Enhancing 3D Point Cloud Classification with Measure-Theoretic Features: Integrating Product Coefficients into LiDAR Analysis

Patricia Medina, Ph.D.
Mathematical Sciences Department
New York City College of Technology
Brooklyn, NY 11201

PMedina@citytech.cuny.edu

Ninth International Conference on Data Management, Analytics and Innovation– ICDMAI 2025, Kolkata, India

January 18th, 2025

Introduction: The data

What is LiDAR?

LiDAR stands for light detection and ranging and it is an optical remote sensing technique that uses laser light to densely sample the surface of the earth, producing highly accurate x, y and z measurements. The collection vehicle of LiDAR data might be and aircraft, helicopter, vehicle, and tripod.

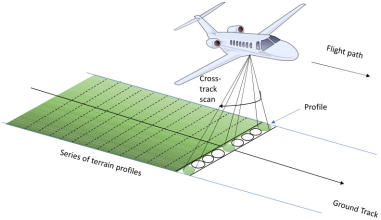
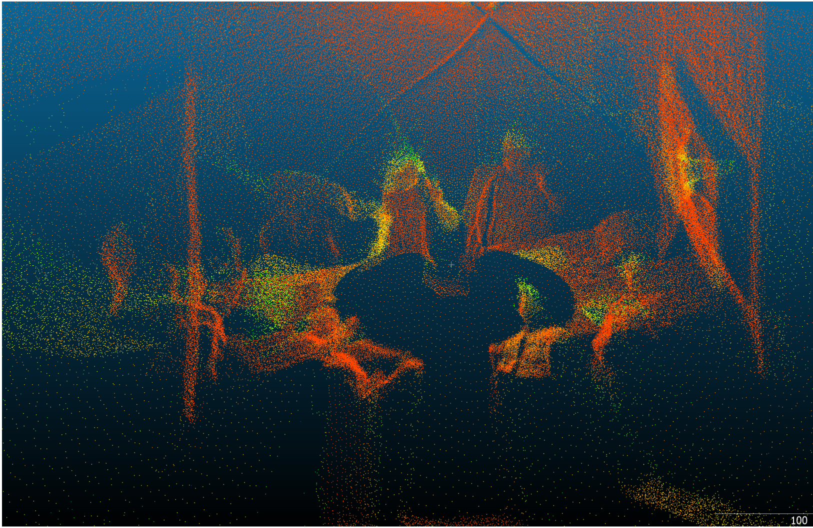


Figure: The profile belonging to a series of terrain profiles is measured in the cross track direction of an airborne platform.

Using a LiDAR Tripod at UW (eScience Institute) with my team at Geohack week, 2018



Outline

- We address the enhancement of classification accuracy for 3D point cloud Lidar data, an optical remote sensing technique that estimates the three-dimensional coordinates of a given terrain.
- Our approach introduces product coefficients, theoretical quantities derived from measure theory, as additional features in the classification process.
- We define and present the formulation of these product coefficients and conduct a comparative study, using them alongside principal component analysis (PCA) as feature inputs.
- Results demonstrate that incorporating product coefficients into the feature set significantly improves classification accuracy within this new framework.

3D point cloud LiDAR Data

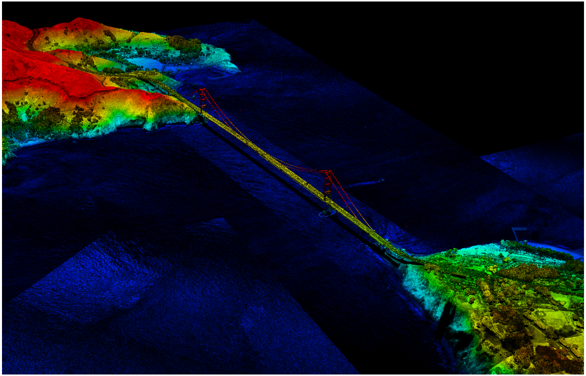


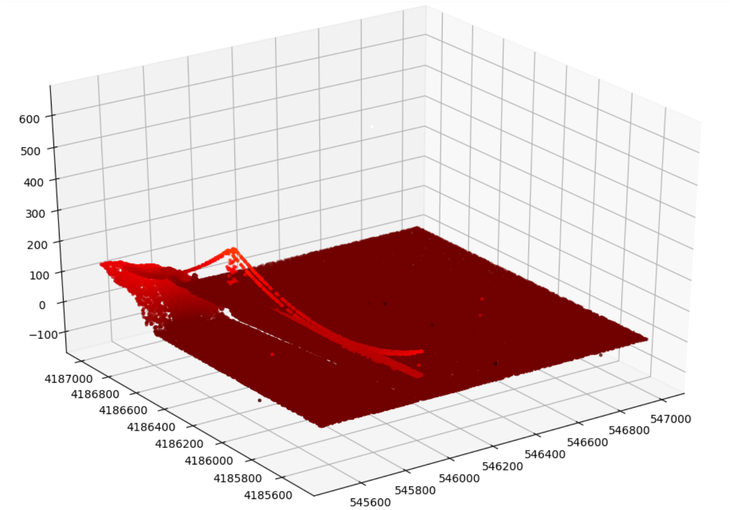
Figure: 3D LiDAR Point Cloud Image of San Francisco Bay and Golden Gate Bridge in California, Courtesy of Jason Stoker, USGS

Goal:

To classify classes such as ground, water, and the bridge structure.

Scatter plot. About 15 million data points

Enhancing 3D Point Cloud Classification with Measure-Theoretic Features: Integrating Product Coefficients into LiDAR Analysis



Attributes

- Intensity. Captured by the LiDAR sensors is the intensity of each return.
- Return number. An emitted laser pulse can have up to five returns depending on the features it is reflected from.
- Number of returns. The number of returns is the total number of returns for a given pulse.
- Point classification. Every LiDAR point that is post-processed can have a classification that defines the type of object that has reflected the laser pulse. The different classes are defined using numeric integer codes in the LAS files.
- Edge of flight line. Points flagged at the edge of the flight line will be given a value of 1, and all other points will be given a value of 0.
- RGB. LiDAR data can be attributed with RGB (red, green, and blue) bands.
- GPS time. The GPS time stamp at which the laser point was emitted from the aircraft. The time is in GPS seconds of the week.
- Scan angle. The scan angle is a value in degrees between -90 and +90.
- Scan direction. The scan direction is the direction the laser scanning mirror was traveling at the time of the output laser pulse.

Attribute example: Number of Returns

Enhancing 3D Point Cloud Classification with Measure-Theoretic Features: Integrating Product Coefficients into LiDAR Analysis

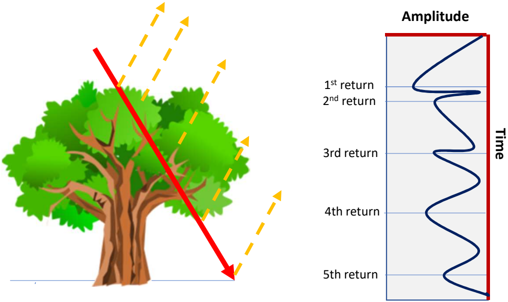


Figure: A pulse can be reflected off a tree's trunk, branches, and foliage as well as reflected off the ground. Karamatou Yacoubou Djima, F. Patricia Medina, Linda Ness and Melanie Weber, *Heuristic Framework for Multi-Scale Testing of the Multi-Manifold Hypothesis* , Accepted, AWM Springer Series.

Classification meaning and value

Classification value	Meaning
0	Never classified
1	Unassigned
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation
6	Building
7	Low Point
8	Reserved
9	Water

The Data

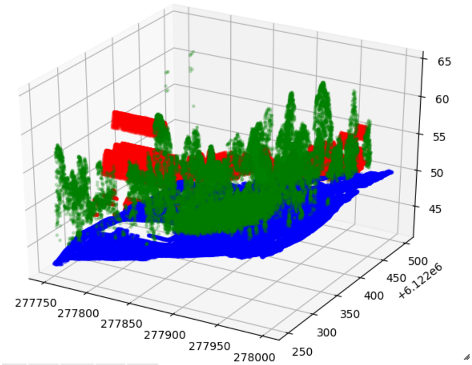


Figure: Python plot representing LiDAR point cloud of a neighborhood in Australia (about 270,000 points). Exact coordinates are available. Just three classes for now.

One way we did feature engineering before. [F. P. Medina and R. Paffenroth, “Classification frameworks comparison on 3D point clouds”, 2021 IEEE High Performance Extreme

We illustrate how to obtain the second row of the neighbor matrix in Fig. 5.

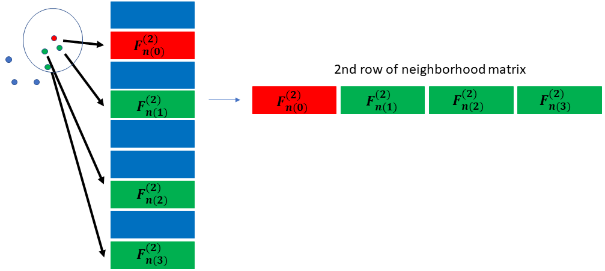
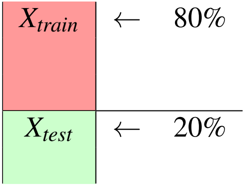


Figure: Forming the second row by concatenating the features of of the 3 nearest neighbors to the the second example in the original data frame. The neighbors are computed respect to the spatial coordinates (x, y, z) of the design point. Observe that if the original data has $N = 7$ features, the neighbor matrix has $(3+1) \times 7 = 28$ features.

Another idea to do feature engineering before applying typical ML algorithm

- The original features include: x, y, z,, intensity, RGB, number of returns and **and we can add new features (feature engineering)**
- We store the vector containing the classification given by the software (e.g. LASTool)
- Choose 80% for testing and 20% for training:



We also store actual classification value y and prediction $\hat{y}$

- We can use <u>product coefficients</u> to generate new features from the original data

An example of a general ML classification framework

1. Feature engineering
2. Perform dimensionality reduction using either PCA (for a linear projection) or a 3-layer auto-encoder (for a non-linear projection)
   - If using PCA, then use the projected features as the predictors for our learning
   - If using an auto-encoder, then use the hidden layer as the predictors for our learning
3. Classifier: K-nearest neighbor, random forest, <u>feed-forward neural network</u>
4. Cross-validation (f1 scores)

The metric that we use to measure precision of our algorithm is given by

$$PRE_{micro} = \frac{\sum_{j=1}^{N} TP_j}{\sum_{j=1}^{N} TP_j + \sum_{j=1}^{N} FP_j}, \tag{1}$$

(known as micro average) where TP_i means true positive on the *ith* class and FP_i means false positive on the *ith* class.
We provide the

$$F_1 \text{ score } = 2\frac{PRE_{micro} \cdot Recall}{PRE_{micro} + Recall}, \tag{2}$$

where the recall (or sensitivity) is given by

$$Recall = \frac{\sum_{j=1}^{N} TP_j}{\sum_{j=1}^{N} TP_j + \sum_{j=1}^{N} FN_j}, \tag{3}$$

where FN_j means false negative on the *jth* class.

Dimensionality reduction: PCA

The low dimensional data representation is obtained by mapping the data via M, i.e.

$$Z = XM.$$

PCA solves the eigen-problem $cov(X)M = \lambda M$.
$cov(X)$: sample covariance matrix of X. The principal components $\phi_1, \phi_2, \ldots, \phi_d$ are the ordered sequence of eigenvectors of $cov(X)$, and the variances of the components are the eigenvalues.
M is the matrix with columns ϕ_i, $i = 1, \ldots d$.

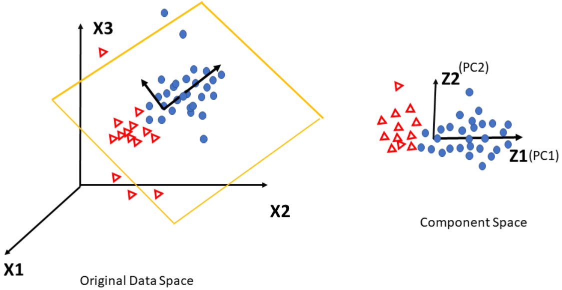
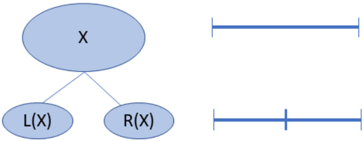


Figure: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \phi_{31}X_3$ and $Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \phi_{32}X_3$

Feature engineering with product coefficients



μ: non-negative measure on X; dy: the naive measure, such that $dy(X) = 1$

$$dy(L(S)) = \frac{1}{2}dy(S), \quad dy(R(S)) = \frac{1}{2}dy(S)$$

μ is additive in the binary set system,

μ be a dyadic measure on a dyadic set X and S be a subset of X. The *product coefficient* parameter a_S is the solution for the following system of equations

$$\mu(L(S)) = \frac{1}{2}(1 + a_s)\mu(S) \tag{4}$$

$$\mu(R(S)) = \frac{1}{2}(1 - a_s)\mu(S) \tag{5}$$

Dyadic Product Formula Representation

X with binary set system B whose non-leaf sets are B_n

$\mu = \mu(X)\prod_{S \in B_n}(1 + a_S h_S)\, dy$

where $a_S \in [-1, 1]$

h_S : Haar-like function

R. FEFFERMAN, C. KENIG, AND J. PIPHER, The theory of weights and the Dirichlet problem for elliptical equations, Annals of Math., 134 (1991), pp. 65–124

Lemma 1 (Dyadic Product Formula Representation)

Let X be a dyadic set with binary set system B whose non-leaf sets are B_n.

1. *A non-negative measure μ on X has a unique product formula representation*

$$\mu = \mu(X) \prod_{S \in B_n} (1 + a_S h_S)\, dy \tag{6}$$

where $a_S \in [-1,1]$ and a_S is the product coefficient for S.

2. *Any assignment of parameters a_S for $(-1,1)$ and choice of $\mu(X) > 0$ determines a measure μ which is positive on all sets S on B with product formula $\mu = \mu(X) \prod_{S \in B_n} (1 + a_S h_S)\, dy$ whose product coefficients are the parameters a_S.*

3. *Any assignment of parameters a_S from $[-1,1]$ and choice of $\mu(X) > 0$ determines a non-negative measure μ with product formula $\mu = \mu(X) \prod_{S \in B_n} (1 + a_S h_S)\, dy$. The parameters are the product coefficients if they satisfy the constraints:*
   1. *If $a_S = 1$, then the product coefficient for the tree rooted at $R(S)$ equals 0.*
   2. *If $a_S = -1$, then the product coefficient for the tree rooted at $L(S)$ equals 0.*

Example 2 (Formula for a scale 0 dyadic measure)

Let $X = [0,1]$ and let there be a non-negative measure μ such that $\mu(X) = 1$, $\mu(L(X)) = \frac{1}{4}$ and $\mu(R(X)) = \frac{3}{4}$. Let $a = a_X$ be the product coefficient which is the solution for the system of equations

$$\mu(L(X)) = \frac{1}{2}(1+a)\mu(X) \tag{7}$$

$$\mu(R(X)) = \frac{1}{2}(1-a)\mu(X). \tag{8}$$

Subtracting (8) from (7) we obtain $a = \dfrac{\mu(L(X)) - \mu(R(X))}{\mu(X)} = -\dfrac{1}{2}$.

Since, $dy(X) = 1$ and $dy(L(X)) = \frac{1}{2} = dy(R(X))$ then by the product formula form,

$$\mu = \mu(X)(1+ah)dy, \tag{9}$$

where h is the Haar-like function with $S = X$.

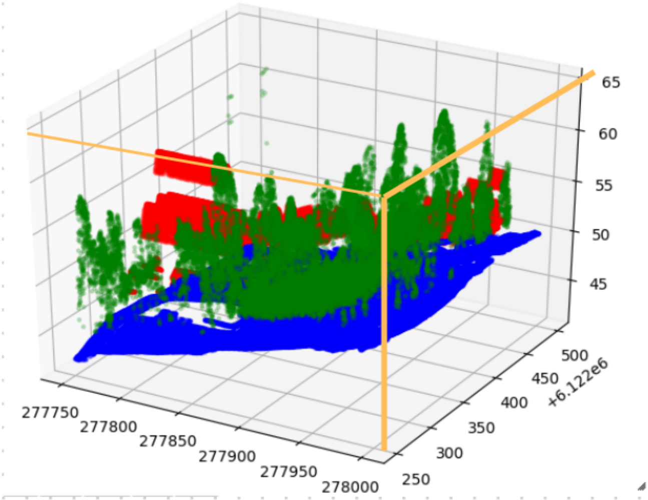
Past work (another Lidar dataset)

- Applying the product formula representation to a counting measure derived from a set of LIDAR sample data (*). This data consists of ten sets of discrete points in 3-dimensional space, representing the surfaces visible to the scanning laser rangefinder in ten nearby scenes. Each point has been labeled as either “vegetation” or “ground”.
- Previous work had examined this same data using a multi-scale SVD approach to build a support vector machine (SVM) based classification rule that could, with high accuracy, reproduce the vegetation/ground labeling. Herein, product coefficient parameters are used as features instead of multi-scale SVD parameters.
- The experiment showed that decision rules for distinguishing two measures (here “vegetation and “ground”) could be approximately inferred from histograms of the product coefficients. While the metrics were not as good as for multi-scale SVD, the method did provide a transparent rationale for the decision rule.

(*) Brodu, N., & Lague, D., 3D terrestrial LiDAR data classication of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 16, pp. 121-134, 2012.

How to compute product coefficients for Lidar?

Put the 3D point cloud in the unit cube and use the counting measure...



Binary tree

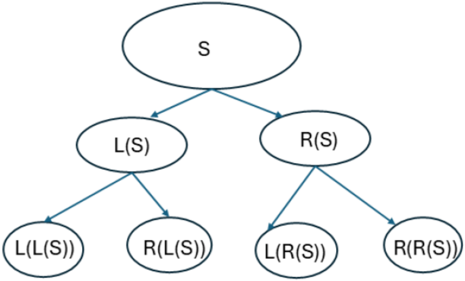


Fig. 5 The diagram represents the binary tree for a general set S. The diagram includes the notation used in the computation of the seven product coefficients. There are $2^0, 2^1$ and 2^2 product coefficients per level.

Computing product coefficients on points inside spheres

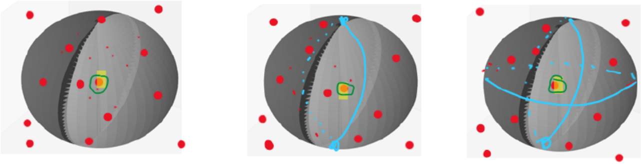


Fig. 6 When computing product coefficients per point (x_i, y_i, z_i), consider a sphere $S + i$of radius 2. We first slice the sphere in along the x-axis and compute the first product coefficient a_S, then slice the sphere along the $y-$ axis and compute two product coefficients. One for the left child $L(S_i)$ and the other for the right child $R(S_i)$. Last, we slice along the $z-$ axis and compute the the last four product coefficients.

F1 scores: Original data vs. Adding product coefficients

	KNN F^1-score	RF F^1-score
Original features $(x.y,z)$	0.33 ($\pm$ 0.18)	0.41 ($\pm$ 0.16)
With PCs	0.33 ($\pm$ 0.18)	0.45 ($\pm$ 0.15)

Table 2 cross validation on F_1- scores using KNN and random forest (RF) on the original data with only x, y, and z spatial coordinates as features and the data with the new seven product coefficient generated features

F1 scores: Performing PCA after adding product coefficient features

# of Principal Components	KNN F^1-score	RF F^1-score
3	0.39 ($\pm$ 0.12)	0.38 ($\pm$ 0.12)
4	0.45 ($\pm$ 0.10)	0.39 ($\pm$ 0.10)
5	0.48 ($\pm$ 0.11)	0.38 ($\pm$ 0.16)
6	0.56 ($\pm$ 0.08)	0.44 ($\pm$ 0.15)
7	0.61 ($\pm$ 0.06)	0.49 ($\pm$ 0.14)
8	0.65 ($\pm$ 0.05)	0.55 ($\pm$ 0.14)
9	0.67 ($\pm$ 0.04)	0.52 ($\pm$ 0.14)
10	0.85 ($\pm$ 0.02)	0.81 ($\pm$ 0.16)

Table 1 F^1 scores using PCA on the data with the new seven product coefficient generated features. Cross validation on F_1- scores using $n = 3, \ldots, 10$ principal components.

Future Directions

- considering more physical features from the original Lidar and computing product coefficients on neighborhood sets of dimension $n \geq 3$.
- considering bigger size data sets to test the robustness of this new framework.
- engineering more features by computing intrisic dimension for example of the dataset and perform more comparison experiments.
- performing dimensionality reduction with an auto-encoder and comparing this new framework with the one including PCA.
- studying other covariance reduction techniques and compare with PCA performance.

References

1. https://www.esri.com/en-us/arcgis/products/.
2. D. Bassu, R. Izmailov, A. McIntosh, L. Ness, and D. Shallcross, *Application of multi-scale singular vector decomposition to vessel classification in overhead satellite imagery*, in Proceedings Volume 9631: Seventh Annual International Conference on Digital Image Processing ICDIP 2015, C. Falco and X. Jiang, eds., 2015.
3. N. Brodu and D. Lague, *3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology*, ISPRS Journal of Photogrammetry and Remote Sensing, 68 (2012), pp. 121 – 134.
4. R. A. Fefferman, C. E. Kenig, and J. Pipher, *The theory of weights and the dirichlet problem for elliptic equations*, Annals of Mathematics, 134 (1991), pp. 65–124.
5. P. M. Mather, *Computer Processing of Remotely-Sensed Images: An Introduction*, John Wiley & Sons, Inc., USA, 2004.
6. F. P. Medina, L. Ness, M. Weber, and K. Y. Djima, *Heuristic framework for multiscale testing of the multi-manifold hypothesis*, in Research in Data Science, Springer, 2019, pp. 47–80.
7. L. Ness, *Dyadic product formula representations of confidence measures and decision rules for dyadic data set samples*, 08 2016, pp. 1–8.
8. G. Petrie and C. Toth, *Introduction to Laser Ranging, Profiling, and Scanning*, 11 2008, pp. 1–28.
9. G. Zheng and L. Moskal, *Retrieving leaf area index (lai) using remote sensing: Theories, methods and sensors*, Sensors (Basel, Switzerland), 9 (2009), pp. 2719–45.


Thank you!!!

Manifold Hypothesis (related to intrinsic dimension)

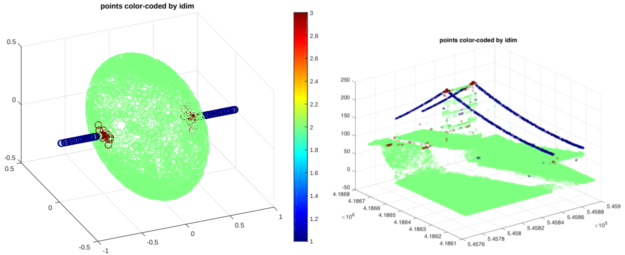


Figure: Intrinsic dimension of the line-sphere sample and the LiDAR data set from the Golden Gate bridge. Karamatou Yacoubou Djima, F. Patricia Medina, Linda Ness and Melanie Weber, *Heuristic Framework for Multi-Scale Testing of the Multi-Manifold Hypothesis* ,Research in Data Science, AWM Springer-Verlag Series.

Proposition (Multi-manifold Hypothesis Test)

Given a data set $X = \{x_i\}_{i \in I}$ in $\mathbb{R}^D$ and a multi-manifold $\mathscr{V}$, is the expected distance of the points in X to $\mathscr{V}$ more than one would expect? If so, reject $\mathscr{V}$ as being a multi-manifold that fits X.

Theoretical framework: C. FEFFERMAN, S. MITTER, AND H. NARAYANAN, Testing the manifold hypothesis, J. Amer. Math. Soc., 29 (2016), pp. 983–1049