Journal of the Korean Institute of Industrial Engineers
Vol. 31, No. 3, pp. 257-264, September 2005.

자기 조직화 신경망을 이용한 클러스터링 알고리듬

이종섭[1†] · 강맹규[2]

[1]우송대학교 IT(경영정보)학과 / [2]한양대학교 정보경영공학과

A Clustering Algorithm using Self-Organizing Feature Maps

Jong-Sub Lee[1] · Maing-Kyu Kang[2]

[1]Department of Technical Management Information Systems, Woosong University, Daejeon, 300-718
[2]Department of Information & Industrial Engineering, Hanyang University, Ansan, 425-791

This paper suggests a heuristic algorithm for the clustering problem. Clustering involves grouping similar objects into a cluster. Clustering is used in a wide variety of fields including data mining, marketing, and biology. Until now there are a lot of approaches using Self-Organizing Feature Maps(SOFMs). But they have problems with a small output-layer nodes and initial weight. For example, one of them is a one-dimension map of *k* output-layer nodes, if they want to make *k* clusters. This approach has problems to classify elaboratively. This paper suggests one-dimensional output-layer nodes in SOFMs. The number of output-layer nodes is more than those of clusters intended to find and the order of output-layer nodes is ascending in the sum of the output-layer node's weight. We can find input data in SOFMs output node and classify input data in output nodes using Euclidean distance. We use the well known IRIS data as an experimental data. Unsupervised clustering of IRIS data typically results in 15 - 17 clustering error. However, the proposed algorithm has only six clustering errors.

***Keywords*:** clustering, Self-Organizing Feature Maps, unsupervised neural network, euclidean distance

1. 서 론

클러스터링(Clustering)은 데이터(Data)를 몇 개의 클러스터(Cluster)로 대응시키는 과정이다(Aldenderfer, 1984; Everitt, 1993; Everitt, 2001). 클러스터링 문제는 고차원을 갖는 n개의 데이터들을 입력으로 하여 유사한 특성을 갖는 k개의 클러스터로 나누는 것을 말한다. 한 클러스터 내의 데이터들은 다른 클러스터 내의 데이터들과 비교하여 높은 유사성을 보이지만, 다른 클러스터 내의 데이터들과는 아주 다르다. 데이터들 사이의 유사한 정도를 평가하는 척도로서 유사도(Similarity)는 데이터의 속성 값(Attribute Value)에 기초한 유클리드 거리(Euclidean Distance)에 의하여 계산된다.

클러스터링의 적용분야는 기계-부품 그룹형성(Machine-Part Grouping), 토량이동 최적화 연구, 고객세분화(Customer Segmentation), 패턴인식(Pattern Recognition) 등과 같이 다양하며, 특히 데이터마이닝, 마케팅, 그리고 생물학 등에 유용하다(Berry *et al*., 1997; Berry *et al*., 2004). 클러스터링은 데이터의 분포에 대한 이해를 구하거나 각 클러스터의 특성을 관찰하는 단독적인 도구로 사용되기도 하고, 다른 알고리듬의 전처리 단계로 사용되기도 한다.

클러스터링 알고리듬의 종류는 크게 두 가지로 나누어지는데, 그 하나는 계층 알고리듬(Hierarchical Algorithms)이고, 나머지는 분할 알고리듬(Partition Algorithms)이다. 최근에는 신경망, 퍼지-신경망 등을 이용한 클러스터링 알고리듬이 활발히 연구되고 있다.

Mangiameli *et al*.(1996)에 의하면 계층 알고리듬에는 두 클러스터 간의 유사성을 최단거리로 측정하는 단일접합 클러스터링(Single Linkage Clustering)과 두 클러스터 간 최장거리로 유사성을 측정하는 완전접합 클러스터링(Complete Linkage Clustering), 두 클러스터 간의 평균거리로 유사성을 측정하는

† 연락저자 : 이종섭 교수, 300-718 대전광역시 동구 자양동 17-2번지 우송대학교 IT(경영정보)학과, Fax : 042-630-9859, E-mail : ljs@wsu.ac.kr
2005년 3월 접수; 2005년 7월, 8월 수정본 접수; 2005년 8월 게재 확정.

평균접합 클러스터링(Group-Average Clustering), 두 클러스터 간의 밀도로서 유사성을 측정하는 Ward's 계층적 클러스터링(Ward's Hierarchical Clustering) 등이 있다. 계층 알고리듬은 초기에 이루어진 부적절한 병합으로 인한 문제점을 보완할 수 없는 단점을 가지고 있다.

분할 알고리듬은 데이터들을 분할하여 같은 클러스터 내에 있는 데이터들 사이의 유사한 정도는 다른 클러스터에 있는 데이터들보다 유사한 정도가 크도록 클러스터를 형성한다. 분할 알고리듬에는 대표적으로 k-Means 알고리듬과 ISODATA 알고리듬이 있다. k-Means 알고리듬은 각 데이터와 각 클러스터의 중심 값과의 거리차이를 최소화시키는 방향으로 알고리듬이 반복됨으로써 각 클러스터에 대한 데이터들이 재배열이 가능하도록 한 알고리듬이다.

이것은 계층 알고리듬에서 초기에 이루어진 부적절한 병합으로 인한 문제점을 보완할 수 없는 단점을 극복하였다. ISODATA 알고리듬은 k-Means 알고리듬과 같이 k개의 중심을 가지고 시작하지만 반드시 클러스터의 개수가 k개가 되는 것은 아니다. 이로써 더욱 유동적으로 알고리듬 수행중에 클러스터의 개수를 변할 수 있다. ISODATA 알고리듬은 고정적인 클러스터 개수인 k-Means 알고리듬의 단점을 어느 정도 보완하였다.

대부분의 클러스터링 알고리듬은 고차원 공간을 갖는 데이터에는 좋은 결과를 제공하지 못하고 있는데 이것은 데이터 자체의 고유한 희박성이 원인이 되고 있다. 이와 같은 문제를 해결하기 위한 방법의 하나로서 우선 특징 추출(Feature Selection)을 먼저 시행하여 그 차원을 줄여서 저차원 공간을 갖는 데이터를 이용하여 클러스터링하는 알고리듬을 적용하기도 한다. 데이터 상에 존재하는 특정 차원만을 선택하고, 나머지 차원들은 잡음으로 간주하여 제거하는 알고리듬은 특정 차원만을 미리 추출하여 사용하게 됨으로써 정보의 손실을 가져올 수 있다.

신경망을 이용한 클러스터링 알고리듬은 Kohonen 네트워크, Carpenter와 Grossberg 네크워크 등이 있다. Kohonen 네트워크 알고리듬은 자기 조직화 신경망과LVQ 알고리듬이 있으며, 이 알고리듬들은 데이터와 클러스터의 중심 값과의 거리를 최소화시키는 학습 알고리듬에 따라 클러스터링한다는 점에서 k-Means 알고리듬과 대응된다고 할 수 있다.

클러스터에 대한 정보를 미리 알고 학습과정에서 정보를 반영할 수 있는 감독학습과 달리 자기 조직화 신경망은 무감독 학습을 사용하는 신경망의 일종으로 출력노드에 입력데이터의 유사성을 반영하는 특성지도(Feature Maps)를 스스로 형성한다. 즉, 자기 조직화 신경망은 각 각의 입력데이터를 출력노드로 대응할 수 있는 능력을 가진다.

Huntsberger and Ajjimarangsee(1989)는 학습률과 이웃범위 등의 파라미터를 변경하면서 Kohonen의 학습방법을 변형한 클러스터링 알고리듬을 제시하였다.

Pal *et al.*(1993)는 초기 연결강도에 의해 입력벡터와 출력노드 간의 거리에 대한 가중값을 준 손실함수를 정의하고, 이 손실함수를 최소화시키는 경쟁학습 신경망 알고리듬을 제시하였다.

Tasao *et al.*(1994)과 Karayiannis(1997)은 신경망에 퍼지(Fuzzy)개념을 결합한 방법을 제시하였다. 특히, Karayiannis (1997)은 FALVQ(Fuzzy Algorithm for Learning Vector Quantization)입력데이터와 LVQ(Learning Vector Quantization) 네트워크의 연결강도 사이의 제곱 유클리디언 거리의 가중합을 최소로 하는 알고리듬을 개발하였다.

Kusiak(2000)은 클러스터링 문제를 NP-Complete 문제로 정의하였다. 이것은 그룹을 형성해야 할 기계의 개수가 많은 경우에는 계산량이 지수적으로 증가하여 많은 시간이 소요된다는 것을 의미한다. 따라서 최적화 해법보다 발견적 해법(heuristic algorithm)을 많이 사용한다.

본 연구에서는 Anderson의 IRIS 데이터 세트와 기계-부품 행렬 데이터를 입력데이터로 하여 자기 조직화 신경망이 무감독 학습을 진행됨에 따라 입력데이터와 유사한 형태로 변형되는 연결강도의 특성을 이용한다. 본 연구에서 결정해야 할 매개변수는 1차원 출력노드의 개수, 학습률, 이웃의 범위 등이다. 제안하는 알고리듬은 이와 같은 매개변수를 이용하여 학습을 진행하고, 출력노드 i와 j 사이의 연결강도 거리(Weight Distance) $WD(i, j)$의 크기에 따라 그룹을 형성하였다. 그 결과 다른 클러스터링 알고리듬과 비교하여 오분류의 개수를 최소화하는 알고리듬을 제안한다.

2. 자기 조직화 신경망의 일반적인 고찰

SOFMs은 Kohonen(1984, 1988, 1997)에 의해 제시된 경쟁학습 신경망(Competitive Learning Neural Network) 모델이다. SOFMs의 구조는 <Figure 1>과 같이 m개의 입력노드로 이루어진 입력층과 n(즉 $p \times p$)개의 출력노드로 이루어진 출력층 두 개의 층으로 구성되어 있다.

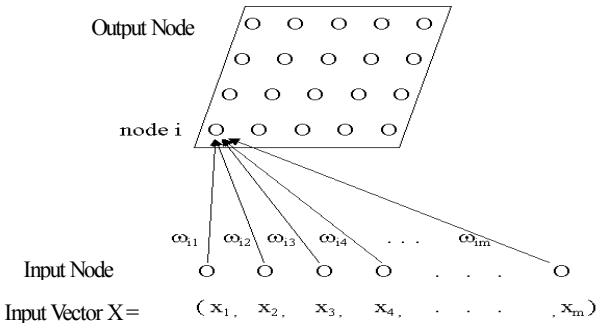

Figure 1. General structure of SOM.

입력층은 입력데이터를 받아 이들을 출력층에 사상(mapping)한다. 출력층은 1차원 또는 2차원 구조를 사용한다.

출력노드의 개수는 입력데이터가 서로 다른 출력노드에 펼쳐질 수 있도록 입력데이터의 개수보다 큰 수로 하거나 사용자가 원하는 임의의 개수 k로 정할 수 있다. 출력층의 각 노드에는 입력데이터가 사상된다.

입력층의 모든 노드와 출력층의 모든 노드는 연결되어 있고, 출력노드 i ($1 \le i \le n$)와 입력노드 j ($1 \le j \le m$) 사이의 연결선은 연결강도(Connecting Weight) ω_{ij}를 가진다. 연결강도는 0과 1 사이의 실수로서, 초기에는 임의로 주어지지만 입력데이터에 따라 조절된다. 각 입력데이터에 대하여 이와 가장 유사한 출력노드인 승자노드 i^*를 결정하는데, 이는 식 (1)과 같이 입력데이터 X와 연결강도 ω_i 사이의 거리 D_i를 계산하여 가장 작은 출력노드 i^*로 정한다.

$$D_i = \sqrt{(x_1 - \ _{i1})^2 + (x_2 - \ _{i2})^2 + \cdots + (x_m - \ _{im})^2} \quad i = 1, 2, ..., n \tag{1}$$

출력노드 i^*의 연결강도(Wi^*)는 식 (2)와 같이 n개의 출력노드 중에서 입력데이터 X와 가장 가까운 연결강도이다.

$$|X - W_{i^*}| = min|X - W_i| \qquad \forall i \tag{2}$$

승자노드 i^*의 앞과 뒤에 위치한 노드를 이웃노드(Neighbor Node)라고 하는데, 이웃범위(Neighborhood) $Ni^*(\delta)$는 승자노드 i^*로부터 δ만큼 떨어진 이웃노드의 집합이다. <Figure 2>는 승자노드를 "#"으로 표시하고, 그 외의 출력노드를 "*"로 표시할 때 2차원 출력층에서 사각그리드(Rectangular Grid)를 사용하여 반경(Radius)이 δ=0, 1, 2인 경우로 이웃노드를 표현하였다.

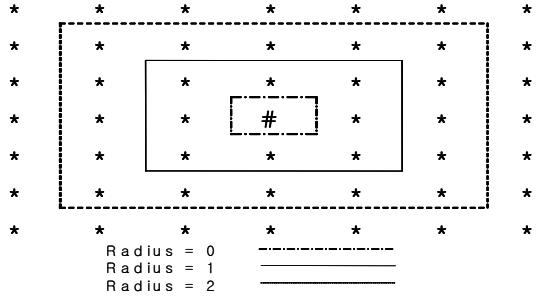

Figure 2. Neighborhood using rectangular grid in two dimension.

각 출력노드 i의 연결강도는 식 (3)과 같이 이웃범위에 속하느냐($i \in N_{i^*}(\delta)$) 그렇지 않느냐($i \notin N_{i^*}(\delta)$)에 따라 다르게 적용된다.

$$\frac{dW_i}{dt} = \alpha(t)(X - W_i) \quad i \in N_i^*(\delta) \tag{3}$$

$$\frac{dW_i}{dt} = 0 \quad i \notin N_i^*(\delta)$$

$N_{i^*}(\delta)$를 이웃범위(neighborhood)라고 하면 자기 조직화 신경망의 연결강도는 각 입력데이터에 대하여 이웃범위와 학습률을 감소시키면서 이웃범위가 승자노드 자신이 될 때까지 승자노드와 그 이웃노드의 연결강도를 식 (4)와 같이 조절한다. 학습률 $\alpha(t)$는 시간 t의 흐름에 따라 입력데이터와 기존의 연결강도 간의 차이를 조정하는 비율이다. 이것은 0과 1 사이의 값을 가지고, 학습이 진행됨에 따라 점차 줄여나간다. 일반적으로 학습률이 너무 크면 학습이 제대로 되지 않고, 너무 작으면 학습시간이 오래 걸린다(Kohonen, 1984).

$$W(new)_{ij} = W(old)_{ij} + \alpha(t)(X_i - W(old)_{ij}) \quad (i \in N_i^*(\delta), j = 1, \ldots, m) \tag{4}$$

여기에서 $W(\text{old})_{ij}$는 조절되기 전의 연결강도이고, $W(\text{new})_{ij}$는 조절된 후의 연결강도이다.

Kohonen의 SOFMs 학습 알고리듬은 다음과 같다

절차 1 : 연결강도를 초기화한다. 학습률과 이웃의 범위를 정한다.
절차 2 : 입력층에 하나의 입력데이터를 입력한다.
절차 3 : 입력데이터와 연결강도 사이의 거리를 식 (1)과 같이 계산한다.
절차 4 : 하나의 승자노드를 결정한다.
절차 5 : 학습규칙에 따라 연결강도를 식 (4)와 같이 조절한다.
절차 6 : 이웃범위와 학습률을 감소시키면서 이웃범위가 승자노드 자신이 될 때까지 절차 2에서 절차 5까지 반복한다.

3. 제안하는 알고리듬

Kohonen의 SOFMs은 초기에 임의로 주어진 연결강도 및 학습률로 인하여 수행시마다 조금씩이지만 해의 변동이나 학습시간이 오래 걸리는 몇 가지 문제를 가지고 있다. 제안하는 SOFMs의 구조는 1차원 선형(linear) 출력층으로 이루어진 SOFMs으로 초기에 임의로 주어진 연결강도를 바탕으로 출력노드별로 연결강도의 합을 구한다. 이를 바탕으로 오름차순으로 정렬한다. 이렇게 구성된 출력노드에 입력데이터를 대응하고, 학습을 통하여 출력노드를 입력노드와 유사한 형태로 변형함으로써 그룹을 형성한다. 특히, 출력노드의 개수를 입력데이터의 개수보다 크게 설정함으로써 출력노드의 분포를 입력데이터의 분포와 유사한 형태로 펼쳐놓을 수 있도록 하였다. 학습과정에서 초기 이웃범위의 절반(1/2) 이 되는 시점에서 그 때까지 승자노드가 되지 못한 출력노드의 연결강도는 식 (4)와 같이 조절하지 않는다. 이와 같이 학습이 진행된 입력데이터가 출력노드에 유사한 순서대로 정렬되면 출력노드 간의 연결강도가 가장 큰 지점을 선형적으로 분리하여 원하는 그룹을 형성할 수 있다. 이것은 학습과정에서 발생할 수 있는 문제점뿐만 아니라 학습 이후에 그룹을 형성하는 과정을 단순하게 처리할 수 있다.

제안하는 클러스터링 알고리듬은 출력노드의 개수, 학습률, 그리고 이웃범위와 같은 매개변수를 결정한다. 출력노드의 개수는 적을수록 학습시간이 짧아지므로 입력데이터가 출력노드에 충분히 펼쳐질 수 있는 최소의 출력노드 개수를 설정한다. 경험적 방법에 의하면 출력노드 개수는 입력데이터 개수의 두 배 이상으로 정하면 학습과정에서 발생할 수 있는 결함을 안전하게 극복할 수 있는 가장 좋은 해를 구할 수 있음이 나타났다. 학습률은 클수록 학습시간이 줄어든다. 그러나 너무 크면 제대로 학습이 되지 않는다. 이웃범위는 적을수록 학습시간이 줄어든다. 초기에는 모든 출력노드를 이웃으로 설정하고, 시간이 지남에 따라 그 범위를 줄여나가고 이웃범위가 자기 자신일 때 알고리듬을 중지한다. 초기 학습률 즉 $\alpha(0)$는 경험적으로 좋은 결과를 제공하는 0.4로, 초기 이웃범위는 출력노드의 개수와 같은 수로 정한다.

각 입력데이터가 사상된 출력노드 i와 i+1 연결강도 간의 거리를 유클리드 거리를 이용하여 선형적으로 분리한다. 여기에서 i와 i+1은 이웃한 출력노드의 번호이다. 사상된 이웃한 출력노드의 연결강도 거리를 유클리드 거리로 사용하는 이유는 연결강도가 입력데이터와 유사한 확률분포함수 형태로 구성되기 때문에 단지 이웃한 출력노드의 연결강도 거리만을 고려하면 된다.

제안하는 클러스터링 알고리듬의 절차는 다음과 같다.

절차 1 : SOFMs의 구조(출력노드의 형태, 입·출력노드의 개수)를 초기화한다.

절차 2 : 각 연결선에 강도를 초기화하고, 초기 학습률 $\alpha(0)$ 및 학습률 함수 $\alpha(t)$ 그리고 초기 이웃범위를 정한다.

절차 3 : 출력노드별 연결강도 합을 구한다. 연결강도 합의 크기에 따라 오름차순으로 출력노드를 정렬한다.

절차 4 : 각 입력데이터에 대하여 출력노드의 연결강도와 입력데이터 사이의 거리를 식 (1)과 같이 계산한다. 그 중에서 가장 짧은 노드를 승자노드로 결정한다.

절차 5 : 승자노드로부터 일정한 범위 내 위치한 이웃노드의 연결강도를 식 (4)와 같이 조절한다.

절차 6 : 이웃범위는 1만큼, 학습률은 $\alpha(t)=(1-t/4950)\times\alpha(t-1)$ 만큼 감소시키면서 최종적으로 이웃범위가 승자노드 자신이 되거나 학습률이 0가 될 때까지 절차 4에서 절차 5까지 반복한다. 단, 이웃범위가 초기 이웃범위의 절반(1/2)이 되는 시점 이후에는 그동안 한 번도 승자노드가 되지 못한 출력노드는 학습을 시키지 않는다.

절차 7 : 각 입력데이터를 가장 가까운 출력노드에 사상시킨다.

절차 8 : 입력데이터가 대응된 이웃한 출력노드 i와 j 간의 연결강도 거리 WD(i, j)를 계산한다.

절차 9 : 선형구조로 이루어진 출력노드에서 연결강도 거리 WD(i, j) 차이가 가장 큰 출력노드 구간을 k-1개 선택하면 k개의 그룹을 형성할 수 있다.

4. 수치예제

본 연구에서 사용한 데이터는 Anderson의 IRIS 데이터와 기계-부품 행렬 데이터를 사용하였다. 특히, Anderson의 IRIS 데이터는 4차원(Petal Width, Petal Length, Sepal Width, Sepal Length) 속성을 가지는 150개 샘플 데이터로 구성된다. 세 개의 클러스터(Iris Setosa, Iris Verisiclolr, Iris Verginica)는 각각 50개의 데이터로 구성되어 있다. IRIS 데이터는 무감독학습을 이용한 클러스터링 알고리듬으로는 15-17개의 오분류가 나타난다고 알려져 있다(Pal *et al*., 1993).

본 연구에서 사용하는 IRIS 데이터에 대하여 제안하는 클러스터링 알고리듬을 적용하면 다음과 같다.

절차 1 : 본 예제에서 사용하는 SOFMs의 구조는 <Figure 3>과 같이 출력노드의 구조는 선형이고, 입력노드와 출력노드의 수가 각각 4, 300이다. 여기에서 출력노드의 구조는 선형이고, 출력노드의 개수는 입력데이터의 2배로 한다. 입력노드의 개수는 4개이다.

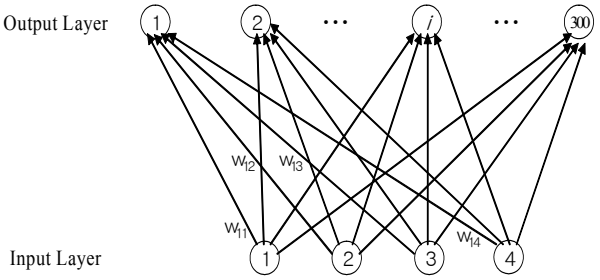

Figure 3. Suggesting structure of SOM.

절차 2 : 첫 번째 출력노드의 연결강도는 W_1={0.5882, 0.2500, 0.2200, 0.1872}이고, 두 번째 출력노드의 연결강도는 W_2={0.9232, 0.4950, 0.8613, 0.7534}이고, 274번째 출력노드의 연결강도 W_{274}={0.0145, 0.2887, 0.0584, 0.0835}이고, 300번째 출력노드의 연결강도는 W_{300}={0.7083, 0.8935, 0.8302, 0.0759}이다. 초기 학습률은 0.4)로 초기화하고, 학습함수는 $\alpha(t)=(1-t/4950)\times\alpha(t-1)$이다.

절차 3 : 첫 번째 출력노드 연결강도 합은 1.245이고, 두 번째 출력노드 연결강도 합은 2.5468이고, 274째 출력 노드 연결강도 합은 0.4453이고, 300번째 출력노드 연결강도 합은 2.5082이다. 크기별로 출력노드의 정렬순서는 274, 48, ... , 44, 89이다.

절차 4 : 첫 번째 입력데이터(0.02, 0.14, 0.33, 0.5)에 대하여 첫 번째 출력노드의 연결강도 W274와의 거리를 식

(1)과 같이 계산하면 0.2748이다. 출력노드의 연결강도 W_1, W_2, W_{300}, ... 에 대하여 계산하면 0.6902, 1.3757, 1.6861 ... 이고, 가장 짧은 출력노드W_{274}를 승자노드라고 한다.

절차5 : 반경 300에 있는 모든 출력노드의 연결강도를 식(4)와 같이 조절한다.

절차6 : 이웃범위를 1씩 줄이고 t=0인 경우, 즉 초기 학습률 $\alpha(0)$을 0.4로 t=1인 경우 0.4×(1-(1/4950))으로 감소시키면서 이웃범위가 승자노드 자신이 될 때까지(반경이 0이 될 때까지) 절차4에서 절차5까지 반

Table 1. The Data in the output nodes

No of Output Node	No of Input Data	No of Output Node	No of Input Data	No of Output Node	No of Input Data
1	12, 15, 24, 26,	87	74	189	104, 109, 149
	31, 41, 43	88	90	195	106
6	30	91	65	196	111
9	17, 50	95	70	198	88
11	6, 8, 33	96	59, 100	200	80
13	16	99	84	206	119
15	11, 49	101	54, 60	207	138
17	34	106	56	208	79
19	37, 44	109	82	212	135
20	22, 47	110	75	214	94
21	10	111	76	220	148
22	38	114	91	221	114, 143
23	28	115	105	223	132
24	23, 36	119	97	226	110
26	46	121	72, 93	231	108
27	4	124	51, 92	233	103
29	40	130	61	235	130
30	1	133	89	239	102
31	48	135	57	241	118
33	20	138	62	243	124
34	2	139	81	244	121
36	35	140	99	251	140
39	25, 29	141	69	252	131, 147
40	19	143	77	258	101
41	27	146	98	260	122
43	42	149	71	263	115, 117, 125
45	5	151	66	266	112
48	21, 45	154	73	267	120
49	13, 18	157	53, 83, 85, 96	275	116
51	14	159	52	277	142
53	9, 32, 39	161	67	280	141
56	3	162	86	281	126
57	7	171	63, 68	285	145
73	64	174	107, 139	288	146
75	55, 58	178	87, 127, 129	294	128
77	78	184	123	300	133, 134, 136,
86	95	186	113, 150		137, 144

복한다.

절차 7 : 각 입력데이터를 가장 가까운 출력노드에 사상시키면 <Table 1>과 같다.

절차 8 : <Table 1>에서 각각의 출력노드와 출력노드 간의 연결강도 거리를 계산하면 WD(1, 6)=0.0112이고, WD(6,9)=0.0074, WD(57, 73)=0.1738, WD(171, 174)=0.02865, ... , WD(294, 300)=0.0139이다.

절차 9 : 연결강도 거리가 가장 큰 출력노드와 출력노드 구간은 57과 73이고, 그 다음으로는 171와 174이다. 각각의 값은 WD(57, 73)=0.1738, WD(171, 174)=0.02865이다. 따라서 3개의 그룹은 <Table 2>와 같다.

Table 2. The data in 3 groups

No of Group	No of Input Data
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
2	51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 89, 90, 91, 92, 93, 95, 96, 97, 98, 99, 100, 105
3	79, 80, 87, 88, 94, 101, 102, 103, 104, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119,120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150

5. 실험 결과와 분석

본 연구에서 제안하는 알고리듬의 성능을 평가하기 위해 기존의 연구에서 인용되는 IRIS 데이터(Anderson, 1939)와 기계-부품 행렬 데이터를 대상으로 실험하였다.

본 연구에서 제안하는 SOFMs의 출력노드 구조는 1차원 선형구조를 하였으나 초기에 임의로 설정되는 연결강도로 인하여 좌측에서 우측으로 출력노드의 연결강도 합이 크기 순으로 정렬되지 못하였다. 이것을 수정하기 위하여 출력노드의 연결강도 합을 기준으로 좌측에서 우측으로 오름차순으로 정렬하였다. 출력노드 개수는 3개에서 시작하여 입력데이터의 4배까지 증가시키면서 실험한 결과 출력노드의 개수가 3개에서 입력데이터의 2배 미만에서는 수행 시마다 다소 다른 결과를 보였으나 입력데이터의 2배 이상일 경우는 동일한 결과를 나타냄을 확인하였다.

<Table 3>과 같은 실험에 의하면 오분류의 개수는 초기 학습률 및 학습함수, 이웃범위 등에 민감한 반응을 보였다. 여기에

Table 3. The number of minimal errors in 30 trials

초기 학습률 / 학습함수	0.1	0.2	0.3	0.4
1-t/3000	17	17	17	17
1-t/4000	17	17	15	15
1-t/5000	15	15	9	6
1-t/6000	17	17	15	12

서 학습함수를 1-t/5000로 한 경우에는 최소(6개)의 오분류를 보였으나 시행 시마다 약간 다른 결과를 보여주었다. 그러나 학습함수를 1-t/4950로 한 경우에는 안정적으로 최소의 오분류를 보장하였다. 이와 더불어 출력노드의 개수가 적은 경우 학습과정에서 발생할 수 있는 결함이 발생할 가능성이 큰 것으로 추정된다. 초기 학습률은 0.4로 학습함수는 (1-t×(1/4950))로 하였다. 초기 이웃범위는 모든 출력노드의 연결강도를 변경할 수 있도록 반경을 300으로 하였다. 학습이 진행되어 초기 이웃범위의 절반(1/2)이 되는 지점 이후에는 그때까지 승자노드가 되지 못한 출력노드는 연결강도를 조절하지 않는다.

<Table 4>는 Anderson의 IRIS 데이터에 대하여 제안하는 클러스터링 알고리듬과 기존 클러스터링 알고리듬의 해를 비교한 결과이다. 제안하는 클러스터링 알고리듬은 Pal *et al.*(1993)이 구한 오분류 17개와 Karayiannis(1997)이 구한 오분류 15개보다 더 좋은 오분류 6개를 갖는 해를 구하였다.

Pal *et al.*(1993)에 의하면 기존의 무감독학습을 이용한 클러스터링 알고리듬에서는 적어도 15-17개의 오분류를 생성한다고 하였다. <Table 4>에서 제안하는 클러스터링 알고리듬의 오분류 개수는 6으로 기존의 알고리듬에서 제시하는 15-17보다 더 적은 오분류를 나타낸다.

Table 4. The number of error comparison for IRIS data

Source of Problem	Source of Algorithms	The Number of Error
Anderson's IRIS Data Set	Proposed algorithm	6
	Pal *et al.* (1993)	17
	Karayiannis (1997)	15

제안하는 클러스터링 알고리듬은 4차원의 실수값을 가지는 IRIS 데이터에 적용한 방법과 같은 파라미터, 즉 초기 학습률을 0.4로, 학습함수를 (1-t×(1/4950))로 하여 제조분야에 널리 알려져 있는 기계-부품 그룹 형성문제에 적용하였다. 여기에 사용된 기계-부품 그룹 형성문제는 예외 요소가 존재하지 않은 기계-부품 행렬로 구성되어 있다. <Table 5>는 기계-부품 그룹을 형성하기 위한 첫 번째 단계로서 기계 그룹을 형성할 때 발생할 수 있는 최적의 그룹 수와 오분류의 개수를 제시한다. 제안하는 클러스터링 알고리듬은 0과 1로 구성된 상호 독립적인 기계-부품 행렬을 사용하여 기계그룹을 형성하는데 여기에

서 발생하는 기계그룹의 수와 오분류의 개수를 <Table 5>와 같이 제시한다. 제안하는 클러스터링 알고리듬은 <Table 5>와 같이 최적의 기계그룹 수와 최소의 오분류인 0개를 제시한다.

6. 결 론

본 연구에서는 IRIS 데이터의 분류 및 기계-부품 그룹을 형성하는 효율적인 클러스터링 알고리듬을 제시한다. 본 연구의 특징은 SOFMs의 구조와 파라미터 연구에 있다. 제안하는 SOFMs의 구조는 1차원 선형구조이다. 출력노드의 개수는 입력노드의 2배로 설정하였다. 이와 같은 출력노드에 연결강도를 임의로 부여하고, 출력노드별 연결강도 합의 크기에 따라 오름차순으로 정렬하였다. 학습이 진행되고 이웃노드의 범위가 초기 이웃노드 범위의 절반(1/2)이 되는 지점에서 승자노드가 되지 못한 출력노드의 연결강도는 조절하지 않는다. 학습이 완료된 입력데이터가 출력노드에 유사한 순서대로 정렬되면 출력노드 간의 연결강도 차이가 가장 큰 지점을 선형적으로 분리하여 원하는 그룹을 형성할 수 있다.

클러스터링 알고리듬은 초기 학습률 및 학습함수 그리고 이웃범위와 같은 매개변수를 어떻게 결정하느냐에 따라 성능이 좌우된다. 초기 학습률은 클수록 학습시간이 줄어든다. 본 연구에서는 초기 학습률을 0.4, 학습함수를 $\alpha(t)=(1-t/4950)\times\alpha(t-1)$로 하였다. 반면 출력노드의 개수와 이웃범위는 적을수록 학습시간이 짧아지나 입력데이터가 출력노드에 충분히 펼쳐질 수 있는 최소의 출력노드 개수 및 이웃범위를 설정하는 것이 매우 중요하다. 경험적 방법에 의하면 출력노드 개수는 입력데이터 개수의 두 배 이상으로 정하면 학습과정에서 발생할 수 있는 결함을 안전하게 극복할 수 있는 가장 좋은 해를 구할 수 있음이 나타났다. 이웃범위는 초기에는 모든 출력노드를 이웃으로 설정하고, 시간이 지남에 따라 그 범위를 줄여나가고 이웃범위가 자기 자신일 때 알고리듬을 중지한다.

본 연구에서는 잘 알려진 IRIS 데이터와 기계-부품 행렬을 가지고 실험하였다. IRIS 데이터를 가지고 실험한 결과 이제까지 알려진 기존의 무감독학습을 이용한 클러스터링 알고리듬이 15-17개의 오분류를 생성하는 데 비해 6개의 오분류를 생성함으로써 더 좋은 해를 구하였다. 기계-부품 행렬을 이용하여 기계그룹을 형성한 결과 모두 최적의 해를 구하였다. 제안하는 알고리듬은 복잡한 연산을 사용하지 않기 때문에 실시간으로 사용이 가능하며 변화하는 상황에 유연하게 적용할 수 있는 장점도 가지고 있다.

참고문헌

Aldenderfer, M. S., and Blashfield, R. K.(1984), *Cluster Analysis*, Saga Publications, London.

Anderson, E.(1939), The IRIS's of the Gaspe Peninsula, *Bull. Amer. IRIS Soc*., **59**, 2-5.

Berry, M. J. A. and Linoff, G.(1997), *Data Mining Techniques for Marketing, Sales, and Customer Support*, John Wiley & Sons, New York.

Berry, M. J. A. and Linoff, G. S.(2004), *Data Mining Techniques for Marketing, Sales, and Customer Relationship Management*, John Wiley & Sons, New York.

Bezdek, J. C.(1981), Pattern Recognition with Fuzzy Objective Function Algorithms, *Plenum*, New York.

Chan, H. M. and Milner, D. A.(1982), Direct clustering algorithm for group formation in cellular manufacturing, *Journal of Manufacturing Systems*, **1**(1), 65-75.

Chandrasekharan, M. P. and Rajagopolan, R.(1987), ZODIAC: An algorithm for concurrent format of part-families and machine-cells, *International Journal of Production Research*, **25**(6), 835-850.

Chandrasekharan, M. P. and Rajagopolan, R.(1989), Groupability: An analysis of the properties of binary data matrices for group technology, *International Journal of Production Research*, **27**(6), 1035-1052.

Everitt, B. S.(1993), Cluster Analysis, Edward Arnold, London.

Everitt, B. S., Laudau, S., and Leese, M.(2001), *Cluster Analysis*, Edward Arnold, London.

Huntsberger, T. L. and Ajjimarangsee, P.(1990), Parallel Self-Organizing Feature Maps for Unsupervised Pattern Recognition, *International Journal of General Systems*, **16**(4), 357-372.

Table 5. The number of error comparison for Machine-Part Incidence Matrix

Size (No of Machine × No of Part)	Source of Problems	No of Group		No of Error	
		Optimal	Proposed Algorithm	Optimal	Proposed Algorithm
4×5	Kusiak(2000)	2	2	0	0
5×7	King *et al*.(1982)	2	2	0	0
7×8	Kusiak *et al*.(1987)	3	3	0	0
10×15	Chan *et al*.(1982)	3	3	0	0
10×20	Srinivasan *et al*.(1990)	4	4	0	0
24×40	Chandraseharan *et al*.(1989)	7	7	0	0
40×100	Chandraseharan *et al*.(1987)	10	10	0	0

Karayiannis, N. B.(1997), A Methodology for Constructing Fuzzy Algorithms for Learning Vector Quantization, *IEEE Trans. Neural Networks*, **8**(3), 505-518.

King, J. R. and Nakornchai, V.(1982), Machine-component group formation in group technology: Review and extension, *International Journal of Production Research*, **20**(2), 117-133.

Kohonen, T.(1984), *Self-Organization and Associative Memory*, Springer, Berlin.

Kohonen, T.(1988), *Self-Organization and Associative Memory*, Springer, Berlin.

Kohonen, T.(1997), *Self-Organizing Maps*, Springer, Berlin.

Kusiak, A.(2000), *Computational Intelligence in Design and Manufacturing*, John Wiley & Sons, New York.

Kusiak, A. and Chow, W. S.(1987), Efficient solving of the group technology problem, *Journal of Manufacturing Systems*, **6**(2), 117-124.

Mangiameli, P., Chen, S. K., and West, D.(1996), A Comparison of SOM Neural Network and Hierarchical Clustering Methods, *European Journal of Operational Research*, **93**(2), 402-407.

Pal, N. N., Bezdek, J. C., and Tasao, E. C. K.(1993), Generalized Clustering Networks and Kohonen's Self-Organizing Scheme, *IEEE Trans. Neural Networks*, **4**(4), 549-551.

Tasao, E. C. K., Bezdek, J. C., and Pal, N. N.(1994), Fuzzy Kohonen Clustering Networks, *Pattern Recognition*, **27**(5), 754-757.