논문 2012-2-4

CfsSubsetEval 속성 선택기를 이용한 주요 속성 조합 예측

김영옥*, 권기태**

Predicting a combination of the key attributes using CfsSubsetEval attribute selector

YoungOk Kim*, KiTae Kwon**

요 약

특징 선택은 기계 학습 및 패턴 인식 분야에서 중요한 이슈 중 하나로, 분류 정확도를 향상시키기 위해 원본 데이터가 주어졌을 때 가장 좋은 성능을 보여줄 수 있는 데이터의 부분집합을 찾아내는 방법이다. 즉, 분류기의 분류 목적에 가장 밀접하게 연관되어 있는 특징들만을 추출하여 새로운 데이터를 생성하는 것이다. 본 논문에서는 소프트웨어 재사용의 성공 요인과 실패 요인에 대한 분류 정확도를 향상시키기 위해 특징 부분 집합을 찾는 실험을 하였다. 그리고 기존 연구들과 비교 분석한 결과 본 논문에서 찾은 특징 부분 집합으로 분류했을 때 가장 좋은 분류 정확도를 보임을 확인하였다.

Abstract

Feature selection is the one of important issues in the field of machine learning and pattern recognition. It is the technique to find a subset from the source data and can give the best classification performance. Ie, it is the technique to extract the subset closely related to the purpose of the classification. In this paper, we experimented to select the best feature subset for improving classification accuracy when classify success and failure factors in software reuse. And we compared with existing studies. As a result, we found that a feature subset was selected in this study showed the better classification accuracy.

한글키워드 : 특징 선택, 주요 속성, 분류, 클러스터링, 데이터 마이닝, 소프트웨어 재사용

1. 서 론

특징 선택이란 속성의 전체집합 중 분류에 결정적 영향을 미치는 속성의 부분 집합을 찾는 기법으로, CfsSubsetEval가 여러 논문들[1][2][3][4]에서 그 성능이 증명되었다. [3]의 연구에서는 기계 학습 문제를 위한 벤치마킹 데이터 세트에 다양한 속성 선택 기법들을 이용하여 분류 실험을

*, ** 강릉원주대학교 컴퓨터공학과
** 교신저자(email: ktkwon@gwnu.ac.kr)
접수일자: 2012.10.11 수정완료: 2012.12.21

하였는데, 그 중 CfsSubsetEval가 가장 우수한 결과를 보임을 증명하였고, [4]의 연구에서도 CfsSubsetEval 기법을 이용하여 독립적인 변수이면서 쌍 간의 상관관계가 가능한 한 낮은 속성의 집합을 선택하여 분류 실험을 하여 그 성능의 우수성을 검증하였다. 즉, CfsSubsetEval 기법은 각 속성들의 예측 능력과 그들 사이의 중복 정도를 평가하여 가장 출력값과 연관성이 높은 집합을 선정하는 것이다[5]. 본 논문도 속성 평가 기법으로 CfsSubsetEval를 이용하였고 검색 방법으로는 GreedyStepwise를 사용하여 분류 실험을 한 후 그 성능을 분석해 보았다.

논문의 구성은 다음과 같다. 2장에서 배경지식을, 3장에서 데이터마이닝 기법을 이용한 실험과 결과를, 4장에서 결과 비교를, 그리고 5장에서 결론 및 향후연구로 끝맺는다.

2. 배경지식

■ SVM : SVM은 1995년 러시아 통계학자인 Vladimir Vapnik에 의해 제안된 커널 기반의 지도 학습 알고리즘으로[6], 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. 선형 SVM은 두 집합 사이의 분리간격(Margin)을 최대로 하는 초평면(hyperplane)을 찾는 분류기로서 최대 마진 분류기(maximal margin classifier)라 불린다. 그러나 실제 입력 데이터를 적용할 경우 분리 불가능한 데이터가 존재하게 되는데, 이러한 오분류 데이터를 제거하기 위한 방법으로 슬랙 변수 ξ와 패널티 값 C를 사용한다. 또 비선형 경계를 갖는 데이터를 SVM으로 분류하기 위해서는 본래의 좌표 공간에 있는 데이터 x를 선형 분류를 가능하게 하는 새로운 차원의 좌표 공간 $\Phi(x)$로 맵핑하여 초평면을 구하여 분류하게 된다[6].

■ RBFNetwork : 통계학의 다변량 분석 및 보간 문제 해결에 이용되었던 RBF(Radial Basis Function)를 Brommhead와 Low가 신경망 모델을 구성하는데 이용함으로써 RBFNetwork가 제안되었다[7]. RBF 신경망은 입력층, 중간층, 출력층의 3계층으로 구성된 전방향 신경망(feedforward neural network)이다. 입력층은 입력 벡터 공간에 해당하고 출력층은 패턴의 부류(class)에 해당한다. 따라서 중간층을 결정하면 전체 신경망의 구조도 결정된다. 입력층과 중간층 사이의 가중치는 중간층이 결정될 때 고정되고, 중간층과 출력층 사이의 가중치는 학습을 통해 구한다[7]. 빠른 학습 시간, 일반화, 단순화의 특징으로 학습 데이터를 분류하는 작업과 비선형 시스템 모델링 등에 적용되고 있다[8].

■ NaiveBayes : NaiveBayes 분류기는 확률에 기반한 베이지언 알고리즘(Bayesian algorithm) 중의 하나이다. 속성 집합과 클래스 변수 사이의 확률적 관계를 모델링하는 접근 방법으로 클래스의 사전 지식과 데이터로부터 획득한 새로운 증거를 결합시키는 통계 원리인 베이스 정리를 기반으로 한다[9].

X와 Y를 한 쌍의 확률 변수라 하면, 이들의 결합 확률 P(X=x, Y=y)는 X가 x의 값을 갖고 Y가 y의 값을 갖는 확률을 말한다. 그리고 조건부 확률은 다른 확률 변수의 값이 이미 알려진 경우, 한 확률 변수가 특정 값을 가질 확률을 말한다. 예를 들어 조건부 확률 P(Y=y|X=x)는 X의 값이 x로 주어졌을 때, Y가 y의 값을 취할 확률을 뜻한다. X와 Y에 대한 결합 확률과 조건부 확률은 아래 (식 1)로 표현된다.

$$P(X,Y) = P(Y|X) \times P(X) = P(X|Y) \times P(Y) \quad \text{(식 1)}$$

위 식을 재배열하면 베이스 정리(식 2)가 된다.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \qquad (식\ 2)$$

조건부 확률은 Y의 사전확률인 P(Y)와 상반되게 Y의 사후확률로 불린다. 베이스 정리는 사후확률 P(Y), 클래스 조건부 확률 P(X|Y), 증거 P(X)의 식으로 표현된다.

NaiveBayes 분류기의 동작원리는 다음과 같다[9]. 조건부 독립성을 가정할 때, X의 모든 조합에 대하여 클래스 조건부 확률을 계산하는 대신에 Y가 주어졌을 때 각 X_i의 조건부 확률을 계산하면 된다. 시험 항목을 분류하기 위해 NaiveBayes 분류기는 각 클래스 Y에 대한 사후확률을 계산한다.

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(X)} \qquad (식\ 3)$$

P(X)가 모든 Y값에 고정되어 있기 때문에 분자를 최대화하는 클래스를 선택하면 된다.

■ BayesNet : 클래스 조건부 확률 P(X|Y)를 계산하기 위하여 베이지안 분류 방법의 또 다른 구현 방법인 BayesNet가 있다. NaiveBayes 분류기가 사용하는 조건부 독립 가정은 속성들 사이에 약간의 관련성이 존재할 때는 매우 엄격하다. 그에 비해 BayesNet는 클래스 조건부 확률 P(X|Y)를 모델링하는 좀 더 융통성 있는 접근 방법이다. 클래스가 주어졌을 때 모든 속성들의 조건부 독립성을 요구하는 대신, 특정 쌍의 속성들이 조건부적으로 독립적임을 명시할 수 있게 해준다[9].

3. 실 험

3.1 데이터 세트

Morisio et al.[10]은 1994년부터 1997년까지 19개 회사로부터 총 24개 프로젝트에 대해 관리자들을 대상으로 구조화된 인터뷰를 실시하였는데, 소프트웨어 재사용에 있어 성공과 실패에 영향을 준 핵심 요소에 대한 경험적 증거를 유도해 내는 것이 목적이다. 이 중 어떤 속성 조합이 성공과 실패에 영향을 주었는지 분석하기 위해 CART 알고리즘을 이용하여 분류 실험을 한 결과 5개의 핵심 속성을 찾았다. 또한, 다른 연구자들이 그들의 결론을 반복 검증할 수 있도록 PROMISE Repository에 데이터 셋을 공개하였고 Menzies et al.[11]의 연구에 다시 사용되었으며 실험을 위해 J4.8 알고리즘을 이용하여 분류한 결과 8개의 핵심 속성을 찾았다. 본 논문은 [10],[11]의 연구결과를 검증해 보고자 다양한 데이터마이닝 알고리즘을 이용하여 분류 실험을 하였다.

3.2 실험방법

자바 기반의 오픈소스 툴킷인 WEKA를 사용하였고, 분류기들의 테스트 옵션으로는 k-분할 교차검증을 사용하였는데, 이 방법은 데이터 셋을 k개의 세트로 나눈 후 k-1개의 데이터 세트로 학습시키고 나머지 하나로 모델을 검증하는 방법이다. 이 과정을 k개의 세트 각각을 대상으로 순환한다. 일반적으로 예측 모델의 에러는 k번 실행한 테스트 데이터들에서 발생한 에러의 평균값이 된다. 알고리즘이나 세팅을 달리함에 따라 다양한 예측 모델이 나오게 되는데, k번의 테스트 후 평균 에러가 가장 작은 모델을 선택한다[12].

3.3 실험결과

<Table 1>은 26개 속성 전체 집합을 이용하여 분류한 것과 CfsSubsetEval가 찾아준 9개 속성 부분 집합으로 분류한 것의 비교이다.

<Table 1> Experiment I

Classifier	Test option (CV)	속성 선택 안함	CfsSubsetEval 속성선택		
			GreedyStepwise 8,9,11,13,14,16,17,19,24	BestFirst 3,8,9,11,13,14,16,17,18,19,24	LinearForwardSelection 3,8,9,11,13,14,16,17,19,24
SVM	**3**	**83.33%**	**100%**	**100%**	**100%**
	5	**79.17%**	**100%**	**100%**	**100%**
	10	**87.5%**	**100%**	**100%**	**100%**
RBFNetwork	**3**	**91.67%**	**100%**	**95.83%**	**95.83%**
	5	**95.83%**	**100%**	**100%**	**95.83%**
	10	**91.67%**	**100%**	**100%**	**100%**
BayesNet	**3**	**95.83%**	**100%**	**100%**	**100%**
	5	**100%**	**100%**	**100%**	**100%**
	10	**100%**	**100%**	**100%**	**100%**
NaiveBayes	**3**	**95.83%**	**100%**	**100%**	**100%**
	5	**95.83%**	**100%**	**100%**	**100%**
	10	**95.83%**	**100%**	**100%**	**100%**

실험 I의 결과를 검증하기 위해 특징 부분 집합의 속성을 하나씩 제거하면서 해당 속성이 분류 성능에 영향을 주는가를 확인해 보았다. 속성을 제거해도 분류에 영향을 주지 않는 8, 11, 13, 17번 속성을 찾았고 이들을 제거한 5개의 속성으로 분류 실험을 하였다. 그러나 4개 분류기에서 공통적으로 1개의 인스턴스를 오분류하여 분류 성능이 떨어졌고, 분류에 영향을 주는 1개의 인스턴스를 찾기 위해 EM 클러스터링을 이용하였다. 클러스터링 알고리즘은 사용자나 콘텐츠의 군집을 발견하는데 가장 많이 사용되는 알고리즘으로 자동으로 관련 아이템 군집을 찾아주며 그 결과는 분류기나 예측기를 만드는데 사용할 수 있다[13]. (Fig. 1)은 속성의 각 도메인들이 클러스터에 기여하는 영향도 점수이다.

```
                                Cluster
Attribute                         0       1
                                (0.61)  (0.39)
=============================================
Development Approach
  OO                            8.5954  8.4046
  proc                          8.0007  1.9993
  not_available                 1.0513  1.9487
  [total]                      17.6474 12.3526
Non-Reuse Processes Modified
  yes                          14.9117  3.0883
  no                             1.724   7.276
  NA                           1.0117  1.9883
  [total]                      17.6474 12.3526
Human Factors
  yes                          15.5157  2.4843
  no                           1.1317  8.8683
  [total]                      16.6474 11.3526
Domain Analysis
  yes                           9.6025  1.3975
  no                            7.0332  8.9668
  NA                           1.0117  1.9883
  [total]                      17.6474 12.3526
Configuration Management
  yes                          14.9117  3.0883
  no                             1.724   7.276
  NA                           1.0117  1.9883
  [total]                      17.6474 12.3526
```

(Fig. 1) EM Clustering run information
(0 : success, 1 : failure)

'success' 클러스터에 속하려면 영향도 점수 합이 60.373이상이어야 하는데 22번 인스턴스의 경우 34.592이고 이는 'failure'에 해당한다. 22번 인스턴스를 노이즈 값으로 보고 실험에서 제외시킨 후 다시 분류 하였다. 즉, 실험 II는 22번 인스턴스를 제외한 23개 인스턴스와 5개의 속성을 이용하여 분류한 것이고 <Table 2>와 같이 실험 II를 검증하는 과정에서 9번 속성 제거 후 분류 성능이 향상됨을 알고 실험 II'에서는 이를 제외한 4개의 속성만으로 분류 실험을 하여 <Table 3>과 같이 주

<Table 2> Experiment II verification process

attribute	Classifier											
	SVM			RBFNetwork			BayesNet			NaiveBayes		
	Cross-validation			Cross-validation			Cross-validation			Cross-validation		
	3	5	10	3	5	10	3	5	10	3	5	10
9,14,16,19,24	95.65%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5개 속성 - 09 제거	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5개 속성 - 14 제거	95.65%	95.65%	95.65%	100%	100%	95.65%	100%	100%	100%	100%	100%	100%
5개 속성 - 16 제거	95.65%	91.30%	82.61%	95.65%	100%	100%	95.65%	95.65%	95.65%	95.65%	95.65%	95.65%
5개 속성 - 19 제거	95.65%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5개 속성 - 24 제거	95.65%	95.65%	95.65%	100%	100%	95.65%	100%	100%	100%	100%	100%	100%

요 속성 조합을 찾을 수 있었고 <Table 4>에 보이는 것처럼 분류기 모두 100%의 분류 정확도를 보임을 확인하였다.

<Table 3> Key attributes

14	Non-Reuse Processes Modified
16	Human Factors
19	Domain Analysis
24	Configuration Management

<Table 4> Experiment I, II, II′

Classifier	Test option (CV)	속성 선택 안함	속성선택함				
			CfsSubsetEval 8,9,11,13,14,16,17,19,24 (I)	실험 I에 대한 검증 후 9,14,16,19,24		22번 인스턴스 제거	
						9,14,16,19,24 (II)	14,16,19,24 (II′)
SVM	3	83.33%	100%	95.83%	14 10 9	95.65%	**100%**
	5	79.17%	100%	91.67%	14 11 8	**100%**	**100%**
	10	87.5%	100%	91.67%	14 11 8	**100%**	**100%**
RBF Network	3	91.67%	100%	95.83%	14 10 9	**100%**	**100%**
	5	95.83%	100%	95.83%	14 10 9	**100%**	**100%**
	10	91.67%	100%	95.83%	14 10 9	**100%**	**100%**
BayesNet	3	95.83%	100%	95.83%	14 10 9	**100%**	**100%**
	5	100%	100%	95.83%	14 10 9	**100%**	**100%**
	10	100%	100%	95.83%	14 10 9	**100%**	**100%**
NaiveBayes	3	95.83%	100%	95.83%	14 10 9	**100%**	**100%**
	5	95.83%	100%	95.83%	14 10 9	**100%**	**100%**
	10	95.83%	100%	95.83%	14 10 9	**100%**	**100%**

4. 결과비교

<Table 5>에 세 논문의 비교결과가 있다. 'n/a'(not analyzed) 표시는 해당 속성을 실험에서

<Table 5> Comparison of the results

attribute	Morisio et al.	Menzies et al.	본 논문		
			I	II	II′
Application Domain	n/a	×	×	×	×
Size of Baseline	n/a	√	√	×	×
Production Type	√	×	×	×	×
Top Management Commitment	√	√	√	×	×
Reuse Approach	×	√	√	×	×
Domain Analysis	×	√	√	√	√
SP maturity	×	×	×	×	×
Software Staff	×	×	×	×	×
Overall Staff	×	×	×	×	×
Staff Experience	×	×	×	×	×
Type of Software	×	×	×	×	×
Development Approach	×	×	√	√	×
Software and Product	×	×	×	×	×
Origin	×	×	×	×	×
# assests	×	×	×	×	×
Qualification	×	×	×	×	×
Rewards Policy	×	×	×	×	×
Work Products	×	×	×	×	×
Independent Team	×	×	×	×	×
When Assests Built	×	×	×	×	×
Configuration Management	×	×	√	√	√
Key Reuse Roles Introduced	×	√	×	×	×
Repository	√	√	√	√	√
Human Factors	√	√	√	√	√
Reuse Processes Introduced	√	√	√	×	×
Non-Reuse Processes Modified	√	√	√	√	√

제외했다는 것인데, Morisio et al.[10]의 연구에서는 해당 속성의 도메인이 너무 많거나 너무 적다는 이유로 'Application domain', 'Size of baseline' 속성을 제외하였으나 Menzies et al.[11]의 연구와 본 논문에서는 제외시킬 이유가 없어 전체 속성을 대상으로 실험하였다. 또 해당 속성이 분류에 영향을 미치지 않았다는 의미로 '×' 표시를, 의미 있는 영향을 주었다는 의미로, '√' 표시를 사용하였다. 'Top Management Commitment', 'Repository', 'Human Factors', 'Reuse Processes Introduced', 'Non-Reuse Processes Modified' 가 세 논문에서 공통적으로 주요 속성으로 선택되었

다. 특히 'Repository' 속성의 경우 전체 속성을 이용하여 분류한 결과와 이 속성 하나를 제거한 후의 분류 결과가 같다. 즉 분류에 전혀 영향을 미치지 않는 속성 중 하나에 해당되지만 'Repository' 항목에 대한 인터뷰 결과를 보면 총 24개 프로젝트 중 23개가 'yes' 이므로 이것을 경험상 주요 속성 중 하나로 인정하여 세 논문 모두에서 '√' 표시를 했다.

기존 논문들에서는 선택되지 않았으나 실험 I, II, II'에서 선택한 것으로 'Configuration Management' 속성이 있다. 이는 '형상관리'로 소프트웨어 소스 코드, 개발 환경, 빌드 구조 등 전반적인 환경 관리 체계를 정의하고 있다. 그리고 하나의 소프트웨어 산출물을 생성하기 위해 필요로 하는 아이템들과 공정 방식의 정의, 그리고 재생성을 위한 전반적인 환경까지 베이스라인화하여 관리하는 방식 전체를 의미함으로[14] 프로젝트 재사용 가능성에 중요한 정보를 줄 수 있고 또 주요 속성임이 본 논문의 실험 결과 검증되었다.

5. 결 론

본 논문은 소프트웨어 재사용의 성공과 실패를 결정하는 주요 요인을 찾고자 했던 기존 연구들과 같은 목적을 갖고 그 결과들을 검증하고 분류 성능을 좀 더 향상시키고자 데이터 마이닝의 속성 선택 기법 중 CfsSubsetEval를 이용하여 실험하였다. 기존 연구들과 결과를 비교했을 때 가장 적은 수로 이루어진 조합을 찾을 수 있었고 이 속성 조합을 이용하여 분류했을 때 더 우수한 분류정확도를 보임을 확인하였다.

향후 연구는 빅 데이터에 다양한 클러스터링 기법과 속성 선택기를 이용하여 노이즈나 이상치를 찾아 이러한 요소들이 데이터 마이닝의 실험 결과에 미치는 영향들을 분석하는 것이다.

참 고 문 헌

[1] MA Hall, "Correlation-based feature selection for machine learning", lri.fr, 1999

[2] Lei Yu, Huan Liu, "Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution", Proceedings of the Twentieth International Conference on Machine Learning(ICML-2003), Washington DC, 2003

[3] K. Selvakuberan, M. Indradevi, Dr. R. Rajaram "Combined Feature Selection and classification - A novel approach for the categorization of web pages", Journal of Information and Computing Science, Volume 3, Issue 2, pp. 083-089, 2008

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

[5] K Michalak, H Kwasnicka, "Correlation-based feature selection strategy in classification problems", Int. J. Appl. Math. Comput. Sci., Volume 16, Issue 4, pp. 503 - 511, 2006

[6] PN Tan, M Steinbach, V Kumar, "Introduction to data mining", Addison-Wesley, 2006

[7] Masahide Watanabe, Kaihei Kuwata, Ryu Katayama., "Adaptive Tree-Structured Self Generating Radial Basis Function and its Application to Nonlinear Identification Problem", PROCEEDINGS of The 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, pp. 167-170, 1994

[8] YoungSup Hwang, Sung-Yang Bang, "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier", Journal of KIISE : Software and Applications, Volume 24, Issue 5,

pp. 451-460, 1997
[9] Satnam Alag, "Collective Intelligence in Action", Manning Publications Co., 2009
[10] Morisio, M., Ezran, M., Tully, C., "Success and failure factors in software reuse", IEEE Transactions on Software Engineering, volume 28, Issue 4, pp. 340-357, 2002
[11] Menzies, T., Di Stefano, J.S., "More success and failure factors in software reuse", IEEE Transactions on Software Engineering, volume 29, Issue 5, pp. 474-477, 2003
[12] IH Witten, E Frank, "Data Mining : Practical Machine Learning Tools and Techniques", Second Edi., Morgan Kaufmann, 2005
[13] Richard J. Roiger, Michael W. Geatz, "Data mining a tutorial-based primer", Addison Wesley, 2003
[14] "CMMI for Development, Version 1.2", Carnegie Mellon University, 2006

저 자 소 개

김 영 옥

1997년 강릉원주대학교
컴퓨터공학과 학사 졸업
2003년 강릉원주대학교
컴퓨터공학과 교육학석사 졸업
2012년~현재 강릉원주대학교
컴퓨터공학과 박사과정 수료

<관심분야: 소프트웨어 신뢰도, 데이터 마이닝 등>

권 기 태

1986년 서울대학교
계산통계학과 학사 졸업
1988년 서울대학교
계산통계학과 이학석사 졸업
1993년 서울대학교
계산통계학과 이학박사 졸업
1996년 Univ. of Southern California, Post-Doc.

<관심분야 : 소프트웨어 비용산정, 소프트웨어 메트릭스, 소프트웨어 아키텍처, 데이터 마이닝 등>