

Hopfield 네트워크를 이용한 데이터 클러스터링

윤면희 정근락
홍익대학교 전자계산학과
{mhyoon, chong}@cs.hongik.ac.kr

Data Clustering Using Hopfield Network

Myun-Hee Youn Kyun-Rak Chong
Dept. of Computer Science, Hongik University

요 약

데이터 클러스터링은 서로 유사한 성질을 갖는 데이터들은 동일한 클러스터에 분류하고, 이질적인 데이터는 다른 클러스터에 분류하여, 클러스터 내의 유사성은 최대한 하고 클러스터와 클러스터사이의 유사성을 최소로 하는 것을 말한다. 데이터 클러스터링은 데이터 마이닝, 기계 학습, 패턴 인식, 통계 분야 등에 다양하게 활용되고 있다. Hopfield 네트워크는 조합적 최적화 문제를 해결하는 데 사용되어 좋은 결과를 나타내고 있다. 본 논문에서는 Hopfield 네트워크를 사용하여 데이터 클러스터링 문제를 해결하는 알고리즘을 연구하였고, 실험을 통해 기존의 방법과 비교하였다.

1. 서론

데이터 클러스터링은 서로 유사한 성질을 갖는 데이터들은 동일한 클러스터에 분류하고, 이질적인 데이터는 다른 클러스터에 분류하여, 클러스터 내의 유사성은 최대한 하고 클러스터와 클러스터사이의 유사성을 최소로 하는 것을 말한다. 클러스터링은 데이터 마이닝, 인공 지능, 통계학, 패턴 인식, 이미지 프로세싱 등에서 널리 연구를 해오고 있다.

클러스터링 알고리즘은 접근 방식에 따라 밀도(density), 분할(partitioning), 계층적(hierarchical) 알고리즘들로 나눌 수 있다 [10]. 밀도를 기반으로 하는 알고리즘들은 공간적으로 비슷한 위치에 있는 데이터는 같은 클러스터에 할당하는 데, 이를 위해 이행성(transitivity)과 같은 근접한 포인트들에 대한 정보를 사용한다. 이 방식은 DBSCAN [5] 등이 있다. 분할 알고리즘은 전체 데이터를 미리 정해진 k 개의 클러스터로 나누는 데, 각 클러스터는 중심 또는 중심 근처의 좌표 값으로 표현되고, 주어진 목적 함수(objective function)를 최소화하기 위해 각 클러스터의 중심을 반복적으로 수정한다. K-means 알고리즘 [1], CLARA [3], CLARANS [4] 등이 분할 알고리즘에 속한다. K. Alsabti는 [10]에서 $k-d$ 트리를 사용해서 K-means 알고리즘의 실행 시간을 개선하였고, 유전자 알고리즘을 사용한 클러스터링 방법도 제안되었다. 계층적 알고리즘은 전체 데이터를 계층적으로 분해하면서 만든다. 대표적인 예는 트리로 표현하는 것인데 트리를 구성하는 방식은 상향식과 하향식이 있다. 예로서 GRIDCLUS [6], BIRCH [7] 등이 있다.

Hopfield 네트워크는 TSP, 모듈 방향 결정 문제와 같은 조합적 최적화 문제(combinatorial optimization problem)에 사용되어 좋은 결과를 나타내고 있다 [8]. Hopfield 네트워크는 에너지 함수가 최소가 되는 점(local minima)으로 수렴하는 성질을 가지고 있으며, 에너지 함수는 주어진 문제의 제약 조건을 나타내는 항과 목적 함수를 최적으로 하는 항으로 표현할 수 있다. 또 이들 항 사이에는 가중 값(weight)이 주어지며, 이 값에 따라 해와 수렴속도가 달라진다.

본 논문에서는 Hopfield 네트워크를 사용하여 데이터 클러스터링 문제를 해결하는 알고리즘을 연구하였고, 실험을 통해 K-means 방법과 비교하였다.

2. Hopfield 네트워크

1943년에 McCulloch 와 Pitts는 두뇌를 계산 조직으로 보고, 각 뉴런들의 동작과 상호 연결에 대한 모델을 제시하여 매우 간단한 구조를 가진 많은 뉴런들이 적당히 연결되면 상당한 계산 능력을 가지게 됨을 밝혀 냈다. Hopfield와 Tank는 에너지 함수의 최소 점에 수렴하는 성질을 가진 신경 회로망을 발견하였으며 TSP를 이 모델을 이용하여 풀 수 있다는 것을 보였다 [8]. Hopfield 네트워크에 대한 설명은 [8][9] 등에 자세히 기술되어 있으므로 여기서는 그 방법만 간단히 언급하고자 한다.

Hopfield 신경 회로망 모델은 모든 처리 요소(processing element)들이 상호 연결된 구조를 갖는다. 하나의 뉴런에 해당되는 각 처리 요소들은, 입력 u_i 를 가지며, 출력 $v_i=g(u_i)$ 는 단

조 증가하는 함수 g 에 의하여 구해진다.

Hopfield 네트워크는 에너지 함수의 값이 감소하는 방향으로 상태 변화하는 성질이 있는데, 아래의 에너지 함수 값이 지역해가 되는 상태가 신경 회로망의 안정 상태(stable state)가 된다. 신경회로망을 통해 최적화 문제를 풀기 위해서는 다음과 같은 형태로 에너지 함수가 표현되어야 한다.

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij} v_i v_j - \sum_i I_i v_i \quad (1)$$

여기서 n 은 뉴런들의 개수이고, v_i 는 뉴런들의 값이고 T_{ij} 는 뉴런 i 와 뉴런 j 의 연결 강도이고 I_i 는 외부 입력 벡터이다.

신경 회로망을 이용하여 최적화 문제를 풀기 위해서는 먼저 해들이 이진수로 표현 가능하여야 한다. 만약 이진 표현이 n 자리를 요구한다면 신경 회로망 내에는 n 개의 뉴런이 필요하게 된다. 다음에는 신경 회로망의 상태가 지역 해에 도달했을 때 각 뉴런들의 값이 가능한 해(feasible solution)를 이루도록 (1)식의 형태로 에너지 함수를 만들어야 하는데, 이 에너지 함수는 최적에 가까운 해에 대해서는 작은 값을 가져야 하며 최적해로부터 멀어질수록 큰 값을 가져야 한다. 이렇게 고안한 에너지 함수로부터 연결 행렬 T 와 외부 입력 벡터 I 를 구하면 우리가 원하는 신경 회로망을 구성할 수 있다.

3. 데이터 클러스터링 문제에 적용

먼저 데이터 클러스터링 방법 중에서 k -means 알고리즘에 대해 간단히 언급하기로 한다. 먼저 고정된 클러스터들의 개수 k 가 주어진다. 그리고, 각 클러스터의 추정된 중심으로 k 개의 임의의 프로토타입(w_1, w_2, \dots, w_k)가 주어진다. 각 입력 패턴(i_1, i_2, \dots, i_n)은 이 중 거리가 가장 가까운 프로토타입에 할당된다. 그 다음 각 클러스터에 속해 있는 입력 패턴들의 평균을 구하여 그 클러스터의 새로운 프로토타입으로 설정하고 위 단계를 반복하게 된다. 보통 클러스터링의 수준은 다음의 오류함수(error function)로 측정된다.

$$E = \sum_{j=1}^k \sum_{i \in C_j} |i_j - w_j|^2 \quad (2)$$

여기서 $|i_j - w_j|$ 는 입력 패턴 i_j 과 클러스터 중심 w_j 사이의 거리를 의미하며, 본 논문에서는 유클리디안 거리를 사용하였다.

Hopfield 네트워크를 데이터 클러스터링 문제에 적용하려면 먼저 뉴런의 개수를 결정하여야 한다. v_i 를 입력 패턴 x 가 클러스터 i 에 속한 것을 나타내는 뉴런이라 정의하면 신경 회로망은 전체 nk 개의 뉴런으로 이루어진다. 즉 신경회로망이 평형 상태에 도달하면 클러스터 당 정확하게 하나의 뉴런만이 "1"의 값을 가져야 하며 나머지 $k-1$ 개의 뉴런들은 "0"의 값을 가지게 된다.

에너지 함수는 다음의 두 가지 특성을 가지도록 만들어져야 한다.

- (1) 에너지 함수가 지역적 최소해(local minima)에 도달했을 때 신경 회로망의 각 클러스터에는 정확히 하나의 뉴런만이 "1"의 값을 가지며 나머지 뉴런들은 "0"의 값을 가져야 한다.
- (2) 신경 회로망의 상태에 따른 에너지 함수의 값은 입력 패턴과 해당 클러스터 사이의 길이 제곱의 합에 비례한다. 따라서 에너지 함수 E 를 다음과 같이 만들 수 있다. 함수

$$E_1 = \sum_x \sum_i \sum_j V_{xi} V_{xj} \quad (3)$$

은 각 뉴런의 값이 "0" 혹은 "1"일 때, 그리고 각 클러스터가 "1"인 뉴런 하나와 "0"인 뉴런 $k-1$ 개로 이루어질 때 최소가 된다. 함수

$$E_2 = [(\sum_x \sum_i V_{xi}) - n]^2 \quad (4)$$

는 신경 회로망 내에 총 n 개의 "1"이 있을 때 최소가 된다. 따라서 식 (2) + (3)은 각 클러스터 당 정확하게 하나의 뉴런만이 "1"의 값을 가질 때 최소가 된다.

d_{ij} 를 입력 패턴 x 와 해당 클러스터 i 사이의 길이라 하면 식 (1)은 다음 식과 같이 표시될 수 있으며, 각 입력 패턴과 해당 클러스터 사이의 길이의 제곱의 합이 최소가 될 때 최소 값을 가지게 된다.

$$E_3 = \sum_x \sum_i V_{xi} d_{xi}^2$$

이론적으로 시스템이 수렴하면 V_{xi} 는 0 또는 1이 되므로

$$E_3 = \sum_x \sum_i V_{xi}^2 d_{xi}^2 \quad (5)$$

을 사용하기로 한다. 따라서 최소화시켜야 할 에너지 함수는

$$E = \frac{1}{2} \alpha E_1 + \frac{1}{2} \beta E_2 + \frac{1}{2} \gamma E_3 \quad (6)$$

이며, 이때 α, β, γ 는 양의 실수 값을 가지는 에너지 가중치이다

(6) 식을 (1) 식의 형태로 바꾸기 위해서는 행렬 T 와 벡터 I 의 값을 결정해야 한다. δ_{ij} 를 $i=j$ 인 경우 $\delta_{ij}=1$, $i \neq j$ 인 경우 $\delta_{ij}=0$ 이 되는 Kronecker delta라 정의하자. 그러면

$$T_{xi,yj} = -\alpha d_{xy}(1 - \delta_{ij}) - \beta - \gamma d_{xy} d_{ij} d_{xi} d_{xj}$$

$$I_{xi,yj} = \frac{1}{2} \beta n$$

로 표현된다. 여기서 (1)식의 i 가 (x, i) 에 해당되고 j 가 (y, j) 에 해당한다.

이 때 중요한 문제점 중의 하나는 에너지 가중치 α, β, γ 의 값을 어떻게 결정하느냐이다. α, β 의 값이 γ 에 비해 너무 작으면 가능한 해를 찾을 수는 있으나 최적해로부터 너무 멀어지는 경향이 있으며, γ 가 너무 커지면 가능한 해를 찾기가 어렵다. 본 논문은 Hopfield 네트워크에서의 spurious 상태(infeasible solution)들을 억제하기 위해 Shigeo Abe에 제안된 방법을 이용하여 가능한 해를 보장하는 범위 하에서 에너지 가중치를 결정하였다 [2].

임의의 상태에서의 각 뉴런들은 $v = (v_1, \dots, v_n)$, $0 \leq v_i \leq 1$, $i = 1, \dots, n$ 로 표현될 수 있다. n 차 hypercube H 를 $H = \{v \mid 0 \leq v_i \leq 1, i = 1, \dots, n\}$ 라 정의하면 H 는 뉴런들의 가능한 모든 상태들을 나타낸다. H 에 속하는 벡터 $v = (v_1, \dots, v_n)$ 에 인접한 벡터 $v(i)$ 를 $v(i) = (v_1, \dots, v_{i-1}, 1-v_i, v_{i+1}, \dots, v_n)$ 으로 정의하고 v 와 $v(i)$ 사이에 존재하는 벡터 x 를

$$x = \mu v + (1 - \mu)v(i), 0 \leq \mu \leq 1 \quad (7)$$

이라 정의하자.

벡터 v 와 $v(i)$ 에 인접한 벡터 $v(i)$ 에서 계산된 에너지 함수 값을 각각 $E_v, E_{v(i)}$ 라 했을 때 만약 $E_v = E_{v(i)}$ 가 성립하면 (7)식에 의해 주어진 값들을 뉴런이 가질 때 Hopfield 네트워크는 안정 상태가 될 것이다. 만약 $E_v < E_{v(i)}$ 가 성립하면 Hopfield 네트워크의 뉴런들의 값은 x 에서 v 로 변화할 것이다. 모든 i 에 대해 $E_v < E_{v(i)}$ 가 성립할 때 벡터 v 는 강한 안정 상태에 있다고 정의한다. 정의에 의하면 v 가 가능한 해들 중의 하나이며 강한 안정 상태에 있게 되면 벡터 $v(i)$ 혹은 v 와 $v(i)$ 사이에 있는

모든 벡터 x 는 가능한 해 v 로 이동하게 된다. 따라서 모든 가능한 해들을 강한 안정 상태로 만들어 주면 그렇게 하지 않았을 경우 Hopfield 네트워크가 수렴해 갈지도 모르는 가능한 해에 인접한 spurious 상태들을 억제할 수 있다.

가능한 해에 인접한 spurious 상태들을 억제하기 위해 임의의 가능한 해를 가정하고 입력 패턴 x 의 클러스터 j 에서 한 뉴런의 상태를 "1"에서 "0"으로 바꾼다. 이 때 (6)식의 에너지 함수 E 에서 첫 번째 항은 $\alpha/2$ 만큼, 두 번째 항은 $\beta/2$ 만큼 증가하고 세 번째 항은 $d_{xi}^2/2$ 만큼 감소하게 된다. 따라서 만약

$$\alpha + \beta > \gamma \max_{x,i} \{d_{x,i}^2\}$$

이 성립하게 되면 가능한 해들은 강한 안정 상태가 된다. 이와 같이 에너지 가중치 α, β, γ 의 값은 입력 패턴과 클러스터 사이의 거리에 의존되어 정해져야 한다.

4. 실험 및 결과

실험을 위해 제안된 시뮬레이터는 다음과 같은 과정을 실행한다.

```
do {
  for every i
    vi = g( ui )
  for every i {
    Δ ui = Δ t ( ∑ Tijvj - ui/R + Ii )
    ui = ui + Δ ui
  }
} while (network not converged);
```

여기서 g 는 sigmoid 함수로서 $g(u) = (1 + \tanh(\lambda u))/2$ 을 사용하였고 R 은 회로 저항에 관계된 변수로 1을 사용하였다.

입력 벡터 v 의 초기 값을 어떻게 정하느냐에 따라 평형 상태로의 수렴 속도와 해가 달라질 수 있다. 평균 수렴 속도를 높이기 위하여 본 논문에서는 kn 차 hypercube를 가정하고 v 의 초기 값이 중심 근처에 있도록 다음과 같이 주었다.

$$v_i = 0.5 + \epsilon \times (\text{RAND} - 0.5)$$

이때 RAND에 의해 발생하는 난수는 0과 1사이에서 균일하게 분포한다. ϵ 은 1보다 작은 수를 사용하게 되는데 본 논문에서는 0.01의 값을 사용하였다.

신경 회로망을 사용했을 때 해의 정확성과 수렴 속도는 에너지 함수, Δt 와 λ 에 밀접한 관계를 가지고 있다. Δt 가 작을수록 일반적으로 좀더 정확한 해를 구할 수 있으나 수렴하는데 걸리는 시간이 많아지므로 본 연구에서는 Δt 의 값을 0.0001로 고정시켰다. λ 의 값은 25로 주고 서서히 $\Delta \lambda = 0.005$ 만큼 변화시켰다.

본 논문에서는 k-means 알고리즘과 Hopfield 네트워크를 이용한 알고리즘을 C로 구현하여 SUN 서버 엔터프라이즈 3000에서 실험하였다. 표 1에 그 결과가 나타나 있다. Hopfield 네트워크를 모의 실험할 때 필요한 T 행렬은 그 크기가 뉴런의 개수의 제곱에 비례하므로 데이터의 크기가 큰 경우에는 모의 실험하기 어려운 단점을 가지고 있지만, VLSI 하드웨어로 구현되면 데이터의 크기를 증가시킬 수 있고 실시간에 실행시킬 수 있다. 실험에서 입력 패턴의 수는 50, 100, 200, 500의 네 가

지 경우로 하였는데 분석적 방법인 K-means 방법과 Hopfield 네트워크는 비슷한 결과를 나타내었다.

<표1> Hopfield 네트워크와 k-means 방법의 비교

방식 \ 개수	50	100	200	500
Hopfield 네트워크	1163.90	1425.54	2383.71	13611.05
k-means	1163.90	1419.33	2407.61	13240.63

5. 결론

데이터 클러스터링은 데이터 마이닝, 기계 학습, 패턴 인식 등 다양한 분야에서 사용되고 있으며, Hopfield 네트워크는 TSP와 같은 조합적 최적화 문제에 적용되어 좋은 결과를 나타내고 있다. 본 논문에서는 Hopfield 네트워크를 사용하여 데이터 클러스터링 문제를 해결하는 방법을 제시하였고, 실험을 통해 기존의 k-means 방법과 비교하였다.

6. 참고 문헌

- [1] R. C. Dubes and A. K. Jain, "Algorithms for Clustering Data", Prentice Hall, 1988
- [2] Shigeo Abe, "Global Convergence and Suppression of Spurious States of the Hopfield Neural Networks", Proc. IJCNN-91, Vol. 2, Nov. 1991.
- [3] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: an Introduction to Cluster Analysis", John Wiley & Sons, 1990
- [4] R. T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. of the 20th Int'l Conf. on very Large Databases, Santiago, Chile, 1994, pages 144-155
- [5] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clustering Clusters in Large Spatial Databases with Noise", Proc1996
- [6] E. Schikuta, "Grid Clustering : An Efficient Hierarchical Clustering Method for Very Large Data Sets", Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, 1996, pp. 226-231.
- [7] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH : An Efficient Data Clustering Method for Very Large Database", In Proc. of ACM SIGMOD International Conference on Management of Data, 1996.
- [8] J. J. Hopfield and D.W. Tank, "Neural Computation of Decision in Optimization Problem," Biol. Cybern. vol. 52, 1985
- [9] J. Freeman and D. Skapura, Neural Networks, Addison-Wesley Publishing Company, 1991
- [10] K. Alsabti, S. Ranka, and V. Singh, "An Efficient K-Means Clustering Algorithm", IPPS/SPDP Workshop on High Performance Data Mining, 1998, Orlando, Florida