

주성분분석을 이용한 소프트웨어 개발노력 추정능력 향상

이 상 운[†]

요 약

Putnam은 소프트웨어 프로젝트에 참여하는 인력이 Rayleigh 분포를 따르는 SLIM 모델을 제시하였다. 이 모델에서 인력분포를 얻기 위해서는 총 개발노력과 개발 난이도를 추정해야 한다. 프로젝트 개발에 참여할 것인지 여부를 결정하기 위해서는 소프트웨어 생명주기의 초기단계에서 이 모수들을 보다 정확히 추정하는 것이 필요하다. Putnam은 시스템 속성들 중 강한 상관관계가 있는 변량을 제거하고 나머지 변량들만으로 총 개발노력과 개발 난이도를 추정하였다. 그러나 통계적 방법에 따라 변량들이 다르게 선택되며 모델의 성능에 차이가 발생한다. 본 논문은 Putnam 방법 대신 주성분분석을 이용하여 최적의 시스템 속성을 선택하였다. 모델의 성능분석 결과 주성분분석 방법이 Putnam의 방법보다 9.85% 성능향상을 보였다. 또한, 제안된 모델은 단순하고 쉽게 구현할 수 있다.

Improving Estimation Ability of Software Development Effort Using Principle Component Analysis

Sang-Un Lee[†]

ABSTRACT

Putnam develops SLIM (Software Lifecycle Management) model based upon the assumption that the manpower utilization during software project development is followed by a Rayleigh distribution. To obtain the manpower distribution, we have to estimate the total development effort and difficulty ratio parameter. We need a way to accurately estimate these parameters early in the requirements and specification phase before investment decisions have to be made. Statistical tests show that system attributes are highly correlation (redundant) so that Putnam discards one and get a parameter estimator from the other attributes. But, different statistical method has different system attributes and presents different performance. To select the principle system attributes, this paper uses the principle component analysis (PCA) instead of Putnam's method. The PCA's results improve a 9.85 percent performance more than the Putnam's result. Also, this model seems to be simple and easily realize.

키워드 : 개발노력(Development Effort), 난이도(Difficulty Ratio), 다중회귀분석(Multiple Regression Analysis), 주성분분석(Principle Component Analysis), 시스템 속성(System Attribute)

1. 서 론

소프트웨어 생명주기(Life-cycle)의 초기단계, 즉, 요구명세(Requirement Specification) 단계에서 우리가 알 수 있는 사항은 사용자들이 요구하는 일반적인 기능 요구사항, 개발 일정과 비용뿐이다. 또한 이 시점에서 알지 못하거나 매우 애매 모호한 것들은 시스템 규모, 실행 가능한 일정, 실행 가능한 개발일정에 일치하는 최소한의 총 개발인력과 비용들이다. 이 시점에서, 프로젝트 개발에 투자를 할 것인가를 결정을 하기 위해서는 보다 정확한 제품의 규모, 실행 가능한 일정, 개발에 필요한 총 인력과 시스템 운영단계에서 소프트웨어 유지보수 비용 추정 데이터가 절실히 요구된다.

소프트웨어의 개발 단계별 및 전체 개발에 투입되는 노력 분포에 대해 Putnam의 SLIM(Software Lifecycle Management) 또는 Norden/Rayleigh 모델[1]과 Pillai et al.[2]의 Gamma 모델이 있다. 이들 모델을 이용하여 개발의 각 시점에서 투입되는 노력의 규모는 총 개발노력이 Rayleigh 분포나 Gamma 분포를 따르므로 총 개발노력(Total Development Effort 또는 Lifecycle Effort)을 추정하면 개발 각 시점에서의 투입되는 노력의 규모를 산정할 수 있다.

소프트웨어 개발에 투입되는 총 개발노력의 규모를 추정하는 방법으로 LOC(Line Of Code)를 이용하는 방법, 기능포인트(Function Point : FP)를 이용하는 방법[3, 4]과 소프트웨어의 시스템 속성들인 입력, 출력, 조회와 화일의 수 등을 직접 이용하는 방법이 있다. LOC를 이용하는 방법에는 대표적으로 Boehm의 COCOMO(Constructive Cost Model)

[†] 정 회 원 : 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당
논문접수 : 2001년 6월 11일, 심사완료 : 2001년 10월 9일

모델[5, 6]이 있다. 이 방법은 소프트웨어의 소스코드인 LOC를 기반으로 하기 때문에 개발 초기단계에서는 사용이 불가능하고, 코딩단계가 끝난 시점에야 정확한 계산이 될 수 있다. FP 기법은 사용자에게 양도될 시스템의 기능으로 소프트웨어 시스템의 규모와 복잡도를 정량화하는 방법이다. 이 기법은 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적, 개발 생명주기의 초기단계인 요구분석 단계에서 측정 가능한 장점을 갖고 있으며, LOC를 사용할 때의 주요 문제점을 극복할 수 있는 접근법이다[7]. 그러나 FP 기법을 이용해 소프트웨어 노력을 추정하기 위한 모델들[7, 8]은 소프트웨어의 정보영역과 복잡도에 대해 주관적인 평가의 계수 측정을 거쳐 얻어지는 알고리즘 기법이다.

Putnam은 시스템 속성들인 화일, 출력과 응용프로그램의 수들 중 통계적 검정(Statistical Test) 결과 화일과 출력이 강한 상관(중복)을 가져 화일 속성을 제거하고 나머지 2개 속성만을 이용하여 총 개발노력을 추정하였다.

주어진 데이터에 적합한 모델을 찾기 위해 회귀분석을 하는 경우, 회귀직선에 의해 종속변수가 설명되는 정도를 결정계수(Coefficient of Determination)라 하며, 모델의 성능을 결정하는 주요 요인이 된다. 즉, 종속변수(총 개발노력)의 값은 독립변수(화일, 출력과 응용 프로그램의 수)에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변동을 설명하는데 있어서 회귀직선에 의해 설명되는 변동이 기여하는 비율이 결정계수이다.

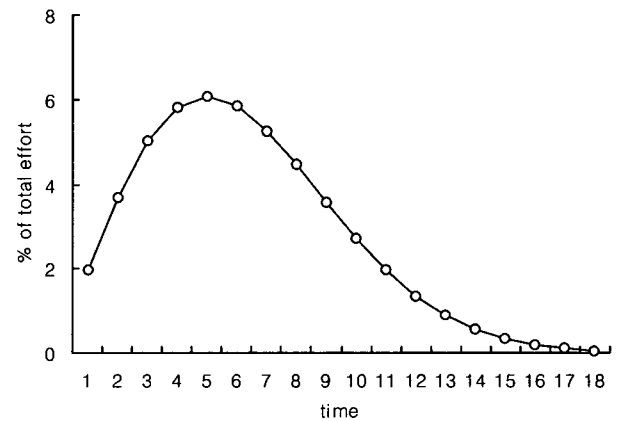
본 논문은 종속변수인 총 개발노력을 추정하는데 영향을 미치는 독립변수들 중에 모든 독립변수들을 사용하지 않고 중복되거나 영향이 적은 변수들을 제거하고 나머지 변수들만을 이용해 모델을 개발하는 방법에 대해 논한다. 통계적 변량(독립변수)을 선택하는 방법 중 Putnam[1]은 상관관계를 이용하였는데 비해 본 논문은 주성분 분석(Principle Component Analysis : PCA)을 이용하여 필요한 시스템 속성들을 선택한다. Putnam[1]이 선택한 시스템 속성들을 이용한 모델과 성능을 비교함으로써, 변량 결정 방법이 모델의 성능에 미치는 영향의 중요성을 제시하며, 주성분 분석으로 선택된 변량들만을 이용할 경우, 모든 변량들을 이용하는 모델보다 모델이 단순한 장점을 갖고 있으면서, 성능도 약간 저하되는 경향을 나타낸다.

2장에서는 Putnam[1]의 SLIM 모델과 총 개발노력을 추정하는 방법들을 살펴본다. 3장에서는 Putnam[1]의 SLIM 모델에 포함된 모수인 총 개발노력과 개발 난이도에 영향을 미치는 주요한 변량들을 선택하는 방법을 제시하며, 4장에서는 Putnam[1]의 방법과 주성분분석 방법으로 얻어진 모델의 성능을 비교, 분석하여 본다.

2. 관련 연구 및 연구배경

Norden[10]은 IBM에서 개발된 다양한 하드웨어 개발과정

에 투입된 인력 (Manpower)의 근사모델로 Rayleigh 분포를 따른다는 사실을 관찰하였다. 이후 이 관찰 결과가 Putnam[1]에 의해 소프트웨어 프로젝트에 적용되었다. 150개의 소프트웨어 프로젝트가 Norden[11]과 Putnam[1]에 의해 연구되었으며, 프로젝트 단계의 개발에 참여하는 전체 인력(단위시간당 소요 인원) 뿐만 아니라 개발 사이클의 각 단계(즉, 요구명세, 분석, 설계, 코딩 등)에 대한 인력도(그림 1)의 Rayleigh 함수로 표현됨을 관찰하였다.



(그림 1) Rayleigh 분포 ($K=1.00$, $a=0.02$)

입의의 시간 t 까지 프로젝트에 투입된 누적 인력 y 를 식 (1)로, 입의의 시간 t 에서 프로젝트에 투입되는 인력 \dot{y} 는 식 (1)을 미분한 식 (2)가 된다.

$$y = K(1 - e^{-at^2})MY \tag{1}$$

$$\dot{y} = 2Kate^{-at^2}MY/YR \tag{2}$$

여기서 K 는 Rayleigh 곡선 아래에 있는 영역으로 연인원 (Man-Year)으로 표시되며 생명주기 전체에 소요되는 총 개발노력이다. $a = \frac{1}{2t_d^2}$ 이며, t_d 는 \dot{y} 이 최대가 되는 시점으로 경험적으로 볼 때, 시스템이 운영되는 시점에 근접한다. 따라서, $t_{y_{max}} = t_d$ 를 시스템의 개발기간이라 한다. 식 (2)에서 $a = \frac{1}{2t_d^2}$ 를 치환하면, 식 (3)이 된다.

$$\dot{y} = \frac{K}{t_d^2} te^{-t \frac{2}{2t_d^2}} \tag{3}$$

식 (3)에서 시스템에 투입되는 인력은 총 개발노력 K , 개발기간 t_d 와 이들 모수들로 이루어진 개발 난이도(Difficulty Ratio, D)인 $\frac{K}{t_d^2}$ 로 구성되어 있다.

Pillai et al. [2]은 Putnam의 SLIM 모델의 단점을 보완하기 위해 Gamma 모델을 이용해 식 (4)의 개발노력 분포를 제안하였다.

$$\dot{y} = \frac{4K}{t_d^3} t^2 e^{-\frac{2t}{t_d}} \text{ MY/YR} \quad (4)$$

Putnam의 SLIM 모델을 이용하여 개발의 각 시점에서 투입되는 노력의 규모를 추정하고자 하려면, 모델이 갖고 있는 총 개발노력 (K) 과 개발 난이도 (D) 를 추정해야만 한다. 총 개발노력을 추정하기 위한 방법들을 살펴보자. 먼저, LOC를 이용한 소프트웨어 생명주기 노력(Life Cycle Effort) 추정 모델은 식 (5)의 형태를 취한다.

$$K = a + b \cdot KLOC^c \quad (5)$$

여기서, 총 개발노력 K 는 예측된 생명주기 노력으로 Man-Months 또는 Man-Years로 측정되고, a, b, c 는 상수이다. KLOC(Thousands of Line Of Code) 최종 코딩된 소프트웨어의 라인 수이다. 소프트웨어 규모 측정 단위로 LOC를 사용할 경우 LOC에 대한 일반적으로 받아들일 수 있는 정확한 정의 부족, 언어에 종속, 요구분석 또는 설계단계에서 정확한 LOC의 추정 어려움, 규모에 대한 특정한 하나의 관점인 길이만 고려하며 소프트웨어의 기능성 또는 복잡도(Complexity)를 고려하지 않는 문제점이 있다.

FP를 이용한 노력 추정 방법은 노력 K 가 FP에 대해 식 (6)의 형태를 취한다.

$$K = a + b \cdot FP \quad \text{또는} \quad \sqrt{K} = a + b \cdot FP \quad (6)$$

FP 기법을 이용해 소프트웨어 노력을 추정하기 위한 모델들은 소프트웨어의 정보영역과 복잡도에 대해 주관적인 평가의 계수 측정을 거쳐 얻어지는 알고리즘 기법으로, 객관적인 평가기준을 적용할 수 없다는 단점이 있다. 따라서, 시스템 속성들로부터 측정된 FP를 이용하는 방법 대신 직접 시스템 속성들을 이용하여 총 개발노력을 추정하는 방법도 개발되었다.

Putnam은 <표 1>의 시스템 속성들인 화일(FILE), 출력(RPT)과 응용프로그램의 수(APP)들을 이용하여 직접 총 개발노력 K 를 추정하였다. Putnam[1]은 K 와 t_d 는 화일 수(FILE), 출력 수(RPT)와 응용 프로그램 수(APP)와 같은 시스템 속성에 선형적으로 독립적이지 못하나, 개발 난이도 D 는 시스템 속성의 개개와 결합된 형태에 매우 선형적인 관계를 가짐을 밝혔다. 즉, K 와 D 를 시스템 속성들에 대한 함수로 표현된다. 즉, $K = f(FILE, RPT, APP)$ 또는 $D = f(FILE, RPT, APP)$. 이들 독립변수들은 프로그램과 시스템 구축과정에 직접 관련되어 있으며, 요구사항 명세화 단계의 초기에 약 10~15% 오차 범위내의 정확도로 추정이 가능하기 때문에 선정되었다[1]. Putnam[1]은 모수 K 와 D 에 영향을 미치는 시스템 속성을 선택하는데 있어 통계적 검정결과 화일 수(FILE)와 출력 수(RPT)가 높은 상관(중복)을 갖고 있어, 2개 변량(Variable) 중 1개를 제거한 나머지 변량과 응용 프로그램 수(APP)변량에 대한 함수로 표현하였

다. 즉, $K = f(FILE, APP)$ 와 $D = f(FILE, APP)$ 또는 $K = f(RPT, APP)$ 와 $D = f(RPT, APP)$ 가 가능하며 $K = f(RPT, APP)$ 와 $D = f(RPT, APP)$ 를 이용하였다. 위에서 얻어진 식으로 <표 1>의 데이터에 대한 다중회귀분석을 이용하여 식 (7)과 식 (8)의 K 와 D 를 추정하였다.

$$K = -0.3447 \cdot RPT + 2.7931 \cdot APP \quad (7)$$

$$D = 0.1991 \cdot RPT + 0.0859 \cdot APP \quad (8)$$

<표 1> U.S. Army Computer System Command 데이터

System	시스템 속성			총 개발노력 K (MY)	개발기간 t_d (YRS)
	화일 수 (FILE)	출력 수 (RPT)	응용프로그램 수(APP)		
MPMIS	94	45	52	73.60	2.28
MRM	36	44	31	84.00	1.48
ACS	11	74	39	33.00	1.67
SPBS	8	34	23	70.00	2.00
COMIS	14	41	35	27.50	1.44
AUDIT	11	5	5	10.00	2.00
CABS	22	14	12	7.74	1.95
MARDIS	6	10	27	91.00	2.50
MPAS	25	95	109	101.00	2.10
CARMOCS	13	109	229	153.00	2.64
SIDPERS	172	179	256	700.00	3.65
VTAADS	155	101	144	404.00	3.50
BASOPS-SUP	81	192	223	591.00	2.73
SAILS AB/C	540	215	365	1028.00	4.27
SAILS-AB/X	670	200	398	1193.00	3.48
STARCIPI	151	59	75	344.00	3.48
STANFINS	270	228	241	741.00	3.30
SAAS	131	152	120	118.00	2.12
COSCOM	33	101	130	214.00	4.25

통계적 변량을 선택하는 방법에는 평균과 중앙값, 표본 평균/분산 비교, 표본 평균에 대한 모멘트, 표본 공분산/상관 행렬과 주성분분석 방법 등이 있다. 처음 3가지 방법은 단일 차원에 대한 변량 선택 방법이며, 마지막 2개 방법이 다차원 성질에 적용된다. 표에서 알 수 있듯이 노력에 영향을 미치는 변량들은 화일 수(FILE), 출력 수(RPT)와 응용프로그램 수(APP) 등 단일차원이 아니라 다차원이다. 따라서, 표본의 공분산/상관 행렬이나 주성분 분석을 이용하여 통계적 변량을 선택할 수 있다. 표본의 공분산/상관 행렬은 다른 차원을 가진 분포들의 관계를 찾을 때 유용하게 사용되며, 변량들 간의 강한 상관관계를 알기 위해 적용한다. Putnam[1]은 주요한 시스템 속성을 선택할 때 이 방법을 사용하였다. 본 논문은 총 개발노력과 개발 난이도를 추정하는데 어떤 시스템 속성들이 주요한 영향을 미치는지에 대해 Putnam의 상관관계를 가진 속성들을 제거하는 방법 대신, 주성분 분석 방법을 이용하고자 한다.

3. 총 개발노력과 개발 난이도 추정 모델에 영향을 미치는 변수 선택

Putnam[1]의 SLIM 모델에서 모수들인 총 개발노력 K 와

개발기간 t_d 를 알고 있다면, 임의의 시간 t 에서 소프트웨어 프로젝트의 노력, 개발인원, 개발비용과 누적 비용 등을 계산할 수 있다. 이와 더불어, 개발과정 초기 단계에서 모수 K 와 t_d 를 결정할 수 있는 방법이 존재한다면, 보다 정확한 투자 결정을 할 수 있을 것이다.

본 논문은 K 와 D (종속변수)에 영향을 미치는 주요한 변량인 시스템 속성(독립변수)들을 결정하는데 상관관계를 이용하지 않고 주성분분석 방법을 적용한다. 주성분분석은 고차원 벡터들을 포함하고 있는 데이터 집합을 분석하는데 유용하게 사용될 수 있는 가장 단순한 다변량 방법 중의 하나이다. 어떤 개체를 설명하는데 p 종의 데이터가 있는 경우, 서로 강한 상관(중복된)이 있는 p 개의 변량 x_1, x_2, \dots, x_p 를 변환하여 상관이 없는 독립된 변량을 구하고, 이들을 x_1, x_2, \dots, x_p 순으로 변동의 설명력이 큰 것부터 선택하면, 최초 변동의 대부분이 설명되는 것으로 생각되는 것이며, 이 데이터로부터 추출한 새로운 변량을 주성분이라 한다 [8]. 예로, <표 1>에서 *FILE*, *RPT*와 *APP* 3개 변량들이 서로 강한 상관관계를 갖고 있다고 할 때, 이들 변량들을 변환시켜 상관이 없는 독립된 변량들로 만들고 최초의 변동 설명력이 큰 것부터 나열하여 (예로, *FILE* > *RPT* > *APP*) 이들 중에서 순서적으로 몇 개만을 선택하면 선택된 변량들이 주성분이 되며, 이 주성분만을 이용하는 경우 최초 변동의 대부분이 설명된다.

<표 1>의 데이터 변동에 대해 MATLAB에서 구현된 주성분분석 프로그램을 이용하여 3% 이하로 기여하는 변량을 제거하는 주성분 분석을 수행하였다. 주성분분석 결과 *APP*가 총 변동의 3% 이하를 설명하여 주성분에서 제거되었으며, *FILE* > *RPT* > *APP* 순으로 변동의 설명력인 큰 결과를 나타내었다. 즉, 총 개발노력 또는 개발 난이도가 Putnam[1]이 제안한 *RPT*와 *APP*의 함수로 표현되는 것이 아니라 식 (9)와 식(10)과 같이 *FILE*과 *RPT*의 함수인 $K = f(\text{FILE}, \text{RPT})$ 와 $D = f(\text{FILE}, \text{RPT})$ 가 보다 좋은 추정치가 됨을 알 수 있다.

$$K = a \cdot \text{FILE} + b \cdot \text{RPT} \quad (9)$$

$$D = a \cdot \text{FILE} + b \cdot \text{RPT} \quad (10)$$

4. 사례연구 및 비교분석

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 추정 정확도를 평가하는 것이 필요하다. 회귀분석의 경우 회귀직선에 의해 종속변수가 설명되는 정도를 나타내는 결정계수 R^2 를 사용한다. 종속변수의 값은 독립변수에 의해 결정되는 부분과 미지의 오차의 합으로 나타나며, 총 변동을 설명하는데 있어서 회귀직선에 의해 설명되는 변동이 기여하는 비율이 R^2 이다.

주성분분석 결과 얻어진 식에 대해 MATLAB의 다중회귀분석 프로그램을 수행하였다. <표 2>에 Putnam[1]의 방법과 주성분분석을 이용해, 선택된 시스템 속성들에 따른 총 개발노력 K 와 개발 난이도 D 를 추정한 다중회귀분석 모델을 제시하였다.

<표 2> 총 개발노력 K 와 개발난이도 D 추정

추정 대상	시스템 속성 선택 방법	선택된 시스템 속성	다중회귀분석 모델	성능 (R^2)
총 개발노력 (K)	Putnam [1]의 방법	RPT APP	$-0.3447RPT + 2.7931APP$	74.21%
	주성분분석 이용	FILE RPT	$1.2878FILE + 1.6363RPT$	84.06%
개발 난이도 (D)	Putnam [1]의 방법	RPT APP	$0.1991RPT + 0.0859APP$	83.09%
	주성분분석 이용	FILE RPT	$0.0475FILE + 0.2484RPT$	88.32%

<표 2>의 모델 성능에 대해 비교 분석한 결과는 <표 3>과 같다.

<표 3> 모델 성능 비교

추정 대상	모델 성능		성능 향상
	Putnam [1]의 방법	주성분분석 이용	
총 개발노력 (K)	74.21%	84.06%	9.85%
개발 난이도 (D)	83.09%	88.32%	5.23%

표에서 Putnam[1]의 상관관계를 이용하여 변량을 선택하는 방법보다 주성분분석을 이용하는 방법이 총 개발노력이나 개발 난이도에 모두 대해 향상된 성능을 보였다. 예로, 개발되는 시스템의 속성이 화일 수(*FILE*)가 13, 출력 수(*RPT*)가 109, 응용 프로그램의 수(*APP*)가 229개일 경우 총 개발노력 K 는 153 MY이었다. 이 시스템에 대해 Putnam[1]의 상관관계를 이용한 모델인 $K = -0.3447RPT + 2.7931APP$ 과 주성분분석으로 선택된 본 제안 모델인 $K = 1.2878FILE + 1.6363RPT$ 의 성능을 비교하여 보자. Putnam[1]의 모델로 추정된 총 개발노력은 $\hat{K} = -0.3447 \times 109 + 2.7931 \times 229$ 로 602.0476이 계산되며, 주성분분석으로 선택된 시스템 속성들을 사용한 본 제안 모델로 추정된 총 개발노력 $\hat{K} = 1.2878 \times 13 + 1.6363 \times 109$ 는 195.0981로 계산된다. 두 모델의 오차를 계산하면 Putnam[1]의 모델은 486.6199, 본 제안모델은 42.0981가 된다. 또한 상대오차 ($\frac{\hat{K} - K}{K} \times 100\%$)는 293.4952%와 27.5151%로 본 제안 모델의 성능이 월등히 좋음을 알 수 있다.

전체 시스템 속성을 사용한 회귀분석 모델 $K = f(\text{FILE}, \text{RPT}, \text{APP})$ 와 $D = f(\text{FILE}, \text{RPT}, \text{APP})$ 인 경우의 R^2 와 평균제곱오차(Mean Squared Error, MSE)는 <표 4>와 같다.

〈표 4〉 전체 시스템 속성을 사용한 경우 모델 성능

추정 대상	다중회귀분석 모델	R ²	MSE
총 개발노력 (K)	1.0176FILE+0.5751RPT+1.0541APP	86.01%	9190.44
개발 난이도 (D)	0.0441FILE+0.2350RPT+0.0132APP	88.42%	164.4524

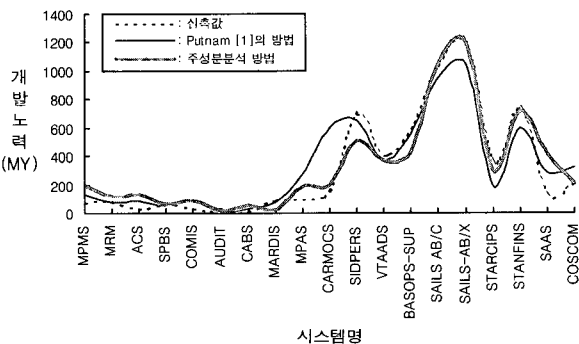
전체 시스템 속성을 사용한 경우에 비해 주성분분석과 Putnam[1]의 상관관계를 이용해 선택된 시스템 속성을 이용한 모델의 성능 저하는 <표 5>에 제시되어 있다.

〈표 5〉 시스템 속성 선택에 따른 모델 성능 비교

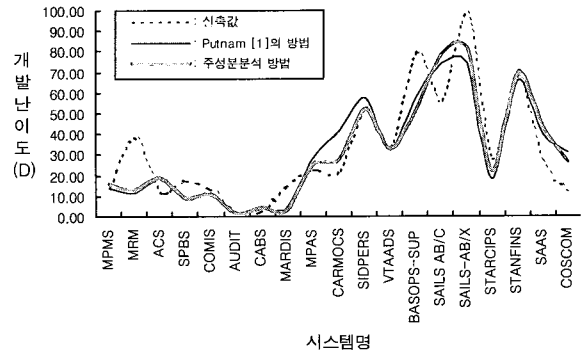
추정대상	전체 시스템 속성 사용 모델	Putnam [1]의 방법		주성분분석 이용	
	성능 ①	성능 ②	성능감소 (①-②)	성능 ③	성능감소 (①-③)
총 개발노력 (K)	86.01%	74.21%	11.80%	84.06%	1.95%
개발 난이도 (D)	88.42%	83.09%	5.33%	88.32%	0.1%

주성분분석을 이용하여 시스템속성들을 선택한 모델의 경우, Putnam[1]의 상관관계로 선택된 시스템 속성으로 추정된 모델에 비해 총 개발노력과 개발 난이도 모두에 대해 약간의 성능 저하만을 나타내었다. 즉, 주성분분석을 사용시 원래의 3개 변량(FILE, RPR, APP)을 2개 변량(FILE, RPR)으로 줄여 모델의 단순성을 취하는 장점이 있으며, 최초 변동의 대부분을 설명하는 주요 변량들만 이용하여도 전체 변량을 사용한 경우에 비해 단지 약간 저하된 추정 능력을 나타낸다.

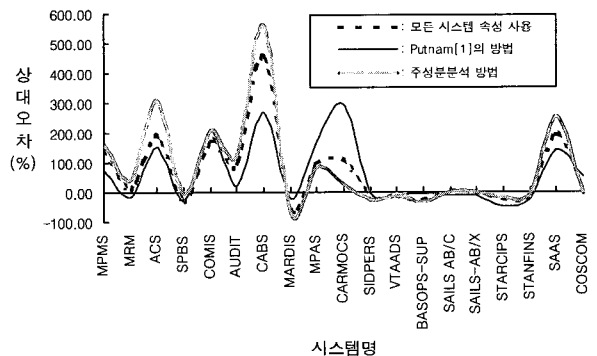
<표 1>의 각 시스템에 대한 총 개발노력과 개발 난이도 추정 결과는 (그림 3)과 (그림 4)에 제시하였다. 또한, 각 시스템에 대한 추정 상대오차는 각각 (그림 5)와 (그림 6)에 표기하였다. 추정치와 상대오차 그림에서 알 수 있듯이 Putnam[1]의 상관관계로 선택된 시스템 속성들을 이용하는



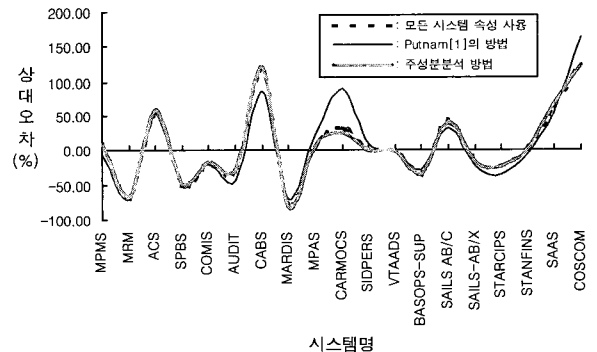
(그림 3) 총 개발노력 추정



(그림 4) 개발 난이도 추정



(그림 5) 총 개발노력 추정 상대오차



(그림 6) 개발 난이도 추정 상대오차

방법보다 주성분분석으로 선택된 시스템 속성들로 추정되는 총 개발노력과 개발 난이도가 실측값에 보다 근사한 값을 얻을 수 있다.

5. 결론 및 향후 과제

본 논문은 소프트웨어 초기 단계에서 개발에 참여할 것인가를 결정하는데 필요한 소프트웨어 생명주기 노력과 비용추정 모델의 모수인 총 개발노력과 개발 난이도에 영향을 미치는 주요한 시스템 속성을 결정하는 방법을 제시하였다. 제시된 방법은 모델의 모수에 영향을 미치는 시스템 속성을 결정하기 위해, 통계적 이론에 기반을 둔 주성분분석 방법이다. 주성분분석 방법으로 선택된 시스템 속성들만

을 이용하여 총 개발노력과 개발 난이도를 추정하는 본 제안 모델이 Putnam[1]의 상관관계로 선택된 시스템 속성들을 이용하는 모델보다 성능이 향상되었으며, 전체 시스템 속성들을 사용하는 경우보다 모델이 단순하면서도 약간의 성능저하만을 나타내었다. 따라서, 소프트웨어의 FILE과 RPT의 수만을 측정하면 본 제안 모델을 이용하여 소프트웨어 개발에 소요되는 노력과 개발 난이도를 추정할 수 있으며, Putnam[1]의 SLIM 모델에 적용하면 소프트웨어 개발 과정에서 투입되는 노력의 분포를 구할 수 있다. 따라서, 본 제안 모델을 이용할 경우, 소프트웨어 개발 초기에 개발에 참여할 것인가 여부를 결정하는데 필요한 정책적 데이터를 보다 정확하고 간단히 얻을 수 있으며, 소프트웨어 개발과정에서의 인력 분배에도 활용이 될 수 있다.

본 실험에 사용된 데이터는 Putnam[1]이 79년에 발표한 논문의 데이터를 사용한 관계로, 최근의 복잡하고 고 신뢰성을 요구하는 소프트웨어 개발환경과 개발기법을 반영하지 못해 실제 적용에 단점을 갖고 있다. 그러나, 총 개발노력과 개발 난이도를 추정하기 위해 필요한 시스템 속성들을 선택하는 방법은 주성분분석 방법을 사용하는 것이 가장 좋은 방법이라 판단된다. 본 제안된 모델을 다양한 개발환경과 조건에서 개발된 다수의 소프트웨어에 적용할 수 있는 일반화된 모델로 발전시키기 위해서는 최근의 다양한 소프트웨어 개발환경과 기법들을 적용한 789개의 소프트웨어 프로젝트 데이터를 보유하고 있는 ISBSG Benchmark Release 6[12]에 적용하여 검증하는 것이 필요하다. 또한, 개발노력을 추정하는데 영향을 미치는 또 다른 변수들이 있는지에 대해서도 추가적인 연구가 필요하다. 따라서, 추후 이 분야에 대한 연구를 수행할 것이다.

참 고 문 헌

[1] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol.SE-4, No.4, 1978.
 [2] K. Pillai and V. S. S. Nair, "A Model for Software Development Effort and Cost Estimation," IEEE Trans. on Software Eng., Vol.23, No.8, pp.485-497, 1997.
 [3] 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.
 [4] 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.
 [5] B. W. Boehm, "Software Engineering Economics," Prentice Hall, 1981.
 [6] B. W. Boehm, "Software Engineering Economics," IEEE Trans. on Software Eng., Vol.10, No.1, pp.7-19, 1984.
 [7] A. J. Albrecht, "Measuring Application Development Productivity," in Programming Productivity : Issues for the Eighties, C. Jones, ed. Washington, DC : IEEE Computer Society Press, 1981.
 [8] C. F. Kemerer, "An Empirical Validation of Software Cost Estimation Models," Communication ACM, Vol.30, No.5, pp. 416-429, 1987.
 [9] W. F. Massy, "Principle Component Regression in Exploratory Statistical Research," Journal of the American Statistical Association, Vol.60, pp.234-246, 1965.
 [10] P. V. Norden, "Curve Fitting for a Model of Applied Research and Development Scheduling," IBM J. Research and Development, Vol.3, No.2, pp.232-248, 1958.
 [11] P. V. Norden, "Project Life Cycle Modeling : Background and Application of the Life Cycle Curves," U. S. Army Computer System Command, 1977.
 [12] ISBSG, "Worldwide Software Development - The Benchmark Release 6," Victoria, Australia International Software Benchmarking Standards Group, 2000.



이 상 운

e-mail : sangun_lee@hanmail.net

1983년~1987년 한국항공대학교 항공전자 공학과(학사)

1995년~1997년 경상대학교 컴퓨터과학과 (석사)

1998년~2001년 경상대학교 컴퓨터과학과 (박사)

1992년~현재 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질보증, 소프트웨어 신뢰성), 소프트웨어 매트릭스, 신경망, 뉴로-퍼지