논문 2019-1-9 http://dx.doi.org/10.29056/jsav.2019.06.09

PCA를 적용한 결함 심각도 기반 차원 축소 모델

권기태*, 이나영

Defect Severity-based Dimension Reduction Model using PCA

Ki-Tae Kwon*, Na-Young Lee

요 약

데이터의 차원축소는 요소들의 공통성을 파악해 영향력 있는 중요한 특징 요소를 추출하여 간소화함으로써 복잡함을 줄이고 다중 공선성 문제를 해결한다. 그리고 중복 및 노이즈 검출을 함으로써 불필요함을 줄인다. 이에 본 논문에서는 PCA(Prinicipal Component Analysis)을 적용한 결함 심각도 기반 차원 축소 모델을 제안한다. 제안된 모델은 결함 심각도가 있는 NASA 데이터 세트인 PC4에 적용하여 결함 심각도에 영향을 주는 속성의 차원수를 검증한다. 그 다음 데이터의 차원을 축소한 후 비교 분석한다. 실험결과, PC4의 적합한 차원수는 2~3개였고 그룹화를 통해 차원 축소가 가능한 것을 보였다.

Abstract

Software dimension reduction identifies the commonality of elements and extracts important feature elements. So it reduces complexity by simplify and solves multi-collinearity problems. And it reduces redundancy by performing redundancy and noise detection. In this study, we proposed defect severity-based dimension reduction model. Proposed model is applied defect severity-based NASA dataset. And it is verified the number of dimensions in the column that affect the severity of the defect. Then it is compares and analyzes the dimensions of the data before and after reduction. In this study experiment result, the number of dimensions of PC4's dataset is 2 to 3. It was possible to reduce the dimension.

한글키워드 : 소프트웨어 품질, 결함 심각도, 주성분 분석, 연관성분석, 차원축소

keywords : software quality, defect severity, PCA, association analysis, dimension reduction

1. 서 론

2000년대 소프트웨어 결함 예측 연구는 NASA 데이터 세트가 공개되고 PROMISE 저장소가 운영되면서 크게 늘어났다[1][2]. 그리고 대부분의 소프트웨어 결함 예측 연구는 결함 경향이 있거나 없는 모델이나 클래스의 결함의 유무만으로 분류하는 지도 학습 모델이었다. 이러한 지도 학습 모델은 입력과 출력 데이터가 있을 경우에 훈련 데이터 세트를 생성하기 때문에 데이터의 수가 작은 경우에 적용하는 것이 어렵다.

* 강릉원주대학교 컴퓨터공학과
(email: ktkwon@gwnu.ac.kr)
접수일자: 2019.05.31. 심사완료: 2019.06.15.
게재확정: 2019.06.20.

그리고 결함 심각도에 기반한 소프트웨어 결함 예측 연구는 데이터의 수집이 쉽지 않고 정확하게 분류할 수 있는 전문가를 찾는 것이 어렵기 때문에 극소수에 불과하다[3].

이러한 이유로 결함 심각도에 기반한 연구에서는 지도학습이 아닌 비지도 학습을 적용한 방법론이 필요하고 차원 축소를 통해 요소들의 공통성을 파악하여 보다 간결하고 효과적인 소프트웨어 품질 관리가 요구된다. 그리고 결함에 영향을 주는 속성을 추출함으로써 상관 관계가 높은 속성 때문에 나타나는 다중 공선성 문제를 해결한다. 또한 데이터의 중복과 노이즈 검출을 통해 불필요한 데이터를 분류하고 제거하여 예측 정확도를 높힌다.

이에 본 논문에서는 소프트웨어 품질을 개선하고 제한점을 보완하기 위해 비지도 학습을 적용하여 소프트웨어 결함 심각도에 영향을 주는 속성의 관계 분석을 하는 모델을 제안한다. 제안 모델은 PCA를 적용한 결함 심각도 기반 차원 축소 모델로 결함 심각도가 있는 NASA 데이터 세트 중에 가장 독립적인 데이터로 구성되어 있는 PC4의 차원을 축소하고 차원수를 검증한다. 그리고 적합한 차원수를 찾아 그래프로 표현함으로써 차원 축소가 가능한 지를 실험하고 분석한다.

본 논문 제 2장에서는 소프트웨어 연관성 분석 및 PCA에 대해 살펴보고 제3장에서는 PCA를 적용한 결함 심각도 기반 차원 축소 모델을 제안한다. 제4장은 실험을 통해 결함 심각도에 영향을 주는 핵심 속성을 추출하여 모델을 평가하고 분석한 후 제 5장에서 결론에 대해 기술한다.

2. 관련 연구

2.1 근사 빈발 패턴 마이닝

체르노프 유계(chernoff bound) 기반인 근사 빈발 패턴 마이닝은 성공과 실패의 두 가지 결과 중 하나를 얻는 베르누이 시행(bernoulli trial)에 의해 이루어지는 확률에 기초한 방법이다. 동전 던지기로 비유되는 베르누이 시행은 어떤 경우 A에 대해 $x_1, x_2, x_3, \cdots, x_n, x_{n+1}, \cdots$로 표현될 때, 임의의 i번째 x_i가 A인 경우를 $\Pr[x_i = 1] = p$, A가 아닌 경우를 $\Pr[x_i = 0] = 1 - p$이라고 한다. 그리고 n번 시도에서 A가 나타난 경우의 수는 r, n번의 시도에서 실제로 A가 나타날 확률인 $\bar{r}$는 r/n, 최소 지지도(minimum support)는 s로 나타내면 이것을 기초로 체르노프 유계를 식으로 정의할 수 있다. 이 식은 식 1과 같다[4].

$$\begin{aligned} &\Pr\{|r - np| \geq np\gamma\} \leq 2e^{\frac{-np\gamma^2}{2}} \ (\gamma > 0) \\ &\Pr\{|\bar{r} - s| \geq s\gamma\} \leq 2e^{\frac{-ns\gamma^2}{2}} \ (\gamma > 0) \\ &\Pr\{|\bar{r} - s| \geq \epsilon\} \leq 2e^{\frac{-n\epsilon^2}{2s}} \ (\epsilon = s\gamma) \end{aligned} \quad (1)$$

식 1에서 ϵ은 A가 발생할 실제 값과 확률값의 차이, $\delta = 2e^{\frac{-n\epsilon^2}{2s}}$ 은 신뢰도를 나타내는데 ϵ은 식 1에 의해 δ보다 작거나 같다. 그리고 식 1을 변형하여 식 2와 같이 ϵ은 나타낼 수 있다[4].

$$\epsilon = \sqrt{\frac{2s \ln\left(\frac{2}{\delta}\right)}{n}} \quad (2)$$

이처럼 근사 빈발 패턴 마이닝은 ϵ의 값을 기준으로 하여 자주 발생하는 패턴을 찾아내는 방법이다.

2.2 최소 지지도 기반 빈발 패턴 마이닝

최소 지지도(minimum support) 기반 패턴 마

이닝은 $I=\{i_1,i_2,\cdots,i_m\}$인 $Items$ 집합, $D=\{T_1,T_2,\cdots,T_n\}$인 n개의 트랜잭션 집합이 있을 때 각 트랜잭션 T는 $T\subseteq I$ 이고 연관된 트랜잭션은 유일한 식별자인 TID로 구분된다[5].

표 1은 이러한 트랜잭션 데이터베이스의 예시이다.

표 1. 트랜잭션 데이터베이스
Tabel 1. Transaction Database

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$Items$ 집합인 $X=\{i_1,i_2,\cdots,i_k\}$ $(1\le k\le m)$는 해당 $Items$에 따라 빈발하게 발생하는 수를 의미한다. 예를 들면 표 1에서 {1}이 발생하는 경우는 2이고 {1, 2}가 발생하는 패턴의 수는 1, {1, 3}이 발생하는 패턴의 수는 2이다. 이 때 최소 지지도에 따라 단일 최소 지지도와 다중 최소 지지도로 구분되는데 단일 최소 지지도 기반 빈발 패턴 마이닝은 하나의 최소 지지도를 기준으로 해당되는 패턴을 찾아내는 방법이고 다중 최소 지지도 기반 빈발 패턴 마이닝은 하나 이상의 각각 $Items$에 대해서 최소 지지도를 평균·중간값 등을 활용하여 값이나 수식을 통해 만들어 결정한다.

이처럼 최소 지지도 기반 패턴 마이닝은 지지도의 값을 기준으로 하여 자주 발생하는 패턴을 찾아내는 방법이다.

2.3 PCA(Principal Component Analysis)

PCA는 선형 상관 관계와 차원 감소에 있어 가장 좋은 도구이고 기계 학습, 신경망, 특징 추출 등에 널리 사용하고 있다[6][7].

다양한 논문에서 PCA의 효율성을 증명하고 있다. [8]에서는 PCA가 데이터와 1차원 사이에 무작위 출력되는 상호 정보를 극대화시키기 위해 가우시안 잡음에 의해 손상된 N차원 가우시안 데이터에 대한 것을 입증했다. [9]에서는 PCA가 차원감소 때문에 잃어버린 정보의 손실을 최소화하는 선형 변환이라고 증명했다.

PCA는 상관 관계가 있는 변수를 결합해 분산이 극대화된 상관 관계가 없는 변수(주성분)으로 축약해서 정보 손실을 최소화하는 방법이다. 이는 3단계의 과정으로 진행되는데 다음과 같이 정리할 수 있다.

제1단계 초기값의 표준화 단계는 각 속성을 동등한 상태로 만들어 주는 것으로 값이 편향되지 않도록 한다. 표준화는 평균을 각 속성의 값에 빼고 분산으로 나누어 계산되는데 그 식은 식 3과 같다.

$$z=\frac{a-mean}{Stv} \quad (Stv=\text{분산}, a=\text{값}, mean=\text{평균}) \tag{3}$$

제2단계 공분산 행렬의 계산은 공분산과 상관계수를 계산해서 상관 관계를 분석한다. 이때, 공분산은 특정 속성 간의 상관 관계를 설명하는 것으로 상관 관계가 높고 낮음을 나타내고 상관 계수를 통해 상관 관계를 설명한다. 공분산과 상관계수의 관계는 식 4로 나타낸다.

모집단
$$\rho=\frac{Cov(X,Y)}{\sqrt{V(X)\,V(Y)}}$$
표본상관계수 (4)
$$r=\frac{\sum(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum(X_i-\overline{X})^2\sum(Y_i-\overline{Y})^2}}$$

이때, V(X)와 V(Y)는 각 X, Y의 분산이고 Cov(X, Y)는 X와 Y에 대한 공분산을 의미한다. 공분산은 변수 X와 Y에 대한 편차의 곱으로 계산한다.

제3단계 주성분 식별 단계는 고유벡터와 고유값으로 고유벡터의 직교화를 통해 선형으로 조합한다. 그리고 고유값, 기여도, 스크리 그래프를 분석하여 최적의 차원수를 결정한다.

3. 제안된 모델

3.1 데이터 세트의 선정

NASA의 13개 데이터 세트 중에서 결함 심각도가 있는 데이터 세트는 아래 5개와 같다[10].

- JM1 : 실시간 C 프로젝트 데이터 세트, 315 KLOC, 2012 모듈
- KC1 : 우주 왕복선을 비행하기 위한 연소 실험 데이터 세트, 63 KLOC, 23526 모듈
- KC3 : 인공위성의 데이터 수집, 전처리, 전송 등에 대한 데이터 세트, 18 KLOC, 458 모듈
- KC4 : Perl 코드로 쓰인 지상 가입 서버 데이터 세트, 25 KLOC, 125 모듈
- PC4 : 지구 궤도 위성 비행 소프트웨어 데이터 세트, 36 KLOC, 1458 모듈

이 중에서 가장 독립적인 데이터로 구성되는 연관성 분석과 CFS방법으로 추출한 PC4의 핵심 7개의 속성에 제안한 모델을 적용하였다[11].

3.2 제안 모델 세부 내용

제안된 모델은 선정된 데이터 세트의 연관성 분석을 통해 데이터를 정제한다. 그리고 PCA를 적용해 결함 심각도에 영향을 주는 핵심 속성에 대한 차원수를 결정하고 차원수에 대한 검증을 한다. 검증된 차원수로 다시 속성을 차원 축소하고 그룹화하여 비교 분석한 후 그래프로 시각화한다. 이러한 과정은 3.2.1에서 3.2.3와 같이 설명된다. 그리고 제안 모델을 토대로 분석도구 R을 이용하여 실험한다.

3.2.1 공분산과 상관 계수 계산

2.3에서 설명한 식 4에 의해 공분산과 상관 계수를 계산한다. 이 때, 공분산에 따라 상관 관계를 알 수 있다. 즉, 공분산이 양수이면 양의 상관관계, 음수이면 음의 상관 관계가 성립한다.

3.2.2 적합한 차원수 선택

PCA의 적합한 차원수를 결정하기 위해서 고유값(Eigen Value), 기여율(Cumulative proportion), 스크리 그래프(Scree plot)를 이용한다. 그리고 결정된 차원수 가운데 카이제곱(χ^2)의 적합도 검증으로 최적의 차원수를 선택한다. 카이제곱은 이론적 확률값에 따른 표본의 동일발생 여부를 검증하는 방법으로 계산식은 식 5와 같다.

$$\chi^2 = \sum \frac{(O-E)^2}{E} \qquad (5)$$

이 때, O는 실험해서 나온 실제값이고, E는 이론적으로 나온 기준값을 의미한다.

3.2.3 데이터 시각화

3.2.2에서 검증된 차원수로 나온 결과를 데이터 세트의 속성을 그룹화한 후 2차원과 3차원 그래프로 표현한다.

4. 실험 결과

4.1 PCA 결과

3.2.1에 따라 공분산은 cov로, 상관 계수는 cor

로 계산하는데 그 결과를 정리하면 그림 1, 그림 2와 같다. 이 결과의 값을 살펴보면 BRANCH_COUNT와 CONDITION_COUNT의 값이 0.98로 상관 관계가 가장 높은 결과가 나왔다. 이는 두 개의 속성이 서로 관계가 있음을 보여준다. 그리고 계산된 공분산과 상관 계수를 토대로 고유 벡터를 직교화하는 방식으로 princomp를 사용하여 PCA를 적용하였다. 적용한 PCA의 결과는 그림 3과 같고 결함 심각도가 있는 데이터 세트인 PC4에 PCA를 적용하는 것이 가능함을 보여준다.

4.2 적합한 차원수의 선택 결과

PCA에서 축소할 차원수의 결정은 매우 중요

	LOC_BLANK	BRANCH_COUNT	CALL_PAIRS	LOC_CODE_AND_COMMENT	LOC_COMMENTS	CONDITION_COUNT	DESIGN_DENSITY
LOC_BLANK	78,19	30,11	15,06	39,28	53,09	56,57	−0,26
BRANCH_COUNT	30,11	52,93	8,53	37,46	30,23	96,54	−1,00
CALL_PAIRS	15,06	8,53	11,22	6,69	10,16	16,12	0,22
LOC_CODE_AND_COMMENT	39,28	37,46	6,69	117,96	33,97	69,33	−0,49
LOC_COMMENTS	53,09	30,23	10,16	33,97	52,38	58,20	−0,27
CONDITION_COUNT	56,57	96,54	16,12	69,33	58,20	181,79	−1,77
DESIGN_DENSITY	−0,26	−1,00	0,22	−0,49	−0,27	−1,77	0,09

그림 1. 공분산 행렬
Fig. 1. Covariance matrix

	LOC_BLANK	BRANCH_COUNT	CALL_PAIRS	LOC_CODE_AND_COMMENT	LOC_COMMENTS	CONDITION_COUNT	DESIGN_DENSITY
LOC_BLANK	1,00	0,47	0,51	0,41	0,83	0,47	−0,10
BRANCH_COUNT	0,47	1,00	0,35	0,47	0,57	0,98	−0,47
CALL_PAIRS	0,51	0,35	1,00	0,18	0,42	0,36	0,22
LOC_CODE_AND_COMMENT	0,41	0,47	0,18	1,00	0,43	0,47	−0,15
LOC_COMMENTS	0,83	0,57	0,42	0,43	1,00	0,60	−0,13
CONDITION_COUNT	0,47	0,98	0,36	0,47	0,60	1,00	−0,45
DESIGN_DENSITY	−0,10	−0,47	0,22	−0,15	−0,13	−0,45	1,00

그림 2. 상관 계수 행렬
Fig. 2. Correlation coefficient matrix

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
LOC_BLANK	-0.4066146	0.31690616	-0.1492366	0.5027710	0.1507682	0.65996916	0.0299070324
BRANCH_COUNT	-0.4617998	-0.25383097	0.2519251	-0.2468744	-0.2691835	0.17576561	-0.7032383713
CALL_PAIRS	-0.2741205	0.52439089	0.4641940	-0.3014790	0.5467925	-0.21100665	0.0008197714
LOC_CODE_AND_COMMENT	-0.3259281	-0.03676913	-0.7775110	-0.4712839	0.2499858	-0.05723919	0.0042195560
LOC_COMMENTS	-0.4333919	0.21965842	-0.1486147	0.4406200	-0.3162725	-0.66732653	-0.0483351348
CONDITION_COUNT	-0.4645515	-0.23674609	0.2481709	-0.2402665	-0.3094613	0.10521017	0.7085131664
DESIGN_DENSITY	0.1907818	0.67416913	-0.1028958	-0.3484282	-0.5895250	0.17132761	-0.0146865106

그림 3. PCA의 결과
Fig. 3. PCA's result

```
$values
[1] 3.71088391 1.49518153 0.88917470 0.69081443 0.61400936 0.31312379 0.24775169
```

그림 4. 고유값
Fig. 4. Eigen value

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     1.9174 1.1928 0.8521 0.80796 0.60824 0.37011 0.12228
Proportion of Variance 0.5252 0.2032 0.1037 0.09326 0.05285 0.01957 0.00214
Cumulative Proportion  0.5252 0.7285 0.8322 0.92544 0.97830 0.99786 1.00000
```

그림 5. 기여율
Fig. 5. Cumulative proportion

하다. 이는 차원 축소한 데이터 세트가 원본 데이터 세트를 얼마만큼 대신 할 수 있는지를 보여주기 때문이다. 이러한 차원수의 결정은 고유값(EigenValue)이 1보다 클 때, 기여율(Cumulative proportion)이 70~80%일 때, 스크리 그래프(Scree plot)가 선에 있을 때의 차원을 선택한다. 실험 결과 적합한 차원수는 그림 4의 고유값이 1보다 클때인 2개, 그림 5의 기여율이 70~80일 때인 2~3개, 그림 6의 스크리 그래프가 선에 있을 때인 2개로 선택되었다.

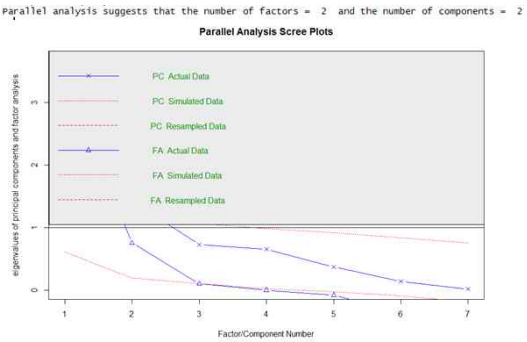
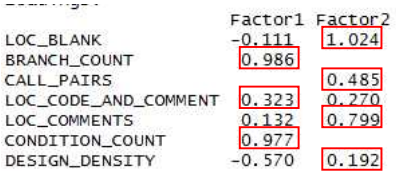


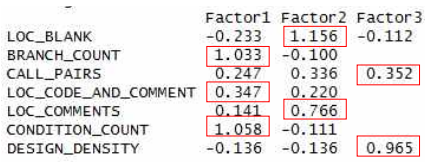
그림 6. 스크리 그래프
Fig. 6. Scree plot

이에 따라 최적의 차원수를 결정하기 위해 카이 제곱으로 적합도 검정한 결과 2개와 3개가 모두 유의한 결과를 보였다. 그리고 검정된 차원수를 기준으로 속성을 분류한 결과 그림 7, 그림 8과 같은 결과를 보였다. 그림 7과 그림 8에서의 속성별로 큰 값을 선택해 표시한 사각형 표현은 차원별 그룹화를 의미한다. 즉, 차원별 그룹화는 차원수가 2일 때 2개의 그룹을 의미하는데 첫 번째 그룹은 DESIGN_DENSITY, LOC_BLANK, CALL_PAIRS, LOC_COMMENT로, 두 번째 그룹은 BRANCH_COUNT, CONDITION_COUNT LOC_CODE_AND_COMMENT로 나눌 수 있다.



	Factor1	Factor2
LOC_BLANK	-0.111	1.024
BRANCH_COUNT	0.986	
CALL_PAIRS		0.485
LOC_CODE_AND_COMMENT	0.323	0.270
LOC_COMMENTS	0.132	0.799
CONDITION_COUNT	0.977	
DESIGN_DENSITY	-0.570	0.192

그림 7. 차원수 2의 결과
Fig. 7. Results of dimension number 2



	Factor1	Factor2	Factor3
LOC_BLANK	-0.233	1.156	-0.112
BRANCH_COUNT	1.033	-0.100	
CALL_PAIRS	0.247	0.336	0.352
LOC_CODE_AND_COMMENT	0.347	0.220	
LOC_COMMENTS	0.141	0.766	
CONDITION_COUNT	1.058	-0.111	
DESIGN_DENSITY	-0.136	-0.136	0.965

그림 8. 차원수 3의 결과
Fig. 8. Results of dimension number 3

4.3 데이터 시각화 결과 및 분석

데이터를 도식화하기 위해 RGL을 사용하여 차원수가 3일 때의 결과를 2차원과 3차원 그래프로 표현하면 그림 9, 그림 10과 같다. 그림 9에서 사각형 표현은 차원별 그룹화의 결과를 그래프로 나타낸 것이다.

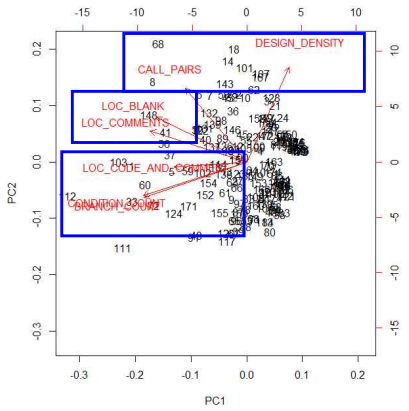


그림 9. 2차원 그래프(차원수=3)
Fig. 9. 2D plot

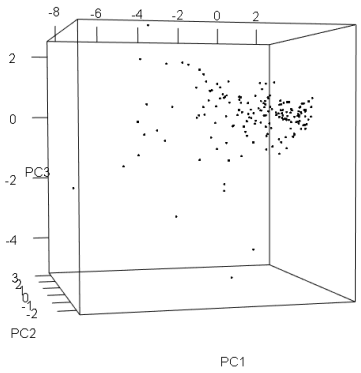


그림 10. 3차원 그래프
Fig. 10. 3D plot

5. 결 론

소프트웨어 결함 예측에 관한 연구는 소프트웨어 결함을 정확히 예측함으로써 소프트웨어 품질을 높이고 프로젝트의 성공에 기여한다. 그러나 결함 심각도에 기반한 소프트웨어 결함 예측에 관한 연구는 지도 학습을 통한 연구가 진행되고 있지만 전문가의 희소성과 데이터 수집의 어려움 때문에 비지도 학습을 통한 차원 축소 및 그룹화를 요구한다.

이에 본 논문에서는 비지도 학습 방법인 PCA를 적용한 결함 심각도 기반 차원 축소 모델을 제안하였다. 그리고 이 모델을 결함 심각도가 있는 NASA 데이터 세트인 PC4에 적용하여 그 결과를 비교 분석하였다. 그 결과 PC4의 차원 축소를 3가지 방법으로 2~3개의 차원을 선택하였고 적합한 차원수를 카이제곱으로 검정하여 차원별 그룹화를 도식화하였다. 즉, 실험 결과 PCA를 적용함으로써 결함 심각도 데이터의 차원 축소가 가능함을 보였다.

이에 향후에는 PCA를 토대로 다양한 핵심 속성을 추출하는 방법론을 제안하고 그에 따른 실험과 검증을 진행할 예정이다. 그리고 여러 방법론을 비교 분석하여 결함 심각도에 최적인 특징 추출 방법을 모색할 계획이다.

참 고 문 헌

[1] R. Malhotra, “A systematic review of machine learning techniques for software fault prediction”, Applied Soft Computing, Vol. 27, pp.504-518, Feb. 2015. https://www.sciencedirect.com/science/article/pii/S1568494614005857

[2] D. Radjenovic, M. Hericko, R. Torkar, and A. Zivkovic, “Software fault prediction metrics: A systematic literature review”, Information Soft. Technology, Vol. 55, pp. 1397-1418, 2013. https://www.sciencedirect.com/science/article/abs/pii/S0950584913000426

[3] Y. Zhou and H. Leung, “Empirical analysis of object-oriented design metrics for predicting high and low severity faults”, IEEE Trans. Software Eng., Vol. 32, No. 10, pp.771-789, 2006. https://ieeexplore.ieee.org/abstract/document/1717471/

[4] J.X. Yu, Z. Chong, H. Lu and A. Zhou, “False Positive or False Negative: Mining Frequent Itemsets from High Speed Transactional Data Streams”, International conference on Very Large Data Bases (VLDB), Vol. 30, pp.204-215, Aug. 2004. https://dl.acm.org/citation.cfm?id=1316709

[5] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, VLDB Conference, 1994.

[6] G. Deco and D. Obradovic, “An Information-Theoretic Approach to Neural Computing”, Springer, New York, 1996. https://www.springer.com/kr/book/9780387946665

[7] J. C. Principe, “Information Theoretic

Learning: Renyi's Entropy and Kernel Perspectives", Information Science and Statistics, Springer, New York, 2010. https://www.springer.com/kr/book/9781441915696

[8] R. Linsker, "Self-organization in a perceptual network", IEEE Computer, Vol. 21, No. 3, pp.105-117, Mar. 1988. https://ieeexplore.ieee.org/abstract/document/36

[9] M. D Plumbley, "On Information theory and unsupervised neural networks", Cambridge University Engineering Department, Tech. Rep. CUED/FINFENG/TR. 78, 1991.

[10] E. S. Hong, "Sortware Quality Prediction based on Defect Severity", Journal of the Korea Society of Computer and Information, Vol. 20, No. 5, pp.73-81, 2015. http://www.koreascience.kr/article/JAKO201517058945623.page

[11] Riju Kaushal, Sunil Khullar, "PSO based neural network approaches for prediction of level of severity of faults in nasa's public domain defect dataset", International Journal of Information Technology and Knowledge Management, July-December Vol. 5, No. 2, pp.453-457, 2012.

저 자 소 개

권기태(Ki-Tae Kwon)

1986년 서울대학교 계산통계학과(학사)
1988년 서울대학교 계산통계학과(석사)
1993년 서울대학교 계산통계학과(박사)
1996년 미국 Univ. of Southern California, 전산학과 Post-Doc.
1990년～현재 강릉원주대학교 컴퓨터공학과 교수.
<주관심분야> 소프트웨어공학, Statistical Learning 등

이나영(Na-Young Lee)

2003년 강릉원주대학교 컴퓨터공학(학사)
2005년 강릉원주대학교 컴퓨터교육(석사)
2015년～현재 강릉원주대학교 컴퓨터공학(박사수료).
<주관심분야> 소프트웨어 품질, 데이터마이닝