허용적 러프집합에 기반한 소프트웨어 분류기준

The Software Classification Criteria based on the Tolerant Rough Set

김상용*, 최완규*, 김영식**, 이성주*
SangYong-Kim*, WanKyoo-Choi*, YoungSik-Kim**, SungJoo-Lee*
*조선대학교 전자계산학과
**서강정보대학교 정보통신과
email: ykchoi@mina.chosun.ac.kr

요 약

소프트웨어의 측정값에 근거하여 소프트웨어 품질에 관한 의사결정을 할 때, 동치관계의 요구조건인 추이적(transitive) 특성이 항상 만족되는 것은 아니다. 순환수(cyclomatic number)가 거의 비슷한 프로그램에서, 하나는 "구조적인" 프로그램 범주에 속하고 또 다른 하나는 비구조적인 프로그램 범주에 속한다고 명확히 분류할 수 있는가하는 점이다.

따라서, 본 연구에서는 동치관계보다는 허용적 관계를 만족하는 허용적 러프집합에 근거한 소프트웨어 분류기준 제시하고자 한다. 분류기준을 생성하기 위한 실험 데이터 집합을 수집하고, 집합 내의 각 원소에 관한 허용적 클래스들을 생성한 후, 각 허용적 클래스들의 중심값을 클러스터링하여 분류기준을 생성한다. 생성된 분류기준을 또 다른 실험 집합에 적용하여 비교 분석하여 생성된 분류기준이 타당함을 보여준다.

Ⅰ. 서론

소프트웨어 측정값에 근거한 소프트웨어 품질에 관한 의사결정은 "품질이 좋다" 또는 "품질이 좋지 않다"라는 언어적 변수를 정의하고 이런 언어적 변수로 분류될 수 있는 측정값의 범위를 결정한 후 임의의 소프트웨어에 대해서 측정값에 근거하여 어떤 부류의 언어적 변수에 속하는 가를 결정하는 것이다.

여러 연구들[1, 3, 6, 7, 9, 12]이 실험을 통하여 소프트웨어 품질 평가를 위한 분류 기준들을 제시하였으나 이러한 분류 기준들은 분류에 관한 동치관계를 가정하고 있다. 그러나 소프트웨어 측정값에 근거하여 소프트웨어를 분류할 때 동치관계의 요구조건인 추이성(transitive)이 항상 만족되는 것은 아니다. 순환수(cyclomatic number)가 19와 20인 프로그램 A와 B에서, A는 "구조적인" 프로그램 범주에 속하고 B는 "구조가 필요 이상으로 복잡한" 프로그램 범주에 속한다고 명확히 분류할 수 있는가하는 문제이다.

따라서 본 연구에서는 소프트웨어 측정값에 근거하여 소프트웨어를 분류할 때, 동치관계보다는 추이적 특성이 없는 허용적(tolerance) 또는 유사(similarity) 관계를 만족하는 허용적 러프집합[10, 11]에 기반하며, 이를 데이터의 유사관계 표현에 적용하여 소프트웨어의 분류기준을 산출하고 제시하고자 한다.

Ⅱ. 소프트웨어 분류기준 생성

본 연구에서는 소프트웨어 메트릭에 근거하여 소프트웨어를 분류할 때 k개의 범주로 분류하기 위하

여 먼저 각 객체들에 대한 허용적 클래스(tolerance class)들을 정의하고, 각 허용적 클래스들을 클러스터링하여 k개의 그룹으로 분류하기 위한 분류기준을 정의한다.

2.1 소프트웨어 분류를 위한 허용적 글래스(Tolerance class)

데이터 분류에 관한 문제의 경우에서는 동치관계를 적용하여 데이터간의 유사관계를 나타내는 것은 불합리하므로[13], 반사성(reflexive)과 대칭성(symmetric)을 만족하는 허용적 관계에 의해 데이터들 간의 유사관계를 나타내는 것이 필요하다[2].

소프트웨어 메트릭에 근거한 소프트웨어의 품질에 관한 의사결정에서도 반드시 추이적 성질이 만족하는 것은 아니다. 일반적으로 "프로그램의 라인수가 30이하이면 적합하다"는 기준을 근거로 할 때, 29와 30라인으로 구성된 프로그램은 적합하고 31라인으로 이루어진 프로그램은 적합하지 않다라고 명확히 분류할 수 있는가라는 문제가 제기된다. 그러므로 소프트웨어의 분류의 경우에 허용적 관계에 의해서 소프트웨어들 간의 유사관계를 나타내는 것이 필요하다.

일반적으로 허용적 관계는 두 원소간의 유사성 정도를 나타내는 유사성 척도(similarity measure)에 의해 표현된다[5]. 데이터 분류와 같은 문제에서는 흔히 유사척도를 Hamming distance, Euclidian diatance 등과 같은 거리 함수를 사용하여 정의하지만, 메트릭 측정값에 근거하여 소프트웨어를 분류하기 위하여 그들을 적용하는 데에는 문제가 있다.

LOC값에 근거하여 소프트웨어를 분류할 때, 10라인과 20라인으로 구성된 소프트웨어 객체들간에는 분명한 차이를 보이지만 100라인과 110라인으로 구성된 프로그램 객체들간에는 분명한 차이를 보이지 않음을 알 수 있다.

따라서, 본 논문에서는 메트릭 측정값에 근거하여 프로그램 객체들을 분류할 때, 퍼지집합의 소속함수를 이용하여 유사성 척도를 다음과 같이 정의한다.

$$s(x_i, x_j) = \mu(x_i) = \frac{1}{1+(x_j-x_i)^2(x_{max}+1-x_i)/x_{max}} \quad (1)$$

$x_i \in U,\ x_j \in U,\ j \neq i,\ j = 1, 2, \cdots, n$

x_{max} : 미리 정의된 최대 값

식(1)의 유사성 척도에 근거하여 임의의 한 원소 $x_i \in U$의 허용적 클래스(tolerance class) $\tau(x_i)$를 식(2)와 같이 정의한다.

$$\tau(x_i) = \{ x_j \mid s(x_i, x_j) = \mu(x_i) > \alpha,\ x_i, x_j \in U,\ j \neq i,\ j = 1, 2, \cdots, n\} \cup \{x_i\} \quad (2)$$

2.2 Clustering

n개의 객체들에 대한 허용적 클래스들은 n개가 생성되는데, n개의 허용적 클래스들을 k개의 ($k \leq n$) 집합으로 묶어서 k개의 분류기준을 생성하기 위해서 n개의 허용적 클래스들에 클러스터링 알고리즘을 적용한다.

클러스터링은 일련의 유사성이 있는 대상 자료들을 하나의 그룹으로 구성될 수 있도록 다른 그룹들과 분리하는데[16], 이를 위하여 각 허용적 클래스들의 중심값을 구한 후, 각 중심값들을 k-means 알고리즘을 사용하여 클러스터링한다.

2.3 분류기준 생성 알고리즘

n개의 허용적 클래스들로부터 k개의 분류기준을 생성하는 알고리즘은 그림 1과 같다.

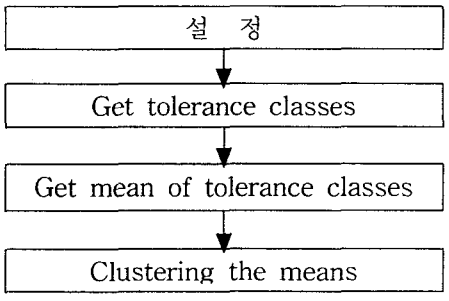

그림 1. 분류기준 생성 알고리즘

Ⅲ. 실험 및 결과

실험을 위해서 본 연구는 C언어로 작성된 18,404개의 모듈을 대상으로 하였다. 실험 대상의 모듈들의 전체 라인수는 533,165이였다. 적정 표본개수[8]인 8,213개를 SPSS를 이용하여 무작위로 선택하여 분류기준 생성을 위한 실험 데이터 집합 T_A를 구성하였고, 실험 결과를 나머지에 실험 데이터 집합 T_B에 적용하여 T_A에서의 결과와 T_B에서의 결과를 비교하였다.

3.1 정규모집단에 대한 검토

T_A에서의 LOC에 관한 누적 분포와 NTT에서의 4년에 걸쳐 실시된 Isoda의 실험[4]에서의 누적 분포가 유사한 형태를 보이고 있으며, SPSS를 이용하여 추출된 각 척도들의 왜도(skewness)와 첨도(kurtosis)가 0에 근사한 작은 값들을 가진다는 것과 중심점의 극한 정리[14, 15]에 의해 실험에 사용된 데이터가 근사적으로 정규 분포가 된다는 것을 알 수 있다.

3.2 분류기준 결정

식(1)과 식(2)을 이용하여 T_A의 각 원소들에 대한 허용적 클래스를 구하기 위해서 각 측정 메트릭들의 최대 값을 (LOC=100, CYC=50, VOL=10,000, DIF=100, EFF=300,000)으로 설정하였으며, 식(2)에서의 α 값을 0.1로 설정하였다.

실험 데이터 집합 T_A에 제안된 알고리즘을 적용하여 산출된 분류 기준은 표 1과 같다

표 1. 산출된 분류 기준

분류 / Metrics	Low	Medium	High
LOC	0<~≤29	24<~≤57	50<~≤100
CYC	0<~≤12	7<~≤19	14<~≤50
VOL	0<~≤568	564<~≤1405	1403<~≤10000
DIF	0<~≤22	15<~≤43	35<~≤100
EFF	0<~≤29734	29636<~≤91667	91456<~≤300000

3.3 분류기준의 적용

표 1의 분류기준을 T_A의 8,213개의 모듈과 T_B의 10,191개의 모듈에 적용하여 T_A와 T_B의 원소들을 분류할 때 두 집단의 특성을 비교한다.

등분산의 가정(H_0: $\sigma_A^2 = \sigma_B^2$)아래서 분류된 데이터들의 평균이 통계적으로 유의한 차이가 있는가(즉, H_0: $\mu_A = \mu_B$)를 검정하기 위한 유의수준 5%에서 T-검정 통계량은 표 2와 같다.

표 2. 각 분류집합에 대한 T-검정 통계량

Low	t 통계량	-0.54139
	P(T<=t) 양측 검정	0.58825
	t 기각치 양측 검정	1.960166
Medium	t 통계량	0.214245
	P(T<=t) 양측 검정	0.830363
	t 기각치 양측 검정	1.96032
High	t 통계량	-0.24485
	P(T<=t) 양측 검정	0.806589
	t 기각치 양측 검정	1.96078

표 2에서 "t 통계량 < t 기각치 양측 검정"을 만족하고, "P(T<=t) 양측 검정>0.05"이므로 유의하지 않으므로 $\mu_A = \mu_B$ 가정을 기각할 수 없으므로 표 1의 분류기준을 통해서 두 집단을 분류할 때, 두 집단간에 차이가 없다는 결론을 내릴 수 있다.

Ⅳ. 결 론

본 연구에서는 동치관계보다는 추이적 특성이 없는 허용적 관계를 만족하는 허용적 러프집합에 근거한 프로그램 분류기준을 제시하였다.

프로그램의 분류기준 제시를 위하여, 각 객체들에 관한 허용적 클래스들을 산출하고, 다음으로 허용적 클래스들의 중심 값을 클러스터링하여 프로그램 분류 기준을 설정하였다.

본 연구는 지금까지 단순한 실험을 통해서 제시되었던 소프트웨어 메트릭에 근거한 소프트웨어의 평가기준을 개선하여 러프집합 및 퍼지집합 이론을 이용하여 인간의 직관에 더욱 유사한 품질 평가기준을

산출할 수 있음을 보여주었다.

V. 참고문헌

[1] Caldiera, G. and V.R. Basili, "Identifying and Qualifying Reusable Software Components", *Computer*, pp.61-70, Feb. 1991,

[2] K.Funakoshi and T.B.Ho, "Information retrieval by rough tolerance relation", *The 4th international Workshop on rough sets, Fuzzy sets, and Machine Discovery*, Tokyo, Nov. 1996.

[3] Horst Zuse, *Software Complexity-Measures and Methods*, New York:Walter de Gruyter, pp.25-37, 1991.

[4] Sadahiro Isoda, "Experience report on software reuse project: its structure, activities, and statistical results", *Proceedings of the 14th international conference on Software engineering*, pp.320-326, 1992.

[5] M.Kretowski and J. Stepniuk, "Selection of objects and attributes a tolerance rough set approach", *ICS Research Reports*, 1994.

[6] Lewis John, Henry Salie, "A Methodology for Integrating Maintainability Using Software Metrics", *Proceedings:Conference on Software Maintenance*, Miami, Florida, IEEE, pp.32-39, Oct. 1989.

[7] Lowell J. Arthur, *Measuring Programmer Productivity and Software Quality*, New York:John Wiley & Sons, pp.138-142, 1985.

[8] D.J.Luck, H.G. Wales, D.H.Taylor, *Marketing Research*, N.J.:Prentice-Hall, pp.611-612, 1970.

[9] T.McCabe, "A Complexity Measure", *IEEE Trans.SE.*, SE-2, pp.308-320, 1976.

[10] Slowwinski R. and Vanderpooten D. "Similarity relations as a basic for rough approximations, *ICS Research Reports*, 1994.

[11] Slowwinski R. and Vanderpooten D. "A Generalized definition of rough approximations", *ICS Research Reports*, 1996.

[12] Szentes J., Gras j., "Some Practical Views of Software - Complexity metrics and a Universal Measurement Tool", *First Australian Software Engineering Conference*, Canberra, pp.14-16, May. 1986.

[13] 김대진, 김철현, "허용적 러프집합을 이용한 필기체 숫자 인식", *한국퍼지및지능시스템학회논문지*, Vol. 9, No. 1, pp.113-123, 1999.

[14] 김우철외, *현대통계학*, 영지문화사, 1989.

[15] 김은정, 박양규, *SPSS 통계분석8*, 21세기사, 2000.

[16]조형기, 민준형, 최종욱, "클러스터링을 이용한 차종인식 모형", *한국정보처리학회논문지*, Vol.3, No.2, pp.369-380, 1996.