Journal of the Korean Data & Information Science Society
2015, **26**(3), 639–652

http://dx.doi.org/10.7465/jkdi.2015.26.3.639
한국데이터정보과학회지

의사결정나무를 이용한 다변량 공정관리 절차[†]

정광영[1] · 이재헌[2]

[12]중앙대학교 응용통계학과

접수 2015년 3월 16일, 수정 2015년 3월 30일, 게재확정 2015년 4월 27일

요 약

현대의 제조공정은 컴퓨터의 발전과 통신 및 네트워크의 발달로 컴퓨터통합제조가 가능해졌다. 이로 인해 고품질 제품의 고속 생산공정이 확대되고, 공정에서 실시간으로 전송되는 다양한 품질변수들의 데이터 축적 또한 가능하게 되었다. 이를 관리하기 위해서는 다변량 통계적 공정관리 절차가 필요하다. 전통적으로 사용하는 다변량 관리도는 이상상태 발생시 이상신호를 주지만, 이상원인이 어떠한 변수에 어떠한 영향을 주는지에 대한 정보를 제공하지 않는다는 단점이 있다. 이를 보완하기 위해 데이터마이닝과 기계학습 기법을 이용할 수 있다. 이 논문에서는 의사결정나무 학습 기법을 이용한 다변량 공정관리 절차를 소개하고, 이변량인 경우 모의실험을 통하여 그 효율을 살펴보았다. 모의실험 결과를 살펴볼 때, 상관계수에 따라 이상상태 탐지 능력은 비슷한 것으로 나타났고, 이상상태에 대한 분류 정확도는 상관계수와 이상원인의 형태에 따라 차이가 있지만 기존의 다변량 관리도에서는 제공하지 않는 이상원인의 정보를 제공하는 장점이 있음을 알 수 있다.

주요용어: 다변량 공정관리, 데이터마이닝, 의사결정나무 학습, 컴퓨터통합제조.

1. 서론

현재 우리는 정보화 사회를 표방하는 시대에 살고 있다. 정부와 기업들은 앞다투어 질 좋은 정보를 독점하고 효율적으로 이용하기 위해서 수없이 경쟁하며 정보의 중요성을 강조하고 있다. 또한 하드웨어와 소프트웨어, 그리고 통신 및 네트워크의 비약적인 발전은 데이터를 기하급수적으로 발생시키는 계기가 되었고, 동시에 이를 효율적으로 수집하고 저장할 수 있는 압축 기술과 데이터베이스 시스템 또한 발전하게 되었다. 이러한 현상은 정보화 사회를 넘어서 정보통신 사회를 이룩하게 되는 기반이 되었다.

정보통신 사회에서는 다양하면서도 방대한 데이터가 모여지고 있다. 여기서 데이터는 일반적으로 가공되지 않은 형태의 자료를 말한다. 즉, 우리는 다듬어지지 않은 정보들 속에서 잠재되어 있는 가치를 찾기 위한 통계적인 기법들이 필요로 하게 되었다. 이 중에서 최근에 가장 대두되고 있는 것이 데이터마이닝 기법이다. 데이터마이닝은 다방면에서 얻은 광대한 자료에 존재하는 패턴, 규칙, 관계 등을 찾아내어 분석함으로써, 우리가 원하는 지식을 추출하고 미래에 대한 예측을 위해 사용된다.

컴퓨터공학의 발전으로 여러 분야에서 데이터를 쉽고 빠르게 저장할 수 있는 환경이 만들어져, 이를 잘 활용할 수 있는 데이터마이닝 기법들이 다방면에서 사용되고 지속적으로 연구되고 있다. 예를 들어,

[†] 이 논문은 2014년도 정부 (교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2014R1A1A2054200).
[1] (156-756) 서울시 동작구 흑석로 84, 중앙대학교 응용통계학과, 석사과정.
[2] 교신저자: (156-756) 서울시 동작구 흑석로 84, 중앙대학교 응용통계학과, 교수.
E-mail: jaeheon@cau.ac.kr

대형마트에서 소비자의 패턴을 분석하여 기저귀와 맥주를 서로 가깝게 진열시켜 매출이 크게 늘어난 사례와 유전자의 정보에 활용하여 생물학과 의약 등 여러 분야에서 사용되는 것은 널리 알려져 있다.

컴퓨터공학의 발전은 제조공정의 환경에도 큰 영향을 주었다. 좋은 제품을 생산하기 위해서는 공정설계와 효율적인 공정관리 (process control)가 필요하다. 목표하고 있는 품질수준을 유지하고 통제하기 위해서는 공정의 변동을 효율적으로 탐지하고 감시할 수 있는 적절한 통계적 공정관리 (statistical process control; SPC)가 필수적이다. 통계적 공정관리에서 기본적으로 이용하는 관리도는 Shewhart 관리도, 누적합 (cumulative sum; CUSUM) 관리도, 지수가중이동평균 (exponentially weighted moving average; EWMA) 관리도 등이 있다.

그러나 위에서 언급한 일변량 관리도는 2개 이상의 속성에 의해 제품의 품질이 결정되는 다변량 공정의 관리에는 적용하기가 어렵다. 그러나 자동 데이터 습득 시스템의 발전과 온라인 모니터 프로그램의 적용으로 서로 상관이 있는 품질변수 (quality variable)들을 동시에 모니터링할 수 있는 다변량 공정관리 (multivariate SPC; MSPC)를 구현할 수 있게 되었다. 다변량 공정관리에서 전통적으로 많이 사용되고 있는 관리도는 Hotelling의 T^2 관리도, 다변량 누적합 (multivariate CUSUM; MCUSUM) 관리도, 다변량 지수가중이동평균 (multivariate EWMA; MEWMA) 관리도 등이 있다. 전통적으로 사용하는 다변량 관리도는 공정의 이상상태 (out-of-control state) 여부를 판단할 수 있으나, 이상상태의 원인이 되는 품질변수를 판별하고 이상원인 (assignable cause)이 품질변수를 어떻게 변화시켰는지에 대한 정보를 제공하지 못한다는 단점이 있다. 즉, 이상원인을 제거하고 수정까지 발생하는 시간과 비용의 손실이 많을 수 밖에 없다. 다변량 관리도에 대한 연구는 Cho (2010)와 Cho와 Park (2013) 등을 참고할 수 있다.

컴퓨터 공학의 발전으로 제조환경은 컴퓨터통합제조 (computer integrated manufacturing; CIM) 시스템을 도입하여 전체적인 제조 상황을 관리하는 방향으로 발전하게 되었고, 이는 통계적 공정관리의 자동화 구현을 가능하게 하였다. 최근 많이 사용되고 있는 기계학습 (machine learning) 기법을 공정관리에 적용할 수 있으며, 과거에 알려진 속성들을 가지고 자동적이나 준자동적으로 현재의 데이터들을 분류하여 공정의 이상 여부를 판단할 수 있다. 기계학습 기법 중 신경망 (neural network) 이론을 공정관리에 적용한 연구는 Guh와 Tannock (1999), Ho와 Chang (1999), Chen과 Wang (2004), Hwarng (2005), 그리고 Guh (2007) 등이 있고, 의사결정나무 (decision tree; DT)를 공정관리에 적용한 연구는 Guh (2005), Guh와 Shiue (2005, 2008) 등이 있다.

이 논문에서는 의사결정나무를 이용한 다변량 공정관리 절차를 소개하고, 그 효율에 대해 알아보고자 한다. 공정관리 절차의 효율을 판단하는 측도로는 분류의 정확도와 평균런길이 (average run length; ARL)를 사용했는데, 여기서 평균런길이는 관리도에서 이상신호를 줄 때까지 관측한 평균 표본의 수를 나타낸다. 이 논문은 Guh와 Shiue (2008)에서 가정한 모형 및 절차와 유사한 점이 있지만, 이상상태의 학습 데이터를 구성하는 방법과 좀 더 다양한 경우에 모의실험을 수행하여 정분류 및 오분류에 대한 결과를 제시했다는 점에서 차이가 있다.

의사결정나무 기반의 학습기술은 간단히 실행가능하며 분류나 예측에 대해 이해하고 설명하기 쉬운 장점이 있어서 실무자나 관리자에게 훌륭한 알고리즘이 될 수 있다. 또한 식별의 문제가 있던 기존의 다변량 관리도와는 다르게, 문제가 되는 품질변수를 식별하고 그 현상을 설명할 수 있는 정보를 제공하여 준다. 따라서 공정의 상태가 원하는 수준에서 벗어났을 경우 보다 빠르게 그 원인을 찾아 제거할 수 있기 때문에, 고품질의 고속 공정에서 시간과 비용을 절감할 수 있는 효과적인 다변량 공정관리의 절차라고 할 수 있다.

2절에서는 이 논문에서 제시한 다변량 공정관리 절차의 주된 방법인 의사결정나무 학습 알고리즘을 설명하고, 3절에서는 의사결정나무 학습 기반의 이변량 공정관리 절차를 제시하고 모의실험을 통하여 그 절차의 효율성을 살펴본다. 마지막으로 4절에서 결론을 제안하고 있다.

2. 의사결정나무를 이용한 다변량 공정관리 절차

2.1. 공정관리 절차의 구조

컴퓨터통합제조 시스템을 통하여 제품의 생산과 동시에 품질변수들의 데이터를 빠르게 수집할 수 있게 되었고, 따라서 고속자동생산 속에서 불량품 생산과 비용 손실을 막기 위해서 공정의 변화를 일찍 탐지하고 이를 수정하는 것이 필요하다. 다변량 공정관리에서 의사결정나무 학습 모델은 공정을 빠르고 자동적으로 모니터링하기 위해 적용되었다. 이 모델은 Figure 2.1과 같이 모의실험을 통해서 의사결정나무 규칙을 생성하고, 그 규칙에 근거하여 실제 품질변수들을 모니터링하여 이상상태 여부를 판단하는 것이다. 이상상태가 탐지될 경우에는 공정관리 시스템에 의해 관리자에게 이상신호와 진단 결과를 전달한다. 여기서 관리상태 (in-control state)인 공정의 평균벡터 $\boldsymbol{\mu}$와 공분산행렬 $\boldsymbol{\Sigma}$는 정확히 알려졌다고 가정한다.

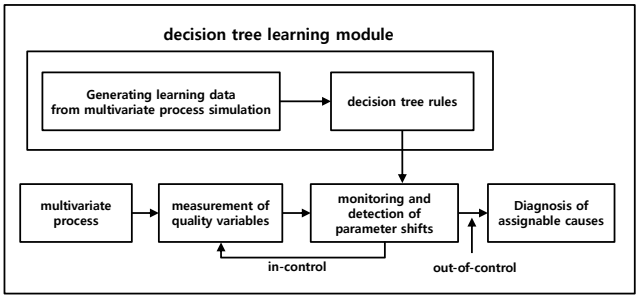

Figure 2.1 Structure of the MSPC procedure using a decision tree

2.2. 의사결정나무 학습 모형

본 연구에 사용한 의사결정나무 학습 모형은 이동식 윈도우 (moving window) 방법을 이용한다. 이는 공정에서 생산되는 제품이 순차적으로 관측될 때, 이동식 식별 윈도우 (moving identification window)도 앞으로 하나씩 움직이는 것을 말한다. Figure 2.2에서 점들은 제품이 표본으로 선택되어 관측되는 품질변수들의 측정값들을 나타낸다.

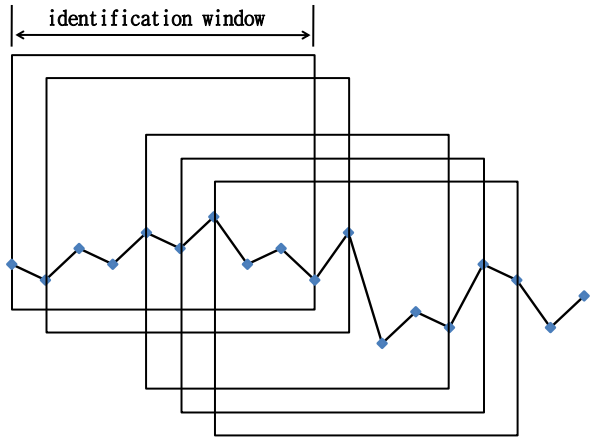

Figure 2.2 Moving identification windows along a sequence of product observations

식별 윈도우는 의사결정나무 학습에 의해 생성된 규칙들을 이용하여 공정의 상태를 파악하게 하는 일종의 단위라고 생각하면 된다. 즉, 현재의 측정값을 포함하여 일정기간 축적된 값들로 공정의 상태를 판단하는 것이다. 품질변수의 수와 윈도우 크기에 따라 공정상태의 판단 근거가 되는 입력 데이터의 크기는 달라진다. 품질변수의 수가 p이고 윈도우 크기가 w인 경우 입력 데이터는 $[(X_{1,1}, X_{1,2}, \cdots, X_{1,p}, \delta_1),\ (X_{2,1}, X_{2,2}, \cdots, X_{2,p}, \delta_2),\ \cdots,\ (X_{w,1}, X_{w,2}, \cdots, X_{w,p}, \delta_w)]$로 구성된 집합으로 표현할 수 있다. 즉, 입력 데이터 벡터는 $(p+1) \times w$ 개의 원소로 구성되어 있다. 여기서 X는 품질변수의 값이고 δ는 p개의 품질변수 값 사이의 마할라노비스 거리 (Mahalanobis distance)를 나타낸다. 예를 들어, 품질변수의 수가 3개이고 윈도우 크기가 10이면 $[(X_{1,1}, X_{1,2}, X_{1,3}, \delta_1),\ (X_{2,1}, X_{2,2}, X_{2,3}, \delta_2),\ \cdots,\ (X_{10,1}, X_{10,2}, X_{10,3}, \delta_{10})]$으로 구성된다. 각 변수들의 모수의 변화는 변화 없음 (no shift), 위로 변화 (upward shift), 아래로 변화 (downward shift)의 세가지를 고려하기 때문에, 품질변수의 수에 따라 의사결정나무 모델에 의해 분류되는 종류는 3^p개가 된다.

2.2.1. 공정 모형과 데이터 구조

의사결정나무 기반의 다변량 공정관리 절차에서 공정 평균의 이상상태를 탐지하기 위해서는 평균 변화에 대한 상황들을 사전에 학습시킬 필요가 있다. 품질변수들의 평균은 관리상태에서 서서히 변화하는 것이 아니라 어느 시점에서 갑자기 변화하는 것으로 가정한다. 이 논문에서 평균 변화에 대한 데이터 생성 모형은 다음과 같이 Guh와 Shiue (2008)에서 사용한 모형을 동일하게 사용하였다.

공정 모형은

$$\boldsymbol{X}_t = \boldsymbol{\mu} + \boldsymbol{n}_t + \boldsymbol{M}_t \tag{2.1}$$

로 표현한다. 여기서 t는 표본추출 시점이고, $\mathbf{X}_t$는 t시점에 추출된 품질변수들의 값을 나타내는 다변량 벡터이다. 만일 공정이 관리상태일 경우 $\mathbf{X}_t$는 다변량 정규분포 $\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$를 따른다고 가정한다. $\boldsymbol{n}_t$는 t시점에 발생하는 우연원인 (chance cause)으로 $\boldsymbol{N}(\mathbf{0}, \boldsymbol{\Sigma})$를 따른다고 가정한다. 우연원인은 공정이 관리상태임에도 발생하는 변동을 나타낸다. 마지막으로 $\boldsymbol{M}_t$는 t시점에 발생하는 평균 변화로 $\boldsymbol{M}_t = u_t \times \boldsymbol{s}$로 표현할 수 있으며, u_t는 평균 변화의 위치를 결정하는 변수이다. 즉, T가 공정의 변화를 나타내는 시점일 경우, $t < T$이면 $u_t = 0$이고, $t \geq T$이면 $u_t = 1$이 된다. 또한 $\boldsymbol{s}$는 변동의 크기를 나타내는 벡터로서 $(k_1\sigma_1, k_2\sigma_2, \cdots, k_p\sigma_p)$로 표현할 수 있다.

다변량 공정의 모의실험을 통하여 생성된 데이터는 표준화 (standardization) 단계와 표준화된 데이터가 속하는 구간 내의 대표값으로 코딩 (coding) 처리하는 2단계 과정을 거쳐 변환된다. $X_{i,j}$ $(i = 1, 2, \cdots, w;\ j = 1, 2, \cdots, p)$를 표준화한 변수를 $Y_{i,j}$라 할 때, $Y_{i,j}$는 Table 2.1과 같이 코딩 처리할 수 있다. 즉, 구간 $[-7.375, +7.375]$를 0.25의 폭으로 나눈 59개의 구역 (zone)을 -29에서 29까지 값으로 코딩하고 그 외 구역을 -30과 30으로 코딩하였다. 코딩된 데이터를 의사결정나무 학습 과정에 직접적으로 사용하며, 이 코딩 작업은 미세한 변동들의 영향을 제거해 주고 의사결정나무에 의한 규칙을 보다 단순하게 만들어 주는 역할을 한다.

Table 2.1 Coding scheme for the standadized data

standadized data	coding value
$Y_{i,j} < -7.375$	-30
$-7.375 \leq Y_{i,j} < -7.125$	-29
$\vdots$	$\vdots$
$-0.375 \leq Y_{i,j} < -0.125$	-1
$-0.125 \leq Y_{i,j} < 0.125$	0
$0.125 \leq Y_{i,j} < 0.375$	1
$\vdots$	$\vdots$
$7.125 \leq Y_{i,j} < 7.375$	29
$Y_{i,j} \geq 7.375$	30

2.2.2. 윈도우 크기 설정

이동식 윈도우 방법을 사용하기 때문에, 윈도우 크기를 설정하는 것은 매우 중요하다. 윈도우 크기가 작을 경우에는 이상상태에 대한 탐지는 빠르나 관리상태의 평균런길이가 짧아져 제1종 오류 (type I error)가 커지고, 반대로 윈도우 크기가 클 경우에는 제2종 오류 (type II error)가 커지는 현상이 있기 때문이다. 이를 살펴보기 위해 윈도우 크기 w가 4, 6, 10, 16, 28인 경우 평균런길이를 Table 2.2에서 비교하였다. 여기서 품질변수의 수는 2개, 관리상태에서 각 변수의 평균은 0, 분산은 1, 상관계수는 $\rho = 0.5$로 설정하였고, 평균런길이는 1000번 반복해서 구한 런길이의 평균값이다. 이상상태에서의 런길이는 3절에서 수행한 여러 가지의 평균변화에 대한 평균런길이의 평균값을 나타낸다. 공정관리 절차에서 평균런길이를 계산하는 과정은 2.3절과 2.4절에서 좀 더 상세하게 설명할 것이다.

이 논문에서 평균런길이를 계산하는 모의실험은 공정의 변화시점 (process change point)으로 $T = 51$을 가정하였다. 즉, 공정은 50번째 시점까지는 관리상태이고 51번째부터 이상상태로 변화하는 것을 가정하였는데, 많은 연구를 통해 이 T값은 아주 작지만 않으면 결과에 큰 영향을 미치지 않는 것으로 알려져 있다.

Table 2.2를 보면 위에서 언급한 바와 같이 윈도우 크기가 커질수록 관리상태의 평균런길이는 커지고, 이상상태에 대한 탐지 능력이 떨어지는 것을 알 수 있다. 제1종 오류 및 제2종 오류 (탐지 능력)를 감안하여 이후 윈도우 크기는 $w = 10$으로 고정하였다.

Table 2.2 ARL values with different window sizes (w)

w	in-control	out-of-control
4	35	2.95
6	78	3.69
10	300	5.15
16	366	6.49
28	783	9.52

2.3. 의사결정나무 학습 과정

2.3.1. 학습 데이터 생성 및 분류 구현

앞에서 언급한 바와 같이 품질변수의 수는 $p = 2$이고, 관리상태의 평균런길이를 고려하여 윈도우 크기는 $w = 10$으로 설정하였다. 따라서 하나의 학습 데이터는 표준화 단계와 Table 2.1의 코딩 과정을 거쳐 $[(Y_{1,1}, Y_{1,2}, \delta_1), (Y_{2,1}, Y_{2,2}, \delta_2), \cdots, (Y_{10,1}, Y_{10,2}, \delta_{10})]$, 즉 30개의 원소를 갖는 벡터로 구성되어 있다.

학습 데이터의 생성을 위해 먼저 변동의 크기 $\mathbf{s}$를 설정하였다. 다변량 데이터는 평균이 0이고, 분산이 1로 표준화되었기 때문에 변동의 크기를 나타내는 벡터 $\mathbf{s}$는 $(k_1, k_2, \cdots, k_p)$가 된다. 이 논문에서는 변동의 크기가 -3, -2, 0, 2, 3인 경우를 고려하였다. 크기가 -1과 1인 경우에는 탐지하기가 쉽지 않아 평균런길이가 크고, 의사결정나무 규칙이 너무 많이 생성되기 때문에 학습 데이터에서 제외하였다. 공정평균의 변동 크기를 5가지로 설정함에 따라 품질변수의 수가 $p = 2$인 경우 25 $(= 5^2)$개 유형의 데이터 상태가 가능하다. 즉, 24 $(= 5^2 - 1)$개의 이상상태와 한개의 관리상태의 학습 데이터를 생성할 수 있다. 학습 데이터는 관리상태의 데이터 2000개와 이상상태의 데이터 2400 $(= 24 \times 100)$개로 총 4400개의 데이터를 만들었다. 이를 좀 더 상세하게 설명하면, 2개의 품질변수의 변동 크기를 (k_1, k_2)라고 할 때, 관리상태, 즉 $(0, 0)$인 경우의 데이터 2000개와 $(0, 0)$를 제외한 $(-3, -3), (-3, -2), \cdots, (3, 3)$인 경우의 데이터를 각 경우마다 100개씩 총 2400개를 생성한 것이다.

공정평균의 25개 유형을 Table 2.3과 같이 9가지 유형으로 다시 구분할 수 있기 때문에, 총 4400개의 학습 데이터는 9개의 집단으로 분류된다고 할 수 있다. 따라서 이 데이터에 의사결정나무를 적용할 경

우 집단의 분류 규칙을 생성할 수 있으며, 실제 생산공정에서 이 규칙을 사용하여 공정의 상태를 판단할 수 있게 된다.

Table 2.3 Shift types of the bivariate process control procedure

shift	$Y_{i,1}$	N	U	U	U	N	N	D	D	D
status	$Y_{i,2}$	N	U	N	D	U	D	U	N	D
type code		T0	T1	T2	T3	T4	T5	T6	T7	T8

* N: no shift, U: upward shift, D: downward shift

여기서 이상상태 데이터의 각 벡터들은 관리상태와 이상상태의 값들로 이루어져 있는데, 관리상태와 이상상태를 구분하는 변화시점을 설정해야 된다. 그 이유는 품질변수를 이동식 윈도우 방법을 이용하여 관측하기 때문에, 처음에는 관리상태이고 어느 시점 이후 이상상태로 변화하는 형태의 학습 데이터를 생성하는 것이 더 타당하기 때문이다. 예를 들어 설명하면, 2개의 품질변수의 변동 크기가 $(-3,-3)$인 경우의 학습 데이터는 윈도우 크기의 모든 원소를 변동 크기가 $(-3,-3)$인 경우 생성하는 것이 아니라, 윈도우 크기의 초반에는 관리상태인 경우 생성하고 어느 시점 이후부터 $(-3,-3)$인 경우 생성해서 구성하게 된다.

변화시점의 영향을 살펴보기 위해 윈도우 크기의 1/4, 1/2, 3/4 시점을 고려하였다. 이 논문에서 윈도우 크기는 10으로 설정했기 때문에, 학습 데이터의 각각 3번째, 5번째, 8번째부터가 이상상태에서 데이터를 생성한 것이다. 평균런길이는 각 경우마다 1000번 반복하여 계산한 평균값을 Table 2.4에 제시하였는데, 이때 상관계수는 $\rho = 0.5$를 가정하였다. 다른 상관계수 값에 대해서도 결과를 얻었지만 유사한 경향을 보이기 때문에 $\rho = 0.5$인 경우에 대한 결과만 제시하였다.

Table 2.4에서 변화시점이 3/4인 경우 관리상태에서의 평균런길이는 24로 아주 작은 값을 갖기 때문에, 즉 오경보의 발생 빈도가 너무 많기 때문에 이 값은 적당하지 않다. 변화시점이 1/4과 1/2를 비교해 보면 이상상태의 탐지 능력에서는 1/2인 경우가 좀 더 좋지만 관리상태에서의 평균런길이에서 큰 차이가 나기 때문에, 학습 데이터에서 관리상태와 이상상태를 구분하는 변화시점은 1/4을 사용하는 것이 바람직하다고 판단된다.

Table 2.4 ARL values with difference shift points of the learning data set

shift size		shift point		
k_1	k_2	1/4	1/2	3/4
0	0	300	105	24
-3	-3	4.34	4.23	2.29
	-2	4.64	4.47	2.49
	0	4.89	4.78	2.51
	2	4.07	4.40	2.40
	3	3.99	4.31	2.36
-2	-3	4.82	4.42	2.42
	-2	5.61	4.84	2.77
	0	6.46	5.39	2.89
	2	4.59	4.44	2.42
	3	4.48	4.34	2.31
0	-3	5.19	4.49	2.37
	-2	6.43	5.18	2.87
	2	6.70	4.97	3.00
	3	5.39	4.32	2.49
2	-3	4.59	3.93	2.17
	-2	5.03	4.24	2.40
	0	6.40	5.39	2.92
	2	5.94	4.54	2.83
	3	5.40	4.12	2.57
3	-3	4.43	3.76	2.12
	-2	4.69	3.96	2.30
	0	5.28	4.60	2.56
	2	5.13	4.31	2.53
	3	5.00	3.94	2.51
average		5.15	4.47	2.52

2.3.2. 구현 방법

학습 데이터 생성 및 모의실험은 통계 소프트웨어인 R을 통해 구현하였다. R은 필요한 분석을 직접 프로그래밍을 통해 실행하거나 연구자들이 개발한 패키지를 사용하여 분석할 수 있다. 여기서 다변량 정규분포 데이터를 생성하기 위해서 'mvtnorm' 패키지와 마할라노비스 거리를 계산해 주는 'mahalanobis' 패키지, 그리고 의사결정나무 규칙을 생성하고 생성된 규칙을 통해서 다변량 공정에서 측정된 입력데이터를 판별하는 'C5.0' 패키지를 사용하였다. C5.0 (Quinlan, 1998) 알고리즘은 목표변수에 따라서 의사결정나무 규칙을 만드는데 가지치기를 통해서 이를 조정할 수 있다. 가지치기는 2가지 옵션으로 구성되어 있는데, 하나는 의사결정나무 끝마디에 포함되는 최소한의 사례의 수를 결정하는 m과 다른 하나는 가지치기의 신뢰수준을 결정하는 c이다. 그리고 의사결정나무 학습을 향상시키기 위해서 Adaboost 기술을 사용하였다. 이 논문에서 의사결정나무를 생성할 때 가지치기 옵션은 $m = 1$과 $c = 0.5$를 사용하였다. 또한 앙상블 기법인 Adaboost의 반복수는 3번으로 설정하였다.

2.4. 예제

이 절에서는 위에서 제시한 내용의 이해를 돕기 위해 이 논문에서 사용한 다변량 공정관리 절차, 즉 학습 데이터 생성, 분류 규칙 생성, 그리고 공정관리에 적용에 대하여 단계별로 설명하고자 한다. 품질변수의 수는 $p = 2$이고, 윈도우 크기는 $w = 10$, 그리고 학습 데이터에서 관리상태와 이상상태를 구분하는 변화시점은 윈도우 크기의 1/4을 가정하였다.

단계 1: 학습 데이터 생성

이 논문에서 $\rho = 0.5$인 경우 생성한 총 4400개의 학습 데이터의 일부를 Table 2.5에 제시하였다. Table 2.5의 데이터는 표준화 및 대표값으로 코딩되는 2단계 과정을 거쳤으며, 두 품질변수 평균의 변화량에 따라 Table 2.3과 같이 T0에서 T8의 9개 집단으로 분류된다.

Table 2.5 Learning data set when $p = 2$, $w = 10$, and $\rho = 0.5$

no.	shift size		class	data									
	k_1	k_2		$Y_{1,1}$	$Y_{1,2}$	δ_1	$Y_{2,1}$	$Y_{2,2}$	δ_2	$\cdots$	$Y_{10,1}$	$Y_{10,2}$	δ_{10}
1	0	0	T0	7	6	1.905	3	0	0.782	$\cdots$	-4	-6	1.452
2	0	0	T0	-1	-1	0.383	-8	-4	1.986	$\cdots$	1	5	1.314
$\vdots$													
2000	0	0	T0	-2	-2	0.650	-3	-1	0.647	$\cdots$	-3	-2	0.800
2001	0	-2	T5	3	4	1.048	-3	0	0.740	$\cdots$	-2	-7	1.782
2002	0	-2	T5	3	-2	1.108	2	-2	1.001	$\cdots$	-2	-14	3.676
$\vdots$													
3001	-3	-2	T8	2	0	0.460	-3	5	1.920	$\cdots$	-15	-9	3.742
3002	-3	-2	T8	5	7	1.770	-3	4	1.680	$\cdots$	-9	-5	2.360
$\vdots$													
4400	3	3	T1	4	2	1.091	-3	-1	0.718	$\cdots$	18	5	4.546

단계 2: 의사결정나무를 이용한 분류 규칙 생성

Table 2.5의 학습 데이터에 대해 의사결정나무 기법을 적용할 경우 9개 집단에 대한 분류 규칙을 생성할 수 있는데, 생성된 분류 규칙의 일부를 Table 2.6에 제시하였다. Adaboost의 반복수는 3번으로 설정하여, 처음에 108개의 규칙, 반복 1에서 53개의 규칙, 반복 2에서 55개의 규칙, 그리고 반복 3에서 72의 규칙 등 총 288개의 분류 규칙이 생성되었다. Table 2.6을 살펴보면 각 분류 기준과 분류되는 소속 집단이 나와있고, 집단 옆의 대괄호에 표기된 수치는 라플라스비 (Laplace ratio)로서 분류 규

칙의 정확도를 나타낸다. 학습 데이터에 대해 총 288개의 분류 규칙을 적용할 경우 정분류율은 99.8% (4400개의 학습 데이터 중 7개가 오분류됨)로 나타났다. 이렇게 정분류율이 높게 나타난 것은 학습 데이터를 생성할 때, 평균의 변화량이 아주 작은 경우 (k_1과 k_2가 1 또는 -1인 경우)를 고려하지 않았기 때문인 것으로 판단된다.

단계 3: 다변량 공정관리에 적용

단계 2에서 생성된 분류 규칙을 다변량 공정관리에 적용할 때, 먼저 10개 시점 (시점 1에서 시점 10)에서 각 시점마다 2개의 품질변수를 측정하여 마할라노비스 거리를 계산하고, 표준화 단계 및 코딩 단계를 거쳐 30개의 원소를 갖는 벡터를 구성한다. 이때 윈도우 크기인 $w = 10$ 시점까지는 공정의 상태를 판단할 수 없는 것이 제안된 방법의 단점이라 할 수 있다. 구성된 벡터에 대해 단계 2에서 생성된 규칙을 적용하여 집단을 분류한다. 이때 관리상태인 T0로 분류될 경우 다음 시점으로 넘어가고, T1에서 T8로 분류된 경우 이상상태의 신호를 주는 것이다. 만일 T0로 분류된 경우 식별 윈도우를 이동시켜, 즉 시점 2에서 시점 11의 품질변수값으로 벡터를 구성하여 분류 규칙을 다시 적용하게 된다. 이와 같은 과정을 계속 반복하여 공정의 이상 유무를 판단하는 것이다.

Table 2.6 Classification rules derived from the decision tree when $p = 2$, $w = 10$, and $\rho = 0.5$

trial	no.	rule
0	1	$\delta_5 \le 2.413, Y_{6,1} \le 7, Y_{7,2} \le 6, \delta_7 > 0.153, \delta_7 \le 2.545, Y_{8,2} > -4, \delta_8 \le 2.361, Y_{9,1} > -8, Y_{9,1} \le 8, Y_{9,2} > -9$ → class T0 [0.998]
		⋮
	108	$Y_{4,1} \le -8, Y_{7,1} \le -8, \delta_7 \le 2.545$ → class T8 [0.628]
1	1	$Y_{7,1} > -7, \delta_7 \le 2.754, Y_{8,2} > -6, \delta_8 \le 2.551, Y_{9,2} \le 7, \delta_{10} \le 2.378$ → class T0 [0.985]
		⋮
	53	$Y_{5,2} \le -2, Y_{6,1} \le -6, Y_{8,2} \le -6, Y_{10,1} \le 3, Y_{10,2} \le -2$ → class T8 [0.784]
2	1	$Y_{4,1} > -6, Y_{4,2} \le -4, Y_{6,2} > -2, \delta_8 \le 3.975, Y_{10,1} \le 5$ → class T0 [0.974]
		⋮
	55	$Y_{4,1} \le -6, Y_{4,2} \le -9, Y_{5,1} \le 0, Y_{6,1} \le -1$ → class T8 [0.945]
3	1	$\delta_4 \le 2.363, Y_{5,1} > 1, \delta_5 \le 2.808, Y_{6,1} \le 7, Y_{7,2} \le 3, \delta_7 \le 2.682, \delta_8 \le 3.875, Y_{10,1} \le 8, Y_{10,2} > -12$ → class T0 [0.994]
		⋮
	72	$Y_{5,1} \le 1, Y_{6,2} \le -8, Y_{7,2} \le 3, Y_{8,1} \le -6$ → class T8 [0.859]

3. 다변량 공정관리 절차의 효율

3절에서는 윈도우 크기가 $w = 10$이고 학습 데이터에서 변화시점은 1/4로 고정한 후, 상관계수 ρ가 ± 0.9, ± 0.5, ± 0.1인 경우 의사결정나무를 이용한 다변량 공정관리 절차의 효율에 대해 살펴보고자 한다. 다른 상관계수에 대해서도 효율을 살펴보았지만, 위의 6가지 상관계수에 대한 결과만 제시하였다.

효율은 다음의 두 가지 측도로 살펴보았다. 첫째는 이상상태의 평균런길이이고, 두번째는 이상상태의 분류율이다. 서론에서 언급한 바와 같이 전통적으로 사용하는 다변량 관리도에서는 이상상태의 신호

를 주지만, 어떠한 변수에 어떠한 형태의 변화가 발생했는지에 대한 정보를 제공하지는 않는다. 그러나 이 논문에서 소개하는 의사결정나무를 이용한 다변량 공정관리 절차에서는 의사결정나무의 분류 규칙을 이용하기 때문에 이상상태의 형태를 분류할 수 있다. Table 3.1부터 Table 3.6에서 정분류율은 Table 2.3에서 구분한 이상상태의 형태를 얼마나 정확하게 분류하였는가에 대한 비율이다. 함께 제공한 오분류율은 잘못 분류된 경우 가장 비율이 높은 두 집단과 그 집단으로 분류된 비율을 나타낸다.

Table 3.1부터 Table 3.6을 살펴보면 이상상태에 대한 탐지 능력, 즉 평균런길이는 상관계수와 상관없이 전반적으로 유사한 것을 알 수 있다. 이상상태의 분류 정확도는 전체적으로는 상관계수에 상관없이 비슷한 성능을 가지지만, 오분류율이 크게 나와서 효율이 좋지 않은 경우도 있다.

모의실험의 결과를 다음과 같이 요약할 수 있다. 첫째, 상관계수의 절대값이 큰 경우 공정평균의 변화가 상관계수와 같은 방향으로 변화했을 때, 즉 상관계수가 양(+)인 경우에는 k_1과 k_2의 부호가 같고 음(−)인 경우에는 k_1과 k_2의 부호가 다를 때, 정분류율이 높게 나타났다. 그러나 공정평균의 변화가 상관계수와 다른 방향으로 변화했을 때에는 정뷴류율이 상대적으로 낮게 나타남을 알 수 있다. 예를 들어 Table 3.2를 살펴보면 k_1과 k_2의 부호가 같은 경우 정분류율이 매우 높게 나타났다.

둘째, 정분류율이 낮고 오분류율이 상대적으로 높은 경우를 살펴보면, 주로 유사한 집단으로 오분류되는 경우가 많음을 알 수 있다. 예를 들어, Table 3.1에서 (k_1, k_2)가 $(-2, 0)$일 경우에는 T7으로 분류되어야 하지만, k_1이 아래로 변하고 k_2가 위로 변하는 집단인 T6으로 오분류되는 비율이 50.2%가 되는 것을 알 수 있다.

결론적으로 의사결정나무를 이용한 다변량 공정관리 절차의 효율은 상관계수와 이상원인의 형태에 따라 달라질 수 있다. 그러나 이 절차의 장점은 이상상태의 신호뿐만 아니라 이상상태의 유형에 대한 정보도 함께 제공하기 때문에, 이상상태에 대한 실무자의 신속한 대처가 가능할 것으로 판단된다.

Table 3.1 Performance of the MSPC procedure when $\rho = -0.9$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	228.47	T0	-	-
	-3	3.48	T8	88.40%	T3 (7.8%), T6 (3.1%)
	-2	3.50	T8	82.40%	T6 (7.3%), T3 (7.1%)
-3	0	3.48	T7	20.00%	T6 (49.7%), T2 (16.5%)
	2	4.16	T6	87.70%	T7 (5.7%), T4 (3.7%)
	3	4.21	T6	91.40%	T4 (4.8%), T7 (2.7%)
	-3	3.54	T8	82.10%	T3 (10.4%), T6 (4.3%)
	-2	3.59	T8	58.50%	T3 (11.7%), T5 (9.7%)
-2	0	3.73	T7	15.50%	T6 (50.2%), T2 (18.5%)
	2	5.63	T6	97.70%	T4 (1.1%), T3 (0.9%)
	3	4.11	T6	84.10%	T4 (9.3%), T7 (4.6%)
	-3	3.75	T5	37.90%	T3 (43.7%), T6 (4.9%)
	-2	3.87	T5	26.50%	T3 (48.7%), T2 (11.1%)
0	2	4.24	T4	50.10%	T6 (27.5%), T2 (17.4%)
	3	4.04	T4	53.90%	T6 (34.6%), T2 (6.7%)
	-3	4.04	T3	89.40%	T2 (5.1%), T5 (3.9%)
	-2	5.61	T3	98.30%	T2 (1.2%), T5 (0.2%)
2	0	4.13	T2	73.00%	T3 (18.7%), T4 (6.3%)
	2	3.97	T1	48.00%	T2 (28.0%), T4 (13.0%)
	3	3.65	T1	71.90%	T8 (8.5%), T2 (7.6%)
	-3	4.14	T3	92.40%	T2 (6.3%), T5 (0.8%)
	-2	4.17	T3	84.40%	T2 (13.2%), T6 (0.9%)
3	0	3.80	T2	79.40%	T3 (13.7%), T4 (4.1%)
	2	3.67	T1	72.80%	T2 (12.3%), T8 (6.4%)
	3	3.62	T1	71.00%	T8 (10.1%), T6 (6.3%)
average		4.01	-	69.03%	-

* CR: classification rate

Table 3.2 Performance of the MSPC procedure when $\rho = 0.9$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	355.22	T0	-	-
	-3	5.01	T8	99.50%	T7 (0.3%), T1 (0.1%)
	-2	5.01	T8	91.40%	T7 (8.1%), T1 (0.3%)
-3	0	4.97	T7	85.50%	T8 (11.3%), T4 (1.4%)
	2	4.33	T6	44.90%	T7 (38.6%), T1 (8.0%)
	3	4.12	T6	45.60%	T7 (33.3%), T1 (13.2%)
	-3	5.03	T8	96.80%	T5 (1.5%), T2 (1.0%)
	-2	6.15	T8	99.60%	T7 (0.2%), T1 (0.1%)
-2	0	5.09	T7	77.10%	T8 (16.3%), T4 (5.0%)
	2	4.58	T6	30.80%	T7 (39.3%), T4 (17.8%)
	3	4.31	T6	42.90%	T7 (34.6%), T1 (11.1%)
	-3	4.96	T5	74.10%	T8 (16.6%), T2 (7.2%)
	-2	5.22	T5	75.40%	T8 (15.7%), T2 (8.1%)
0	2	4.97	T4	65.40%	T1 (24.8%), T7 (8.8%)
	3	4.76	T4	61.80%	T1 (30.9%), T7 (3.8%)
	-3	4.42	T3	47.80%	T5 (22.7%), T6 (16.9%)
	-2	4.64	T3	30.00%	T5 (34.1%), T2 (15.5%)
2	0	5.07	T2	42.80%	T5 (28.9%), T1 (27.6%)
	2	5.48	T1	99.20%	T8 (0.5%), T2 (0.2%)
	3	4.50	T1	97.90%	T4 (1.1%), T8 (0.6%)
	-3	4.05	T3	42.90%	T6 (28.2%), T1 (19.0%)
	-2	4.20	T3	34.20%	T6 (28.4%), T1 (15.3%)
3	0	4.77	T2	32.20%	T1 (36.2%), T5 (29.6%)
	2	4.31	T1	98.70%	T2 (1.1%), T5 (0.1%)
	3	4.15	T1	99.60%	T8 (0.2%), T2 (0.1%)
average		4.75	-	67.34%	-

* CR: classification rate

Table 3.3 Performance of the MSPC procedure when $\rho = -0.5$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	155.52	T0	-	-
	-3	3.92	T8	77.50%	T5 (9.7%), T7 (8.9%)
	-2	4.17	T8	71.80%	T7 (14.0%), T5 (9.7%)
-3	0	5.02	T7	65.00%	T6 (18.6%), T8 (8.0%)
	2	4.46	T6	81.00%	T4 (8.7%), T7 (8.2%)
	3	4.25	T6	81.80%	T4 (11.3%), T7 (5.0%)
	-3	4.14	T8	65.30%	T5 (18.7%), T7 (13.4%)
	-2	4.42	T8	51.50%	T7 (23.8%), T5 (20.8%)
-2	0	5.92	T7	72.20%	T6 (19.3%), T5 (3.8%)
	2	5.20	T6	74.80%	T4 (17.2%), T7 (7.0%)
	3	4.58	T6	62.40%	T4 (31.7%), T7 (3.6%)
	-3	4.98	T5	80.00%	T8 (9.2%), T3 (5.2%)
	-2	6.17	T5	83.80%	T3 (5.7%), T7 (3.8%)
0	2	6.00	T4	89.30%	T6 (5.4%), T2 (2.5%)
	3	5.03	T4	91.80%	T6 (2.9%), T1 (2.5%)
	-3	4.90	T3	47.20%	T5 (30.1%), T2 (20.1%)
	-2	5.26	T3	52.90%	T2 (28.8%), T5 (16.8%)
2	0	5.97	T2	88.40%	T3 (5.8%), T4 (2.1%)
	2	4.91	T1	31.10%	T4 (43.1%), T2 (22.9%)
	3	4.69	T1	40.50%	T4 (45.4%), T2 (11.9%)
	-3	4.31	T3	43.20%	T2 (32.7%), T5 (20.6%)
	-2	4.68	T3	42.00%	T2 (43.8%), T5 (12.0%)
3	0	4.77	T2	87.00%	T3 (3.4%), T1 (3.1%)
	2	4.56	T1	41.70%	T2 (29.3%), T4 (27.0%)
	3	4.27	T1	48.70%	T4 (25.9%), T2 (23.0%)
average		4.86	-	65.45%	-

* CR: classification rate

Table 3.4 Performance of the MSPC procedure when $\rho = 0.5$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	300.12	T0	-	-
	-3	4.34	T8	47.50%	T5 (25.8%), T7 (24.2%)
	-2	4.64	T8	41.90%	T7 (40.1%), T5 (14.4%)
-3	0	4.89	T7	80.40%	T6 (10.2%), T8 (4.2%)
	2	4.07	T6	53.80%	T7 (37.2%), T4 (6.6%)
	3	3.99	T6	61.50%	T7 (30.7%), T4 (6.2%)
	-3	4.82	T8	53.80%	T5 (32.5%), T7 (9.7%)
	-2	5.61	T8	64.90%	T7 (17.2%), T5 (16.2%)
-2	0	6.46	T7	86.00%	T6 (5.2%), T8 (4.8%)
	2	4.59	T6	51.00%	T4 (24.3%), T7 (23.1%)
	3	4.48	T6	63.40%	T4 (19.0%), T7 (16.9%)
	-3	5.19	T5	84.30%	T8 (5.6%), T3 (3.6%)
	-2	6.43	T5	86.60%	T8 (5.2%), T3 (3.0%)
0	2	6.70	T4	88.70%	T1 (5.2%), T6 (3.9%)
	3	5.39	T4	86.60%	T6 (6.7%), T1 (4.4%)
	-3	4.59	T3	44.40%	T5 (33.6%), T2 (15.4%)
	-2	5.03	T3	38.50%	T2 (34.5%), T5 (21.3%)
2	0	6.40	T2	90.60%	T1 (3.5%), T3 (2.1%)
	2	5.94	T1	61.30%	T2 (19.0%), T4 (18.5%)
	3	5.40	T1	54.60%	T4 (29.8%), T2 (14.9%)
	-3	4.43	T3	48.30%	T5 (26.5%), T2 (20.6%)
	-2	4.69	T3	42.00%	T2 (35.0%), T5 (16.8%)
3	0	5.28	T2	87.60%	T1 (3.9%), T4 (3.5%)
	2	5.13	T1	46.70%	T2 (36.8%), T4 (15.1%)
	3	5.00	T1	50.50%	T2 (25.7%), T4 (22.5%)
average		5.15	-	63.12%	-

* CR: classification rate

Table 3.5 Performance of the MSPC procedure when $\rho = -0.1$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	142.01	T0	-	-
	-3	3.94	T8	41.50%	T7 (39.1%), T5 (15.4%)
	-2	4.12	T8	39.20%	T7 (48.7%), T5 (8.0%)
-3	0	4.25	T7	86.80%	T6 (6.6%), T8 (4.3%)
	2	4.42	T6	59.50%	T7 (35.0%), T4 (3.2%)
	3	4.27	T6	71.50%	T7 (21.7%), T4 (4.4%)
	-3	4.19	T8	42.50%	T5 (26.9%), T7 (26.7%)
	-2	4.57	T8	45.00%	T7 (33.8%), T5 (17.7%)
-2	0	5.83	T7	82.60%	T6 (9.6%), T8 (4.9%)
	2	5.13	T6	67.80%	T7 (21.5%), T4 (8.7%)
	3	4.83	T6	73.90%	T7 (13.1%), T4 (11.3%)
	-3	4.59	T5	87.70%	T3 (4.9%), T8 (3.9%)
	-2	5.78	T5	90.40%	T8 (3.9%), T3 (2.9%)
0	2	6.49	T4	86.70%	T6 (5.8%), T1 (5.2%)
	3	5.76	T4	80.90%	T1 (8.9%), T6 (8.5%)
	-3	4.31	T3	36.40%	T5 (57.9%), T2 (4.2%)
	-2	4.92	T3	48.50%	T5 (41.4%), T2 (7.9%)
2	0	6.60	T2	85.30%	T3 (7.4%), T1 (2.9%)
	2	5.46	T1	67.10%	T2 (19.2%), T4 (11.8%)
	3	5.12	T1	74.70%	T2 (12.6%), T4 (10.0%)
	-3	4.09	T3	41.30%	T5 (47.7%), T2 (8.0%)
	-2	4.52	T3	56.00%	T5 (28.2%), T2 (13.3%)
3	0	5.59	T2	81.10%	T3 (9.9%), T1 (4.4%)
	2	4.99	T1	66.30%	T2 (27.6%), T4 (3.8%)
	3	4.85	T1	75.00%	T2 (19.3%), T4 (4.0%)
average		4.94	-	66.15%	-

* CR: classification rate

Table 3.6 Performance of the MSPC procedure when $\rho = 0.1$

k_1	k_2	ARL	class	correct CR	incorrect CR
0	0	109.46	T0	-	-
-3	-3	4.37	T8	77.80%	T5 (14.8%), T7 (4.4%)
	-2	4.81	T8	76.20%	T7 (12.4%), T5 (9.9%)
	0	5.53	T7	79.80%	T6 (8.8%), T8 (7.3%)
	2	4.42	T6	79.30%	T7 (13.2%), T4 (5.5%)
	3	4.14	T6	87.70%	T7 (5.7%), T4 (4.9%)
-2	-3	4.62	T8	59.70%	T5 (34.7%), T7 (4.6%)
	-2	5.19	T8	64.00%	T5 (20.5%), T7 (14.6%)
	0	6.20	T7	84.90%	T8 (6.1%), T6 (4.7%)
	2	4.86	T6	76.20%	T4 (11.4%), T7 (9.7%)
	3	4.41	T6	76.90%	T4 (16.9%), T7 (4.3%)
0	-3	4.67	T5	89.90%	T3 (3.9%), T8 (3.9%)
	-2	6.16	T5	86.90%	T3 (4.4%), T8 (4.4%)
	2	6.55	T4	84.00%	T6 (7.4%), T1 (5.0%)
	3	5.69	T4	82.70%	T6 (9.2%), T1 (4.5%)
2	-3	4.20	T3	52.30%	T5 (34.0%), T2 (12.1%)
	-2	4.79	T3	49.70%	T2 (24.3%), T5 (24.3%)
	0	5.91	T2	82.70%	T1 (6.1%), T3 (5.3%)
	2	4.77	T1	62.40%	T2 (18.1%), T4 (16.4%)
	3	4.60	T1	68.20%	T4 (18.1%), T2 (10.4%)
3	-3	4.17	T3	62.50%	T5 (19.7%), T2 (14.7%)
	-2	4.55	T3	59.00%	T2 (25.6%), T5 (12.9%)
	0	4.96	T2	81.50%	T1 (8.3%), T3 (5.1%)
	2	4.26	T1	66.50%	T2 (20.0%), T4 (9.7%)
	3	4.14	T1	77.10%	T4 (14.4%), T2 (6.1%)
average		4.92	-	73.66%	-

* CR: classification rate

4. 결론

컴퓨터의 발전과 통신 속도의 향상, 그리고 이를 처리하는 데이터베이스 시스템이 발전함으로써 사회 각 분야에서 데이터의 축적 속도가 기하급수적으로 늘어났고 이에 대한 처리 또한 가능하게 되었다. 제조 환경 역시 공정의 자동화와 동시에 각 품질변수에서 생성된 데이터를 실시간으로 저장할 수 있게 되었다. 이에 따라 실시간으로 들어오는 다변량 데이터를 신속하게 처리하기 위해서 데이터마이닝 기법을 적용한 공정관리 절차에 대한 연구가 최근에 많이 진행되고 있다.

이 논문에서는 데이터를 처리하는 방법으로 이동식 윈도우를 사용하였고, 의사결정나무를 이용한 다변량 공정관리 절차를 소개하고 그 효율에 대하여 알아보았다. 모의실험을 통하여 얻은 결과를 정리하면 다음과 같다.

첫째, 윈도우 크기에 따라서 관리 절차의 효율이 달라진다. 즉, 윈도우 크기가 커질수록 관리상태의 평균런길이는 커지지만 이상상태의 탐지 성능화이 떨어지며, 작아질 경우에는 그 반대가 된다. 모의실험 결과 윈도우 크기는 10이 적당한 것으로 나타났다.

둘째, 학습 데이터를 생성할 때, 관리상태와 이상상태를 구분하는 변화시점 또한 관리 절차의 효율에 영향을 준다. 변화시점을 윈도우 크기의 1/4, 1/2, 3/4으로 설정하여 모의실험을 수행한 결과 1/4을 사용하는 것이 가장 바람직하다는 결론을 얻었다.

셋째, 의사결정나무의 분류 규칙을 이용한 다변량 공정관리 절차는 전통적으로 사용하는 관리도와는 달리 이상상태의 유형에 대한 정보를 제공하기 때문에, 실제 공정관리에서 유용하게 사용될 수 있을 것이라 판단된다.

향후 품질변수의 수가 3개 이상인 경우와 다른 기계학습 기법을 이용한 다변량 공정관리 절차에 대한 연구가 진행되어야 한다고 생각한다.

References

Chen, L. H. and Wang, T. Y. (2004). Artificial neural networks to classify mean shifts from multivariate chi^2 chart signals. *Computers & Industrial Engineering*, **47**, 195-205.

Cho, G. Y. (2010). Multivariate Shewhart control charts with variable sampling intervals. *Journal of the Korean Data & Information Science Society*, **21**, 999-1008.

Cho, G. Y. and Park, J. S. (2013). Parameter estimation in a readjustment procedure in the multivariate integrated process control. *Journal of the Korean Data & Information Science Society*, **24**, 1275-1283.

Guh, R. S. (2005). A hybrid learning-based model for on-line detection and analysis of control chart patterns. *Computers & Industrial Engineering*, **49**, 35-62.

Guh, R. S. (2007). On-line identification and quantification of mean shifts in bivariate processes using a neural network-based approach. *Quality and Reliability Engineering International*, **23**, 367-385.

Guh, R. S. and Shiue, Y. R. (2005). On-line identification of control chart pattern using self-organizing approaches. *International Journal of Production Research*, **43**, 1225-1254.

Guh, R. S. and Shiue, Y. R. (2008). An effective application of decision tree learning for on-line detection of mean shifts in multivariate control charts. *Computers & Industrial Engineering*, **55**, 475-493.

Guh, R. S. and Tannock, J. D. T. (1999). Recognition of control chart concurrent pattern using a neural network approach. *International Journal of Production Research*, **37**, 1743-1765.

Ho, E. S. and Chang, S. I. (1999). An integrated neural network approach for simultaneous monitoring of process mean and variance shifts - a comparative study. *International Journal of Production Research*, **37**, 1743-1765.

Hwarng, H. B. (2005). Simultaneous identification of mean shift and correlation change in AR(1) processes. *International Journal of Production Research*, **43**, 1761-1783.

Quinlan, J. R. (1998). *C5.0: An informal tutorial*, RuleQuest, Australia.

Journal of the Korean Data & Information Science Society
2015, **26**(3), 639–652

http://dx.doi.org/10.7465/jkdi.2015.26.3.639
한국데이터정보과학회지

Multivariate process control procedure using a decision tree learning technique[†]

Kwang Young Jung[1] · Jaeheon Lee[2]

[12]Department of Applied Statistics, Chung-Ang University

Received 16 March 2015, revised 30 March 2015, accepted 27 April 2015

Abstract

In today's manufacturing environment, the process data can be easily measured and transferred to a computer for analysis in a real-time mode. As a result, it is possible to monitor several correlated quality variables simultaneously. Various multivariate statistical process control (MSPC) procedures have been presented to detect an out-of-control event. Although the classical MSPC procedures give the out-of-control signal, it is difficult to determine which variable has caused the signal. In order to solve this problem, data mining and machine learning techniques can be considered. In this paper, we applied the technique of decision tree learning to the MSPC, and we did simulation for MSPC procedures to monitor the bivariate normal process means. The results of simulation show that the overall performance of the MSPC procedure using decision tree learning technique is similar for several values of correlation coefficient, and the accurate classification rates for out-of-control are different depending on the values of correlation coefficient and the shift magnitude. The introduced procedure has the advantage that it provides the information about assignable causes, which can be required by practitioners.

Keywords: Computer integrated manufacturing, data mining, decision tree learning, multivariate process control.

[†] This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A1A2054200).
[1] Graduate student, Department of Applied Statistics, Chung-Ang University, Seoul 156-756, Korea.
[2] Corresponding author: Professor, Department of Applied Statistics, Chung-Ang University, Seoul 156-756, Korea. E-mail: jaeheon@cau.ac.kr