J. Soc. Korea Ind. Syst. Eng Vol. 41, No. 2 : 56-64, June 2018
https://doi.org/10.11627/jkise.2018.41.2.056
ISSN : 2005-0461(print)
ISSN : 2287-7975(online)

An Efficient One Class Classifier Using Gaussian-based Hyper-Rectangle Generation

Do Gyun Kim · Jin Young Choi† · Jeonghan Ko

Department of Industrial Engineering, Ajou University

가우시안 기반 Hyper-Rectangle 생성을 이용한 효율적 단일 분류기

김도균 · 최진영† · 고정한

아주대학교 산업공학과

In recent years, imbalanced data is one of the most important and frequent issue for quality control in industrial field. As an example, defect rate has been drastically reduced thanks to highly developed technology and quality management, so that only few defective data can be obtained from production process. Therefore, quality classification should be performed under the condition that one class (defective dataset) is even smaller than the other class (good dataset). However, traditional multi-class classification methods are not appropriate to deal with such an imbalanced dataset, since they classify data from the difference between one class and the others that can hardly be found in imbalanced datasets. Thus, one-class classification that thoroughly learns patterns of target class is more suitable for imbalanced dataset since it only focuses on data in a target class. So far, several one-class classification methods such as one-class support vector machine, neural network and decision tree there have been suggested. One-class support vector machine and neural network can guarantee good classification rate, and decision tree can provide a set of rules that can be clearly interpreted. However, the classifiers obtained from the former two methods consist of complex mathematical functions and cannot be easily understood by users. In case of decision tree, the criterion for rule generation is ambiguous. Therefore, as an alternative, a new one-class classifier using hyper-rectangles was proposed, which performs precise classification compared to other methods and generates rules clearly understood by users as well. In this paper, we suggest an approach for improving the limitations of those previous one-class classification algorithms. Specifically, the suggested approach produces more improved one-class classifier using hyper-rectangles generated by using Gaussian function. The performance of the suggested algorithm is verified by a numerical experiment, which uses several datasets in UCI machine learning repository.

Keywords : Imbalanced Dataset, One-Class Classification, Classifier Using Hyper-Rectangle, Classification Rate, Interpretability

1. 서 론

최근 산업 현장에서 품질 관리를 위해 고려해야 하는 가장 큰 이슈 중 하나는 불균형 데이터(imbalanced data)이다. 그 이유는 기술의 발달과 고도로 발전하는 품질 관리 기법들로 인해 불량률은 매우 적은 수준으로 줄어들었으며, 공정에서 발생하는 데이터에는 불량 데이터가 거의 존재하지 않게 되었기 때문이다. 즉, 하나의 클래스(양품)가 다른 클래스(불량품)에 비해 매우 크기 때문에

Received 28 February 2018; Finally Revised 15 May 2018; Accepted 7 June 2018
† Corresponding Author : choijy@ajou.ac.kr

불균형 데이터가 된다.

이러한 데이터를 분류하기 위한 방법으로 다루고자 하는 데이터에 존재하는 클래스의 수에 따라 단일 분류(One class classification)와 다중 분류(Multi-class classification)를 고려할 수 있다[18]. 다중 분류는 데이터 집합이 다양한 클래스로 구성되며, 새로 발생하는 인스턴스가 여러 개의 클래스들 중 어떤 클래스의 데이터인지 예측하는 것이다[11]. 이를 위해 데이터 상에 존재하는 여러 클래스의 차이점을 이용하여 분류의 기준이 되는 분류기를 생성한다. 예를 들면, 대표적인 분류 기법인 support vector machine(SVM)[4, 13]은 양(positive)의 클래스와 음(negative)의 클래스의 차이점을 이용하여 두 개의 클래스를 구분할 수 있는 hyper-plane을 찾는다. 반면, 단일 분류는 주어진 데이터 집합이 오직 하나의 클래스로만 구성되어 있으며, 해당 클래스에 소속되지 않은 데이터는 모두 이상치(outlier)로 가정한다. 이 때, 단일 분류는 클래스가 하나만 존재하기 때문에 해당 클래스의 패턴을 보다 정밀하게 묘사할 수 있는 분류기를 생성하는 것이 필요하다.

그러나 만일 불균형 데이터에 대해 다중 분류를 이용한 접근을 고려한다면 클래스와 다른 클래스 간의 차이를 통해 분류기를 학습하는 특성에 의해 분류기를 생성하기 위해 보다 많은 연산이 요구되고 분류기의 정확도가 떨어진다. 따라서, 불균형 데이터에 대해서는 검출하고자 하는 대상 클래스에 대해 면밀한 학습을 통해 분류기를 생성하는 단일 분류를 통한 접근이 더 적합하다. 그 외에도, 단일 분류는 클래스 간의 차이를 통해 학습하는 것이 아니기 때문에, 모든 클래스의 정보가 주어지지 않은 경우에도 올바른 분류기를 생성할 수 있다는 장점을 지닌다.

현재 존재하는 대부분의 단일 분류 알고리즘들은 SVM과 같이 다중 분류에서 사용되는 방법론들을 단일 분류 상황에 적용할 수 있도록 수정한 것이다. 이러한 방법들은 단일 분류 상황에서도 준수한 분류 정확도를 보이지만, 많은 방법들이 분류 결과에 대한 요인을 파악할 수 없다는 한계점을 보인다. 즉, 주어진 데이터 집합에 대해 분류 정확도가 높은 분류기를 얻을 수 있으나, 특정 인스턴스가 왜 해당 클래스 혹은 이상치로 분류되었는지에 대한 명확한 해석이 어려운 블랙박스 형태를 갖는다는 것이다[2]. Decision tree(DT)와 같이, 명확한 규칙(rule)에 의해 데이터를 분류함으로써 사용자가 도출된 규칙을 토대로 분류 결과에 대한 요인을 파악할 수 있는 방법들도 있으나, 이러한 방법들은 규칙을 생성하기 위해 사용되는 지표들이 다중 분류에 초점이 맞춰져 있어 단일 분류에 적용되기 어렵다는 단점이 있다.

한편, 실제 산업 현장에서 발생하는 문제점들을 근본적으로 해결하기 위해서는 단순히 데이터를 분류하는 것이 아니라, 도출된 분류기에 대한 해석이 가능해야 한다. 따라서, 본 논문에서는 분류 정확도를 유지하면서도 해석력을 제공할 수 있는 단일 분류기인 hyper-rectangle(H-RTGL)을 이용한 단일 분류기를 제안한다. H-RTGL을 이용한 단일 분류기는 Jeong and Choi[9]에 의해 제안된 바 있지만, H-RTGL을 생성하기 위해서 사용되는 인터벌 생성 시 데이터의 분포가 고려되지 않는다는 단점이 존재한다. 따라서 본 논문에서는 이러한 H-RTGL 기반 단일 분류기들의 단점을 개선할 수 있는 새로운 H-RTGL 기반 단일 분류 알고리즘을 제시하고자 한다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 단일 분류에 대한 연구 동향을 기술하고, 제 3장에서는 본 논문에서 제안하는 개선된 H-RTGL 기반 단일 분류기를 설명한다. 이후, 제 4장에서 실제 데이터에 기반한 수치실험을 통해 본 논문에서 제안하는 단일 분류 알고리즘의 성능을 검증하고, 분류 요인에 대한 해석을 제공한다. 제 5장에서는 본 논문의 결론을 제시하고 추후 연구 방향을 제시한다.

2. 관련 연구 동향

지금까지 연구된 단일 분류 알고리즘은 크게 (i) 밀도 추정(density estimation) 기반 단일 분류, (ii) 결정 경계(decision boundary) 도출 기반 단일 분류, (iii) DT 기반 단일 분류, (iv) H-RTGL 기반 단일 분류 등으로 구분할 수 있다. 이러한 방법은 단일 분류를 위한 알고리즘을 도출하기 위한 기본적인 아이디어와 접근 방법에서 다음과 같은 차이점을 발견할 수 있다.

첫째, 밀도 추정을 통한 단일 분류는 주어진 데이터 집합의 확률 밀도 함수(probability density function)를 추정하고, 각 인스턴스가 해당 클래스에 속할 확률을 계산하여 분류하는 방법이다. 계산된 함수 값이 사전에 정의된 threshold 값 보다 크면 해당 클래스로, 그렇지 않으면 이상치로 분류를 수행한다. Tarassenko[15]는 유방암 검진을 위한 X선 영상에서 이상 물질을 식별하기 위해 가우시안 혼합 분포(Gaussian mixture)와 파젠(parzen) 밀도 추정을 이용하여 단일 분류 모델을 수립하였다. de Ridder et al.[6]은 가우시안 추정(Gaussian approximation)과 파젠 밀도 추정 방법을 적용하여 다른 단일 분류 알고리즘과 비교하였다. 밀도 추정 기반의 단일 분류 방식은 준수한 분류 정확도를 보장하지만 해당 클래스와 이상치 데이터를 분류하는 기준인 threshold 값을 선택하기 위한 명확한 기준이 없다는 문제점을 지닌다. 이를 극복하기 위해 Hempstalk et al.[8]은 데이터에 대한 밀도 뿐 아니라, 클래스 확률에 대한 추정을 함께 수행하고 인공 데이터

(artificial data)를 생성하여 단일 분류를 수행하는 방법을 제안하였다. 이를 통해 threshold 선정의 문제는 해결하였지만, 인공 데이터를 생성해야 하기 때문에 계산을 위한 복잡도가 상승하며 얼마나 많은 인공 데이터를 어떤 기준으로 생성할 것인지에 대한 문제가 남는다.

둘째, 데이터 집합의 패턴을 학습하여 이를 포함하는 결정 경계를 도출하여 분류하는 방법이 있다. 만약 새롭게 발생한 인스턴스가 도출된 경계면의 안쪽에 위치한다면 해당 클래스에 속한 인스턴스로, 밖에 있다면 이상치로 분류하게 된다. Tax and Duin[16, 17, 18]에 의해 제안된 Support Vector Data Description(SVDD)는 이와 같이 주어진 데이터를 바탕으로 폐곡선 형태의 분류기를 설계하는 대표적인 방법이다. SVDD는 데이터 집합을 분류하기 위해 데이터의 외곽에 존재하는 데이터들을 서포트 벡터(support vector)로 하여 서포트 벡터와 더 안쪽의 데이터 내부까지 포함하는 hyper-sphere를 생성한다. 생성된 hyper-sphere를 경계로 해당 클래스와 이상치가 분류되는 것이다. SVDD는 소수의 데이터만으로도 분류기를 생성할 수 있으며, 높은 분류 정확도를 갖는다. Schölkopf et al.[14]이 제안한 1-SVM은 원래의 SVM이 한 클래스와 다른 클래스 사이의 hyper-plane을 통해 분류를 수행하는 것과 달리 원점과 데이터 집합 사이를 나누는 hyper-plane을 찾는 방법이다. 1-SVM은 SVDD와 마찬가지로 적용하기 쉽고 높은 분류 정확도를 유지할 수 있는 방법이며, 두 방법은 매개 변수에 따라 서로 비슷하거나 동일한 결과를 나타내기도 한다[18].

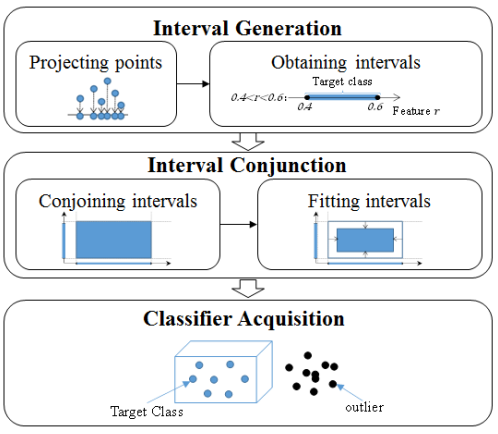
셋째, DT는 데이터 집합으로부터 규칙을 추출하고, 추출된 규칙을 각각의 노드에 할당하여 할당된 규칙을 토대로 분류를 수행한다. 이 때, 생성된 규칙은 사용자에 의해 해석될 수 있으며 이를 통해 데이터 집합에 대한 분석이 가능해진다. 그러나, 앞서 언급한 것처럼 DT의 규칙을 생성하기 위한 지표가 단일 분류에 적합하지 않다는 단점이 있다. 예를 들어, DT를 이용한 데이터 분류 기준으로 사용되는 불순도(impurity)의 경우, 하나의 노드에 서로 다른 클래스의 인스턴스들이 얼마나 존재하는지를 통해 계산된다. 이와 같은 특성을 극복하기 위해 De Comite et al.[5]과 Letouzey et al.[12]은 가장 대표적인 DT 형태인 C4.5를 이용하여 특정 클래스(정확히는 positive class)의 데이터를 단일 분류로 검출하는 연구를 수행하였다. Désir et al.[7]은 DT의 규칙이 지나치게 훈련 데이터에 과적합(overfitting)된다는 단점을 극복하기 위해 다수의 tree를 생성하여 다양한 rule을 얻을 수 있는 one-class random forests 방법을 제안하였다. 그러나 이러한 방법들은 다른 클래스의 역할을 수행할 수 있는 인공 데이터나 이상치 데이터의 역할을 수행하기 위한 클래스 불명(unlabeled) 데이터가 필요하다는 단점을 가진다.

넷째, H-RTGL 기반의 단일 분류기[9]는 데이터 집합의 각 속성에 대해 인터벌(interval) 형태의 기하학적(geometric) 규칙을 도출하고, 이러한 규칙들의 결합을 통해 결정 경계인 H-RTGL을 생성함으로써 단일 분류를 수행한다. Jeong and Choi[9]에서는 인터벌을 생성하는 방법에 따라 여러 개의 겹치는 인터벌을 병합(merging)하는 병합 기반 H-RTGL(merging based hyper-rectangle : *MbH*)와 데이터 집합을 군집화하고 군집 별로 인터벌을 생성하는 군집 기반 H-RTGL(clustering based hyper-rectangle : *CbH*)의 2가지가 제안되었다. 그러나 *MbH*는 같은 사영점을 공유하는 인스턴스가 적은 경우, 작은 H-RTGL이 매우 많이 생성되어 데이터의 산포를 반영하기 어렵다. 또한 *CbH*도 데이터를 군집화하고, 도출된 군집으로부터 바로 인터벌을 생성하기 때문에, 역시 데이터의 산포를 고려하지 않은 분류기가 생성될 위험이 있다.

3. 개선된 Hyper-Rectangle 기반 단일 분류기 설계

3.1 문제 정의 및 프레임워크

단일 분류 문제란 오직 하나의 클래스만이 존재하는 데이터 집합에 대해 특정 인스턴스가 해당 클래스에 속하는지 아니면 outlier인지 판단하는 문제라고 할 수 있다. 즉, n개의 인스턴스 $x_i(i=1, 2, \cdots, n)$로 구성된 데이터 집합 $X=\{x_1, x_2, \cdots, x_n\}$이 주어졌을 때, 해당 클래스에 소속되는 인스턴스와 outlier를 분류하는 문제이다. 이 때, 하나의 인스턴스 x_i가 m개의 속성 값을 갖는다고 가정하면, r번째 속성 값을 $y_{ir}(r=1, 2, 3, \cdots, m)$로 나타낼 때, $x_i=(y_{i1}, y_{i2}, \cdots, y_{im})$로 표현될 수 있다.



<Figure 1> Framework for Generating One-Class Classifier Using H-RTGLs

본 논문에서는 H-RTGL을 이용한 단일 분류기를 <Figure 1>과 같은 프레임워크를 따라서 생성한다. 먼저, 각 속성별로 기본적인 인터벌들을 생성하고, 도출된 인터벌들의 결합(conjunction)을 통해 기본적인 H-RTGL을 생성한다. 이렇게 생성된 H-RTGL은 각각의 속성마다 독립적으로 생성된 인터벌을 이용하였기 때문에 적당한 크기로 조정하는 피팅(fitting) 과정이 필요할 수 있다. 이를 통해 새로운 H-RTGL을 생성하면 이들을 이용하여 분류기를 생성한다. 이에 대한 자세한 절차는 제 3.2절에서 설명한다.

3.2 가우시안 기반 H-RTGL 단일 분류기 설계

본 논문에서는 앞서 언급되었던 *MbH*와 *CbH*의 단점을 개선할 수 있는 새로운 가우시안 기반 H-RTGL(Gaussian based H-RTGL : *GbH*) 단일 분류기를 제안한다. *GbH*를 이용한 단일 분류는 주어진 데이터가 특정한 가우시안 혼합(Gaussian mixture)임을 가정하고, 각각의 가우시안 분포로부터 인터벌을 생성함으로써 데이터의 산포를 고려한 분류기를 얻을 수 있다. *GbH*를 이용한 단일 분류기를 생성하는 과정은 <Figure 1>의 단계에 따라 다음과 같이 설명될 수 있다.

3.2.1 인터벌 생성

먼저, 인스턴스들을 속성 r에 대해 사영하는 함수를 다음과 같이 정의한다.

$$proj_r(x_i) = y_{ir},\ \forall i, r$$

다음으로는 각각의 속성 $r(r=1, 2, \cdots, m)$에 대해 사영점 집합을 k-means clustering을 통해 군집화한다. 군집화를 수행하기 때문에 *CbH*와 유사한 점이 있지만, 데이터 집합 전체에 대한 군집화를 수행하는 것이 아니라 한 속성에 대한 사영점들을 군집화한다는 점이 다르다. 이후, 도출된 사영점으로부터 인터벌을 생성하기 위해서는 각 사영점의 군집을 하나의 가우시안 분포와 대응시키는 것이 필요하다. 즉, 군집 내의 인스턴스들이 특정한 가우시안 분포를 따르는 데이터 집합이라고 가정하는 것이다. 이에 대한 결과로서, 속성 r에 대한 $q_r(q_r=1, 2, \cdots, k)$번째 사영점 군집 $C_r^{q_r}$에 대응되는 가우시안 분포 $g_r^{q_r}$은 다음과 같이 정의할 수 있다.

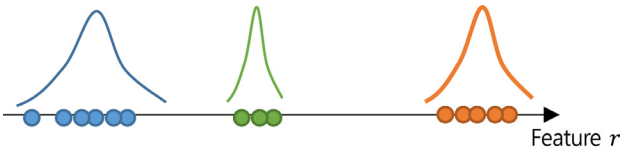
$$g_r^{q_r} \sim N(\mu_r^{q_r}, \sigma_r^{q_r})$$

여기서 $\mu_r^{q_r}$과 $\sigma_r^{q_r}$은 각각 $C_r^{q_r}$에 대응되는 가우시안 분포의 평균과 표준편차를 나타낸다.

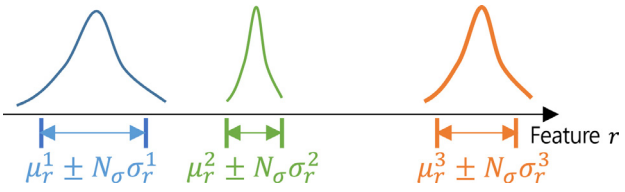
H-RTGL의 도출을 위해 필요한 인터벌은 군집 내의 인스턴스가 따르는 분포의 통계량에 기반하여 만들어진다. 즉, 속성 r의 q_r번째 사영점 군집 $C_r^{q_r}$에 대한 가우시안 분포 $g_r^{q_r}$에 의해 생성되는 인터벌은 분포의 평균 $\mu_r^{q_r}$과 표준 편차 $\sigma_r^{q_r}$에 의해 다음과 같이 정의된다.

$$ITVL_r^{g_r^{q_r}} = [\mu_r^{q_r} - N_\sigma \cdot \sigma_r^{q_r},\ \mu_r^{q_r} + N_\sigma \cdot \sigma_r^{q_r}]$$

이 때, N_σ는 가우시안 분포의 표준편차를 얼마나 반영할 지 결정하여 인터벌의 길이를 조절해주기 위한 매개변수로서 모든 사영점 군집에 대해서 동일한 값을 사용한다. <Figure 2>는 *GbH*를 이용한 단일 분류기의 인터벌 생성 과정의 예를 나타낸다. 먼저, <Figure 2>의 (a)와 같이 데이터 집합의 속성 r에 대한 사영점들을 3개의 사영점 군집 C_r^1, C_r^2, C_r^3으로 나누었다면, 이러한 정보로부터 도출되는 인터벌은 각 군집에 대응되는 가우시안 분포의 통계량인 평균 $\mu_r^{q_r}$과 표준 편차 $\sigma_r^{q_r}$을 고려하여 <Figure 2>의 (b)와 같이 생성될 수 있다. 인터벌의 길이를 결정하는 N_σ 값은 실험을 통해서 정해질 수 있다.



(a) Clustering projection points and assuming distribution



(b) Interval generation based on Gaussian statistics

<Figure 2> Interval Generation for *GbH*

주어진 데이터 집합이 각각의 속성에 대해서 k-means clustering을 통해 군집화되므로 각각의 속성에 대해 k개의 가우시안 분포가 정의되고, 총 $k \cdot m$개의 인터벌이 생성된다. 또한, *GbH*를 이용한 단일 분류기 생성을 위해서 필요한 인터벌은 몇 개의 가우시안 분포를 사용할 것인지와 분포의 평균으로부터 얼마나 넓은 구간을 인터벌에 포함시킬지에 따라 다르게 생성될 수 있다.

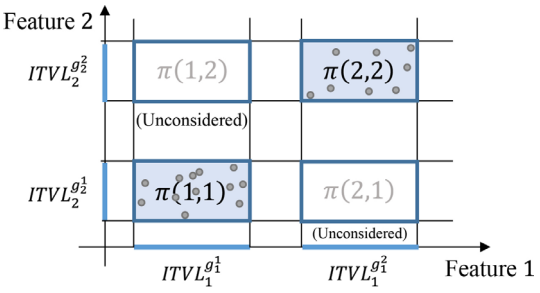
3.2.2 인터벌 결합

다음은 단계 1에서 생성된 인터벌들을 결합(conjunction)하여 H-RTGLs을 도출한다. 이를 위해 각각의 속성에 대한 인터벌 m개에 대한 m차원 곱집합(m-fold Cartesian

product) 연산을 수행한다. 이 때, 각 속성마다 k개의 인터벌이 존재하기 때문에 이들에 대한 m차원 곱집합은 최대 k^m개의 인터벌 결합을 발생시킬 수 있다. 그러나, 모든 경우에 대해 인터벌 결합을 생성하는 것은 계산 효율성을 매우 저해시키며, 인스턴스를 포함하지 않는 인터벌 결합은 의미가 없다. 따라서, 단계 2에서는 전체 가능한 인터벌 결합 중에서 모든 속성에 대하여 인스턴스가 존재하는 인터벌들의 조합$(q_1, q_2, \cdots, q_m)$만을 이용하여 생성되는 인터벌 결합 $\pi(q_1, q_2, \cdots, q_m)$를 다음과 같이 생성한다.

$$\pi(q_1, q_2, \cdots, q_m) = \prod_{r=1}^{m} ITVL_r^{g_r q_r}$$
$$= ITVL_1^{g_1 q_1} \times ITVL_2^{g_2 q_2} \times \cdots \times ITVL_m^{g_m q_m}$$

예를 들어, <Figure 3>은 2개의 속성을 가진 데이터 집합에 대해 2개의 가우시안 분포를 가정한 경우의 인터벌 결합을 나타낸다. 그러나 전체 가능한 인터벌 결합 4가지 중에서, $\pi(1,2)$와 $\pi(2,1)$은 인스턴스를 포함하고 있지 않기 때문에 단계 2에서 인터벌 결합으로 생성되지 않게 된다.



<Figure 3> An Example of Interval Conjunction

한편, 인터벌 결합 $\pi(q_1, q_2, \cdots, q_m)$은 다수의 속성을 반영하는 H-RTGL 형태를 가지고 있으나, 각 속성마다 독립적으로 인터벌을 생성하고 그에 대한 논리곱을 취하여 도출되었기 때문에 데이터 집합에 대한 과적합(overfitting) 문제를 야기할 수 있다. 따라서, 데이터 집합의 통계량과는 독립적으로 H-RTGL의 크기를 조정하여 과적합 문제를 해결할 필요가 있다. 이를 위해, 각 속성별로 인터벌 길이를 조정함으로써 H-RTGL의 부피를 결정하는 매개 변수 v를 도입하여 피팅(fitting) 과정을 수행한다. 인터벌 결합 $\pi(q_1, q_2, \cdots, q_m)$의 속성 r에 대한 피팅 값은 다음과 같이 얻을 수 있다.

$$Fit_r(\pi(q_1, q_2, \cdots, q_m))$$
$$= [\min(ITVL_r^{g_r q_r}) - v, \max(ITVL_r^{g_r q_r}) + v], \ \forall r$$

3.2.3 분류기 생성

모든 속성에 대한 피팅 과정을 수행하고, 피팅 값들의 논리곱을 통해 최종적으로 다음과 같이 각 인터벌 결합 $\pi(q_1, q_2, \cdots, q_m)$로부터 GbH를 정의한다.

$$GbH(\pi(q_1, q_2, \cdots, q_m)) = Fit_1(\pi(q_1, q_2, \cdots, q_m))$$
$$\wedge Fit_2(\pi(q_1, q_2, \cdots, q_m))$$
$$\wedge Fit_m(\pi(q_1, q_2, \cdots, q_m))$$

정의된 GbH를 바탕으로, 새롭게 발생한 인스턴스가 GbH의 안쪽에 위치하면 해당 클래스에 속한 것으로 판정하고, 그렇지 않은 경우 이상치로 결정하는 단일 분류기를 생성할 수 있다.

4. 수치 실험을 통한 분류기 검증

4.1 실험 설계

본 논문에서 제안된 가우시안 기반 H-RTGL 단일 분류기 GbH의 성능을 평가하기 위해서 UCI machine learning repository[1]에서 제공되는 데이터 집합들을 이용한 수치 실험을 설계하였다. 사용된 데이터 집합은 Iris, Breast, Liver, Biomed의 4가지였으며, 각 데이터 집합에 대한 정보는 <Table 1>과 같다. 그러나 이러한 데이터들은 2개 이상의 클래스를 포함하고 있기 때문에, 단일 분류 실험을 위한 목적으로 바로 사용될 수가 없다. 따라서 본 논문에서는 각각의 데이터 집합에 대해서 클래스들 중 하나를 분류의 목적이 되는 목표 클래스(target class)로 정의하고 나머지 클래스들은 이상치 데이터로 분류하는 One-versus-All(OvA) 방식을 사용하였다. 또한, 실험의 다양성을 위해서 Iris 데이터 집합에 대해서는 3가지 클래스 모두를 각각 목표 클래스로 사용한 3번의 독립적인 실험을 수행하였다.

<Table 1> Summary of Datasets Used from UCI Repository

Datasets	Iris	Breast	Biomed	Liver
Number of attributes(m)	4	9	5	6
Target class	(Various)	Malignant	Normal	Healthy
Size of target class	50	241	127	145
Size of outliers	100	458	67	200

일반적인 분류 문제에서 분류기 생성을 위한 학습을 수행하기 위해 사용되는 데이터를 훈련 데이터(training data)라고 하며, 생성된 분류기의 성능 측정을 위해 사용

되는 데이터를 시험 데이터(test data)라고 한다. 본 논문에서는 목표 클래스에 속한 데이터 중 50%를 임의로 추출하여 훈련 데이터로 사용하였다. 또한, 훈련 데이터로 선정되지 않은 목표 클래스의 데이터와 이상치는 모두 시험 데이터로 사용하였다. 예를 들면, Breast 데이터 집합에 대한 실험의 경우, 목표 클래스에 속한 241개의 인스턴스 중에서 50%인 120개의 인스턴스를 학습 데이터로 사용하였고, 목표 클래스에 있지만 선택되지 않은 나머지 121개의 인스턴스와 458개의 이상치를 모두 시험 데이터로 사용하였다. 이와 같이 시험 데이터를 목표 클래스와 이상치를 모두 포함하도록 정의함으로써, 분류기가 목표 클래스의 데이터를 얼마나 잘 반영시키고 이상치 데이터를 얼마나 잘 배제시킬 수 있는지 파악할 수 있다.

제안된 단일 분류기의 성능을 검증하기 위한 성능 지표로 Area Under the Receiver Operating Characteristics curve (AUC)를 채택하였다. Receiver Operating Characteristics(ROC) 곡선은 분류 모델이 양의 데이터로 분류한 인스턴스 중 실제로 양의 데이터의 비율인 true positive rate(TPR)과 실제로는 음의 데이터이지만 양의 데이터로 분류된 인스턴스의 비율인 false positive rate(FPR)를 기반으로 그려진다. 이처럼 ROC 곡선은 2가지 요소에 대해 모두 고려하기 때문에, 분류 정확도뿐 아니라 모델이 얼마나 효율적인지를 검증하기 위해서도 사용 가능하다. 만일, 어떤 분류기가 TPR이 매우 높은 분류 결과를 나타내지만, FPR 또한 동시에 높다면 이 분류기를 분류 성능이 좋은 분류기라고 할 수 없다. 즉, 검출해야 하는 인스턴스를 포함하는 비율과 더불어 배제해야 하는 인스턴스를 포함하고 있지 않은 비율에 대해서도 함께 고려해야 한다. 일반적으로, AUC는 이러한 조건에 부합하는 성능 지표이며, 분류기의 AUC가 1에 가까울수록 바람직한 분류 모델이라고 할 수 있다.

본 논문에서는 *GbH* 생성을 위해 고려되는 매개 변수 값들을 다양하게 변화시키면서 다양한 성능의 분류기와 그에 대응하는 ROC 곡선을 생성하였다. 보다 구체적으로는 인터벌 생성 시 인터벌의 길이를 결정하는 매개 변수 v의 값을 변화시키며 ROC 곡선을 그렸다. 만약 v가 0의 값을 갖는다면, 생성되는 인터벌들이 피팅 함수에 의해 학습 데이터 집합에 매우 과적합된 H-RTGL이 생성될 수 있다. 반대로 v가 매우 큰 값을 갖는다면, 목표 클래스의 데이터가 아닌 이상치까지 무분별하게 포함하는 매우 큰 H-RTGL이 생성될 수 있다. 이와 같이 인터벌 생성을 위한 핵심 매개 변수인 v의 값을 변화시키면 생성되는 H-RTGL의 크기를 조절할 수 있으며, 목표 클래스의 데이터를 더 많이 포함시키는 분류기를 생성할 수 있다. 따라서, 본 논문에서는 v의 값을 점차 증가시키며 ROC 곡선을 그리되, 시험 데이터 중 목표 클래스의 데이터가 모두 포함되었다면 실험을 종료하였다.

*GbH*에는 v 외에도 가정되는 가우시안 분포의 개수 k와 해당 가우시안 분포의 평균 값을 중심으로 얼마나 넓은 인터벌을 생성할 지 결정하는 매개 변수 N_σ가 존재한다. 따라서, 이 2가지 매개 변수들의 값을 바꿔가며 ROC 곡선을 그림으로써 가장 좋은 AUC 값을 나타내는 매개 변수 조합을 찾을 수 있도록 하였다.

<Table 2> Experimental Results of *GbH*-based One-Class Classifier Compared to Others

Datasets / Classifiers	Iris (Setosa)	Iris (Vesricolor)	Iris (Virginica)	Breast	Biomed	Liver
	AUC×100(standard deviation)					
GbH	100(0.0) ($k=1$, $N_\sigma=1.00$, $T=7ms$)	99.1(0.8) ($k=2$, $N_\sigma=0.87$, $T=10ms$)	97.9(0.9) ($k=2$, $N_\sigma=1.18$, $T=12ms$)	95.9(0.8) ($k=3$, $N_\sigma=1.46$, $T=11ms$)	89.4(1.5) ($k=4$, $N_\sigma=1.15$, $T=21ms$)	61.3(1.5) ($k=4$, $N_\sigma=2.38$, $T=31ms$)
MbH	100(0.0)	98.5(0.8)	96.1(1.1)	95.1(1.2)	89.3(1.2)	61.6(1.6)
CbH	100(0.0)	98.6(0.8)	93.9(1.4)	85.8(11.1)	88.3(5.3)	60.3(2.7)
Naïve Parzen	100(0.0)	98.3(0.6)	95.4(1.1)	96.5(0.4)	93.1(0.2)	61.4(0.7)
Parzen	100(0.0)	99.0(0.3)	96.5(0.8)	72.3(0.5)	90.0(1.1)	59.0(0.3)
k-means	100(0.0)	98.4(1.0)	95.4(0.5)	84.6(3.5)	87.8(1.2)	57.8(1.0)
1-NN	100(0.0)	98.3(0.2)	97.0(0.8)	69.4(0.6)	89.1(0.8)	59.0(0.9)
Auto-Encoder	100(0.0)	97.3(0.5)	95.6(1.4)	38.4(0.9)	85.6(2.2)	56.4(0.9)
PCA	97.3(0.8)	92.6(2.4)	90.9(4.7)	30.3(1.0)	89.7(0.5)	54.9(0.5)
SOM	100(0.0)	98.3(0.5)	96.6(0.3)	79.0(2.3)	88.7(0.8)	59.6(0.7)
MST_CD	100(0.0)	98.5(0.1)	97.0(0.7)	75.6(1.8)	89.8(1.0)	58.0(0.9)
k-Centers	100(0.0)	97.3(1.0)	96.9(0.7)	71.5(12.4)	87.8(2.4)	53.7(4.1)
SVDD	100(0.0)	98.2(0.6)	98.1(0.8)	70.0(0.6)	2.2(0.3)	4.7(1.4)
LPDD	100(0.0)	97.8(0.5)	98.6(0.4)	80.0(0.5)	86.5(2.6)	56.4(2.6)

4.2 실험 결과 분석

<Table 2>는 본 논문에서 제안된 *GbH*를 이용한 단일 분류기를 <Table 1>의 4가지 데이터 집합에 대해 각각 20번씩 적용한 결과와 동일 데이터 집합에 대하여 적용된 다른 단일 분류기들의 성능을 나타낸다. *GbH* 기반 단일 분류기의 경우, 제 4.1절에서 설명된 방법을 적용하여 구한 가장 높은 AUC 값을 갖는 k와 N_σ 값을 표현하였으며, 이러한 환경에서 각 시행에 대해 계산된 20회의 AUC 값에 대한 평균과 표준 편차를 기록하였다. 특히, *GbH* 기반 단일 분류기 생성을 위해 소요되는 시간 T 값은 데이터 집합의 크기 변화에 따라 큰 변동이 없었으며, 이를 통해 빠른 시간에 제안된 알고리즘이 수행됨을 알 수 있었다. 다른 분류 알고리즘의 AUC는 [9, 10]을 참고하여 기록하였다. 해당 논문에서는 MATLAB toolbox의 일종인 Data Description toolbox(dd_tools)를 통한 매개 변수 최적화를 수행하여 가장 최적의 AUC 값을 기록하였다.

제안된 *GbH*를 이용한 단일 분류기는 Parzen, Naïve Parzen과 더불어 가장 높은 수준의 AUC 값을 가졌다. 이를 통해, *GbH* 기반 단일 분류기가 복잡도가 높은 방법들과 비교하였을 때 분류 정확도의 측면에서 결코 부족하지 않음을 알 수 있었다. 또한, 기존에 제안되었던 *MbH*와 *CbH*를 이용한 단일 분류기와 비교했을 때, 개선된 AUC 값을 나타냄을 알 수 있었다. 결과적으로, 본 논문에서 제안한 단일 분류기가 높은 분류 정확도와 강건성(robustness)까지 갖추고 있음을 검증할 수 있었다. 특히, 데이터 집합의 크기가 비교적 작은 편인 Iris 데이터 집합에서는 모든 목표 클래스에 대해 가장 좋은 AUC 값을 보였다.

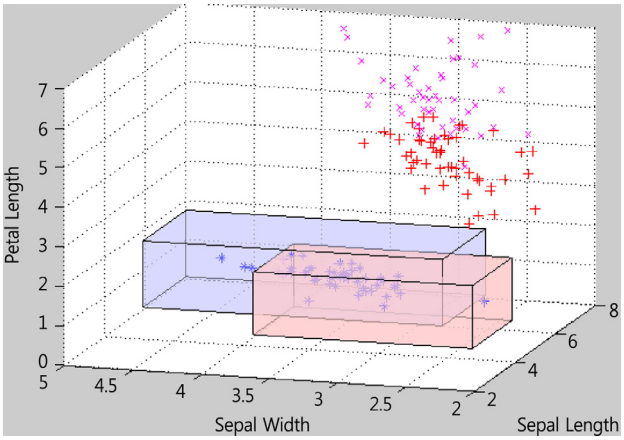
매개 변수의 측면에서는 데이터 집합에 따라 최적 매개 변수 k와 N_σ 값이 달라지고 있는 것을 알 수 있는데, 이는 *GbH*를 이용한 단일 분류기가 매개 변수를 조정하여 데이터의 산포를 반영할 수 있음을 나타낸다. 예를 들면, Iris 데이터 집합에서는 2개 이내의 가우시안 분포를 가정할 때 높은 AUC 값을 얻을 수 있었다. 그러나 Biomed 데이터 집합에서는 4개의 가우시안 분포를 가정할 때 높은 AUC 값을 얻을 수 있었다. 데이터가 분산되어 있는 정도를 반영할 수 있는 매개 변수 N_σ에 대해서도 데이터 집합에 따른 변화가 나타났다. Iris 데이터 집합은 0.8~1.2σ를 기준으로 인터벌을 생성할 때 가장 높은 AUC 값을 나타낸 반면, Breast 데이터 집합에서는 1.46σ를 기준으로 생성된 인터벌이 가장 높은 AUC 값을 보였다.

4.3 분류기의 해석력 검증

본 논문에서 제안하는 *GbH*를 이용한 단일 분류기는 높은 분류 정확도를 보장하면서도 분류 결과에 대한 해석이 함께 가능하다는 장점을 갖는다. 분류에 대한 해석은 DT와 같이 해석력을 제공하는 알고리즘을 사용하거나, 해석력을 제공하지 않는 방법들로부터 규칙을 추출하여 사용하는 방법이 있다[3]. *GbH*를 이용한 단일 분류기에서는 인터벌이 규칙의 역할을 수행한다. 즉, 사용자는 각 속성들로부터 생성된 인터벌들을 이용하여 데이터 집합에 대한 해석이 가능하다.

<Figure 4>는 Iris 데이터 집합의 Setosa 종을 목표 클래스로 하여 2개의 가우시안 분포를 가정한 *GbH* 결과를 나타낸다. 가시화를 위해 Petal Length, Sepal Width, Sepal Length 등의 3가지 속성만으로 그래프를 그렸으나, 실제로 *GbH* 생성 과정에서는 4가지 속성이 모두 고려되었다.

<Figure 4>에서 생성된 2개의 *GbH*는 각각 <Table 3>과 같은 속성 값을 갖는다. 이 때, GbH_1이 뒤쪽, GbH_2가 앞쪽에 있는 H-RTGL을 나타낸다. 이러한 2개의 *GbH*는 모든 Setosa 종의 인스턴스를 포함하고 있으며, 이는 각각의 *GbH*를 구성하는 인터벌이 Setosa 종의 대표적인 패턴임을 나타낸다. 예를 들어, Sepal Width의 값이 2.7과 5.0 사이에 있고, Sepal Length가 5.3과 7.8 사이에 있으며, Petal Length는 1.9보다 작은 미식별된 Iris 인스턴스가 있다면 해당 인스턴스는 Setosa 종이라고 할 수 있다. 특히, <Figure 4>를 보면, Sepal Width나 Sepal Length에 대한 사영을 수행할 경우 (x-y 평면) setosa 종과 다른 종들이 구분되지 않지만 Petal Length에 대한 사영 (z축)을 수행하면 다른 종들과 setosa 종이 매우 유의미한 차이를 보인다. 이를 통해 Petal Length는 다른 종들과 Setosa 종을 구분 짓는 핵심 특성임을 파악할 수 있다. 이처럼, *GbH*를 통해서 목표 클래스의 인스턴스들이 갖는 특정한 속성 값 구간을 도출하거나, 목표 클래스만의 고유한 핵심 특성을 분석할 수 있다. 즉, 인터벌이 해석력 제공을 위한 규칙의 역할을 수행하고 있는 것이다.



<Figure 4> *GbH* Generation Result from Iris Dataset

<Table 3> Interpretation of *GbH* from Iris dataset

	GbH_1	GbH_2
Petal Length	0.0~1.9	0.0~1.9
Sepal Width	2.7~5.0	2.4~4.0
Sepal Length	5.3~7.8	4.2~6.3

추가적인 예시로서, Breast 데이터 집합에 대해 <Table 4>와 같이 3개의 *GbH*를 생성하였다.

<Table 4> Interpretation of *GbH* from Breast Dataset

	GbH_1	GbH_2	GbH_3
Feature 1	2.8~10.2	1.9~9.9	2.0~10.1
Feature 2	1.8~9.8	3.2~7.5	1.8~9.8
Feature 3	1.9~10.2	2.8~9.8	2.7~10.1
Feature 4	0.9~10.9	0.8~10.4	1.0~10.3

다른 특성 값들에 비해, 특성 4에서 나타난 인터벌들의 값이 공통적으로 데이터 스케일(1-10)의 모든 값을 포함하여 매우 넓은 인터벌을 가짐을 확인할 수 있다. 이는 특성 4가 학습 데이터의 특정 패턴을 규정하는 데에 사용되고 있지 못하며, 데이터의 특성을 결정짓는 주요한 속성이 아님을 알 수 있다. 실제로, 특성 4는 Marginal Adhesion으로서 추후 연구를 통해 악성 유방암의 유발 요인이 아님이 밝혀졌다. 이를 통해 *GbH*가 학습 데이터의 패턴을 올바르게 반영하고 있으며, *GbH*의 기반이 되는 인터벌이 분류 결과의 해석을 위한 규칙으로 사용될 수 있음을 추가로 확인할 수 있었다.

5. 결 론

본 논문에서는 산업 현장에서 점차 중요성이 대두되고 있는 불균형 데이터에 대한 클래스 분류와 같은 단일 분류 문제를 위한 효율적인 알고리즘을 제안하였다. 특히, 분류 정확도와 더불어 분류 요인에 대한 해석력까지 함께 제공할 수 있는 H-RTGL 기반 단일 분류기인 *GbH*를 설계하였다. *GbH*는 주어진 인스턴스들이 특정한 가우시안 분포를 따른다고 가정하며, 가우시안 분포의 통계량인 평균과 표준 편차를 이용하여 인터벌을 생성, 결합하여 H-RTGL을 얻는다. 결과적으로, *GbH*를 이용한 단일 분류기는 기존에 제안되었던 H-RTGL 기반의 분류 방법인 *MbH*와 *GbH*를 이용한 단일 분류기보다 개선된 분류 정확도를 보였으며, 분류 요인에 대한 해석 또한 가능했다.

본 논문에서 제안된 *GbH*는 비교적 간단한 매개 변수 조합을 통해 생성되었다. 따라서, 추후 연구로서 인터벌 생성에 영향을 줄 수 있는 추가적인 매개 변수에 대한 탐색이 이루어질 수 있다. 또한, 사용할 분포의 수나 표준 편차에 대해서도 임의 탐색이 아니라 데이터의 패턴에 따른 탐색을 수행할 수 있는 방법론이나 k-means가 아닌 다른 군집화 방법 등을 함께 고려할 수 있다.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. NRF-2017R1A2B4009841) and by the Ajou University research fund.

References

[1] Asuncion, A. and Newman, D., UCI machine learning repository, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[2] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Muller, K.R., How to explain individual classification decisions, *The Journal of Machine Learning Research*, 2010, Vol. 11, pp. 1803-1831.

[3] Barakat, N. and Bradley, A.P., Rule extraction from support vector machines : a review, *Neurocomputing*, 2010, Vol. 74, No. 1-3, pp. 178-190.

[4] Cortes, C. and Vapnik, V., Support-vector networks, *Machine Learning*, 1995, Vol. 20, No. 3, pp. 273-297.

[5] De Comite, F., Denis, F., Gilleron, R., and Letouzey, F., Positive and unlabeled examples help learning, *Proceedings of International Conference on Algorithmic Learning Theory*, 1999, Berlin, Germany, pp. 219-230.

[6] De Ridder, D., Tax, D., and Duin, R.P., An experimental comparison of one-class classification methods, *the 4th Annual Conference of the Advanced School for Computing and Imaging*, 1998, Delft, Netherlands.

[7] Desir, C., Bernard, S., Petitjean, C., and Heutte, L., A random forest based approach for one class classification in medical imaging, *Machine Learning in Medical Imaging, Lecture Notes in Computer Science*, 2012, Vol. 7588, pp. 250-257.

[8] Hempstalk, K., Frank, E., and Witten, I.H., One-class classification by combining density and class probability estimation, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008, Berlin, Germany, pp. 505-519.

[9] Jeong, I.K. and Choi, J.Y., Design of One-Class Classi-

fier Using Hyper-Rectangles, *Journal of the Korean Institute of Industrial Engineers*, 2015, Vol. 41, No. 5, pp. 439-446.

[10] Juszczak, P., Tax, D.M., Pe, E., and Duin, R.P., Minimum spanning tree based one-class classifier, *Neurocomputing*, 2009, Vol. 72, No. 7-9, pp. 1859-1869.

[11] Kang, B.S. and Kim, S.S., Combined Artificial Bee Colony for Data Clustering, *Journal of Society of Korea Industrial and Systems Engineering*, 2017, Vol. 40, No. 4, pp. 203-210.

[12] Letouzey, F., Denis, F., and Gilleron, R., Learning from positive and unlabeled examples, *Proceedings of 10th International Conference on Algorithmic Learning Theory*, Berlin, German, 2000, pp. 71-85.

[13] Park, Y.J., Kim, G.Y., and Jang, S.W., Traffic Anomaly Identification Using Multi-Class Support Vector Machine, *Journal of the Korea Academia-Industrial Cooperation Society*, 2013, Vol. 14, No. 4, pp. 1942-1950.

[14] Schölkopf, B., Williamson, R., Smola, A., Taylor, J.S., and Platt, J., Support vector method for novelty detection, *Advances in Neural Information Processing Systems*, 2000, Vol. 12, pp. 582-588.

[15] Tarassenko, L., Hayton, P., Cerneaz, N., and Brady, M., Novelty detection for the identification of masses in mammograms, *4th International Conference on Artificial Neural Networks*, 1995, pp. 442-447.

[16] Tax, D.M.J. and Duin, R.P.W., Data domain description using support vectors, *Proceedings of European Symposium on Artificial Neural Networks*, 1999a, Brussels, Belgium, pp. 251-256.

[17] Tax, D.M.J. and Duin, R.P.W., Support vector domain description, *Pattern Recognition Letters*, 1999b, Vol. 20, pp. 1191-1199.

[18] Tax, D.M.J., One-class Classification, [dissertation], [Delft, Netherlands] : Delft University of Technology, 2001.

ORCID

Do Gyun Kim | http://orcid.org/0000-0002-5149-9417
Jin Young Choi | http://orcid.org/0000-0001-6397-3107
Jeonghan Ko | http://orcid.org/0000-0002-2625-7889