One-class Classification in Multimodal Biometric Authentication

Panos Liatsis1, Quang Duc Tran2

1Department of Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
2Department of Data Communications and Computer Networks, Hanoi University of Science and Technology
Hanoi, Vietnam
1pliatsis@pi.ac.ae, 2ductq@soict.hust.edu.vn

Abstract: Class imbalance is a major challenge in biometric authentication, particularly in the context of two-class classification, i.e., distinguishing between genuine users and impostors. Traditional classifiers assume near-balanced class distributions and as such they do not work well when the samples of one class outnumber those of the other. Indeed, class imbalance is a common problem in multimodal biometrics, where typically impostor samples are in the order of 500:1 compared to those of genuine users. In this work, we present the use of one-class classification to enhance multimodal biometric performance in the presence of class imbalance. We consider well-known one-class classifiers, such as the Gaussian Mixture Model, k-Nearest Neighbour, etc in learning the user-specific and user-independent descriptions for the biometric decision inference. We conclude that the user-specific approach is powerful in overcoming the within-class sub-concepts problem, which commonly occurs in multimodal biometrics due to user variation.

Keywords: Multimodal biometrics, One-class classification, User-independent description, User-specific description, Within-class sub-concepts problem.

I. INTRODUCTION

Unimodal biometric authentication systems suffer with a variety of problems including non-universality, noise in the sensing data, intra-class variations, inter-class similarities and spoof attacks [1]. These challenges gave rise to multi-modal biometric authentication systems, which seek to alleviate some of the aforementioned problems by reconciling multiple, fairly independent sources of evidence. Such systems tend to be robust against individual sensor failure, solving the problem of non-universality, and deterring spoof attacks [1], [2]. Different sensors, algorithms, instances, samples, and/or biometric modalities can be used to obtain multi-biometric data [1], [2], [3]. Various fusion strategies can be performed on the multiple sources of evidence at the various levels of the biometric system, i.e., sensor, feature, match score and decision. Sensor and feature level fusion are generally expected to be most effective, since they contain richer information about the biometric data. However, this is challenging as most commercial biometric systems do not provide access to the raw data (and their features) for the security purposes, but also because the feature sets of the different modalities may be non-homogenous or incompatible [1], [2]. On the other hand, decision level fusion is considered to be the least informative [2]. In this work, we focus on match score level fusion, as it is relative easy to access and allows the combination of match scores generated by the different matchers.

A common practice in many reported works on multimodal biometrics is to view fusion at match score level as a two-class classification problem, where the vector of match scores is treated as a feature vector, and thus, can be classified into one of two classes, i.e., genuine user and impostor. Next, based on the training set of match scores, the classifier learns the decision boundary between the two classes [1]. The decision boundary can be learned at the operational stage, regardless of the claimed identity [4] or for each user, enrolled in the system [5]. A number of two-class classifiers, such as HyperBF [4], k-Nearest Neighbours using vector quantization [6], C4.5 decision tree, Fisher linear discriminant, Bayesian classifier [7], [8], multilayer perceptrons [7], Support Vector Machines (SVM) [7], linear classifiers [5], and discriminative classifiers based on reduced polynomial expansion [9] have been used to render the decision in a multimodal biometric verification system. However, the performance of most conventional two-class classifiers deteriorates, when they are applied to problems characterised by class imbalance [10]. Class imbalance is a common problem to many application domains, including multimodal biometric authentication. It is not uncommon for the class imbalance to be in the order between 49:1 and 500:1.

II. LITERATURE REVIEW

Most conventional two-class classifiers assume or expect balanced class distributions, and generally create suboptimal classification models in the presence of complex imbalanced data sets. In particular, Bayesian networks learn, according to certain scoring functions, to approximate the dependency patterns, which dominate the data, while those in the minority class are usually hard to be encoded [11]. In [11], it was reported that Back propagation (BP) and Radial Basis Function (RBF) neural networks may perform sub-optimally with imbalanced data sets, since the contribution of the minority class is inadequately weighted in the networks. K-Nearest Neighbours was shown to give higher probabilities to samples from the prevalent class, and hence, the test cases
from the minority class are prone to being incorrectly classified [11]. Support Vector Machines (SVM) has significantly advanced decision boundary design. SVM tries to minimize the total error, which is inherently biased towards the prevalent class, since the samples of the minority class tend to be relatively far from the decision boundary, thus contributing less to the final hypothesis [10]. The most obvious characteristic of data sets with imbalanced class distributions is the skewed data distribution between classes. However, it has been shown that a skewed data distribution is not the only negative effect of class imbalance on classification performance [10].

Since one-class classification is capable of learning the description using samples exclusively from one-class, it is known to be naturally quite robust to the class imbalance, which poses serious difficulties to traditional two-class classifiers [13]. In [14], the authors suggested that one-class classifiers are particularly useful in handling highly imbalanced data sets with high feature space dimensionality, while two-class classifiers are more suitable for a relatively moderate imbalance data sets. In [15], a number of one-class and two-class classifiers were evaluated on a selection of credit score datasets. It is important to note that one-class classifiers offer a viable solution to the low-default portfolio problem when the minority class constitutes approximately 4% of less of the data (i.e., the imbalance rate is severe). In [16], it was observed that the auto-encoder may be superior to the MLP neural networks under certain conditions, such as multimodal domains.

III. OVERVIEW OF ONE-CLASS CLASSIFIERS

In this section, we will present some representative one-class classifiers, i.e., Gaussian Mixture Model, k-Nearest Neighbour (k-NN), K-means clustering (K-means), and Support Vector Data Description (SVDD), which can be employed the render the biometric decision. It should be note that all the one-class classifiers require the fraction rejection $f_T$, which has to be specified by the user. By definition, this parameter controls the percentage of target samples, rejected by the classifier during training. Without loss of generality, we assume that $w_T$ is the target class, $x_T$ is a test object, $||x||$ is the Euclidean length of vector $x$, and $p(x_T, \mu_j, \Sigma_j)$ is the Gaussian distribution characterized by mean $\mu_j$ and covariance matrix $\Sigma_j$. These notations will be used throughout this work.

A. Gaussian Mixture Models

Gaussian Mixture Models (GMM) are often employed in multimodal biometric systems due to their ability of forming smooth approximations to arbitrary shape densities of the match scores. The Gaussian Mixture Model (GMM) is a linear combination of Gaussian distributions as follows:

$$p_{GMM}(x_T | w_T) = \frac{1}{N_{GMM}} \sum_{j=1}^{N_{GMM}} \alpha_j p(x_T, \mu_j, \Sigma_j)$$

(1)

where $\alpha_j$ are the mixing coefficients, $N_{GMM}$ is the number of mixture components, which is used to model the match score.
distributions of either genuine user class or impostor class. When the number of mixture components $N_{\text{GMM}}$ is known, $\mu_j$ and $\Sigma_j$ of the individual mixture components can be estimated using the Expectation Maximization (EM) algorithm [17]. The value of $N_{\text{GMM}}$ can be determined through cross validation and grid search [17].

**B. K-Nearest Neighbors**

When limited training data is available, k-Nearest Neighbors (k-NN) [44-13] is preferred to GMM. In general, k-NN can be derived from local density estimation as follows:

$$p_{k-\text{NN}}(x_{\tau}|w_{\tau}) = \frac{k}{N} \frac{1}{V_k(||x_{\tau} - NN_k(x_{\tau})||)}$$

(2)

where $N$ is the number of training objects, $NN_k(x_{\tau})$ is the $k^{th}$ Nearest Neighbor of the test pattern $x_{\tau}$ in the training set, and $V_k$ is the volume surrounding $x_{\tau}$. This algorithm requires the user to define the number of nearest neighbors $k$, and heavily relies on the distance between the objects as it is sensitive to the scaling of match scores, provided by multiple matchers [13].

There is little computational cost in training k-NN, but testing is expensive. Such classifiers require all the training samples to be stored and, during testing, distances to all training samples have to be calculated and sorted [13]. This effect can be offset by constructing tree-based search structures, which allow for the nearest neighbors to be found efficiently without carrying out an exhaustive search on the training data [17].

**C. K-Nearest Neighbors**

K-means clustering (K-means) [17] is one of the simplest methods for one-class classification. It works by clustering the data into groups and described by a set of prototype objects $\mu_k$. The position of these prototype objects is determined by minimizing the following error:

$$e_{k-\text{means}} = \sum_{t} \min_{k} ||x_t - \mu_k||^2$$

(3)

The classification of a test object $x_{\tau}$ is based on its distance to the nearest prototype object, defined as:

$$d_{k-\text{means}}(x_{\tau}|w_{\tau}) = \min_{k} ||x_{\tau} - \mu_k||^2$$

(4)

K-means uses the squared Euclidean distance as the measure of similarity between the test and prototype objects. Not only does this limit the scaling of data, which can be considered, but it may also impact on the robustness of the cluster means estimation, with regards to outliers [17]. The error in (3) can be then minimized using a so-called batch algorithm, which is comparable to the EM algorithm of GMM [13]. A distinct advantage of K-means over other one-class classifiers is that it has no free parameters, which have to be specified by the user apart from the fraction rejection.

![Fig. 2. User-independent descriptions around the impostor class (red continuous lines) and genuine user class (blue continuous lines), learned by (a) GMM, (b) k-NN, (c) K-means, and (d) SVDD using the samples from face and iris modalities of the BioSecure DS2 database.](image)
B. Support Vector Data Description

Support Vector Data Description (SVDD) [13] aims to directly fit a closed boundary around the target data set, without estimating a complete probability density. Thus, such a method has the ability to obtain the data boundary from a limited data set. The boundary can then be described by a few training objects, known as, the support vectors. In order to obtain more flexible data description, SVDD replaces the normal inner products by kernel functions, $K(x_i, x_j)$.

Polynomial and Radial Basis Functions (RBF) are the most common kernel functions used in practice. The classification of a test object $x_T$ is then based on its distance from the center of the hypersphere, which is calculated as follows:

$$
\begin{align*}
    d_{SVDD}(x_T | \omega_T) &= \|x_T - a\|^2 \\
    &= K(x_T, x_T) - 2 \sum_i \alpha_i K(x_T, x_i) + \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j)
\end{align*}
$$

(5)

where $a = \sum \alpha_i x_i$ is the center of the sphere, and $\alpha_i \geq 0$ are the Lagrange multipliers, which can be determined as the solution of a Quadratic Programming problem [13]. Obviously, the center of the sphere is a linear combination of the samples with weights $\alpha_i$. Since a large fraction of the weights are close to zero, the description can be characterized by a few samples with positive weights. These samples are called the support vectors of the description. It was observed that when an insufficient number of samples is available, the number of support vectors remains high, indicating that more data is necessary. Hence, for very small sample sizes, the SVDD breaks down due to its requirement for support vectors [13].

In general, SVDD is different from the v-SVM, proposed in [18], for one-class classification. While the former computes a closed hypersphere around the data, the latter estimates a largest margin hyperplane, used to separate the data and the origin of the space, where the data resides. However, it has been shown that when RBF is used as the kernel function, these methods are equivalent and produce similar performances [13], [18].

IV. MULTIMODAL FUSION USING ONE-CLASS CLASSIFIERS

Essentially, one-class classifiers can provide two types of outcomes, i.e., (1) a distance, $d(x_T | \omega_T)$, or (2) a probability estimate, $p(x_T | \omega_T)$, of the test object $x_T$ to the target class. K-means and SVDD are examples of classifiers belonging to the first group, while GMM and k-NN are examples of classifiers of the second group. Assume that $x_T$ is the test match score vector, $k \in \{G, I\}$ denotes the genuine user and/or imposter classes. This section will explain how to design
user-independent and user-specific score fusion based on one-class classifiers. Illustrative examples are also presented to highlight the advantages of user-specific fusion over the user-independent counterpart.

A. User-independent Score Fusion

In user-independent score fusion, a single description is learned around the target class $w_k$ using the match score vectors from a number of different users. Since the target class can be either impostor or genuine user class, one-class classifiers applied separately to each of these classes can produce two different descriptions. Fig. 2 shows graphical representations of the decision boundaries, learned by GMM, k-NN, K-means and SVDD. Specifically, the red continuous lines correspond to the descriptions of the impostor class, while the blue continuous lines correspond to those of the genuine user class. Obviously, each of the one-class classifiers has a different ability to learn the characteristics of the training data, leading to a difference (including sometimes contrary/overlapped sets) in their description, and the associated error rates. Since samples of the two classes are scattered into several small regions, the decision boundaries enclose a large empty area, which could augment the probability of accepting outliers. This situation has been referred to as the problem of within-class sub-concepts [11], which becomes even more notable when the description is learned around the genuine class. As illustrated in Fig. 2, a large number of impostor samples are accepted by such description, leading a significantly higher error rates.

In Figs. 3 and 4, we provide examples of the distributions of the outcomes of GMM and K-means in the cases, when the descriptions are learned around the impostor and the genuine user classes. The red solid line corresponds to the distribution of the outcomes, obtained when classifying the test samples from the impostor class. The blue solid line corresponds to the distribution of the outcomes for those samples from the genuine user class. The verification performance depends much on the overlapping zone between the distributions of these outcomes. It is clear that impostor class provides a more reliable source of information. Its corresponding description is able to produce a much smaller overlapping zone (see Figs. 3(a) and 4(a)), compared to that when the description is learned around the genuine user class, as illustrated in Figs. 3(b) and 4(b).

In general, the combined match scores can be directly obtained based on the outcomes of the one-class classifiers. It should be noted that due to the choice of the target class, such match scores, even when generated by the same one-class method, are inhomogeneous. When the target class is impostor $w_i$ (see Figs. 3(a) and 4(a)), user-independent score fusion produces low probability or high distance for a test object $x_T$ of a genuine user, while higher probability or lower distance are obtained when the test object belongs to an impostor. Thus, the distance in this case is a similarity measure, while the probability is a dissimilarity measure and has to be transformed into a similarity one. To do this, the combined match scores $s_i(x_T)$ can be defined as:

$$s_i(x_T) = \begin{cases} -p(x_T|w_i) & \text{if the measure is a probability} \\ d(x_T|w_i) & \text{otherwise} \end{cases} \quad (6)$$

In contrast, when the target class is the genuine user class $w_G$ (see Figs. 3(b) and 4(b)), the probability becomes a similarity measure, while the distance turns out to be a dissimilarity measure. The combined match scores $s_G(x_T)$ are as follows:

$$s_G(x_T) = \begin{cases} p(x_T|w_G) & \text{if the measure is a probability} \\ -d(x_T|w_G) & \text{otherwise} \end{cases} \quad (7)$$

The former case was considered using $\nu$-SVM [126-18], and was observed to be comparable to the two-class SVM, surpassing other conventional combination rules, including the sum of scores in the experiments, carried out on the NIST BSSR1 and MCYT databases.

B. User-specific Score Fusion

In user-specific score fusion, different descriptions around the target class are determined for each user enrolled in the system using exclusively their corresponding match score vectors. Similarly to user-independent score fusion, the target class can be either the genuine user ($w_{ji}$) or impostor ($w_{ij}$).

Subsequently, the combined match score, related to the user $j$ is defined as:

$$s_{j}(x_T) = \begin{cases} -p(x_T|w_{ji}) & \text{if the measure is a probability} \\ d(x_T|w_{ji}) & \text{otherwise} \end{cases} \quad (8)$$

if the target class is the impostor, and:

$$s_{jG}(x_T) = \begin{cases} p(x_T|w_{jG}) & \text{if the measure is a probability} \\ -d(x_T|w_{jG}) & \text{otherwise} \end{cases} \quad (9)$$

if the target class is the genuine user. The latter case is impractical to achieve due to the limited availability of genuine match scores per user. GMM requires a large number of target samples to converge to the true density, while SVDD suffers from the lower bound on the number of support vectors required for its description. In [13], the authors demonstrated that GMM and SVDD do not work at all with a sample size of ten. Due to this reason, we only consider the user-specific score fusion in (8), where the descriptions are learned using the impostor class samples.

As the impostor is the target class, user-specific fusion can be generally considered as a better alternative, relative to the user-independent counterpart for the following reasons:
User-specific fusion is faster in testing time, since fewer samples are used to construct the classifier. This is particularly true in the case of k-NN, where, during testing, distances of the test object from all training samples have to be calculated and sorted. A similar observation can also be made when SVDD is used as a classifier, where the reduction in the number of training samples leads to a smaller number of support vectors, and associated computational savings.

One-class classifiers fail to achieve high verification performance when the impostor class is composed of various sub-clusters. A straightforward explanation for this is that such methods train classifiers using the matching score patterns from different users, while the literature suggests that users of a biometric system may have differing degrees of accuracy within the system [19]. Particularly, sheepish users can be easily recognized, matching poorly against others, and well against themselves, while other users (i.e., lambish and wolfish users) are particularly successful at impersonation, receiving high matching scores for all verifications even when matching against others. Enrollment of such users amplifies the within-class sub-concepts problem in the impostor class distribution. User-specific fusion may partially alleviate this problem by defining a different reference model for each user, thus forming a more reliable and compact scatter.

V. CONCLUSIONS

Biometric authentication is the process of verifying a human identity using behavioural and physiological characteristics. Multimodal biometric systems can improve verification performance by combining evidence from multiple biometric traits. A common practice in multimodal biometric is to view fusion at match score level as a two-class classification problem. However, conventional two-class classifiers are inadequate, when applied to problems characterized by class imbalance. Class imbalance occurs when the samples of the impostor class vastly outnumber those of the genuine user class. In this work, the paradigm of one-class classification methods was exploited in imbalanced biometric data sets. One-class classifiers were employed to learn the user-specific and user-independent descriptions around either the impostor or genuine user classes. The user-specific approach consistently demonstrates a better authentication performance with respect to the user-independent counterparts. It is also able to overcome or partially overcome the problem of within-class sub-concepts, which arises when the target class is scattered into several small regions due to the existence of the user-variations.

REFERENCES