각도 기반 이상치 탐지 방법의 분석과 성능 개선

신용준○ 박정희
충남대학교 컴퓨터공학과
yjsin@cnu.ac.kr, cheonghee@cnu.ac.kr

Analysis and Performance enhancement of angle-based outlier detection

Yong Joon Sin○ Cheong Hee Park
Department of Computer Engineering Chungnam National University

요 약

고차원 공간에서 효과적인 이상치 탐지 방법으로 제안되었던 각도 기반 이상치 탐지(Angle Based Outlier Detection)는 객체와 객체를 비교하는 척도로 각도 개념을 사용하여 고차원 공간에서도 일반적인 거리기반 이상치 측정 방법보다 좋은 이상치 탐지 성능을 가진다. 그러나 어떤 이상치가 다른 이상치에 의해 둘러싸인 경우 정상객체와 구분하기 어렵다는 문제가 있다. 이 논문에서는 기존의 이상치 탐지 방법을 개선한 방법을 제안하고 실험을 통하여 기존의 방법과 제안한 새로운 방법을 비교하여 향상된 성능을 입증한다.

1. 서 론

이상치 탐지 방법의 목적은 어떤 데이터 셋에서 대부분의 객체들과 다른 특이한 패턴을 가지는 객체를 찾는 것이다. 이상치 탐지가 사용되는 분야의 예로는 신용카드 사기 탐지, 네트워크 침입 탐지 등이 있다. 이 논문에서는 다양한 이상치 탐지 방법을 간단하게 설명하고 이어서 고차원 공간에서 효과적인 각도 기반 이상치 탐지 방법에 대해서 설명한다. 다음으로 기존에 제안되었던 각도 기반 이상치 탐지 방법을 자세히 분석하고 분석 결과를 기반으로 기존의 각도 기반 이상치 탐지 방법의 취약점을 알아보고 이를 극복하기 위한 개념을 설명한다. 그리고 이 개념을 이용하여 이상치 탐지 성능과 속도가 향상된 새로운 각도 기반 이상치 탐지 방법을 제안한다. 또한 실제 데이터를 이용한 실험을 통하여 기존의 각도 기반 이상치 탐지 방법과 비교한 결과를 보이고 제안한 방법이 기존 방법보다 좋은 성능을 가진다는 것을 증명한다.

2. 관련연구

지금까지 데이터 마이닝, 기계학습, 통계 등의 분야에서 다양한 종류의 이상치 탐지 방법이 제안되어져 왔다. 이상치 탐지 방법의 하나인 통계 기반 이상치 탐지는 이상치 탐지의 초기 방법으로 주어진 데이터를 특정한 분포나 확률모델로 가정해 모델을 설정하고 그 모델에 따라 이상치를 구분하는 방법이다.[2] 거리 기반 이상치 탐지 방법은 한 객체의 이상치 정도를 판단하는 척도로 거리를 이용하는 방법이고[3] 밀도 기반 이상치 탐지 방법은 이상치를 나타내는 척도로 밀도를 사용하는 것으로 일반적으로 밀도는 거리의 역수로 정의된다.[4] 표본 기반 이상치 탐지는 전문가에 의해 확실한 이상치, 정상객체 표본을 구성한 후에 그 표본을 이용하여 이상치 탐지 방법의 성능을 개선한 방법이다.[5] 고립트리(isolation tree) 이상치 탐지 방법은 무작위 공간 분할 트리를 이용하여 대량의 데이터에도 빠른 속도로 이상치 탐지를 수행하도록 하는 방법이다.[10] 그러나, 이상치 탐지 방법들은 고차원에서 이상치 탐지 성능이 떨어진다는 단점이 있고 이를 개선한 방법인 각도기반 이상치 탐지 방법이 제안되었다. 차원이 증가함에 따라 거리 개념의 의미가 없어지기 때문에 객체의 이상치 정도를 측정하는 방법으로 거리벡터 쌍이 이루는 각도를 사용한다.[1]

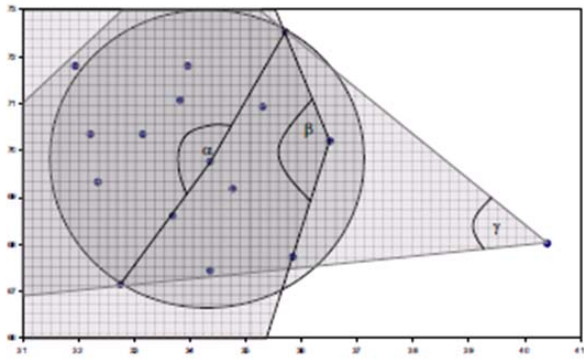

그림 1. 객체의 위치에 따른 각도[1]

그림 1을 보면 한 객체를 중심으로 다른 두 객체와 이루는 각을 측정할 때 군집 내부 객체를 중심으로 측정한

각 α는 각도 측정 대상에 따라 변화량이 크고, 군집 경계에 위치한 객체에서 측정한 각 β는 변화량이 α에 비해 작게 나타난다. 군집 외부에 위치한 객체에서 측정한 각 γ는 α, β와 비교하면 그 변화량이 매우 작다. 이러한 특징을 이용하여 한 객체를 중심으로 나머지 객체들과의 거리벡터 쌍이 이루는 각도를 측정하고 각도의 변화량을 나타내는 분산에 거리가중치를 적용하여 계산한 값을 각도 기반 이상치 정도(Angle Based Outlier Factor, ABOF)라고 하며 $ABOF$는 식 (1)과 같이 정의 된다.[1]

$$ABOF(A) = VAR_{B,C \in D}\left(\frac{\langle \overline{AB}, \overline{AC} \rangle}{\| \overline{AB} \|^2 \cdot \| \overline{AC} \|^2}\right) \quad (1)$$

모든 객체에 대해 각도 기반 이상치 정도를 계산하고 $ABOF$값이 작은 상위 α%의 객체를 이상치로 분류하는 것이 각도 기반 이상치 탐지의 기본 개념이다.

각도 기반 이상치 탐지 방법은 고차원 공간에서도 의미있는 벡터 사이의 각도를 이상치 정도 측정에 사용하여 고차원 데이터에 대해서도 잘 작동하는 알고리즘으로 다음과 같은 특징이 있다. $ABOD$는 이상치 정도를 계산하기 위해서 객체 쌍의 각도를 이용하여 고차원에 대해서 이상치 분류 정확도는 좋다. 하지만 모든 객체 쌍에 대해서 각도를 측정하기 때문에 시간복잡도가 $O(n^3)$로 데이터의 크기에 대해서 확장성을 가지지 못한다. 이러한 문제점을 개선한 방법으로 각도를 측정해야 할 객체 쌍의 수를 줄인 $FastABOD$가 제안되었다. $FastABOD$는 k개의 인접이웃만을 각도측정 객체 쌍으로 이용하여 속도 문제를 개선하여 시간복잡도가 $O(n^2 + n \cdot k^2)$이다. $ABOD$가 거리가중치를 사용하기 때문에 저차원 공간에서는 인접이웃을 이용해도 모든 객체를 이용하는 경우와 큰 차이가 없다. 하지만 객체 사이의 거리를 이용하기 때문에 고차원에서는 $FastABOD$ 정확도가 $ABOD$에 비해 많이 떨어진다. $LBABOD$는 $FastABOD$가 고차원에서 가지는 문제를 해결하기 위해 이상치로 분류한 상위 α%의 객체에 대해서 $ABOD$처럼 모든 객체 쌍을 이용하여 다시 이상치 정도를 측정하여 정제기법을 적용한 방법이다. 이 논문에서는 $ABOD$의 문제점을 보완하고 $FastABOD$와 알고리즘 수행 시간은 비슷한 방법을 제안한다.

3. 각도 기반 이상치의 문제점

각도 기반 이상치 탐지는 객체를 내부점, 경계점, 이상치로 나누는데 내부점은 $ABOF$ 값이 가장 크고 경계점은 중간, 이상치는 작은 값을 가진다. 그림 2와 3은 3차원 공간에 반지름이 0.3인 구안에 균등하게 분포된 군집과 그 옆에 2차원 형태로 평면에 분포된 이상치가 존재할 때, 내부점은 파란색(○), 경계점은 녹색(△), 이상치는 빨간색(×)으로 시각화한 것이다. 일반적인 $ABOF$는 그림 2와 같은 상황에서는 빨간색의 이상치 객체를 정확하게 분류하는 성능을 보이지만 그림 3처럼 이상치가 다른 이상치에 의해 둘러싸인 경우에는 빨간색 이상치 객체의 $ABOF$ 값이 정상객체보다 각도 분산이 커지게 되고 그로 인해 정상 객체(내부점, 경계점)들과 구별 할 수 없게 됨으로써 이상치 탐지 성능을 나쁘게 한다.

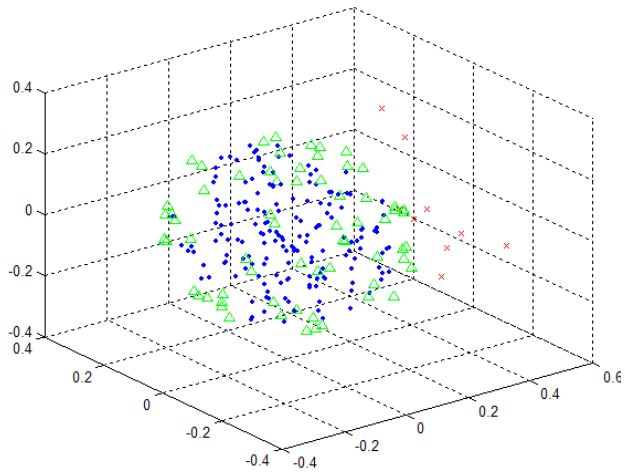

그림 2. 3차원 인공데이터 1

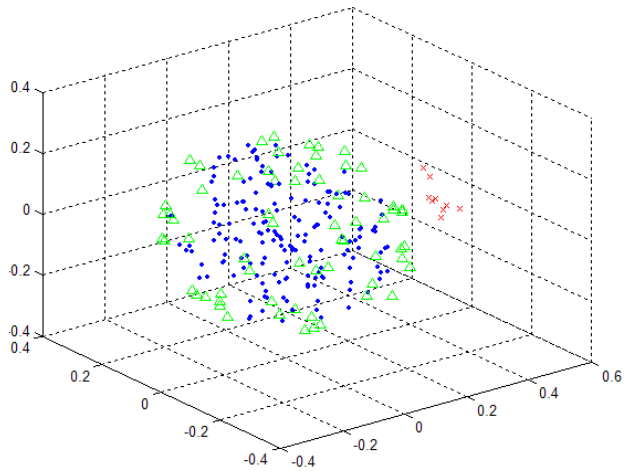

그림 3. 3차원 인공데이터 2

4. 각도 기반 이상치 탐지의 개선

이 장에서는 기존의 각도 기반 이상치 탐지 방법의 정확도를 향상 시킨 새로운 각도 기반 이상치 측정을 위한 개념을 설명한다.

4.1 각도 측정 대상 선택

각도기반 이상치 탐지 방법은 이상치 정도를 정확하게 나타내기 위해서는 내부점, 경계점 같은 정상객체는 $ABOF$가 크고 이상치는 $ABOF$가 작아야한다. 객체의 종류 별로 각도 측정에 사용되는 객체들이 다음과 같다면 기존의 각도 기반 이상치 탐지 방법보다 더 정확한 이상치 정도를 측정 할 수 있다.

정상객체(내부점, 경계점) : 거리벡터 쌍의 각도 분산이 크도록 객체를 중심으로 여러 방향에 각도 측정 후보가 존재하는 경우.

이상치 : 거리벡터 쌍의 각도 분산이 작은 값을 가지도록 객체를 중심으로 한쪽 방향에 밀집해서 각도 측정 후보가 존재하는 경우.

위의 조건을 만족하는 기준으로 “정상객체 근접이웃“를 사용할 수 있다. 여기서 언급한 정상객체 근접이웃은 임의의 객체 o로부터 가까이 있는 k개의 정상객체로 이루어진 집합을 의미한다. 일반적으로 정상객체는 군집을 형성하고 있고 내부점이나 경계점은 군집 내에 존재하므로 정상객체 근접이웃을 선택하여 선택된 객체만을 이용해 각도 분산을 측정 하게 되면 그 분산이 크다. 또한 이상치는 가까이 존재하는 다른 이상치들을 거리 측정 대상 객체로 선택하지 않고 더 멀리있는 정상 객체 군집을 선택하므로 각도의 분산이 상대적으로 작게 된다. 또한 주위에 존재할지도 모르는 이상치를 제외 할 수 있어서 그림 3과 같은 경우에도 영향을 받지 않는다. 이러한 기준으로 k개의 각도 측정 후보를 선택하여 $FastABOD$에 적용하면 이상치 분류 정확도는 보다 향상되고 $FastABOD$와 비슷한 수행시간을 가질 수 있다.

4.2 정상객체를 구분하기 위한 방법

정상객체의 구분을 위해 여러 가지 방법을 사용 할 수 있는데 대표적으로 군집화 기법의 사용과 이상치 탐지 기법으로 구분할 수 있다. 군집화 기법에 의해 군집을 형성한 객체를 정상객체로 분류하는 방법[7, 8, 9], 이상치 탐지 기법[4, 5, 10]에 의해 이상치로 분류되지 못한 객체를 정상객체로 분류하는 방법이다. 이 논문에서는 정상객체의 구분을 위해 새로운 기준을 사용한다. 바로 “근접이웃 참조 횟수“로 다른 객체의 근접이웃으로 포함되는 횟수를 저장하여 포함되는 경우가 많은 순서대로 정렬하여 사용자가 원하는 수량만큼 정상객체로 분류하는 것이다. 근접이웃 참조를 다른 논문에서는 역 근접이웃으로 표현하기도 한다.[6] 근접 이웃 참조 횟수를 구하는 알고리즘은 표 1과 같다.

```
근접이웃 참조 횟수
입력 : k, D
출력 : rcnn(Reference Count of Nearest Neighborhood)
rcnn : 크기가 n(D)이고 0으로 초기화된 int형 배열.
For each object p∈D
  knn_p = get_knn(p);//Find p's k-nearest neighbors
  For each object q∈knn_p
    rcnn[q.index]++;
  end
end
return rcnn;
```

표 1. 근접이웃 참조 횟수 알고리즘

그림 4는 정규분포를 따르는 200개의 인공데이터에 대해 근접이웃 참조 횟수를 0에서 1사이의 값으로 정규화하여 표현한 것이다.

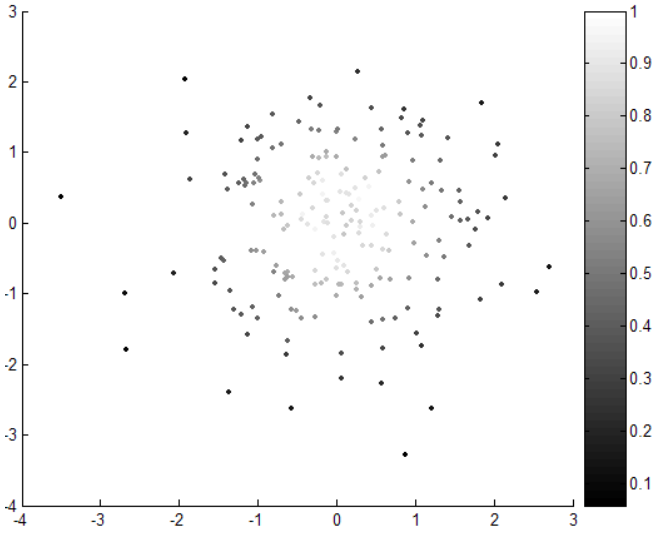

그림 4. 근접이웃 참조 횟수 시각화

이 기준을 사용하면 LOF를 기준으로 분류하는 것과 유사하게 정상 객체를 분류할 수 있다는 결과를 인공데이터의 실험으로 알 수 있었다.

4.3 제안하는 방법

이 장에서는 기존의 각도 기반 이상치 탐지 방법의 약점을 보완하여 향상된 이상치 탐지 성능과 속도를 가지는 개선된 각도 기반 이상치 탐지 방법인 $ABOD_{ns}$에 대한 설명을 한다. 다음은 $ABOD_{ns}$에서 사용될 용어에 대

한 설명으로 4.2 절에서 설명한 방법을 이용하여 분류한 정상객체 집합을 NS(normal set)로 하고 이 NS을 이용하여 선택되는 각도 측정에 사용되는 정상객체 근접이웃을 knn_{ns}이라 한다.

정의 1. knn_{ns}

임의의 객체 o에 대하여 NS의 객체 중에서 가장 가까운 거리에 존재하는 k개의 원소들의 집합을 $knn_{ns}(o)$로 나타낸다.

이상치 정도를 나타내는 $ABOF_{ns}$는 다음과 같다.

정의 2. knn_{ns}**를 이용하는 각도 기반 이상치 정도**

$$ABOF_{ns}(A) = VAR_{B,C \in knn_{ns}(A)}\left(\frac{\langle \overline{AB}, \overline{AC} \rangle}{\| \overline{AB} \|^2 \cdot \| \overline{AC} \|^2}\right)$$

다음 표 2는 근접이웃 참조 횟수를 기준으로 분류한 정상객체를 각도 측정 근접이웃으로 사용한 $ABOF_{ns}$와 $ABOF$의 차이가 인공 데이터에서 어떻게 나타나는지 정리한 내용이다. X는 정상객체 중에서 $ABOF$ 값이 가장 낮은 30개의 평균을 계산한 것이고 Y는 모든 이상치의 $ABOF$ 값의 평균을 계산한 것이다. 그림 2에서는 $ABOF$나 $ABOF_{ns}$가 Y가 X에 비해 낮아서 이상치 탐지 성능이 좋다는 것이 나타나지만 그림 3과 같은 경우에는 $ABOF$의 경우 Xㅂ다 Y가 높은 $ABOF$ 값을 가지는 것을 볼 수 있다.

		X	Y	Y/X
그림2	$ABOF$	1101.13	40.83	0.037
	$ABOF_{ns}$	1048.87	1.78	0.0016
그림3	$ABOF$	1070.16	1163.59	1.0873
	$ABOF_{ns}$	970.47	6.09	0.0062

표 2. $ABOF$ 와 $ABOF_{ns}$의 차이

이러한 성능개선 효과가 실제 데이터 셋에서는 어떻게 작용하는지 다음 실험결과 부분에서 다루었다.

5. 실험 결과

이 장에서는 실제 데이터 셋인 Satlog, ISOLET (http://archive.ics.uci.edu/ml/datasets.html)을 이용하여 고차원 데이터에서 기존의 각도 기반 탐지 방법과 우리가 제안한 새로운 방법의 성능을 비교한다. 이 논문에서 새로 제안한 각도 기반 이상치 탐지 알고리즘에는 각도 분산 측정에 이용하는 이웃의 수 k와 각도 측정 후보로 사용될 정상객체 집합의 수 $n(NS)$는 사용자에 의해 결정된다.

5.1 이상치 탐지 정확도

ISOLET는 26개의 클래스와 617개의 속성으로 구성된 데이터 셋으로 각 클래스별 240개의 데이터가 존재한다. 여기에서는 1개의 클래스 전부를 정상객체, 나머지 25개의 클래스에서 무작위로 2개씩 이상치로 선택하여 실험 데이터를 생성하였다. Satlog 데이터 셋은 클래스 6개, 속성 36개로 각 클래스별로 1533, 703, 1358, 626, 707, 1508개의 객체가 존재하는데 1개의 클래스에서 무작위로 300개를 정상객체로 추출하고, 나머지 5개의 클래스에서 무작위로 5개씩 이상치로 선택하여 실험 데이터를 생성하였고 알고리즘 입력 변수 k는 50로 설정하였고 $n(NS)$의 수는 전체 데이터의 70%로 설정하였다. 표 3은 각 데이터 셋에 기존 $ABOD$와 $ABOD_{ns}$를 수행한 결과를 나타낸다. $accuracy$는 모든 데이터를$ABOF$값으로 낮은 것부터 높은 것까지 정렬했을때 실제 이상치 개수(n)에 대한 $ABOF$값이 낮은 상위 n개 중 이상치의 개수의 비율이다. AUC는 수신자 동작 특성(Receiver Operating Characteristic) 곡선의 면적을 계산한 값으로 분류기의 성능을 평가하는데 사용되는 방법이다.[11] 수신자 동작 특성 곡선은 거짓 긍정률(false positive rate)의 변화에 따른 참 긍정률(true positive rate)을 측정한 그래프이다. 참 긍정률과 거짓 긍정률은 다음 식 (2)와 (3)에 의해 구할 수 있고,

$$TPR = TP/(TP+FN) \quad (2)$$
$$FPR = FP/(TN+FP) \quad (3)$$

여기에서 TP(true positive)는 참 긍정으로 이상치 탐지 방법에 의해 이상치 분류된 객체 중 실제 이상치의 수, FN(false negative)는 거짓 부정으로 정상객체로 분류된 객체중 실제 이상치의 수, FP(false positive)는 거짓 긍정으로 이상치로 분류된 객체 중 실제 정상객체의 수, TN(true negative)는 정상객체로 분류된 객체중 실제 정상객체의 수를 의미한다. 즉 참 긍정률은 실제 이상치중에서 이상치로 분류한 객체의 비율이고 거짓 긍정률은

실제 정상객체 중에서 이상치로 분류한 객체의 비율을 의미한다. 표 3에서 대부분의 경우에서 $ABOD$보다 $ABOD_{ns}$가 $accuracy$와 AUC(area under ROC curve)가 비슷하거나 높은 값을 가진다는 것을 볼 수 있다.

데이터 셋	정상 클래스	$ABOD$		$ABOD_{ns}$	
		$accuracy$	AUC	$accuracy$	AUC
Satlog	1	68	96.97	80	99.02
	2	56	92.46	88	99.72
	3	68	94.06	72	94.98
	4	52	87.96	52	91.01
	5	48	86.57	56	88.16
	7	48	90.53	68	91.54
	평균	56.68	91.43	69.32	94.07
ISOLET	1	76	97.96	88	99.48
	2	72	93.03	76	93.88
	3	88	99.00	96	99.43
	4	84	96.40	84	98.33
	5	84	98.48	88	99.08
	6	84	97.98	84	98.28
	7	76	95.58	80	98.28
	8	80	99.20	88	99.28
	9	92	98.61	92	98.66
	10	80	98.13	88	98.75
	11	76	96.23	76	96.48
	12	80	98.18	88	99.60
	13	72	94.85	76	96.55
	14	76	96.48	76	98.30
	15	84	96.83	92	98.18
	16	52	87.38	52	89.01
	17	88	99.33	92	99.61
	18	96	99.85	100	100
	19	88	96.26	92	97.28
	20	72	93.23	72	95.58
	21	80	98.43	84	98.98
	22	60	89.70	72	92.85
	23	36	88.06	44	83.43
	24	88	99.68	96	99.95
	25	92	99.13	96	99.81
	26	88	98.01	88	98.98
	평균	78.60	96.38	83.08	97.23

표 3. 이상치 탐지 정확도 비교

5.2 각도 측정 후보와 정확도

이 절에서는 각도 측정 후보가 어떻게 선택 되어지는가에 따라서 이상치 탐지 정확도에 미치는 영향을 알아본다. 이 실험에서는 Satlog 데이터의 1 클래스를 정상객체로 설정한 실험데이터를 사용하고 정상 객체 구분 방법은 근접이웃 참조 횟수를 이용하였다. 그림 5는 각도 측정 후보인 정상객체 집합의 크기가 전체 데이터의 30%부터 90%까지 변화에 따른 이상치 탐지 성능을 측정한 것이다.

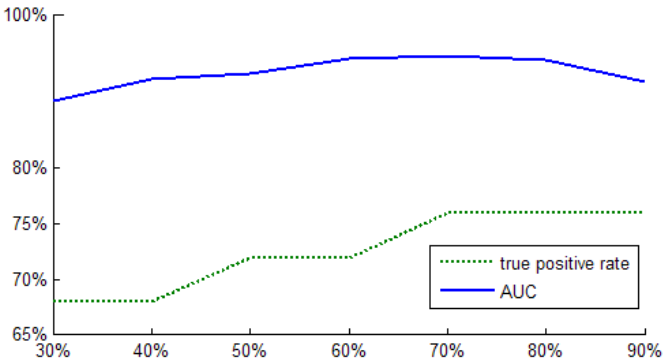

그림 5. 정상객체 집합의 크기 변화에 따른 정확도

다음 그림 6은 정상 객체 구분 성능 측정을 위해 정상객체 집합의 크기에 따라 정상객체 집합에 포함된 이상치의 수를 나타낸 그래프이다.

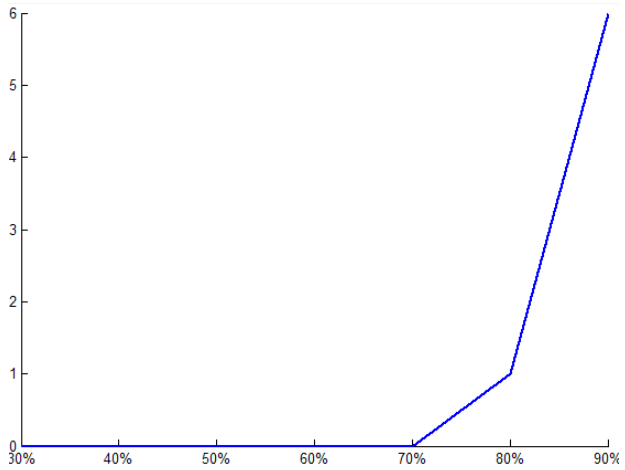

그림 6. 정상객체 집합에 포함된 이상치의 수

그림 5를 보면 정상객체 집합의 크기를 전체 데이터의 70%로 설정할 때 이상치 탐지 성능이 가장 좋았는데 그 이유는 그림 6의 그래프에서 정상객체 집합의 크기가 70%이상이 되면 근접이웃 참조 횟수에 의해 구분되는 정상객체에 이상치가 포함되게 되고 결과적으로 이상치 탐지 성능이 떨어진다는 것을 확인할 수 있다.

5.3 알고리즘 수행 시간

2절에서 우리는 각도 기반 이상치 탐지인 $ABOD$의 시간복잡도가 $O(n^3)$이고 $FastABOD$는 시간복잡도가 $O(n^2 + n \cdot k^2)$라고 설명하였다. 이 논문에서 제안한 $ABOD_{ns}$는 총 세 가지 부분으로 나눌 수 있는데 첫째로 정상객체를 분류하는 과정, 둘째로 분류한 정상객체에 대해 근접이웃을 구하는 과정, 셋째로 근접이웃에 대한 각도 측정을 하는 과정이다. 정상객체를 분류하는 과정은 각각의 객체에 대해 근접이웃을 구하면 되는 것으로 $O(n^2)$의 시간복잡도를 가진다. 둘째로 분류한 정상객체에 대한 각도 측정은 앞에서 구한 근접이웃을 정제하여 정상객체 근접이웃을 구하면 되므로 $O(n)$이다. 셋째로 근접이웃에 대한 각도 측정 시간은 $O(n \cdot k^2)$이므로 $ABOD_{ns}$의 시간복잡도는 $O(n^2 + n \cdot k^2)$로 $ABOD$보다 작은 시간복잡도를 가진다.

6. 결론

이 논문에서는 이상치의 주변에 이상치가 분포되어있는 경우, 기존 $ABOD$가 가지는 문제점을 해결하기 위해 정상객체를 각도 측정 후보로 이용하는 개념을 소개하였고 정상객체를 구별하는 기준으로 근접이웃 참조 횟수를 제안하였다. 또한 실험을 통하여 우리가 제안한 각도 기반 이상치 탐지 방법이 기존의 방법보다 좋은 이상치 탐지 정확도와 수행 시간의 측면에서 모두 개선되었다는 것을 보였다.

참고문헌

[1] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-Based Outlier Detection in High-dimensional Data. in KDD, 2008

[2] V. Barnett and T. Lewis. *Outliers in statistical data*. John Wiley, 1994

[3] E. Knorr and R. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, pp. 211 - 222, 1999

[4] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In SIGMOD, pp. 93 - 104, 2000

[5] Cui Zhu, Hiroyuki Kitagawa, and Christos Faloutsos. Example-Based Robust Outlier Detection in High Dimensional Datasets. In ICDM, 2005

[6] Wen Jin, Anthony K. H. Tung, Jiawei Han, and Wei Wang. Ranking Outliers Using Symmetric Neighborhood Relationship. in PKADD, 2006

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In ACM SIGKDD, pages 226 - 231, 1996.

[8] Christian Böhm, Karin Kailing, Peer Kröger, and Arthur Zimek. Computing Clusters of Correlation Connected Objects. in SIGMOD, 2004

[9] Vineet Chaoji, Mohammad Al Hasan, Saeed Salem, and Mohammed J. Zaki. SPARCL: Efficient and Effective Shape-based Clustering. In ICDM, 2008

[10] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In ICDM, 2008.

[11] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining. Addison Wesley, 2006