Journal of Computational and Applied Mathematics 477 (2026) 117142

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Integrating subdivision schemes into SVM for improved signal classification ☆

V. Bruni [a,*], F. Pelosi [b], D. Vitulano [a]

[a] Sapienza University of Rome, Department of Basic and Applied Sciences for Engineering, via Antonio Scarpa 16, Roma, 00161, Italy
[b] University of Siena - Department of Information Engineering and Mathematics, via Roma 56, Siena, 53100, Italy

ARTICLE INFO

Keywords:
Subdivision schemes
Tension parameter
Support Vector Machine
Signal classification
Wavelet Scattering Transform

ABSTRACT

The integration of advanced signal processing techniques into machine learning models has gained increasing attention due to its potential to improve model performance, particularly for classification tasks. Support Vector Machine (SVM) is widely recognized as a powerful tool for signal classification due to its robust mathematical foundation and effectiveness in handling high-dimensional data. Subdivision schemes, originally developed in computer graphics for geometric modeling, offer a novel and parametric approach to feature preprocessing by iteratively refining input data through an efficient computational procedure. This paper studies the impact of subdivision schemes on SVM performance in terms of class separability and provides insights into the relationship between feature transformation and SVM response. Specifically, it investigates the theoretical and empirical implications of applying subdivision schemes to input features in SVM-based classification. The conditions under which these schemes preserve or enhance class separability are analyzed, focusing on the tension parameter which governs both the smoothness properties of the limit curve and the subdivision rule at each iteration. An estimation method for the tension parameter from the training data is also provided. Experimental results, performed in the context of signal classification based on the wavelet scattering transform, demonstrate that the appropriate selection of the tension parameter of the scheme can significantly enhance class separability, highlighting that subdivision schemes are a promising tool for improving classification accuracy in machine learning workflows.

1. Introduction

Signal classification is a critical task in decision-making processes in various applications fields, including communications, medical diagnostics, and multimedia processing. In recent years, the rapid growth of machine learning applications have highlighted the crucial role of data preprocessing in the improvement of model performance. Preprocessing methods aim to transform input data into representations that are more suitable for learning tasks, addressing challenges such as noise, sparsity, nonlinearity, and dimensionality [1–6]. They include scaling, normalization, dimensionality reduction, and feature extraction. Representative examples are z-score normalization, Principal Component Analysis (PCA), embeddings, wavelet or graph-based representations [7–13]. One notable area of advancement is the use of kernel-based transformations, where input features are implicitly mapped into higher-dimensional spaces to make data linearly separable [14,15]. Additionally, feature augmentation strategies, such as polynomial expansions or Fourier feature mappings, are often used to enrich the feature space, enabling models to capture complex relationships

☆ This article is part of a Special issue entitled: 'NAMAS-24' published in Journal of Computational and Applied Mathematics.

* Corresponding author.
E-mail addresses: vittoria.bruni@sbai.uniroma1.it, bruni@iac.rm.cnr.it (V. Bruni).

https://doi.org/10.1016/j.cam.2025.117142
Received 13 March 2025; Received in revised form 12 September 2025
Available online 8 October 2025

that may be hidden in the original representation [16–18]. More recently, feature ranking and automated machine learning frameworks began to incorporate automated feature selection or preprocessing as a core component of the optimization process using, for example, greedy strategies for classifier-dependent feature selection (see [19] and references therein) or evolutionary strategies and reinforcement learning for effective dataset-dependent preprocessing [20]. Despite these advancements, preprocessing remains a delicate task. Over-transforming features can lead to information loss or overfitting, particularly in scenarios with limited data. Among the methods that transform input features to improve machine learning models, subdivision schemes [21] are emerging as a promising approach for improving feature quality since they integrate a flexible and easily manageable formal model with an efficient computational procedure. Specifically, they are designed to generate continuous or differentiable functions from discrete data points through a recursive refinement procedure, thus producing smoother or higher-resolution representations of the data. A key characteristic is the tension parameter, which controls the degree of smoothness in the resulting signal. This parameter significantly influences the geometry of the transformed feature space, directly affecting the separability of data points associated with different classes. In a recent study [22], the fractal properties of 4-point interpolatory subdivision schemes have been exploited to interpolate wavelet scattering coefficients [23] for signal classification purposes. Specifically, the formal relationship between the tension parameter and the fractal dimension of the limit curve of the scheme has been studied, providing empirical evidence of the benefits given by tension parameters reproducing the fractal properties of the scattering matrix of the input signals. However, a direct study of how the tension parameter affects class separability in the feature space was not undertaken.

This paper contributes to a more rigorous understanding of how the geometric properties induced by subdivision schemes, particularly those controlled by the tension parameter, interact with machine learning models and their ability to separate classes. To this aim, Support Vector Machine (SVM) classifier [24–26] is considered. The latter is a cornerstone in machine learning-based classification tasks, where maximizing the margin between classes in a feature space is critical for generalization. A detailed mathematical analysis is conducted to identify the conditions under which subdivision schemes preserve or enhance class separability. Exploiting their analytical formulation, the investigation primarily concentrates on determining the tension parameter that provides a good balance between the separation of the center of mass of each class and the within class dispersion. The outcomes of the study also include a systematic iteration-wise tuning process for the tension parameter, which ensures that the feature transformation aligns with the optimal decision boundaries of the SVM classifier. Experimental results demonstrate that variations in the tension parameter significantly impact the performance of SVM in classification tasks, and indicate that, by carefully tuning the tension parameter, one can improve classification accuracy even after a single iteration of the subdivision scheme. This paves the way for future research into automated and data-driven parameter tuning in non-stationary subdivision schemes [27–29], with the goal of refining features for better class separability and increased generalization capabilities over different datasets.

The remainder of the paper is organized as follows. Section 2 provides a background on SVM, subdivision schemes and wavelet scattering transform. Section 3 presents the theoretical analysis of subdivision schemes in the context of class separability. Section 4 describes the experimental setup and results, and discusses the implications and limitations of our findings. Finally, Section 5 draws the conclusions.

2. Mathematical background

This section provides a formal yet concise explanation of the essential concepts of the two main elements of the proposed model: Support Vector Machines [24] and the Subdivision Schemes (the reader may refer to [30–32] and references therein for a comprehensive treatment of univariate subdivision schemes). As the numerical simulations are within the context of signal classification using wavelet scattering transform, a brief overview of this transform is also given.

2.1. Support vector machine

Support Vector Machines (SVM) is a supervised machine learning algorithm primarily used for classification and regression tasks [24–26]. It aims to find the optimal hyperplane that separates data points of different classes in a feature space, as detailed in the following for the binary classification problem.

Given a labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the ith feature vector and $y_i \in \{-1, +1\}$ denotes the corresponding label, SVM finds a hyperplane that maximally separates the two classes. More formally, let the support vectors be the closest data points from each class, and let the margin γ be defined as the distance between the hyperplane and the support vectors; then, for linearly separable data, SVM solves the following optimization problem

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i, \tag{1}$$

where $\mathbf{w}^\top \mathbf{x} + b = 0$ is the hyperplane equation, with $\mathbf{w} \in \mathbb{R}^d$ the normal vector to the hyperplane, and $b \in \mathbb{R}$ the bias term, $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$ is the classification rule, and $\gamma = \frac{2}{\|\mathbf{w}\|}$ is the margin.

Hence, under the constraint that a point $(\mathbf{x}_i, y_i)$ is correctly classified if

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1, 2, \dots, n\},$$

SVM maximizes the margin γ, and the above optimization problem refers to the *hard-margin*. A larger margin typically promotes generalization capabilities of the model. In addition the following result holds [24,26].

Proposition 2.1. *If the training data is linearly separable, there exists a unique hyperplane* $(\mathbf{w}, b)$ *that maximizes the margin.*

To handle more real cases, the hard margin can be relaxed, as for example in the case of data that are not linearly separable, where a point $(\mathbf{x}_i, y_i)$ is considered correctly classified if

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall\, i,$$

where $\xi_i \geq 0$ acts as a relaxation variable and represents the degree to which the ith point violates the margin. In other words, ξ_i allows for some misclassifications. The optimization problem then refers to a *soft-margin* SVM and can be reformulated as it follows

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \quad \text{subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i, \tag{2}$$

where $C > 0$ is a penalty term that controls the trade-off between maximizing the margin and minimizing misclassification errors. A high value of C forces the model to have fewer misclassifications, leading to a smaller margin, whereas a smaller C allows for a larger margin but more misclassifications [24,26].

It is worth observing that in both hard and soft margin optimization problems, the number of parameters (i.e., the dimension of $\mathbf{w}$) grows linearly with the number of features. To define the learning problem in a way that does not scale with the number of features, the optimization problem is often solved in its dual form where the Lagrange multipliers $\alpha_i \geq 0, \quad i = 1, \ldots, n$ are introduced for each training data point. Specifically, the dual form of the soft-margin SVM is:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i^\top \mathbf{x}_j), \tag{3}$$
$$\text{subject to } \sum_{i=1}^{n}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall i.$$

where

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b\right)$$

is the decision function; and whose optimal solution is given by the Representer Theorem [33], i.e.

Proposition 2.2. *The optimal solution* $\mathbf{w}$ *for the SVM problem in Eq. (3) can be expressed as a linear combination of the training data points:*

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i,$$

where $\alpha_i \geq 0$ *are the Lagrange multipliers and the Karush–Kuhn–Tucker conditions are satisfied, i.e.:*

1. $\alpha_i \geq 0$,
2. $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$,
3. $\alpha_i[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i] = 0$.

Therefore, only the support vectors ($\alpha_i > 0$) contribute to the hyperplane. Moreover, the number of parameters increases with the number of examples in the training set. This formulation also enables the use of kernel functions to tackle with data that are not linearly separable in the original feature space. Kernel functions are used to implicitly map the data into a higher-dimensional space, where it may become linearly separable, and without the need to explicitly compute the coordinates in that space. This is known as the *kernel trick* [26,34]. The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ computes the dot product in the transformed space, i.e.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j),$$

where $\varphi(\cdot)$ is a mapping to the higher-dimensional space, and then the decision function changes as it follows:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right).$$

Common kernel functions include linear kernel, polynomial kernels, radial basis functions, sigmoidal functions.

SVM can be extended to handle multi-class classification problems using different strategies, whose benefits depend on the number of classes and the computational efficiency [26]. Two commonly adopted strategies are *One-vs-All* and *One-vs-One*. The former consists of training one SVM for each class, with the objective of distinguishing that class from all the others, and selecting the SVM with the highest output score. The latter consists of training a binary SVM for every pair of classes, and then applying majority voting to select the final class.

2.2. 4-Point subdivision schemes

A general subdivision scheme [30–32,35–37] defines a curve or a surface out of an initial control polygon or control mesh by recursively subdividing them according to some refining rules which approximate or interpolate the points generated at each

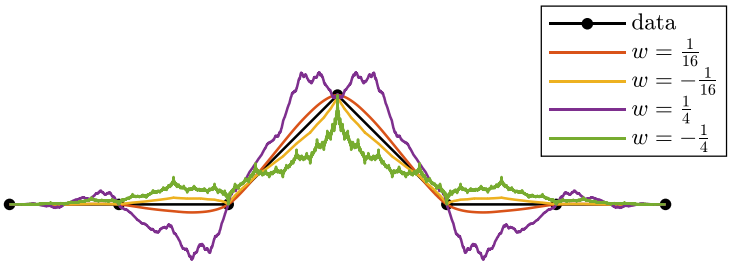

Fig. 1. Curves obtained after 10 steps on the initial points (black dots) for different values of the tension parameter: $w = \frac{1}{16}$ (orange), $w = -\frac{1}{16}$ (yellow), $w = \frac{1}{4}$ (purple) and $w = -\frac{1}{4}$ (green).

iteration. In general, approximating schemes give smoother curves or surfaces, while interpolatory schemes preserve the initial data points. This property makes interpolatory schemes able to preserve the original features when applied for increasing the number of discriminative features. Motivated by the results in [22], this section focuses on the 4-points interpolatory subdivision scheme (4SS) [21], while its use for classification purposes is further studied in the next section.

The 4SS is the first and most popular interpolatory scheme introduced in [38]. It is a binary scheme since it duplicates the number of control points, and thus it is in perfect matching with the upsampling/downsampling of a multiresolution setting. The recursive steps involve a 4-point rule, defined as it follows. Starting on initial control points $\mathbf{x}^{(0)} = \{\mathbf{x}_j^{(0)} \in \mathbb{R}^d\}_{j=-2}^{N+1}$, let $\mathbf{x}^{(k)} = \{\mathbf{x}_j^{(k)} \in \mathbb{R}^d\}_{j=-2}^{2^k N+1}$ be the set of control points at level k ($k \geq 0,\ k \in \mathbb{Z}$), then the set of control points at level $k+1$ $\{\mathbf{x}_j^{(k+1)} \in \mathbb{R}^d\}_{j=-2}^{2^{k+1}(N+1)}$ is defined by the following 2 rules:

$$\begin{cases} \mathbf{x}_{2j}^{(k+1)} = \mathbf{x}_j^{(k)}, \qquad -1 \leq j \leq 2^k N + 1, \\ \mathbf{x}_{2j+1}^{(k+1)} = \left(\frac{1}{2} + w\right)\left(\mathbf{x}_j^{(k)} + \mathbf{x}_{j+1}^{(k)}\right) - w\left(\mathbf{x}_{j-1}^{(k)} + \mathbf{x}_{j+2}^{(k)}\right), \qquad -1 \leq j \leq 2^k N, \end{cases} \tag{4}$$

where w is the tension (shape) parameter. By letting k tend to infinity, this process defines an infinite set of points in $\mathbb{R}^d$ whose smoothness properties depend on the values of the parameter w. In particular, it is proved that the 4-point interpolatory scheme is C^0-continuous for $-\frac{1}{4} < w < \frac{1}{4}$ and C^1-continuous for $0 < w < \frac{\sqrt{5}-1}{8}$ (see [21,38,39]), it generates the piecewise linear interpolant in the initial control polygon for the trivial value $w = 0$, whereas, for $w = \frac{1}{16}$, the limit curve has the best possible Hölder regularity (i.e., it is $C^{2-\varepsilon}$ for every $\varepsilon > 0$). In addition, the limit curve of the 4-point binary can be fractal for $-\frac{1}{2} < w < 0$ or $\frac{1}{4} \leq w < \frac{1}{2}$ [40], thus it allows to interpolate the coefficients of the feature matrix in a fractal context. Fig. 1 shows examples of curves obtained with the scheme (4) using different values of the parameter w: for $w = \frac{1}{16}$ (orange line) the curve is almost C^2 and it is only continuous for $w = -\frac{1}{16}$ (yellow line), whereas fractal behaviors are obtained for $w = \frac{1}{4}$ (purple line) and for $w = -\frac{1}{4}$ (green line). The initial points are marked with black bullets.

Subdivision can also be implemented in a level dependent way, that is by using different rules at different iterations, leading to the *non-stationary* subdivision schemes (NSS) while preserving computational efficiency. Also, the definition of convergence and regularity is not affected by the level dependence of the rules. Compared to the stationary case, NSS are most flexible and allow to reconstruct special types of curves including polynomial functions, conic sections and spiral curves [41,42]. Accordingly, in [42] a binary four-point interpolating non-stationary scheme generating limit curves of C^1-continuity has been defined. Fig. 2 compares the curves generated at different steps ($k = 1, 3, 5$) by the 4SS using two different tension parameters ($w = 0.3$ (top) and $w = \frac{1}{2}$ (bottom)) and by a non stationary four-point scheme where the tension parameters w_k are randomly sampled in $[\frac{1}{4}, \frac{1}{2}]$ (center).

2.3. Wavelet scattering transform

Wavelet Scattering Transform (WST) [23,43] is a non-linear time–frequency transform which is implemented by cascading convolutional, non-linear and pooling operators in an iterative manner. It can be graphically represented with a tree structure, as shown in Fig. 3, where the 0th layer is defined as the convolution product of the original input signal x with a regularizing scaling function ϕ [44]. In the successive layers the regularizing function ϕ is applied to the absolute value of the bands of a constant-Q factor wavelet decomposition. Its formal definition is in the following [23,43,45].

Definition 2.1. Let $x(t), \quad t \in \Omega_t \subset \mathbb{R}$ be a time-dependent signal, $\phi(t)$ a scaling function, $I_m \subset \mathbb{Z}$ a subset of indices, and $\psi_{\lambda_m}(t) := \{\psi_{m,k}(t)\}_{k \in I_m \subset \mathbb{Z}}$ the constant Q_m-factor wavelet filter bank generated by the mother wavelet $\psi(t)$, with $\psi_{m,k}(t) = \sqrt{\lambda_{m,k}}\psi(\lambda_{m,k}\, t)$, where $\lambda_{m,k} = 2^{\frac{k}{Q_m}}$ is the dilation parameter and $\{Q_m\}_{m=1,\dots,L}$ is a sequence of integer Q–factors representing the number of wavelets per

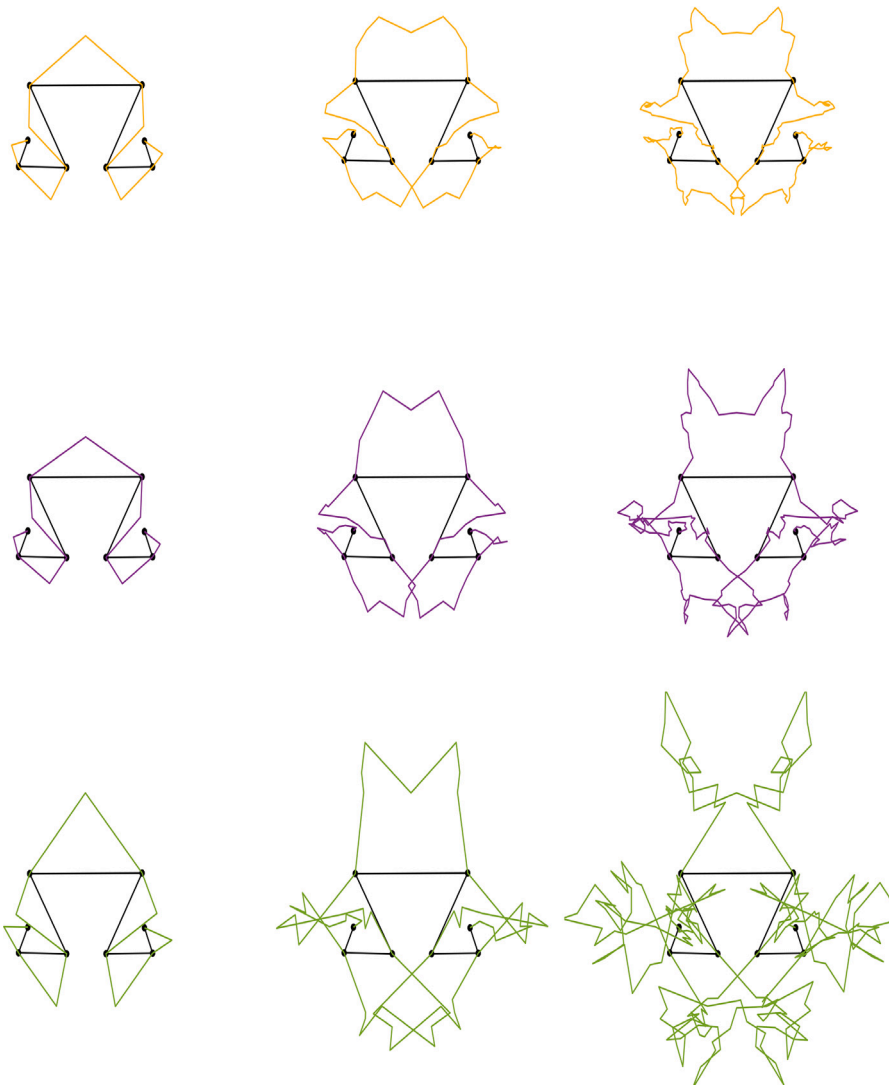

Fig. 2. Curves obtained after 1, 3 and 5 steps of 4SS *(left to right)*. *Top row:* stationary 4SS with $w = 0.3$ (yellow line). *Central row:* nonstationary 4SS where the values of w_k are randomly sampled in $[\frac{1}{4}, \frac{1}{2}]$ (purple line). *Bottom row:* stationary 4SS with $w = 0.5$ (green line).

octave. By setting $\lambda_m = \{\lambda_{m,k}\}_{k\in\mathbb{Z}}$, the *Wavelet Scattering Transform* (WST) $\mathbf{S}$ of x is the vector aggregation of the mth order scattering coefficients $\mathbf{S}_m x$ with $m = 0, \dots, L$, i.e.

$$\mathbf{S}x = \{\mathbf{S}_0 x, \mathbf{S}_1 x, \dots, \mathbf{S}_L x\}, \tag{5}$$

where $\mathbf{S}_0 x = x * \phi(t)$, and $\mathbf{S}_m x = \||\|x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \ldots| * \psi_{\lambda_m}| * \phi(t), \quad m = 1, \dots, L.$

Each mth order component $\mathbf{S}_m x$ can be arranged in matrix form whose dimension depends on the constant Q-factor wavelet transform. In particular, the temporal dimension (*number of columns*) depends on the critical sampling that the transform allows to apply, while the frequency dimension (*number of rows*) depends on the number of octaves Q_m. As a result, $\mathbf{S}x$ is the matrix obtained as the vertical concatenation of the mth order scattering matrices, i.e.

$$\mathbf{S}x = \begin{bmatrix} \mathbf{S}_0 x \\ \mathbf{S}_1 x \\ \vdots \\ \mathbf{S}_L x \end{bmatrix}. \tag{6}$$

WST is a non expansive transform that computes the deep spectrum of x and is stable to additive noise and time warping [23,46]. In addition, it preserves signal energy that, under appropriate assumptions on ψ, is concentrated in the first layers. This enables

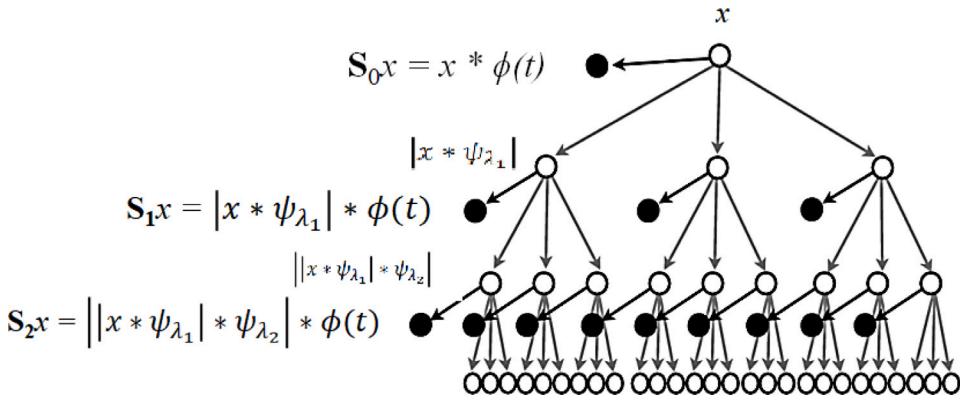

Fig. 3. Graphical representation of the tree-structure of the Wavelet Scattering Transform till the 2nd order component. Black dots represents the scattering subbands.

to reduce depth of the transform, making WST a powerful tool for signal classification when embedded in machine learning architectures [2,11,47].

3. The proposed data-driven method for feature pre-preprocessing in signal classification

SVM aims to find a decision boundary that separates the data into different classes while maximizing the margin between the support vectors [26,48]. The intuition behind SVM is that a larger margin leads to better generalization and, consequently, improved performance on unseen data. A larger margin reduces model complexity, leading to a better trade-off between bias and variance, and generally results in lower overfitting. The relationship between the centers of mass (or centroids) of classes and the dispersion within each class is crucial for understanding the geometry of class separability. In particular, it is expected that classes centroids are sufficiently separated while the dispersion within each class is sufficiently limited [26]. More precisely, let define

$$s = \{\mathbf{x}_i^{(s)}, i = 1, 2, \dots, n_s\}$$

a class of signals with n_s data points $\mathbf{x}_i^{(s)} \in \mathbb{R}^d$. The centroid $\mathbf{c}^{(s)}$ of the class s is defined as

$$\mathbf{c}^{(s)} = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{x}_i^{(s)}, \tag{7}$$

while the within-class dispersion is defined as [26], i.e.

$$\sigma^{(s)} = \frac{1}{n_s} \sum_{i=1}^{n_s} \|\mathbf{x}_i^{(s)} - \mathbf{c}^{(s)}\|^2. \tag{8}$$

The latter quantifies the variability of class data points around the centroid and it is commonly measured as the average squared distance of the points from their centroid. Hence, by denoting with

$$d(\mathbf{c}^{(s_1)}, \mathbf{c}^{(s_2)}) = \|\mathbf{c}^{(s_1)} - \mathbf{c}^{(s_2)}\|^2 \tag{9}$$

the squared distance between the centroids of the classes s_1 and s_2, a larger distance improves separability, while classes with higher dispersion exhibit greater overlap, reducing separability. With reference to SVM-based binary classification, a large margin is achievable when the distance between class centroids is significant compared to their dispersions, according to the *Fisher's Ratio* [26]

$$F_R^{(s_1,s_2)} = \frac{d(\mathbf{c}^{(s_1)}, \mathbf{c}^{(s_2)})}{\sigma^{(s_1)} + \sigma^{(s_2)}}, \tag{10}$$

which is a standard measure of separability in the linear case [26]. Nonlinear kernels accomplish this task by mapping the data into a higher-dimensional space where the class centroids are farther apart, and dispersions are reduced. Preprocessing methods aim to reduce within-class dispersion and enhance the separation between centroids. The next section focuses on the latter strategy by providing constraints on the tension parameter of the 4-point interpolatory subdivision scheme applied to the wavelet scattering feature matrix. To simplify the analysis, the investigation is restricted to classification problems involving two classes of signals, say s_1 and s_2. Class separation is evaluated through the Fisher ratio, as defined in Eq. (10), and its dependence on w is studied by considering a single subdivision step, transitioning from level k to level $k + 1$.

3.1. A numerical procedure for the selection of the tension parameter of stationary subdivision schemes

As mentioned in Section 2.3, the WST applied to a given signal $x(t)$ can be represented by a feature matrix where the rows represent the frequency bands while the columns the time samples. Since we are interested in interpolating time samples, in this section each signal $\mathbf{x}_i^{(s)}$ is associated with the wavelet scattering feature matrix $\mathbf{Sx}_i^{(s)} \in \mathbb{R}^{M\times N}$, which is rewritten as follows for convenience

$$\mathbf{Sx}_i^{(s)} = [\mathbf{x}_{i,1}^{(s)},\ \mathbf{x}_{i,2}^{(s)}, \dots, \mathbf{x}_{i,N}^{(s)}], \qquad \mathbf{x}_{i,j}^{(s)} \in \mathbb{R}^M,$$

where $\mathbf{x}_{i,j}^{(s)}$ represents the j column of the corresponding feature matrix, referred as *component*, of the signal $\mathbf{x}_i^{(s)}$. The subdivision scheme described in Section 2.2 is then applied to the columns of the feature matrix. Therefore, in this sequel the symbols have the following meaning:

- superscripts respectively indicate the class s (or the classes s_1 and s_2) and the iteration k of the subdivision scheme;
- subscripts indicate the element i of the class and the column index j of the feature matrix; for the centroid the first subscript is obviously missing.

As a result, the *components* of the signal $\mathbf{x}_i^{(s)}$ at the level $k+1$ of the subdivision scheme are denoted by $\mathbf{x}_{i,j}^{(s,k+1)}$ and are recursively defined as

$$\begin{cases} \mathbf{x}_{i,2j}^{(s,k+1)} = \mathbf{x}_{i,j}^{(s,k)}, & -1 \le j \le 2^k N + 1, \\ \mathbf{x}_{i,2j+1}^{(s,k+1)} = \left(\frac{1}{2} + w\right)\left(\mathbf{x}_{i,j}^{(s,k)} + \mathbf{x}_{i,j+1}^{(s,k)}\right) - w\left(\mathbf{x}_{i,j-1}^{(s,k)} + \mathbf{x}_{i,j+2}^{(s,k)}\right), & -1 \le j \le 2^k N, \end{cases} \tag{11}$$

for a proper choice of the tension (shape) parameter w.

The linearity of the scheme allows us to write the functional dependence of the entities in Eq. (10) on w, starting from class centroids at level $k+1$ of the subdivision scheme.

Proposition 3.1. *Let $\mathbf{c}^{(s,k)} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{x}_i^{(s,k)}$ be the centroid of the class s at level k of the four-point subdivision scheme, and let $\mathbf{x}_i^{(s,k+1)}$ be the feature matrix at level $k+1$ of the ith signal, as defined in Eq. (11), then*

$$\begin{cases} \mathbf{c}_{2j}^{(s,k+1)} = \mathbf{c}_j^{(s,k)}, & -1 \le j \le 2^k N + 1, \\ \mathbf{c}_{2j+1}^{(s,k+1)} = \left(\frac{1}{2} + w\right)\left(\mathbf{c}_j^{(s,k)} + \mathbf{c}_{j+1}^{(s,k)}\right) - w\left(\mathbf{c}_{j-1}^{(s,k)} + \mathbf{c}_{j+2}^{(s,k)}\right), & -1 \le j \le 2^k N. \end{cases} \tag{12}$$

The proof is in Appendix A. Thus, the centroid of feature vectors in a class at level $k+1$ is obtained by applying the subdivision scheme to the centroid of the same class at level k. This ensures consistency between the subdivision process applied to the signal and its corresponding centroid representation.

Eq. (12) enables to express the distance between $\mathbf{c}^{(s_1,k+1)}$ and $\mathbf{c}^{(s_2,k+1)}$ as function of w, and then to study the dependence of the *Fisher ratio* at level $k+1$, i.e.

$$F_R^{(s_1,s_2,k+1)} = \frac{d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)})}{\sigma^{(s_1,k+1)} + \sigma^{(s_2,k+1)}}, \tag{13}$$

on the tension parameter w, as shown in the following propositions.

Proposition 3.2. *Let consider $d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)})$ as in Eq. (9), and let set*

$$\boldsymbol{\Delta}^{(s_1,s_2,k+1)} := \mathbf{c}^{(s_1,k+1)} - \mathbf{c}^{(s_2,k+1)}$$

and

$$\mathbf{D}_j^{(s_1,s_2,k)} := \boldsymbol{\Delta}_j^{(s_1,s_2,k)} + \boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)} - \boldsymbol{\Delta}_{j-1}^{(s_1,s_2,k)} - \boldsymbol{\Delta}_{j+2}^{(s_1,s_2,k)}.$$

If $\overline{w}^{(s_1,s_2,k)} = argmin_w d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)})$, then

$$\overline{w}^{(s_1,s_2,k)} = -\frac{1}{2}\frac{\sum_j \left(\boldsymbol{\Delta}_j^{(s_1,s_2,k)} + \boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}\right)\mathbf{D}_j^{(s_1,s_2,k)}}{\sum_j \left(\mathbf{D}_j^{(s_1,s_2,k)}\right)^2}, \tag{14}$$

The proof is in Appendix B.

As observed in the proof, $d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)}) = \|\boldsymbol{\Delta}^{(s_1,s_2,k+1)}\|^2$ is a convex paraboloid with respect to w; therefore, larger values can be obtained for w far enough from $\overline{w}^{(s_1,s_2,k)}$.

A similar procedure allows us to study the denominator of the Fisher ratio.

Proposition 3.3. *Let* $\sigma^{(s,k+1)} = \frac{1}{n_s}\sum_{i=1}^{n_s} \|\mathbf{x}_i^{(s,k+1)} - \mathbf{c}^{(s,k+1)}\|^2$ *be the dispersion within the class* s *at a level* $k+1$ *of the four-point subdivision scheme, and let*

$$\widetilde{w}^{(s_1,s_2,k)} = argmin_w(\sigma^{(s_1,k+1)} + \sigma^{(s_2,k+1)}),$$

then

$$\widetilde{w}^{(s_1,s_2,k)} = -\frac{\alpha^{(s_1,k)}\widetilde{w}^{(s_1,k)} + \alpha^{(s_2,k)}\widetilde{w}^{(s_2,k)}}{\alpha^{(s_1,k)} + \alpha^{(s_2,k)}} \tag{15}$$

where $s = s_1, s_2$, $\widetilde{w}^{(s,k)} = argmin_w \sigma^{(s,k+1)}$, $\alpha^{(s,k)} := \sum_{i=1}^{n_s}\sum_j \left(\widetilde{\mathbf{D}}_{i,j}^{(s,k)}\right)^2$, *with* $\widetilde{\mathbf{D}}_{i,j}^{(s,k)} := \widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)} - \widetilde{\boldsymbol{\Delta}}_{i,j-1}^{(s,k)} - \widetilde{\boldsymbol{\Delta}}_{i,j+2}^{(s,k)}$ *and* $\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} := \mathbf{x}_{i,j}^{(s,k)} - \mathbf{c}_j^{(s,k)}$.

The proof is in Appendix C.

Collecting the results in Propositions 3.2 and 3.3, the following proposition provides an explicit expression for $F_R^{(s_1,s_2,k+1)}$ in (13), as a function of w. Hence, the tension parameter that maximizes $F_R^{(s_1,s_2,k+1)}$ can be easily defined.

Proposition 3.4. *Let* s_1 *and* s_2 *be two classes of signals having the same cardinality* ($n_{s_1} = n_{s_2} = n$). *Assume to perform one step of the four-point subdivision scheme from level* k, *and let define* $\boldsymbol{\Delta}^{(s_1,s_2,k)}$, $\mathbf{D}^{(s_1,s_2,k)}$, $\widetilde{\boldsymbol{\Delta}}^{(s,k)}$ *and* $\widetilde{\mathbf{D}}^{(s,k)}$ *as in Propositions 3.2 and 3.3, and* $F_R^{(s_1,s_2,k+1)}$ *as in Eq. (13), then*

$$F_R^{(s_1,s_2,k+1)} = \frac{\beta^{(s_1,s_2,k)}w^2 + \gamma^{(s_1,s_2,k)}w + \delta^{(s_1,s_2,k)}}{(\alpha^{(s_1,k)} + \alpha^{(s_2,k)})w^2 + (\xi_k^{(s_1)} + \xi^{(s_2,k)})w + \eta^{(s_1,k)} + \eta^{(s_2,k)}}, \tag{16}$$

where

$$\beta^{(s_1,s_2,k)} := \sum_j \left(\mathbf{D}_j^{(s_1,s_2,k)}\right)^2, \gamma^{(s_1,s_2,k)} = \sum_j (\boldsymbol{\Delta}_j^{(s_1,s_2,k)} + \boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)})\mathbf{D}_j^{(s_1,s_2,k)},$$

$$\delta^{(s_1,s_2,k)} := \sum_j (\boldsymbol{\Delta}_j^{(s_1,s_2,k)})^2 + \sum_j \left(\frac{\boldsymbol{\Delta}_j^{(s_1,s_2,k)} + \boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}}{2}\right)^2,$$

$$\alpha^{(s,k)} := \sum_{i=1}^{n}\sum_j \left(\widetilde{\mathbf{D}}_{i,j}^{(s,k)}\right)^2, \xi^{(s,k)} = \sum_{i=1}^{n_s}\sum_j \left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)}\right)\widetilde{\mathbf{D}}_{i,j}^{(s,k)},$$

$$\eta^{(s,k)} := \sum_{i=1}^{n_s}\sum_j \left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)}\right)^2 + \sum_{i=1}^{n_s}\sum_j \left(\frac{\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)}}{2}\right)^2.$$

The proof is in Appendix D.

The optimal tension parameter is then

$$w^{(s_1,s_2,k)} := argmax_w F_R^{(s_1,s_2,k+1)}. \tag{17}$$

Therefore, the estimate of the best tension parameter only depends on the initial feature matrices, i.e. the one composing the training set, and does not require the application of the subdivision scheme. Moreover, its evaluation is straightforward, making it computationally efficient and almost costless with respect to the successive learning process.

It is also worth observing that a tighter condition can be used for the estimation of the optimal parameter. It mainly rely on maximizing the quantity

$$R^{(s_1,s_2,k+1)} = d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)}) - \sigma^{(s_1,k+1)} - \sigma^{(s_2,k+1)}, \tag{18}$$

instead of the Fisher ratio. $R^{(s_1,s_2,k+1)}$ geometrically defines a sort of margin between the two classes and we are interested in finding the values of w for which the maximum is reached. In particular, the more positive $R^{(s_1,s_2,k+1)}$ the more separated the classes.

Proposition 3.5. *Let* s_1 *and* s_2 *be two classes of signals having the same cardinality* ($n_{s_1} = n_{s_2} = n$). *Assume to perform one step of the four-point subdivision scheme from level* k, *and let set*

$$\beta^{(s_1,s_2,k)} := \sum_j \left(\mathbf{D}_j^{(s_1,s_2,k)}\right)^2, \quad \text{and} \quad \alpha^{(s,k)} := \sum_{i=1}^{n}\sum_j \left(\widetilde{\mathbf{D}}_{i,j}^{(s,k)}\right)^2,$$

with $\mathbf{D}^{(s_1,s_2,k)}$ *and* $\widetilde{\mathbf{D}}^{(s,k)}$ *as defined in Propositions 3.2 and 3.3. If*

$$\beta^{(s_1,s_2,k)} > \alpha^{(s_1,k)} + \alpha^{(s_1,k)}.$$

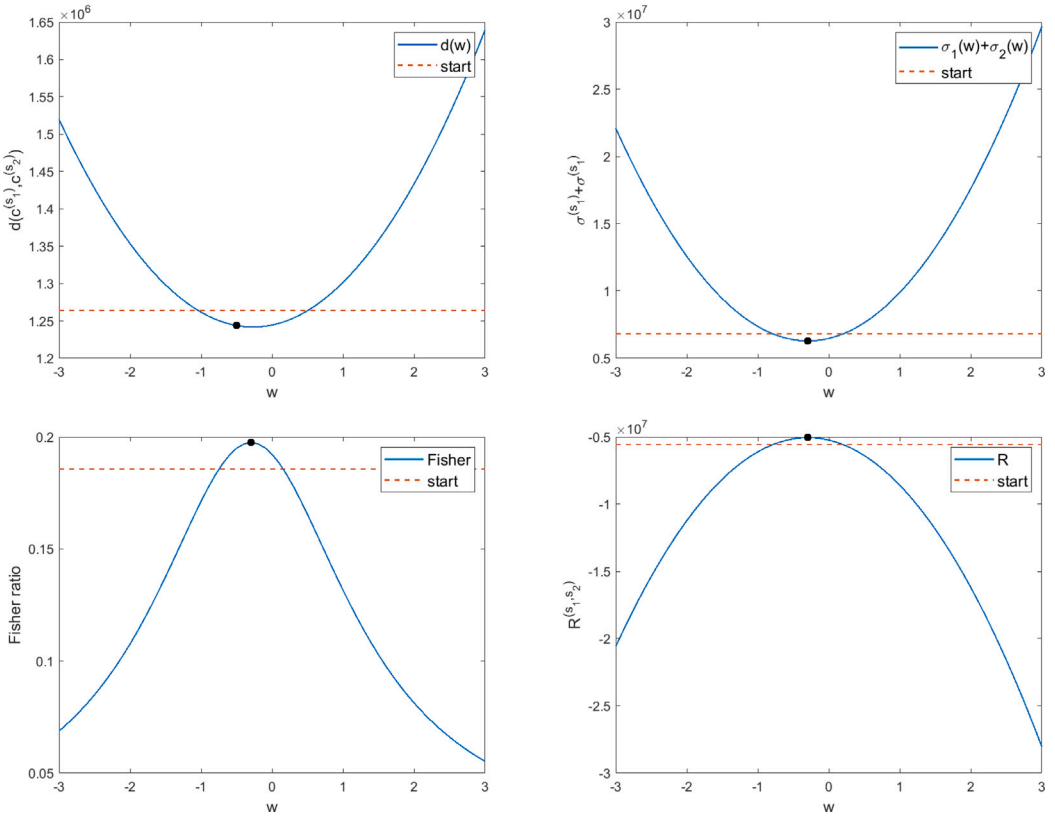

Fig. 4. Music Genres dataset. Plot of *(i)* the distance between the centroids versus the shape parameter of the 4SS of blues and jazz classes (*topleft*); *(ii)* the sum of the within-class dispersions (*topright*); *(iii)* the Fisher ratio (*bottomleft*); *(iv)* the separation index $R^{(s_1,s_2)}$ as defined in Eq. (18) (*bottomright*). The black dot indicates the estimated optimal tension parameter using Eq. (17), while the dashed line indicates the value of each index for the initial feature matrix. 60% of signals for each class have been considered.

then $w = \hat{w}^{(s_1,s_2,k)}$ *with*

$$\hat{w}^{(s_1,s_2,k)} = \frac{\beta^{(s_1,s_2,k)}\overline{w}^{(s_1,s_2,k)} - \alpha_k^{(s_1)}\widetilde{w}^{(s_1,k)} - \alpha^{(s_2,k)}\widetilde{w}^{(s_2,k)}}{\beta^{(s_1,s_2,k)} - \alpha^{(s_1,k)} - \alpha^{(s_2,k)}}. \tag{19}$$

maximizes $R^{(s_1,s_2,k+1)}$*, as defined in* (18).

The proof is in Appendix E.

4. Experimental results

This section aims at numerically evaluating the main theoretical findings of the previous section. To this aim, the WST matrix, as defined in Section 2.3, has been used as input feature for the SVM classifier. In fact, it is a critically sampled transform and then it is consistent with the considered operational scenario, where features are properly interpolated to enhance the discrimination capability of the classifier. All tests have been performed in Matlab environment. To provide quantitative results, the music genre recognition task has been considered and the well-known dataset GTZAN [49] has been used for both the training and testing phases. The dataset includes 10 genres, each represented by 100 clips having duration of 30 s and sampled at 22 050 Hz. Different percentages of clips have been considered for building the training and testing sets and will be specified for each experiment. Clips have been randomly selected from the whole initial dataset ensuring that all classes were equally represented. A two-layer WST ($L = 2$) has been employed in all tests and the hard critically sampled feature matrices, as in the standard implementation of the transform, have been considered as initial control points. Most of the results presented in this section refers to the couple of Q-factors (Q_1, Q_2), with $Q_1 = 8$ and $Q_2 = 1$. With these settings, WST provides the matrix $\mathbf{x}$ having dimension 334×32, where 334 are the WST bands, and 32 are the time samples of each band.

The first test refers to the binary classification problem where two genres s_1 and s_2 are considered. Fig. 4 refers to blues and jazz genres and shows the curve of the distances between the two centers of mass of the two classes, as defined in Eq. (9), as a function of the shape parameter w; similarly the within-class dispersion of the two classes $\sigma^{(s_1)}$ and $\sigma^{(s_2)}$, as defined in Eq. (8), the Fisher Ratio

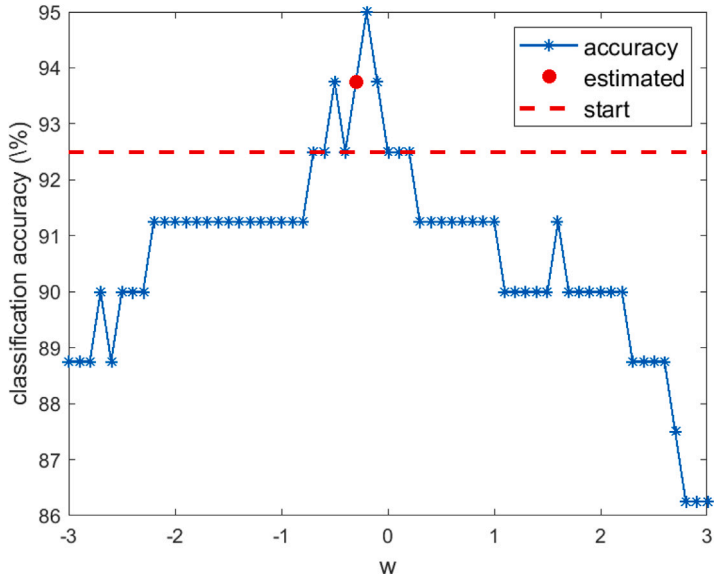

Fig. 5. Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy (measured in terms of correct assignments) versus the tension parameter w. The largest dot indicates the estimated w using the proposed approach. 60% of signals for each class are considered for the training set, 40% for the testing set.

$F_R^{(s_1,s_2,k+1)}$, as defined in Eq. (13), and the separability index $R^{(s_1,s_2,k+1)}$, as defined in Eq. (18), are considered. The plots have been obtained by applying the 4SS for different values of w and directly evaluating these quantities on the interpolated feature matrices. The evaluation refers to 60% of the available clips for the two classes. As can be observed, the curves show the expected convex behavior. In particular, the distance between the centroids is larger for shape parameters far away from $w = -0.30$. Conversely, the sum of the within-class dispersions attains its minimum at $w = -0.31$. The shape parameter that provides a tradeoff between the two different effects is the one realizing the maximum of the Fisher ratio, as estimated using Eq. (17), and it corresponds to $w = -0.3065$. In all plots, this value has been indicated with the dot point, while the starting value of the plotted quantities, i.e. the one evaluated on the initial feature matrix, is indicated by the dashed line. From the first two plots it is evident that the Fisher ratio grows especially because the overall within dispersion is reduced using the selected shape parameter. In particular, the gain provided by the reduced within dispersion is much larger than the penalty provided by closer centroids. It is also worth observing that the estimated shape parameter also maximizes the separability index $R^{(s_1,s_2,k+1)}$, that can be considered in place of the Fisher ratio for the selection of the best tension parameter. As a matter of fact, there exists a range of values of w for which the Fisher ratio is greater than the starting one, suggesting a possible improvement of the classification performance if tension parameters in this range are used for feature pre-processing. For instance, the trend of the Fisher ratio in the figure indicates the range $[-0.8, 0.1]$. Similar arguments are valid when commenting the separability index $R^{(s_1,s_2)}$ and the within-class dispersions.

The second test aims at evaluating the accuracy of the prediction using the proposed estimation method. To verify how the prediction aligns with the classification accuracy, which is measured in terms of percentage of correct class assignments, the linear kernel is used for SVM. Fig. 5 shows the classification results obtained using SVM where the input features are interpolated using one iteration of the subdivision scheme. The same figure indicates the estimated w using the proposed approach. As can be observed, the estimated w falls in the range of parameters providing the best classification performance. It is also worth highlighting that the range of tension parameters giving the best performance almost coincides with the one providing better Fisher ratios than the initial control feature matrix, as identified in the previous test. The same happens when the percentage of signals in the training set changes. Fig. 6 refers to 20% of signals in the training set, confirming the adaptivity of the proposed approach to the training set. Similar comments apply when using a different couple of Q factors (see Fig. 7), different initial input features (see Fig. 8) and different couples of genres (see Fig. 9).

Presented results confirm that interpolating input features using subdivision schemes can improve class separability and then classification performance. Moreover, the proposed estimation method can be quite successful in selecting good tension parameters. The gain in classification accuracy provided by the proposed method is about up to 2%. It is also worth observing that negative values of the tension parameter often provide the best classification results, in agreement with the results in [22] concerning the ability of the adopted subdivision scheme to reproduce some distinctive fractal properties.

It is also worth outlining that the proposed approach enables to avoid the selection of non linear kernels, providing a simple and straightforward procedure for the estimation of the optimal shape parameter to use for features transformation. An example is shown in Fig. 10, where classification accuracies obtained using different kernels are compared with the proposed approach. As can be observed, preprocessing the input feature vectors using the proposed procedure can reach performance comparable to a Radial Basis

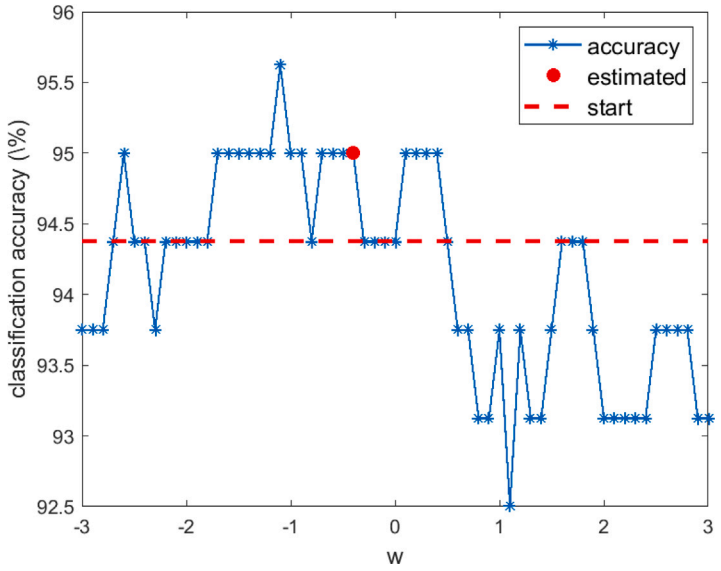

Fig. 6. Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy (measured in terms of correct assignments) versus the tension parameter w. The largest dot indicates the estimated w using the proposed approach. 20% of signals for each class are considered for the training set, 80% for the testing set.

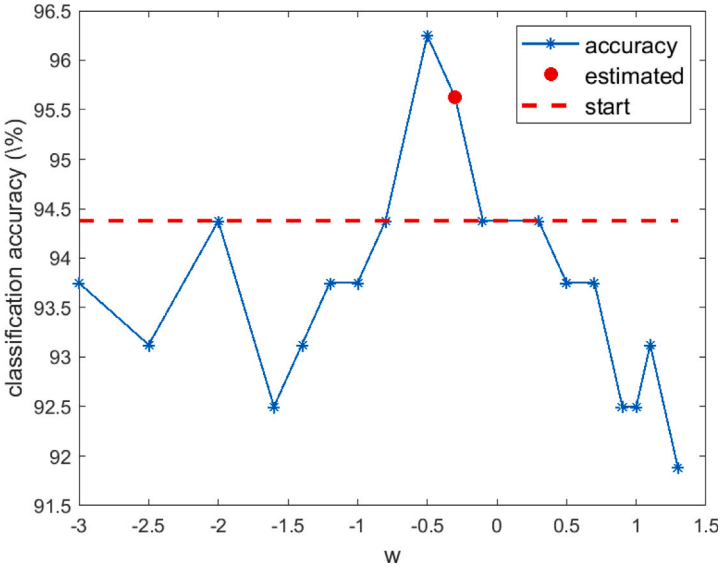

Fig. 7. Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy (measured in terms of correct assignments) versus the tension parameter w. The largest dot indicates the estimated w using the proposed approach. 20% of signals for each class are considered for the training set, 80% for the testing set. The couple $(3, 2)$ of Q-factors has been used in the WST.

Function (RBF) Kernel for solving specific classification tasks with small training sets, making it an effective alternative strategy to kernel selection. It is also worth observing that one iteration of the subdivision scheme improves the classification accuracy of the linear kernel. However, the use of just one iteration of the scheme cannot always be sufficient for class separation; more iterations can often provide higher separability. It is also worth highlighting that when using more iterations, the level-dependent parameter optimization cannot provide, as expected, the best result. A global optimization involving all the steps should be applied, as it is evident in Table 1. Likewise, feature interpolation alone cannot assure class separation in more complex contexts, thus requiring non linear kernels. However, similar behavior of the classification accuracy with respect to the tension parameter are expected. These observations motivate future studies concerning the generalization of the results to handle more iterations of the scheme, where the matrix formulation of subdivision schemes might be exploited, and the extension to non linear kernels.

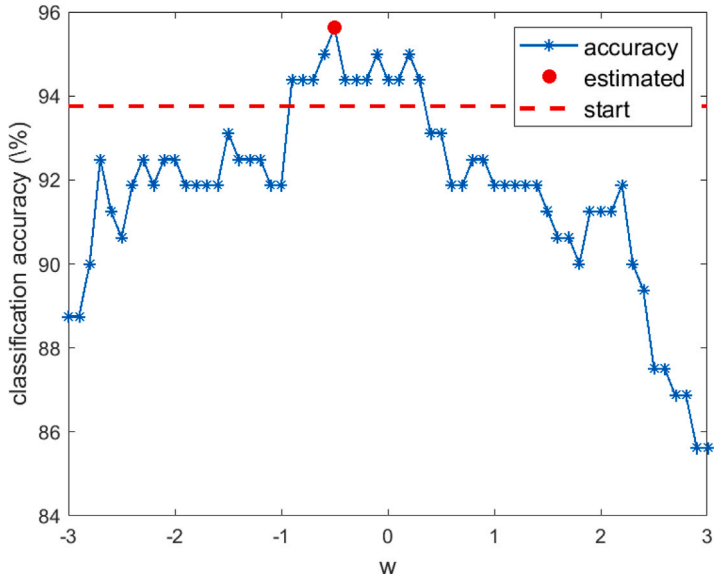

Fig. 8. Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy (measured in terms of correct assignments) versus the tension parameter w. The largest dot indicates the estimated w using the proposed approach. 20% of signals for each class are considered for the training set, 80% for the testing set. The couple $(8, 1)$ of Q-factors has been used in the WST. The input feature matrix has 16 samples along the temporal direction.

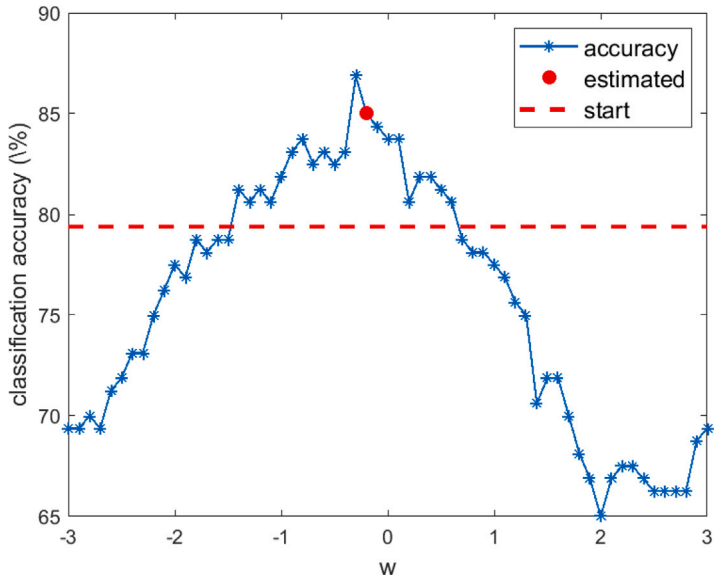

Fig. 9. Music Genres dataset: *blues* and *rock* genres. SVM classification accuracy (measured in terms of correct assignments) versus the tension parameter w. The largest dot indicates the estimated w using the proposed approach. 20% of signals for each class are considered for the training set, 80% for the testing set. The couple $(8, 1)$ of Q-factors has been used in the WST.

5. Conclusion

This paper presented a study on the integration of subdivision schemes in the workflow of machine-learning based classification. Specifically, subdivision schemes are used to properly interpolate the input features provided by a critically sampled transform with the aim of improving class separability and then increase the classification rates of the adopted classifier. The tension parameter of the scheme is identified as a critical factor that significantly shapes the performance of the classifier. Through a case study on signal classification using the wavelet scattering matrix as input to a linear SVM, it has been established that fine-tuning the tension parameter positively impacts classification accuracy. A method for automatically determining this parameter from the training

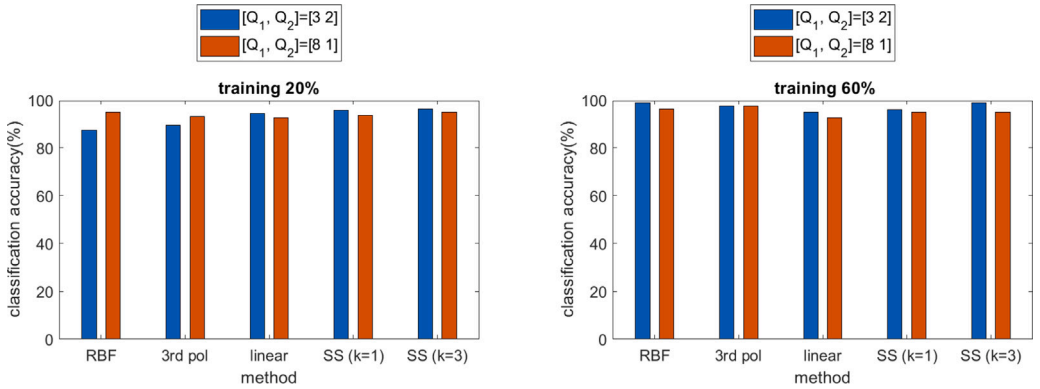

Fig. 10. Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy (measured in terms of correct assignments) using a radial basis function (RBF), third degree polynomial (3rd pol) and linear kernels compared to the proposed method, applied to the linear case in a level-dependent fashion, after one ($k = 1$) and three iterations ($k = 3$) of the four point subdivision scheme. Plots refer to 20% (*left*) and 60% of signals in the training set, and to two different couples of Q-factors for the WST: [3,2] (*blue bars*) and [8,1] (*red bars*).

Table 1
Music Genres dataset: *blues* and *jazz* genres. SVM classification accuracy for increasing number of iterations of the subdivision scheme and for some selected tension parameters. The reference value for the initial feature matrix is 94.3750%. As can be observed, using more than one iteration class separability is enhanced. The couple $(8, 1)$ of Q-factors has been used in the WST. 20% of signals for each class are considered for the training set, 80% for the testing set.

w	-1	0.1	0.7
$k = 1$	93.7500	94.3750	93.7500
$k = 2$	95.0000	96.2500	95.0000
$k = 3$	97.5000	96.8750	96.2500

set was proposed, and numerical experiments confirmed that the estimated values fall within the optimal range for improved performance. Importantly, the proposed approach introduces minimal computational overhead, as it allows for efficient estimation and implementation. These results open avenues for further research, including extending the estimation procedure to multiple iterations of the subdivision scheme, exploring non-stationary schemes, and adapting the approach for non-linear kernel classifiers. Future work could also investigate the application of this method to broader machine-learning tasks, reinforcing the potential of subdivision schemes as a valuable tool for feature enhancement in classification frameworks.

Acknowledgments

This work has been partially supported by PNRR-CN1 SPOKE 6 - Multiscale modeling and engineering applications B83C2200294 0006. This research has been accomplished within RITA (Research ITalian network on Approximation), UMI-TAA research group and the Italian national research group GNCS (INdAM).

Appendix A. Proof of Proposition 3.1

Proof. By definition $\mathbf{c}_j^{(s,k+1)} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{x}_{i,j}^{(s,k+1)}$. Using Eq. (11), for suitable values of j, we get

$$\mathbf{c}_{2j}^{(s,k+1)} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{x}_{i,2j}^{(s,k+1)} = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{x}_{i,j}^{(s,k)} = \mathbf{c}_j^{(s,k)},$$

and

$$\begin{aligned}\mathbf{c}_{2j+1}^{(s,k+1)} &= \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{x}_{i,2j+1}^{(s,k+1)}\\ &= \frac{1}{n_s}\sum_{i=1}^{n_s}\left(\frac{1}{2}+w\right)\left(\mathbf{x}_{i,j}^{(s,k)}+\mathbf{x}_{i,j+1}^{(s,k)}\right)-w\left(\mathbf{x}_{i,j-1}^{(s,k)}+\mathbf{x}_{i,j+2}^{(s,k)}\right)\\ &= \left(\frac{1}{2}+w\right)\left(\mathbf{c}_{j}^{(s,k)}+\mathbf{c}_{j+1}^{(s,k)}\right)-w\left(\mathbf{c}_{j-1}^{(s,k)}+\mathbf{c}_{j+2}^{(s,k)}\right). \quad \square\end{aligned}$$

Appendix B. Proof of Proposition 3.2

Proof. Let consider $\boldsymbol{\Delta}^{(s_1,s_2,k+1)} := \mathbf{c}^{(s_1,k+1)} - \mathbf{c}^{(s_2,k+1)}$ at a level $k+1$ component-wise. Using (12) we have

$$\boldsymbol{\Delta}_{2j}^{(s_1,s_2,k+1)} = \mathbf{c}_{2j}^{(s_1,k+1)} - \mathbf{c}_{2j}^{(s_2,k+1)} = \mathbf{c}_{j}^{(s_1,k)} - \mathbf{c}_{j}^{(s_2,k)} = \boldsymbol{\Delta}_{j}^{(s_1,s_2,k)},$$

and the linearity of the subdivision rules provide

$$\begin{cases}\boldsymbol{\Delta}_{2j}^{(s_1,s_2,k+1)} = \boldsymbol{\Delta}_{j}^{(s_1,s_2,k)},\\ \boldsymbol{\Delta}_{2j+1}^{(s_1,s_2,k+1)} = \left(\frac{1}{2}+w\right)\left(\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}\right)-w\left(\boldsymbol{\Delta}_{j-1}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+2}^{(s_1,s_2,k)}\right)\end{cases} \tag{B.1}$$

for $-1 \le j \le 2^k N + \ell$ where $\ell = 1, 0$ for even or odd rules. It is worth observing that $\boldsymbol{\Delta}_{2j}^{(s_1,s_2,k+1)}$ is independent of w while $\boldsymbol{\Delta}_{2j+1}^{(s_1,s_2,k+1)}$ can be rewritten as it follows

$$\boldsymbol{\Delta}_{2j+1}^{(s_1,s_2,k+1)} = \frac{\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}}{2} + w\mathbf{D}_{j}^{(s_1,s_2,k)},$$

with $\mathbf{D}_{j}^{(s_1,s_2,k)} = \left(\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}-\boldsymbol{\Delta}_{j-1}^{(s_1,s_2,k)}-\boldsymbol{\Delta}_{j+2}^{(s_1,s_2,k)}\right)$,

It turns out that

$$\|\boldsymbol{\Delta}^{(s_1,s_2,k+1)}\|^2 = \sum_j\left(|\boldsymbol{\Delta}_{2j}^{(s_1,s_2,k+1)}|^2 + |\boldsymbol{\Delta}_{2j+1}^{(s_1,s_2,k+1)}|^2\right)$$

is a second order polynomial with respect to w. Specifically,

$$\|\boldsymbol{\Delta}^{(s_1,s_2,k+1)}\|^2 = \beta^{(s_1,s_2,k)}w^2 + \gamma^{(s_1,s_2,k)}w + \delta^{(s_1,s_2,k)}, \tag{B.2}$$

where

$$\beta^{(s_1,s_2,k)} := \sum_j\left(\mathbf{D}_{j}^{(s_1,s_2,k)}\right)^2,$$

$$\gamma^{(s_1,s_2,k)} := \sum_j(\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)})\mathbf{D}_{j}^{(s_1,s_2,k)},$$

$$\delta^{(s_1,s_2,k)} := \sum_j\left(\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}\right)^2 + \sum_j\left(\frac{\boldsymbol{\Delta}_{j}^{(s_1,s_2,k)}+\boldsymbol{\Delta}_{j+1}^{(s_1,s_2,k)}}{2}\right)^2.$$

Since $\beta^{(s_1,s_2,k)} > 0$, then

$$\overline{w}^{(s_1,s_2,k)} = argmin_w\|\boldsymbol{\Delta}^{(s_1,s_2,k+1)}\|^2 = -\frac{\gamma^{(s_1,s_2,k)}}{2\beta^{(s_1,s_2,k)}} \tag{B.3}$$

and (14) immediately holds. $\square$

Appendix C. Proof of Proposition 3.3

Proof. By setting $\widetilde{\boldsymbol{\Delta}}_{i}^{(s)} := \mathbf{x}_{i}^{(s)} - \mathbf{c}^{(s)}$, the dispersion $\sigma^{(s,k+1)}$can be rewritten as

$$\sigma^{(s,k+1)} = \frac{1}{n_s}\sum_{i=1}^{n_s}\sum_j\left(\left|\widetilde{\boldsymbol{\Delta}}_{i,2j}^{(s,k+1)}\right|^2 + \left|\widetilde{\boldsymbol{\Delta}}_{i,2j+1}^{(s,k+1)}\right|^2\right), \tag{C.1}$$

where

$$\begin{cases}\widetilde{\boldsymbol{\Delta}}_{i,2j}^{(s,k+1)} = \mathbf{x}_{i,2j}^{(s,k+1)} - \mathbf{c}_{2j}^{(s,k+1)} = \mathbf{x}_{i,j}^{(s,k)} - \mathbf{c}_{j}^{(s,k)} = \widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)},\\ \widetilde{\boldsymbol{\Delta}}_{i,2j+1}^{(s,k+1)} = \left(\frac{1}{2}+w\right)\left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)}+\widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)}\right)-w\left(\widetilde{\boldsymbol{\Delta}}_{i,j-1}^{(s,k)}+\widetilde{\boldsymbol{\Delta}}_{i,j+2}^{(s,k)}\right),\end{cases} \tag{C.2}$$

for suitable $j = -1, 0, 1, \ldots, 2^k N + \ell, \ell = 1, 0$ for even or odd rule.

Hence, $\sigma^{(s,k+1)}$ is a second order polynomial with respect to w, i.e.

$$\sigma^{(s,k+1)} = \alpha^{(s,k)} w^2 + \xi^{(s,k)} w + \eta^{(s,k)}, \tag{C.3}$$

with

$$\alpha^{(s,k)} := \sum_{i=1}^{n_s} \sum_j \left(\widetilde{\mathbf{D}}_{i,j}^{(s,k)} \right)^2,$$

$$\xi^{(s,k)} := \sum_{i=1}^{n_s} \sum_j \left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)} \right) \widetilde{\mathbf{D}}_{i,j}^{(s,k)},$$

and

$$\eta^{(s,k)} := \sum_{i=1}^{n_s} \sum_j \left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} \right)^2 + \sum_{i=1}^{n_s} \sum_j \left(\frac{\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)}}{2} \right)^2$$

where $\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} := \mathbf{x}_{i,j}^{(s,k)} - \mathbf{c}_j^{(s,k)}$, and $\widetilde{\mathbf{D}}_{i,j}^{(s,k)} := \widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)} - \widetilde{\boldsymbol{\Delta}}_{i,j-1}^{(s,k)} - \widetilde{\boldsymbol{\Delta}}_{i,j+2}^{(s,k)}$.

Since $\alpha^{(s,k)} > 0$, then

$$\widetilde{w}^{(s,k)} = argmin_w \sigma^{(s,k)} = -\frac{\xi^{(s,k)}}{2\alpha^{(s,k)}}, \tag{C.4}$$

and then

$$\widetilde{w}^{(s,k)} = -\frac{1}{2} \frac{\sum_{i=1}^{n_s} \sum_j \left(\widetilde{\boldsymbol{\Delta}}_{i,j}^{(s,k)} + \widetilde{\boldsymbol{\Delta}}_{i,j+1}^{(s,k)} \right) \widetilde{\mathbf{D}}_{i,j}^{(s,k)}}{\sum_{i=1}^{n_s} \sum_j \left(\widetilde{\mathbf{D}}_{i,j}^{(s,k)} \right)^2} \tag{C.5}$$

Accordingly, assuming that $n_{s_1} = n_{s_2} = n$, the sum of the dispersion of the two classes s_1 and s_2, i.e.

$$\begin{aligned} \sigma^{(s_1,k+1)} + \sigma^{(s_2,k+1)} = & \frac{1}{n_{s_1}} \sum_{i=1}^{n_{s_1}} \left(\sum_j \left| \widetilde{\boldsymbol{\Delta}}_{i,2j}^{(s_1,k+1)} \right|^2 + \left| \widetilde{\boldsymbol{\Delta}}_{i,2j+1}^{(s_1,k+1)} \right|^2 \right) \\ & + \frac{1}{n_{s_2}} \sum_{i=1}^{n_{s_2}} \left(\sum_j \left| \widetilde{\boldsymbol{\Delta}}_{i,2j}^{(s_2,k+1)} \right|^2 + \left| \widetilde{\boldsymbol{\Delta}}_{i,2j+1}^{(s_2,k+1)} \right|^2 \right), \end{aligned}$$

is the second order polynomial

$$\sigma^{(s_1,k+1)} + \sigma^{(s_2,k+1)} = (\alpha^{(s_1,k)} + \alpha^{(s_2,k)}) w^2 + (\xi^{(s_1,k)} + \xi^{(s_2,k)}) w + \eta^{(s_1,k)} + \eta^{(s_2,k)} \tag{C.6}$$

with respect to w and then it attains its minimum at

$$\widetilde{w}^{(s_1,s_2,k)} = -\frac{\xi^{(s_1,k)} + \xi^{(s_2,k)}}{2(\alpha^{(s_1,k)} + \alpha^{(s_2,k)})}. \tag{C.7}$$

The proof follows by comparing the last equation with Eq. (C.4) . □

Appendix D. Proof of Proposition 3.4

Proof. By definition $d(\mathbf{c}^{(s_1,k+1)}, \mathbf{c}^{(s_2,k+1)}) = \|\boldsymbol{\Delta}^{(s_1,s_2,k+1)}\|^2$. Putting Eqs. (B.2) and (C.6) into Eq. (13), we get the proof. □

Appendix E. Proof of Proposition 3.5

Proof. From Eqs. (B.2) and (C.6) it follows that $R^{(s_1,s_2,k+1)}$ is a second order polynomial with respect to w, where the coefficient of the second order monomial is $\beta^{(s_1,s_2,k)} - (\alpha^{(s_1,k)} + \alpha^{(s_2,k)})$. Hence, if

$$\beta^{(s_1,s_2,k)} > (\alpha^{(s_1,k)} + \alpha^{(s_2,k)})$$

then

$$w^{(s_1,s_2,k)} = argmax_w R^{(s_1,s_2,k+1)} = -\frac{1}{2} \frac{\gamma^{(s_1,s_2,k)} - (\xi^{(s_1,k)} + \xi^{(s_2,k)})}{\beta^{(s_1,s_2,k)} - (\alpha^{(s_1,k)} + \alpha^{(s_2,k)})}.$$

Using Eqs. (C.7) and (B.3), we get the proof. □

Data availability

Data will be made available on request.

References

[1] A.A. Abdulsatar, V.V. Davydov, V.V. Yushkova, A.P. Glinushkin, V. Yu Rud, Age and gender recognition from speech signals, J. Phys.: Conf. Ser. 1410 (1) (2019).
[2] J. Anden, S. Mallat, Multiscale scattering for audio classification, in: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, 2011.
[3] V. Bruni, M.L. Cardinali, D. Vitulano, A short review on minimum description length: An application to dimension reduction in PCA, Entropy 24 (2) (2022).
[4] V. Bruni, G. Monteverde, D. Vitulano, An entropy-based speed up for hyperspectral data classification via CNN, in: Proceedings of the 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing, WHISPERS 2022, Rome, Italy, 2022.
[5] D. Klepl, M. Wu, F. He, Graph neural network-based EEG classification: A survey, IEEE Trans. Neural Syst. Rehabil. Eng. 32 (2024) 493–503.
[6] K. Zaman, M. Sah, C. Direkoglu, M. Unoki, A survey of audio classification using deep learning, IEEE Access 11 (2023) 106620-106649.
[7] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.
[8] M. Cox, T. Cox, Multidimensional scaling, in: Handbook of Data Visualization, in: Springer Handbooks Comp. Statistics, Springer, Berlin, Heidelberg, 2008.
[9] A.J. Ferreira, M.A.T. Figueiredo, Efficient feature selection filters for high-dimensional data, Pattern Recognit. Lett. 33 (13) (2012) 1794–1804.
[10] I. Jolliffe, J. Cadima, Principal component analysis: A review and recent developments, Philos. Trans. A 374 (2016).
[11] Z. Liu, G. Yao, Q. Zhang, J. Zhang, X. Zeng, Wavelet scattering transform for ECG beat classification, Comput. Math. Methods Med. (2020).
[12] V. Lostanlen, A. Cohen-Hadria, J. Pablo Bello, One or two frequencies? The scattering transform answers, in: Proceedings of the 28th European Signal Processing Conference, EUSIPCO, 2021.
[13] D. Theng, K.K. Bhoyar, Feature selection techniques for machine learning: A survey of more than two decades of research, Knowl. Inf. Syst. 66 (2024) 1575–1637.
[14] C. Campi, F. Marchetti, E. Perracchione, Learning via variably scaled kernels, Adv. Comput. Math. 47 (2021).
[15] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.
[16] W. Li, L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 1134–1148.
[17] J. Wang, M. Ye, Y. Kuang, R. Yang, W. Zhou, H. Li, Long-term feature extraction via frequency prediction for efficient reinforcement learning, IEEE Trans. Pattern Anal. Mach. Intell. 47 (4) (2025) 3094–3110.
[18] Y. Zhang, C. Shang, Q. Shen, Interpolating destin features for image classification, in: Proceedings of the 13th UK Workshop on Computational Intelligence, UKCI, Guildford, UK, 2013.
[19] F. Camattari, S. Guastavino, F. Marchetti, M. Piana, E. Perracchione, Classifier-dependent feature selection via greedy methods, Stat. Comput. 34 (2024).
[20] F. Hutter, L. Kotthoff, J. Vanschoren, Automated machine learning: Methods, systems, challenges, in: The Springer Series on Challenges in Machine Learning, Springer Nature, 2019.
[21] N. Dyn, Subdivision schemes in computer-aided geometric design, in: W. Light (Ed.), Advances in Numerical Analysis, vol. 2, 1992, pp. 36–104.
[22] V. Bruni, F. Pelosi, D. Vitulano, Fractal properties of 4-point interpolatory subdivision schemes and wavelet scattering transform for signal classification, Appl. Numer. Math. 208 (2025) 256–270.
[23] J. Anden, S. Mallat, Deep scattering spectrum, IEEE Trans. Signal Process. 62 (16) (2014) 4114–4128.
[24] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[25] R. Cristianini, E. Ricci, Support vector machines, in: M.Y. Kao (Ed.), Encyclopedia of Algorithms, Springer, Boston, MA, 2008.
[26] C. Gambella, B. Ghaddar, J. Naoum-Sawaya, Optimization problems for machine learning: A survey, European J. Oper. Res. 290 (3) (2021) 807–828.
[27] V. Bruni, M. Cotronei, F. Pitolli, A family of level-dependent biorthogonal wavelet filters for image compression, J. Comput. Appl. Math. 367 (2020).
[28] M. Charina, C. Conti, N. Guglielmi, V. Protasov, Limits of level and parameter dependent subdivision schemes: A matrix approach, Appl. Math. Comput. 272 (2016) 20–27.
[29] C. Conti, N. Dyn, Non-stationary subdivision schemes: State of the art and perspectives, in: G.E. Fasshauer, M. Neamtu, L.L. Schumaker (Eds.), Proceedings in Mathematics & Statistics, in: Approximation Theory XVI. AT 2019, vol. 336, Springer, 2021.
[30] M. Sabin, Recent progress in subdivision: a survey, in: N.A. Dodgson, M.S. Floater, M.A. Sabin (Eds.), Advances in Multiresolution for Geometric Modelling, in: Mathematics and Visualization, Springer, Berlin, Heidelberg, 2005.
[31] M. Sabin, Analysis and Design of Univariate Subdivision Schemes, Springer, 2010.
[32] J. Warren, H. Weimer, Subdivision Methods for Geometric Design: A Constructive Approach, Morgan Kaufmann Publishers Inc., San Francisco, 2001.
[33] B. Scholkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: Computational Learning Theory, in: Lecture Notes in Computer Science, vol. 2111, 2001, pp. 416–426.
[34] R. Herbrich, Learning Kernel Classifiers: Theory and Algorithms, MIT Press, 2001.
[35] E. Catmull, J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, Comput. Aided Des. 10 (6) (1978) 350–355.
[36] G. Chaikin, An algorithm for high speed curve generation, Comput. Graph. Image Process. 3 (1974) 346–349.
[37] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, Comput. Aided Des. 10 (6) (1978) 356–360.
[38] N. Dyn, D. Levin, J.A. Gregory, A 4-point interpolatory subdivision scheme for curve design, Comput. Aided Geom. Design 4 (4) (1987) 257–268.
[39] N. Dyn, J.A. Gregory, D. Levin, A butterfly subdivision scheme for surface interpolation with tension control, ACM Trans. Graph. 9 (2) (1990) 160–169.
[40] H. Zheng, Z. Ye, Y. Lei, X. Liu, Fractal properties of interpolatory subdivision schemes and their application in fractal generation, Chaos Solitons Fractals 32 (2007) 113–123.
[41] M.K. Jena, P. Shunmugaraj, P.C. Das, A non-stationary subdivision scheme for curve interpolation, ANZIAM J. 44 (2003) E216–E235.
[42] C. Beccari, G. Casciola, L. Romani, A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics, Comput. Aided Geom. Design 24 (1) (2007) 1–9.
[43] J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1872–1886.
[44] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 2008.
[45] J. Anden, V. Lostanlen, S. Mallat, Joint time–frequency scattering, IEEE Trans. Signal Process. 67 (14) (2019) 3704–3718.
[46] S. Mallat, Group invariant scattering, Commun. Pure Appl. Math. 65 (10) (2012) 1331–1398.
[47] V. Bruni, M.L. Cardinali, D. Vitulano, An MDL-based wavelet scattering features selection for signal classification, Axioms 11 (8) (2022).
[48] C. Aggarwal, Neural networks and deep learning: A textbook, 2019.
[49] G. Tzanetakis, P. Cook, Music genre classification of audio signals, IEEE Trans. Speech Audio Process. 10 (5) (2002) 293–302.