

RBF망을 이용한 소프트웨어 유지보수 비용 추정

(Software Maintenance Cost Estimation using RBF Network)

박주석[†] 정기원^{**}

(Juseok Park) (Kiwon Chong)

요약 소프트웨어 회사들은 새로운 개발보다는 기존 프로젝트의 유지보수와 성능향상 프로젝트를 보다 많이 수행한다. 기존의 비용 추정 모델들은 유지보수 프로젝트들에 적용할 수 있지만, 유지보수 분야에 적용시키기 위해서는 변경이 필요하다. 본 논문은 개발 프로젝트와 유지보수 프로젝트의 기능점수 계산 방법을 분류하고 ISBSG의 벤치마킹 자료를 회귀분석한 결과를 토대로 유지보수 프로젝트의 비용을 측정할 수 있는 방법을 제안하였다. 먼저, ISBSG 자료를 소프트웨어 비용에 영향을 미치는 요소인 프로그램 추가, 변경과 삭제 3가지 요소의 8가지 중에서 실제 유지보수가 나타나는 4가지 그룹으로 분류하였다. 그리고, 그룹별로 통계적 모델과 RBF 망(Radial Basis Function Network)을 이용한 모델을 개발하여 각각의 성능을 분석 평가한 결과 RBF 망이 통계적 모델보다 좋은 성능을 보였다.

키워드 : 개발비용, 소프트웨어 규모, 개발 기능점수, 유지보수 기능점수, RBF 망

Abstract Software industry has put more emphasis on maintenance and enhancement work than on the new development. The existing effort estimation models can still be applied to maintenance projects, though modifications are needed. This paper suggests a way to estimate the size of a maintenance project from the regression analysis of ISBSG's benchmarking data. First of all, among the 3 elements(addition, modification and deletion of the program) which influences the software cost, we selected and classified 4 groups from a total of 8 which shows actual maintenance cost from ISBSG's data. Moreover, we developed statistical model and a model which uses RBF(Radial Basis Function) Network and after evaluating each functions we concluded that the RBF Network is superior to the statistical model.

Key words : Development Cost, Software Sizing, Development Function Point, Maintenance Function Point, Radial Basis Function Network

1. 서론

소프트웨어공학 분야에서 해결해야 할 많은 어려운 문제들 중 하나는 소프트웨어 시스템의 개발(Development) 또는 유지보수(Maintenance) 활동에 소요되는 비용(Effort 또는 Person-hour)를 계획 수립 단계에서 보다 정확히 추정하는 것이다[1]. 많은 회사들은 개발 또는 유지보수 비용에 가장 큰 영향을 미치는 요인들에 대한 연구를 통해 사업 계획의 정확성을 향상시키는 것을 바라고 있다. 소프트웨어 개발과 유지보수 비용에 영향을 미치는 요인들을 추정하는 것은 고비용이 소

되며, 모든 요인들을 계획단계에서 상세하게 측정하는 것은 적절한 방법이 될 수 없다[1]. 따라서 어떤 요인들을 측정하고, 어떤 요인들을 측정하지 않아도 되는지 문제가 되는 것이다.

실제로 소프트웨어 개발과 유지보수에 소요되는 비용을 추정하려면 소프트웨어의 규모(Size)를 측정해야 한다[2,3]. 소프트웨어의 규모를 측정하는 척도로 라인 수(Lines Of Code : LOC)와 기능점수(Function Point : FP)가 있다[2]. 기능점수에 기반하여 비용을 추정하는 모델들[2,4-7]은 대부분 개발에 소요되는 비용에만 관심을 가졌으며, 유지보수 비용 추정은 등한시하였다. 90년대 이후 유지보수를 수행하는 프로젝트 비중이 점차 커지고 있으나, 이 분야에 대한 비용을 추정할 수 있는 연구로는 IFPUG[8], Boehm[9,10], NESMA[11], Cote et al.[12]와 권기태 et al.[13]에 불과하다. 이로 인해, 유지보수 비용은 소프트웨어의 규모와 무관하게 개발비용의

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

† 학생회원 : 숭실대학교 컴퓨터학과

iage2k@dreamwiz.com

** 종신회원 : 숭실대학교 컴퓨터학부 교수

chong@comp.ssu.ac.kr

논문접수 : 2003년 11월 19일

심사완료 : 2004년 1월 27일

10 - 15% 범위내에서 설정되고 있는 실정이다. 소프트웨어의 규모를 고려하여 유지보수 비용을 추정하고자 할 때 기존 개발비용 추정 모델을 적용하면 개발과 유지보수의 차이로 인해 비용 추정 결과의 신뢰성이 저하될 수 있다. 실제로 동일한 규모의 프로젝트라도 개발과 유지보수에 투입되는 비용 사이에는 차이가 발생할 수 있으므로 유지보수 비용을 적절히 추정할 수 있는 모델에 대한 연구가 절실히 요구되고 있다. 본 논문은 소프트웨어 유지보수 비용에 영향을 미치는 요인들을 식별하고, 이들 요인들에 기반을 둔 유지보수 비용 추정 모델을 개발하고자 한다. 2장에서는 유지보수 활동과 비용 추정에 관한 관련연구 및 문제점을, 3장에서는 소프트웨어 유지보수 비용 추정을 위해 적용 가능한 모델을 제시한다. 4장에서는 제안된 모델의 성능을 평가해 본다.

2. 관련 연구 및 연구배경

2.1 유지보수 활동

시간과 비용의 제한으로 인해 일반적으로 소프트웨어에 남아 있는 모든 결함을 발견하고 제거할 때까지 시험을 수행하는 것은 현실적으로 거의 불가능하다. 따라서 일정한 시간이 경과하면 시험을 중단하고 결함이 잠재된 상태에서 소프트웨어가 고객에게 양도된다.

양도된 소프트웨어를 운영하면서 발생하는 유지보수 활동은 정정 유지보수(Corrective Maintenance), 완전 유지보수(Perfective Maintenance)와 적응 유지보수(Adaptive Maintenance)로 구분된다. 정정 유지보수는 운영 단계에서 발견된 결함을 제거하는 활동이며, 새로운 기능을 추가하여 성능을 향상시키는 경우를 완전 유지보수라 한다. 이에 비해, 데이터베이스 또는 운영체제의 업그레이드, 컴파일러 버전 변경 등과 같은 운영환경 변화로 요구되는 소프트웨어 수정을 적응 유지보수라 한다.

안정상태(Steady State)의 연간 유지보수 비용(Annual Maintenance Effort)은 개발비용에 비해한다[9,17]. Roetzheim[17]은 전형적인 시스템(Typical System)에 대해, 납품 시점부터 안정상태의 유지보수 비용이 투입되는 시점에 도달할 때까지 4년간의 유지보수 활동 관찰에 의하면 안정상태에서 소요되는 유지보수 비용의 2배가 납품시점에 투입되기 시작하여 안정상태 시점까지 지속적으로 감소한다는 것이다. 즉, 납품시점에서는 개발비용의 30%가 투입되어야 하고, 지속적으로 감소하다가 안정화된 시점부터는 매년 개발비용의 15%를 유지보수 비용으로 투입해야 함을 의미한다.

Jorgensen[18]은 노르웨이의 대형 회사들로부터 124개 유지보수 프로젝트 데이터들을 획득하여 분석한 결과 정정 유지보수 비용이 적응 또는 완전 유지보수 비

용보다 4-5배 적게 투입됨을 밝혔다. Abran와 Nguyenkim[19]도 캐나다의 대형 회계연구소의 사용자 지원 활동, 정정, 적응과 완전 유지보수 비용분포를 관찰한 결과 유사한 결론을 얻었다. 유지보수의 많은 부분을 차지하는 완전 유지보수는 대부분이 새로운 기능의 추가에 기인하므로 개발 프로젝트의 규모 프로필과 유사한 점이 많다고 할 수 있으나 기존에 존재하는 기능에 대한 변경과 삭제도 발생할 수 있다. 따라서 개발 프로젝트의 기능점수와는 다른 방법으로 측정된 소프트웨어 규모에 기반하여 유지보수 비용을 추정하는 모델이 필요하다.

2.2 기능점수 소프트웨어 규모

70년대 말 IBM 회사는 사용되는 프로그램 언어에 영향을 받지 않는 소프트웨어 규모 척도를 개발하고자 하였으며, 그 결과 Albrecht[5]가 기능점수 기법을 제안하였다. 기능점수는 5개의 소프트웨어 기능요소(Function Element)들인 입력(External Input : EI), 출력(External Output : EO), 조회(External Inquiry : EQ), 파일(Internal Logical File : ILF)과 인터페이스(External Interface File : EIF)를 식별한다. 시스템의 기능적 규모는 각 개별 기능요소들의 복잡도에 따른 가중치 할당으로 계산되며, 이를 조절이 안된 기능점수(Unadjusted FP : UFP)라 한다. 5개 기능요소 각각에 대한 UFP를 UFP_{EI} , UFP_{EO} , UFP_{EQ} , UFP_{ILF} , UFP_{EIF} 라 하자. 시스템 수준의 복잡도는 14개 항목에 대한 프로그램 특성들의 영향 정도를 평가하여 가치 조절 인자(Value Adjustment Factor : VAF)를 산출한다. UFP와 VAF를 곱하여 조절된 기능점수(Adjusted FP : AFP)를 얻는다[14]. 이 AFP를 일반적으로 기능점수라 부른다. AFP는 개발비용 추정과 예측을 위해 널리 알려진 방법으로 개발되는 소프트웨어의 규모(Development FP : AFP_D)는 식 (1)과 같이 표현된다.

$$AFP_D = (UFP_{EI} + UFP_{EO} + UFP_{EQ} + UFP_{ILF} + UFP_{EIF}) \times VAF \quad (1)$$

개발 프로젝트들은 기능이 새로 생성(추가)되며, 5개의 기능요소로부터 측정된 기능점수를 사용한다. 반면에 유지보수 프로젝트들은 기존 프로그램에서 기능들이 추가(Added), 변경(Changed) 또는 삭제(Deleted) 되는가에 관심을 가지며, 각각에 대한 기능점수만을 고려하여 규모를 측정한다[1,15,16]. 추가되는 UFP를 UFP_{ADD} , 변경되는 UFP를 UFP_{CHG} , 삭제되는 UFP를 UFP_{DEL} 라면, 유지보수가 수행되는 소프트웨어의 규모(Maintenance FP : AFP_M)는 식 (2)로 표현된다.

$$AFP_M = (UFP_{ADD} + UFP_{CHG} + UFP_{DEL}) \times VAF \quad (2)$$

$UFP_{ADD} \times VAF$ 를 AFP_{ADD} , $UFP_{CHG} \times VAF$ 를 AFP_{CHG} ,

$UFP_{DEL} \times VAF$ 를 AFP_{DEL} 이라 하면 AFP_M 은 식 (3)으로 표현된다.

$$AFP_M = AFP_{ADD} + AFP_{CHG} + AFP_{DEL} \quad (3)$$

2.3 유지보수 비용 추정

독립변수인 비용을 추정하는데 필요한 종속변수인 소프트웨어 규모를 결정하는 방법에 관한 연구에서 제시된 유지보수 방법별 가중치를 표 1에 제시하였다.

표 1 기존 연구에서 제시된 유지보수 방법별 가중치

모델	소프트웨어 규모		
	추가	수정	삭제
COCOMO[9]	1.0	1.0	-
IFPUG[10]	1.0	1.0	1.0
NESMA[11]	1.0	0.25~1.5	0.4
Cote et al.[12]	1.0	0.8	0.33
권기태 et al. [13]	1.0	0.8	0.33

기능점수와 유지보수 비용간의 관계를 나타내는 회귀 모델을 유도하기 위해서는 적절한 독립변수의 결정, 모델의 가능한 함수 형태 결정, 결정된 모델의 평가 과정을 거쳐야 한다. 이때 AFP와 UFP를 독립변수로 사용할 수 있다. 기능점수 방법을 적용하는 실무자들은 UFP가 AFP와 같이 정확성을 가지고 있기 때문에 VAF 값을 무시하고 있다[20]. 또한, Kitchenham과 KnsI [21]도 VAF 값이 비용 추정 능력을 향상시키지 못함을 보였다. 따라서 유지보수 비용을 추정하는데 필요한 독립변수로 UFP와 AFP를 모두 적용하는 회귀모델을 유도할 수 있다.

회귀분석 결과 제안된 모델이 좋은지 여부를 평가하는 기준으로 결정계수(Coefficient of determination, R^2)와 상대오차를 적용한다. 회귀모델의 경우, 종속변수의 총 변동(Total Sum of Squares : SST) 값은 독립변수에 의해 결정되는 부분(Regression Sum of Squares : SSR)과 미지의 오차의 합(Residual Sum of Squares : SSE)으로 나타난다. 총 변동 중에서 회귀직선으로 설명할 수 있는 비율을 결정계수($R^2 = \frac{SSR}{SST}$)라 한다. 따라서 $R^2(0 \leq R^2 \leq 1)$ 이 0에 가까우면 추정된 회귀직선은 쓸모가 없으며, 값이 클수록 쓸모 있는 회귀직선이 된다. 상대 오차(Relative Error : RE)는 $\frac{실측치 - 추정치}{실측치} \times 100(\%)$ 이다. MRE(Magnitude of the RE)는 $|RE|$ 이며, n 개의 데이터에 대한 MMRE(Mean MRE) = $\frac{1}{n} \sum_{i=1}^n MRE_i$ 로 계산된다. MMRE가 작은 값이면 모델은 평균적으로 좋은 모델이 된다.

Jorgensen[18]은 109개 프로젝트에 대한 유지보수 비

율을 추정하기 위해 평균 생산성으로 데이터들을 분류하였으며, 회귀 모델과 신경망을 적용하였다. 신경망으로는 역전파 알고리즘(Backpropagation Algorithm)의 전향망(FeedForward Network : FFN)을 사용하였다. 그 결과, 신경망이 100%의 평균 상대오차를 보여 최적의 회귀 모델보다 좋지 못한 결과를 얻었다. 권기태 et al.[13]의 경우 국내 대기업에서 수행된 31개 유지보수 프로젝트에 대해 회귀모델을 적용한 결과 결정계수는 0.84, MMRE는 23%의 상당히 좋은 모델을 얻었다.

90년대 개발된 내용량의 유지보수 프로젝트를 대상으로 기존에 제안된 모델들의 성능을 분석하여 보자.

표 1로부터 NESMA[11]의 경우 수정되는 기능점수의 가중치를 결정하기 어려움으로 이를 제외한 나머지 가중치를 적용한 3가지 AFP와 3가지 UFP를 독립변수로 하는 회귀모형을 표 2와 같이 생각할 수 있다. UFP의 경우 1의 기능점수 계산은 삭제에 대해 고려하지 않고 추가와 수정에 대한 가중치가 1이므로 $UFP_M = UFP_{ADD} + UFP_{CHG}$ 이 된다. 또한, 경우 3은 $UFP_M = UFP_{ADD} + 0.8 * UFP_{CHG} + 0.33 * UFP_{DEL}$ 이 된다. AFP를 독립변수로 하는 경우 4는 식 (3)에 의해 $AFP_M = AFP_{ADD} + AFP_{CHG}$ 가 된다. 나머지 경우에 대해서도 동일한 방법으로 기능점수가 계산된다.

표 2 유지보수 비용 추정을 위한 독립변수 선정 방법

가중치			독립변수	
추가	수정	삭제	UFP	AFP
1.0	1.0	-	경우 1	경우 4
1.0	1.0	1.0	경우 2	경우 5
1.0	0.8	0.33	경우 3	경우 6

ISBSG Benchmark Release 6[15]의 234개 유지보수 프로젝트를 대상으로 이들 6가지 경우에 대해 유지보수 비용을 추정한 결과를 표 3에 제시하였다.

이와 같이 통계적 모델로 추정한 노력은 독립변수를

표 3 유지보수 비용 추정 모델 성능

독립변수 경우	유지보수 비용 추정	모델 성능	
		결정계수	MMRE
1	$E = 15.1399 UFP_M^{0.8902}$	0.4180	99.98%
2	$E = 15.3158 UFP_M^{0.8929}$	0.4108	101.43%
3	$E = 14.5484 UFP_M^{0.9113}$	0.4318	99.09%
4	$E = 17.2058 AFP_M^{0.9678}$	0.4196	100.14%
5	$E = 17.5885 AFP_M^{0.8665}$	0.4117	101.68%
6	$E = 16.5419 AFP_M^{0.8886}$	0.4334	99.05%

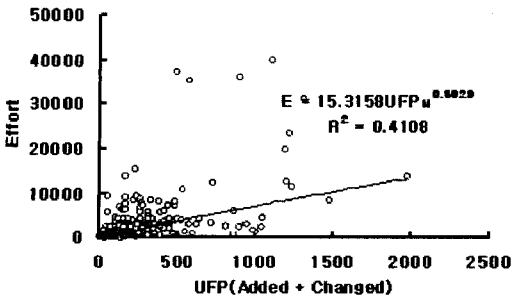


그림 1 경우 1을 독립변수로 유지보수 비용 추정 회귀 모델

UFP와 AFP로 고려하는 것에 상관없이 결정계수가 약 40%이다. 이는 실제 투입된 노력의 약 40%를 설명할 수 있다는 의미이며, MMRE가 약 100%로 실제 투입 노력에 대해 100% 오차를 나타냄을 의미한다. 예로 경우 1로 유지보수 비용을 추정하는 모델은 그림 1에 표현하였다.

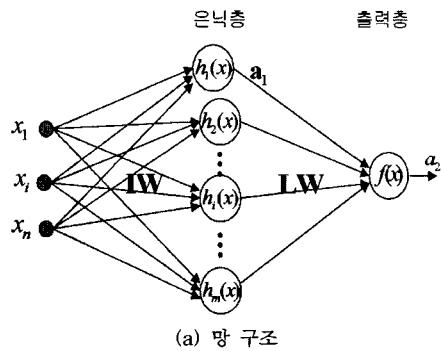
결론적으로 표 2와 같은 독립변수를 적용하여 유지보수 비용을 추정하는데 한계가 있음을 알 수 있다. 따라서 다른 방법을 이용하여 통계적 알고리즘 모델이나 비 알고리즘 모델을 제시하는 것이 보다 좋은 결과를 얻을 수 있을 것이다.

3. 유지보수 비용 추정 모델

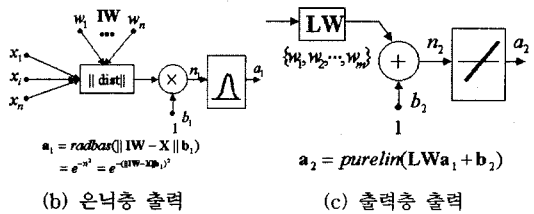
3.1 적용 모델

유지보수 비용을 추정하기 위한 모델로 통계적 회귀 모델과 비알고리즘 모델로 신경망은, 소프트웨어 규모는 AFP 대신 UFP만을 고려하기로 한다. 신경망 중에서 가장 일반적으로 많이 적용하고 있는 망이 전향 망이다. 그러나 역전파 알고리즘을 사용하는 전향망은 망이 가지고 있는 정보인 가중치의 정확한 값을 모르기 때문에 랜덤한 작은 값으로 초기화하고 훈련을 수행하여 적절한 가중치를 얻는다. 이와 같이 랜덤한 값을 초기 값으로 설정함으로써 인해 훈련을 수행할 때마다 동일한 결과를 얻지 못하는 단점이 있다. 이에 비해 RBF(Radial Basis Function)망은 LMS(Least Mean Squares) 알고리즘을 사용하며 훈련을 수행할 때마다 동일한 결과를 얻을 수 있어 본 논문에서는 RBF 망을 적용한다.

RBF 망은 주어진 입력들을 분류할 수 있는 군집(Clustering)을 결정하는 기저함수(Basis Function)를 찾는 것으로, 기저함수만 있으면 입력벡터 공간의 모든 점들을 기저함수의 선형결합으로 표현할 수 있다는 개념에서 유래되었다. 벡터공간 V 의 벡터집합을 S 라하고, S 에 있는 모든 벡터가 서로 1차 독립이며, V 의 모든 벡터는 S 에 있는 모든 벡터의 1차 결합(선형결합)으로



(a) 망 구조



(b) 은닉층 출력

(c) 출력층 출력

그림 2 RBF 망

표현될 수 있을 때 S 를 V 의 기저함수라 한다. RBF 망은 그림 2와 같이 2개 층(은닉층과 출력층)으로 구성되어 있다[22].

전형적인 다층 신경망의 은닉층 뉴런은 시그모이드 작동함수를 갖고 있고 입력의 가중합을 취한다. 이에 비해 RBF망은 입력의 가중합을 취하지 않으며, 은닉 뉴런의 작동함수는 가우시안(Gaussian) 함수(e^{-x^2})와 같은 방사 대칭(Radially Symmetric) 기저함수를 사용한다. 각 은닉층 뉴런의 출력(a_1)은 기저함수의 중심과 망의 입력간 거리에 의해 결정되는 기저함수로 표현되어 기저함수에 근접된 입력에만 응답하여 지역적으로 응답하는 영역을 가진다. 출력층은 선형 작동함수를 가지며 선형결합으로 은닉층 출력의 가중합을 생성한다. RBF 망은 FFN에 비해 최적의 은닉 뉴런 수(기저함수 개수)를 결정함에 있어 주어진 문제에 적합하도록 은닉 뉴런 수를 점진적으로 추가시키면서 주어진 문제에 적합한 뉴런 수를 찾는 방법(은닉 뉴런 수가 입력되는 표본의 수가 될 때까지) 또는 모델의 오차가 주어진 Goal인 SSE의 값 이하가 될 때까지 망을 훈련시킨다. 따라서 이 2가지 중에 어느 하나의 기준을 먼저 충족시키면 망의 훈련이 종료된다. 또한, RBF 망은 전형적인 다층 FFN 보다 빠른 수렴속도, 보다 적은 오차와 높은 신뢰성으로 인해 공학분야에서 점점 더 관심의 대상이 되고 있다.

3.2 유지보수 발생 경우 수 그룹화 및 모델 개발

유지보수는 추가, 변경과 삭제의 3가지가 조합되어 발

표 4 유지보수 활동별 발생유무에 따른 프로젝트 분류

경우 수	유지보수 활동별 발생 유무			그룹	프로젝트 수
	추가 (Added)	변경 (Changed)	삭제 (Deleted)		
0	0	0	0	-	-
1	0	0	1	D	0
2	0	1	0	C	10
3	0	1	1	CD	0
4	1	0	0	A	121
5	1	0	1	AD	2
6	1	1	0	AC	67
7	1	1	1	ACD	34

생할 수 있다. 따라서 표 4와 같이 8개의 경우 수가 발생할 수 있으며, 이들 경우가 실제로 발생되었는지를 살펴보자.

ISBSG Benchmark Release 6[15]의 유지보수 관련 234개 프로젝트를 검토한 결과 121개 프로젝트가 기능이 추가된 경우만 발생하였고, 67개 프로젝트는 새로운 기능 추가와 변경이 발생하였다. 34개 프로젝트는 기능 추가, 변경과 삭제가 모두 발생하였으며, 10개 프로젝트는 기능변경만 발생하였다. 또한 2개 프로젝트는 기능 추가와 삭제가 이루어 졌다. 2개의 데이터로는 회귀분석 모델 개발이 불가능하므로 기능 추가와 삭제가 발생한 프로젝트는 본 유지보수 비용 추정 모델 연구대상에서 제외시킨다. 새로운 기능이 추가 된 경우만 발생한 프로젝트들을 A 그룹, 새로운 기능 추가와 변경이 발생한 프로젝트들을 AC 그룹, 기능 추가, 변경과 삭제가 모두 발생한 프로젝트들을 ACD 그룹, 기능 변경만 발생한 프로젝트들을 C 그룹이라 하자. ACD 그룹의 추가, 변경과 삭제를 합한 기능점수를 $UFP_{ADD+CHG+DEL}$ 라 하고, AC 그룹의 추가와 변경을 합한 기능점수를 $UFP_{ADD+CHG}$ 라 하자. A, AC, ACD와 C 그룹에 대해 입력되는 변량이 1개인 경우와 2개 이상인 다변량인 경우를 고려할 수 있다. 각각의 경우에 대한 통계적 회귀모델과 RBF 망을 고찰한다. 소프트웨어의 규모가 커질수록 투입되는 노력은 선형적인 관계를 갖지 않는다. 따라서 표 5와 같이 통계적 모델인 경우, 변량이 1개인 경우는 비선형 회귀를, 변량이 2개 이상인 경우는 다중회귀분석을 수행한다. 비알고리즘 방법은 비선형관계를 표현하는 능력을 갖고 있으며, 변량의 수는 입력의 수로 치환이 된다. 비알고리즘 방법으로는 RBF 망을 이용하여 유지보수 비용을 추정하여 결과를 비교해 본다.

표에서 각 그룹에 대한 통계적 모델과 RBF망에 대한 모델명은 그룹명과 아래첨자를 이용하여 모델명을 분류하였다. 아래첨자의 첫째 자리는 모델 구분으로 통계적 모델은 S, RBF 망은 R로 명명하였다. 둘째 자리는 변

표 5 유지보수 그룹별 모델 개발

그룹	입력 수	입력	모델명	
			통계적 모델	RBF 망
A	1	UFP_{ADD}	A_{SS}	A_{RS}
AC	1	$UFP_{ADD+CHG}$	AC_{SS}	AC_{RS}
	2	UFP_{ADD}, UFP_{CHG}	AC_{SM}	AC_{RM}
ACD	1	$UFP_{ADD+CHG+DEL}$	ACD_{SS}	ACD_{RS}
	3	$UFP_{ADD}, UFP_{CHG}, UFP_{DEL}$	ACD_{SM}	ACD_{RM}
C	1	UFP_{CHG}	C_{SS}	C_{RS}

량 개수 구분으로 입력이 1개인 경우는 S, 입력이 2개 이상인 경우에는 다변량(Multivariate)으로 M을 적용하였다.

3.3 유지보수 비용 추정

먼저 통계적 회귀모델을 구하기 위해, A, AC, ACD와 C 그룹 각각을 대상으로 하였다. 이들 그룹에 대해 비선형 회귀(입력이 1개인 경우)와 중회귀(입력이 2개 이상인 경우) 분석을 통해 유지보수 비용 E_M 을 적절히 추정할 수 있는 모델을 찾고자 하였다. 이와 같은 방법으로 얻은 통계적 회귀모델들은 표 6에 제시되어 있다.

표 6 유지보수 그룹별 비용 추정 통계적 모델 성능

그룹	모델명	유지보수 비용 추정 모델	결정계수	MMRE
A	A_{SS}	$E_A = 0.1490 \cdot UFP_{ADD}^{0.8463}$	0.2993	112.11%
AC	AC_{SS}	$E_{AC} = 0.1336 \cdot UFP_{ADD+CHG}^{0.7815}$	0.1239	86.30%
	AC_{SM}	$E_{AC} = 0.0657 UFP_{ADD} + 0.0180 UFP_{CHG}$	0.0999	77.39%
ACD	ACD_{SS}	$E_{ACD} = 0.1539 \cdot UFP_{ADD+CHG+DEL}^{0.8071}$	0.5960	75.03%
	ACD_{SM}	$E_{ACD} = 0.1012 UFP_{ADD} + 0.0382 UFP_{CHG} + 0.0503 UFP_{DEL}$	0.6155	106.97%
C	C_{SS}	$E_C = 0.0357 \cdot UFP_{CHG}^{0.9683}$	0.6638	47.31%

A, AC, ACD와 C 그룹에 대한 RBF 망도 변량이 1개인 경우와 다변량인 경우에 대해 유지보수 비용 E_M 을 적절히 추정할 수 있는 모델을 찾고자 하였다. 예로, AC 그룹에 대해 RBF 망을 적용하여 유지보수 비용을 추정하는 모델은 그림 3에 제시되어 있으며, 각 그룹별 RBF 망 모델의 성능은 표 7과 같다.

RBF망을 적용시 기저함수의 개수는 데이터 개수보다 약간 적어 모수의 수가 많아 보인다. 이 경우만을 고려해 볼 때, 모수 2~3개를 이용하는 통계적 모델이 보다 간단하여 좋은 모델로 선정될 수 있다. 그러나 통계적 모델은 비선형 회귀분석으로부터 모수의 값을 결정하는 어려움이 있는 반면, RBF망은 주어진 데이터를 입력만 하면 자동으로 망을 구성하고 기저함수의 값을 결정하는 장점이 있다. 따라서 복잡한 문제를 풀고자 할 경우

표 7 유지보수 그룹별 비용 추정 RBF망 성능

그룹	모델명	입력	GOAL	기저함수 개수	결정 계수	MMRE
A	A_{RS}	UFP_{ADD}	10,000	31	0.9525	114.88%
AC	AC_{RS}	$UFP_{ADD+CHG}$	1,000	23	0.9182	91.46%
	AC_{RM}	UFP_{ADD}, UFP_{CHG}	100	37	0.9932	35.94%
ACD	ACD_{RS}	$UFP_{ADD+CHG+DEL}$	100	21	0.9979	18.68%
	ACD_{RM}	$UFP_{ADD}, UFP_{CHG}, UFP_{DEL}$	100	22	0.9983	17.14%
C	C_{RS}	UFP_{CHG}	5	7	0.9998	4.68%

표 8 모델 성능 종합

그룹	MMRE			
	통계적 모델		RBF 망	
	XXX_{SS}	XXX_{SM}	XXX_{RS}	XXX_{RM}
A	-	-	114.88%	-
AC	86.30%	-	91.46%	-
ACD	-	106.97%	18.68%	-
C	47.31%	-	-	-

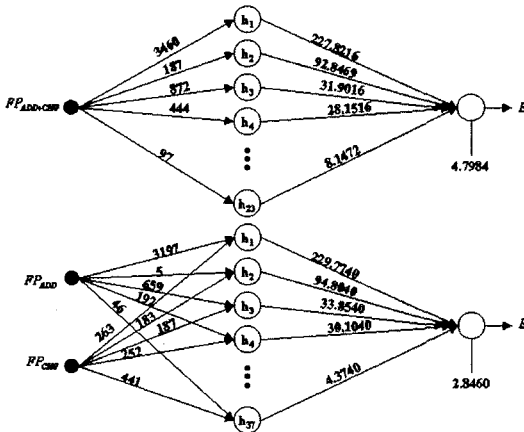


그림 3 그룹 AC에 대한 유지보수 비용 추정 RBF 망

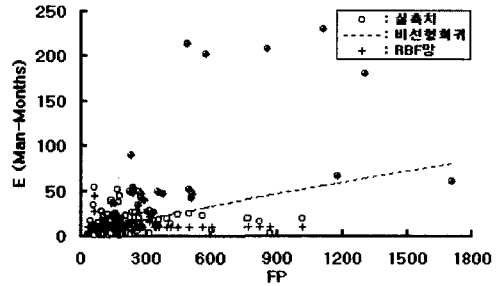
비선형함수를 근사시킬 수 있는 RBF망이 통계적 모델보다 좋은 장점이 있다.

4. 모델 평가

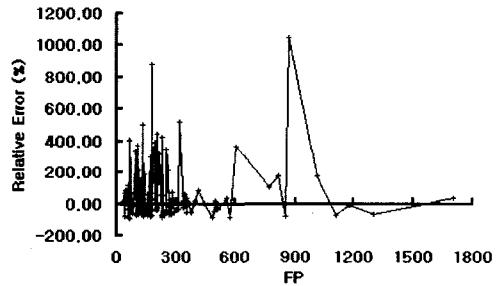
A, AC, ACD와 C 그룹에 대한 통계적 회귀모델과 RBF 망 모델의 성능을 종합한 결과는 표 8에 제시되어 있다. 예로 A 그룹에 대한 유지보수 비용 추정치와 상대오차는 그림 4에 제시되어 있다.

통계적 모델은 4개 그룹 중에서 단지 A 그룹에서만 RBF망 보다 좋은 성능을 보이고 있다. 따라서 RBF 망이 통계적 모델보다 좋은 모델임을 알 수 있다.

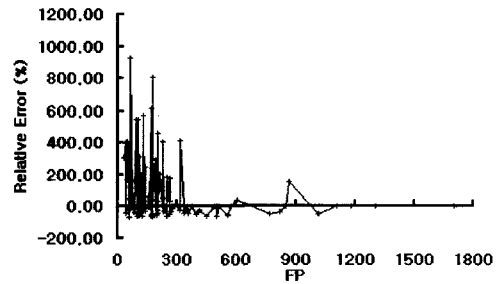
AC와 ACD 그룹에 대해, 입력이 1개인 경우와 2개 이상인 다변량인 경우를 비교하면 통계적 모델과 RBF 망 모두 AC 그룹은 다변량인 경우가, ACD 그룹은 입력이 1개인 경우가 보다 좋은 결과를 얻고 있다. 따라서 모델의 성능은 그룹의 특성에 따라 입력 변량을 다르게 선택하는 방법이 좋다. 위 그룹들에서 그룹 A는 순수한 기능 추가만을 수행하는 유지보수로 신규 개발되는 프로젝트와 동일한 형태로서 개발프로젝트로 분류가 가능하다.



(a) 비용 추정 결과



(b) 회귀모델 상대오차



(c) RBF망 상대오차

그림 4 A 그룹 유지보수 비용

따라서 A 그룹은 유지보수에서 발생하는 경우 수에서 삭제가 가능하다. 결론적으로 A 그룹을 제외한 나머지 3개 그룹의 경우, 표 3의 유지보수 비용 추정 모델들 보다 적은 상대오차를 보인다. 따라서 유지보수 발생 경우 별로 그룹을 분할하는 방법을 이용하여 비용을 추정하

는 본 제안 모델이 단일 그룹으로 고려하는 경우 보다 타당함을 알 수 있다.

5. 결론 및 향후 연구과제

유지보수를 수행하는 프로젝트 비중이 점차 커지고 있으나 비용을 추정하기 위한 모델들은 대부분 개발에 소요되는 비용에만 관심을 가지고 있다. 이로 인해, 개발비용 추정 모델을 유지보수 비용 추정에 적용함에 따라 추정 결과에 대한 신뢰성이 저하될 수 있어 유지보수 비용을 적절히 추정할 수 있는 모델에 대한 연구가 절실히 요구되고 있다.

본 논문은 먼저, 프로젝트 수행에 소요되는 비용을 추정하는데 필수적 요인인 규모 측정을 위해, 개발 프로젝트와 유지보수 프로젝트의 기능점수 계산방법을 분류하여 유지보수 프로젝트의 규모를 측정할 수 있는 방법을 제시하였다. 또한, 유지보수 발생 경우 수를 고려하여 그룹화한 결과 추가, 변경과 삭제 3가지로 발생할 수 있는 8가지 경우 수 중 실제로는 4개 경우 수인 A, AC, ACD와 C 그룹을 관찰하였다. 통계적 회귀모델과 RBF 망을 적용하였으며, 모델의 입력이 1개인 경우와 2개 이상인 다변량인 경우로 분류하여 그룹별로 모델을 개발하였다.

A 그룹은 새로운 기능이 추가된 경우로, 개발 프로젝트와 동일한 프로파일을 가지고 있다. 따라서 A 그룹을 개발 프로젝트로 분류하고 유지보수 프로젝트 경우에서 제외시킬 수 있다. A 그룹을 제외하고 AC, ACD와 C 그룹에 대해 분석한 결과 제안된 모델들이 전체 데이터를 단일 그룹으로 하는 모델보다 성능이 향상되어 제안된 방법이 타당함을 보였다.

앞으로, 규모 추정에 개발 프로젝트의 기능점수 계산 방법을 적용할 수 있는지 여부에 대한 연구를 수행할 것이다.

참고 문헌

[1] F. Niessink and H. V. Vliet, "Two Case Studies in Measuring Software Maintenance Effort," In Proceedings of the International Conference on Software Maintenance, Bethesda, Maryland, USA, pp. 76-85, 1998.

[2] J. E. Matson, B. E. Barrett and J. M. Mellichamp, "Software Development Cost Estimation Using Function Points," IEEE Trans. on Software Eng., Vol.20, No.4, pp. 275-287, 1994.

[3] F. Niessink and H. V. Vliet, "Predicting Maintenance Effort with Function Points," In Proceedings of ICSM97, Bari, Italy, 1997.

[4] L. A. Laranjeira, "Software Size Estimation of Object-Oriented Systems," IEEE Trans. Software

Eng., Vol.16, pp. 64-71, 1990.

- [5] A. J. Albrecht, "Measuring Applications Development Productivity," Proceedings of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA., pp. 83-92, 1979.
- [6] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Line of Code and Development Effort Prediction : A Software Science Validation," IEEE Trans. on Software Eng., Vol. SE-9, No.6, pp. 639-648, 1983.
- [7] C. F. Kemerer, "Reliability of Functional Point Measurement - A Field Experiment," Communications of ACM, 1993.
- [8] IFPUG, "Function Point Counting Practice Manual," Release 4.1.1 IFPUG, Troy Michigan, 2000.
- [9] B. W. Boehm, "Software Engineering Economics," Prentice-Hall, 1981.
- [10] B. W. Boehm et al, "Software Cost Estimation with COCOMO II," Prantice-Hall, 2000.
- [11] NESMA, "Function Point Analysis for Software Enhancement," NESMA, 2001.
- [12] Cote and St-Pierre, "A Model for Estimating Perfective Software Maintenance Projects," Proceedings of Conference on Software Maintenance, Vol. 11, pp. 328-334, 1990.
- [13] 권기태, 신수정, "소프트웨어 유지보수 비용산정 모델 개선," 한국정보과학회 소프트웨어공학회지 제16권 제2호, pp. 31-42, 2003.
- [14] M. Bradley, "Function Point Counting Practices Manual, Release 4.1," International Function Point Users Group (IFPUG), 1999.
- [15] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.
- [16] O. S. J. Lima, P. P. M. Farias, and A. D. Belchier, "Maintenance Project Assessments Using Fuzzy Function Point Analysis," Department of Computer Science, University of Fortaleza, 2001.
- [17] W. H. Roetzheim, "Creating The Project Plan," Cost Xpert Group, Inc., 2001.
- [18] M. Jorgensen, "An Empirical Study of Maintenance Tasks," Software Maintenance: Research and Practice, Vol. 7, pp. 27-48, 1995.
- [19] A. Abran and H. Nguyenkim, "Measurement of the Maintenance Process from a Demand-based Perspective," Software Maintenance: Research and Practice, Vol. 5, pp. 64-90, 1993.
- [20] K. Ribu, "Estimating Object-oriented Software Projects with Use Cases," University of Oslo Department of Informatics, Master of Science Thesis, 2001.
- [21] B. Kitchenham and K. Knsl, "Inter-item Correlation Among Function Points," National Computing Centre Ltd, UK and VTT, Finland, 1997.
- [22] S. Chen, C. F. N. Cowan, and P. M. Grant,

"Orthogonal Least Square Learning for Radial Basis Function Networks," IEEE Trans. on Neural Networks, Vol. 2, No. 2, pp. 302-309, 1991.



박 주 석

1984년 해군사관학교 전자공학과(공학사). 1995년 국방대학원 전자계산학과(전산학 석사). 2001년~현재 국방대학교 직무연수부 정보화과정 담임교수. 2001년~현재 숭실대학교 일반대학원 컴퓨터학과 박사과정 수료. 관심분야는 비용산정, 표

준 및 프로세스, 정보채계사업관리, 소프트웨어 품질보증, 정보전



정 기 원

1967년 서울대학교 전기공학과(공학사). 1981년 미국 알라바마주립대 전산학과(전산학석사). 1983년 미국 텍사스주립대 전산학과(전산학박사). 1966년~1968년 미 8군(IBM 기계정비 담당). 1971년~1975년 한국과학기술연구소. 1975년1990년

국방과학연구소(책임연구원). 1990년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 소프트웨어 프로세스, 소프트웨어 개발방법론, 정보시스템, 전자거래(CALS/EC)