데이터 마이닝을 위한 LVQ 기반 신경 트리 분류기

김세현⁰ 김은주 이일병
연세대학교 컴퓨터과학과
{astaroth, outframe, yblee}@csai.yonsei.ac.kr

Neural Tree Classifier based on LVQ for Data Mining

Sehyun Kim⁰ Eun-Ju Kim Yillbyung Lee
Dept. of Computer Science, Yonsei University

요 약

신경 트리는 신경망과 결정 트리의 구조를 결합한 형태의 분류기로서 비선형적 결정 경계 형성이 가능하며 기존 신경망에 비해 학습, 출력시 계산량이 적다는 장점을 갖는다. 본 논문에서는 신경 트리의 노드를 구성하는 신경망을 학습하기 위하여 기존의 방법들과는 달리 교사 학습 방법인 LVQ3 알고리즘을 사용하는 신경 트리 분류기를 제안한다. 학습 과정을 통해 생성된 트리는 오인식율 추정을 이용한 가지치기를 통하여 효율적인 트리로 재구성된다. 제안하는 방법은 실제 데이터 집합들을 이용한 실험을 통하여 그 성능을 검증하였다.

1. 서론

신경 트리(Neural Tree)는 신경망(Neural Network)과 결정 트리(Decision Tree)를 결합하여 특성이 우수한 분류기를 만들려는 노력에서 나온 시도 중 한 가지 방법이다[6]. 신경 트리는 결정 경계가 신경망에 의해 형성되므로 벡터 공간의 축에 수직일 필요가 없으며, 결정 트리와 같은 divide & conquer의 하향 처리 방식으로 문제를 분할해 나가므로 큰 문제를 한꺼번에 푸는 방식인 기존 신경망보다 학습 속도가 우수하다. 또한 기존 신경망은 학습하기에 앞서 신경망의 구조를 결정해야 한다는 문제점이 있지만 신경 트리는 학습하는 과정에서 트리가 성장하면서 구조적응 과정을 거치므로, 구조를 미리 결정할 필요가 없다는 장점을 갖는다.

본 논문에서 제안하는 분류기인 LVQ 기반 신경 트리(Lvq-based Neural Tree ; LNT)는 경쟁식 신경망으로 구성된 신경 트리 분류기로, 트리의 노드 안의 경쟁식 신경망을 LVQ3[1][15] 알고리즘으로 학습하는 것이 특징이다. LVQ3 알고리즘은 비교사학습 경쟁식 신경망인 자기조직화지도(Self-Organizing Map)[1][15]에 분류 성능을 높이기 위한 것으로 소개되었으나, 단독으로 경쟁식 신경망을 학습하는 데에 사용되기도 한다. 또한 LNT는 신경 트리에서 흔히 발생되는 과학습(overtraining) 문제를 해결하기 위해 트리의 성장이 끝난 후에 C4.5 시스템에서 구현된 오류 기반 가지치기 방법(Error-Based Pruning ; EBP)[2]을 사용하여 가지치기한다.

본 논문의 구성은 다음과 같다. 2장에서는 신경 트리와 관련된 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 LNT에 대해 설명한다. 4장에서는 공개된 데이터에 대한 실험을 통해 다른 분류기들과의 성능을 비교하며, 5장에서 결론을 맺는다.

2. 관련연구

Sankar 등은 출력 뉴런이 한 개인 단층 feed forward 신경망을 트리 구조로 배치한 신경 트리를 제안하였다[3]. 이 신경 트리는 트리의 성장이 끝난 후에 CART[4]의 가지치기 방법인 CCP(Cost-Complexity Pruning)를 사용하여 효율적인 트리를 구성하도록 하였다.

Li 등의 신경 트리는 경쟁식 신경망을 사용한 신경 트리의 초기 연구로서, 노드의 분류 성능을 트리의 성장 기준으로 한 구조적응 분류기이다[5]. Karrayiannis 등은 경쟁식 신경 트리 분류기에서 다양한 트리 탐색 방법을 제안함으로써 계산량을 크게 증가시키지 않으면서도 신경 트리의 결정 경계 형성의 유연성을 증대하였다[6].

Song 등은 대용량 데이터 학습에 알맞은 경쟁식 신경 트리 알고리즘을 제안하였다[7]. 이 신경 트리는 비교사 경쟁 학습을 하면서 승자 노드에 누적되는 distortion error의 크기를 기준으로 성장시킴으로써, 벡터 공간에서 많은 데이터가 존재하는 영역에 복잡한 결정 경계 형성이 가능하게 하였다.

Waizumi 등이 제안한 HLVQ[10]는 경쟁식 신경 트리의 학습에 LVQ2.1[1][15]을 도입한 신경 트리이다. 이것은 기본적으로 문자 인식에서의 대분류를 위해 사용된 것으로, 각 노드에서 승자가 하나 이상일 수 있도록 하여 greedy 탐색의 한계를 극복하려 하였다.

신경 트리를 이용한 계층적 군집화 방법도 제안되었다. Adams 등이 제안한 알고리즘[8][9]은 트리 성장에 의한 구조 적응은 물론 노드 안의 경쟁식 신경망도 구조적응을 할 수 있도록 한 것이다.

3. LVQ based Neural Tree(LNT)

본 논문에서 제안하는 LNT는 경쟁식 신경망을 사용한 신경 트리로서, 하향식 접근 방법을 사용하여 데이터를 분할 및 학습함으로써 신경망의 가중치인 코드북 벡터(codebook vector)를 트리 형태로 구성한다. 경쟁식 신경망은 LVQ3[1][15] 알고리즘으로 학습하며, 트리의 성장이 완전히 끝나면 EBP[2] 방법으로 가지치기한다.

3.1 분류기 구조

LNT의 한 노드 안에는 n개의 입력 뉴런과 m개의 출력 뉴런으로 구성된 단일 계층의 경쟁식 신경망이 존재한다. 그리고 각 출력 뉴런에는 자식 노드가 할당될 수 있는 구조이다.

노드에 입력 벡터가 들어오면 출력 뉴런들이 경쟁을 벌여 최근접 가중치 벡터를 갖는 뉴런이 승자로 결정된다. 승자 뉴런에 자식 노드가 할당되어 있으면 입력 벡터는 그 자식 노드로 다시 입력된다. 출력 뉴런에 자식이 할당되어 있지 않다면 그 뉴런의 클래스를 출력한다. 결과적으로 가중치 벡터는 코드북 벡터로 간주되어, 경쟁식 신경망은 코드북 벡터를 중심으로 한 Voronoi 영역으로 구성되는 결정 경계를 형성한다.

LNT의 구체적인 구조는 그림 1과 같다.

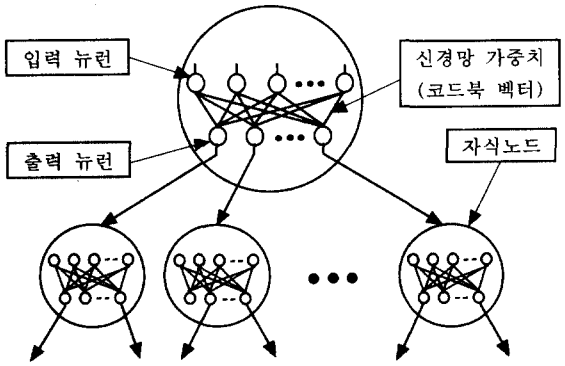

그림 1. LNT의 구조

3.2 학습방법

3.2.1 노드 내의 신경망 학습

학습은 근 노드로만 구성된 트리에서부터 시작하며, 각 노드별로 이루어진다. 노드에 학습 데이터가 주어지면, 먼저 경쟁식 신경망의 가중치를 초기화한다. LVQ3을 위한 신경망의 초기화는 무작위로 학습 데이터에서 선택하는 방법도 가능하지만 본 논문에서는 비교사학습인 (unlabeled) LVQ[15]를 사용하였다. 신경망의 초기화가 끝나면 학습 데이터 전체를 신경망에 입력하여 각각의 코드북 벡터에 속하는 학습 벡터들의 다수결 방법으로 코드북 벡터의 클래스를 결정한 후, 다음과 같은 LVQ3 알고리즘[1][15]으로 학습한다. 먼저 입력 벡터 $\boldsymbol{x}$에 대해서 두 개의 승자 뉴런의 가중치 벡터 $\boldsymbol{v}_i$와 $\boldsymbol{v}_j$를 찾는다. 가중치 벡터는 다음 두 가지 경우에만 갱신된다.

1. $\boldsymbol{x}$의 클래스가 $\boldsymbol{v}_i$와는 다르고 $\boldsymbol{v}_j$와 같은 경우

$$\boldsymbol{v}_i^{new} = \boldsymbol{v}_i^{old} - \alpha(\boldsymbol{x} - \boldsymbol{v}_i^{old}) \quad (1a)$$

$$\boldsymbol{v}_j^{new} = \boldsymbol{v}_j^{old} + \alpha(\boldsymbol{x} - \boldsymbol{v}_j^{old}) \quad (1b)$$

이때 벡터 $\boldsymbol{x}$는 $\boldsymbol{v}_i$와 $\boldsymbol{v}_j$의 윈도우에 존재하여야 한다. 윈도우는 두 가중치 벡터를 코드북 벡터로 하여 형성되는 결정 경계의 부근 영역으로, $\boldsymbol{x}$가 윈도우 내부에 존재하는지의 여부는 $\boldsymbol{x}$와 $\boldsymbol{v}_i$, $\boldsymbol{v}_j$ 사이의 거리를 각각 d_i, d_j라고 하고 두 가중치 벡터 사이의 길이에 대한 윈도우의 상대적 폭을 w라고 할 때, 다음 식에 의해 결정된다.

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > \frac{1-w}{1+w} \quad (2)$$

2. $\boldsymbol{x}$의 클래스가 $\boldsymbol{v}_i$, $\boldsymbol{v}_j$와 같은 경우

$$\boldsymbol{v}_k^{new} = \boldsymbol{v}_k^{old} + \varepsilon\alpha(\boldsymbol{x} - \boldsymbol{v}_k^{old}), \quad k \in \{i, j\} \quad (3)$$

여기서 w는 0.2 정도의 값이 적당하고 ε은 0.1에서 0.5 사이의 값이면 가능하다.

3.2.2 트리의 성장

트리의 성장 조건은 Karrayiannis 등의 신경 트리[6]와 유사한 방법을 사용한다. 학습중인 노드의 신경망 학습이 종료되면, 각 출력 뉴런의 오인식율을 검사한다. 오인식율의 검사에는 학습에 사용되었던 데이터를 그대로 사용한다. 만약 어떤 출력 뉴런의 오인식율이 임계값 매개변수 β보다 크다면 학습이 더 필요하다고 판단하여 해당 출력 뉴런에 자식 노드를 생성한다. 따라서 매개변수 β를 변화시킴으로써 트리의 성장을 조절할 수 있다. β는 0과 0.5 사이의 값이 가능하다.

새로 자식 노드가 생성되면 그 출력 뉴런을 승자로 하는 데이터를 가지고 자식 노드를 학습한다. 그런데 데이터의 수가 너무 적을 경우에는 자식 노드를 만들지 않고 이 부분의 학습을 종료한다. 모든 노드가 학습을 종료할 때까지 계속한다.

3.2.3 트리의 가지치기

트리의 모든 노드의 성장이 끝나면 가지치기를 하게 된다. 가지치기는 결정 트리 분류 시스템인 C4.5에서 사용된 EBP[2]를 사용한다.

EBP는 학습 데이터에 대한 오인식을 바탕으로 학습에 쓰이지 않은 데이터에 대한 오인식율을 추정하여, 그 결과를 바탕으로 가지치기하는 방법이다. 각 노드에서, 그 노드를 근 노드로 하는 서브트리의 오인식 수와, 그 노드에서 데이터가 가장 많이 접근하는 가지로 원래의 노드를 대체하였을 경우의 오인식 수를 추정한다. 만약 가지로 대체하였을 경우의 오인식 추정값이 더 작다면 원래의 노드를 제거하고 그 가지로 대체한다.

오인식 추정 방법을 구체적으로 설명하면 다음과 같다. 한 개의 자식이 없는 출력 뉴런이 N개의 데이터에 의해 승자가 되고 그 중에서 E개의 데이터가 오인식이었다고 할 때, 이것을 N번의 시도(trial)에서 E번의 사건(event)이 발생하였다는 관점으로 볼 수 있다. 그러면 데이터에 대한 오인식의 분포는 이산 분포를 따른다고 할 수 있으며, 매개변수로 주어지는 신뢰 수준(confidence level)에 대해서 실제 오인식율의 신뢰 구간(confidence interval)의 상한, 하한값을 구할 수 있다. EBP는 이 신뢰 구간의 상한값을 택하여 오인식율로 추정한다.

이렇게 추정된 오인식율에 데이터의 수 N을 곱하여 오인식 수를 추정한다. 그리고 트리에서 자식이 없는 모든 출력 뉴런들의 오인식 추정값을 합하여 트리의 오인식 수를 추정해 낸다.

4. 실험

실험에 사용된 데이터는 UCI 기계학습 DB repository[1] [11]에 공개되어 있는 것이며, 실험은 n-fold cross-validation 방법으로 수행하였다. 실험에 사용된 데이터 중 Pima Indian Diabetes는 12 fold로, 나머지 데이터는 10 fold로 실험하였다.

4.1 실험 결과

데이터	속성 수	클래스 수	데이터 수	오인식률		
				LNT	LVQ	C4.5
Australian credit	14	2	690	**0.145**	0.197	0.155
Image segmentation	19	7	2310	0.049	0.046	**0.040**
Pima Indian D.	8	2	768	**0.246**	0.272	0.250
Br. cancer W.	9	2	699	**0.033**	0.034	0.053
Cleveland heart	13	2	303	**0.188**	0.392	0.258
Glass	9	6	214	0.299	0.393	**0.298**
Ionosphere	34	2	351	0.094	0.114	**0.084**

표 1. 공개된 데이터에 대한 LNT와 LVQ, C4.5의 분류 오인식률 결과.

표 1에서 LNT의 결과는 가지치기를 한 경우와 하지 않은 경우 중에서 더 좋은 결과를 택한 것이며, 다른 분류기의 결과는 [12][13][14]에서 참고하였다.

5. 결론 및 향후 연구 방향

본 논문에서는 LVQ3과 EBP를 적용한 경쟁식 신경 트리의 학습 방법인 LNT를 제안하였다. 기존 신경망은 분류 성능은 뛰어나지만 신경망의 구조를 미리 결정해야 하며 풀어야 할 문제가 커질수록 복잡한 구조를 필요로 하여 결과적으로 계산량이 매우 많아진다. 그러나 LNT는 트리 성장으로 구조 결정이 자동화되며, 비교적 단순한 구조의 신경망을 사용하여 주어진 문제를 하향 처리 방식으로 분할하므로 상대적으로 계산량이 적다는 장점이 있다. 그리고 생성된 트리 구조에 어느 정도 구조적인 정보가 있을 수 있다. 또한 LNT는 기존의 결정 트리에 비해 결정 경계가 비선형적으로 형성될 수 있다는 장점 이외에, 출력 뉴런의 코드북 벡터와 입력 벡터 사이의 거리를 기반으로 하여 결정 트리가 제공하지 못하는 신뢰도를 제공할 수 있다. 즉 LNT는 기존 신경망에는 트리 구조를 통한 어느 정도의 설명력 부여 효과를, 결정 트리에는 결정 경계의 비선형화 및 결과에 대한 신뢰도 제공 기능을 부여함으로써, 대용량 데이터를 학습하여 사용자의 의사 결정을 지원하는 데이터 마이닝 분야에 효과적으로 사용될 수 있을 것이라 생각된다.

향후 연구 계획으로, 트리 구조 자체를 변형시킴으로써 greedy 탐색이 갖는 한계를 극복하도록 할 것이다. 그리고 경쟁식 신경망의 학습 결과가 신경망의 가중치 초기화 방법에 영향을 받는데, 데이터에 따른 적절한 신경망 가중치 초기화 방법을 도입하여 신경망이 안정된 결과를 낼 수 있도록 할 예정이다.

1) 다음 URL에서 구할 수 있다.
http://www.ics.uci.edu/~mlearn/MLRepository.html

참고문헌

[1] T. Kohonen, "The Self-Organizing Map", Proceedings of the IEEE, vol. 78, no. 9, pp 1464-1480, 1990.

[2] J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, Calif.: Morgan Kaufmann, 1993.

[3] A. Sankar and R. J. Mammone, "Growing and Pruning Neural Tree Networks", IEEE Trans. Computers, vol42, no. 3, pp 291-299, 1993.

[4] L. Breiman, J. Friedman, R. Olshen and C. Stone, Classification and Regression Trees, Belmont, Calif.: Wadsworth Int'l, 1984.

[5] T. Li, L. Fang and Q. -Q. Li, "Hierarchical Classification and Vector Quantization with neural trees", Neurocomputing, 5, pp 119-139, 1993.

[6] S. Behnke and N. B. Karrayiannis, "Competitive Neural Trees for Pattern Classification", IEEE Trans. Neural Networks, vol 9, no. 6, pp 1352 -1369, 1998.

[7] H. -H. Song and S. -W. Lee, "A Self- Organizing Neural Tree for Large-set Pattern Classification", IEEE Trans. Neural Networks, vol 9, no. 3, pp 369-380, 1998

[8] R. G. Adams, K. Butchart and N. Davey, "Hierarchical Classification with a competitive evolutionary neural tree", Neural Networks 12, pp 541-551, 1999.

[9] N. Davey, R. G. Adams and S. J. George, "The Architecture and Performance of a Stochastic Competitive Evolutionary Neural Tree Network", Applied Intelligence 12, pp75-93, 2000.

[10] Y. Waizumi, N. Kato, K. Saruta and Y. Nemoto, "High Speed and High Accuracy Rough Classification for Handwritten Characters Using Hierarchical Learning Vector Quantization", IEICE Trans. Inf. & Syst., vol E83-D, no. 6, pp 1282-1290, 2000.

[11] P. M. Murphy and D. W. Aha, "UCI Repository of Machine Learning Databases [Machine Readable Data Repository]", Univ. of California, Dept of Information and Computer Science, Irvine, Calif., 1996.

[12] D. Michie, D. J. Spiegelhalter and C. C. Taylor (eds), Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.

[13] P. Eklund, A. Hoang, "A Performance Survey of Public Domain Machine Learning Algorithms", School of Information Technology, Griffith University. 1998.

[14] B. Ster and A. Dobnikar, "Neural networks in medical diagnosis: Comparison with other methods". In A. Bulsari et al., editor, Proceedings of the International Conference EANN '96, pages 427-430, 1996.

[15] T. Kohonen, Self-Organization and Associative memory 3rd ed., Berlin: Springer-Verlag, 1989.