HDoutlier 정리

* Nearest neighbor distance만을 사용하여 이상치 분류 및 탐지하기 때문에 False negative rate를 증가시키는 경향이 있다 (왜 NND가 FN을 증가시키는지 추후 설명 - )
* R 라이브러리 ‘HDoutliers’
* Leader Algorithm을 사용하여 clustering 과정이 중간에 추가되어 연산이 느리다 (왜 해당 알고리즘을 사용하면 느려지는지 추후 설명)
* 거리 기반 이상치탐지 기법 – 거리 기반이기 때문에 차원이 얼마나 높은가에 상관없이 일차원 문제로 축소됨
* 버전 1 : 각 점마다 nearest neighbor distance만을 계산하고, 이에 따라 이상치탐지. Small sample에 권장됨
* 버전2 : 중간에 클러스터링 과정이 포함된 버전. 선 군집화 후, 각 클러스터마다 대표점을 추출하여 대표점 간의 거리를 고려하여 이상치 탐지

알고리즘은 3개의 한계점을 가지고 있음

1. 오직 nearest neighbor distance만을 사용하여 이상치탐지를 진행한다.

* 이 방법이 유효하기 위해서는 이상치 클러스터(들)가 정상 클러스터(들)로부터 isolated되어 있어야 한다는 가정이 필요하다.
* 만약 2개 이상의 이상치 클러스터들이 서로 가깝게 형성되어 있다면, 이들 간의 nearest neighbor distance가 작게 산출되면서 정상 클러스터로 판별되어버리고 말 것이다.

1. Leader algorithm으로 중간 클러스터링 과정이 추가된다.

* Leader algorithm으로 자체적인 클러스터를 형성한다. Each cluster is a ball in the high dimensional data space.
* The radius of the ball is selected such that it is well below the expected value of the distances between n(n-1)/2 pairs of points distributed randomly in a d-dimensional unit hypercube.
* 위 과정으로 클러스터들이 형성이 되고, 각 클러스터에서 대표점을 산출하여 각 대표점에 한해서만 nearest neighbor distance를 계산하고, 이를 기준으로 이상치를 탐지한다. (정상치와 이상치 간의 거리가 멀 것이라는 가정)
* 위 가정 하에, 이상치는 nearest distance가 매우 크게 나타날 것이고, 이를 이상치로 판별한다.
* 문제는, 클러스터링과 대표점을 산출하는 과정에서 데이터가 가지고 있는 density적 특징이 완벽히 무시된다.
* (논문 41 페이지 그래프 e-ii)를 예로 들어보겠다. bimodal인 해당 데이터는 총 2001개의 data points가 있고, 좌상단에 1000개의 정상 데이터가 위치하고, 중앙에 1개의 이상치 데이터가, 그리고 우하단에 1000개의 정상 데이터가 위치한다. 즉, 두개의 정상 클러스터 사이에 이상치 클러스터가 자리한다고 하자. 그렇게 되면 leader algorithm을 거치게 되면 우하단에 위치한 정상 클러스터는 데이터 50%가 할당되었음에도 불구하고 이상치 클러스터로 판별되어버리고 만다.

1. Threshold Calculation에서 next potential candidate을 사용하는데, 이 과정에서 false negative rate가 증가하는 경향이 있다.

* **이거는 왜 그런지 설명을 못하겠는데…?**

Stray Algorithm 정리

피피티에 이미 알고리즘의 특징과 차별점 슬라이드를 넣어놨으니까, 이제 알고리즘이 어떻게 돌아가는지 전체적으로 먼저 설명하고, 설명하는 과정에서 HD아웃라이어의 문제점을 어떻게 타파했는지 설명하자

Input data

* Numerical, categorical 모두 사용 가능하다. 단, scale invariant하게 만들어주기 위해 normalization이 필요하다. (scale invariant인 것이지 affine invariant인 것은 아니다. Affine invariant 설명?)
* Numerical 🡪 min-max normalization, categorical 🡪 correspondence analysis (0~1로 normalize)
* Correspondence Analysis : 일단 피피티 만들어보고, 설명을 추가할지 말지 결정하자…
* ;;;;;;’