그래프 기반 아웃라이어 검출 방법

정 서, 김 상욱
한양대학교 전자컴퓨터통신공학부
e-mail : {xaveng, wook}@agape.hayang.ac.kr

A Graph-Based Algorithm for Outlier Detection

Seo Jeong, Sang-Wook Kim
Dept. of Electronics and Computer Engineering, Hanyang University
{xaveng, wook}@agape.hanyang.ac.kr

요 약

아웃라이어란 데이터 셋 내에서 다른 객체들과 상대적으로 이질적인 객체를 의미한다. 본 논문에서는 기존 그래프 기반 아웃라이어 검출 방법의 문제점을 분석한다. 이를 통해, HITS 를 기반으로 하는 새로운 그래프 기반 아웃라이어 검출 방법을 제안한다. 마지막으로, 다양한 실험을 통하여 제안하는 방법이 아웃라이어 검출에 적합함을 보인다.

1. 서론

아웃라이어(outlier)란 데이터 셋에서 다른 객체들과 비교해 상대적으로 이질적인 객체를 의미한다[1]. 이러한 아웃라이어를 검출하는 것은 여러 도메인에서 유용하게 사용될 수 있다. 예를 들어 금융 거래 사기를 발견해야 하는 도메인에서는 한 고객의 금융 거래 패턴이 다른 고객들과 상이한 경우, 아웃라이어 검출을 통해 이를 검출할 수 있으며, 해당 고객의 거래 내역을 자세히 조사해 볼 수 있다.

아웃라이어 검출에 대한 다음과 같은 기존 방법들이 존재한다. 통계적인 방법[2]은 다양한 통계 모델 중 해당 데이터 셋이 따르는 통계 모델에서 벗어나는 객체를 아웃라이어로 검출하는 방법이다. 거리 기반 방법[1, 3]은 데이터 셋 내의 객체간의 거리를 척도로 하여 상대적으로 동떨어져 있는 객체를 아웃라이어로 검출하는 방법이다. 밀도 기반 방법[4]은 각 객체의 밀도가 해당 객체 주변에 존재하는 다른 객체의 밀도와 차이가 많이 나는 경우 해당 객체를 아웃라이어로 검출하는 방법이다.

기존의 방법들에는 다음과 같은 문제점들이 존재한다. 통계적 방법은 다차원 데이터의 경우 각 차원에 대한 통계 모델을 각각 구한 후, 이를 하나의 모델로 결합하여야 하는 문제가 발생하므로 적용하기 힘들다. 거리 기반 방법들은 아웃라이어를 검출하고자 할 때 거리만을 아웃라이어 척도로 사용하므로 local density 문제[5]가 발생될 수 있다. 밀도 기반 방법들은 각 객체의 주변 객체들만 비교하여 아웃라이어 여부 판단 기준으로 사용하기 때문에 multi-granularity 문제[5]가 발생될 수 있다.

최근, 이를 해결하기 위해 그래프 기반 방법이 제안되었다[6]. 참고 문헌 [6]에서는 주어진 데이터 셋을 그래프로 모델링 한 후, 해당 그래프에 Random walks with restart(RWR)[7]을 수행한다. 이를 통해 각 객체가 다른 객체들과 얼마나 동떨어져 있는지를 의미하는 점수를 부여할 수 있게 된다. 그러나 참고문헌 [6]에서는 주어진 데이터를 완전 그래프로 모델링 함으로써, 그래프의 무게중심의 객체에 가장 높은 점수가 부여되는 문제가 발생한다.

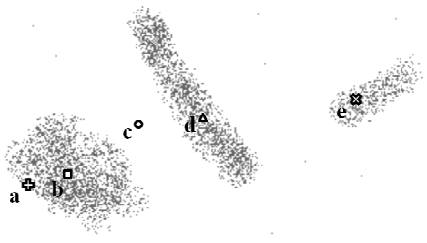
본 논문에서는 정확한 아웃라이어 검출을 위한 새로운 그래프 기반 방법을 제안한다. 첫째, 제안하는 방법의 수행을 위해, 주어진 데이터를 완전 그래프 대신 k-nn 그래프로 모델링 한다. 둘째, k-nn 그래프에서 RWR 수행 시 발생하는 근본적인 문제점을 분석하고, 이를 해결하기 위해 RWR 대신 가중치가 부여된 HITS 알고리즘을 적용한다. 마지막으로, 다양한 실험을 통해 제안 방법의 정확함을 보인다.

2. 제안하는 방법

2.1 연구 동기

기존의 그래프 기반 아웃라이어 검출 방법[6]은 주어진 데이터를 완전 그래프로 모델링 한 후 간선의 가중치를 기반으로 점수를 파급하도록 수정된 RWR을 적용하여 각 객체에 아웃라이어 점수를 부여한다. 아웃라이어 점수란 각 객체가 다른 객체와 얼마나 동질적인지를 나타내는 척도로써, 부여된 점수가 낮은 객체들을 아웃라이어로 간주한다. 객체의 in-degree가 높을수록, 그리고 객체를 가리키는 간선에 부여된 가중치가 높을수록 해당 객체는 높은 아웃라이어 점수를 갖는다.

참고문헌 [6]은 주어진 데이터 셋을 완전 그래프로 모델링 하였기 때문에 전체 그래프의 무게중심에 존재하는 객체가 다른 모든 객체보다 높은 점수를 가지게 되는 문제가 발생한다. 완전 그래프는 모든 객체들의 in-degree가 동일하다. 따라서 각 객체에 부여되는 점수는 해당 객체를 가리키고 있는 간선들의 가중치에 따라 결정된다. 이 때, 각 간선의 가중치는 두 객체 사이의 거리 기반 유사도로서, 두 객체가 가까울수록 큰 값을 가진다. 결과적으로 그래프의 전체 무게 중심에 가까운 객체일수록 다른 모든 객체들과의 거리의 합이 작아지므로 더 높은 점수를 가지게 된다.



(그림 1) 아웃라이어 검출 시 완전 그래프의 문제점

그림 1에서 아웃라이어로 판단되는 점 c의 점수는 점

a보다 낮아야 한다. 그러나 실제로는 전체 무게 중심에 존재하는 점 c의 점수가 가장 높으며, 점 b, d, a, e 순으로 점수가 낮아진다.

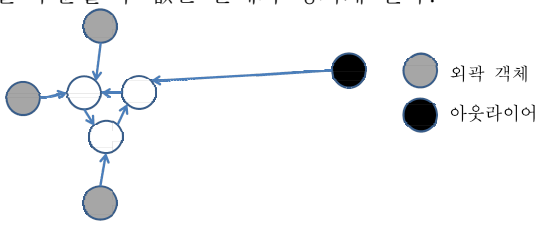
2.2 HITS 기반 아웃라이어 검출 방법

본 논문에서는 주어진 데이터 셋을 완전 그래프로 모델링 하는 대신 유사도[1]가 높은 k개의 객체로만 간선을 연결하는 k-nn 그래프로 모델링 한다. 즉, 객체 A의 k-nn에 객체 B가 포함될 때, A→B로의 간선을 연결한다. 각 간선의 가중치는 간선이 연결하는 두 객체의 유사도이다.

이를 통하여 본 논문에서는 각 객체의 in-degree를 주변 객체의 수에 비례하게 그래프를 모델링 할 수 있다. 즉, 주변에 객체들이 많은 객체의 in-degree는 높게, 주변에 객체가 거의 없는 객체의 in-degree가 낮게 그래프를 모델링 한다.

그러나 k-nn 그래프에 RWR을 적용시키는 것은 다음과 같은 새로운 문제를 발생시킨다. RWR의 경우, in-degree가 없는 객체들에게 restart probability에 해당하는 점수만을 동일하게 부여하기 때문에, 클러스터 외곽 객체와 아웃라이어를 구분할 수 없다는 근본적인 문제점이 존재한다.

예를 들어 그림 1에서 RWR을 적용하면 클러스터 외곽 객체와 아웃라이어 객체의 점수가 동일하게 부여되어 두 객체를 구분할 수 없는 문제가 생기게 된다.



(그림 2) 아웃라이어 검출 시 RWR 의 문제점.

따라서 본 논문에서는 RWR 대신 HITS[8]를 적용한다. HITS는 간선 정보를 토대로 하나의 객체에 허브점수와 권위점수를 각각 부여하는 알고리즘이다. RWR은 각 객체에 점수를 부여할 때 in-degree만을 이용한다. 그러나 HITS는 허브점수를 구할 때 out-degree를 이용해 점수를 계산하므로 위와 같은 문제를 해결할 수 있다.

HITS 알고리즘에서 객체 p의 허브점수 h와 권위점수 a는 다음과 같이 구할 수 있다[8]. 이 때, 집합 B(x)는 객체 x를 가리키고 있는 객체들의 집합을 의미하고, F(x)는 객체 x가 가리키고 있는 객체들의 집합을 의미한다.

$$a_p = \sum_{q \in B(p)} h_q \qquad h_p = \sum_{q \in F(p)} a_q$$

본 논문에서는 기존 HITS 알고리즘에서 허브점수와 권위점수를 계산할 때 각 객체 간의 유사도를 적용하도록 다음과 같이 수정하였다. 이는 객체들 간의 유사도에 비례하게 점수를 파급시키기 위해서이다. 이 때 W_{ij}는 객체 i와 j의 유사도이다.

$$a_p = \sum_{q \in B(p)} h_q (W_{pq} / \sum_{i \in B(p)} W_{pi})$$
$$h_p = \sum_{q \in F(p)} a_q (W_{pq} / \sum_{i \in F(p)} W_{pi})$$

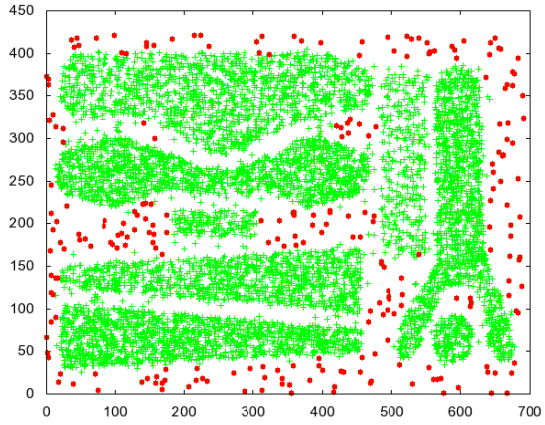
각 객체에 부여된 권위점수는 해당 객체주변에 객체가 얼마나 많은지를 의미하게 된다. 또한, 각 객체의 허브점수는 해당 객체 주변에 아웃라이어가 아닌 객체가 얼마나 많은지를 의미하게 된다. 본 논문에서는 각 객체에 부여된 허브점수와 권위점수의 합이 작은 n 개의 객체들을 아웃라이어로 검출한다.

[1] 본 논문에서는 각 객체 사이의 거리를 기반으로 하여 가까울수록 높은 유사도를 가지게 하였다. 그러나 이 외의 다른 어떤 방법을 유사도로 사용하여도 된다.

3. 실험 및 결과 분석

본 논문에서는 Chameleon[9] 에서 사용된 데이터들을 이용하여 아웃라이어 검출을 수행하였다. 그 중 도식된 데이터 셋은 8,000 개의 객체로 구성되어 있고 그림 3 과 같은 분포를 보인다.

실험을 위해, 해당 데이터 셋에 제안하는 방법을 적용한 후, 허브점수와 권위점수의 합이 낮은 250 개의 객체들을 검출하였다. 이 때, k 는 80 으로 설정하였다.



(그림 3) 아웃라이어 검출 결과.

실험 결과, 데이터 외곽의 아웃라이어와 클러스터들 사이의 아웃라이어를 검출하였고 클러스터 외곽 객체들은 검출하지 않는 모습을 볼 수 있다. 이 외에도 다수의 데이터에 대한 아웃라이어 검출을 수행하였지만, 유사한 경향을 보이므로 본 논문에는 지면 관계상 생략한다.

4. 결론

본 논문은 그래프를 기반으로 하는 새로운 아웃라이어 검출 방법을 제안하였다. 제안하는 방법은 주어진 데이터를 k-nn 그래프로 모델링 한 후, 가중치가 부여된 HITS 알고리즘을 적용하였다. 마지막으로, 실험을 통해 이 방법이 아웃라이어 검출에 적합함을 보였다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 'IT 융합 고급인력과정 지원사업'의 연구결과로 수행되었음 (NIPA-2010-C6150-1001-0005)

참고문헌

[1] S. Ramaswamy, “Efficient Algorithms for Mining Outliers from Large Data Sets,” In *SIGMOD*, pp. 427-438, 2000.
[2] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley & Sons, 1994
[3] E. Knorr, “Algorithms for Mining Distance-Based Outliers in Large Datasets,” In *VLDB*, pp. 392-403, 1998.
[4] M. Breunig, “LOF: Identifying Density-Based Local Outliers,” In *ACM SIGMOD Record*, Vol. 29, No. 2, pp. 93-104, 2000.
[5] S. Papadimitriou, “LOCI: Fast Outlier Detection using the Local Correlation Integral,” In *ICDE*, pp. 315-326, 2003.
[6] H. Moonesinghe, “Outlier Detection using Random Walks,” In *ICTAI*, pp. 532-539, 2006.
[7] S. Brin and L. Page, "The Anatomy of Large-Scale Hypertextual Web Search Engine," In *WWW*, pp.107-117, 1998.
[8] J. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” In *JACM*, Vol. 46, No. 5, pp. 604-632, 1999.
[9] G. Karypis, “Chameleon: Hierarchical Clustering using Dynamic Modeling,” In *IEEE Computer*, Vol. 32, No. 8, pp. 68-75, 1999.