비정상 행위 탐지를 위한 신경망 기반의 데이터 클러스터링

Data Clustering using a Neural Network for Anomaly Detection

김인영, 장병탁
서울대학교 컴퓨터공학부
{iykim, btzhang}@scai.snu.ac.kr

In-Young Kim and Byoung-Tak Zhang
Seoul National University

요 약

코호넨 자기조직 신경망을 사용하면 클러스터링뿐만 아니라 그 데이터가 할당된 클러스터의 대표값(Centroid)과의 거리 차이(Quantization Error)를 알아볼 수 있다. 이를 이용하면 어떤 데이터가 정상적인 분포를 따르는지 정상적인 분포에서 벗어나는 비정상적인 데이터인지 알 수 있고, 유닉스 시스템 사용자의 명령어 사용 패턴에 적용하여 어떤 사용자의 명령어 사용 패턴이 정상적인 것인지 비정상적인 것인지 알 수 있다. 본 논문에서는 유닉스 시스템 사용자 8명의 명령어 패턴을 클러스터링한 후 Quantization Error를 이용하여 비정상 패턴을 탐지하는 오프라인에서의 비정상 행위를 탐지하는 시스템을 구현하였다. 그리고 통계적인 학습 방법을 적용한 비정상 패턴 탐지와의 비교를 통하여 두 가지 비정상 패턴 탐지 결과가 동일함을 확인하였다.

Ⅰ. 서론

시스템 사용자의 행위를 관찰하고 분석하는 일은 시스템 관리자에게 많은 정보를 줄 수 있다. 예를 들어 각 사용자가 시스템에 로그온하여 사용한 명령어들을 보면 그 사용자의 명령어 사용 패턴을 알 수 있는데 이 데이터를 학습하여 사용자의 비정상적인 명령어 사용을 탐지할 수 있다. 이러한 연구는 오용 탐지(Misuse detection)와 비정상 탐지(Anomaly detection)로 구성되는 침입 탐지 시스템(IDS)에서의 비정상 탐지에 해당한다.

본 논문에서는 신경망과 통계적인 방법을 이용한 기계 학습 기법을 통해 시스템 사용자의 비정상적인 행위를 탐지하였다. 여기서 사용자의 비정상적인 행위라는 것은 그 사용자가 지금까지의 명령어 사용 패턴과는 다른 명령어 패턴을 보이는 것을 말한다. 비정상적인 명령어 패턴을 탐지하기 위해서 각 사용자의 명령어 패턴을 학습한 후 새로운 명령어 패턴에 대해 학습된 결과를 이용하여 정상적인 패턴인지 비정상적인 패턴인지 구별할 수 있다.

Ⅱ. 본론

1. 문제 정의 및 데이터 설명

본 실험에서 관심을 가지는 것은 유닉스 시스템 사용자의 행동을 분석하고 각 사용자마다의 특정한 행동 패턴, 즉 명령어 사용에 있어서의 패턴을 학습하는 것이다. 이것은 각 사용자마다 사용하는 명령어와 그 빈도수가 다른 것을 이용하여 사용자의 명령어 사용 패턴을 학습하려는 것이다.

실험에 사용한 데이터는 8명의 사용자를 대상으로 2년 동안 수집된 데이터이다. 이 데이터는 유닉스 시스템에서 각 사용자의 명령어 사용 기록(History)을 이용하였으며 각 세션(Session)마다 구분이 되어 있다. 하나의 세션은 한 번의 로그인부터 로그아웃 사이의 모든 명령어 사용을 저장하고 있다. 전부 9개의 집합(Set)으로 구성되어 있는 이 데이터는 각 사용자마다 하나의 집합을 유지하며 그 중 다른 시스템에서 다른 프로젝트를 수행하던 한 명의 사용자의 경우에 각각을 다른 집합에 저장하였다. 이 데이터에 관한 자세한 설명은 "http://kdd.ics.uci.edu/databases/UNIX_user_data/README"에서 찾을 수 있으며 데이터는 "http://kdd.ics.uci.edu/databases/UNIX_user_data/UNIX_user_data.html"에서 얻을 수 있다.

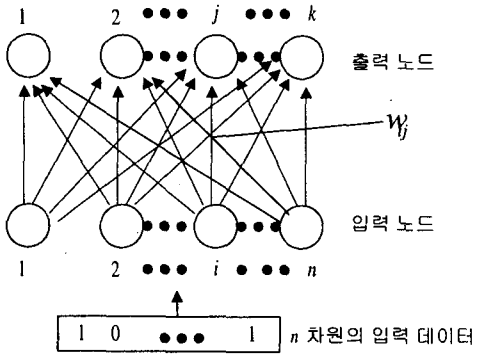
본 실험에서의 데이터 표현은 신경망 등의 학습에서 주로 사용하는 방법인 벡터 표현 방법을 사용하였다. 각 세션별로 하나의 벡터를 만들어 사용하였고 실험에 사용된 벡터는 100개의 명령어에 대한 사용 빈도수를 이용한 100차원의 벡터이다.

2. 학습 방법

사용자의 명령어 패턴을 학습하는 것은 기존의 명령어 패턴을 각 사용자의 프로파일(Profile)에 저장을 하고 새로운 패턴이 들어왔을 때 그 사용자의 프로파일과 비교하여 갱신하는 작업을 하는 것을 의미한다. 다시 말하면 사용자의 명령어 패턴이 점차적으로 변해가는 것을 프로파일에 반영하여 현재의 명령어 패턴을 잘 표현하도록 하는 것이다.

위의 학습 방법을 두 가지 다른 실험에 적용시켜 보았다. 첫 번째 실험은 신경망을 이용한 코호넨 자기조직 신경망(Self-Organization Map)에서의 클러스터링(Clustering)이며 두 번째 실험은 각 사용자마다 여러 개의 명령어 패턴을 유지하면서 프로파일을 갱신하는 과정을 보여주는 실험이다.

코호넨 자기조직 신경망을 이용한 클러스터링은 무감독(Unsupervised) 학습 방법으로 스스로 n차원의 입력 데이터들을 클러스터링하여 2차원 상에 표현시켜준다. 코호넨 자기조직 신경망은 신경망을 이용하여 구현한 것으로 그 내부는 아래의 그림과 같다.



[그림1] 코호넨 네트워크의 구조

[그림1]은 2-layer 신경망으로 n차원의 입력 데이터를 표현하는 n개의 입력 노드들과 k개의 분류 영역(Decision Region)을 표현하기 위한 k개의 출력 노드로 구성되어 있다. 모든 입력 노드들은 모든 출력 노드들과 연결되어 있고 연결 가중치(Weight)를 가진다.

초기 상태에서는 연결 가중치들을 임의로 할당한다. 임의의 연결 가중치를 할당한 후 유클리드 거리를 이용하여 입력 벡터와의 유사성을 측정한다. 입력 벡터와 k개의 연결 가중치 벡터사이의 유클리드 거리를 구하여 입력 벡터와 가장 유사한 j번째 연결 가중치 벡터를 찾으면 그 입력 벡터에 대해서 j번째 출력 노드가 승자가 된다. 이렇게 승자를 선택하면 승자의 연결 가중치 벡터는 [식1]과 같이 갱신된다.

$$w^j(t+1) = w^j(t) + \alpha(t)[x(t) - w^j(t)] \quad [식1]$$

$$\alpha(t) = 0.1(1 - t/10^4) \quad [식2]$$

$\alpha(t)$는 시간이 경과함에 따라 연결 가중치 벡터가 안정화되도록 하는 역할을 하는 파라미터이고 [식2]와 같이 표현된다. [식1]의 의미는 j번째 출력 노드가 승자가 되었으면 그 노드의 연결 가중치 벡터는 입력 데이터 쪽으로 약간 이동한다. 결과적으로 연결 가중치 벡터를 입력 데이터에 근사하도록 만들어 가는 것이다.

각 사용자마다 여러 개의 패턴을 프로파일에 저장하고 있으면서 새로운 패턴에 대해 학습하고 프로파일을 갱신하는 알고리즘은 아래와 같다.

사용자마다 최대 몇 개(M)의 패턴을 프로파일에 저장할 것인지 가정한다. 이 실험에서는 각 사용자마다 최대 10개의 패턴을 프로파일로 가진다고 정하였다. 즉, 각각의 사용자는 최대 10가지의 상이한 명령어 사용 패턴을 보일 수 있다.

i. M개의 패턴이 되기 전까지는 들어오는 패턴을 모두 프로파일에 저장한다.

ii. 프로파일에 저장된 패턴의 개수가 M개를 넘은 경우는 다음의 과정을 거친다.

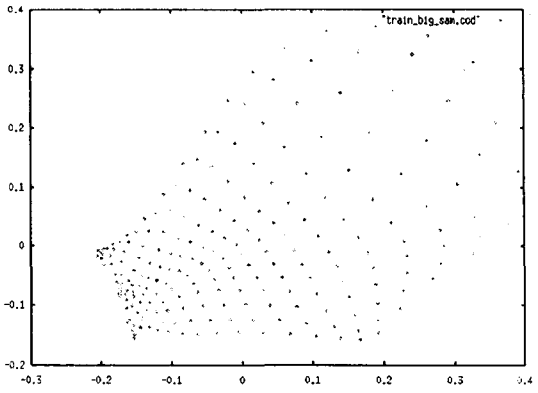
1. 새로 들어온 패턴을 포함한 $M+1$개의 패턴간 차이를 계산한다.
2. 두 패턴간 차이가 가장 작은 두 패턴을 통합한다.

위의 알고리즘을 통해서 M개의 상이한 패턴을 포함하는 프로파일을 유지할 수 있다. 새로 들어온 패턴이 프로파일에 속해 있는 어느 패턴과 유사한 패턴이라면 그 패턴과 함께 통합(Merge)될 것이며 전혀 새로운 패턴이라면 다른 유사한 패턴의 쌍 2개가 통합되면서 새로 들어온 패턴이 프로파일에 추가될 것이다. 이 때 벡터간의 유클리드 거리로 표현되는 패턴간 차이를 보면 새로 들어온 패턴이 기존에 학습된 프로파일에 있는 다른 패턴들과 얼마나 차이가 나는지 알 수 있다. 결국 사용자가 지금까지의 명령어 사용 패턴과는 다른 비정상적인 명령어 사용을 보임을 알 수 있다.

3. 실험 및 결과

실험은 위에서 제시한 두 가지 방법으로 수행하였다. 첫 번째로 코호넨 자기조직 신경망을 이용한 실험에서는 전체 사용자들의 명령어 패턴을 이용하여 클러스터링 하였다. 클러스터링 결과 [그림2]와 [그림3]을 통해 데이터들의 전체적인 분포를 살펴볼 수 있다. [그림2]에서 인접한 노드들은 서로 비슷한 유형의 패턴들이 이루는 클러스터들이고 거리가 멀고 명암의 차이가 많이 나는 노드들은 서로 다른 패턴을 나타내는 클러스터들이다. [그림3]에서는 n차원의 데이터 벡터로 표현되는 각각의 클러스터를 2차원으로 표현한 것으로 데이터들의 전체적인 분포와 밀집 정도를 볼 수 있다.

[그림2] 클러스터링 결과

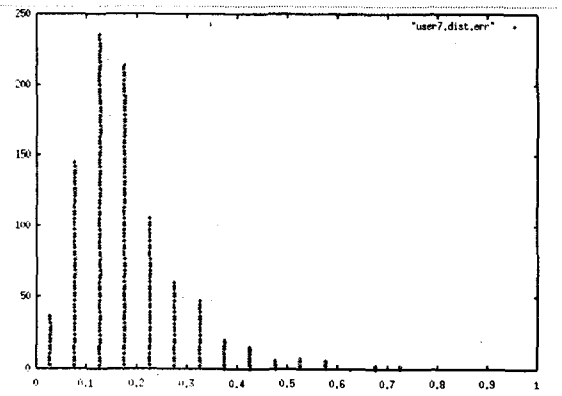


[그림3] 2차원상의 클러스터 분포

또한 각 사용자의 명령어 패턴의 분포를 알아보기 위해 각각의 사용자별 클러스터링도 하였다. 이 경우 한 명의 사용자로부터 나온 여러 가지 명령어 패턴에 대해서 클러스터링하며 각 클러스터를 대표하는 Centroid 벡터값과 해당 클러스터에 속한 각각의 데이터와의 차이를 표현하는 Quantization Error값을 통해서 비정상적인 패턴을 탐지할 수 있다.

두 번째 실험에서는 통계적인 방법을 이용하여 사용자의 명령어 패턴 학습 및 비정상적인 명령어 사용을 탐지한다. 앞에서 설명한 알고리즘대로 각 사용자마다 최대 M개의 명령어 패턴을 포함하는 프로파일을 유지하고 있다. 새로운 명령어 패턴이 주어지면 프로파일에 속해 있는 다른 명령어 패턴과의 유클리드 거리를 측정하여 그 차이가 Threshold θ 이상이면 비정상 패턴으로 간주한다. 그 후 새로 들어온 패턴을 포함한 $M+1$개 패턴 사이의 유사성을 비교하여 가장 유사한 두 개의 패턴은 하나의 패턴으로 통합한다. 그러면 다시 전체적으로 M개의 패턴이 유지되면서 새로운 패턴에 대한 학습이 끝나게 된다.

[그림4]는 Threshold θ를 정하기 위해서 새로 들어온 명령어 패턴 벡터와 프로파일내의 M개의 벡터들과의 거리 중 최소 거리를 도수분포표로 나타낸 것이다. 아래 그림에서의 사용자는 새로 들어온 명령어 패턴이 기존의 패턴과 평균적으로 유클리드 거리 0.25의 차이가 나며 분산은 0.15라는 것을 알 수 있는데 만약 새로 들어온 패턴의 최소 거리가 θ 이상이면 비정상적인 명령어 사용이라고 탐지한다.



[그림4] Mean : 0.25, Variance : 0.15

Ⅲ. 결론

신경망에 근거한 방법과 통계적인 방법에 근거한 두 가지 실험으로 확인할 수 있었던 것은 두 실험 모두 비정상적인 명령어 사용을 탐지하는 데 효과적이라는 것이다. 즉, 기존에 학습된 프로파일과 상이한 패턴이 들어오면 코호넨 자기조직 신경망의 경우 Quantization Error값이 정상 분포에서 벗어나는 것을 알 수 있었고 마찬가지로 통계적으로도 프로파일 내의 다른 패턴들과의 최소 거리가 정상 분포에서 벗어난 큰 값을 가지는 것을 확인할 수 있었다.

지금까지의 실험에서는 사용자의 명령어 패턴을 명령어 사용 빈도수만을 이용하여 학습하였으나 사용된 명령어들 사이의 순서를 학습한다면 온라인으로 비정상적인 명령어 패턴을 탐지할 수 있을 것이며 더욱 효과적인 방법이 될 것이다.

감사의 글

본 연구는 학술진흥재단 자유공모과제 (1999-001-E01025)에 의해 지원되었음.

Ⅳ. 참고 문헌

[1] Jake Ryan, Meng-Jang Lin, and Risto Miikkulainen. Intrusion Detection with Neural Networks. *Proceedings of NIPS-98*, pp. 943-949, 1998.

[2] Michael Chester. *Neural Networks: A Tutorial*, Prentice Hall. pp. 42-49, 1993.

[3] Simon Haykin. *Neural Networks: A Comprehensive Foundation*, Prentice Hall. pp. 443-483, 1999.

[4] Teuvo Kohonen, Jussi Hynninen, Jari Kangas and Jorma Laaksonen. *SOM_PAK The Self-Organizing Map Program Package*, 1995.

[5] Teuvo Kohonen. *Self-Organizing Maps: Second Edition*, Springer. pp. 48-51, 1997

[6] Aurobindo Sundaram. *ACM Crossroads*, April 1996.

[7] http://kdd.ics.uci.edu/databases/UNIX_user_data/UNIX_user_data.html

[8] Liren Chen, Katia Sycara. WebMate: A personal agent for browsing and searching, *Proceedings of the International Conference on Autonomous Agents (AA-98)*, 1998.

[9] Neil C. Rowe, Sandra Schiavo. An Intelligent tutor for intrusion detection on computer systems. *Computers & Education 31*. pp. 395-404, 1998.