ϵ) 유전자로 하여 10개의 초기 개체군(initPopulation=10)을 생성하고 개체군의 유전자형을 결정한다 ② 12개 속성, 35개 인스턴스로 구성된 실제 데이터를 이용하여 ③ 2/3는 훈련데이터로, 나머지 1/3은 테스트 데이터로 하는 교차 검증(3-fold cross validation)을 한 후, 가장 좋은 교차검증정확도(cross validation accuracy) 값을 유전알고리즘에 전달하고 ④ 유전알고리즘에서는 이 정확도 값을 이용하여 적합도 평가를 하는데, 더 좋은 결과를 위해 유전자형을 복제하고, 돌연변이(mutants=0.1) 및 교배(incest=5)를 통해 변형한 후 새로운 개체군을 생성해 낸다. 이 유전자형을 이용하여 다시 ②, ③ 과정을 반복한다. 세대를 진화시켜 알고리즘을 반복 실행하고 적합도 평가를 한 결과, 더 이상 값이 향상되지 않으면 ⑤ 가장 마지막 사용된 유전자형을 SVM의 최종 파라미터로 선정하고, 32개의 테스트데이터를 분류한다.

교차검증정확도란 분류의 정확도인데, 이 값이 클수록 적합도 값은 작아지게 된다. Fig. 5는 총 3000 세대를 진화하

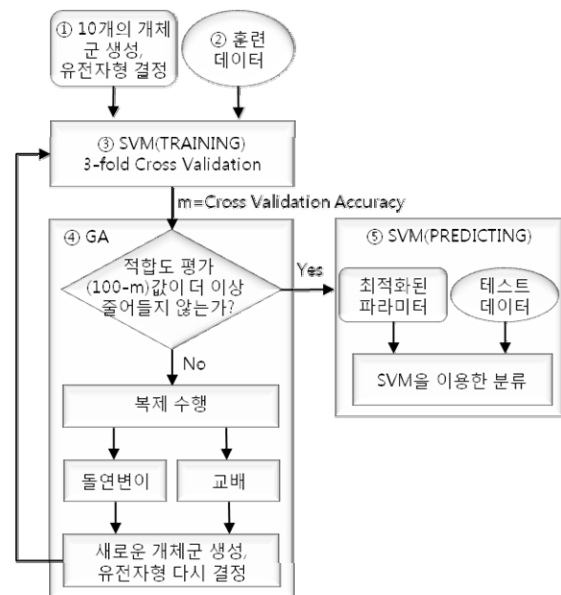


Fig. 4. GA-SVM model

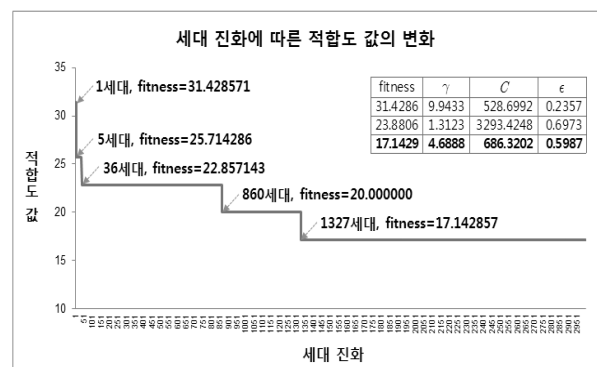


Fig. 5. Generations-evolution for improving of fitness value

는 동안 적합도 값이 향상되고 있음을 보여준다. 그러나 1327 세대부터 더 이상 적합도 값의 변화가 없는데, 이 구간의 유전자형(4.6888, 686.3202, 0.5987)을 SVM 모델의 최종 파라미터로 선정하여 테스트데이터를 분류한다.

GA-SVM 모델의 분류 성능을 평가하기 위해 사용된 데이터에 대한 설명은 다음과 같다.

SungBack Hong et al.[2]에서 사용한 M 프로젝트 데이터는 35개의 훈련데이터와 32개 테스트데이터가 있는데, 테스트데이터는 결함경향이 있는 모듈 6개와 결함경향이 없는 모듈 26개로 구분된다. 이 데이터의 각 메트릭에 대한 설명을 아래 Table 1에 정리하였다.

Table 1. Module complexity metrics

Complexity	Metrics	Means
Module Internal Complexity	m1	Halsteads Program Volume
	m2	Halsteads Program Length
	m3	Halsteads Program Vocabulary
	m4	Halsteads Program Difficulty
	m5	Halsteads Program Effort
	m6	McCabe Cyclomatic Complexity
	m7	Line of Code
	m8	Module Internal Task Complexity
Module External Complexity	m9	External Module Communication Complexity
	m10	External Library Interface Complexity
	m11	External Database Usage Complexity
CR		Change Request

데이터를 구성하고 있는 각 복잡도 메트릭은 모듈 내부 복잡도와 모듈 외부 복잡도를 나타낸다. 내부 복잡도는 모듈 내부의 수행 복잡도이기 때문에 모듈의 코드나 상세 설계를 필요로 한다. 제어 흐름 또는 구문 분석과 크기에 기초한 메트릭은 소프트웨어의 품질 측정을 위한 내부 복잡도 측정에 사용된다. 외부 복잡도 측정을 위한 메트릭은 정보 흐름 메트릭과 엔트로피를 이용한 메트릭이 있다. 이들은 모듈 간에 주고받는 정보의 양이 많을수록 모듈간의 결함도가 높아져 모듈의 복잡도가 커지게 되므로 역시 소프트웨어 품질에 영향을 미친다.

마지막 항목인 CR 값은 변경 요구의 개수를 나타낸다. 일반적으로 변경 요구가 많으면 많을수록 시스템의 결함이 있을 가능성이 높기 때문에 변경 요구서가 결함의 기준이 되고, 함수 블록에서 변경 요구서가 21개 이상인 것을 결함 경향이 있는 것으로 SungBack Hong et al.[2]에서 이미 정의하였는데, 같은 조건에서 성능 비교를 하기 위해 GA-SVM 모델에서도 그대로 적용하였다.

CR 값을 포함하여 12개의 속성과 35개의 인스턴스로 구성된 M 프로젝트 관련 데이터들은 각 메트릭 간의 값의 편차가 매우 크다. 이런 경우 알고리즘 수행 시간이 길어지므로 훈련데이터를 정규화 할 필요가 있다. 따라서 데이터의 최고값을 1로, 최하값을 -1로 바꾸고 나머지 값들은 그 사이에 뒀으로써 GA-SVM 알고리즘의 수행 시간을 단축시킬 수 있었다. 그리고 CR 값은 21 이상과 미만을 구분해 주면

되므로 21 이상 값들은 1(결함경향 있음)로, 미만 값들은 -1(결함경향 없음)로 정규화 하였다.

4. 실험결과

Table 2와 Table 3은 실제 결함 경향이 없는 그룹(26개 모듈)과 실제 결함 경향이 있는 그룹(6개 모듈)을 GA-SVM 모델의 테스트데이터로 이용하여 분류한 결과이다. 그리고 Table 4는 혼동 행렬이고 Table 5와 Table 6은 역전파알고리즘을 이용한 분류 결과와 GA-SVM 모델을 이용한 분류 결과를 혼동 행렬 형태로 정리한 것이다.

Table 2. Classification result of defect-prone modules

CR	Backpropagation Algorithm	GA-SVM Model
22	no defect-prone	no defect-prone
24	defect-prone	defect-prone
31	no defect-prone	defect-prone
24	defect-prone	defect-prone
28	defect-prone	defect-prone
32	defect-prone	defect-prone

Table 3. Classification result of no defect-prone modules

CR	Backpropagation Algorithm	GA-SVM Model
19	no defect-prone	no defect-prone
9	no defect-prone	no defect-prone
1	no defect-prone	no defect-prone
5	no defect-prone	no defect-prone
5	no defect-prone	no defect-prone
12	no defect-prone	no defect-prone
15	no defect-prone	no defect-prone
2	no defect-prone	no defect-prone
7	no defect-prone	no defect-prone
18	no defect-prone	no defect-prone
9	defect-prone	no defect-prone
9	no defect-prone	no defect-prone
5	no defect-prone	no defect-prone
4	no defect-prone	no defect-prone
8	no defect-prone	no defect-prone
3	no defect-prone	no defect-prone
19	no defect-prone	no defect-prone
20	no defect-prone	no defect-prone
16	no defect-prone	no defect-prone
10	no defect-prone	no defect-prone
13	defect-prone	no defect-prone
13	no defect-prone	no defect-prone
4	no defect-prone	no defect-prone
16	defect-prone	defect-prone
10	defect-prone	defect-prone
10	no defect-prone	defect-prone

Table 4. Confusion matrix

	Predicted	no defect-prone	defect-prone
Actual			
no defect-prone		TN = True Negative	FP = False Positive
defect-prone		FN = False Negative	TP = True Positive

- $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
- $Precision = \frac{TP}{TP + FP}$
- $Recall = \frac{TP}{TP + FN}$

Table 5. Prediction using the backpropagation algorithm

	Predicted	no defect-prone	defect-prone	total
Actual				
no defect-prone		22(84.62%)	4(15.38%)	26
defect-prone		2(33.33%)	4(66.67%)	6
total		24	8	32

Table 6. Prediction using the GA-SVM algorithm

	Predicted	no defect-prone	defect-prone	total
Actual				
no defect-prone		23(88.46%)	3(11.54%)	26
defect-prone		1(16.67%)	5(83.33%)	6
total		24	8	32

아래 Table 7은 역전파알고리즘과 GA-SVM 모델의 분류 성능을 비교한 것이다. 분류 성능 평가에서 일반적으로 [10][11] 사용되고 있는 Accuracy, Precision, Recall 값을 측정치로 하였다.

전체 테스트데이터 중 결함 경향 여부가 올바르게 예측된 모듈수의 비율을 보여주는 Accuracy 값에서 역전파알고리즘은 81.25%를, GA-SVM 모델에서는 87.5%의 분류결과를 보였다.

예측에서 결함 경향이 있다고 분류된 모듈 중 실제 결함 경향이 있는 모듈수의 비율을 보여주는 Precision 값에서는 역전파알고리즘에서 50%를, GA-SVM 모델에서 62.5%를 적중하였다.

실제 결함 경향이 있는 모듈 중에 예측에서도 결함 경향이 있으므로 분류된 모듈수의 비율을 보여주는 Recall 값에서는 역전파알고리즘에서 66.67%를, GA-SVM 모델에서 83.33%의 예측 분류율을 보여 전체적으로 보다 더 나은 분류를 하였음을 확인하였다.

다음은 분류 error에 대한 분석이다. Type I error(FP)는 실제 결함 경향이 아닌데 연구에서 제안한 모델이 결함 경향으로 분류한 것이고, Type II error(FN)는 실제로 결함 경향이 아닌데 결함 경향이 아닌 것으로 모델이 분류한 것이다. Type I error는 해당 모듈에 불필요한 많은 노력이 들어가고,

Table 7. Comparison of the classification performance

	Backpropagation Algorithm	GA-SVM Model
Accuracy	(26/32) 81.25%	(28/32) 87.5%
Precision	(4/8) 50%	(5/8) 62.5%
Recall	(4/6) 66.67%	(5/6) 83.33%

Type II error는 제품의 품질이나 사업 일정 등과 밀접한 관계가 있으므로 Type I error보다 더 중요한 의미를 갖는다.

아래 Table 8은 Type I error와 Type II error에 대한 비교이다. Type I error의 경우 역전파알고리즘에서는 15.38%, GA-SVM 모델에서는 11.54%의 error율을, Type II error의 경우 역전파알고리즘에서는 33.33%, GA-SVM 모델에서는 16.67%의 error율을 나타내어 Type I, II error율이 GA-SVM 모델에서 약간 더 낮음을 확인하였다.

Table 8. Comparison of CR values(Type I, Type II error)

Type I error		Type II error	
Backpropagation Algorithm	GA-SVM Model	Backpropagation Algorithm	GA-SVM Model
9, 13, 16, 10	16, 10, 10	22, 31	22
(4/26) 15.38%	(3/26) 11.54%	(2/6) 33.33%	(1/6) 16.67%

5. 결론

본 논문에서는 분류문제에 있어 일반화 성능이 뛰어나 많은 분야에서 응용되고 있는 SVM을 이용하여 결함 경향 여부를 예측하는 실험을 하였다. 그러나 SVM은 사용자가 임의로 조절해야 할 파라미터들이 존재하는데, 더 좋은 결과 예측을 위해서는 반복적인 실행과정을 통해 적절한 파라미터 값을 찾아야 하는 불편함이 있다.

본 논문에서 제안된 GA-SVM 모델에서는 유전알고리즘을 이용하여 SVM에서 사용될 최적의 파라미터들을 스스로 찾게 하였다. 그리고 그 분류 결과를 신경망의 역전파알고리즘을 이용한 실험과 비교 분석해 보았다.

그 결과 GA-SVM 모델에서 약간 더 좋은 분류를 하였음을 확인하였다. 특히 Type II error의 경우, 소프트웨어 신뢰성이나 제품의 품질, 사업 일정과 밀접하게 관련되어 있는데, 이러한 실제 결함 경향이 있는 모듈의 예측 분류에 있어 역전파알고리즘보다 GA-SVM 모델이 더 우수한 분류를 하였음을 확인하였다.

이러한 연구를 통해 소프트웨어를 시험하기 전, 구현이 끝난 소프트웨어 복잡도 메트릭들을 이용하여 어떤 기능 블록에 고장이 있을 것인지를 더 정확히 예측 분류하게 되면 신뢰도가 낮은 부분을 미리 인식할 수 있어 이 블록에 대한 재설계 여부에 정량적 판단 기준을 제시해 줄 수 있고, 기능 블록의 소프트웨어 시험 시 주의도 등의 평가에 중요한 정보를 줄 수 있다.

향후 연구는 다양한 클러스터링 기법을 이용하여 데이터를 클러스터링한 후, 각 속성이 해당 클러스터에 미치는 영향도 평가를 실험하고, 이 속성들이 소프트웨어 신뢰도 검사에 미치는 영향 등을 알아보고자 한다.

참 고 문 헌

[1] Khoshgoftaar, T.M., Allen, E.B., Kalaichelvan, K.S., Goel, N., "Early quality prediction: a case study in telecommunications", Software, IEEE, Vol.13, Issue 1, pp.65-71, 1996.

[2] SungBack Hong, KapSu Kim, YungGun Lee, Chisu Wu, "An Early Software Reliability Prediction Method using Backpropagation Algorithm", Journal of KIISE : Software and Applications, Vol.25, Issue 11, pp.1608-1617, 1998.

[3] Ebert, C., "Fuzzy classification for software criticality analysis", Expert Systems with Applications, Vol.11, Issue 3, pp.323-342, 1996.

[4] ByungRo Moon, "Easily Learning Genetic Algorithms", Hanbit Media, ISBN 978-89-7914-576-2, 2008.

[5] Toby Segaran, "Programming collective intelligence", O'relly, 2007.

[6] Heesung Lee, Euntai Kim, and Mignon Park, "A genetic feature weighting scheme for pattern recognition", Integrated Computer-Aided Engineering, Vol.14, Issue 2, pp.161-171, 2007.

[7] Yu, L., Wang S., Lai K. K., "Mining Stock Market Tendency Using GA-Based Support Vector Machines", Internet and Network Economics, Vol.3828, pp.336-345, 2005.

[8] Burges, C., "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol.2, Issue 2, pp.121-167, 1998.

[9] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to data mining", Addison Wesley, 2007.

[10] Witten, I., Frank, E., "Data Mining: Practical Machine Learning Tools and Techniques", second ed., Morgan Kaufmann, 2005.

[11] Karim O. Elish, Mahmoud O. Elish, "Predicting defect-prone software modules using support vector machines", The Journal of Systems and Software, Vol.81, Issue 5, pp.649-660, 2008.



김 영 옥

e-mail : kim052@gwnu.ac.kr

1997년 강릉원주대학교 컴퓨터공학과(학사)

2003년 강릉원주대학교 컴퓨터공학과

(교육학석사)

2012년 강릉원주대학교 컴퓨터공학과

(박사수료)

관심분야: 소프트웨어 신뢰도, 데이터 마이닝 등



권 기 태

e-mail : ktkwon@gwnu.ac.kr

1986년 서울대학교 계산통계학과(학사)

1988년 서울대학교 계산통계학과(이학석사)

1993년 서울대학교 계산통계학과

(이학박사)

1996년 Univ. of Southern California,

Post-Doc.

1990년~현 재 강릉원주대학교 컴퓨터공학과 교수

관심분야: 소프트웨어 비용산정, 소프트웨어 매트릭스, 소프트웨어 아키텍처, 소프트웨어 신뢰도, 데이터 마이닝 등