Support Kernel Classification: A New Kernel-Based Approach

Ouiem Bchir[1], Mohamed M. Ben Ismail[2]*, Sara Algarni[3]
College of Computer and Information Sciences, King Saud University, Riyadh, KSA

***Abstract*—In this paper, we introduce a new classification approach that learns class dependent Gaussian kernels and the belongingness likelihood of the data points with respect to each class. The proposed Support Kernel Classification (SKC) is designed to characterize and discriminate between the data instances from the different classes. It relies on the maximization of the intra-class distances and the minimization of the intra-class distances to learn the optimal Gaussian parameters. In fact, a novel objective function is proposed to model each class using one Gaussian function. The experiments conducted using synthetic datasets demonstrated the effectiveness of the proposed algorithm. Moreover, the results obtained using real datasets proved that the proposed classifier outperforms the relevant state of the art approaches.**

Keywords*—Supervised learning; classification; kernel based learning

I. Introduction

Classification finds applications in many real-world problems related to different fields. Such applications include, for example, analyzing customer data in the areas of commerce [1-2], detecting fraud to benefit industry and government [3], improving the learning process in education [4], predicting the climate for crop production [5], and assisting doctors in detecting anomalies in the healthcare [6]. In order to solve these classifications problems, many approaches have been reported in the literature [7]. However, most approaches assume that the different categories can be separated using linear boundaries, and thus, they are effective when the data has a simple geometric characteristic with well-separated categories.

Kernel-based approaches [8] have been proposed as an alternative solution. They map the data into a new feature space in such a way that categorizing classes with complex boundaries can be reduced to a simple categorization problem in the new feature space. Nevertheless, the choice of an optimal kernel that allows separating linearly the different categories of the data is a challenging problem [9]. The most common used kernel is the Gaussian kernel due to its statistical and geometrical proprieties [10]. However, it is sensitive to the choice of the Gaussian parameters. An exhaustive search of these parameters requires training the classifier many times to consider different possible values. In addition to the problems related to the selection of the set of possible values, and to the time complexity, the exhaustive search may also lead to an over-fitting problem. In fact, the Gaussian parameters are selected based on the value of a criterion function that is computed on the training data [11-13]. Moreover, a global parameter over the entire data may be inappropriate when the different categories have large characteristics variations.

In this paper, we propose a novel classification algorithm named the Support Kernel Classifier (SKC). It categorizes the data by learning a Gaussian kernel for each category. SKC is designed to learn the Gaussian parameters from the intrinsic geometric characteristics of the data. More specifically, the kernel parameters are learned by minimizing the intra-class distances and maximizing the inter-class distances simultaneously. Moreover, the proposed classifier learns the probability of each data point to belong to each class. In fact, it does not use crisp assignment where an instance belongs or not to a class, but rather learns its likelihood to belong to it. This is intended to better describe the data. Besides, it allows avoiding the over-fitting problem.

The rest of the report is as follows: In Section II, we present the related works. Section III describes the proposed approach. The experimental results and analysis are outlined in Section IV. Finally, we conclude the report and highlight the future works in Section V.

II. Related Works

In this review, we focus on the main classification approaches based on the Gaussian kernel. More specifically, we focus on the approaches that learn the Gaussian parameters such as the support vector machine [15], the Gaussian mixture [16], and the radial basis function neural network [17].

A. Parameters Selection for the Support Vector Machine

The typical *SVM* [15] algorithm was extended by mapping the input data vectors into high-dimensional feature space [18]. This mapping can be obtained by using a kernel function $K(x_i, x_j)$ [19]. Thus, the SVM discriminant function becomes:

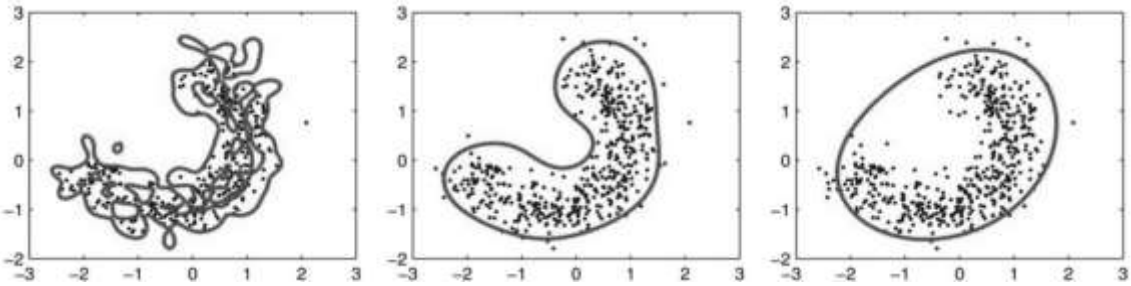
$$g(x) = \sum_{i \in SV} \alpha_i K(x_i, x_j) + b \quad (1)$$

Although any kernel function can be used, the Gaussian kernel is widely used. It is defined as:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2)$$

The choice of the Gaussian parameter, σ, affects the *SVM* performance. As shown in Fig. 1, when σ is too small, the discriminant function surrounds each data point, which may lead to an over-fitting problem. Yet, if σ is too large, the discriminant function surrounds all points, which yields mapping all points into a single one [20].

*Corresponding Author



(a) (b) (c)

Fig. 1. Hyperplane Affected by the Value of σ: (a) σ=0.1, (b) σ=2.6 and (c) σ=10 [20].

Since the value of the Gaussian parameter has a large impact on the *SVM* classification results, several attempts have been proposed in the literature to determine this parameter.

The authors in [21] proposed a Gaussian parameters selection approach for a the outlier detection problem. They used the dual function as a criterion to find the optimal kernel parameters. In particular, the introduced the criterion [21]:

$$\max \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \quad (3)$$

Similarly, the authors in [20] used a dual function as an extension of the criterion in (3). They proposed the following objective function:

$$\max \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - \max \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \quad (4)$$

Inspired by the "Fisher linear discrimination" (*FLD*) [11], the authors in [22] proposed to find the Gaussian parameters using an objective function that minimizes the intra- class distances and maximizes the inter-class distances. The proposed criterion was defined as:

$$f(\sigma) = \frac{s_1^2 + s_2^2}{\|m_1 - m_2\|^2} \quad (5)$$

Where and m_1 and m_2 are the mean of the two classes while s_1^2 and s_2^2 are the coincidences of two classes defined as:

$$s_k^2 = N_k - \frac{1}{N_k} \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} K(x_{ki}, x_{kj}) \quad k = \{1,2\} \quad (6)$$

The authors in [23] proposed an approach to learn the Gaussian parameters based on maximizing the "kernel target alignment" (*KTA*) objective function [12]. *KTA* maximizes the intra-class similarities and minimizes the intra-class similarities using given the following expression:

$$A(K, y) = \frac{\langle K, y\dot{y} \rangle_F}{\sqrt{\langle K, K \rangle_F \langle y\dot{y}, y\dot{y} \rangle_F}} \quad (7)$$

where $K: x^2 \longrightarrow [-1, +1]$ and $y \in \{-1, +1\}^m$. Note that $\langle K, y\dot{y} \rangle_F$ is the difference between the intra-class similarities and the intra-class similarities as defined below:

$$\langle K, y\dot{y} \rangle_F = \sum_{y_i = y_j} k(x_i, x_j) - \sum_{y_i \neq y_j} k(x_i, x_j) \quad (8)$$

where $k(x_i, x_j)$ is the Gaussian kernel function. The optimal σ is obtained by maximizing the following objective function:

$$\sigma_{opt} = argmax \left(\sum_{y_i = y_j} exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2} \right) - \sum_{y_i \neq y_j} exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2} \right) \right) \quad (9)$$

The optimization problem formulated in (9) is solved by computing the partial derivative and setting it to zero. Thus,

$$\frac{\partial \langle K, y\dot{y} \rangle_F}{\partial \sigma} = \frac{1}{\sigma^3} \left(\sum_{y_i = y_j} \|x_i - x_j\|^2 exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2} \right) - \sum_{y_i \neq y_j} \|x_i - x_j\|^2 exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2} \right) \right) \quad (10)$$

For the SVM-based approaches, various criterion have been proposed to select the optimal Gaussian parameters [20][21][22]. These approaches are similar to the exhaustive search. Yet, the approach in [31] learns the Gaussian parameter by maximizing the intra-class similarities and minimizing the intra-class similarities. However, this approach is suitable for the two-class problems only.

B. Parameters Selection for the Gaussian Mixture Models

Typical Gaussian Mixture Model (*GMM*) classifier [16] relies on Bayesian framework, Gaussian probabilistic modelling and the expectation maximization (*EM*) algorithm [13]. In particular, it assumes that the data can be modelled as a mixture of a finite number of Gaussian functions. GMMs compute the probability density functions (*PDF*), $P(X \mid C_i)$ [24], for the data instance *X* given the class C_i. Then, they classify the test instances using the Bayes' rule [25] using theses PDfs as:

$$P(C_i|X) = P(X|C_i) \cdot \frac{P(C_i)}{P(X)} \quad (11)$$

where $P(C_i)$ is the class i prior probability and $P(X)$ serves as a normalization term. Note that the GMM assumes that the probability density functions, $P(X \mid C_i)$, are a weighted sum of multiple Gaussians as:

$$P(X|C_i) = \sum_{k=1}^{NG} w_k G_k \quad (12)$$

In (12), NG is the number of Gaussians and w_k is the weight associated with the k^{th} Gaussian G_k constrained to:

$$\sum_{k=1}^{Nc} w_k = 1 \quad (13)$$

The k^{th} Gaussian G_k is formulated as:

$$G_k = \frac{1}{(2\pi)^{n/2}|S_k|^{1/2}} \cdot e^{[-1/2(X-M_k)^T S_k^{-1}(X-M_k)]} \quad (14)$$

where M_k and S_k are the mean and the covariance, respectively. The *GMMs* can be defined through three parameters. Namely, the set of mean, $\{M_k\}$, the set of covariances, $\{S_k\}$, and the weights, $\{w_k\}$ represent the model parameters. The *EM* [13] is the iterative optimization approach typically used to estimate these parameters.

In order to estimate the three parameters M_k, S_k, and w_k, the Maximum Likelihood Estimation (*MLE*) algorithm [26] can be used.

Let $\{x_i\}_{i\in\{1,2,\dots,N\}}$ be a set of data points and let $P_c(x_i|\Theta_c)$ be the conditional probability of x_i belonging to cluster *c* defined by $\Theta_c = \{M_c, S_c\}$ where M_c is the centroid of the cluster and S_c its covariance matrix. One should note that the set of mean $\{M_k\}$ is initialized using the *K*-means clustering algorithm [14].

GMM [24] defines the total probability distribution of x_i as:

$$P_{total}(x_i) = \sum_{c=1}^{C} A_c P_c(x_i|\Theta_c) \quad (15)$$

where *C* is the number of clusters and A_c which represents the ratio of the number of data instances in the cluster *c* is computed as:

$$A_c = \frac{N_c}{N} \quad (16)$$

with N_c is the number of instances assigned to the cluster *c*. *GMM* [27] optimizes the log-likelihood of the total probability distribution of x_i below

$$G = \sum_{i=1}^{N} \log P_{total}(x_i) \quad (17)$$

The probability that the instance x_i belongs to cluster *c*, is defined as:

$$w_{ic} = \frac{A_c P_c(x_i|\Theta_c)}{P_{total}(x_i)} \quad (18)$$

In fact, w_{ic} is considered as the membership of the data instance x_i to the cluster *c* such that:

$$\sum_{c=1}^{C} w_{ic} = 1 \quad (19)$$

The number of data points assigned to cluster *c* can be expressed as

$$N_c = \sum_{i=1}^{N} w_{ic} \quad (20)$$

The covariance matrix, S_c, is then defined as

$$S_c = \left(\frac{1}{N_c}\right) \sum_{i=1}^{N} w_{ic} (x_i - M_c)(x_i - M_c)' \quad (21)$$

with

$$M_c = \left(\frac{1}{N_c}\right) \sum_{i=1}^{N} w_{ic} x_i \quad (22)$$

Similarly to the *MLE* [35] approach, the maximum a posterior (*MAP*) approach [28] computes the *GMM* parameters M_k, S_k, and w_k. However, it estimates $\Theta_c = \{M_c, S_c\}$ by maximizing the posterior probability function not the likelihood function. More specifically, *MAP* finds Θ_c that maximizes:

$$L_{MAP} = \prod_{i=0}^{N_{train}} P(\theta_c|x_i) \quad (23)$$

MAP assumes that θ_c is a random variable with a distribution. In fact, it relies on the equation:

$$P(\theta_c, x_i) = P(\theta_c|x_i)P(\theta_c) \quad (24)$$

Note $P_c(\boldsymbol{x}_i|\Theta_c)$ represents the conditional probability that an instance $\boldsymbol{x}_i$ belongs to a cluster c defined by $\Theta_c = \{M_c, S_c\}$. Where M_c is the centroid of the cluster and S_c is its covariance matrix. For each mixture i in the prior model, *MAP* [28] defines the posterior probability $P_c(\Theta_c|x_i)$ as:

$$P(\theta_c|x_i) = \frac{P(x_i|\theta_c)P(\theta_c)}{P(x_i)} \quad (25)$$

The optimal θ_c is a random variable defined by:

$$\theta_c = \arg\max_{\theta_c} \prod_{x_i} P(\theta_c|x_i) \quad (26)$$

Since $P(x_i)$ is not dependent of θ_c, one can write

$$\theta_c = \arg\max_{\theta_c} \prod_{x_i} P(x_i|\theta_c)P(\theta_c) \quad (27)$$

This yields

$$\theta_c = \arg\max_{\theta_c} \sum_{x_i} \log P(x_i|\theta_c) + \log(P(\theta_c)) \quad (28)$$

If X follows a normal distribution $N(\mu, \sigma)$, where μ is random and σ^2 is fixed. Then,

$$P(x_i|\theta_c)P(\theta_c) = \left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}\right] \times \left[\frac{1}{\sigma_0\sqrt{2\pi}} \exp\left\{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right\}\right] \quad (29)$$

This gives

$$f(x|\mu)\pi(\mu) = \frac{1}{2\pi\sigma\sigma_0} \exp\left\{-\frac{1}{2\sigma^2\sigma_0^2}[\sigma_0^2(x_i-\mu)^2 + \sigma^2(\mu-\mu_0)^2]\right\} \quad (30)$$

As it can be seen, $P(x|\theta_c)P(\theta_c)$ is proportional to

$$\exp\left\{-\frac{1}{2\sigma^{*2}}(\mu-\mu^*)^2\right\} \quad (31)$$

where

$$\mu^* = \frac{\sigma_0^2}{\sigma^2+\sigma_0^2}x + \frac{\sigma_0^2}{\sigma^2+\sigma_0^2}\mu_0 \quad (32)$$

and

$$\sigma^{*2} = \frac{\sigma^2\sigma_0^2}{\sigma^2+\sigma_0^2} = \left(\frac{1}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1} \quad (33)$$

Consequently, it follows a normal distribution with μ^* and σ^{*2} as parameters.

One can claim that *MLE* [26] and *MAP* [38] are efficient approaches that provide interpretable results. However, *MLE* [36] based solutions are prone to over-fitting [33]. On the other hand, *MAP* [28] addresses the over-fitting problem through the assumption that the parameters of the Gaussian distribution that fits the data are known.

C. Parameters Selection for Radial basis Function Network (RBFN)

The Radial Basis Function Network (*RBFN*) is a particular neural network where the Gaussian distribution is used as activation functions [29]. Besides, the network output is a combination of Gaussian functions of the inputs:

$$O_d(x) = w_0 + \sum_{i=0}^{m} w_i * G(x, \mu, \sigma) \quad (34)$$

where $O_d(x)$ is the output corresponding to the input x, w_i are the weights, and $G(x, \mu, \sigma)$ is the Gaussian function characterized by the parameters μ and σ. Figure 2 displays the architecture of a *RBFN*.

Training the *RBFN* involves learning the optimal weights $w_0, w_1, \dots, w_n$. These weights are learned using gradient descent. Therefore, the iterative learning process requires deriving the training error, which is defined as:

$$E = \frac{1}{2}\sum_d (t_d - O_d)^2 \quad (35)$$

where t_d is the target label and O_d is the output label.

The authors in [31] used the Gradient descent to learn iteratively the Gaussian parameters. Let V = [μ σ w] be the vector including the mean μ, the standard deviations σ, and the set of weighs w respectively. The update equation of V is defined as follows:

$$V^{new} = V^{old} + \alpha \frac{\partial \varepsilon}{\partial V} \quad (36)$$

where α is the learning rate.

In [32], the researchers proposed the learning of the Gaussian parameters based on the intra-class and inter-class structures in the training data. Specifically, the mean C^k of each class k is computed. Then, the distance d_k is defined as the distance between the mean C^k and furthest sample P belonging to the class k based on the distance.

$$d_k = \|P^k - C^k\| \quad (37)$$

The second step consists in computing the distance between each mean and the closest mean to it.

$$d_c(k) = \text{argmin}\big(d_c(k,j)\big)\ j = 1, \dots, s, j \neq k \quad (38)$$

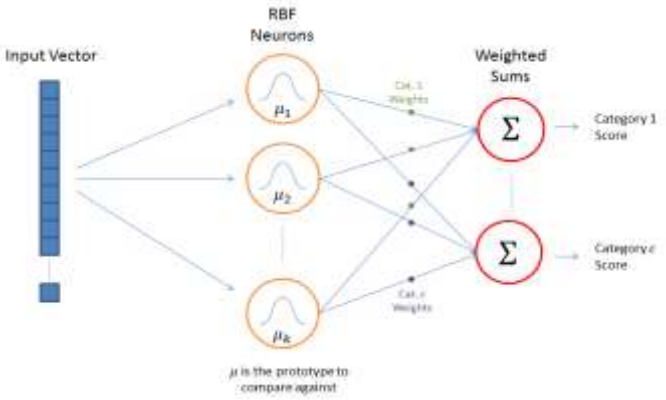


Fig. 2. Radial basis Function Network Architecture [30].

Given a confidence parameter β, the width of class k, σ_w^k is:

$$\sigma_w^k = \frac{d_k}{\sqrt{|\ln \beta|}} \quad (39)$$

The overlap between class k and class I (σ_B^k) is:

$$\sigma_B^k = \eta \times d_{min}(k, l) \quad (40)$$

where η is a factor that controls the overlap between the classes. The Gaussian parameter, σ, with respect to class k, is defined in [32] as the largest value between σ_w^k and σ_B^k:

$$\sigma^k = max(\sigma_B^k\ ,\ \sigma_w^k) \quad (41)$$

As the choice of η is not straightforward, the authors in [32] suggested the following approximation:

$$\eta \approx \frac{\sum_{i=1}^{c} \frac{d_k}{\sqrt{|\ln \beta|}}}{\sum_{k=1}^{c} d_{min}(k,l)} \quad (42)$$

The value of the Gaussian parameter, σ, is then updated using gradient descent by deriving the training error defined in (35).

These approaches may be prone to local minima. The approach in [20] tries to avoid the problem by suggesting a way to initialize the parameter based on the intra- and inter-class similarities. However, the suggested approach requires the estimation of other parameters.

III. The Proposed Support Kernel Classification

Kernel classification approaches are intended to categorize the data by mapping it into a new feature space. This mapping reduces the complex classification task to a simpler problem in the new feature space. The Gaussian kernel function is commonly used due to its analytical characteristics. However, the performance of the Gaussian kernel based classifiers depends on the setting of the Gaussian parameters. In this work, we propose a new kernel-based classification approach where each class is modeled using a Gaussian function. The optimal Gaussian parameters are learned by optimizing a novel objective functions.

Let a Gaussian function be defined as:

$$G_{ijk} = \exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (43)$$

where σ_i is the scaling parameter, and e_{jk}^2 represents the distance between the data points x_j and x_k. In this work, we use the squared Euclidian distance defined as:

$$e_{jk}^2 = |x_j - x_k|^2 = (x_j - x_k)(x_j - x_k)^T \quad (44)$$

The optimal set of Gaussian parameters $\{\sigma_i\}$ is obtained by minimizing the intra-class distances and maximizing the inter-class distances. More specifically, the proposed approach formulates and minimizes the intra-class distances as follows:

$$J_i^{intra} = \sum_{j=1}^{Ntrain} \sum_{k=1}^{Ntrain} p_{ij}^m p_{ik}^m \left(1 - \exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right)\right) + K\sigma_i^2 \quad (45)$$

Similarly, it maximizes the inter-class distances below:

$$J_i^{inter} = \sum_{j=1}^{Ntrain}\sum_{k=1}^{Ntrain}\left(p_{ij}^m(1-p_{ik}^m)+p_{ik}^m(1-p_{ij}^m)\right)\left(1-\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right)\right)+K\sigma_i^2 \quad (46)$$

In (45) and (46), N_{train} represents the number of observations in the training set $\{x_j\}_{j=1,..,N\ train}$, $m \in]1,\infty)$ is a constant that determines the degree of overlapping between classes, $K\sigma_i^2$ represents a regularization term, and p_{ij} expresses the likelihood that the observation x_j belongs to the class i. Note that p_{ij} satisfies:

$$0 \le p_{ij} \le \text{ and } \sum_{i=1}^{C} p_{ij} = 1 \text{ for } j \in \{1,\dots,N_{train}\} \quad (47)$$

where C is the number of classes. Notice that the regularization term is integrated in (45) in order to avoid the trivial solution of large scaling parameter, σ_i, which would map all data instances into one single point. On the other hand, the regularization term in (47) is intended to avoid the trivial solution of a scaling parameter equal to zero. Besides, K allows ensures the tradeoff between the minimization of J^{intra} and the maximization of J^{inter}. Note that K is also learned through the optimization of J^{intra} and J^{inter}.

The distance, r_{jk}, between the data points x_j and x_k, in the new feature space is defined as:

$$r_{jk} = 1-\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (48)$$

The Gaussian parameters $\{\sigma_i\}$ are defined with respect to each class in order to better handle the distribution and the geometric characteristics of each class. The proposed approach learns the scaling parameter σ_i for each class and the likelihood for each observation x_i to belong to class i using the given the training set. Moreover, the objective functions J_i^{intra} and J_i^{inter} are based on the relational distances between pairs of data instances rather than the distance between the data instances and the classes. This relaxes the assumption that each class fits a spherical shape [34]. In fact, the objective functions J_i^{intra} and J_i^{inter} do not use the class means/centroids. In the proposed approach, the mean is used only in the testing phase. It is computed after learning $\{\sigma_i, p_{ij}\}$ of each class i.

A. Optimization with Respect to p_{ij}

In order to optimize J_i^{intra} and J_i^{inter} with respect to p_{ij}, we use the relational dual described in [35]. It defines the relation between the relational distance $\{r_{jk}\}$ and the distance between point x_j and class i, d_{ij}, using the probabilities $\{p_{ij}\}$ as follows:

$$d_{ij} = \sum_{j=1}^{Ntrain} e_{jk}\frac{p_{ij}^m}{\sum_{v=1}^{Ntrain} p_{iv}^m} - \frac{1}{2}\sum_{j=1}^{Ntrain}\sum_{q=1}^{Ntrain}\frac{p_{ij}^m e_{jq}\, p_{iq}^m}{(\sum_{v=1}^{Ntrain} p_{iv}^m)^2} \quad (49)$$

Using the relational dual, we rewrite (45) and (46) and obtain the following set of equations system:

$$\begin{cases} J_i^{intra} = \sum_{j=1}^{N_{train}} p_{ij}^m d_{ij} + K\,\sigma_i^2 \\ J_i^{inter} = \sum_{j=1}^{N_{train}} (1-p_{ij}^m) d_{ij} + K\,\sigma_i^2 \end{cases} \quad (50)$$

In order to optimize J_i^{intra} and J_i^{inter} with respect to p_{ij}, subject to the constraint in (47), we use the language multipliers to obtain:

$$\begin{cases} J^{intra} = \sum_{j=1}^{N_{train}} p_{ij}^m d_{ij} + K\,\sigma_i^2 - \lambda_i\left(\sum_{i=1}^{C} p_{ij} - 1\right) \\ J^{inter} = \sum_{j=1}^{N_{train}} \left(1-p_{ij}\right)^m d_{ij} + K\,\sigma_i^2 - \lambda_i\left(\sum_{i=1}^{C} p_{ij} - 1\right) \end{cases} \quad (51)$$

where d_{ij} is as defined in (49) and λ_i is the Lagrange multiplier variable. Setting the derivatives with respect to p_{ij} of the set of equations in (51) to zero yields:

$$m p_{ij}^{m-1} d_{ij} - \lambda_i = 0. \quad (52)$$

Thus,

$$p_{ij} = \left(\frac{\lambda_i}{m d_{ij}}\right)^{\frac{1}{m-1}} \quad (53)$$

and,

$$\left(\frac{\lambda_i}{m}\right)^{\frac{1}{m-1}} \sum_{i=1}^{C}\left(\frac{1}{d_{ij}}\right)^{\frac{1}{m-1}} = 1 \quad (54)$$

This results in

$$\lambda_i^{\frac{1}{m-1}} = \frac{m^{\frac{1}{m-1}}}{\sum_{i=1}^{C}\left(\frac{1}{d_{ij}}\right)^{\frac{1}{m-1}}} \quad (55)$$

and,

$$p_{ij} = \frac{\left(\frac{1}{d_{ij}}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{C}\left(\frac{1}{d_{kj}}\right)^{\frac{1}{m-1}}} \quad (56)$$

B. Optimization with Respect to σi

In order to optimize J^{intra} and J^{inter} with respect to σ_i, we derive J_i^{intra} and J_i^{inter} with respect to σ_i^2 and set the derivatives to zero. First, we set the derivative of J_i^{intra} to zero and to obtain:

$$\frac{\partial J_i^{intra}}{\partial \sigma_i^2} = -\sum_{j=1}^{Ntrain}\sum_{k=1}^{Ntrain} p_{ij}^m p_{ik}^m \frac{e_{ij}^2}{\sigma_i^4}\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) + K = 0 \quad (57)$$

which yields:

$$K = \sum_{j=1}^{N_{train}}\sum_{k=1}^{Ntrain} p_{ij}^m p_{ik}^m \frac{e_{jk}^2}{\sigma_i^4}\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (58)$$

Substituting (58) in (46) gives

$$J_i^{inter} = \sum_{j=1}^{N_{train}}\sum_{k=1}^{Ntrain}\left(p_{ij}^m(1-p_{ik}^m)+p_{ik}^m(1-p_{ij}^m)\right)\left(1-\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right)\right) + \frac{1}{\sigma_i^2}\sum_{j=1}^{N_{train}}\sum_{k=1}^{Ntrain} p_{ij}^m p_{ik}^m e_{jk}^2 \exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (59)$$

The derivative of (59) with respect to σ_i^2 can be written as:

$$\frac{\partial J_i^{inter}}{\partial \sigma_i^2} =$$

$$-\frac{1}{\sigma_i^4}\sum_{j=1}^{N_{train}}\left(p_{ij}^m(1-p_{ik}^m)+p_{ik}^m(1-p_{ij}^m)\right)e_{jk}^2\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right)$$

$$-\frac{1}{\sigma_i^4}\sum_{j=1}^{N_{train}}\sum_{k=1}^{Ntrain}p_{ij}^m p_{ik}^m e_{jk}^2\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right)$$

$$+\frac{1}{\sigma_i^6}\sum_{j=1}^{N_{train}}\sum_{k=1}^{Ntrain}p_{ij}^m p_{ik}^m\left(e_{jk}^2\right)^2\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (60)$$

Which results in:

$$\sigma_i^2=\frac{T_1^i}{T_2^i} \quad (61)$$

where

$$T_1^i=\sum_{j=1}^{Ntarin}\sum_{k=1}^{Ntrain}p_{ij}^m p_{ik}^m\left(e_{jk}^2\right)^2\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (62)$$

and

$$T_2^i=\sum_{j=1}^{Ntrain}\sum_{k=1}^{Ntrain}\left(p_{ij}^m+p_{ik}^m-p_{ij}^m p_{ik}^m\right)e_{jk}^2\exp\left(-\frac{e_{jk}^2}{\sigma_i^2}\right) \quad (63)$$

Based on an iterative optimization approach and the assumption that σ_i and p_{ij} do not change significantly from one iteration to another, we σ_i, and p_{ij} can be updated alternatively using (61), and (56), respectively. Once σ_i and $\{p_{ij}\}$ are optimized with respect to each class i, we define the prototype for each category. Then, it can be used during the testing phase in order to predict the class value for any unlabeled data point. Along with the standard deviation σ_i, we propose to use the mean μ_i of each class i as its prototype. Specifically, we define it as:

$$\mu_i=\frac{\sum_{j=1}^{Ntrain}p_{ij}^m d_{ij}}{\sum_{j=1}^{Ntrain}p_{ij}^m} \quad (64)$$

where d_{ij} is as defined in (49). The proposed training algorithm is depicted below:

Algorithm 1: SKC training phase

Input: *training set* $\{x_j\}_{j=1,..,N_{train}}$
Output: $\{\mu_i,\sigma_i\}_{i=1,...,C}$
1- *Initialize the probability according to the class labels such that* $p_{ij}=1$ *if* x_j *belongs to class* i*, and 0 otherwise.*
2- *Initialize* $\sigma_i^{(0)}$ *to 1.*
3- *Set* ε *to* 10^{-5}.
Repeat
1- *Compute* $d_{ij}^{(t)}$ *using (49)*
2- *Compute* $\sigma_i^{2(t)}$ *using (51)*
3- *Compute* $p_{ij}^{(t)}$ *using (56)*
Until $\|\sigma_i^t-\sigma_i^{t-1}\|<\varepsilon\ \&\ t\le 100$
Compute μ_i *using (64).*

The set of Gaussian parameters $\{\mu_i,\sigma_i\}_{i=1,...,C}$ defines the model with respect to each class i.

Using the learned models from the training set, we classify the unlabeled data point x_j using

$$class(x_j)=arg\max_i(q_{ij}) \quad (65)$$

where

$$q_{ij}=1-\exp\left(\frac{\|x_j-\mu_i\|^2}{\sigma_i^2}\right) \quad (66)$$

The latter equation (66) represents the distance between the test data point x_j and the center μ_i as defined in (48). Note that the parameters $\{\mu_i,\sigma_i\}$ were learned during the training phase. The proposed *SKC* testing algorithm is detailed below:

Algorithm 2: SKC testing phase

Input: $\{\mu_i,\sigma_i\}_{i=1,...,C}$, *an unknown observation* x_j
Output: $class\ value\ of\ (x_j)$
1- *Compute* $\{q_{ij}\}$ *using (66)*
2- *Predict the class value of* (x_j) *using (65)*

IV. Experiments

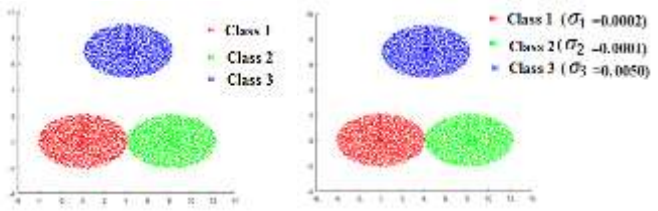
In order to assess the performance of the proposed approach, we conducted several experiments using both synthetic and real datasets. The synthetic datasets are 2-D datasets generated to represent different geometric characteristics. They were used to illustrate visually how the proposed classifier categorizes them. Moreover, they were intended to analyze and interpret the learned Gaussian parameters. Besides, the proposed approach was evaluated using real benchmark datasets. Specifically, 10 data sets from the UCI repository [46] were used to analyze the performance of the proposed approach. Namely, these datasets are: Handwritten Digits [36], Mammographic Mass [37], E.coli [38], and Haberman's Survival [39], Frogs MFCCs [40], Blood Transfusion Service Center [41], HCC Survival [42], Adolescent Autistic Spectrum Disorder Screening [43], Libras Movement [44], and Seeds of wheat [45] data sets. Table I summarizes the considered real datasets.

A. Experiments using Synthetic Datasets

In order to show that *SKC* succeeds to learn the optimal Gaussian parameters for each class, and simultaneously classifies accurately the data instances. Therefore, we set the fuzzifier *m* to *2*, and the maximum number of iterations to *100*. Then, we run *SKC* on the synthetic 2-D datasets in Fig. 3. As it can be seen, the datasets include 3 classes with the same intrinsic characteristics. However, class 1 and class 2 exhibit low inter-class distances, while they show large inter-class distances with class 3. *SKC* classifies correctly data set 1 as shown in Fig. 3(b). It learns two similar Gaussian parameters for class 1 and class 2 (σ_1 = 0.0002 and σ_2 = 0.0001), and a larger Gaussian parameter (σ_3 = 0.0050) for class 3. Indeed, σ_1 is not too large so that points from class 2 get assigned to class 1. Similarly, σ_2 is not too large so that points from class 1 are not labeled as class 2. On the other hand, σ_3 is relatively larger because its intra-class distances are larger.

TABLE I. CHARACTERISTICS OF THE 10 REAL DATASETS

Name	N° of elements	Feature size	N° of categories	Categories
Handwritten Digits [36]	7494	17	10	Digit numbers from 0 to 9
Mammographic Mass [37]	961	6	2	Benign malignant
E.coli [38]	336	8	8	Protein localization sites
Haberman's Survival [39]	306	4	2	5 years < 5 years
Frogs MFCCs [40]	7195	22	4	Bufonidae Dendrobatida Hylidae Leptodactylida
Blood Transfusion Service Center [41]	748	5	2	Donating Not donating
HCC Survival [42]	165	50	2	Survives Dies
Adolescent Autistic Spectrum Disorder Screening [43]	104	64	2	ASD Not ASD
Libras Movement [44]	135	91	15	Hand movement types
Seeds of wheat [45]	210	8	3	Kama, Rosa Canadian



(a) (b)

Fig. 3. Classifying Synthetic Dataset using SKC. (a) The Synthetic Dataset, (b) The Classification Result Obtained using SKC. The Learned Gaussian Parameters are: $\boldsymbol{\sigma_1}$ = 0.0002, $\boldsymbol{\sigma_2}$ = 0.0001 and $\boldsymbol{\sigma_3}$ = 0.005.

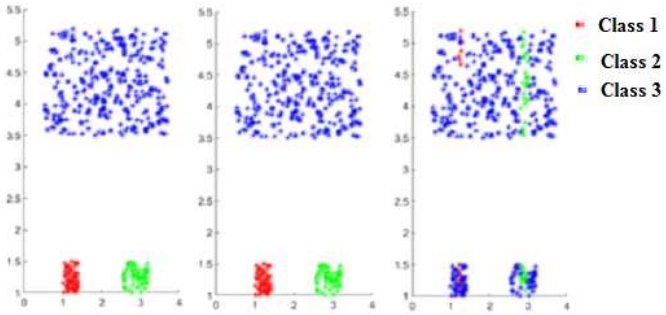
The Gaussian Mixture Models (*GMM*) based classification [16] is the most similar approach to SKC because it learns a Gaussian mixture for each class. Therefore, we compare the classification results obtained using *SKC* and to those achieved using *GMM* on various different datasets.

As shown in Fig. 4(a), the dataset includes has three classes where class 1 and 2 have similar size and density while class 3 which is larger and less dense. *SKC* succeeds to learn the optimal Gaussian parameters for each class (σ_1 = 3.35 10^{-04}, σ_2 = 8.07 10^{-04}, and σ_3 = 9.31 10^{-01}), and classifies accurately dataset 2 as shown in Figure 4-(b). In fact, the classification problem gets easier if the classes have similar volume and density. Such performance is attained through the Gaussian parameters learned by *SKC* where a larger σ_3 allows shrinking class 3 so it is less sparse and has a comparable volume to class 1 and class 2. Since class 1 and class 2 have comparable intra/inter class distances, similar Gaussian parameters are learned by *SKC*. On the other hand, s reported in Fig. 4(c), *GMM* is not able to classify accurately dataset 2.

Another synthetic dataset is shown in Fig. 5(a). As one can notice, despite the good classification results obtained using *SKC*, some border points are misclassified as reported in Fig. 5(b). On the other hand, as shown in Fig. 5(c), *GMM* yields poor classification results because it learns larger a Gaussian parameter for class 1 compared to class 2 which results in similar density and volume for both classes.

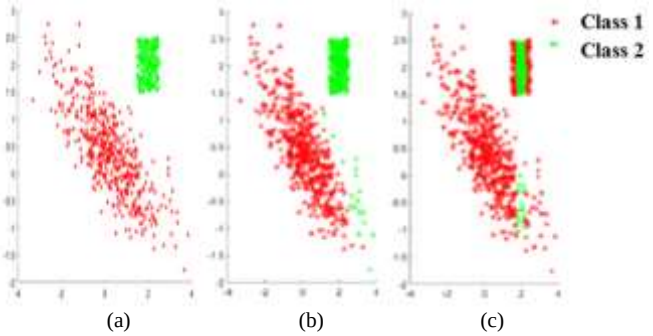
Fig. 6 reports the classification results obtained by *SKC* and *GMM* using a different synthetic dataset. Although the inter-class distances are too small for the border points, *SKC* classifies correctly this dataset as displayed in Fig. 6(b). In fact, *SKC* learns double the value of σ_2 for class 1 to shrink it more than class 2. This yields a considerable separation between both classes. On the other hand, as reported in Figure 6-(b), *GMM* misclassifies the border points which degrades the overall classification performance.

Similarly, for the synthetic dataset in Fig. 7, both classifiers misclassify some border points from class 2 which the inter-class distances with class 1 is lower than the intra-class distance. In particular, *SKC* learns similar Gaussian parameters for class 1 and class 2, cannot discriminate accurately between the border points. Whereas, some of the points from class 2 that are misclassified by *GMM* have relatively large inter-class distance with class 1.



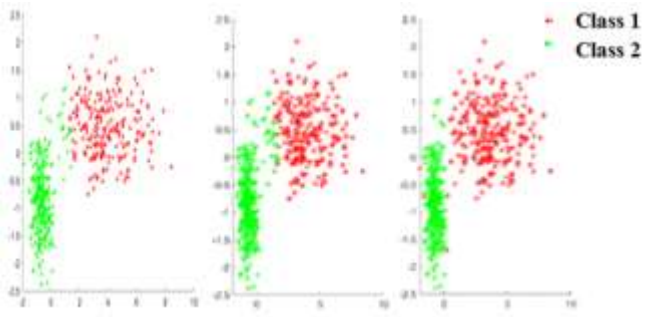
(a) (b) (c)

Fig. 4. Classification Results Obtained using SKC and GMM. (a) The Synthetic Dataset, (b) SKC Classification Results with $\boldsymbol{\sigma_1}$ = 3.35 10-04, $\boldsymbol{\sigma_2}$ = 8.07 10-04, and $\boldsymbol{\sigma_3}$ = 9.31 10-01, and (c) GMM Classification Results.



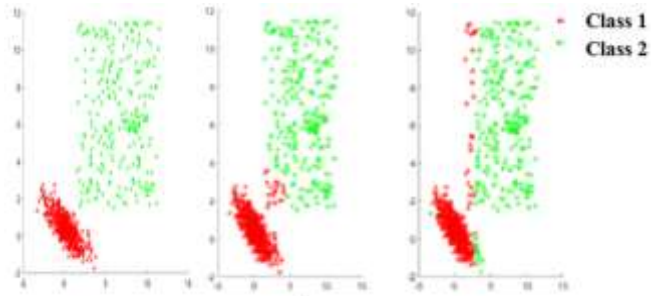
(a) (b) (c)

Fig. 5. Classification Results Obtained using SKC and GMM.(a) The Synthetic Dataset, (b) SKC Classification Results with σ_1 = 7.62 10-02, and σ_2 = 8.82 10-04, and (c) GMM Classification Results.



(a) (b) (c)

Fig. 6. Classification Results Obtained using SKC and GMM. (a) The Synthetic Dataset, (b) SKC Classification Results with σ_1 = 5.11 10-03, and σ_2 = 2.29 10-03, and (c) GMM Classification Results.



(a) (b) (c)

Fig. 7. Classification Results Obtained using SKC and GMM. (a) The Synthetic Dataset, (b) SKC Classification Results with σ_1 = 2.45 10-05, and σ_2 = 3.88 10-05, and (c) GMM Classification Results.

Based on the comparison of the classification results obtained by *SKC* and *GMM* using the different synthetic datasets, one can claim that the learning of the optimal Gaussian parameters using the intra-class and the inter-class characteristics of each dataset, makes SKC outperform *GMM*. Besides, in case of large volume variations for the different classes, *GMM* misclassifies a considerable proportion of the dataset. In fact, *GMM* does not include the inter-class distance in the learning process of the Gaussian parameters of the large classes. This yields the misclassification of some points from the other near classes. Moreover, *GMM* fails to classify the border points when the two classes are too close. This can be attributed to the fact that it learns the Gaussian parameters based on the intra-class distances only.

B. Experiments using Real Dataset

In this section, we report the classification results of the benchmark datasets from the UCI repository [46]. The classification task was conducted using the proposed *SKC*, the Gaussian Mixture Model classifier (GMM) [16], the *K*-nearest Neighbour classifier (*KNN*) [47], the kernel Support Vector machine (SVM) with Gaussian kernel [18], and the Naïve Bayes approach [19]. Note that for *KNN* we set three different values of *K* (*K=1*, *K=3*, and *K=5*). For the Gaussian kernel *SVM* we varied the Gaussian parameter by setting 6 different values (10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, and 1). For SKC, we set the fuzzifier *m* to *2*, and the maximum number of iterations to *100*.

We adopted a 10-folds cross validation approach, along with the accuracy, the sensitivity, and the specificity as performance measures to report the classification performance. Thus, Tables II and III show the average scores over the 10 training iterations. Moreover, a t-test was conducted to evaluate the statistical significance of the obtained results. In Tables II and III, the best results are shown in red. On the other hand, the green color represents the results that are not significantly different according to the t-test. As it can be seen, *SKC* overtakes all classifiers on Handwritten Digits [36] and Mammographic Mass [37] datasets. Moreover, it yields the best performances on the remaining 7 data sets.

Similarly, *SKC* outperforms *KNN* [47] on Handwritten Digits [46] and Mammographic Mass [47] data sets. However, it yields the same performance attainment on the other datasets. Even though *KNN* does not use the Gaussian kernel, it uses the local characteristics of the data by labelling the unknown instances based on their neighbouring points in the training set. Moreover, it requires a prior setting of the number of neighbours (*K*).

TABLE II. PERFORMANCES MEASURES OBTAINED USING THE DIFFERENT CLASSIFIERS ON HANDWRITTEN DIGITS, MAMMOGRAPHIC MASS, E.COLI, AND HABERMAN'S SURVIVAL, AND FROGS MFCCS DATASETS

		Accuracy	Sensitivity	Specificity	t-test
Handwritten Digits	***SKC***	0.84	0.91	0.99	
	GMM	0.47	0.45	0.99	1
	KNN	0.78	0.78	1	1
	SVM	0.16	0.17	0.89	1
	NB	0.83	0.83	0.98	1
Mammographic Mass	***SKC***	0.78	0.79	0.85	
	GMM	0.76	0.73	0.8	1
	KNN	0.77	0.77	0.77	1
	SVM	0.53	0.56	0.48	1
	NB	0.76	0.71	0.73	1
E.coli	***SKC***	0.82	0.94	0.98	
	GMM	0.55	0.63	0.94	0
	KNN	0.72	0.83	0.95	1
	SVM	0.7	0.89	0.95	1
	NB	0.75	0.87	0.95	1
Haberman's Survival	***SKC***	0.71	0.77	0.77	
	GMM	0.49	0.38	0.54	1
	KNN	0.61	0.65	0.51	1
	SVM	0.62	0.75	0.26	1
	NB	0.62	0.69	0.42	0
Frogs MFCCs	***SKC***	0.76	0.96	1	
	GMM	0.4	0.48	1	1
	KNN	0.69	0.77	1	0
	SVM	0.23	0.88	1	1
	NB	0.72	0.8	1	1

TABLE III. PERFORMANCES MEASURES OBTAINED USING THE DIFFERENT CLASSIFIERS ONBLOOD TRANSFUSION SERVICE CENTER, HCC SURVIVAL, ADOLESCENT AUTISTIC SPECTRUM DISORDER SCREENING, LIBRAS MOVEMENT, AND SEEDS OF WHEAT DATASETS

		Accuracy	Sensitivity	Specificity	t-test
Blood Transfusion Service Center	***SKC***	0.68	0.75	0.72	
	GMM	0.46	0.38	0.71	1
	KNN	0.53	0.51	0.6	0
	SVM	0.64	0.74	0.34	1
	NB	0.66	0.73	0.43	0
HCC Survival	***SKC***	0.58	0.48	0.65	
	GMM	0.49	0.6	0.43	1
	KNN	0.52	0.53	0.51	0
	SVM	0.58	0.23	0.55	1
	NB	0.55	0.44	0.62	0
Adolescent Autistic Spectrum Disorder Screening	***SKC***	0.99	0.98	1	
	GMM	0.81	0.67	0.9	0
	KNN	0.85	0.74	0.92	0
	SVM	0.64	0.53	0.7	1
	NB	0.9	0.8	0.96	1
Libras Movement	***SKC***	0.72	0.78	1	
	GMM	0.41	0.39	0.98	1
	KNN	0.62	0.64	0.99	0
	SVM	0.18	0.43	0.96	0
	NB	0.67	0.67	0.99	0
Seeds of wheat	***SKC***	0.9	0.86	0.93	
	GMM	0.89	0.85	0.91	0
	KNN	0.9	0.84	0.92	0
	SVM	0.9	0.89	0.99	0
	NB	0.89	0.83	0.93	0

In addition SKC yields the same results as SVM [18] on HCC survival [42] and Seeds of wheat [45], while it beats SVM [18] on the remaining datasets. In fact, although Kernel *SVM* [18] relies on the inter-class distances through the learning the optimal hyperplanes that guarantee the best inter-class margin, it uses one global sigma for all the data in the original features space. In other words, it assumes that all classes follow the same distributions.

Also, one can see that SKC attains similar results as NB [19] on Haberman's survival [39], Blood Transfusion Service Center [41], HCC survival [42], Libras movement and Seeds of wheat [45]. On the other hand, it outperforms NB [19] on the remaining 5 datasets. This results can be attributed to the fact that NB [19] learns a sigma for each feature with respect to each class. Therefore, if the features are not independent it fails to discover the correct structure of the data. Moreover, NB doesn't take into consideration the inter-class distances.

SKC yields similar performance to GMM [16] on E.coli [38], Adolescent Autistic Spectrum Disorder Screening [43], and Seeds of wheat [45] data sets. It overtakes GMM [16] on the 7 other data sets. Even though, GMM learns a Gaussian parameter with respect to each feature and the corresponding covariance matrix, it does not take into consideration the inter-class dissimilarities**.** Therefore, the Gaussians parameters are learned based on the intra–class distances only.

V. CONCLUSIONS

Despite the researchers' efforts to address the supervised learning challenges, most of the classification algorithms exhibit some limitations. The classification task is even more acute when the data classes show different distribution characteristics. Kernel-based classifiers were introduced to overcome this problem through the mapping of the data into a new feature space using a specific kernel function. This mapping is intended to obtain better separation between the data classes and simplify the classification task. Even though the Gaussian function proved to yield reasonable classification accuracy, its performance depends on the choice of its parameters' values. Moreover, if the data include highly variant classes in terms of size, density, and shape, the data mapping into a new feature space using one global Gaussian is not effective. Typically, the tuning of the Gaussian parameters is done though some search strategy that is intended to optimize a predefined criterion function. In this paper, we proposed a new classification algorithm that learns a Gaussian function for each data class. The proposed Support Kernel Classification (SKC) is designed to characterize and separate the data instances from the different classes. It relies on the maximization of the intra-class distances and the minimization of the intra-class distances to learn the optimal Gaussian parameters. In fact, a novel objective function is optimized to model each class using one Gaussian function. The experiments conducted using synthetic datasets demonstrated the effectiveness of the proposed algorithm. Moreover, the results obtained using real datasets proved that the proposed classifier outperforms the relevant state of the art approaches.

ACKNOWLEDGMENT

This work was supported by the Research Center of the College of Computer and Information Sciences at King Saud University, Riyadh, Saudi Arabia. The authors are grateful for this support.

REFERENCES

[1] G. P. Zhang, "Neural networks for classification: a survey," IEEE Trans. Syst. Man Cybern. Part C (Applications Rev., vol. 30, no. 4, pp. 451–462, 2000.

[2] D. L. García, À. Nebot, and A. Vellido, "Intelligent data analysis approaches to churn as a business problem: a survey," Knowl. Inf. Syst., vol. 51, no. 3, pp. 719–774, 2017.

[3] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," Journal of Network and Computer Applications, vol. 68. pp. 90–113, 2016.

[4] C. Angeli, et al., "Data mining in educational technology classroom research: Can it make a contribution?," Comput. Educ., vol. 113, pp. 226–242, 2017.

[5] N. Gandhi and L. J. Armstrong, "A review of the application of data mining techniques for decision making in agriculture," in Int. Conf. on Contemporary Computing and Informatics, 2016, pp. 1–6.

[6] W. Sun, at al., "A survey of data mining technology on electronic medical records," in e-Health Networking, Applications and Services (Healthcom), IEEE 19th International Conference on, 2017, pp. 1–6.

[7] J. Han and M. Kamber, Data Mining: Concepts and Techniques, vol. 12. 2011.

[8] D. Tien Bui, T. A. Tuan, H. Klempe, B. Pradhan, and I. Revhaug, "Spatial prediction models for shallow landslide hazards: a comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree," Landslides, vol. 13, no. 2, pp. 361–378, 2016.

[9] M. E. Abbasnejad, D. Ramachandram, and R. Mandava, "A survey of the state of the art in learning the kernels," Knowl. Inf. Syst., vol. 31, no. 2, pp. 193–221, 2012.

[10] M. Tian and W. Wang, "An efficient Gaussian kernel optimization based on centered kernel polarization criterion," Inf. Sci. (Ny)., vol. 322, pp. 133–149, 2015.

[11] K. Fukunaga, Introduction to Statistical Pattern Recognition, vol. 22. 1990.

[12] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," Adv. Neural Inf. Process. Syst. 14, pp. 367--373, 2002.

[13] A. P. Dempster, et al., "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. R. Stat. Soc. Ser. B, vol. 39, no. 1, pp. 1–38, 1977.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Proc. fifth Berkeley Symp. …, vol. 233, no. 233, pp. 281–297, 1967.

[15] C. Cortes and V. Vapnik, "Support-Vector Networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[16] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, vol. 7. 1973.

[17] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," 1988.

[18] K. R. Müller, et al., "An introduction to kernel-based learning algorithms," IEEE Transactions on Neural Networks, vol. 12, no. 2. pp. 181–201, 2001.

[19] M. A. Aizerman, E. A. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," Autom. Remote Control, vol. 25, pp. 821–837, 1964.

[20] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of gaussian kernel for one-class SVM," IEEE Trans. Cybern., vol. 45, no. 5, pp. 927–939, 2015.

[21] D. Kakde, et al., "Peak criterion for choosing Gaussian kernel bandwidth in Support Vector Data Description," in IEEE Int Conf on Prognostics and Health Management, ICPHM, 2017, pp. 32–39.

[22] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the Gaussian kernel for classification and regression," Neurocomputing, vol. 55, no. 3–4, pp. 643–663, 2003.

[23] S. Zhong, D. Chen, Q. Xu, and T. Chen, "Optimizing the Gaussian kernel function with the formulated kernel target alignment criterion for two-class pattern classification," Pattern Recognit., vol. 46, no. 7, pp. 2045–2054, 2013.

[24] G. McLachlan and D. Peel, "Mixtures of factor analyzers," Finite Mix. Model., pp. 238–256, 2000.

[25] H. Jeffreys, "Scientific Inference," Library (Lond)., vol. 2, 1931.

[26] J. Pfanzagl and H. R., Parametric Statistical Theory. Walter de Gruyter, 1994.

[27] A. Ortiz-Rosario, H. Adeli, and J. A. Buford, "MUSIC-Expected maximization gaussian mixture methodology for clustering and detection of task-related neuronal firing rates," Behav. Brain Res., vol. 317, pp. 226–236, 2017.

[28] C. P. Robert, The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation, vol. 91, no. 433. 2007.

[29] D. Broomhead, D. S. and Lowe, "Multivariable Functional Interpolation and Adaptive Networks," Complex Syst., vol. 2, pp. 321–355, 1988.

[30] C. McCormick, "Radial Basis Function Network (RBFN) Tutorial," 2013. [Online]. Available: http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/. [Accessed: 17-Apr-2018].

[31] L. W. Kang, et al, "A new neural network model for the state-of-charge estimation in the battery degradation process," Appl. Energy, vol. 121, pp. 20–27, 2014.

[32] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," IEEE Trans. Neural Networks, vol. 13, no. 3, pp. 697–710, 2002.

[33] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for machine learning.," Gaussian Processes for Machine Learning. 2006.

[34] O. Bchir, et al., "Fuzzy clustering with learnable cluster-dependent kernels," Pattern Anal. Appl., 2016.

[35] R. J. Hathaway, et al., "Relational duals of the c-means clustering algorithms," Pattern Recognit., 1989.

[36] F. Alimoglu, "Combining multiple classifiers for pen-based handwritten digit recognition," Institute of Graduate Studies in Science and Engineering, Bogazici University, 1996.

[37] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process," Med. Phys., 2007.

[38] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins.," Int. Conf. Intell. Syst. Mol. Biol., 1996.

[39] D. DeCoste, "Anytime Query-Tuned Kernel Machines via Cholesky Factorization," in SDM, 2003.

[40] J. G. Colonna, M. Cristo, M. Salvatierra, and E. F. Nakamura, "An incremental technique for real-time bioacoustic signal segmentation," Expert Syst. Appl., 2008.

[41] I. C. Yeh, K. J. Yang, and T. M. Ting, "Knowledge discovery on RFM model using Bernoulli sequence," Expert Syst. Appl., 2009.

[42] M. S. Santos, P. H. Abreu, P. J. García-Laencina, A. Simão, and A. Carvalho, "A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients," J. Biomed. Inform., 2015.

[43] F. Thabtah and Fadi, "Autism Spectrum Disorder screening: Machine learning adaptation and DSM-5 fulfillment," in Proceedings of the 1st International Conference on Medical and Health Informatics 2017 - ICMHI '17, 2017.

[44] D. B. Dias, R. C. B. Madeo, T. Rocha, H. H. Bíscaro, and S. M. Peres, "Hand movement recognition for Brazilian Sign Language: A study Using distance-based neural networks," in Proceedings of the International Joint Conference on Neural Networks, 2009.

[45] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Zak, "Complete gradient clustering algorithm for features analysis of X-ray images," Adv. Intell. Soft Comput., 2010.

[46] D. Dheeru and E. Karra Taniskidou, "{UCI} Machine Learning Repository." 2017.

[47] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," IEEE Trans. Syst. Man. Cybern., no. 4, pp. 580–585, 1985.