




## Article

# A Distance Correlation Approach for Optimum Multiscale Selection in 3D Point Cloud Classification

Manuel Oviedo-de la Fuente <sup>1,\*</sup>, Carlos Cabo <sup>2,3</sup>, Celestino Ordóñez <sup>3</sup> and Javier Roca-Pardiñas <sup>4</sup><sup>1</sup> CITIC, Research Group MODES, Department of Mathematics, Universidade da Coruña, 15001 Coruña, Spain<sup>2</sup> Faculty of Science and Engineering, Swansea University, Swansea SA2 8PP, UK; carlos.cabo@swansea.ac.uk<sup>3</sup> Department of Mining Exploitation and Prospecting, Universidad de Oviedo, 33004 Oviedo, Spain; ordonezcelestino@uniovi.es<sup>4</sup> Department of Statistics and OR, SiDOR Research Group & CINBIO, Universidade de Vigo, 36310 Vigo, Spain; roca@uvigo.es

\* Correspondence: manuel.oviedo@udc.es

**Abstract:** Supervised classification of 3D point clouds using machine learning algorithms and handcrafted local features as covariates frequently depends on the size of the neighborhood (scale) around each point used to determine those features. It is therefore crucial to estimate the scale or scales providing the best classification results. In this work, we propose three methods to estimate said scales, all of them based on calculating the maximum values of the distance correlation (DC) functions between the features and the label assigned to each point. The performance of the methods was tested using simulated data, and the method presenting the best results was applied to a benchmark data set for point cloud classification. This method consists of detecting the local maximums of DC functions previously smoothed to avoid choosing scales that are very close to each other. Five different classifiers were used: linear discriminant analysis, support vector machines, random forest, multinomial logistic regression and multilayer perceptron neural network. The results obtained were compared with those from other strategies available in the literature, being favorable to our approach.

**Keywords:** 3D point clouds; multiclass classification; feature selection; distance correlation; functional data



**Citation:** Oviedo-de la Fuente, M.; Cabo, C.; Ordóñez, C.; Roca-Pardiñas, J. A Distance Correlation Approach for Optimum Multiscale Selection in 3D Point Cloud Classification. *Mathematics* **2021**, *9*, 1328. <https://doi.org/10.3390/math9121328>

Academic Editors: Ana M. Aguilera and M. Carmen Aguilera-Morillo

Received: 14 April 2021

Accepted: 2 June 2021

Published: 9 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As a result of the advances in photogrammetry, computer vision and remote sensing, accessing massive unstructured 3D point cloud data is becoming easier and easier. Simultaneously, there is a growing demand for methods for the automatic interpretation of these data. Machine learning algorithms are among the most used methods for 3D point cloud segmentation and classification given their good performance and versatility [1–4]. Many of these algorithms are based on defining a set of local geometric features obtained through calculations on the vicinity of each point (or the center of a voxel when voxelization is carried out to reduce computing time) as explanatory variables. Local geometry depends on the size of the neighborhood (scale) around each point. Thus, a point can be seen as belonging to an object of different geometry, such as a line, a plane or a volume, depending on the scale [5]. As a result, the label assigned to each point can also change with this variable. The neighborhood around a point is normally defined by a sphere of fixed radius centered on each point [6] or by a volume limited by a fixed number of the closest neighbors to that point [7]. A third method, which is normally applied to airborne LiDAR (Light Detection and Ranging) data, consists of selecting the points in a cylinder of a fixed radius [8]. The local geometry of each point on the point cloud is mainly obtained from the covariance matrix, although other alternatives are possible, such as in [9], where Delaunay triangulation and tensor voting were used to extract object contours from LiDAR point clouds.

Given the importance of the scale in the result of the classification, there has been a great interest in estimating the scale (size of the neighborhood) or combination of scales (multiscale approach) [10–13] that provide the best results (the smallest error in the classification). Unfortunately, this is not a trivial question, and several methods have been proposed to determine optimum scales. A heuristic but largely used method is to try a few scales considering aspects such as the density of the point cloud, the size of the objects to be classified and the experience of the people solving the problem and select the scale or scales that provide the best solution. Differences in the density through the point cloud lead to errors in the classification [14]. In [15], the sparseness in the point cloud was addressed through upsampling by a moving least squares method. Another possibility is to select several scales at regular intervals or at intervals determined by a specific function, such as in [16], where a quadratic function of the radius of a sphere centered in each point was used. Obviously, these procedures cannot guarantee an optimal solution, so some more objective and automatic alternatives have been proposed. One of them is to estimate the optimum scales taking into account the structure of the local covariance matrix obtained from the coordinates of the points, and a measure of the uncertainty, such as the Shannon entropy [17,18]. Another alternative is to relate the size of the neighborhood with the point density and the curvature at each point [19]. The size of the neighborhood can be fixed across the point cloud, but the results can improve when it is changed from point to point [20].

There are different approaches for feature selection in classification problems. Some of them, for example, the meta algorithm optimal feature weighting [21], random forests [22] or regularization methods, such as Lasso [23], select the features at the time of performing the classification. By contrast, other methods (known as filter methods), such as Anova, Kruskal, AUC test [24] or independent component analysis [25], perform the selection prior to establishing the classification model. In this work, we propose a method inside this second category. As in [26], we assume that a good approach to the optimum scale selection should be that for which the distance correlation [27] between the features and the labels of the classes has high values, but in this case we look for a combination of scales that provides the best classification, instead of selecting just one scale. In summary, we propose a simple, objective and model independent approach to address an unsolved problem: the determination of the optimal scales in 3D point cloud multiscale supervised classification.

## 2. Methodology

Given a sample  $\{X_i^j, Y_i\}_{i=1}^n$ , where  $X^j$  represents the predictors,  $j = 1, \dots, p$ , and  $Y_i \in Z^+$ , we are interested in determining a model that assigns values to  $Y$  given the corresponding features  $X^j$ . Each feature depends on the values of a variable  $k$  observed in  $k = 1, \dots, K$  discretization points. Accordingly,  $X^j$  is a vector in  $\mathbb{R}^K$ . For each  $X^j$ , the response variable  $Y$  follows a multinomial distribution with  $L$  possible levels  $C_1, \dots, C_L$  and associated probabilities  $P_l(X^j) = P(Y = C_l | X^j)$ ,  $l = 1, \dots, L$ .

In the context of our particular problem,  $X^j(k)$  represents each of the features obtained from the point cloud to be classified, whose values are calculated, for each training point, at a finite number of scales  $k = 1, \dots, K$ , each of them representing the size of the local vicinity around each point. In contrast to the standard procedure, by which only a few scales are used, here the features are calculated at a much larger number of scales in order to be able to select those containing the relevant information to solve the classification problem.

The initial hypothesis is that for each feature only a few values of  $k$  (scales) provide useful information to perform the classification, and that these scales correspond to high values of the distance correlation between the features and the labels representing each category. Generally speaking, the distance correlation [27,28] between two random vectors  $X \in \mathbb{R}^p$  and  $Y \in \mathbb{R}^q$  is defined as

$$\mathcal{R}^2(X, Y) = \begin{cases} \frac{\mathcal{V}^2(X, Y)}{\sqrt{\mathcal{V}^2(X)\mathcal{V}^2(Y)}}, & \mathcal{V}^2(X)\mathcal{V}^2(Y) > 0 \\ 0, & \mathcal{V}^2(X)\mathcal{V}^2(Y) = 0 \end{cases} \tag{1}$$

$$\mathcal{V}^2(X, Y) = \|f_{X,Y} - f_X f_Y\|^2$$

where  $\mathcal{V}^2(X, Y) = \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|f_{X,Y}(t,s) - f_X(t)f_Y(s)|^2}{|t|^{1+p}|s|^{1+q}} dt ds$  represents distance covariance (see [29] for an application of distance covariance for variable selection in functional data classification), a measure of the distance between  $f_{X,Y} = \int_{\mathbb{R}} \int_{\mathbb{R}} e^{i(sx+ty)} f(x,y) dx dy, \{s,t\} \in \mathbb{R}$ , the joint characteristic function of random vectors  $X$  and  $Y$ , and the product  $f_X f_Y$  of the characteristic functions of  $X$  and  $Y$ , respectively. For their part,  $c_p$  and  $c_q$  are constant depending on the dimensions  $p$  and  $q$ , respectively.

A distance correlation has some advantages over other correlation coefficients, such as the Pearson correlation coefficient: (1) it measures non-linear dependence, (2)  $X$  and  $Y$  do not need to be one dimensional variables, (3)  $\mathcal{R}(X, Y) = 0 \Leftrightarrow X, Y$  are independent, that is, independence is a necessary and sufficient condition for the nullity of distance correlation.

Given that  $\mathcal{R}(\mathcal{X}, \mathcal{Y})$  is defined for  $X$  and  $Y$  random vector variables in arbitrary finite dimension spaces, in this study, the distance correlation was calculated both for a single scale  $\mathcal{R}(k) = \{\mathcal{R}(X(k_s), Y)\}_{s=1}^K$  or for a set of scales (discretization points)  $\mathcal{R}(k_1, \dots, k_v) = \mathcal{R}(X(k_1, \dots, k_v), Y)$ . Our aim is to select a set of critical scales (those which contain transcendental information for the classification)  $\tilde{k} = \tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_{\tilde{K}}$ , with  $\tilde{K} < K$ , which maximize the distance correlation. Solving this problem can be time consuming given that it requires calculating the distance correlation from a very high number of combinations. In this work, three different approaches have been proposed to determine the arguments providing the maximum distance correlation values.

The first approach (Algorithm 1) is an iterative method that looks for the best combinations of all the combinations of the scales  $K$ , taking a number  $\tilde{K}$  of scales at a time. It is a procedure that in each iteration fix the last  $\tilde{K} - 1$  values of vector  $\tilde{k}$  and randomly selects the remaining value until a maximum of  $\mathcal{R}(\tilde{k})$  is reached. This is a force brute procedure that does not look for local maximums directly.

The second approach (Algorithm 2.a) calculates the DC at each value of  $k$  for each feature and looks for the local maximums of the distance correlation, sorting them in decreasing order, and finally selecting the first  $\tilde{K}$  values of  $k$ .

Finally, the third approach (Algorithm 2.b) only differs from the second approach in that the distance correlation is smoothed before calculating the local maxima. Then, the values of DC calculated at discrete scales are considered as coming from a smooth function  $m(k)$  so that  $\mathcal{R}(k) = m(k) + \varepsilon$ ,  $\varepsilon$  being a zero mean independent error term and  $k \in [1, K]$ . In this way, close local maximums providing redundant information are avoided. In particular, this work follows the same idea as [29] but using a B-spline base instead of kernel-type smoothing.

Each algorithm is run several times, and the values of  $k$  in  $\tilde{k} = (\tilde{k}_1, \dots, \tilde{k}_{\tilde{K}})$  obtained in each step are stored. Those values of  $\{\tilde{k}_j\}_{j=1}^{\tilde{K}}$  that are the most frequent are considered as the critical points (scales). Once these critical scales have been selected, a classification model is fitted to the features at these scales, avoiding the inherent drawbacks of high-dimensional feature spaces. The algorithms for each of the methods are written below, although the second and third methods are written together, since they only differ in a sentence corresponding to the adjustment of a smooth function to the vector of distance correlation. In each algorithm,  $n$  represents the size of the data,  $K$  the number of scales used to calculate the features and  $\tilde{K}$  the number of critical points selected.

---

**Algorithm 1** For scale selection by brute force of the DC.

---

**Step 0:** Randomly select the initial estimates  $\tilde{k}^0 = (\tilde{k}_1^0, \dots, \tilde{k}_{\tilde{K}}^0)$  taking a sample of  $\tilde{K} \ll K$  size without replacement.

**Step 1:** Cycle  $s = 1, \dots, \tilde{K}$  calculating the update

$$\hat{k}_s = \underset{k}{\operatorname{argmax}} \mathcal{R}(\tilde{k}_1, \dots, \tilde{k}_{s-1}, k, \tilde{k}_{s+1}, \dots, \tilde{k}_{\tilde{K}})$$

**Step 2:** Repeat **Step 1** replacing  $\tilde{k}^0 = (\tilde{k}_1^0, \dots, \tilde{k}_{\tilde{K}}^0)$  by  $\tilde{k} = (\tilde{k}_1, \dots, \tilde{k}_{\tilde{K}})$  until there is no change between  $\tilde{k}^0$  and  $\tilde{k}$ .

---



---

**Algorithm 2** For scale selection by local maxima of the DC.

---

(Algorithm 2.a)

**Step 1:** Calculate  $\mathcal{R}(k)$  for  $k = 1, 2, \dots, K$ .

**Step 2:** Smooth  $\mathcal{R}(k)$  as a function of  $k$  (only for the third approach-Algorithm 2.b).

**Step 3:** Compute  $(\tilde{k}_1, \dots, \tilde{k}_{\tilde{K}})$  corresponding to local maxima of  $\mathcal{R}(k)$ , sorting them in decreasing order:  $\mathcal{R}(\tilde{k}_1) > \mathcal{R}(\tilde{k}_2), > \dots > \mathcal{R}(\tilde{k}_{\tilde{K}})$ .

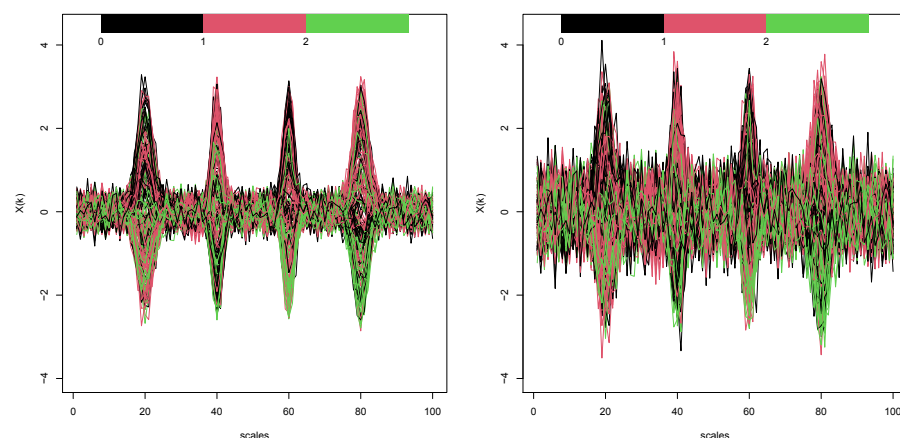
---

### 3. Simulation Study

This section reports the procedure followed in order to evaluate the practical performance of the proposed methodology using simulated data. We consider each simulated feature  $\{X_i(k)\}_{i=1}^n$  drawn from

$$X_i(k) = \sum_{j=1}^4 w_{ij} \exp\left(\frac{k_j - k}{\sigma_j}\right)^2 + \varepsilon_i(k) \quad k \in (0, 100)$$

with  $(k_1, k_2, k_3, k_4) = (20, 40, 60, 80)$  being the critical points or points of interest to be detected,  $(\sigma_1, \sigma_2, \sigma_3, \sigma_4) = (3, 2, 2, 3)$  a measure of how much sharpness is  $X_i(k)$  around  $k_j$ , and  $w_{ij}$  weights that have been simulated independently from a uniform distribution in the interval  $[-2.5, 3.0]$ . The errors  $\varepsilon_i(k)$  were generated from a random Gaussian process with covariance matrix  $\Sigma = \sigma_0 \mathbf{I}$ , being  $\sigma_0$  a scalar. A set of predictors (features) for  $\sigma_0 = 0.05$  and  $\sigma_0 = 0.25$  is represented in Figure 1.



**Figure 1.** Functional predictors  $\mathcal{X}_i(k), i = 1, \dots, n$ , colored by the corresponding outcome variable ( $C_1$  in black,  $C_2$  in red and  $C_3$  in green) for two different values of the standard deviation in the error term ( $\sigma_0 = 0.05$  in the left, and  $\sigma_0 = 0.25$  in the right).

Given  $X_i$ , the corresponding outcome variable  $Y_i$  was generated from a multinomial distribution with three possible results  $C_1, C_2$  and  $C_3$ , with associated probabilities  $P_j(X_i) = P(Y_i = C_j | X_i), j = 1, 2, 3$ , given by

$$P_j(X_i) = \frac{\exp(\eta_j(X_i))}{\exp(\eta_1(X_i)) + \exp(\eta_2(X_i))} \quad \text{for } j = 1, 2$$

and  $P_3(X_i) = 1 - P_1(X_i) - P_2(X_i)$ . We specifically define  $\eta_1(X_i) = X_i(20) + X_i(60)$  and  $\eta_2(X_i) = X_i(40) + X_i(80)$ .

A number  $n = n_e + n_p$  of independent samples  $\{(X_i, Y_i)\}_{i=1}^n$  were generated under the above scenario for different variance parameters  $\sigma_0$ . Moreover, we randomly split the sample into a training set ( $n_e = 140$ ), used in the estimation process, and a test set ( $n_p = 60$ ), for prediction. The curves were discretized in  $k = 100$  equi-spaced points in  $k \in (0, 100)$ .

The performance of the algorithms was evaluated for each of the critical points (as mentioned above, the theoretical critical points in the simulation scenario were  $k = (20, 40, 60, 80)$ ) by means of the mean square error on a number of repetitions  $R$  (we specifically chose  $R = 400$ ):

$$MSE = \frac{1}{R} \sum_{r=1}^R (\hat{k}_j^r - k_j)^2, j = 1, \dots, 4$$

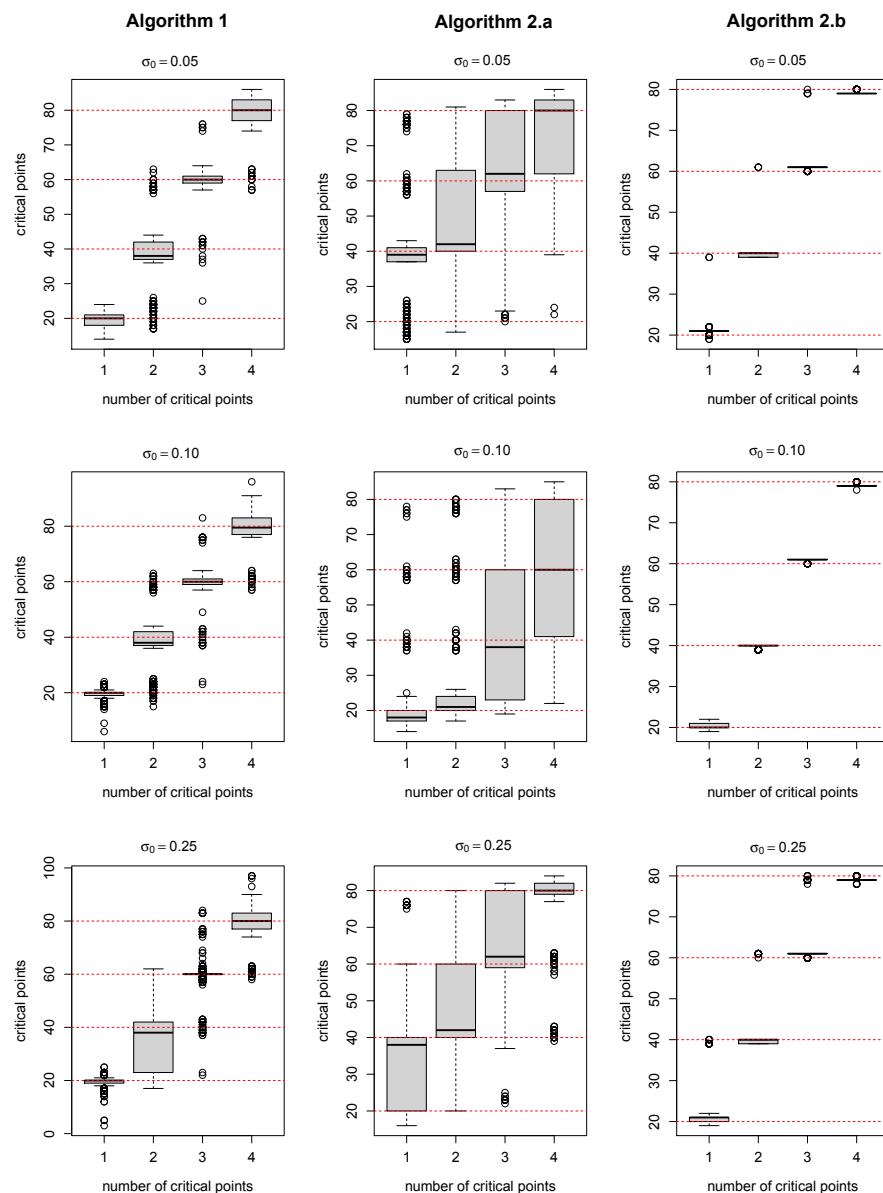
where  $\hat{k}^r$  is the estimated optimal subset of scale points in the simulation  $r$ .

$MSE(k)$  values for the different proposed algorithms and different values of  $\sigma_0$  are summarized in Table 1. For the three algorithms, the mean squared error and the dispersion of the estimated critical points around the mean increases with  $\sigma_0$ , as expected. Algorithm 2.a produced the worst results, both in detecting the critical points as well as in the uncertainty of the points detected. In contrast, Algorithm 2.b presented the best performance, so smoothing the distance correlation functions before calculating the critical points had a positive effect. Despite the fact that Algorithm 1 produces reasonable results in terms of accuracy in estimating the critical points, it is striking that the standard deviation is too large in opposition to Algorithm 2.b.

**Table 1.** Mean values of  $MSE^j, j = 1, \dots, 4$  (standard deviation in brackets) for different values of  $\sigma_0$ .

$\sigma_0$	Algorithm	$MSE^1$	$MSE^2$	$MSE^3$	$MSE^4$
0.05	1	1 (6.0)	9 (138.3)	1 (93.1)	9 (72.5)
	2.a	361 (932.6)	289 (578.3)	361 (282.1)	16 (515.5)
	2.b	1 (28.9)	0 (35.4)	1 (46.5)	1 (0.3)
0.10	1	1 (13.7)	9 (172.9)	1 (136.9)	9 (93.2)
	2.a	9 (492.8)	400 (282.1)	484 (595.2)	400 (991.9)
	2.b	0 (0.5)	0 (0.4)	1 (0.2)	1 (0.3)
0.25	1	0 (8.1)	289 (179.0)	0 (164.5)	9 (128.8)
	2.a	324 (649.3)	324 (474.1)	361 (240.4)	4 (388.7)
	2.b	1 (74.7)	0 (98.0)	1 (100.3)	1 (0.5)

Figure 2 shows boxplots representing the results of the simulation to detect each of the critical points (20, 40, 60, 80) for three different values of  $\sigma_0$  and for each algorithm. Clearly Algorithm 2.b is the most accurate, while the worst results corresponded to Algorithm 2.a. Increasing  $\sigma_0$  produces uncertainty in the determination of the critical points for Algorithms 1 and 2.a, but hardly affects Algorithm 2.b.



**Figure 2.** Boxplots of the estimated critical scales together with the theoretical scales (red lines) for  $\sigma_0 = 0.05$  (top),  $\sigma_0 = 0.10$  (middle) and  $\sigma_0 = 0.25$  (bottom) under Algorithm 1 (left), Algorithm 2.a (middle) and Algorithm 2.b (right).

In addition, to measure the effect of the errors in the detection of the critical points in the estimation of the probabilities, the mean squared error of the probabilities for each class was also calculated:

$$MSE(P) = \frac{1}{n_p} \sum_{i=1}^{n_p} \frac{1}{3} \sum_{j=1}^3 (\hat{P}_j(X_i) - P_j(X_i))^2$$

The results are shown in Table 2, considering a different number  $\tilde{K}$  of critical points and values of  $\sigma_0$ , again, the best results, that is, small values of MSE (and its standard deviation) and total accuracy close to that of the theoretical model, corresponded to Algorithm 2.b, whereas Algorithm 2.a provided the worst results.

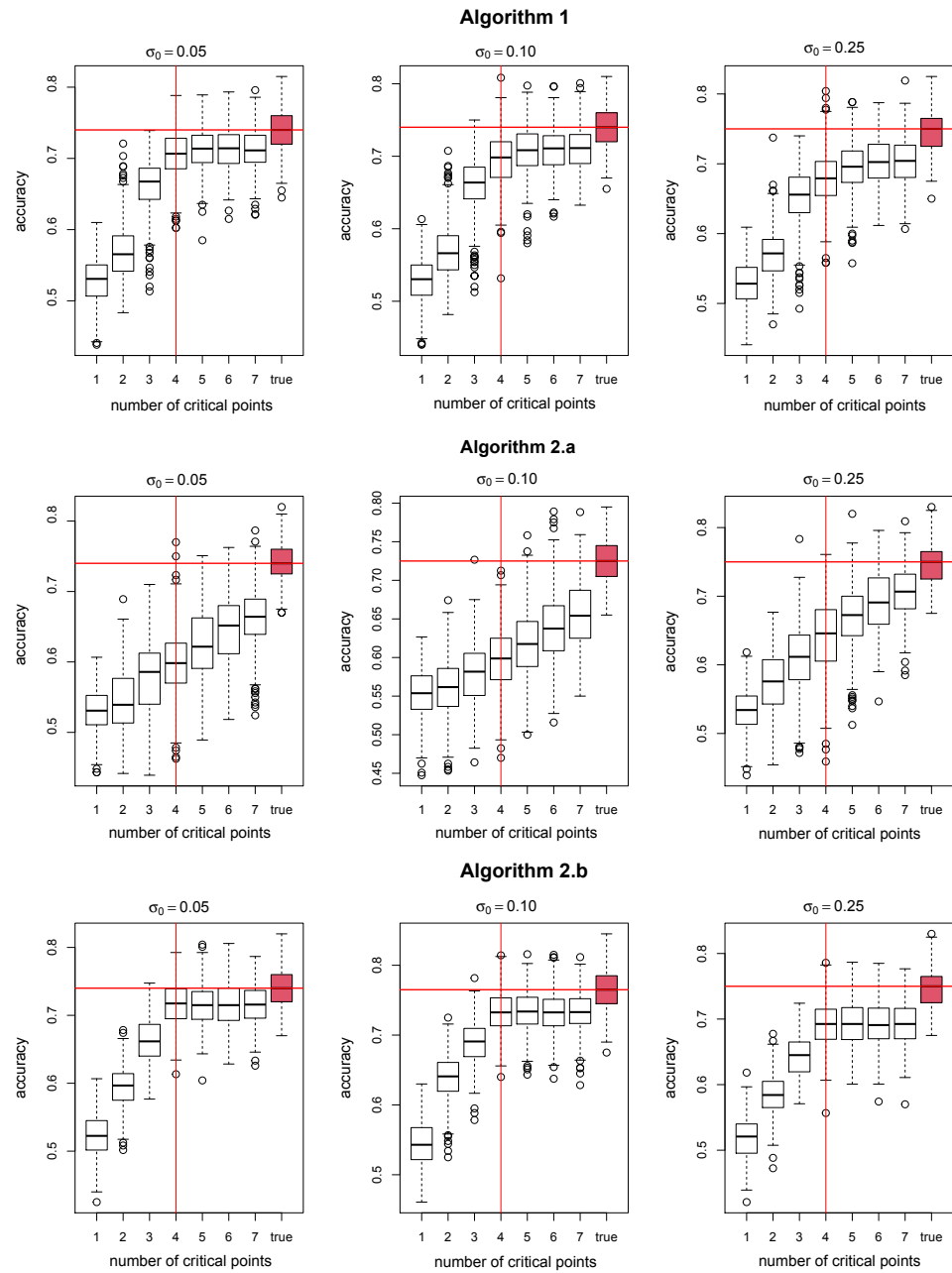
**Table 2.** Mean values of  $MSE(P) = (1/400) \sum_{r=1}^{400} MSE(P)_r$  and accuracy (standard deviation in brackets) for a different number of critical points.  $Acc_{th}$  makes reference to the accuracy obtained by using the true model, while  $Acc_{sim}$  represents the accuracy values obtained from the simulation.

$\tilde{K}$	$\sigma_0$	Alg.	MSE		$Acc_{th}$	$Acc_{sim}$	
3	0.05	1	0.033	(0.007)	0.74	0.67	(0.03)
		2.a	0.060	(0.012)	0.74	0.59	(0.05)
		2.b	0.035	(0.003)	0.74	0.66	(0.03)
	0.10	1	0.035	(0.008)	0.73	0.66	(0.04)
		2.a	0.058	(0.009)	0.73	0.58	(0.04)
		2.b	0.034	(0.003)	0.73	0.69	(0.03)
	0.25	1	0.038	(0.010)	0.75	0.65	(0.04)
		2.a	0.059	(0.014)	0.75	0.61	(0.05)
		2.b	0.042	(0.004)	0.75	0.65	(0.03)
4	0.05	1	0.013	(0.011)	0.74	0.71	(0.03)
		2.a	0.058	(0.012)	0.74	0.60	(0.05)
		2.b	0.008	(0.004)	0.74	0.72	(0.03)
	0.10	1	0.022	(0.013)	0.73	0.70	(0.04)
		2.a	0.054	(0.009)	0.73	0.60	(0.04)
		2.b	0.010	(0.003)	0.73	0.73	(0.03)
	0.25	1	0.035	(0.012)	0.75	0.68	(0.04)
		2.a	0.042	(0.015)	0.75	0.65	(0.06)
		2.b	0.019	(0.008)	0.75	0.69	(0.03)
5	0.05	1	0.013	(0.007)	0.74	0.71	(0.03)
		2.a	0.052	(0.014)	0.74	0.62	(0.05)
		2.b	0.008	(0.004)	0.74	0.72	(0.03)
	0.10	1	0.017	(0.009)	0.73	0.71	(0.03)
		2.a	0.051	(0.011)	0.73	0.62	(0.04)
		2.b	0.010	(0.003)	0.73	0.73	(0.03)
	0.25	1	0.026	(0.012)	0.75	0.69	(0.04)
		2.a	0.036	(0.015)	0.75	0.67	(0.05)
		2.b	0.019	(0.008)	0.75	0.69	(0.04)

Figure 3 shows boxplots of total accuracy for all the repetitions and for the three algorithms tested, according to the number of critical points. The red items correspond to the theoretical model. Again, the best results correspond to Algorithm 2.b followed by Algorithm 1. Note that more than four critical points were tried, but, either way, for Algorithms 1 and 2.b these two metrics stabilized at  $\tilde{k} = 4$ .

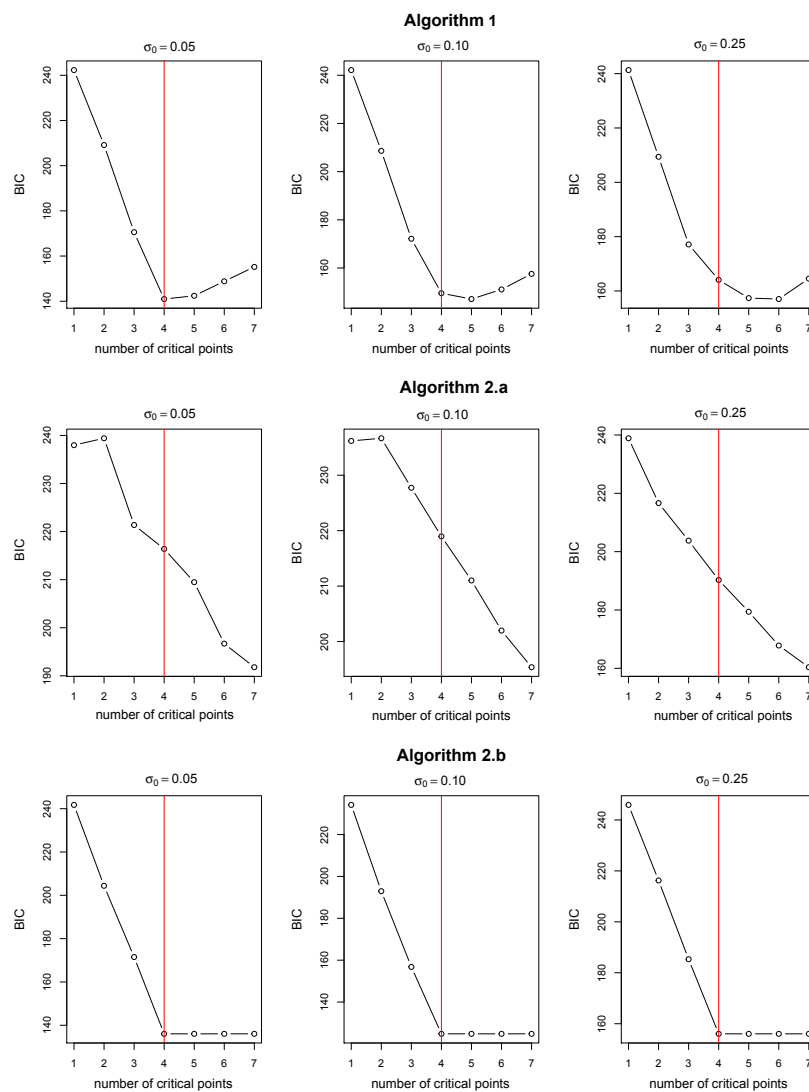
The Bayesian Information Criterion for each Algorithm is represented versus the number of critical points, for three different values of  $\sigma_0$  in Figure 4. As can be appreciated, Algorithm 2.a does not work properly: the number of critical points is not detected, and there is a great dispersion in accuracy. Algorithm 2.b reaches a minimum value of BIC at the correct number of critical points  $\tilde{K} = 4$ , regardless of the value of  $\sigma_0$ .

A comparison of the total accuracy for the three algorithms tested with the theoretical model (in red), fixing  $\tilde{K} = 4$ , for different values of  $\sigma_0$ , is shown in Figure 5. A decrease in the accuracy is accompanied by an increase in the standard deviation of the error. Based on the results of the classification for each algorithm it seems that there is a relationship between inaccuracy in the estimation of the critical points and errors in the classification. As before, the best algorithm is Algorithm 2.b, followed by Algorithm 1, while Algorithm 2.a behaves badly. Note that inaccuracy in Algorithm 2.b is mainly due to bias (see right panel of Figure 2), since there is hardly any variability, while for the other two algorithms there is both bias and variability, especially for Algorithm 2.a (see center panel of Figure 2).

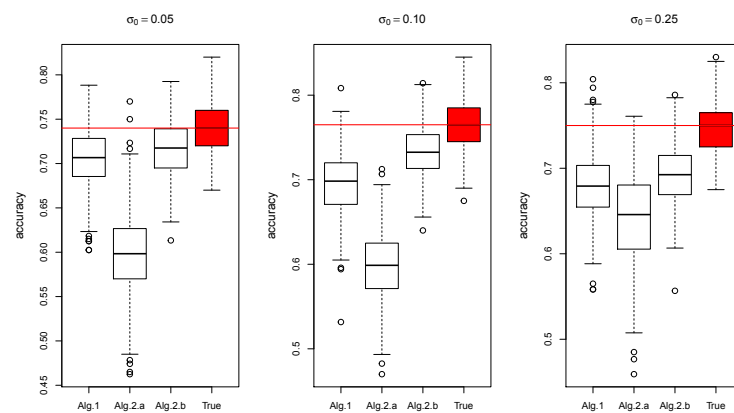


**Figure 3.** Boxplots of the total accuracy obtained with the three algorithms: **1** (top), **2.a** (middle), **2.b** (bottom). The theoretical value of these metrics is given by the red line. The red boxplot corresponds to the theoretical probabilities.





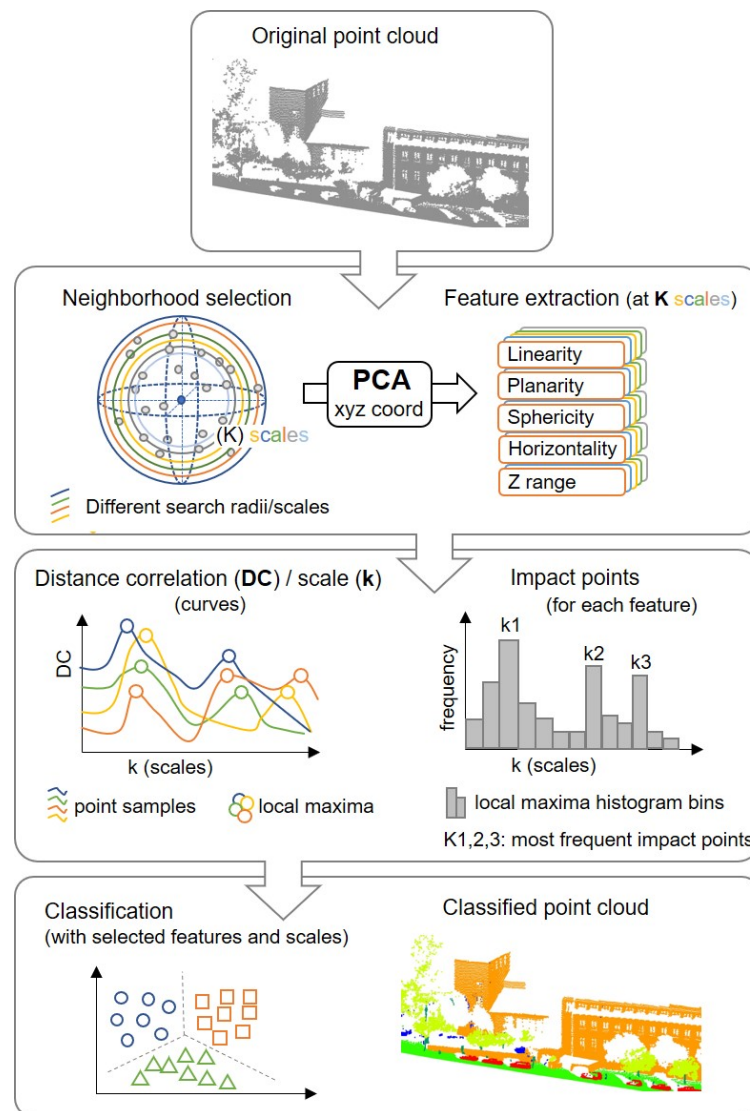
**Figure 4.** BIC vs. number of critical points  $\tilde{K}$  under the Algorithm 1 (left), Algorithm 2.a (middle) and Algorithm 2.b (right), for  $\sigma_0 = 0.05$  (top),  $\sigma_0 = 0.10$  (middle) and  $\sigma_0 = 0.25$  (bottom).



**Figure 5.** Total accuracy for the three Algorithms 1 and 2 under study at  $\sigma_0 = 0.05$  (top),  $\sigma_0 = 0.10$  (middle) and  $\sigma_0 = 0.25$  (bottom). The boxplots on the right (in red) correspond to the theoretical model.

#### 4. Case Study

In addition to testing our method on simulated data, we also applied it to a real dataset where the objects to be labeled are elements of an urban environment. Figure 6 shows the operations followed to perform the classification of the point cloud using the proposed methodology.



**Figure 6.** Workflow of the proposed methodology to select the optimum scales (impact points) and perform the multiscale classification.

##### 4.1. Dataset and Feature Extraction

The real dataset used to test our approach was the Oakland 3D point cloud [30], a benchmark dataset of 1.6 million points that has been previously used in other studies concerning point cloud segmentation and classification. The objective is to automatically assign a label to each point in the point cloud from a set of features obtained from the coordinates ( $X, Y, Z$ ) of a training dataset. Specifically, there are six categories (labels) of interest, as is shown in Figure 7.

The point cloud was collected around the CMU campus in Oakland-Pittsburgh (USA) using a Mobile Laser Scanner system (MLS) that incorporates two-dimensional laser scanners, an Inertial Measurement Unit (IMU) and a Global Navigation Satellite receptor (GNSS), all calibrated and mounted on the Nablav 11 vehicle. Figure 7 shows a small part

of the point cloud, where a label represented with a color has been assigned to each point. A total of six labels have been considered.



**Figure 7.** Oakland MLS point cloud. The classes to be extracted have been represented in different colors.

The features representing the local geometry around each point were obtained through the eigendecomposition of the covariance matrix  $\Sigma$  [13,31]:

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (\mathbf{p}_i - \bar{\mathbf{p}})^T (\mathbf{p}_i - \bar{\mathbf{p}}) = \mathbf{V} \Lambda \mathbf{V}^{-1} \tag{2}$$

where vector  $\mathbf{p}_i = (X_i, Y_i, Z_i)$  represents each point in the point cloud,  $\mathbf{V}$  a matrix whose columns are the eigenvectors  $\mathbf{v}_i, i = 1, \dots, 3$ , and  $\Lambda$  a diagonal matrix whose non-zero elements are the eigenvalues  $\lambda_1 > \lambda_2 > \lambda_3 > 0$ .

The three eigenvalues and the eigenvector  $\mathbf{v}_3$  were used to calculate the five features registered in Table 3. Z range for each point is calculated considering the points in a vertical column of a specific section (scale) around that point. In order to avoid the negative effect of outliers, instead of using the range of Z coordinates we used the values between the 5th and 95th percentiles. An explanation of the geometrical meaning of these and other local features can be obtained in [12,16].

**Table 3.** Features extracted from the point cloud.

Name	Formula
Linearity	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity	$\frac{\lambda_3}{\lambda_1}$
Horizontality	$\frac{\text{acos}(\mathbf{v}_3 \cdot \mathbf{z})}{\ \mathbf{v}_3\ }$
Z range	$Z_{max} - Z_{min}$

The spatial distribution of these features at different scales are shown in Figure 8.

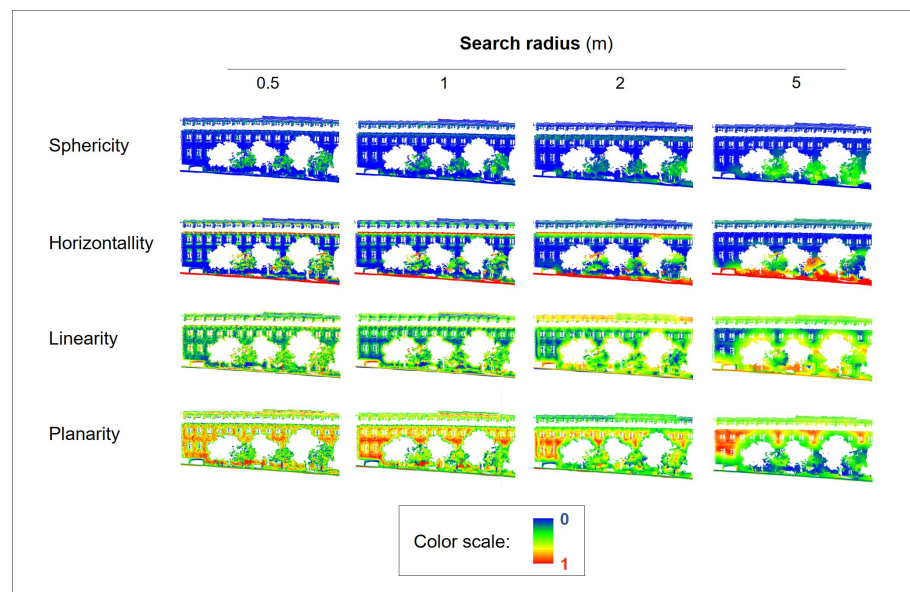


Figure 8. Example of the features extracted at different scales.

#### 4.2. Neighborhood Selection

The proposed methodology for optimum scale selection (detection of critical points) was applied to solve the previous classification problem. Thus, we have a vector of input variables  $\mathcal{X} = (X_L(k), X_P(k), X_S(k), X_H(k), X_Z(k))$  representing the features (*linearity L*, *planarity P*, *sphericity S*, *horizontality H* and *Z range Z*) measured at different scales, and an output variable  $Y = \{C_1, \dots, C_5\}$ , which can also take five discrete values, the labels assigned to each type of object (cars, buildings, canopy, ground and poles). Our aim is to estimate an optimum neighborhood (scale) for each feature by means of distance correlation (DC), taking into account its advantage with respect to the Pearson coefficient. For each sample, each feature was evaluated at a regular grid of  $K = 100$  scales measured in centimeters in a linear scale from 50 to 300 cm. Figure 9 shows a sample of  $n = 150$  curves for each feature registered  $X_i(k), i = 1, \dots, 5$  in the interval  $k \in [k_1 = 50, k_{100} = 300]$  and the corresponding functional mean  $\bar{X}_i(k)$ , both colored by a class label.

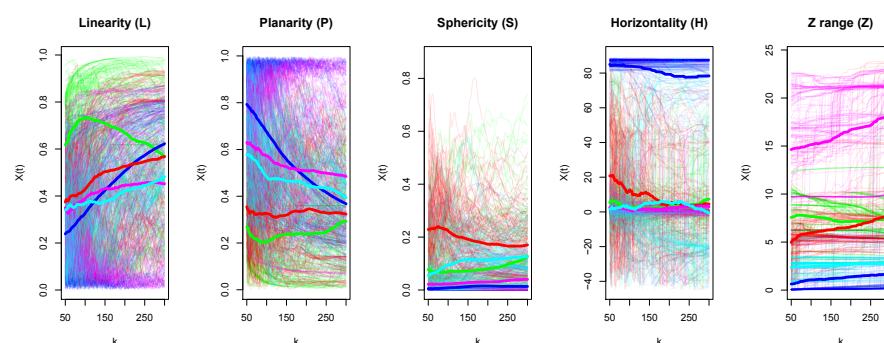
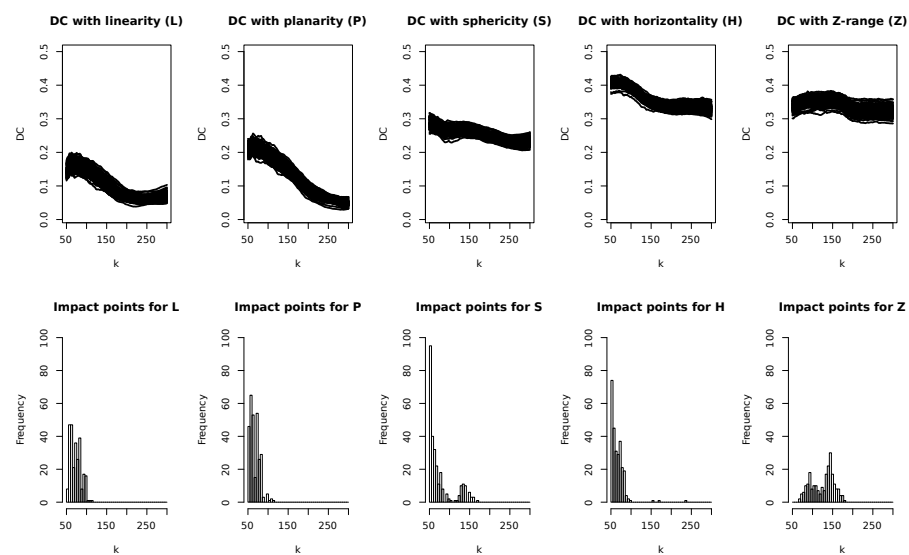


Figure 9. A sample of  $n = 150$  curves representing the features measured on  $K = 100$  scales. Each color represents a label or type of object: poles (green), ground (blue), vegetation (red), buildings (magenta), and vehicles (cyan). Mean values for each class are represented as wider lines.

Note the different performance of the features for the different classes and scales. For instance, *horizontality* takes high values for the ground, and it is uniform at different scales. However, this feature shows abrupt jumps at certain scales for the poles, that could correspond to edge effects. As expected, *linearity* takes high values for the poles and low values for the buildings, while planarity is high for buildings and low for poles.

Figure 10 shows the distance correlation functions between a dummy matrix representation of the categorical response and the covariates ( $X_L(k), X_P(k), X_S(k), X_H(k), X_Z(k)$ ) for 100 repetitions of random samples of size  $n_e = 750$  (150 per class), corresponding to each of the features extracted. DC functions were calculated using the `fda.usc` package [32]. They are quite uniform, with not many peaks, and some of them are close together. This could cause problems in finding the relevant scales, similar to the effect of increasing the standard deviation observed with the simulated data. A histogram of the global maximum of distance correlation curves for those repetitions is depicted at the bottom of the figure. As can be appreciated, most of the relative maximums correspond to low scales (impact points), except for the *Z range* variable (5th-95th range of *z* axis).



**Figure 10.** Distance correlation functions for each of the features (**top**) and histogram of critical points. There are 100 curves, hence the same number of impact points, that were obtained by random sampling the data.

Again, smoothing the DC function before searching for the local maximums helped to discriminate the most important points. However, even when the number of local maximums decreases after smoothing DC, some of them have little influence on the classification. In fact, no more than three critical points were needed to obtain the best results of the classification for the different classifiers tested. The three most frequent scales in each histogram are shown in Table 4. As can be appreciated, the important scales take low values for all the features, with the exception of the *Z range*, for which important scales correspond to low to medium grades.

**Table 4.** Most frequent values of scales selected using Algorithm 2.b.

	$\tilde{k}_1$	$\tilde{k}_2$	$\tilde{k}_3$
Linearity	60.1	72.7	82.8
Planarity	60.1	50.0	75.2
Sphericity	50.0	70.1	133.2
Horizontality	50.0	67.7	85.3
Z range	146.0	108.1	156.1

The performance of our approach was contrasted with the proposal in [18], on which the optimal scale  $k_\lambda$  was calculated as the minimum of the Shannon entropy  $E_\lambda$ , which depends on the normalized eigenvalues  $e_i = \lambda_i / \Sigma_\lambda, i = 1, \dots, 3$ , of the local covariance matrix  $\Sigma$ :

$$E_\lambda = -e_1 \ln(e_1) - e_2 \ln(e_2) - e_3 \ln(e_3) \tag{3}$$

### 4.3. Classification

In this step we evaluate how the classifier translates the information of critical points of  $X$  (all features) into a classification error. This is a quite an interesting question in the functional context or when dealing with high-dimensional spaces. However, new important questions arise related to how representative the selected critical points (scales) shown before are or how to select a useful classifier for the classification problem among all the possibilities. Unfortunately, detecting the critical scales is not a simple task, as was proved in the simulation study by introducing Gaussian errors with standard deviation of different magnitude in a model relating features and classes. The greater the standard deviation, the greater the error in determining the exact values of the relevant scales. This is because these scales correspond to the peaks of  $\mathcal{R}_i(k)$ , and they are less sharp as the standard deviation of the error increases. In this situation, the best results were obtained when DC was smoothed before searching for the local maximum, and this was the method applied to perform the classification with real data.

Now, we have to deal with a purely multivariate classification problem in dimension  $K_N$  (the number of critical points for each functional variable) and many procedures are known to handle this problem (see, for example, [33]). Many classification methods have been described in the literature, but we limit our study to four proven methods that are representative of most of the types of classifiers, linear or non-linear, hard or probabilistic, ensemble or not. Specifically, the chosen methods are: linear discriminant analysis (LDA) [34,35], multiclass logistic regression (LR) [36,37], multiclass support vector machines (SVMs) [38,39], random forest (RF) [22] and feed-forward neural networks with a single hidden layer (ANN) [40]. They are among the top classification algorithms in machine learning, although there are some other classification methods that could be employed here such as, for example, Quadratic Discriminant Analysis (QDA) or Generalized Additive Models (GAMs).

The choice among the different classifiers could be influenced by their theoretical properties and/or the easiness to draw inferences. Better inferences can be drawn from more simple classifiers such as LDA or LR models. Furthermore, as is discussed in [41], such simple classifiers are usually hard to beat in real life scenarios, adding interpretability to the classification rule which is sometimes more important than predictability.

We consider three possibilities for the classification, depending on the number of critical scales used, according to Table 4:

1. A unique scale (impact point),  $\tilde{k}_{N_1} = \{\tilde{k}_1^L, \tilde{k}_1^P, \tilde{k}_1^S, \tilde{k}_1^H, \tilde{k}_1^Z\}$ , for each of the features, corresponding to the most frequent value of the impact points.
2. Two scales,  $\tilde{k}_{N_2} = \{\tilde{k}_{N_1}^L, \tilde{k}_2^L, \tilde{k}_2^P, \tilde{k}_2^S, \tilde{k}_2^H, \tilde{k}_2^Z\}$ , corresponding to the two most frequent critical scales for each of the features.
3. Three scales,  $\tilde{k}_{N_3} = \{\tilde{k}_{N_2}^L, \tilde{k}_3^L, \tilde{k}_3^P, \tilde{k}_3^S, \tilde{k}_3^H, \tilde{k}_3^Z\}$ , corresponding to the three most frequent critical scales for each of the features.

In order to contrast the performance of our approach, the same classification algorithms were applied to the feature values obtained using other scales. Specifically, we used the following values of the scale  $k$ :

- a.  $k_\lambda = \{k_\lambda^L, k_\lambda^P, k_\lambda^S, k_\lambda^H, k_\lambda^Z\}$ , obtained according to Equation (3).
- b.  $k_{seq} = \{k_{seq}^L, k_{seq}^P, k_{seq}^S, k_{seq}^H, k_{seq}^Z\}$ , linearly spaced scales corresponding to the following values of  $k$  in centimeters (cm): 50, 112, 175, 237, 300.

Training data ( $n_e = 200$  per class) and test data ( $n_p = 400$  per class) were sampled from different areas of the point cloud in order to ensure their independence. Table 5 compares the total accuracy obtained using LR, LDA, SVM, RF and ANN classifiers for all the scales studied: no important discrepancies between them were appreciated, with SVM having a narrow advantage over the others. A quick look at this table informs us that our proposal is better than using sequential scales or  $k_\lambda$ . Furthermore, using a multiscale scheme provides a slight improvement in accuracy with respect to using just one scale

( $\tilde{k}_{N1}$ ) for almost all the classifiers. The feature selection and classifier functions available in the R package `fda.usc` were used with the default parameters (without previous tuning).

**Table 5.** Total accuracy (and standard deviation) in % of the classification using five different classifiers depending on the scales evaluated in  $n_p = 2000$  samples in  $B = 100$  repetitions. The values were calculated by averaging the total accuracy in each repetition. The numbers in bold represent maximum values.

	LR	LDA	SVM	RF	ANN
$k_{50}$	80.7 (1.0)	79.6 (0.8)	82.0 (0.8)	80.5 (1.3)	79.6 (2.3)
$k_{112}$	76.8 (1.1)	78.9 (0.9)	81.8 (0.9)	81.0 (1.4)	79.0 (4.2)
$k_{175}$	69.6 (1.3)	71.8 (1.3)	74.4 (1.4)	72.6 (1.7)	71.8 (4.6)
$k_{237}$	58.7 (4.6)	57.1 (3.3)	72.3 (1.9)	69.4 (2.2)	66.0 (4.7)
$k_{300}$	51.9 (4.1)	51.1 (1.7)	71.5 (1.4)	67.3 (1.7)	62.8 (5.3)
$k_\lambda$	74.6 (9.0)	75.5 (8.7)	79.9 (4.6)	78.3 (5.4)	76.2 (6.7)
$k_{N1}$	83.1 (1.0)	81.9 (0.8)	84.9 (0.9)	82.9 (1.3)	<b>82.7</b> (3.0)
$k_{N2}$	<b>83.6</b> (0.9)	82.4 (0.8)	85.6 (0.7)	82.9 (1.6)	79.9 (6.4)
$k_{N3}$	83.3 (1.0)	<b>82.7</b> (0.7)	<b>85.7</b> (0.7)	<b>83.3</b> (1.4)	74.8 (7.8)

Table 6 shows the results of the classification for the test sample, in terms of precision and recall, for each of the scales tested, using a multinomial logistic regression LR and SVM classifiers. We limited the results to these two classifiers because there are few differences with the other three classifiers. Although some non-optimal scales provided the best results for some types of objects (ground and buildings), in global terms it can be concluded that the largest values of precision and recall correspond to  $\tilde{k}_{N2}$  and  $\tilde{k}_{N3}$ , and that they are practically the same in both cases.

**Table 6.** Metrics (precision and recall) of the classification by classes using LR and SVM for different scales in a test sample  $n_p = 2000$ . The numbers in bold represent maximum values in each column. Both metrics correspond to the average values in 100 repetitions.

	LR Classifier											
	Precision %						Recall %					
	Pol	Gro	Veg	Bui	Veh	Ave (sd)	Pol	Gro	Veg	Bui	Veh	Ave (sd)
$k_{50}$	70.6	96.9	73.7	88.0	75.7	81.0 (10.2)	73.6	<b>99.1</b>	71.4	77.0	82.3	80.7 (10.2)
$k_{112}$	63.3	<b>98.0</b>	69.7	74.5	79.7	77.0 (12.0)	77.6	97.7	49.5	78.4	80.7	76.8 (15.7)
$k_{175}$	51.1	96.8	66.6	63.7	75.2	70.7 (15.4)	63.4	92.4	32.5	77.9	82.1	69.6 (21.0)
$k_{237}$	36.1	97.3	57.8	53.6	62.5	61.4 (21.2)	48.5	90.0	29.4	78.8	46.6	58.7 (24.5)
$k_{300}$	27.8	95.3	49.5	48.2	46.8	53.5 (23.5)	33.1	87.3	32.1	<b>80.3</b>	26.5	51.9 (27.8)
$k_\lambda$	61.0	97.2	69.0	73.8	74.9	75.2 (15.8)	70.3	96.2	52.5	78.1	75.7	74.6 (19.1)
$k_{N1}$	75.0	97.8	75.8	89.1	79.2	83.4 (9.2)	78.2	99.0	72.4	78.6	87.5	83.1 (9.5)
$k_{N2}$	<b>75.1</b>	97.8	76.3	<b>90.9</b>	79.4	<b>83.9</b> (9.2)	78.5	98.7	<b>73.3</b>	78.7	88.6	<b>83.6</b> (9.2)
$k_{N3}$	74.8	96.9	<b>77.0</b>	89.0	<b>79.9</b>	83.5 (8.6)	<b>79.1</b>	98.3	72.9	78.5	<b>87.7</b>	83.3 (9.2)

Table 6. Cont.

SVM Classifier												
	Precision %						Recall %					
	Pol	Gro	Veg	Bui	Veh	Ave (sd)	Pol	Gro	Veg	Bui	Veh	Ave (sd)
$k_{50}$	70.5	97.4	77.0	<b>96.5</b>	73.7	83.0 (11.7)	78.4	<b>99.0</b>	71.1	77.2	84.6	82.0 (9.8)
$k_{112}$	67.3	98.7	77.0	84.4	83.1	82.1 (10.5)	77.9	96.5	60.0	80.0	94.4	81.8 (13.4)
$k_{175}$	55.0	99.1	70.9	74.7	76.8	75.3 (14.4)	67.1	91.2	41.6	76.9	<b>95.0</b>	74.4 (19.5)
$k_{237}$	51.2	98.6	65.7	78.9	73.2	73.5 (16.1)	64.3	88.9	40.5	76.3	91.4	72.3 (19.1)
$k_{300}$	49.3	98.1	58.6	87.7	70.5	72.8 (18.2)	60.0	86.6	41.2	77.0	92.7	71.5 (19.0)
$k_{\lambda}$	65.7	98.6	74.8	86.0	77.8	80.6 (13.0)	74.5	95.2	60.1	78.2	91.2	79.9 (14.8)
$k_{N1}$	75.8	98.9	80.1	93.8	79.0	85.5 (9.4)	80.0	98.8	75.1	79.4	91.4	84.9 (9.1)
$k_{N2}$	76.8	<b>99.2</b>	80.3	94.7	79.8	<b>86.2</b> (9.2)	<b>80.6</b>	98.9	<b>75.5</b>	<b>80.4</b>	92.7	85.6 (9.0)
$k_{N3}$	<b>77.3</b>	<b>99.2</b>	<b>80.6</b>	92.9	<b>80.4</b>	86.1 (8.7)	80.5	98.8	75.3	80.2	93.6	<b>85.7</b> (9.1)

Figure 11 represents boxplots of F1-index for the LR classifier, for each of the classes depending on the scale. The plot at the bottom right is the average value of F1 for the five classes. With a few exceptions, the highest F1 values correspond to the case where two ( $k_{N2}$ ) or three ( $k_{N3}$ ) optimum scales for each feature were used. However, the values of the median and the interquartile range are almost the same, so we cannot claim that there are significant differences between both options. For its part,  $k_{\lambda}$  led to the worst performance with low mean values and great dispersion because sometimes the extreme values of the scale were selected.

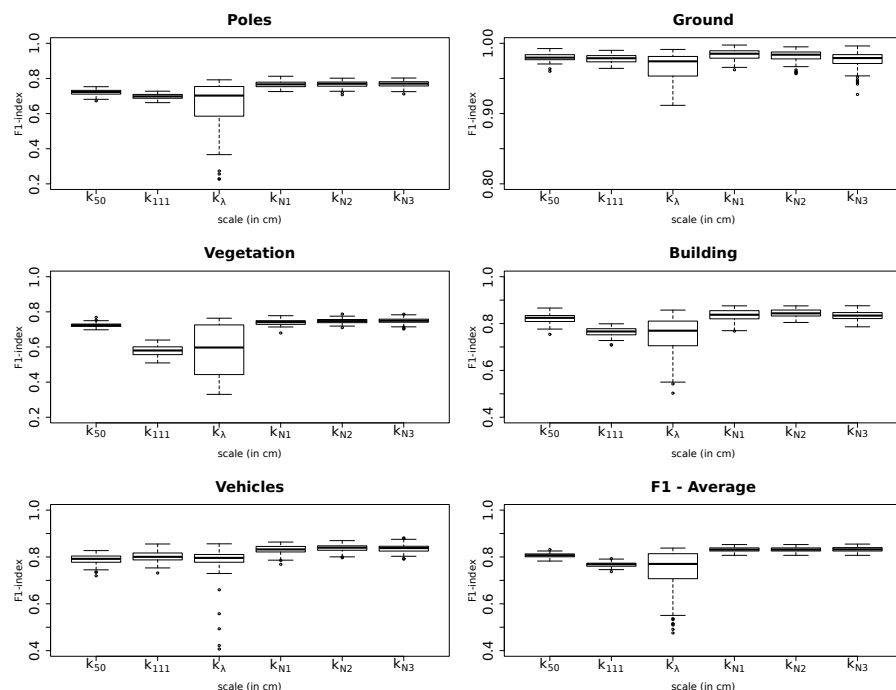


Figure 11. F1 index for each class and average F1 for all the classes (bottom right).



A drawback for the practical implementation of the proposed algorithms, specially for Algorithm 1, is the memory consumption. The overall consumption of memory when storing the distance matrices for  $p$  covariates with  $n$  elements each is  $(p + 1)(n(n - 1)/2 - n)$ . As an example, Table 7 shows the maximum memory consumed (in megabytes) and the execution time (in seconds) when the algorithms for different sample sizes and two different levels of error is executed in an Intel Core i7-1065G7 with 16MB of RAM.

**Table 7.** Computation times (in sec) and memory consumption (in MB) for computing the three algorithms using different parameters in  $B = 100$  repetitions.

$\sigma_0$	Algo.	TIME Mean (sd)		MEM Mean (sd)	
		$n = 400$	$n = 1000$	$n = 400$	$n = 1000$
0.05	1	59.17 (4.05)	448.21 (19.61)	132.10 (4.08)	210.45 (9.24)
	2a	3.09 (0.18)	23.17 (0.62)	110.40 (8.92)	194.90 (2.30)
	2b	3.20 (0.18)	23.22 (0.48)	113.50 (9.35)	199.75 (1.88)
0.25	1	56.03 (0.69)	523.77 (23.80)	133.90 (4.06)	208.55 (9.07)
	2a	3.12 (0.22)	23.28 (0.45)	109.50 (9.36)	196.75 (1.88)
	2b	3.08 (0.18)	22.97 (0.47)	112.55 (9.01)	199.75 (1.88)

## 5. Conclusions

In this work we propose three different algorithms to select optimum scales in multi-scale classification problems with machine learning. They are based on determining the arguments (scales) of the features with high distance correlation with the labels assigned to the objects. Distance correlation provides a measure of the association, linear or non-linear, between two random vectors of arbitrary dimensions, so it was expected that high correlations correspond to low classification errors. First, the proposed algorithms were tested simulating the distance correlation function and its relationship with the labels. The results were encouraging, supporting the validity of our proposal, and allowing us to establish the order of performance of the three algorithms. The best results were obtained when the distance correlation functions for each feature were smoothed before calculating the local maximum. Then, the algorithm that provided the best results with the simulated data was tested in a real classification problem involving a 3D point cloud collected using a mobile laser scanning system. Determining the optimum scales simplifies the classification problem for other point clouds, since they give us important information to limit the scales at which the features are determined, assuming the quality and density of the point clouds are similar to those of the training data and, of course, that we are classifying the same type of objects. The results obtained for the real problem were also positive, outperforming those obtained with other methods reported in the literature that use unique sequentially defined scales or the Shannon entropy. A maximum of three scales for each feature was sufficient to obtain the best results in the classification, measured in terms of precision, recall and F1-index. In addition, non-significant differences were found between the five classifiers tested.

**Author Contributions:** Conceptualization, C.O.; Software, M.O.-d.l.F.; Methodology, M.O.-d.l.F. and J.R.-P.; Resources, C.C.; Data preparation, C.C.; Validation: M.O.-d.l.F. and J.R.-P.; Writing original draft, C.O. and J.R.-P.; Writing-review & editing, C.C., C.O. and M.O.-d.l.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** Manuel Oviedo-de la Fuente acknowledges financial support by (1) CITIC, as Research Center accredited by Galician University System, funded by “Consellería de Cultura, Educación e Universidade from Xunta de Galicia”, supported in an 80% through European Regional Development Funds (ERDF), Operational Programme Galicia 2014-2020, and the remaining 20% by “Secretaría Xeral de Universidades” (Grant ED431G 2019/01), (2) by the Spanish Ministry of Science, Innovation and Universities grant project MTM2016-76969-P and (3) by “the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2020-14)”, all of them through the ERDF Funds. Celestino Ordóñez acknowledges support from the University of Oviedo (Spain) for recognized research

groups. Javier Roca-Pardiñas acknowledges financial support by the Grant MTM2017-89422-P (MINECO/AEI/FEDER, UE).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data source is cited in the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Bonneau, D.A.; Hutchinson, D.J. The use of terrestrial laser scanning for the characterization of a cliff-talus system in the Thompson River Valley, British Columbia, Canada. *Geomorphology* **2019**, *327*, 598–609. [[CrossRef](#)]
- Zhou, J.; Fu, X.; Zhou, S.; Zhou, J.; Ye, H.; Nguyen, H.T. Automated segmentation of soybean plants from 3D point cloud using machine learning. *Comput. Electron. Agric.* **2019**, *162*, 143–153. [[CrossRef](#)]
- Xie, Y.; Tian, J.; Zhu, X.X. Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [[CrossRef](#)]
- de Oliveira, L.M.C.; Lim, A.; Conti, L.A.; Wheeler, A.J. 3D Classification of Cold-Water Coral Reefs: A Comparison of Classification Techniques for 3D Reconstructions of Cold-Water Coral Reefs. *Front. Mar. Sci.* **2021**, *8*, 640713. [[CrossRef](#)]
- Brodu, N.; Lague, D. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 121–134. [[CrossRef](#)]
- Lee, I.; Schenk, T. Perceptual organization of 3D surface points. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2002**, *34*, 193–198.
- Linsen, L.; Prautzsch, H. Local versus global triangulations. *Proc. Eurographics* **2001**, *1*, 257–263.
- Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [[CrossRef](#)]
- Sreevalsan-Nair, J.; Jindal, A.; Kumari, B. Contour extraction in buildings in airborne lidar point clouds using multiscale local geometric descriptors and visual analytics. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2320–2335. [[CrossRef](#)]
- Mallet, C.; Bretar, F.; Roux, M.; Soergel, U.; Heipke, C. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 71–84. [[CrossRef](#)]
- Weinmann, M.; Jutzi, B.; Mallet, C. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry. Remote Sens. Spat. Inf. Sci.* **2013**, *5*, 1.
- Dittrich, A.; Weinmann, M.; Hinz, S. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 195–208. [[CrossRef](#)]
- Thomas, H.; Goulette, F.; Deschaud, J.E.; Marcotegui, B.; LeGall, Y. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In Proceedings of the 2018 International Conference on 3D vision (3DV), Verona, Italy, 5–8 September 2018; pp. 390–398.
- Du, J.; Jiang, Z.; Huang, S.; Wang, Z.; Su, J.; Su, S.; Wu, Y.; Ca, G. Point Cloud Semantic Segmentation Network Based on Multi-Scale Feature Fusion. *Sensors* **2021**, *21*, 1625. [[CrossRef](#)]
- Kumar, S.; Raval, S.; Banerjee, B. A robust approach to identify roof bolts in 3D point cloud data captured from a mobile laser scanner. *Int. J. Min. Sci. Technol.* **2021**, *31*, 303–312. [[CrossRef](#)]
- Demantké, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality based scale selection in 3D lidar point clouds. In Proceedings of the Laserscanning 2011, Calgary, AB, Canada, 29–31 August 2011.
- Shannon, C. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
- Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [[CrossRef](#)]
- Mitra, N.J.; Nguyen, A. Estimating surface normals in noisy point cloud data. In Proceedings of the Nineteenth Annual Symposium on Computational Geometry, San Diego, CA, USA, 8–10 June 2003; pp. 322–328.
- Blomley, R.; Weinmann, M.; Leitloff, J.; Jutzi, B. Shape distribution features for point cloud analysis—a geometric histogram approach on multiple scales. *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2014**, *2*, 9. [[CrossRef](#)]
- Gadat, S.; Younes, L. A stochastic algorithm for feature selection in pattern recognition. *J. Mach. Learn. Res.* **2007**, *8*, 509–547.
- Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
- Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* **1996**, *58*, 267–288. [[CrossRef](#)]
- Bommert, A.; Sun, X.; Bischl, B.; Rahnenführer, J.; Lang, M. Benchmark for filter methods for feature selection in high-dimensional classification data. *Comput. Stat. Data Anal.* **2020**, *143*, 106839. [[CrossRef](#)]
- Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [[CrossRef](#)]
- de la Fuente, M.O.; Cabo, C.; Ordóñez, C.; Roca-Pardiñas, J. Optimum Scale Selection for 3D Point Cloud Classification through Distance Correlation. In *International Workshop on Functional and Operatorial Statistics (IWFOS)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–220.

27. Székely, G.J.; Rizzo, M.L.; Bakirov, N.K. Measuring and testing dependence by correlation of distances. *Ann. Stat.* **2007**, *35*, 2769–2794. [[CrossRef](#)]
28. Székely, G.J.; Rizzo, M.L. Partial distance correlation with methods for dissimilarities. *Ann. Stat.* **2014**, *42*, 2382–2412. [[CrossRef](#)]
29. Berrendero, J.R.; Cuevas, A.; Torrecilla, J.L. Variable selection in functional data classification: A maxima-hunting proposal. *Stat. Sin.* **2016**, *26*, 619–638.
30. Munoz, D.; Bagnell, J.A.; V.; apel, N.; Hebert, M. Contextual classification with functional max-margin markov networks. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 975–982.
31. Ordóñez, C.; Cabo, C.; Sanz-Ablanedo, E. Automatic detection and classification of pole-like objects for urban cartography using mobile laser scanning data. *Sensors* **2017**, *17*, 1465. [[CrossRef](#)]
32. Febrero Bande, M.; Oviedo de la Fuente, M. Statistical Computing in Functional Data Analysis: The R Package fda.usc. *J. Stat. Softw.* **2012**, *51*, 1–28. [[CrossRef](#)]
33. Ripley, B.D. *Pattern Recognition and Neural Networks*; Cambridge University Press: Cambridge, UK, 2007.
34. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [[CrossRef](#)]
35. Aguilera-Morillo, M.C.; Aguilera, A.M. Multi-class classification of biomechanical data: A functional LDA approach based on multi-class penalized functional PLS. *Stat. Model.* **2020**, *20*, 592–616. [[CrossRef](#)]
36. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B Methodol.* **1958**, *20*, 215–232. [[CrossRef](#)]
37. Zelen, M. Multinomial response models. *Comput. Stat. Data Anal.* **1991**, *12*, 249–254. [[CrossRef](#)]
38. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
39. Crammer, K.; Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.* **2001**, *2*, 265–292.
40. Venables, W.N.; Ripley, B.D. *Modern Applied Statistics with S*; Springer: Berlin/Heidelberg, Germany, 2002.
41. Hand, D.J. Classifier technology and the illusion of progress. *Stat. Sci.* **2006**, *21*, 1–14. [[CrossRef](#)]