Title

Explainable and Class-Revealing Signal Feature Extraction via Scattering Transform and Constrained Zeroth-Order Optimization

Journal

IEEE Workshop on Statistical Signal Processing Proceedings, 00

ISSN

2373-0803

Authors

Saito, Naoki

Weber, David

Publication Date

2025-06-11

DOI

10.1109/ssp64130.2025.11073432

Explainable and Class-Revealing Signal Feature Extraction via Scattering Transform and Constrained Zeroth-Order Optimization

Naoki Saito
Department of Mathematics
University of California
Davis, CA 95616 USA
nsaito@ucdavis.edu

David Weber
The Delphi Group
Carnegie Mellon University
Pittsburgh, PA 15213 USA
dsweber2@protonmail.com

***Abstract*—We propose a new method to extract discriminant and explainable features from a particular machine learning model, i.e., a combination of the scattering transform and the multiclass logistic regression. Although this model is well-known for its ability to learn various signal classes with high classification rate, it remains elusive to understand why it can generate such successful classification, mainly due to the nonlinearity of the scattering transform. In order to uncover the meaning of the scattering transform coefficients selected by the multiclass logistic regression (with the Lasso penalty), we adopt zeroth-order optimization algorithms to search an input pattern that maximizes the class probability of a class of interest given the learned model. In order to do so, it turns out that imposing sparsity and smoothness of input patterns is important. We demonstrate the effectiveness of our proposed method using a couple of synthetic time-series classification problems.**

***Index Terms*—Nonlinear discriminant feature extraction, scattering transform, wavelets, zeroth-order optimization, sparsity and smoothness of signals**

I. Introduction

In signal and image classifications, the *Scattering Transform* (ST) [1], [2], which cascades wavelet transform convolutions with modulus nonlinearities (i.e., absolute values) and averaging operators, has emerged as an alternative to the popular Convolutional Neural Networks (CNNs)/Deep Learning (DL) [3], [4]. Since only two or three layers of the cascades in the ST are sufficient and since it uses the predefined wavelet convolution filters, it has a number of advantages over CNNs/DL for signal classification problems: 1) its training process is computationally faster; 2) it does not require a large number of training samples; 3) it automatically generates multiscale/multifrequency feature representations of input data in an *explicit* manner via wavelet filters; 4) the computed ST coefficients can be fed to any classifier of choice, e.g., Multiclass Logistic Regression (MLR), Support Vector Machine (SVM), k-Nearest Neighbors (kNN), etc.; and 5) it is more mathematically tenable. Yet, it still extracts robust and quasi-invariant features of input signals relative to certain types of deformations or perturbations (e.g., a small amount of shifts, warps, noise, etc.) so that it provides comparable classification performance as DL models. In fact, for a small number of training samples, it outperforms DL models [2], [5], [6].

Our earlier work has obtained excellent results using the ST and its variant in a number of signal classification problems including: underwater object classification from acoustic wavefields [7], [8]; texture image classification [9]; and music genre classification [10]. However, one critical thing remains to be understood: *Why does this combination of the features and the classifier work so well? What are the features in the original time domain that contributed to these excellent classification results?* Because the ST is a nonlinear transform like DL, even if we can identify which ST coefficients significantly contributed to the correct classification, the true meaning of such coefficients in the original time domain has been elusive: the only explicitly available information is the so-called *path* information, i.e., a set of indices of the wavelet filters (e.g., passband info) used in all the previous layers and the current layer in order to generate that particular ST coefficient.

Hence, our aim here is to *extract explainable or class-revealing features using the logistic regression coefficients learned on the ST coefficients of given training samples.* More specifically, we plan to *estimate signals that maximize the class probability using their ST coefficients and the trained MLR classifier for a signal class of interest.* If we are successful, then we can deepen our understanding of the features in the original time domain that are responsible for discriminating one class from the other. Such understanding and insight may uncover the secret of physical nature of a given problem, e.g., detection and classification of underwater objects via scattered acoustic wavefields; see [7], [8] for the background information and our previous effort. This is quite important since it may provide us with an opportunity to *design new transmitter signal patterns and/or sensors that specifically focus on discriminating a particular class of objects.*

II. Our Proposed Method

To be concrete, let us consider a typical signal classification problem. Let $\{\boldsymbol{x}_i\}_{i=1:N}$ be available training signals each of which has d time samples. Each $\boldsymbol{x}_i \in \mathbb{R}^d$ has a class label, $k \in \{1,\ldots,K\}$, i.e., $\boldsymbol{x}_i \in X_k$, where $X_k \subset \mathbb{R}^d$ is a space of class k signals. Our proposed procedure is the following.

Step 1: Apply the ST to the training samples;

Step 2: Train the GLMNet classifier (= the MLR classifier with the Lasso penalty [11]), which can efficiently select a small number of the ST coefficients as key features; let $(\alpha_k, \boldsymbol{\beta}_k)$, $k = 1:K$ be the resulting intercepts and regression coefficient vectors;

Step 3: Find an input pattern $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ for class k that minimizes the following criterion:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{p_k(\boldsymbol{x})} + \mu\|\boldsymbol{x}\|_1 + \nu\|\nabla\boldsymbol{x}\|_2, \tag{1}$$

where $p_k(\boldsymbol{x}) := \exp(\alpha_k + \boldsymbol{\beta}_k \cdot \mathscr{S}[\boldsymbol{x}]) / \sum_{j=1}^K \exp(\alpha_j + \boldsymbol{\beta}_j \cdot \mathscr{S}[\boldsymbol{x}])$ is the probability of a signal $\boldsymbol{x}$ belonging to class k (according to the trained GLMNet classifier); $\mathscr{S}[\boldsymbol{x}]$ is the ST coefficient vector computed from $\boldsymbol{x}$; and $\mu > 0$, $\nu > 0$ are the Lagrange multiplier parameters to be adjusted. Note that the minimization of the first term $1/p_k(\boldsymbol{x})$ is clearly equivalent to the maximization of $p_k(\boldsymbol{x})$ while the second and third terms promote *sparsity* and *smoothness* of $\boldsymbol{x}$, respectively.

We now describe the ST a bit more precisely. Since we only consider the second-layer ST coefficients, we define it for a given input 1D signal $\boldsymbol{x} \in \mathbb{R}^d$ as follows:

$$\mathscr{S}[\boldsymbol{x}] := \left\{\rho\left(\rho\left(\boldsymbol{x} \circledast_{r_1} \boldsymbol{\psi}^{(1)}_{\lambda_1}\right) \circledast_{r_2} \boldsymbol{\psi}^{(2)}_{\lambda_2}\right) \circledast_{r_a} \boldsymbol{\phi}_J\right\}_{\lambda_1\in\Lambda_1;\lambda_2\in\Lambda_2}, \tag{2}$$

where ρ is the elementwise modulus operator, $\rho : \boldsymbol{y} = (y_1,\ldots,y_d)^\mathsf{T} \in \mathbb{R}^d \mapsto |\boldsymbol{y}| := (|y_1|,\ldots,|y_d|)^\mathsf{T} \in \mathbb{R}^d_{\geq 0}$; $\circledast_r$ indicates a 1D circular convolution (typically done via multiplication in the Fourier domain) *followed by subsampling with rate* $r \geq 1$; $\boldsymbol{\psi}^{(l)}_{\lambda_l}$, is the mother wavelet filter with frequency-band index $\lambda_l \in \Lambda_l$ for the lth layer, $l = 1,2$; and finally $\boldsymbol{\phi}_J$ is the father wavelet (i.e., lowpass) filter at scale $J \in \mathbb{N}$. Note that we allow different subsampling rates for each layer, i.e., r_1, r_2 and for the final averaging process r_a. For the details, see [1], [6], [12] as well as [8, Chap. 3].

Step 3 needs a bit more explanation here. To solve the minimization problem Eq. (1), we use the so-called *zeroth-order (ZO) or derivative-free optimization* [13]–[15]. For our proposed method, it is better to use ZO optimization algorithms than the popular first-order (FO) optimization algorithms that require computing the gradient of the objective function to be minimized. This is because the gradient $\nabla\mathscr{S}[\boldsymbol{x}]$ is highly discontinuous, leading update steps to converge slowly, as shown in [8, Chap. 4]. In this paper, we use the popular ZO method called *Differential Evolution* (DE) [16]–[18]. We also note that we perform the updates of solution candidates in the frequency domain (via DCT) with the *pink* (a.k.a. $1/f$) noise as the initial randomized candidates instead of time domain updates starting from white noise. This is because the updates of solution candidates in the time domain are too local in time and many naturally-occurring signals follow the $1/f$ power spectra [19]; see also [8, Chap. 4]. The DE is certainly not the only ZO method to use for our problems; see more discussion of the choice of ZO methods in Section IV. Finally, we point out the importance of using the sparsity and the smoothness constraints in Eq. (1): 1) the landscape of Eq. (1) with $\mu = \nu = 0$ becomes too rough for the ZO optimization to converge; and 2) the sparsity and the smoothness provide us with easy and intuitive interpretation of signal features.

III. Examples

We now demonstrate the usefulness of our proposed method using two synthetic signal classification problems.

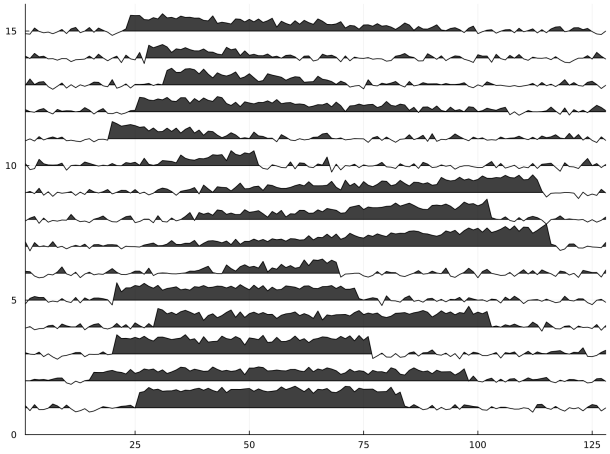
A. "Cylinder-Bell-Funnel" Signal Classification

We first tested our idea described above on a simple yet well-known time-series classification dataset, called the "Cylinder-Bell-Funnel" dataset that was introduced by one of us [20], [21] and has become quite well known in the time-series analysis literature; see, e.g., [22], [23]. In this example, we want to classify synthetic noisy signals with various shapes, amplitudes, lengths, and positions into three possible classes ($K = 3$). More precisely, sample signals of the three classes were generated by:

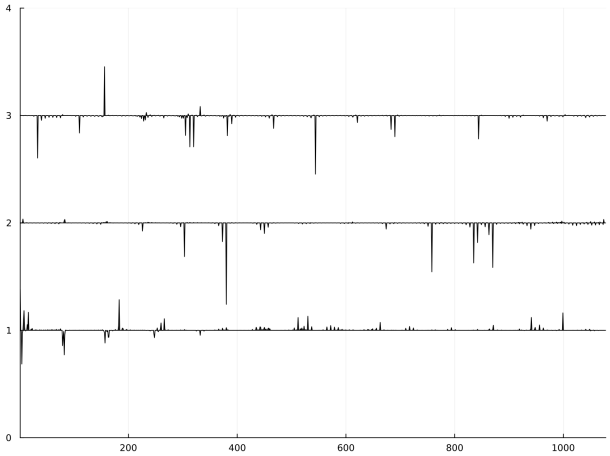
$$\begin{aligned}
c(i) &= (6+\eta)\cdot\chi_{[a,b]}(i) + \epsilon(i) && \text{for "cylinder"}\\
b(i) &= (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(i-a)/(b-a) + \epsilon(i) && \text{for "bell"}\\
f(i) &= (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(b-i)/(b-a) + \epsilon(i) && \text{for "funnel"}
\end{aligned}$$

where $i = 1,\ldots,128$, a is an integer-valued uniform random variable on the interval $[16,32]$, $b-a$ also obeys an integer-valued uniform distribution on $[32,96]$, η and $\epsilon(i)$ are the standard normal variates, and $\chi_{[a,b]}(i)$ is the characteristic (or indicator) function on the interval $[a,b]$. Figure 1a shows five sample waveforms from each class. If there is no noise, we can characterize the "cylinder" signals by two step edges and constant values around the center, the "bell" signals by one ramp and one step edge in this order with positive slopes around the center, and the "funnel" signals by one step edge and one ramp in this order with negative slopes around the center. In our numerical experiments below, we used the Julia programming language [24]; more specifically, for the ST computation and the MLR with the Lasso penalty, we used our own `ScatteringTransform.jl` [25] and the publicly-available `GLMNet.jl` [26]. Then, for the DE algorithm, we used the `BlackBoxOptim.jl` [27] package.

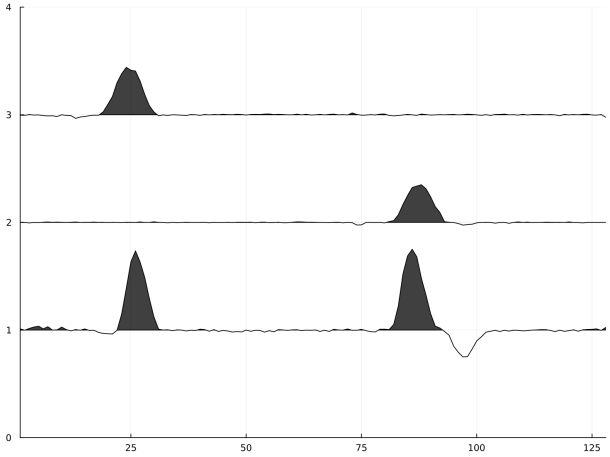
We generated 100 training signals per class and used the ST with the famous "Mexican-hat" wavelet function (i.e., the 2nd derivative of the Gaussian) as its base filter. Finally, we trained the GLMNet classifier on the 2nd layer ST coefficients $\{\mathscr{S}[\boldsymbol{x}_i] \in \mathbb{R}^{1078}_{\geq 0}\}_{i=1:300}$ of these training signals $\{\boldsymbol{x}_i \in \mathbb{R}^{128}\}_{i=1:300}$, to obtain the regression coefficient vector $\boldsymbol{\beta}_k \in \mathbb{R}^{1078}$ and the intercept, $\alpha_k \in \mathbb{R}^1$, $k = 1,2,3$,



(a) Five samples/class in the "CBF" classification problem: "Cylinder" (bottom 5); "Bell" (middle 5); "Funnel" (top 5)



(b) $\boldsymbol{\beta}$ coefficients: "Cylinder" (bottom); "Bell" (middle); "Funnel" (top)



(c) Solutions of Eq. (1): "Cylinder" (bottom); "Bell" (middle); "Funnel" (top)

Fig. 1: Extracted features via Eq. (1) reveal the decisive characteristics of each class in the "CBF" signal classification problem

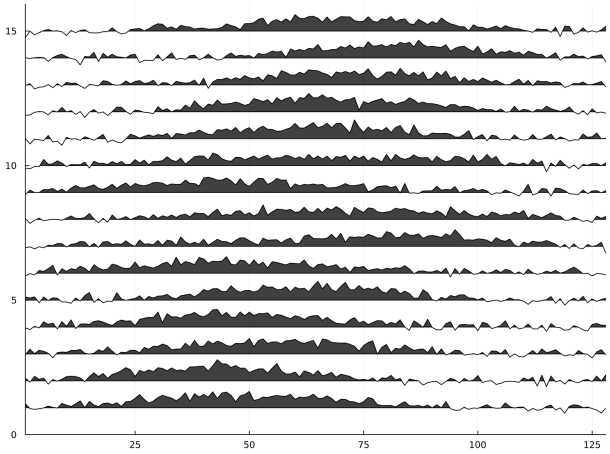
corresponding to cylinder, bell, and funnel classes. The subsampling rates $r_1 = r_2 = 1.5$ and $r_a = 8$ were used while the number of wavelet filters was set as $|\Lambda_1| = 14$ and $|\Lambda_2| = 11$. Hence, the final output size at the 2nd layer of the ST for each input signal of length 128 *per filter pair* (λ_1, λ_2) was $\lfloor 128/1.5/1.5/8 \rfloor = 7$, which led to $7 \times 14 \times 11 = 1078$ ST coefficients in total. The classification accuracy of this method was almost perfect ($\approx 99\%$) on newly-generated test dataset of size 3000. Figure 1b displays the learned $\boldsymbol{\beta}_k$ coefficient vectors that are quite sparse thanks to the Lasso penalty while Fig. 1c shows the solutions of Eq. (1) using the DE algorithm. It is amazing to see that *those estimated signals pinpoint the distinguishing features of those three classes*: the one for the bell class and that for the funnel class are measuring the local activities around the discontinuous regions where those classes abruptly ends (bell) or starts (funnel) while it checks *both* edge locations for the cylinder class. They are the *class-revealing* features and can be viewed as the *essence* of the prototype signals.

B. Triangular Waveform Classification

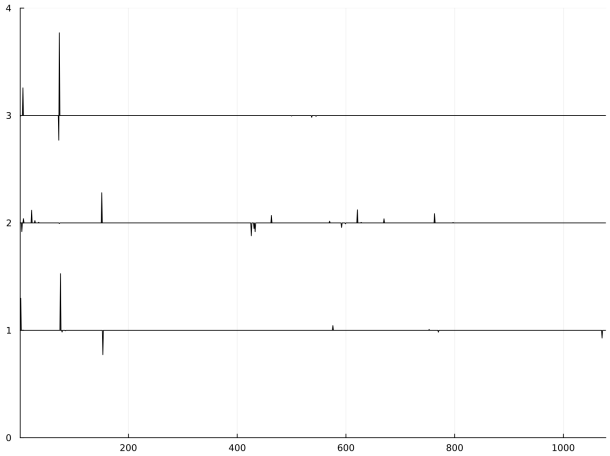
This is another well-known classification problem, first appeared in the classic book [28, Sec. 2.6.2]. The dimensionality (or length) of each signal was extended from the original 21 in [28] to 128 in order to fully utilize the power of the ST: 21 was just too short for cascades of wavelet convolutions. Three classes of signals were generated by:

$$\begin{aligned} x^{(1)}(i) &= u h_1(i) + (1-u) h_2(i) + \epsilon(i) \qquad \text{for Class 1,} \\ x^{(2)}(i) &= u h_1(i) + (1-u) h_3(i) + \epsilon(i) \qquad \text{for Class 2,} \\ x^{(3)}(i) &= u h_2(i) + (1-u) h_3(i) + \epsilon(i) \qquad \text{for Class 3,} \end{aligned}$$

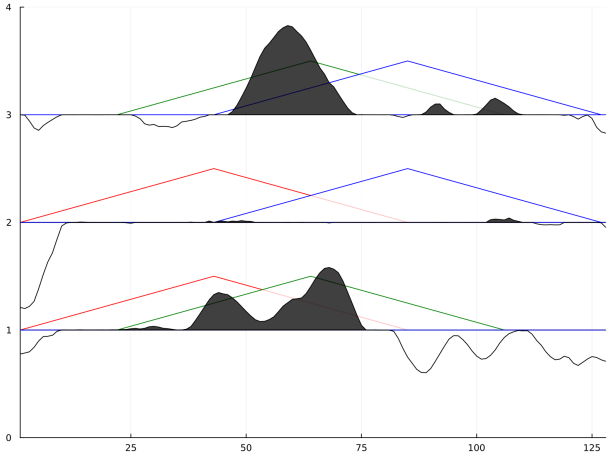
where $i = 1, \ldots, 128$, $h_1(i) = \max(6 - |i - 43|/7, 0)$, $h_2(i) = h_1(i-21)$, $h_3(i) = h_1(i-42)$, u is a uniform random variable on the interval $(0,1)$, and $\epsilon(i)$ are the standard normal variates. Simply speaking, each class consists of random convex linear combination of two triangles with additive white Gaussian noise. Notice that the noiseless version of each class forms an edge of a triangular manifold in $\mathbb{R}^{128}$ whose vertices are those three triangles $\boldsymbol{h}_k$, $k = 1,2,3$. Hence, the noisy versions are distributed within Gaussian balls around this triangular manifold. Figure 2a shows five sample waveforms from each class; Fig. 2b shows the sparse $\boldsymbol{\beta}_k$ coefficient vectors; and Fig. 2c displays the extracted feature/the best input pattern for each class; the red, green, and blue triangles in the background indicate the vectors $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, and $\boldsymbol{h}_3$, respectively. We used the same parameter settings as those of the "CBF" signal classification problem. The classification rate using the ST plus MLR was 88.9% using 1000 newly generated signals per class as the test dataset while each of the estimated features in Fig. 2c gave us essentially perfect classification rate of 99.99%. In other words, these estimated features essentially discovered the nature of class signal generation mechanism as in the "CBF" signal classification problem. The Class 1 and the Class 3 features in the bottom and top rows in Fig. 2c



(a) Five samples/class of the triangular waveform problem: Class 1 (bottom 5); Class 2 (middle 5); Class 3 (top 5)



(b) $\boldsymbol{\beta}$ coefficients: Class 1 (bottom); Class 2 (middle); Class 3 (top)



(c) Solutions of Eq. (1): Class 1 (bottom); Class 2 (middle); Class 3 (top)

Fig. 2: Extracted features via Eq. (1) reveal the significant characteristic of each class in the triangular signal classification problem

indicate the signal energy concentrates around the apices of $\{\boldsymbol{h}_1, \boldsymbol{h}_2\}$ and $\{\boldsymbol{h}_2, \boldsymbol{h}_3\}$, respectively while the Class 2 feature shown in the middle row of Fig. 2c does not have such energy concentration around $\boldsymbol{h}_2$, indicating that the signals of this class do not have the contribution from $\boldsymbol{h}_2$. This is quite interesting: if the Class 2 feature had energy concentration around $\{\boldsymbol{h}_1, \boldsymbol{h}_3\}$, it would have generated worse classification rate because it would confuse some Class 1 and Class 3 signals; hence our algorithm decided to use the "nonexistence" of the $\boldsymbol{h}_2$ component. This has given us a serendipity: in certain situations, it would be better to consider feature "suppressors" instead of feature extractors!

IV. Summary and Discussion

We have described our effort to understand nonlinear signal features computed by ST and the MLR classifier using the ZO optimization with constraints on sparsity and smoothness of the features in the original time domain. Our numerical results in Section III were quite promising.

We have used the DE algorithm as our ZO method to solve Eq. (1). Clearly, other methods may work well with even faster computational speed than the DE algorithm. Hence, it is important to evaluate various ZO algorithms that is robust and computationally efficient. There are two different strategies in ZO optimization: *single-particle* methods and *multi-particle* methods. The advantage of multi-particle methods [29]–[31] is its population-based adaptive and evolutionary solution-search capability while the disadvantage is their difficulty to derive theoretical guarantee of convergence. Hence, we will also investigate the single-particle methods, which are theoretically more tenable and can fully utilize the advanced FO algorithms once the gradient estimation is done at each iteration. In particular, we plan to investigate the so-called *ZO proximal methods* [32], [33] because these can handle the constraints consisting of a nonconvex and smooth function and a nonsmooth and convex function like our Eq. (1). Our preliminary numerical experiments using the `PRIMA.jl` [34] is quite promising: it generated similar results as those in Section III with an order of magnitude faster than the DE algorithm.

Furthermore, we will investigate: 1) automatic determination of the optimal values of μ and ν in Eq. (1); 2) other effective constraints beyond sparsity and smoothness; and 3) the influence of the wavelet filter parameters (e.g., type of wavelets, number of filters, their frequency overlaps, etc.).

Finally, we note that our optimization strategy Eq. (1) should work in principle with any other learning model as long as it outputs the class probability $p_k(\boldsymbol{x})$ for a given input signal $\boldsymbol{x} \in \mathbb{R}^d$, e.g., a DL model equipped with "softmax" output units [4, Sec. 6.2.2.3].

Acknowledgment

This research was partially supported by the US National Science Foundation grants DMS-1912747 and CCF-1934568 as well as the US Office of Naval Research grant N00014-20-1-2381.

REFERENCES

[1] S. Mallat, "Group invariant scattering," *Comm. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, 2012.

[2] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 35, no. 8, pp. 1872–1886, 2013.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[5] J. Bruna and S. Mallat, "Classification with scattering operators," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1561–1566.

[6] J. Andèn and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, 2014.

[7] N. Saito and D. S. Weber, "Underwater object classification using scattering transform of sonar signals," in *Wavelets and Sparsity XVII, Proc. SPIE 10394*, Y. M. Lu, D. Van De Ville, and M. Papadakis, Eds., 2017, Paper # 103940K.

[8] David S. Weber, *On Interpreting Sonar Waveforms via the Scattering Transform*, PhD dissertation, Appl. Math., Univ. California, Davis, Dec. 2021.

[9] Wai Ho Chak and Naoki Saito, "Monogenic wavelet scattering network for texture image classification," *JSIAM Letters*, vol. 15, pp. 21–24, 2023.

[10] Wai Ho Chak, Naoki Saito, and David Weber, "The scattering transform network with generalized Morse wavelets and its application to music genre classification," in *Proc. 2022 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Toyama, Japan, 2022, pp. 25–30.

[11] Trevor Hastie, Robert Tibshirani, and Martin Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, vol. 143 of *Monographs on Statistics and Applied Probability*, CRC Press, Boca Raton, FL, 2015.

[12] Mathieu Andreux et al., "Kymatio: Scattering transforms in python," *Journal of Machine Learning Research*, vol. 21, no. 60, pp. 1–6, 2020.

[13] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization. SIAM, 2009.

[14] Jeffrey Larson, Matt Menickelly, and Stefan M. Wild, "Derivative-free optimization methods," *Acta Numerica*, vol. 28, pp. 287–404, 2019.

[15] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.

[16] Mohamad Faiz Ahmad, Nor Ashidi Mat Isa, Wei Hong Lim, and Koon Meng Ang, "Differential evolution: A recent review based on state-of-the-art works," *Alexandria Eng. J.*, vol. 61, no. 5, pp. 3831–3872, 2022.

[17] Bilal, Millie Pant, Hira Zaheer, Laura-Hernandez Garcia, and Ajith Abraham, "Differential Evolution: A review of more than two decades of research," *Eng. Appl. Artif. Intell.*, vol. 90, pp. Article #103479, 2020.

[18] Karol R. Opara and Jarosaw Arabas, "Differential Evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, 2019.

[19] W. H. Press, "Flicker noises in astronomy and elsewhere," *Comments on Modern Physics, Part C - Comments on Astrophysics*, vol. 7, no. 4, pp. 103–119, 1978.

[20] N. Saito and R. R. Coifman, "Local discriminant bases and their applications," *J. Math. Imaging Vis.*, vol. 5, no. 4, pp. 337–358, 1995, Invited paper.

[21] N. Saito, "Local feature extraction and its applications using a library of bases," in *Topics in Analysis and Its Applications: Selected Theses*, R. Coifman, Ed., pp. 269–451. World Scientific Pub. Co., Singapore, 2000.

[22] Pierre Geurts, "Pattern extraction for time series classification," in *Principles of Data Mining and Knowledge Discovery (PKDD 2001)*, Luc De Raedt and Arno Siebes, Eds., 2001, vol. 2168 of *Lecture Notes in Computer Science*, pp. 115–127.

[23] Eamonn Keogh and Shruti Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, pp. 349–371, 2003.

[24] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[25] David Weber and Naoki Saito, "ScatteringTransform.jl," https://github.com/dsweber2/ScatteringTransform.jl, 2022–25.

[26] Julia Statistics, "GLMNet.jl," https://github.com/JuliaStats/GLMNet.jl, 2015–25.

[27] Robert Feldt, "BlackBoxOptim.jl," https://github.com/robertfeldt/BlackBoxOptim.jl, 2015–25.

[28] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman & Hall, Inc., New York, 1993, previously published by Wadsworth & Brooks/Cole in 1984.

[29] José A. Carrillo, Shi Jin, and Yuhua Zhu, "A consensus-based global optimization method for high dimensional machine learning problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 27, pp. # S5, 2021.

[30] Massimo Fornasier, Lorenzo Pareschi, Hui Huang, and Philippe Sünnen, "Consensus-based optimization on the sphere: Convergence to global minimizers and machine learning," *Jour. Mach. Learn. Res.*, vol. 22, no. 237, pp. 1–55, 2021.

[31] Hui Huang, Jinniao Qiu, and Konstantin Riedl, "On the global convergence of particle swarm optimization methods," *Appl. Math. Optim.*, vol. 88, pp. Article #30, 2023.

[32] Spyridon Pougkakiotis and Dionysis Kalogerias, "A zeroth-order proximal stochastic gradient method for weakly convex stochastic optimization," *SIAM J. Sci. Comput.*, vol. 45, no. 5, pp. A2679–A2702, 2023.

[33] E. Kazemi and L. Wang, "Efficient zeroth-order proximal stochastic method for nonconvex nonsmooth black-box problems," *Mach. Learn.*, vol. 113, pp. 97–120, 2024.

[34] Zaikun Zhang and Alexis Montoison, "PRIMA.jl," https://github.com/libprima/PRIMA.jl, 2023–25.