Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 3** : 1–12.
ISSN-0037-8712
doi:10.5269/bspm.79206

Difference of Convex Functions Optimization for Feature Selection in Granular Ball Support Vector Machine

Najoua Aafar*, Abdellatif El Ouissari and Bouchaib Ferrahi

ABSTRACT: Feature selection constitutes a critical optimization problem within the domain of supervised pattern classification. It involves selecting an optimal subset of features that maximizes the retention of the data's salient information. Granular Ball Support Vector Machine (GBSVM) has proven to be a powerful technique for enhancing the predictive accuracy and computational tractability of classification models, by exploiting the concept of granular structures in the feature space, through the generation of a set of granular balls, enabling complex decision boundary modeling and adaptability to data variability. This paper presents a novel embedded feature selection approach in the context of granular ball SVM, directly enhancing classifier performance. Our approach to the resulting optimization problem is to apply Difference of Convex (DC) functions programming to effectively handle the non-convex nature of the problem. Genetic algorithm is used to tune the model's parameters. Experimental results on UCI datasets show the efficiency of the proposed method.

Key Words: DC Programming, non-convex optimization, feature selection, Support Vector Machine (C-SVM), Granular Ball Support Vector Machine (GBSVM), genetic algorithm.

Contents

1. Introduction

Support Vector Machines (SVMs) [1] is among the most recognized machine learning algorithms for classification. This method demonstrated strong predictive accuracy due to its capaciy to reduce the model complexity [1]. In contrast to models that chase every data nuance and might overfit, support vector machine tries to find the simplest, most robust hypothesis that can explain the data clearly. This is achieved through three core mechanisms: *The pursuit of the maximum margin*: support vector machine classifier isn't just about finding any decision boundary that separates classes. It specifically seeks the optimal hyperplane that maximizes the margin. *Using the Kernel Trick*: The data is often not linearly separable specially the real-world data. The SVM gets around this pitfall with a mathematical

* Corresponding author.
2020 *Mathematics Subject Classification:* 90C26, 62H30, 68T05.
Submitted September 27, 2025. Published January 22, 2026

workaround called the kernel trick. Instead of going through the computationally expensive hassle of transforming all the data points into a high-dimensional space (which is computationally expensive), the kernel function calculates the similarity (or dot product) between pairs of points as if they had been transformed. The most common kernels are the Polynomial Kernel and the Radial Basis Function (RBF) Kernel. *Built-in regularization*: A parameter C in SVM dictates the penalty for each data point that end up falling on the wrong side of the margin (a misclassification or a violation of the margin). If the value of C value is low, the model prioritizes maximizing the overall margin above all else, even if it means tolerating a few misclassified training points, and if C takes high values, that means the model would avoid misclassifying training points. This can lead to a narrower margin and a more complicated decision boundary. Because of its widespread use, a broad range of SVM extensions and applications have been presented in the literature [2,3,4,5,6,7]. Within supervised pattern recognition frameworks, feature selection focuses on choosing a subset of the original input dimensions (features) for different purposes such as performance problems through enabling easier data gathering and minimizing storage requirements and classification time, feature selection also helps conducting semantics analysis and yet understanding the problem, it also helps to avoid the "curse of dimensionality" which leads to a better prediction accuracy. Feature selection methods may be classified into filters, wrappers and embedded approaches [8,9,10]. Filter methods represent the most prevalent approach, it operates as a preprocessing stage independent of the classifier [11,12]. On the other hand, wrappers treat the classifier as a black box [9,13]. For embedded approaches, they jointly optimize both feature selection and classifier parameters during model training. SVM classifiers is limited by its inability to perform embedded feature selection during the classifier construction [14,15]. Researchers have proposed various methods to integrate feature selection with SVM frameworks [14] such as penalty functions e.g. LASSO (Least Absolute Shrinkage and Selection Operator) and concave approximations of the zero norm [10,16], and several studies have adopted a balancing methodology with double regularizers to address this same issue [17,18].

Granular Ball Computing [19] proposed by Xia et al. is the approach based on the generation of granular balls of different sizes from raw datasets, completelty or partially covering the data manifold while conserving the essential characteristics of the data, enabling the treatement of problems with limited noise robustness. Granular Ball Support Vector Machine (GBSVM) [20] is a new extension of SVM based on granular ball computing, founded on the generation of granular balls to use as the input instead of individual data points. Notwithstanding its novel introduction, GBSVM proved its resilience in terms of computational efficiency, system scalability, model adaptability and algorithmic robustness, and showed promising potential for future development through several works and extensions e.g. [21,22].

The present work aims to explore the idea of double regularized SVM classifiers to granular ball support vector machine using l_2-norm and l_0-"norm". The first objective of our work is to combine between the benefits of the new adoption of input data gained by GBSVM and the double regularization approach, by adding an other regularization term to the l_2-norm, and yet gaining a simultanious classification and attributes selection, in the aim of creating a classifier that benefits from both sides. Our approach requires the solution of a non-convex optimisation problem, we hence implement a Difference-of-Convex (DC) programming approach [23,24] with appropriate problem decomposition. The remainder of this paper is structured as follows: we give in section 2 an overview of related works. In section 3, we introduce our approach. The results of the experiments are presented in section 4.

2. Related work

2.1. Support Vector Machines (SVM)

SVM are a class of supervised learning algorithms, they represent a standard technique for classification and regression tasks. The primary goal of SVM is to determine the optimal partitioning hyperplane of different classes by maximizing the margin. Given a set of training data:

$$(x_1, y_1), \ldots, (x_d, y_d) \in \mathbb{R}^n \times \{\pm 1\}$$

where x_i represents an n-dimensional input variable and y_i are their labels, SVM seeks to find a hyperplane $f(x) = w^T x + b$ that maximizes the margin between the two classes, where $w, x \in \mathbb{R}^n$, $b \in \mathbb{R}$. A

hyperplane $f(x) = w^T x + b = 0$ linearly separates the training instances of the two classes, if and only if, $y_i(w^T x_i + b) \geq 1, i = 1, \ldots, d$. The SVM optimization problem is defined as [27,28]:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\text{s.t. } y_i(w^\top x_i + b) \geq 1, \quad i = 1, \ldots, d \end{aligned} \tag{2.1}$$

For non-separable cases, slack variables ξ are introduced to allow for misclassifications, and the optimization problem becomes:

$$\begin{aligned} &\min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{d} \xi_i \\ &\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, d \\ &\qquad \xi_i \geq 0, \quad i = 1, \ldots, d \end{aligned} \tag{2.2}$$

where $C > 0$ is a regularization parameter that controls the trade-off between maximizing the margin and minimizing classification errors.

2.2. Granular ball computing

Drawing on granular computing principles, Xia [29] et al. introduced granular ball computing in the aim of enhancing system robustness and reduce uncertainty [25]. This approach is based on transforming the original sample into granular balls that either completely or partially cover the sample and use them as input for the model instead of individual sample points, leading to a reduction in training samples and significantly improving algorithm efficiency. Let consider a dataset $D \in \mathbb{R}^d$, and let GB_i be a granular ball that has a center s_i and a radius r_i. $GB_1, GB_2, \ldots, GB_N$ is the list of granular balls. The center s_i of each ball denotes the centroid of all data points in GB_i, and r_i is the mean distance from the centroid to all points in GB_i. The label of GB_i is selected based on which label occurs the most within GB_i. It's worth noting that as r_i decreases, the granular ball becomes finer, and if it increases the granular ball becomes coarser. The coverage of a granular ball can be formally modeled in the following model. Given a dataset $D = \{x_1, x_2, \ldots, x_d\}$. Granular ball list $T = \{GB_1, GB_2, \ldots, GB_N\}$ denotes the set of granular balls constructed from dataset D. The corresponding optimization problem is formulated as follows:

$$\begin{aligned} &g(x, \beta) \to h(GB, \gamma) \\ &\text{s.t.} \quad \min\left(\frac{d}{\sum_{j=1}^{N} |GB_j|} + N\right) \\ &\qquad \text{Quality}(GB_j) \geq pur \end{aligned} \tag{2.3}$$

where β and γ are the parameter vectors and pur is the purity threshold.

2.3. Granular Ball Support Vector Machines (GBSVM)

Introduced by Xia et al. [26], based on granular ball computing, and in the purpose of enhancing robustness and adressing uncertainties in classification problems, Granular Ball Support Vector Machines (GBSVM) represents a recent extension of the baseline SVM. The core concept of GBSVM is to model data points as granular balls (clusters of data points) instead of individual points. This approach yields a classification model that is both robust and efficient, particularly useful in situations involving noisy data or outliers. This method enables the model to consider the internal structure of data clusters, ending up with accurate and stable classification boundaries. The primal optimization problem for an inseparable GBSVM is as follows:

$$\begin{aligned} &\min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \\ &\text{s.t.} \quad y_i(w^\top s_i + b) - \|w\| r_i \geq 1 - \xi_i, \quad \forall i \in \{1, \ldots, m\} \\ &\qquad \xi_i \geq 0, \quad \forall i \in \{1, \ldots, m\} \end{aligned} \tag{2.4}$$

m is the number of granular balls, ξ_i are slack variables, C is a penalty coefficient, s_i is the center, and r_i is the radius of the granular ball GB_i.

3. Feature Selection for Granular Ball Support Vector Machine

3.1. l_2-l_0-GBSVM

While the l_2 regularization is crucial for achieving high predictive accuracy in the SVM models, l_0 penalty term is employed to induce sparsity and perform feature selection [30]. Inspired by the work of Neumann et al. [17], we suggest to combine these terms in Granular Ball Support Vector Machine to atain double income. It is important to note that the l_0-"norm," $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$ is actually a pseudo-norm because, when l_p-norms ($p > 0$) verify the triangle inequality, the former does not. The optimization problem is written as follows:

$$\begin{aligned} &\min_{w,b,\xi_i} \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\xi_i + \nu\|w\|_0 \\ &\text{s.t.} \quad y_i(w^\top s_i + b) - \|w\|r_i \geq 1 - \xi_i, \quad \forall i \in \{1,\dots,m\} \\ &\qquad \xi_i \geq 0, \quad \forall i \in \{1,\dots,m\} \end{aligned} \tag{3.1}$$

where $\mu, \nu \in \mathbb{R}^+$ are two weight parameters
The associated unconstrained optimization problem is written this way:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\left(1 - y_i(w^\top s_i + b) + \|w\|r_i\right)_+ + \nu\|w\|_0 \tag{3.2}$$

where $x_+ := max(x, 0)$.

Over the years, three approaches have emerged to adress the discontinuity at the origin of the l_0-"norm", these pertain to convex approximation, nonconvex approximation and nonconvex exact reformulation. Since the l_0-"norm" is non-smooth, we are going to use the concave approximation proposed in [10] for the sake of the method thereon:

$$\|w\|_0 \approx e^T\left(e - exp(-\alpha|w|)\right) \tag{3.3}$$

$\alpha \in \mathbb{R}^+$ is an approximation parameter and e is the vector of ones. And the problem we name $l_2 - l_0$-GBSVM reads:

$$\begin{aligned} &\min_{w,b,\xi_i,v} \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\xi_i + \nu e^T\left(e - exp(-\alpha v)\right) \\ &\text{s.t.} \quad y_i(w^\top s_i + b) - \|w\|r_i \geq 1 - \xi_i, \quad \forall i \in \{1,\dots,m\} \\ &\qquad \xi_i \geq 0, \quad \forall i \in \{1,\dots,m\} \\ &\qquad -v \leq w \leq v \end{aligned} \tag{3.4}$$

So, this yields the mathematical program:

$$\min_{w,b,v} \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\left(1 - y_i(w^\top s_i + b) + \|w\|r_i\right)_+ + \nu e^T\left(e - exp(-\alpha v)\right) + \chi_{[-v;v]}(w) \tag{3.5}$$

here χ_Ω is given by $\chi_\Omega(x) = 0$ if $x \in \Omega$, and $+\infty$ otherwise.

The resulting problem is a non-convex problem that can be written as difference of convex (DC) functions programming problem.

3.2. DC programming and DCA

At its core, DC (Difference of Convex functions) programming involves decomposing a function $f(x)$ into difference of convex functions for tractable optimization. DC programming represents a powerful optimization framework [31,32,33,34], notably, for training Support Vector Machine, especially for formulations deviating from classical convexity assumptions. To handle these nonconvex problems, DC programming and DCA are often utilized, see e.g. [35,36,37,38]. DCA [39] is an iterative numerical procedure designed to address challenging non convex optimization problems. It works by breaking down the problem into a series of simpler convex optimizations. Extensive related research contributions have been presented in the litterature by Le Thi Hoai An, Pham Dinh Tao, see e.g. [24,23,40,41]. For more details of the uses of DC programming and DCA in the context of SVM see [42].

We denote $\Gamma_0(\mathbb{R}^n)$ the set of functions $u : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that u is a lower-semicontinuous, proper and convex function, Consider the following optimization problem:

$$\min_{x\in\mathbb{R}^n} \quad \big\{f(x) = g(x) - h(x)\big\} \tag{3.6}$$

Definition 3.1 *Let C be a subset of $\mathbb{R}^n$, and C is convex. And let f be a function that maps from $\mathbb{R}^n$ to the real numbers $\mathbb{R}$. We call this function a DC -that's short for Difference of Convex - function on C, if there exists a pair of convex functions g and h (also from $\mathbb{R}^n$ to $\mathbb{R}$), such that f can be expressed as:*

$$f(x) = g(x) - h(x) \tag{3.7}$$

Any function that can be expressed in the form of equation (3.7) is is called a DC decomposition of the original function f.

Definition 3.2 *We say a function $f : \mathbb{R}^n \to \mathbb{R}$ is a locally DC function, if for any given point x_0 in $\mathbb{R}^n$, you can always find a ball $B(x_0;\epsilon) = \{x \in \mathbb{R}^n : \|x - x_0\| \le \epsilon\}$ where the function f is DC.*

A DC problem, such that $x \in \Omega$ is a convex contraint, can be rewritten in the following form:

$$\min_{x\in\Omega}\{f(x) = g(x) - h(x)\} = \min_{x\in\mathbb{R}^n}\{g(x) + \chi_\Omega(x) - h(x)\},$$

with χ_Ω is the indicator function of Ω.

For $\phi \in \Gamma_0(\mathbb{R}^n)$, the following notations are used : [43,44]

- $\operatorname{dom}\phi = \{x \in \mathbb{R}^n \mid \phi(x) < \infty\}$ is the domain of ϕ .
- $\phi^*(\tilde{x}) = \sup_{x\in\mathbb{R}^n}\{\langle x, \tilde{x}\rangle - \phi(x)\}$ is the conjugate function of ϕ.
- $\partial\phi(z) = \{\tilde{x} \in \mathbb{R}^n \mid \phi(x) \ge \phi(z) + \langle x - z, \tilde{x}\rangle, \forall x \in \mathbb{R}^n\}$ is the subdifferential of ϕ.

for $z, \tilde{x} \in \mathbb{R}^n$. Given a differentiable function, we have $\partial\phi(z) = \{\nabla\phi(z)\}$. The necessary local optimality condition for DC programming 3.6 is :

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*).$$

A point $x^* \in \operatorname{dom}(f)$ is a critical point of 3.6 if it verifies $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$. If the function f is locally convex at x^*, or if the DC program is polyhedral, then for local optimality, the necessary condition is also sufficient.
[Theorem 23.5] in [43] assures:

$$\partial f(x) = \arg\max_{\tilde{x}\in\mathbb{R}^d}\{x^T\tilde{x} - f^*(\tilde{x})\}, \quad \partial f^*(\tilde{x}) = \arg\max_{x\in\mathbb{R}^d}\{\tilde{x}^T x - f(x)\} \tag{3.8}$$

Difference of Convex functions Algorithm (DCA) $(g, h, \mathbf{tol})$

Initialize $x^0 \in \operatorname{dom} g$
for $k \in \mathbb{N}_0$ **do**
  Select $y^k \in \partial h(x^k)$
  Select $x^{k+1} \in \partial g^*(y^k)$
  if $\min\left(|x_i^{k+1} - x_i^k|, \frac{|x_i^{k+1} - x_i^k|}{|x_i^k|}\right) \leq \text{tol} \quad \forall i = 1, \ldots, d$ **then**
    return (x^{k+1})
  end if
end for

Theorem 3.1 (DCA convergence) *For $g, h \in \Gamma_0(\mathbb{R}^n)$ so that $\operatorname{dom} g \subset \operatorname{dom} h$ and $\operatorname{dom} h^* \subset \operatorname{dom} g^*$, then for the DC algorithm the following holds true:*

- *The sequence $(x^k)_{k\in\mathbb{N}_0}$, $(y^k)_{k\in\mathbb{N}_0}$ are well defined.*
- *The sequence $(f(x^k) = g(x^k) - h(x^k))_{k\in\mathbb{N}_0}$ is monotonously decreasing.*
- *Every limit point of the sequence $(x^k)_{k\in\mathbb{N}_0}$ is a critical point of $f = g - h$. Furthermore, if the algorithm stalls such that $f(x_{k+1}) = f(x_k)$, then the iterate x_k is itself a critical point in 3.6.*

The algorithm's convergence leads to a local minimum influenced by both the initial point x^0 and the specific DC decomposition of the objective function 3.6. When the solution obtained is not global, the DC algoritm can be reinitialized with different starting points. Empirical evidence from Pham and Hoai's 1998 [45] research suggests that DCA implementations frequently achieve global optimal solutions despite the local convergence properties.

3.3. DCA for l_2-l_0-GBSVM

It is clear that the problem studied can be written as a difference of convex programming problem. A viable DC decomposition reads:

$$g(w, b, v) = \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\left(1 - y_i(w^\top s_i + b) + \|w\|r_i\right)_+ + \chi_{[-v;v]}(w)$$

and

$$h(v) = \nu e^T\left(e - exp(-\alpha v)\right)$$

h is a differentiable function, so the first step of DCA iteration $(k \in \mathbb{N}_0)$ becomes:

$$y^k = \nabla h(x^k)$$

By combining the two steps of DCA and using 3.8 for each k, we have:

$$x^{k+1} \in \partial g^*(\nabla h(x^k)) = \arg\max_x \left\{\nabla h(x^k)^T x - g(x)\right\}$$

This gives us the constrained convex quadratic problem:

$$\begin{aligned} \min_{w,b,\xi_i,v} \ & \frac{1}{2}\|w\|^2 + \frac{\mu}{m}\sum_{i=1}^{m}\xi_i + \nu\alpha v^T exp(-\alpha v^k) \\ \text{s.t.} \quad & y_i(w^\top s_i + b) - \|w\|r_i \geq 1 - \xi_i, \quad \forall i \in \{1, \ldots, m\} \\ & \xi_i \geq 0, \quad \forall i \in \{1, \ldots, m\} \\ & -v \leq w \leq v \end{aligned} \tag{3.9}$$

The function f is bounded below, therefore by using Theorem 3.1, the sequence generated by solving these quadratic programs (QPs) is guaranteed to converge. Since $y^k \in \partial h(x^k)$ and consequently $y^{k-1} \in \partial g(x^k)$,

since $g^*(y^{k-1}) = \langle x^k, y^{k-1}\rangle - g(x^k)$ then $f(x^k) = \langle x^k, y^{k-1}\rangle - g^*(y^{k-1}) - h(x^k) \leq f^*(y^{k-1}) \leq h^*(y^{k-1}) + g(x^{k-1}) - \langle x^{k-1}, y^{k-1}\rangle \leq g(x^{k-1}) - h^{**}(x^{k-1}) \leq f(x^{k-1})$. In case the convergence is not attained in finitely many iterations, having $f(x)$ bounded from below, this assures convergence as there exists a such that $\lim_{k\to\infty} f(x^k) = a$. Take $\tilde{x}, \tilde{y}$ to be two accumulation points of x^k and y^k respectively, then $f(\tilde{x}) = f^*(\tilde{x})$. Similarly $\langle \tilde{x}, \tilde{y}\rangle = h(\tilde{x}) + h^*(\tilde{x})$, this gives us the desired results $\tilde{x} \in \partial g(x^k) \cap \partial h(x^k)$.

4. Experimental results

4.1. Genetic Algorithm for Parameter Optimization

Genetic Algorithms (GAs) are evolutionary computation techniques inspired by biological evolution, employing concepts such as selection, crossover, and mutation to optimize solutions to complex problems [46]. Genetic Algorithms are population-based metaheuristics, they maintain a set of candidate solutions (individuals) that evolve iteratively through the strategic application of genetic operations. Each individual's fitness is evaluated according to an objective function, and the better solutions have higher probabilities of being selected for reproduction. The fundamental components are as follows: *Representation* in which the solutions are encoded as chromosomes (typically binary strings or real-valued vectors), *Selection* where fitter individuals are chosen to pass their genes to the next generation, *Crossover* where pairs of selected individuals combine their genetic material to produce offspring, and *Mutation* in which random modifications introduce diversity into the population. We used Genetic Algorithms for hyperparameter optimization as they are effictive in exploring large parameter spaces without requiring gradient information, besides the fact that they can escape local optima thanks to their stochastic nature [47].

Table 1: Datasets used

Dataset	Number of samples	Number of features	Classes	Class distribution
Breast Cancer	569	30	2	(357, 212)
Ionosphere	351	57	2	(225, 126)
Heart	1025	13	2	(499, 526)
Credit Approval	690	46	2	(383, 307)
Statlog (Heart)	270	13	2	(150, 120)
Parkinson's	195	22	2	(147, 48)
SPECT Heart	349	44	2	(254, 95)

4.2. Experimental Setup

For parameter optimization, we implemented a genetic algorithm with the following configuration: for selection, we used tournament selection with size 3, and employed blend crossover with ($\beta = 0.5$), as for mutation, we applied Gaussian mutation ($\sigma = 0.1$). We used a population size of 20 individuals. This balance between exploration and computational efficiency was chosen based on common practices in medium-dimensional optimization problems [48]. In addition, we used 10 generations, given that optimization showed convergence within this number of iterations. The crossover probability (CXPB), and the mutation probability were set repectively to 0.7 and 0.2. The parameter search space was constrained as follows:

For $l_2 - l_0$-GBSVM and $l_2 - l_0$-SVM, μ is explored in $[0.01, 1.0]$, as for α in $[0.001, 10.0]$ and ν in $[0.001, 0.1]$. For $l_2 - l_0$-GBSVM the purity threshold (pur) is chosen in $[0.7, 1.0]$ and the minimal number of samples within a granular ball (num) is in $\{2, ..., 10\}$. The tolerance of DCA is is set to 10^{-5}. For l_1-SVM, C is explored in $[0.01, 10.0]$. The number of features is determined as $|\{j = 1, \ldots, d : |w_j| \geq 10^{-8}\}|$. For each dataset, we conducted training and testing in an 7:3 ratio. The fitness function used 3-fold stratified cross-validation accuracy to evaluate each candidate solution, with class weights fixed at 2.0 for the minority class and 1.0 for the majority class to address imbalance.

4.3. Results and discussion

The numerical results for the algorithm are presented in this section. The experiments were conducted on a computer running with Windows 11 operating system with the configuration of 11th Gen Intel(R)

Table 2: Parameters selected by the genetic algorithm and average accuracy

Dataset	l_1-SVM Average accuracy (C)	l_2-l_0-SVM Average accuracy (μ,α,ν)	$l_2-l_0-GBSVM$ Average accuracy (pur,μ,α,ν,num)
Breast Cancer	**0.9708**	0.9520	0.9333
	(0.0448)	(0.9385, 0.4789, 0.0126)	(0.7971, 0.6339, 2.5616, 0.0248, 4)
Ionosphere	**0.8981**	0.8660	0.8708
	(2.0304)	(0.9130, 0.2088, 0.0098)	(0.7278, 0.4051, 8.7220, 0.0553, 6)
Heart	0.8364	0.8273	**0.8398**
	(7.9070)	(0.4341, 3.9421, 0.0287)	(0.7301, 0.7905, 0.1594, 0.0370, 6)
Credit Approval	0.8773	0.8213	**0.8980**
	(0.7507)	(0.1796, 2.9025, 0.0061)	(0.6378, 0.6808, 8.5460, 0.1331, 5)
Statlog (Heart)	0.8395	**0.8494**	0.8247
	(1.8902)	(0.7232, 6.1031, 0.0052)	(0.7393, 0.4134, 2.9088, 0.1683, 8)
Parkinson's	**0.8475**	0.8203	0.7458
	(7.1263)	(0.0284, 0.6341, 0.0063)	(0.7060, 0.6418, 0.2013, 0.0673, 6)
SPECT Heart	0.8076	0.7238	**0.8138**
	(0.7557)	(0.5570, 1.0529, 0.0566)	(0.7916, 0.3837, 2.2876, 0.0996, 7)

Core(TM) i5-1145G7 @ 2.60GHz 1.50 GHz, 8 GB RAM memory. We conducted our experiments on 7 benchmark datasets: Breast Cancer, Ionosphere, Heart, Credit approval, Statlog (Heart), Parkinson's [49] and SPECT Heart see Table 1. We compare the results of our model with two other models that apply embedded feature selection, which are l_1-SVM [50] and l_2-l_0-SVM presented in [17]. The genetic algorithm was used to tune all the parameters for the three models to gain fair comparison.

Table 3: Average number and percentage of selected features

Dataset	l_1-SVM	l_2-l_0-SVM	l_2-l_0-GBSVM
Breast Cancer	**9.2**	28.6	14.8
	30.66%	95.33 %	49.33 %
Ionosphere	31.0	32.4	**19.6**
	54.38 %	56.84 %	**34.38**%
Heart	13.0	12.3	**11.8**
	100 %	94.61 %	**90.77**%
Credit Approval	**31.4**	35.2	33.2
	68.26%	76.52 %	72.17 %
Statlog (Heart)	13	**11.8**	12.8
	100 %	**90.78**%	98.46 %
Parkinson's	20.6	21.4	**2.4**
	93.64 %	97.27 %	**10.91**%
SPECT Heart	36.2	28.0	**4.5**
	82.27 %	63.64 %	**12.27**%

The best parameters for each model are represented in table 2. The best parameters are applied for 5 times run, the average values of several metrics for each model are presented in tables 2, 3 and 4, including average accuray, average number and average percentage of selected features and average training time.

According to table 2, the three models show very good performance in terms of average accuracy, with comparable results, for example, for the credit approval dataset l_2-l_0-GBSVM has the best results (89.80%) versus 87.73% and 82.13% respectively for l_1-SVM and l_2-l_0-SVM, similarly it has the highest accuracy for heart and SPECT heart datasets 83.98% and 81.38% respectively. For breast cancer, ionosphere and parkinson's datasets, l_1-SVM has the highest accuracy with 89.81% followed by l_2-l_0-SVM and l_2-l_0-GBSVM with very close outputs (86.60% and 87.08% respectively) for ionosphere, 97.08%

for breast cancer and 84.75% for parkinson's. According to table 3, $l_2 - l_0$-GBSVM selects the lowest number of features for most of the datasets including ionosphere, heart, parkinson's and SPECT heart, with respective percentages of 19.6%, 11.8%, 10.91% and 12.27%. while the l_1-SVM and $l_2 - l_0$-SVM have the following respective percentages for these same datasets (54.38% and 56.84% for ionosphere, 100% and 94.61% for Heart , 93.64% and 97.27% for parkinson's and 82.27% and 63.64% for spect heart). As for breast cancer and credit approval l_1-SVM has the best feature selection results (30.66% and 31.4% respectively) versus 95.33% and 76.52% respectively for breast cancer and credit apprival for $l_2 - l_0$-SVM, and for $l_2 - l_0$-GBSVM, the respective percentages of selected features for breast cancer and credit approval are 49.33% and 72.17%. As for statlog dataset, $l_2 - l_0$-SVM has the the minimum number of selected features 90.78% , whereas $l_2 - l_0$-GBSVM selected 98.46% and l_1-SVM 100%. Over all, and based on the aforementionned results, we can say that $l_2 - l_0$-GBSVM model makes a good compromise between accuracy and number of selected features despite the use of just the centers of the balls as the input data.

Table 4: Average Training Time

Dataset	l_1-SVM	$l_2 - l_0$-SVM	$l_2 - l_0$-GBSVM
Breast Cancer	**0.0082s**	0.1485s	0.0552s
Ionosphere	**0.0453s**	0.1006s	0.1398s
Heart	**0.0016s**	0.4746s	0.1365s
Credit Approval	0.0661s	0.1684s	**0.0472s**
Statlog (Heart)	**0.0022s**	0.2988s	0.0305s
Parkinson's	0.1575s	0.1954s	**0.0265s**
SPECT Heart	**0.0233s**	0.6481s	0.0913s

Table 4 shows the average training time results for the three models, the output results demonstrate that l_1-SVM has the shortest training time followed by the two other models for all datasets except credit approval and parkinson's in favor of $l_0 - l_2$-GBSVM ($0.0472s$ and $0.0265s$ respectively). For example for breast cancer, the average training time is $0.0082s$, for ionosphere, $0.0453s$, for heart dataset $0.0016s$ and $0.0233s$ for the SPET heart dataset. The advantage of the l_1-SVM algorithm in training time is due to the fact that $l_2 - l_0$-SVM and $l_2 - l_0$-GBSVM both solve an iterative algorithm, which is the difference of convex functions algorithm, to make classification and specifically for the $l_2 - l_0$-GBSVM model that necessities a prestep of generation of granular balls before going through training the model; 2-means clustering, calculation of the radius and the center, assigning a label to each generated ball, ... , and then using these information as input for the model. Overall, in contrast to the two other models that use the whole data points as input, and despite the use of a small number of data as input (the centers of the balls), $l_2 - l_0$-GBSVM model guarantees a good compromise between the number of selected features and the average accuracy and all in a short amount of time.

5. Conclusion

In this paper, we presented a new embedded feature selection approach that combines Granular ball support vector machine variant with the double regularizaion approach combining l_2 and l_0 norms to attain two simultanious purposes; a good classfification and a good feature selection, l_2 norm is responsible for the good classification outputs, while l_0-"norm" is dedicated for feature selection. By adding the approximation of the l_0-"norm (concave approximation) to the problem winds it up being non convex. Despite the non convexity nature of the problem formulated, it could be written in the form of a difference of convex functions and solved using the difference of convex functions algoritm. The choice of the models' parameters was conducted using a genetic algorithm and the results were obtained on some benchmark datasets. The overall output shows that our approach presented in this paper could attain good results and have a favorable trade-off between accuracy, feature selection and training time, despite the fact that it uses a very small sized sample (the centers of granular balls). Several promising directions for future research emerge from this work, first there are several l_0 approximations that could be used instead of the approximation presented in this paper, other embedded feature selection approaches could be used

instead of the combination of l_0 and l_2. The granular ball generation algorithm can be replaced by other alternatives. The problem can be solved by other approaches, other than DC programming, besides the possibility to explore the feature selection for multi-class GBSVM. $l_2 - l_0$-GBSVM has many prameters to be tuned, therefore the choice of their values strongly affects the classifier, hence a much stronger way to select parameters would be an important investigation point. In future work, we will explore these directions, and initiatives will be performed on more datasets, from synthetic to real datasets, these being from different repositories, and with higher sized datasets. In addiion, different statistical analytics will be done to assert the efficiency of this method.

References

1. V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, (1998).
2. A. El Ouissari, K. El Moutaouakil, *Density based fuzzy support vector machine: application to diabetes dataset*, *Mathematical Modeling and Computing*, Vol. 8, No. 4, 747–760 (2021).
3. S. Abdualkarim, M. Kasihmuddin, M. Marsani, *Enhancing flood forecasting accuracy through improved SVM and ANFIS techniques*, *Mathematical Modeling and Computing*, Vol. 12, No. 2, 447–460 (2025).
4. V. Someetheram, M. F. Marsani, M. S. M. Kasihmuddin, N. E. Zamri, *Hybrid least squares support vector machine for water level forecasting*, *Mathematical Modeling and Computing*, Vol. 8, No. 4, 761–773 (2021).
5. F. Bazikar, S. Ketabchi, H. Moosaei, *DC Programming and DCA for Parametric Margin ν-Support Vector Machine*, *Applied Intelligence*, Vol. 50, No. 6, 1763–1774 (2020).
6. B. Richhariya, M. Tanveer, A. H. Rashid, Alzheimer's Disease Neuroimaging Initiative, *Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE)*, *Biomedical Signal Processing and Control*, Vol. 59, Article 101903 (2020).
7. X. Yang, Z. Hua, L. Zhang, X. Fan, F. Zhang, Q. Ye, et al., *Preferred vector machine for forest fire detection*, *Pattern Recognition*, Vol. 143, Article 109722 (2023).
8. I. Guyon, A. Elisseeff, *An Introduction to Variable and Feature Selection*, J. Mach. Learn. Res. 3, 1157–1182, (2003).
9. G. H. John, R. Kohavi, K. Pfleger, *Irrelevant Features and the Subset Selection Problem*, In R. S. Michalski, G. Tecuci (Eds.), Proc. 11th Int. Conf. Mach. Learn., 121–129, Morgan Kaufmann, San Francisco, CA, (1994).
10. P. S. Bradley, O. L. Mangasarian, *Feature Selection via Concave Minimization and Support Vector Machines*, In J. Shavlik (Ed.), Proc. 15th Int. Conf. Mach. Learn., 82–90, Morgan Kaufmann, San Francisco, CA, (1998).
11. L. Hermes, J. M. Buhmann, *Feature Selection for Support Vector Machines*, In Proc. Int. Conf. Pattern Recognit. (ICPR'00), Vol. 2, 716–719, (2000).
12. R. Duda, P. Hart, D. Stork, *Pattern Classification*, John Wiley and Sons, New York, NY, 2nd edition, (2000).
13. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik, *Feature Selection for SVMs*, In T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Adv. Neural Inf. Process. Syst. 13, 668–674, MIT Press, Cambridge, MA, (2001).
14. I. Guyon, S. Gunn, M. Nikravesh, L. A. Zadeh, *Feature Extraction: Foundations and Applications*, Springer, Berlin, (2006).
15. S. Maldonado, R. Weber, J. Basak, *Kernel-Penalized SVM for Feature Selection*, Inf. Sci. 181(1), 115–128, (2011).
16. H. A. Le Thi, H. M. Le, V. V. Nguyehn, T. Pham Dinh, *A dc programming approach for feature selection in support vector machines learning*, *Journal of Advances in Data Analysis and Classification*, Vol. 2, 259–278 (2008).
17. J. Neumann, C. Schnorr, G. Steidl, *Combined SVM-Based Feature Selection and Classification*, Mach. Learn. 61(1-3), 129–150, (2005).
18. J. Lopez, S. Maldonado, M. Carrasco, *Double regularization methods for robust feature selection and SVM classification via DC programming*, *Inf. Sci.*, Vol. 429, 377–389 (2017).
19. S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, *Granular ball computing classifiers for efficient, scalable and robust learning*, *Inform. Sci.*, Vol. 483, 136–152 (2019).
20. S. Xia, G. Wang, X. Lian, X. Gao, *GBSVM: An efficient and robust support vector machine framework via granular-ball computing*, *J. Mach. Learn. Res.*, Vol. 25, No. 1, 1–30 (2024).
21. A. Quadir, M. Sajid, M. Tanveer, *Granular Ball Twin Support Vector Machine*, *IEEE Transactions on Neural Networks and Learning Systems*, (2024).
22. L. Zhao, W. Ding, D. Miao, G. Lang, *Granular-Balls based Fuzzy Twin Support Vector Machine for Classification*, arXiv preprint arXiv:2408.00699, (2024).
23. T. Pham Dinh, H. A. Le Thi, *A DC Optimization Algorithm for Solving the Trust-Region Subproblem*, SIAM J. Optim. 8(2), 476–505, (1998).

24. T. Pham Dinh, H. A. Le Thi, *Convex Analysis Approaches to DC Programming: Theory, Algorithms and Applications*, Acta Math. Vietnam. 22(1), 287–367, (1997).

25. J. Xie, L. Jiang, S. Xia, X. Xiang, G. Wang, *An adaptive density clustering approach with multi-granularity fusion*, *Inf. Fusion*, Vol. 106, Article 102273 (2024).

26. S. Xia, G. Wang, X. Gao, X. Peng, *GBSVM: Granular-ball support vector machine*, arXiv preprint arXiv:2210.03120 (2022).

27. B. Scholkopf, A. J. Smola, F. Bach, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT Press, Cambridge (2002).

28. V. Vapnik, A. Chervonenkis, *Theory of pattern recognition*, Nauka, Moscow (1974).

29. S. Xia, X. Dai, G. Wang, X. Gao, E. Giem, *An efficient and adaptive granular-ball generation method in classification problem*, *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 35, No. 4, 5319–5331 (2024).

30. H. Moosaei, M. Hladík, *Sparse solution of least-squares twin multi-class support vector machine using l0 and Lp-norm for classification and feature selection*, *Neural Networks*, Vol. 166, 471–486 (2023).

31. T. N. Pham, M. N. Dao, N. Amjady, R. Shah, *A proximal splitting algorithm for generalized DC programming with applications in signal recovery*, *European Journal of Operational Research*, Article in Press (2025).

32. V. T. Huong, D. T. Kim Huyen, N. D. Yen, *Generalized polyhedral DC optimization problems*, *Journal of Optimization Theory and Applications*, Article in Press (2025).

33. Y. Zhang, I. Yamada, *An inexact proximal linearized DC algorithm with provably terminating inner loop*, *Optimization*, Article in Press (2025).

34. F. Hashemi, S. Ketabchi, *Optimal correction of infeasible equations system as $Ax + B|x| = b$ using ℓ_p-norm regularization*, *Boletim da Sociedade Paranaense de Matemática*, Vol. 40, 1–16 (2022).

35. V. T. Pham, H. P. H. Luu, H. A. Le Thi, *A block coordinate DCA approach for large-scale kernel SVM*, In: *Lecture Notes in Computer Science*, Vol. 13822, 123–135 (2022).

36. H. A. Le Thi, M. C. Nguyen, *DCA based algorithms for feature selection in multi-class support vector machine*, *Ann. Oper. Res.*, Vol. 249, No. 1–2, 273–300 (2017).

37. H. A. Le Thi, X. T. Vo, T. Pham Dinh, *Feature selection for linear SVMs under uncertain data: Robust optimization based on difference of convex functions algorithms*, *Neural Networks*, Vol. 59, 36–50 (2014).

38. G. Li, L. Yang, Z. Wu, C. Wu, *D.C. programming for sparse proximal support vector machines*, *Information Sciences*, Vol. 547, 187–201 (2021).

39. T. Pham Dinh, E. B. Souad, *Algorithms for Solving a Class of Nonconvex Optimization Problems: Methods of Subgradients*, In J.-B. Hiriart-Urruty (Ed.), Fermat Days 85: Math. Optim., North-Holland Math. Stud. 129, 249–271, North-Holland, (1986).

40. H. A. Le Thi, T. P. Dinh, *DC Programming and DCA: Thirty Years of Developments*, *Mathematical Programming*, Vol. 169, No. 1, 5–68 (2018).

41. H. A. Le Thi, T. Pham Dinh, *The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems*, Ann. Oper. Res. 133(1), 23–46, (2005).

42. N. Aafar, A. El Hilali Alaoui, B. Ferrahi, *Exploring the applications of DC programming in support vector machine learning*, In: *Algebra, Analysis, Modelling and Optimization*, Trends in Mathematics (2024).

43. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, (1970).

44. T. Pham Dinh, H. A. Le Thi, *A D.C. Optimization Algorithm for Solving the Trust-Region Subproblem*, SIAM J. Optim. 8(2), 476-505, (1998).

45. T. Pham Dinh, S. Elbernoussi, *Duality in d.c. (difference of convex functions) optimization. Subgradient Methods*, In: *Trends in Mathematical Optimization*, Vol. 84 of Int. Series of Numer. Math., 277–293, Birkhäuser Verlag, Basel (1988).

46. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).

47. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989).

48. A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer (2015).

49. M. Lichman, *UCI machine learning repository*, University of California, School of Information and Computer Science, Irvine (2013).

50. J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, *Use of the zero-norm with linear models and kernel methods*, *J. Mach. Learn. Res.*, Vol. 3, 1439–1461 (2003).

Najoua Aafar,
Department Mathematics,
LaR2A, Faculty of Sciences, Abdelmalek Essaadi University,
Morocco.
E-mail address: najoua.aafar@etu.uae.ac.ma

and

Abdellatif El Ouissari,
Department Mathematics,
LaR2A, Faculty of Sciences, Abdelmalek Essaadi University,
Morocco.
E-mail address: a.elouissari@uae.ac.ma

and

Bouchaib Ferrahi,
Department Mathematics,
LaR2A, Faculty of Sciences, Abdelmalek Essaadi University,
Morocco.
E-mail address: bferrahi@uae.ac.ma