Machine Learning in LiDAR 3D point clouds

F. Patricia Medina
Computer Science Department
Yeshiva University
New York, NY 10033

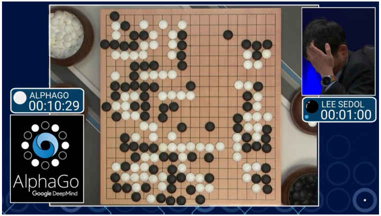
patricia.medina@yu.edu

REU talk
University of Rochester

August 13, 2020

Machine Learning → Deep Learning

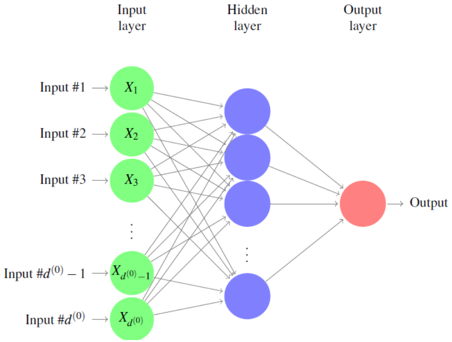
https://www.digitaltrends.com/cars/toyota-robot-car-extreme-tests-california/



https://www.shellypalmer.com/2016/03/alphago-vs-not-fair-fight/

- Machine learning and algorithms are currently having a direct impact on many aspects of our lives
- Lack of mathematical understanding

Neural Networks

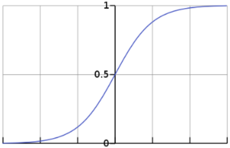


Input layer: vector $X = \left(X_1, X_2, X_3, \ldots, X_{d^{(0)}}\right) \in \mathbb{R}^{d^{(0)}}$ (the original input vector is augmented with $X_0 = 1$)

- Find optimal weights
- Every node has a transformation function θ.
- From layer $l-1$ to layer l we have a weight matrix $W^{(l)}$ (weights in) of size $d^{(l-1)} \times d^{(l)}$, and the matrix $W^{(l+1)}$ (weights out) of dimension $d^{(l)} \times d^{(l+1)}$. We
- Put all weight matrices together. Weight parameter: $\mathbf{w} = \{W^{(1)}, W^{(2)}, \ldots W^{(L)}\}$.
- Approximation to target function $h_{\mathbf{w}}(X)$

Neural Networks

- Derived features (S_m) (hidden units) are created from linear combinations of the inputs (X_i), and the targets (Y_k) are modeled as functions of linear combinations (S_m)
- The activation function θ is the sigmoid σ given by $\sigma(u) = \dfrac{1}{1+e^{-u}}$



- To find the weight in $\mathbf{w}$, it is common to use the *batch gradient descent* algorithm

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.

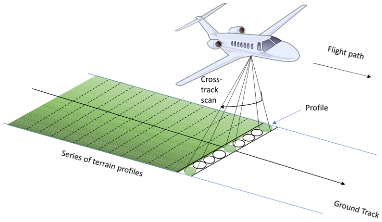


Figure: The profile belonging to a series of terrain profiles is measured in the cross track direction of an airborne platform.

3D point cloud LiDAR Data

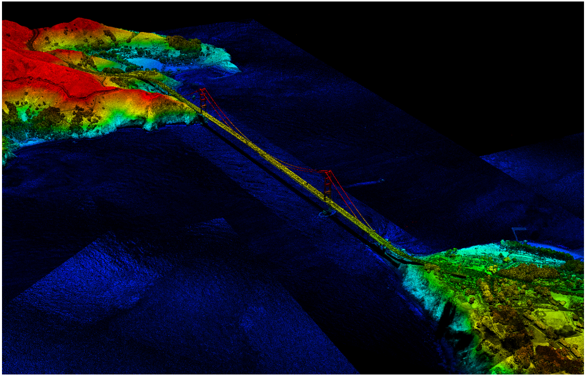


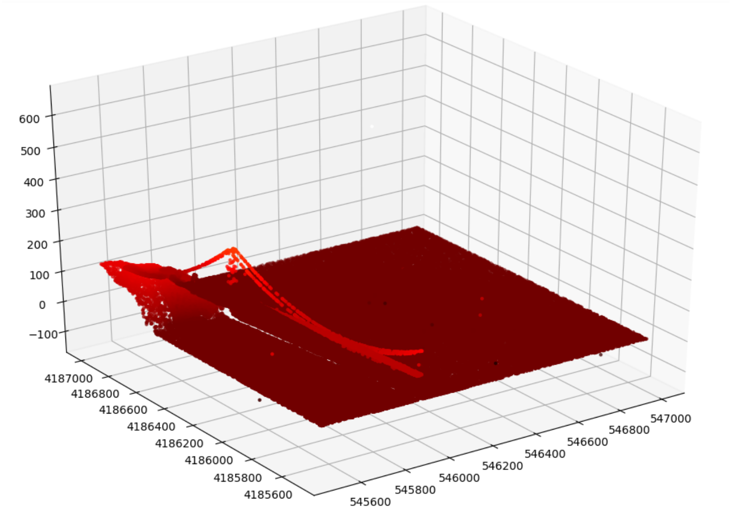
Figure: 3D LiDAR Point Cloud Image of San Francisco Bay and Golden Gate Bridge in California, Courtesy of Jason Stoker, USGS

Goal:

To classify ground, water, and the bridge structure.

Scatter plot. About 15 million data points

Machine Learning in LiDAR 3D point clouds.



Attributes/Features

- Intensity. Captured by the LiDAR sensors is the intensity of each return.
- 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

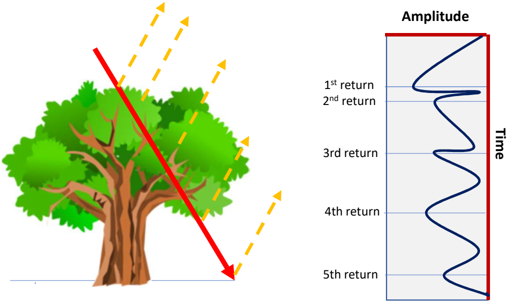


Figure: A pulse can be reflected off a tree's trunk, branches, and foliage as well as reflected off the ground. Karamatou Yacoubou Djima, F. Patricia Medina, Linda Ness and Melanie Weber, *Heuristic Framework for Multi-Scale Testing of the Multi-Manifold Hypothesis* , AWM Springer Series.

Classification meaning and value

0	Never classified	
1	Unassigned	
2	Ground	$\longleftarrow$
3	Low vegetation	
4	Medium vegetation	
5	High vegetation	
6	Building	
7	Noise	$\longleftarrow$
8	Model key/ Reserved	
9	Water	$\longleftarrow$
10	Rail	$\longleftarrow$
11	Road surface	
$\vdots$	$\vdots$	
17	Bridge deck	$\longleftarrow$
18	High noise	$\longleftarrow$

Main outline of experiments

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)

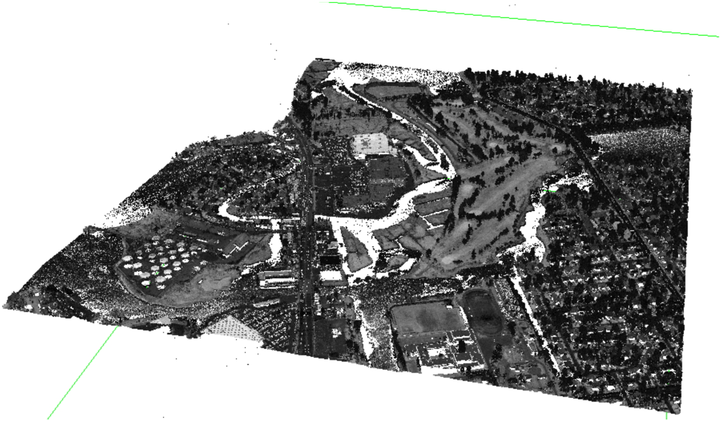


Figure: 3D LiDAR point cloud graphed by intensity for a location close to the JFK airport, NY.

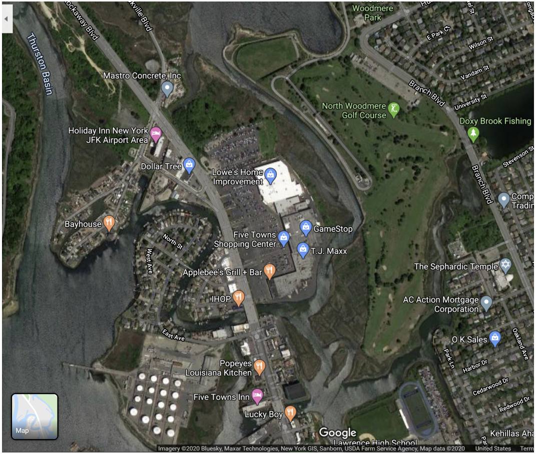


Figure: Google map satellite image of the location of associated to the 3D point cloud in the JFK airport, NY. Coordinates: $40^\circ 38' 38.6" N 73^\circ 44' 46.9" W$ Rockaway Blvd, Rosedale, NY 11422 See Fig.4

Feature engineering: neighbor matrix construction

For each LiDAR data point (example) we consider k nearest neighbors based on spatial coordinates (x_i, y_i, z_i) and create a new example which is in higher dimensions.
More precisely, let $F^{(i)}_{n(0)}$ the set of N features associated to the ith example (the first three features are spatial.) Now let $F^{(i)}_{n(j)}$ the set of N features associated to the jth nearest neighbor to the ith example. We end up with set of set of features associated to the ith example:

$$F^{(i)}_{n(0)}, F^{(i)}_{n(1)}, \dots, F^{(i)}_{n(k)},$$

where $i = 1, \dots, s$. Here $F^{(i)}_{n(j)} \in \mathbb{R}^{1\times N}$ for each $j = 1, \dots, k$.
We concatenate the features

$$\left[\begin{array}{c|c|c|c} F^{(i)}_{n(0)} & F^{(i)}_{n(1)} & \dots & F^{(i)}_{n(k)} \end{array}\right] \in \mathbb{R}^{1\times(k+1)\cdot N}$$

for each $i = 1, \dots, s$.

Neighbor matrix

We then put all the rows together and get what we call the *neighbor matrix*

$$\left[\begin{array}{c|c|c|c} F_{n(0)}^{(1)} & F_{n(1)}^{(1)} & \dots & F_{n(k)}^{(1)} \\ F_{n(0)}^{(2)} & F_{n(1)}^{(2)} & \dots & F_{n(k)}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n(0)}^{(s)} & F_{n(1)}^{(s)} & \dots & F_{n(s)}^{(1)} \end{array}\right] \in \mathbb{R}^{s\times(k+1)\cdot N}$$

We illustrate how to obtain the second row of the neighbor matrix in Fig. 6.

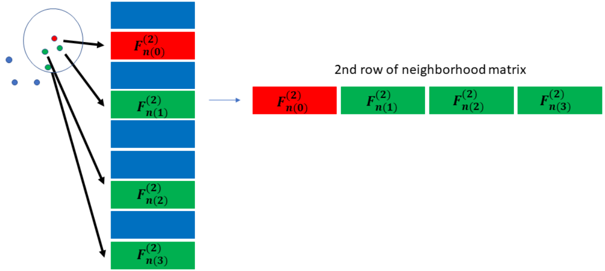
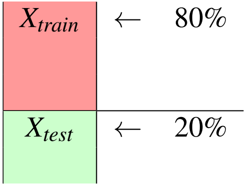


Figure: Forming the second row by concatenating the features of of the 3 nearest neighbors to the the second example in the original data frame. The neighbors are computed respect to the spatial coordinates (x, y, z) of the design point. Observe that if the original data has $N = 7$ features, the neighbor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

- The original features include: x, y, z,, intensity, number of returns and at most the new features $(X_1 = x, X_2 = y, X_3 = z, X_4 = F_1, \dots, X_{10} = F_7)$
- We store the vector containing the classification given by the software (e.g. LASTool)
- Construct the "neighbor matrix". Find 10 nearest neighbors for each (x, y, z). One row of the neighbor matrix is a concatenation of $x, y, z, F_1, F_2, \dots, F_7$ and its nearest neighbors with their corresponding features
- Choose 80% for testing and 20% for training:



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

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, $i = 1, \ldots d$.

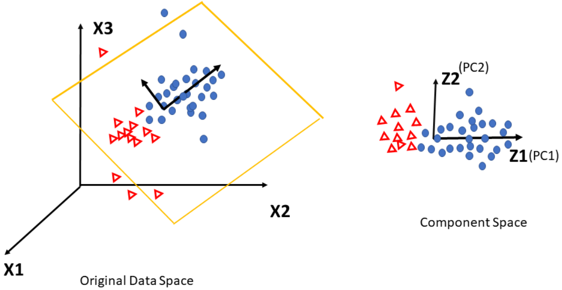
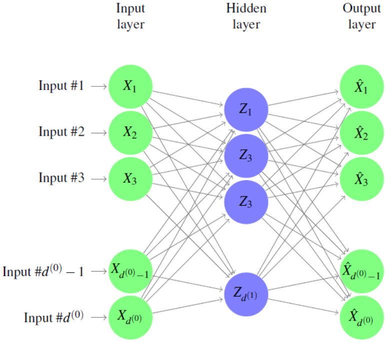


Figure: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \phi_{31}X_3$ and $Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \phi_{32}X_3$

Dimensionality reduction: Auto-encoders



$$\underset{W^{(1)},b_1,c_1}{\operatorname{argmin}} \; \|X - f^+(f(X))\|_2^2 = \underset{W^{(1)},b_1,c_1}{\operatorname{argmin}} \; \|X - \hat{X}\|_2^2, \qquad (1)$$

where $W^{(1)}$ are the weights, and b_1 and c_1 are coming from the bias term. They determine the non-linear maps f (encoder) and f^+ (decoder). Note that the nonlinearity comes from the introduction of some activation function θ :

$$f(X) = \theta\left(W^{(1)}X + b_1\right), \quad f^+(Z) = \theta\left(\left(W^{(1)}\right)^T Z + c_1\right), \qquad (2)$$

The auto-encoder $h_{\mathbf{w}}(X) = f^+(f(X))$ "tries to learn" the identity function

$$\hat{X} = h_{\mathbf{w}}(X) \approx X. \qquad (3)$$

An auto-encoder is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.

Grim, A., Iskra, B., Ju, N., Kryshchenko, A., Medina, F.P., Ness, L. , Ngamini, M., Owen, M., Paffenroth, R., Tang, S. Representation of Data as Multi-Scale Features and Measures, To appear in AWM Series Springer Volume.

Dimensionality reduction: Auto-encoders

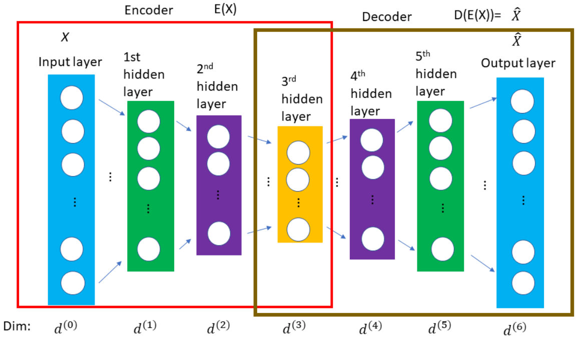


Figure: 5-layer auto–encoder diagram. The input layer has dimension $d^{(0)}$, the five inner layers have dimensions $d^{(1)}$, $d^{(2)}$, $d^{(3)}$, $d^{(4)}$ and $d^{(4)}$, respectively. The dimension of the outer layer $\hat{X}$ has dimension $d^{(6)} = d^{(0)}$ since this is an auto-encoder. The 5th hidden layer has dimension $d^{(5)} = d^{(1)}$ and the 4th hidden layer has dimension $d^{(4)} = d^{(2)}$. The 3rd layer is the most inner layer with dimension $d^(3)$ which is the reduced dimension we use in some of the frameworks for classification.

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

$$PRE_{micro} = \frac{\sum_{j=1}^{N} TP_j}{\sum_{j=1}^{N} TP_j + \sum_{j=1}^{N} FP_j}, \tag{4}$$

(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

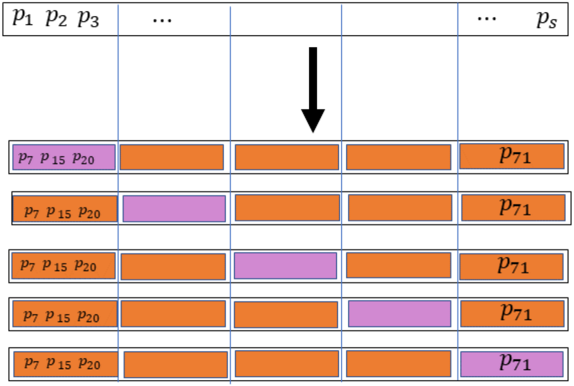
$$F_1 \text{ score } = 2\frac{PRE_{micro} \cdot Recall}{PRE_{micro} + Recall}, \tag{5}$$

where the recall (or sensitivity) is given by

$$Recall = \frac{\sum_{j=1}^{N} TP_j}{\sum_{j=1}^{N} TP_j + \sum_{j=1}^{N} FN_j}, \tag{6}$$

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

K-fold cross validation

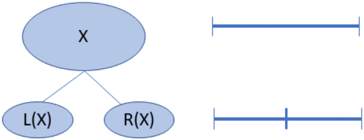


Some results

	KNN	RF	RF-Ens	NN
Raw	0.8670 (+/- 0.0004)	0.8701 (+/- 0.0007)	0.8564 (+/- 0.0019)	0.8241 (+/- 0.0018)
PCA	0.8399 (+/- 0.0002)	0.8384 (+/- 0.0010)	0.8212 (+/- 0.0011)	0.7791 (+/- 0.0069)
Enc	0.8223 (+/- 0.0004)	0.8160 (+/- 0.0003)	0.7902 (+/- 0.0041)	0.6331 (+/- 0.0110)
Neig+PCA	0.8291 (+/- 0.0032)	0.8445 (+/- 0.0029)	0.8361 (+/- 0.0031)	0.9748 (+/- 0.0042)
Neig+Enc	0.7366 (+/- 0.0045)	0.7816 (+/- 0.0044)	0.7700 (+/- 0.0049)	0.6770 (+/- 0.0059)
Neig	0.8303 (+/- 0.0025)	0.9497 (+/- 0.0101)	0.9499 (+/- 0.0118)	0.9792 (+/- 0.0044)

Figure: 5-fold cross validation of F_1 scores for different classification frameworks; number of classes=6; RAW+ Norm= Standardized and normalized raw data (includes pre-processing step) Enc= Encoder (using inner layer of auto encoder for dimension reduction); PCA and Enc have already been standardized and normalized. *Machine Learning in Lidar 3D point cloud*, accepted (Springer) with Randy Paffenroth (WPI)

Generating more features: 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

$$\mu(L(S)) = \frac{1}{2}(1+a_s)\mu(S) \qquad (7)$$

$$\mu(R(S)) = \frac{1}{2}(1-a_s)\mu(S) \qquad (8)$$

Dyadic Product Formula Representation

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

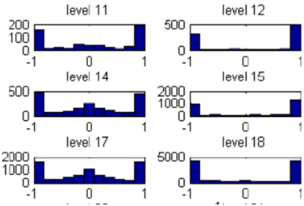
$\mu = \mu(X)\prod_{S\in B_n}(1+a_S h_S)\,dy$

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

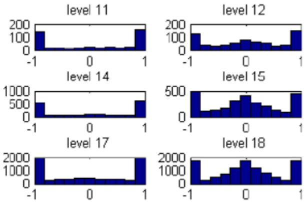
h_S : Haar-like function

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

Ground vs Vegetation

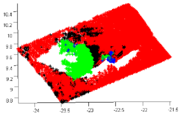


(a) Ground



(b) Vegetation

Figure: Histogram of coefficients by level for sets of ground points and vegetation points



D. Bassu, P. W. Jones, L. Ness, D. Shallcross, Product Formalisms for Measures on Spaces with Binary Tree Structures: Representation, Visualization, and Multiscale Noise, Mathematics in Data Science Workshop, 2015

Manifold hypothesis (related to intrinsic dimension)

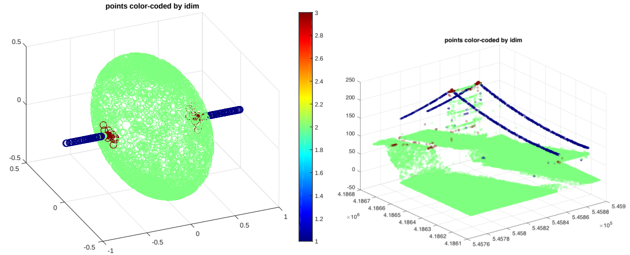


Figure: Intrinsic dimension of the line-sphere sample and the LiDAR data set from the Golden Gate bridge. Karamatou Yacoubou Djima, F. Patricia Medina, Linda Ness and Melanie Weber, *Heuristic Framework for Multi-Scale Testing of the Multi-Manifold Hypothesis* , Accepted,Research in Data Science, AWM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

Microsoft AI for Earth Grant (Azure Services): Climate Change

Differentiation of photosynthetic components (leaf, bushes or grasses) and non-photosynthetic components (branches or stems) by 3D terrestrial laser scanners (TLS) is of key importance to understanding the spatial distribution of the radiation regime, photosynthetic processes, and carbon and water exchanges of the forest canopy.
YU students collaborating: Tony Arriaza, Yudi Meltzer (IBM), Ezra Splaver

---

F. Patricia Medina, Mathematical Sciences Department, Worcester Polytechnic Institute (P.I.); Jonathan Batchelor, Remote Sensing and Geospatial Laboratory, School of Environment and Forestry Science, University of Washington (Co-P.I.); L. Monika Moskal, Remote Sensing and Geospatial Laboratory, School of Environment and Forestry Science, University of Washington (Co-P.I.); Randy Paffenroth, Mathematical Sciences Department, Computer Science Department, Data Science Program, Worcester Polytechnic Institute (Co-P.I.); Guang Zhen, Jiangso Provincial Key Laboratory of Geographic Information Science and Technology, International Institute of Earth System Science, Nanjing University, China (Co-P.I.)

Thanks!

Machine Learning in LiDAR 3D point clouds.