# 아웃라이어 검출 방법을 위한 매개변수 설정

정서, 김상욱, 배덕호
한양대학교 전자컴퓨터통신공학부
e-mail : {xaveng, wook, dhbae}@agape.hanyang.ac.kr

# Parameter Setting for an Outlier Detection Algorithm

Seo Jeong, Sang-Wook Kim, Duck-Ho Bae
Dept. of Electronics Computer Engineering, Hanyang University

**요 약**

아웃라이어란 데이터 셋에서 다른 객체들과 비교해 상대적으로 이질적인 객체를 의미한다. 본 논문에서는 HITS 기반 아웃라이어 검출 방법의 매개변수 변화에 따른 성능 변화를 분석하고, 매개변수 설정을 위한 가이드라인을 제안한다.

## 1. 서론

아웃라이어 (outlier)란 데이터 셋에서 다른 객체들과 비교해 상대적으로 이질적인 객체를 의미한다[1]. 이러한 아웃라이어를 검출하는 기존의 방법에는 통계적, 거리 기반[1], 밀도 기반[2], 그래프 기반 방법[3] 등이 존재한다.

통계적 방법은 해당 데이터 셋이 따르는 통계 모델에서 벗어나는 객체들을 아웃라이어로 검출하는 방법으로 다차원의 데이터에 적용하기 어렵다는 문제점을 가지고 있다.

거리 기반 방법은 데이터 셋 내의 객체들 간의 거리를 척도로 하여 상대적으로 동떨어져 있는 객체들을 아웃라이어로 검출하는 방법으로 객체간의 거리만을 아웃라이어 척도로 사용하여 발생하는 local density 문제점이 있다[4]. *k*-Dist 방법[1]이 대표적이다.

밀도 기반 방법은 어떤 객체의 밀도가 해당 객체 주변에 존재하는 다른 객체의 밀도와 차이가 많이 나는 경우 해당 객체를 아웃라이어로 검출하는 방법이다. 이 방법은 적은 수의 아웃라이어들이 유사한 밀도를 가질 때 해당 객체들을 정상 객체로 판단하는 small outlying cluster 문제점이 있다[3]. 일반적으로 밀도 기반 방법에서 객체의 밀도는 해당 객체로부터 일정 범위 내에 속한 다른 객체의 수로 정의되며, LOF 방법[2]이 대표적이다.

그래프 기반 방법은 주어진 데이터 셋을 그래프로 모델링 한 후 링크 분석 알고리즘을 통하여 각 객체의 아웃라이어 점수를 부여하는 방법으로, Outrank 방법[3]이 대표적이다. Outrank 방법은 전체 데이터 셋을 완전 그래프로 모델링 하여 Random walks with restart (RWR)[5]를 적용하는 Outrank-a 방법과, Outrank-a 방법을 통해 모델링 된 그래프에서 일정 임계값 이하의 유사도를 가지는 간선들을 제거한 후 RWR 을 적용하는 Outrank-b 방법이 존재한다. Outrank-a 방법의 경우 전체 데이터 셋의 중심에 가까이 존재하는 아웃라이어를 검출하지 못하는 문제를 가지고 있다. Outrank-b 방법의 경우 전체 데이터 셋의 외곽에 존재하는 객체들에게 높은 아웃라이어 점수를 부여하는 문제를 가지고 있다.

본 논문의 저자들은 기존의 아웃라이어 검출 방법들이 가지는 문제를 해결하기 위해 HITS 기반 아웃라이어 검출 방법[6]을 제안한 바 있다. 본 논문에서는 매개 변수의 변화에 따른 HITS 기반 아웃라이어 검출 방법의 성능 변화를 분석하고, 매개변수 설정을 위한 가이드라인을 제안한다.

## 2. HITS 기반 아웃라이어 검출 방법

HITS 기반 아웃라이어 검출 방법은 먼저 주어진 데이터 셋을 *k*-NN 그래프로 모델링 한다. 이 때, 각 간선에는 다음과 같은 (0,1]의 값을 갖는 가중치를 부여한다.

$$\text{Euclidean\_Similarity}(A, B) = 1 - \frac{\sqrt{\sum_i^k (a_i - b_i)^2} - min}{max - min}$$

이 때, *max* 는 전체 데이터 셋에서 가장 멀리 존재하는 두 객체의 거리 (Euclidean distance)를, *min* 은 가장 가까이 존재하는 두 객체의 거리를 의미한다.

이를 통하여 두 객체 사이의 간선은 해당 객체들이 가까이 존재할 경우 높은 유사도를 갖게 되고, 해당 객체들이 멀리 존재할 경우 낮은 유사도를 갖게 된다.

구성된 그래프에 간선들의 가중치를 이용하도록 변형된 HITS 를 적용하여 각 객체에 중심도 (Centrality)와 중심_근접도 (Center_closeness) 점수를 부여한다. 중심도는 해당 객체가 얼마나 클러스터의 중심에 위치하는가를 의미하며, 중심 근접도는 해당 객체가 클러스터의 중심에서 얼마나 가까이 존재하는가를 의미한다.

중심도와 중심 근접도 점수를 계산하기 위해, 모든 객체에 동일한 초기값을 부여한 후, 아래 두 식을 순

차적으로 반복한다. 각 객체에 부여된 두 점수들이 수렴하면 반복을 종료한다.

$$Centrality_{i+1}(p) = \sum_{q \in In(p)} w_{q \to p} * \frac{Center\_Closeness_i(q)}{Z_{Out(q)}}$$

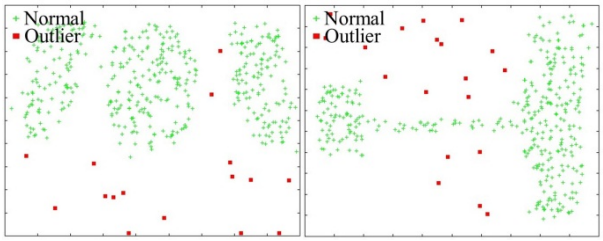
$$Center\_Closeness_{i+1}(p) = \sum_{q \in Out(p)} w_{p \to q} * \frac{Centrality_i(q)}{Z_{In(q)}}$$

이 때, $In(p)$는 객체 $p$ 를 가리키고 있는 객체들의 집합을, $Out(p)$는 객체 $p$ 가 가리키고 있는 객체들의 집합을 의미한다. 또한 $Z_{Out(q)}$는 객체 $q$ 가 가리키고 있는 객체들과의 유사도 총합을, $Z_{In(q)}$는 객체 $q$ 를 가리키고 있는 객체들과의 유사도 총합을 의미한다. 그리고 $w_{q \to p}$는 객체 $q$ 에서부터 $p$ 를 연결하는 간선의 가중치를 의미한다.

HITS 기반 아웃라이어 검출 방법에서는 중심-근접도 점수가 낮은 객체들을 아웃라이어로 검출한다.

## 3. 실험 및 결과 분석

실험을 위해 그림 1 의 2 개의 데이터 셋들을 사용하였다. 해당 데이터 셋들은 임의로 생성된 2 차원 데이터 셋들로서 해당 데이터 셋에서 십자 (+)로 표시된 객체들은 정상 객체이고, 사각형으로 표시된 객체들은 아웃라이어이다. 데이터 셋들은 각각 435 개와 378 개의 객체들로 구성되어 있으며, 그 중 16 개와 17 개의 아웃라이어를 각각 포함하고 있다.

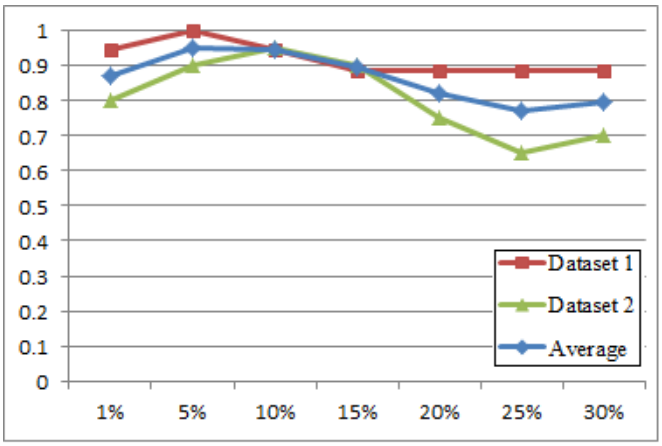



(a) 데이터 셋 1. (b) 데이터 셋 2.

(그림 1) 실험 대상 데이터 셋.

본 논문에서는 $k$-NN 그래프의 변화에 따른 HITS 기반 아웃라이어 검출 방법의 성능 변화를 분석하였다. 이를 위해, 매개변수 $k$ 를 전체 객체 수의 1%부터 30%까지 증가시키며 그래프를 구성한 뒤, 각 그래프에서 아웃라이어를 검출하였다. 이 때, 해당 데이터 셋에 존재하는 아웃라이어 수만큼의 객체를 추출하였으며, 추출된 객체들 중 실제로 아웃라이어인 객체들의 수인 정확도(precision)를 측정하였다.

그림 2 는 실험 결과를 나타낸다. $x$ 축은 매개변수 $k$ 값을 나타내며, $y$ 축은 측정한 정확도를 나타낸다. $k$ 가 증가함에 따라 평균 정확도가 향상되다가 k 가 5~10% 일 때 가장 좋은 정확도를 보였다. $k$ 가 10% 이상으로 증가하면 오히려 정확도가 낮아지는 경향을 보였다. 이는 $k$ 가 작을 경우, 그래프의 연결성이 낮아 정상 객체가 아웃라이어로 취급되는 문제가 발생하며, $k$ 가 일정 이상 커질 경우, 그래프의 연결성이 높아져 아웃라이어도 정상 객체로 취급되는 문제가 발생하기 때문이다.




(그림 2) 정확도 측정 결과.

이렇듯, $k$ 의 변화에 따라 HITS 기반 아웃라이어 검출 방법의 성능이 변화하였다. 비록 성능 변화의 폭이 크지 않으나, 데이터 셋에 적합한 $k$ 를 설정하는 것은 중요하다. 위의 결과를 근거로, 본 논문에서는 HITS 기반 아웃라이어 검출 방법의 $k$ 를 데이터 셋에 포함된 전체 객체 수의 5~10%로 설정할 것을 추천한다.

## 4. 결론

본 논문에서는 HITS 기반 아웃라이어 검출 방법의 매개변수 $k$ 의 변화에 따른 성능 변화를 분석하였다. 실험 결과, 전체 객체 수의 5~10%에 해당하는 값으로 매개변수 $k$ 를 설정할 때, 가장 좋은 성능을 보였다.

## 감사의 글

본 연구는 2010 년도 정부(교육과학기술부)의 재원으로 한국연구재단 (No.2008-0061006) 및 지식경제부 및 정보통신산업진흥원의 'IT 융합 고급인력과정 지원사업' (NIPA-2011-C6150-1101-0001) 및 IT/SW 창의연구과정의 연구결과로 지식경제부와 삼성전자주식회사에 의해 지원된 과제 (NIPA-2010-(C1810-1003-0007))로 수행되었음

## 참고 문헌

[1] S. Ramaswamy, “Efficient Algorithms for Mining Outliers from Large Data Sets,” In *SIGMOD*, pp. 427-438, 2000.

[2] M. Breunig, “LOF: Identifying Density-Based Local Outliers,” In *ACM SIGMOD Record*, Vol. 29, No. 2, pp. 93-104, 2000.

[3] H. Moonesinghe, “Outlier Detection using Random Walks,” In *ICTAI*, pp. 532-539, 2006.

[4] S. Papadimitriou, “LOCI: Fast Outlier Detection using the Local Correlation Integral,” In *ICDE*, pp. 315-326, 2003.

[5] S. Brin and L. Page, “The Anatomy of Large-Scale Hypertextual Web Search Engine,” In *WWW*, pp. 107-117, 1998.

[6] 정서, 김상욱, “그래프 기반 아웃라이어 검출 방법,” *한국정보처리학회 추계학술 발표대회*, Vol. 17, No. 2, pp. 173-174, 2010.