MATHEMATICAL MODELING AND COMPUTING, Vol. 8, No. 4, pp. 747–760 (2021)

Density based fuzzy support vector machine: application to diabetes dataset

El Ouissari A., El Moutaouakil K.

Engineering Science Laboratory (LSI), Faculty Polydisciplinary of Taza, USMBA, Morocco

(Received 23 May 2021; Accepted 7 June 2021)

In this work, we propose a deep prediction diabetes system based on a new version of the support vector machine optimization model. First, we determine three types of patients (noisy, cord, and interior) basing on specific parameters. Second, we equilibrate the clinical data sets by suppressing noisy and cord patients. Third, we determine the support vectors by solving an optimization program with a reasonable size. Our system is performed on the well-known diabetes dataset PIMA. The experimental results show that the proposed method improves the prediction accuracy and the proposed system significantly outperforms all other versions of SVM as well as literature methods of classification.

Keywords: *support vector machine, machine learning, optimization, classification, diabetes.*

2010 MSC: 68T01, 68T99, 62H30 **DOI:** 10.23939/mmc2021.04.747

1. Introduction

Diabetes is a major global health problem. According to the report of WHO (World Health Organization), diabetes is among the most prevalent diseases among the elderly in the country, and also each year, an estimated 1.6 million people die from the effects of diabetes [1]. According to the International Diabetes Federation report, 451 million individuals worldwide have diabetes in 2017, and this number is expected to increase to 693 million people in the next 26 years [1]. Diabetic is a major chronic disease that occurs when the pancreas does not produce sufficient insulin or when the body does not properly use the insulin it produced. There are three types of DM, the pathogenesis of type 1 diabetes mellitus (T1DM), the pathogeneses of type 2 diabetes mellitus (T2DM) [2] and Gestational Diabetes Mellitus (GDM). It is very important the diagnosis to prevent the diabetic of type 2 and treat it in time. Many works have used machine learning (ML) methods to predict diabetes [3]. One of the most efficient methods for prediction is Support Vector Machine SVM [4–7], it is a machine learning method and one of the most efficient methods for solving classification problems. SVM has attracted the attention of researchers because of its good performance in practical applications and solid theoretical foundations [5–7]. SVM has been developed and different versions have emerged and among these versions we have the following [8–23]: **C-SVM**: If the data is linearly non-separable, Vapnikh and Cortes [24] introduce the notion of a soft margin. We introduce N additional variables ξ_i called slack variables, where $\xi_i \geqslant 0$. Then, we solve the following problem:

$$\begin{cases} \min \dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{N}\xi_i, \\ \text{subject to } y_i(x_i.w + b) \geqslant 1 - \xi_i, \\ \xi_i \geqslant 0. \end{cases} \tag{1}$$

Define the Lagrangian of the soft margin SVMs:

This work was supported by MENFPESRS and the DDA of Morocco (Nos. Alkhawarizmi/2020/23).

$$L(w,\alpha_i,\xi_i)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i\left[y_i(x_i.w+b)-1+\xi_i\right]-\sum_{i=1}^{N}\beta_i\xi_i. \tag{2}$$

By performing the transformation into a dual problem, we obtain the following:

$$\begin{cases}\max\limits_{\alpha}\sum\limits_{i=1}^{N}\alpha_i-\dfrac{1}{2}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i x_j,\\ \text{subject to }\sum\limits_{i=1}^{N}\alpha_i y_i=0,\\ 0\leqslant\alpha_i\leqslant C.\end{cases} \tag{3}$$

G-SVM: We also have problems of classification of m points in the real space of n dimension $\mathbb{R}^n$, representing by the matrix A of dimension $m\times n$, knowing for each point A_i is belonging to the class $+1$ or -1 specified by a given diagonal matrix D of dimension $m\times m$. For this type of problem we have the linear problem of support vector machine is given by:

$$\begin{cases}\min\|w\|+\nu e'y,\\ \text{subject to } D(A.w-e\gamma)+y\geqslant e,\\ y\geqslant 0,\end{cases} \tag{4}$$

where $\nu>0$.

For the non-linear problem of the support vector machine we can do the transformation $w=A'Du$ then the problem (6) must be in the following form:

$$\begin{cases}\min\|A'Du\|+\nu e'y,\\ \text{subject to } D(AA'Du-e\gamma)+y\geqslant e,\\ y\geqslant 0.\end{cases} \tag{5}$$

We replace AA' by a nonlinear kernel function $K(A,A')$ and ADu by u with u a convex function of $\mathbb{R}^m$. Then we get the next generalized support vector machines (G-SVM) [9]:

$$\begin{cases}\min\|A'Du\|+\nu e'y,\\ \text{subject to } D(k(A',A)Du-e\gamma)+y\geqslant e,\\ y\geqslant 0.\end{cases} \tag{6}$$

F-SVM [10]: In SVM we find that all the training data belong to one of the classes, that is to say, each entry point belongs to this class or the other class, but in real problems it does not work good because we find a difference between the training points and that there are more important points than the other, and also misclassed training points which makes the classification task difficult and full of errors. To remedy this problem Chun–Fu Lin proposed the notion fuzzy SVM, this technique is based on each input data is added a value s_i called fuzzy membership such that $0\leqslant s_i\leqslant 1$. This fuzzy membership s_i can be considered as the position of the corresponding training point x_i toward a class.

ν-SVM [11]: One of the difficulties we face in solving a non-separable problem is finding or predicting the C value, since the parameter C takes a large value from 0 to infinity, it is very difficult to determine, Bernhard Scholkopf proposes an extension of C-svm called ν-SVM.

The idea is replaced C by another more intuitive parameter, this parameter is confined between 0 and 1. This parameter roughly represents the fraction of the expected support vectors, so for any

given $\nu \in [0,1]$, we will know a priori how the class will be formed. One of the excellent property of ν-SVM is that ν control the number of training errors and consequently the support vectors.

LS-SVM [12]: Least Squares Support Vector Machine Classifiers is a version of SVM which proposes by Suykens [12] and it is close to the classical SVM method, but instead of solving the quadratic problem one seeks to solve a system of linear equations.

OC-SVM [13]: One class Support Vector Machine is a version of SVM [13, 20]. This version is concerned with only one class, i.e. the target class, and any data that does not belong to this class is considered extreme. In one-class SVM the convex optimization problem is solved using the quadratic programming method.

T-SVM [14]: Another version of SVM which solves the problem when we have input data are corrupted by noise, this version called total support vector machine (TSVM), which is motivated by the method of total least squares regression.

W-SVM [15]: The idea of the weighted support vector machine (W-SVM) version is as follows: at each entry point a different weight is added in terms of their importance in the class such that each different point has a different contribution to the learning of the decision area.

Gr-SVM: Granular Support Vector Machines (G-SVM) is a version that systematically and formally combines the theory of statistical learning with the theory of granular computing. For more detail and advantage of G-SVM see [17].

S-SVM [18]: another reformulation of SVM called smooth support vector machine(S-SVM). In the problems of unconstrained optimization of SVM [18], the objective function is not twice differentiable, and to remedy this problem we use smoothing techniques to fall in a new version vector support machine (S-SVM).

P-SVM [21]: support vector machine classifiers (P-SVM) is classified the new input data according to the proximity to one of the two parallel hyperplane that are pushed as far apart as possible. And to obtain a linear or non-linear P-SVM classifier we solve a system of linear equations.

GEP-SVM [22]: Multisurface proximal support vector machine classification via generalized eigenvalues (GEP-SVM) is an extension of P-SVM but instead of solving a single linear equation system, we formulate two eigenvalue problems to generate two non-parallel planes; with a necessary condition is that the input data of each class is as close as possible to their own class and as far as possible from the other class.

T-SVM [23]: Twin support vector machines (T-SVM) is very similar to GEP-SVM version; but, they are based on an absolutely different formulation. The T-svm also find to obtain two non-parallel planes; with a necessary condition is that the input data of each class is as close as possible to their own class and as far as possible from the other class.

To overcome these drawbacks, we propose a new SVM version called Density Based Support Vector Machine (DBF-SVM). This version is based on two parameters to know: the radius of the neighborhood of the data and the number of points contained in this neighborhood. Basing on these parameters, we determine three types of samples: noisy, cord and interior point. To ensure more consistency of DBF-SVM, we select a random subset of a noise data, randomly chosen, are injected in the learning set and during the resolution of the model (RD) we reject them taking according to their weights. As we show, theoretically, that the cord points data cannot be support vectors, they are destitute form the learning set. By doing this, we don't loss generality because we show, mathematically, that a cord sample keeps its nature even if we introduce any kind of kernel function.

This paper is organized as follows: the second section presents the fuzzy support vector machine, the third section presents our proposed approach called density based Support Vector Machine (DBF-SVM), where we explain the mathematical theories of the approach. The fourth section discusses the experiment results obtained with our DBF-SVM approach on two types of data and its comparison with other known classifiers. Finally, we conclude the paper by section 5.

2. Fuzzy Support Vector Machine Theory

Vapnikh and Cortes introduced the notion of a soft margin to overcome the problem of the sutured constraints, this version is called C-SVM [24]. In this sense, they used N additional slack variables $\xi_i \geqslant 0$ to each constraint $y_i(x_i.w + b) \geqslant 1$. The sum of the slack variables is penalized and integrated in the objective function:

$$\begin{cases} \text{Min}\dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{N}\xi_i, \\ \quad \text{Subject to:} \\ \qquad y_i(\phi(x_i).w + b) \geqslant 1 - \xi_i, \\ \qquad \xi_i \geqslant 0, \quad \forall i = 1, \ldots, N. \end{cases}$$

Where ϕ is the transformation function extracted from the kernel function K. By performing the transformation into a dual problem, basing on the slack variables $\alpha_1, \ldots, \alpha_N$, we obtain the following optimization problem:

$$\begin{cases} \text{Max}\sum\limits_{i=1}^{N}\alpha_i - \dfrac{1}{2}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j), \\ \quad \text{Subject to:} \\ \qquad \sum\limits_{i=1}^{N}\alpha_i y_i = 0, \\ \qquad 0 \leqslant \alpha_i \leqslant C, \quad \forall i = 1, \ldots, N. \end{cases}$$

In SVM we find that all the training data belongs to one of the classes, that is to say, each entry point belongs to this class or to the other class, but in real problems it doesn't work well because we find a difference between the training data, for example we find that some training data is more important than others, and also misclassified training data, it makes the task of classification difficult and full of errors. To remedy this problem Chun–Fu Lin [10] proposed the fuzzy SVM concept, this technique is based on the fact that in each input data we add a value m_i called fuzzy membership such that $0 < m_i < 1$. This fuzzy membership m_i can be considered as the position of the corresponding training point of a class in the classification problem, and based on the importance of the data in their class.

Several works have been done to develop this extension for example A New Fuzzy Support Vector Machine to Evaluate Credit Risk by Wang [25] which is based on each entry point, the latter has added two membership values, also the works [26–28] are developments of the fuzzy extension SVM.

We reformulate the problem of classification and take the data set as follows:

$$(x_1, y_1, m_1),\ (x_2, y_2, m_2),\ \ldots, (x_N, y_N, m_N),$$

such as $x_i \in \mathbb{R}^n$.

We add to each entry point a fuzzy membership for this reason all the input data contributes to the learning of the decision terminal, so the reformulation of the SVM is like this:

$$\begin{cases} \text{Min}\dfrac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{N} m_i\xi_i, \\ \quad \text{Subject to:} \\ \qquad y_i(\phi(x_i).w + b) \geqslant 1 - \xi_i, \\ \qquad \xi_i \geqslant 0, \forall i = 1, \ldots, N. \end{cases}$$

To solve this problem, we use the Lagrange multiplier method:

$$L(w, \rho, \xi, \alpha, \beta, b, \delta) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} m_i\xi_i - \sum_{i=1}^{N}\alpha_i\big(y_i((w.x_i) + b) - 1 + \xi_i\big) - \sum_{i=1}^{N}\beta_i m_i\xi_i.$$

Finally, we have the dual problem:

$$\begin{cases} \text{Max} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \\ \quad \text{Subject to:} \\ \quad\quad \sum_{i=1}^{N} \alpha_i y_i = 0, \\ \quad\quad 0 \leqslant \alpha_i \leqslant m_i C, \forall i = 1, \ldots, N. \end{cases}$$

To determine fuzzy membership, we look for the main properties of the data set, and we relate those properties to the fuzzy membership, for example Chun [10] assumes that time is the main property of the dataset, the latter gives the value m_i as a function of time and expresses the value of m_i as follows:

$$m_i = f(t_i)$$

with $0 < \sigma \leqslant m_i \leqslant 1$ and $t_1 \leqslant t_2 \leqslant \ldots \leqslant t_N$.

Chun [10] proposes to take that $s_1 = f(t_1) = \sigma$ and $s_N = f(t_N) = 1$ and also proposes two types of fuzzy value,the first one is a linear function type value given by:

$$m_i = f(t_i) = at_i + b,$$

and we use board conditions to determine a and b, one finds:

$$m_i = f(t_i) = \frac{1-\sigma}{t_N - t_1} t_i + \frac{t_N \sigma - t_1}{t_N - t_1},$$

and the second is a quadratic function type value given by:

$$m_i = f(t_i) = a(t_i - b)^2 + c,$$

and we use board conditions to determine a, b and c:

$$m_i = f(t_i) = (1-\sigma) \left(\frac{t_i - t_1}{t_N - t_1} \right)^2 + \sigma.$$

3. Literature review

To predict diabetes by using the Indian Pima Diabetes Dataset (PIDD), many works have used machine learning (ML) methods. A number of closely related works are discussed in this section [29–32].

In [29] the authors used different methods of machine learning and neural network on the PIMA dataset to predict diabetics. They used seven machine learning methods which are (LR, KNN, SVM, NB, DT, RF and AB), and noticed that SVM and LR are giving the best results for the prediction of diabetics. They also built a neural network with different hidden layers (1, 2 and 3 hidden layers) and with different epochs (200, 400 and 800 epochs), and found that the neural network with two hidden layers gave a better result than the others.

Another study which used the logistic regression method for the classification of superseded learning to predict the risk of type 2 diabetes in individuals is that of Tigga et. al. In [30]. the idea of this study is to improve the prediction of the Logistics Regression method for predicting the risk of diabetes.

Shuja and all [31] they used a model of classification to predict diabetes based on two phases: the first phase is preprocessing the data using Smote method to equilibrium the data unbalanced, and the second phase is to feeding the most efficacy techniques of data mining of classification (Bagging, SVM

(Support Vector Machine), MLP (Multi-Layer Perceptron), Simple Logistic and Decision Tree) with the preprocessed data, to obtain an efficient model for predicting type 2 diabetes mellitus.

Delshi and all in [32] they proposed an approach that was used to diagnose Diabetes mellitus (DM). This method benefits from the Farthest First (FF) clustering algorithm and the Sequential Minimal Optimization (SMO) classifier algorithm. They used Farthset First (FF) to group the data into number of clusters and Support Vector Machine (SVM) to classify the output to diabetic and non-diabetic patients.

4. Density Based Fuzzy Support Vector Machine (DBF-SVM)

Let BD be a set of N samples $x_1, \ldots, x_N$ labeled, respectively, by $y_1, \ldots, y_N$, distributed via k class $C_1, \ldots, C_k$. In this section, we set a non-negative real r and a positive integer number mp, which denotes min-points, that allows us to determine three types of data: noise, border and interior. In this sense, we show that kernel function doesn't change the interior points nature. In addition, we proof that this kind of points can not be selected as a support vector. Finally, we show, theoretically, the influence of the parameter mp on the samples nature.

Definition 1. *Let $S \subseteq \mathbb{R}^n$. A point $a \in \mathbb{R}^n$ is said to be an Interior Point of S if there exists an $r > 0$ such that $B(a, r) \subseteq S$. The set of all interior points of S is denoted by* $\text{int}(S)$ *or* S°.

Definition 2. *For a given dataset BD, a non-negative real r and an integer mp, there exist three kind of samples.*

1. *A sample x is called C_i-Noise Point (NP_i) if $|C_i \cap B(x, r)| < mp$.*
2. *A sample x is called C_i-Cord Point (CP_i) if $|C_i \cap B(x, r)| \geqslant mp$ and $x \in \overbrace{\text{envol}(C_i)}^{o}$*
3. *A sample x is called C_i-Border Point (BP_i) if $|C_i \cap B(x, r)| < mp$ and there exists a C_i-cord point y such as $x \in B(y, r)$.*

Let K be a kernel function that permits to pass from $\mathbb{R}^n$ to $\mathbb{R}^N$ via a transformation denoted ϕ ($n < N$).

Lemma 1. *If a is a C_i-Cord point for a given ε and minpoints (mp), then $\phi(a)$ is also a C_i-Cord point with appropriate ε' and the same minpoints (mp).*

Proof. As a is a C_i-Cord point, there exist $x_1, \ldots, x_p$ such as $\forall l \in \{1, \ldots, p\}$, $\|x_l - a\| \leqslant \varepsilon$ and $p \geqslant mp$.

Let by ϕ the function defined by the equation $k(x, y) = \phi(x)\phi(y)$.

We have

$$\phi(x) - \phi(a) = \sum_{i=1}^{n} (x_i - a_i)\frac{\partial \phi}{\partial x_i}(a) + \frac{1}{2}\sum_{i,j=1}^{n} (x_i - a_i)(x_j - a_j)\frac{\partial^2 \phi}{\partial x_i \partial x_j}(a) + \|x - a\|^2 \varepsilon(x - a).$$

Thus

$$\|\phi(x) - \phi(a)\| \leqslant \sum_{i=1}^{n} |x_i - a_i| \left\| \frac{\partial \phi}{\partial x_i}(a) \right\| + \frac{1}{2}\sum_{i,j=1}^{n} |x_i - a_i||x_j - a_j| \left\| \frac{\partial^2 \phi}{\partial x_i \partial x_j}(a) \right\| + \|x - a\|^2 \|\varepsilon(x - a)\|,$$

where $\lim\limits_{x \to a} \varepsilon(x - a) = 0$.

Then

$$\|\phi(x) - \phi(a)\| \leqslant nK_1\|x - a\|_\infty + \frac{1}{2}nK_2\|x - a\|_\infty^2 + M^2 K_3 \|x - a\|_\infty^2,$$

where K_1, K_2, K_3 and M are defined by:

$$K_1 = \text{Max}\left\{ \left\| \frac{\partial \phi}{\partial x_i}(z) \right\|,\ i = 1, \ldots, n \text{ and } j = 1, \ldots, n \text{ and } z \in BD \right\},$$

$$K_2 = \text{Max}\left\{\left\|\frac{\partial^2\phi}{\partial x_i \partial x_j}(z)\right\|, i = 1,\ldots,n \text{ and } j = 1,\ldots,n \text{ and } z \in BD\right\},$$
$$K_3 = \text{Sup}\big\{\|\varepsilon(x-a)\|, z \in B, \text{ where } B \text{ is the smallest bole containing } BD\big\},$$
$$M'\|x-y\|_\infty \leqslant \|x-y\| \leqslant M\|x-y\|_\infty, \quad \forall x, y \in \mathbb{R}^n.$$

If we set

$$K = nK_1 + \left(\frac{1}{2}nK_2 + M^2K_3\right)\text{Max}_{x,y\in BD}\|x-y\|_\infty.$$

Then

$$\|\phi(x) - \phi(a)\| \leqslant K'\|x-a\|,$$

where $K' = \frac{K}{M'}$.

Finally

$$\forall l \in \{1,\ldots,p\},\ \|\phi(x_l) - \phi(a)\| \leqslant \varepsilon \quad \text{and} \quad p \geqslant \text{MinPoints}.$$

■

Theorem 1. *A cord point can never be selected to be a support vector in the sense of SVM.*

Proof. As it is always possible to find an integer N such as DB is linearly separable, we demonstrate the demanded result for the linear case. Suppose that SVM selects a cord point cp (said it is $a+sample$) as support vector. Since the data is linearly separable, the separation zone S determined by SVM must not contain any sample. Let Δ^+ and Δ^- be the two lines delimiting S.

As cp is $a + cord$ point, thus $cp \in \text{int}(\text{envol}(C^+ \cap B(cp, r)))$.

Then there exist x^+ and y^+ from $C^+ \cap B(cp, r)$ which are not from the same side of Δ^+ said x^+.

Thus $x^+ \in S$ absurd.

As consequence a cord point can never be selected to be a support vector in the sense of svm. ■

Proposition 6. For ε fixed, cordPoints(minPoints) is decreasing function for the inclusion operator.

Proof. Let ε be a nonnegative real number. For $\forall mps1 \in \mathbb{N}$, $\forall mps2 \in \mathbb{N}$ such as $mps1 \leqslant mps2$, we have

$$\text{cordPoints}(mps2) \subseteq \text{cordPoints}(mps1).$$

In fact, let $x \in \text{cordPoints}(mps2)$.

Then $|C_i \cap B(x,r)| \geqslant mps2$ and $x \in \overbrace{\text{envol}(C_i)}^{o}$.

As $mps1 \leqslant mps2$, we have $|C_i \cap B(x,r)| \geqslant mps1$.

Thus x fulfilled, both, the condition 1) and 2) for $mps1$. Finely $x \in \text{cordPoints}(mps1)$. ■

Recall the last dual that consists finding multipliers $\alpha_1, \ldots, \alpha_n$ such that:

$$\begin{cases} \text{Max} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j K(x_i x_j), \\ \quad \text{Subject to:} \\ \quad\quad \sum_{i=1}^{N} \alpha_i y_i = 0, \\ \quad\quad 0 \leqslant \alpha_i \leqslant m_i C, \quad \forall i = 1,\ldots,N \end{cases}$$

and let $\{\alpha_1,\ldots,\alpha_n\} = BM \cup CM \cup NM$ be the set of the Lagrange multipliers, where: BM is the set of the Lagrange multipliers associated to the boundary samples; CM is the set of the Lagrange multipliers associated to the Cord samples; NM is the set of the Lagrange multipliers associated to the noise samples.

As the elements of BM can't be a support vector and to ensure a large generalization, we rewrite the objective function and the constraint of the dual problem:

$$\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i x_j)=\sum_{\alpha_i\in BD}\alpha_i-\frac{1}{2}\sum_{\alpha_i\in BD}\sum_{\alpha_j\in BD}\alpha_i\alpha_j y_i y_j K(x_i x_j)$$

and

$$\sum_{i=1}^{N}\alpha_i y_i=0\Leftrightarrow\sum_{\alpha_i\in BD}\alpha_i y_i=0.$$

Our approach consists solving the following dual model:

$$(RD)\begin{cases}\text{Max}\sum\limits_{\alpha_i\in D}\alpha_i-\frac{1}{2}\sum\limits_{\alpha_i\in BM}\sum\limits_{\alpha_j\in BM}\alpha_i\alpha_j y_i y_j K(x_i x_j),\\ \quad\text{Subject to:}\\ \qquad\sum\limits_{\alpha_i\in BD}\alpha_i y_i=0,\\ \qquad 0\leqslant\alpha_i\leqslant m_i C\quad\forall\alpha_i\in BM\quad\forall i=1,\ldots,N.\end{cases}$$

To ensure more consistency of DBF-SVM, we select a random subset of noisy data that are injected in the learning set and we reject them according to their weights (x having the smallest $|BD\cap B(x,r)|$) during the resolution of the model (RD).

Basing on these theoretical results, we give the main steps of the proposed DBF-SVM that improves the classical SVM:
1) We set the parameters r (radius of the neighborhood) and mp (the neighborhood size) statistically;
2) We determine the noisy-subset, cord-subset and interior-subset;
3) We solve the dual problem basing on the cord-subset data only.

5. Experimentation

The proposed method is tested on several datasets, to know One-Gaussian dataset and Skin-noskin described, respectively, in the section 5.1 and 5.2. This choice is founded on the large numbers law.

Several experimentations have been conducted for all algorithms with different configurations under a compatible HP, Intel (R) Core (TM) i5- 4210U CPU@ 1.70GHz, 2.40 GHz, and 6.00Go GB of RAM through Matlab.

As DBF-SVM depends on two parameters (Min-Points mp and radius r of the neighborhood of a given sample), it will be interesting to study the DBF-SVM consistency. To this end, we calculate the margin, for several values of those parameters, using the formula:

$$M=\frac{2}{\|w\|},\tag{7}$$

where $w=\sum_{i=1}^{ncp}\alpha_i y_i\phi(x_i)$, then $\|w\|^2=\sum_{i=1}^{ncp}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$ with ncp is the number of cord points. Then we estimate the number of vector support for different values of the radius r.

5.1. DBF-SVM test on 1-Gaussian dataset

In the following, we give the description of the 1-Gaussian dataset; see Figure 1.

Dataset Name: 1-Gaussian.
Attribute types: Real numbers.

Instances: Variant $[1000, 50000]$.
Attributes number: 3 (2 inputs+class).
Class number: 2 class.
Description: the first class is obtained by generating $500 \leqslant N \leqslant 25000$ points for a red class distributed as 2-D independent normal with mean (3,0) and unit variance. The second class is obtained by generating $500 < N < 25000$ points for a green class distributed as 2-D independent normal with mean (0,3) and unit variance.
Notation: $1|G|$(instances).

Figure 1 illustrates the obtained results by the classical SVM(kernel=RBF) applied to $1|G|1000$. This example shows that to determine the decision boundary, we need only the border points which reinforces the theoretical result shown in Theorem 1.

Figure 2 illustrates the obtained results by the DBF-SVM ($r = 0.4$; $mp = 12$; kernel=RBF) applied to $1|G|1000$. First, the DBF-SVM determines the three kind of data: cord, border and noise points. Then, we keep only border points (the $1|G|1000$ instance becomes $1|G|151$; see table 1 row 1.

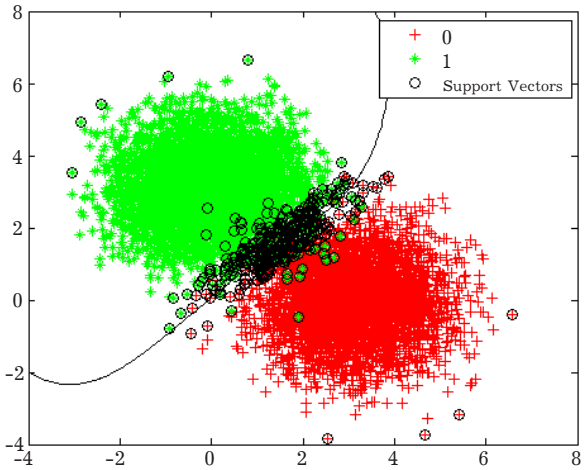

Fig. 1. Boundary decision using classical SVM.

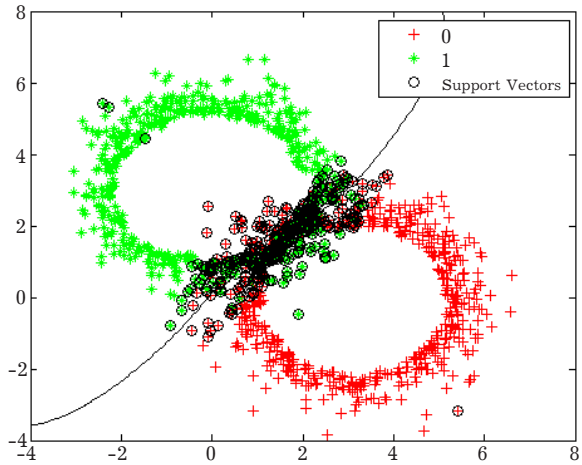

Fig. 2. Boundary decision using DBF-SVM.

Then, the dual problem of this latter is solved using an adequate optimization method (subject discussed at the end of the paragraph 2).

Table 1 gives a statistical study about the remained samples most of which are border points and represent, in mean, 3.34% of the initial data; this percent increases with the size of the 1-Gaussian dataset size (Figure 4). A such diminution permits a large gain in term of cpu time to solve the dual problem. In addition, solving the dual problem basing on, only, the border points reduces the number of local minimum allowing the selection of the optimal support vectors.

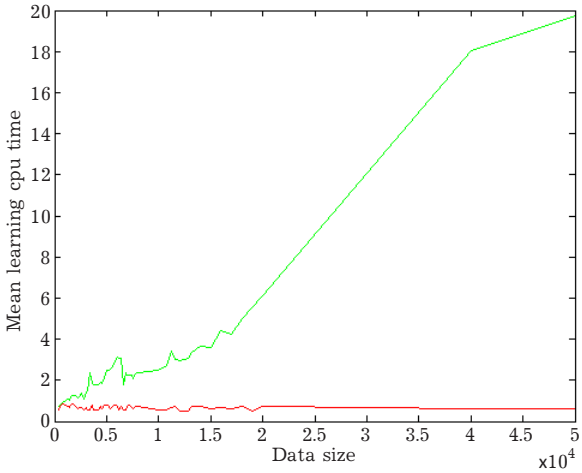

Fig. 3. Learning cpu time versus data size on $1|G|$N instances.

Table 1 presents a comparison between the classical and the proposed DBF-SVM. First, the DBF-SVM provides the best classification rate (see Table 1 columns 3 and 6). Second, the reduction of the dual size permits a remarkable gain in term of time (see Table 1 columns 4 and 7). Third, the performance of the DBF-SVM increases with size of the 1-Gaussian instances (Figure 3); then we recommend the use of the DBF-SVM for large instances of $1|G|N$.

Table 1. Classical SVM vs Density Based SVM on 1-Gaussians with: RBF=kernel function, $r = 0.4$, $mp = 12$.

Method	**Classical SVM**			**Density Based SVM**		
Data size	Dual problem	Mean test correct rate(%)	Mean learning CPU time (s)	Dual problem	Mean test correct rate(%)	Mean learning CPU time (s)
1|G|1000	800	100	0.9375	151	100	0.7656
1|G|1600	1280	98.24	1.1875	161	98.24	0.8125
1|G|2000	1600	97.84	1.2500	129	97.84	0.7188
1|G|2600	2080	96.93	1.3594	143	96.52	0.6719
1|G|3000	2400	97.04	1.4063	119	98.08	0.6719
1|G|3600	2880	96.12	1.8750	119	95.69	0.7969
1|G|4000	3200	96.68	1.7969	96	98.21	0.5781
1|G|4600	3680	95.18	1.8594	103	94.96	0.5938
1|G|5000	4000	94.88	2.5000	95	94.18	0.7813
1|G|5600	4480	95.77	2.6563	121	97.30	0.7344
1|G|6000	4800	95.92	3.1094	125	97.67	0.7500
1|G|6600	5280	96.37	1.7656	103	97.58	0.5313
1|G|7000	5600	95.91	2.2344	107	96.41	0.7500
1|G|7600	6080	95.73	2.0938	116	96.78	0.6406
1|G|8000	6400	95.24	2.3594	89	96.59	0.7188
1|G|10000	8000	95.52	2.4844	102	96.39	0.5313
1|G|11600	9280	95.63	3.0313	104	96.54	0.7031
1|G|15000	12000	96.10	3.6250	86	96.94	0.6094
1|G|16000	12800	95.68	4.3906	105	97.51	0.6563
1|G|17000	13600	96.12	4.2188	109	97.37	0.5938
1|G|18000	14400	95.71	5.0000	101	97.48	0.7031
1|G|19000	15200	95.69	5.5000	110	97.37	0.4844
1|G|20000	16000	94.63	6.0938	90	96.78	0.7344
1|G|40000	28767	95.11	18.0156	103	97.02	0.6094
1|G|50000	27767	94.00	19.7500	106	94.00	0.5938

Table 2. Consistency of the dual solution (Vector support number and SVM marge) associated to 1|G|N instances.

Data size	2000	3000	4000	5000	6000	7000	8000	9000	10000	15000	20000
Marge mean	3.49	3.95	1.11	4.91	3.16	5.82	1.82	2.76	5.34	4.52	2.08
Marge Standard deviation	4.68e-16	0.14	0.02	0	4.68e-16	0	0.14	0	9.36e-16	9.36e-16	4.68e-16

As DBF-SVM depends on the parameter mp, it was obligatory to study it's consistency. In this regard, experiments carried out on several instances of 1-g show that DBF-SVM becomes strongly consistent for values $15 < mp < 30$; see Figure 5.

Table 4 and 3 give the DBF-SVM results compared to the other classifiers (Niave Bayes, MLP, Knn, AdaBoostM1, Random Forest, Random Tree and Classical SVM).

As shown in Table 3, the DBF-SVM confusion matrix contains the most larger number of true positive samples. Basing on this matrix, we have calculated the most known performance measures such as the Precision, the Recall and the F-measure; see Table 4. Our proposed method has the best accuracy.

Table 5 shows the comparison of six classifier methods and our BD-FSVM method, evaluated by the PIMA dataset, also presents how much our model is capable of distinguishing between classes.

From Table 5 it is obvious to notice that our model BD-FSVM tested on the PIMA datasets gives better results, and it shows how our algorithm significantly outperforms classical SVM.

Table 6 shows the comparison of six version of SVM and our version BD-FSVM, evaluated by the PRIMA dataset, also presents how much our model is capable of distinguishing between classes.

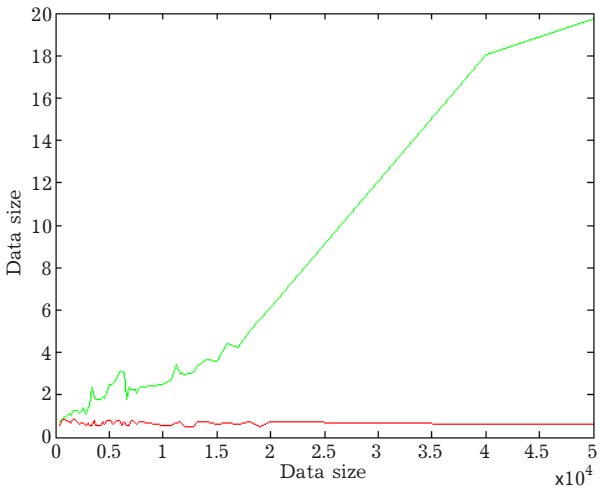

Fig. 4. Dual size versus the instances sizes on $1|G|$N instances.

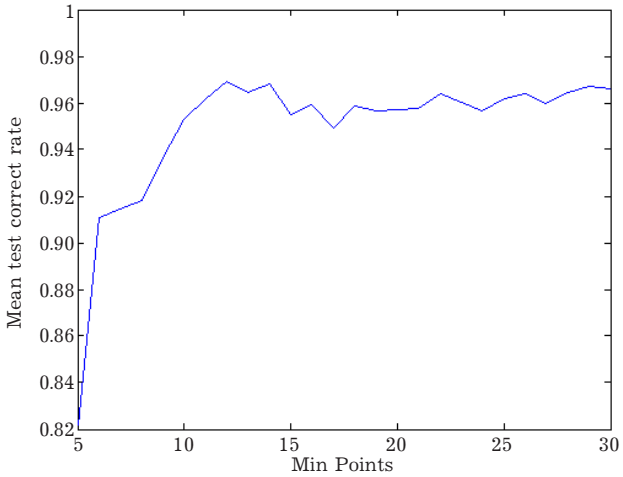

Fig. 5. Test correct rate versus min points on $1|G|$N instances.

Table 3. Confusion matrix of Niave Bayes, MLP, Knn, AdaBoostM1, Random Forest, Random Tree and Classical SVM.

Classical SVM			DB-SVM		
no	yes		no	yes	
596	8	no	592	12	no
453	11785	yes	0	12238	yes
NB			MLP		
no	yes		no	yes	
362	242	no	586	18	no
799	11439	yes	2257	9981	yes
KNN			Adaboost		
no	yes		no	yes	
604	0	no	269	335	no
2024	10214	yes	3068	9170	yes
Random Forest			Random Tree		
no	yes		no	yes	
591	13	no	601	3	no
3101	9137	yes	3653	8585	yes

Table 4. Performance of Classical SVM, DBF-SVM, parametric, non-parametric, and hybrid classifiers on skinnoskin dataset.

Method	Accuracy	Precision	Recall	F-measure
Niave Bayes	91.8 9	31.18	59.93	41.02
MLP	82.28	20.61	97.01	34
Knn	84.23	22.98	100	37.37
AdaBoostM1	73.50	8.06	44.53	13.65
Random Forest	75.75	16.01	97.84	27.51
Random Tree	71.53	14.12	99.50	24.740
Classical SVM	96.41	56.81	98.67	72.11
DB SVM	**99.90**	**100**	**98.01**	**98.99**

From Table 6 it is obvious to notice that our model BD-FSVM tested on the PRIMA datasets gives better results.

Table 5. Comparison between different classification methods and our approach on diabetes dataset.

Method	Accuracy	Precision	Recall	F-measure
Niave Bayes	79.3	67.2	62.50	69.40
MLP	76.6	61.1	57.40	68.30
Knn	80.90	68.4	64.50	66.60
AdaBoostM1	81.10	69.2	70.60	66.50
Decision Tree	79.90	64.8	72.80	68.70
Classical SVM	79.70	70.7	55.6	62.7
DB SVM	**89.90**	**83.4**	**81.6**	**78.6**

Table 6. Comparison between different version of SVM and our approach for diabetes dataset.

Method	Accuracy	Precision	Recall	F-measure
Twin SVM	0.74	0.58	0.75	0.66
OC-SVM	0.51	—	—	—
Least Squares SVM	0.79	0.67	0.61	0.68
Nu-SVM	0.77	0.66	0.64	0.64
Fuzzy SVM	0.81	0.65	0.66	0.70
Classical SVM	0.79	0.70	0.55	0.62
DB-FSVM	**0. 90**	**0.83**	**0.82**	**0.79**

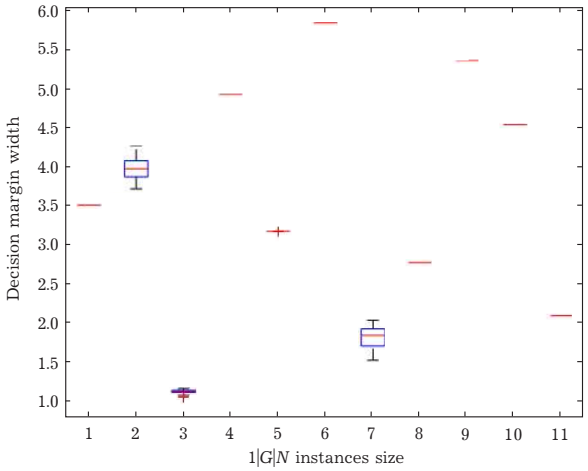

Fig. 6. Box plot of 11 instances of $1|G|N$ for for different values of r.

To study the consistency of the DBF-SVM, with respect to the r parameter, the radius of the neighborhood, we carried out a small perturbation to r, for several instances of $1|G|N$ for different values of N. It should be noted that the perturbation is generated randomly between 0.0001 and 0.001. This perturbation leads to different separation margins that have different widths. These latter are calculated using the formula (7). Then a statistical study were carried out by calculating the arithmetic mean and the standard deviation of the widths of the obtained separation margins; see Table 2. The small size of the box plot, given in Figure 6, shows the strong consistency of our method DBF-SVM when changing the parameter r.

6. Conclusion

In this work, we proposed a new version of SVM called the DB-FSVM as improvement of the classical one. The method considers a non-negative real number allowing the data division to three types (noise, border and cord). We used the Pima Indian Diabetes (PID) dataset to evaluate our method. The experimental results indicate that the proposed method improves the prediction accuracy and DB-FSVM significantly outperforms all other versions of SVM as well as literature methods of classification. We show that our model BD-FSVM tested on the PIMA datasets gives better results and an accuracy of 90%.

[1] WHO. Diabetes. Available online: https://www.who.int/news-room/fact-sheets/detail/diabetes (accessed on 1 June 2020).

[2] IDF Diabetes Atlas, A.D. Type 2 Diabetes. Available online: https://www.idf.org/aboutdiabetes/type-2-diabetes.html (accessed on 20 March 2020).

[3] El Moutaouakil K., Touhafi A. A New Recurrent Neural Network Fuzzy Mean Square Clustering Method. 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech). 1–5 (2020).

[4] Vapnik V. N. The Nature of Statistical Learning Theory. Springer Science and Business Media (1999).

[5] Burges C. J. C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery. **2**, 121–167 (1998).

[6] Vapnik V. N., Chervonenkis A. Ya. A class of algorithms for pattern recognition learning. Avtomat. i Telemekh. **25** (6), 937–945 (1964).

[7] El Moutaouakil K., El Ouissari A., Touhafi A., Aharrane N. An Improved Density Based Support Vector Machine (DBSVM). 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech). 1–7 (2020).

[8] Mercer J. XVI. Functions of positive and negative type, and their connection the theory of integral equations. Philosophical Transactions of The Royal Society of London. Series A. **209** (441–458), 415–446 (1909).

[9] Mangasarian O. L. Generalized Support Vector Machines. Advanced in Large Margin Classifiers. 135–146 (2000).

[10] Lin C. F., Wang S. D. Fuzzy support vector machines. IEEE transactions on neural networks. **13** (2), 464–471 (2002).

[11] Schölkopf B., Smola A. J., Williamson R. C., Bartlett P. L. New support vector algorithms. Neural computation. **12** (5), 1207–1245 (2000).

[12] Suykens J. A. K., Vandewalle J. Least squares support vector machine classifiers. Neural processing letters. **9** (3), 293-300 (1999).

[13] Schölkopf B., Platt J. C., Shawe-Taylor J., Smola A. J., Williamson R. C. Estimating the support of a high-dimensional distribution. Neural computation. **13** (7), 1443–1471 (2001).

[14] Bi J., Zhang T. Support vector classification with input data uncertainty. Advances in neural information processing systems. 161–168 (2005).

[15] Yang X., Song Q., Cao A. Weighted support vector machine for data classification. Proceedings. 2005 IEEE International Joint Conference on Neural Networks. **2**, 859–864 (2005).

[16] Bi J., Vapnik V. N. Learning with rigorous support vector machines. Learning Theory and Kernel Machines. 243–257 (2003).

[17] Tang Y., Jin B., Sun Y., Zhang Y. Q. Granular support vector machines for medical binary classification problems. 2004 Symposium on Computational Intelligence in Bioinformatics and Computational Biology. 73–78 (2004).

[18] Lee Y. J., Mangasarian O. L. SSVM: A smooth support vector machine for classification. Computational optimization and Applications. **20** (1), 5–22 (2001).

[19] Lee Y. J., Mangasarian O. L. RSVM: Reduced support vector machines. Proceedings of the 2001 SIAM International Conference on Data Mining. 1–17 (2001).

[20] Schölkopf B., Smola A. J., Bach F. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2002).

[21] Mangasarian O. L., Wild E. W. Proximal support vector machine classifiers. Proceedings KDD-2001: Knowlborder discovery and data mining (2001).

[22] Mangasarian O. L., Wild E. W. Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE transactions on pattern analysis and machine intelligence. **28** (1), 69–74 (2005).

[23] Khemchandani R., Chandra S. Twin support vector machines for pattern classification. IEEE Transactions on pattern analysis and machine intelligence. **29** (5), 905–910 (2007).

[24] Cortes, C., Vapnik, V. Support-vector networks. Machine learning, 20(3), 273-297 (1995).

[25] Wang Y., Wang S., Lai K. K. A new fuzzy support vector machine to evaluate credit risk. IEEE Transactions on Fuzzy Systems. **13** (6), 820–831 (2005).

[26] Huang H. P., Liu Y. H. Fuzzy support vector machines for pattern recognition and data mining. Int. J. Fuzzy Syst. **4**, 826–835 (2002).

[27] Batuwita R., Palade V. FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Transactions on Fuzzy Systems. **18** (3), 558–571 (2010).

[28] Yu H., Sun C., Yang X., Zheng S., Zou H. Fuzzy support vector machine with relative density information for classifying imbalanced data. IEEE transactions on fuzzy systems. **27** (12), 2353–2367 (2019).

[29] Khanam J. J., Foo S. Y. A comparison of machine learning algorithms for diabetes prediction. ICT Express. (2021).

[30] Tigga N. P., Garg S. Prediction of type 2 diabetes using machine learning classification methods. Procedia Computer Science. **167**, 706–716 (2020).

[31] Shuja M., Mittal S., Zaman M. Effective prediction of type ii diabetes mellitus using data mining classifiers and SMOTE. Advances in computing and intelligent systems. 195–211 (2020).

[32] Devi R. D. H., Bai A., Nagarajan N. A novel hybrid approach for diagnosing diabetes mellitus using farthest first and support vector machine algorithms. Obesity Medicine. **17**, 100152 (2020).

Адаптивний метод опорних векторів на основі функції щільності: застосування до набору даних про діабет

Ель Уісарі А., Ель Мутауакіл К.

Лабораторія інженерних наук, Полідисциплінарний факультет Тази, Марокко

У роботі запропоновано систему глибокого прогнозування діабету, засновану на новій версії моделі машинної оптимізації опорних векторів. Спочатку визначаються три типи пацієнтів (шум, зв'язкові та внутрішні) на основі конкретних параметрів. Далі врівноважуються набори клінічних даних, пригнічуючи шумних та зв'язкових пацієнтів. Після того визначаються вектори підтримки, розв'язуючи програму оптимізації розумного розміру. Запропонована система виконується на добре відомому наборі даних про діабет PIMA. Результати експериментів показують, що запропонований метод покращує точність прогнозування, а запропонована система значно перевершує всі інші версії SVM, а також літературні методи класифікації.

Ключові слова: *метод опорних векторів, машинне навчання, оптимізація, класифікація, діабет.*