Machine Leaning on LiDAR 3D point clouds

Dr. F. Patricia Medina
Mathematics Department
New York City College of Technology
Brooklyn, NY

FMedina@citytech.cuny.edu

REU mentor prensentation

June 15, 2023

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.

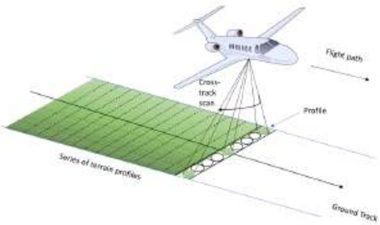


Figure: The profile belonging to a series of terrain profiles is measured in the cross track direction of an airborne platform.

3D point cloud LiDAR Data

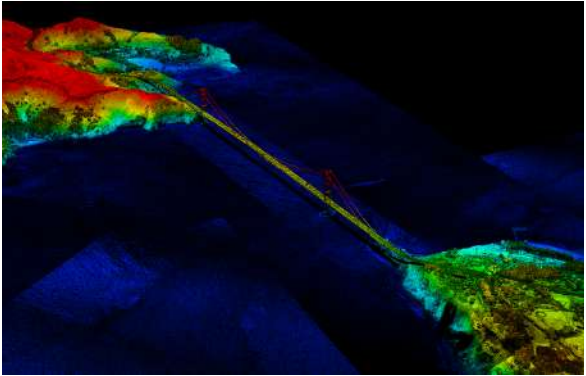
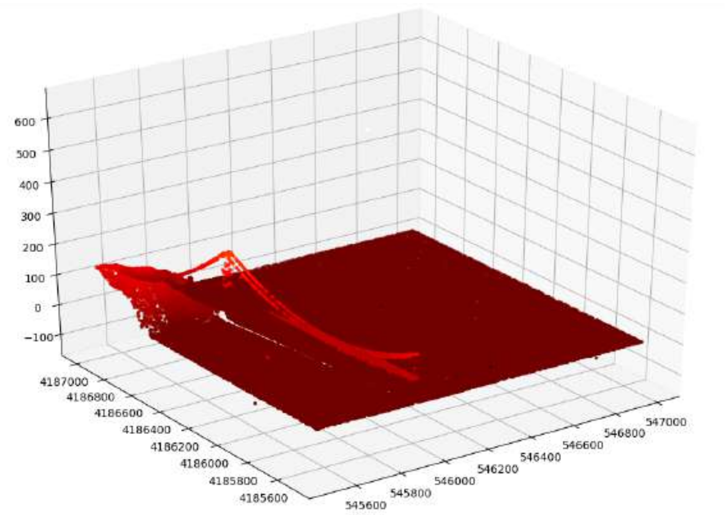


Figure: 3D LiDAR Point Cloud Image of San Francisco Bay and Golden Gate Bridge in California, Courtesy of Jason Stoker, USGS

Goal:

To classify ground, water, and the bridge structure.

Scatter plot. About 15 million data points



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

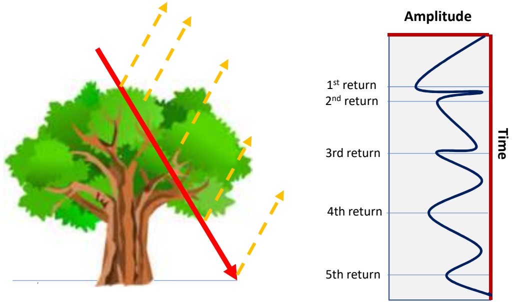


Figure: A pulse can be reflected off a tree's trunk, branches, and foliage as well as reflected off the ground. Karamatou Yacoubou Djima, F. Patricia Medina, Linda Ness and Melanie Weber, *Heuristic Framework for Multi-Scale Testing of the Multi-Manifold Hypothesis* , AWM Springer Series.

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	$\longleftarrow$

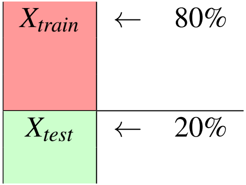
Main outline of experiments

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: $40^{\circ}38'38.6\text{''}N73^{\circ}44'46.9\text{''}W$ Rockaway Blvd, Rosedale, NY 11422 See Fig.4

- The original features include: x, y, z,, intensity, number of returns and at most the new features $(X_1 = x, X_2 = y, X_3 = z, X_4 = F_1, \ldots, X_{10} = F_7)$
- 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 $x, y, z, F_1, F_2, \ldots, F_7$ 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, $i = 1, \ldots d$.

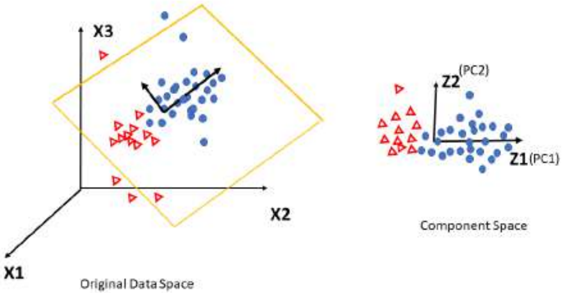
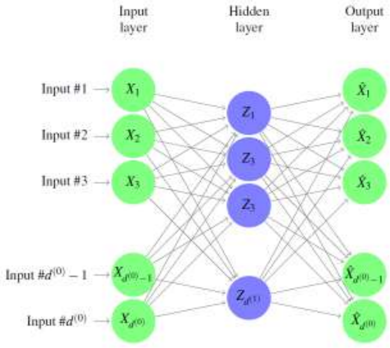


Figure: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \phi_{31}X_3$ and $Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \phi_{32}X_3$

Dimensionality reduction: Auto-encoders



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

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 :

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

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

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

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

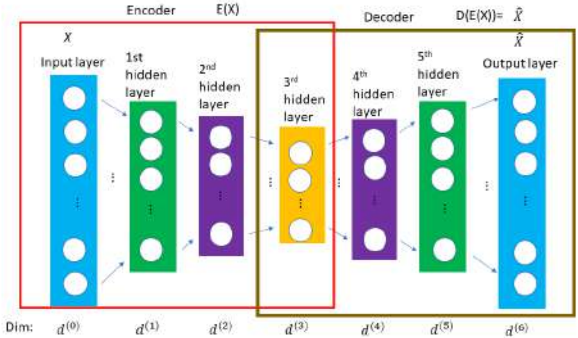


Figure: 5-layer auto–encoder diagram. The input layer has dimension $d^{(0)}$, the five inner layers have dimensions $d^{(1)}$, $d^{(2)}$,$d^{(3)}$,$d^{(4)}$ and $d^{(4)}$, respectively. The dimension of the outer layer $\hat{X}$ has dimension $d^{(6)} = d^{(0)}$ since this is an auto-encoder. The 5th 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

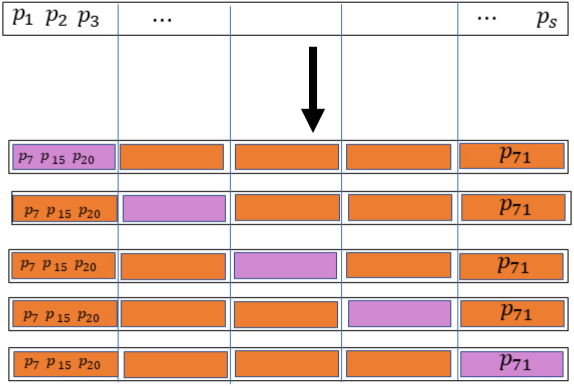
$$F_1 \text{ score } = 2\frac{PRE_{micro} \cdot Recall}{PRE_{micro} + Recall}, \tag{5}$$

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



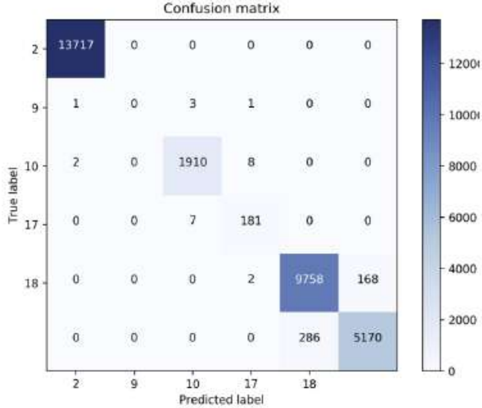
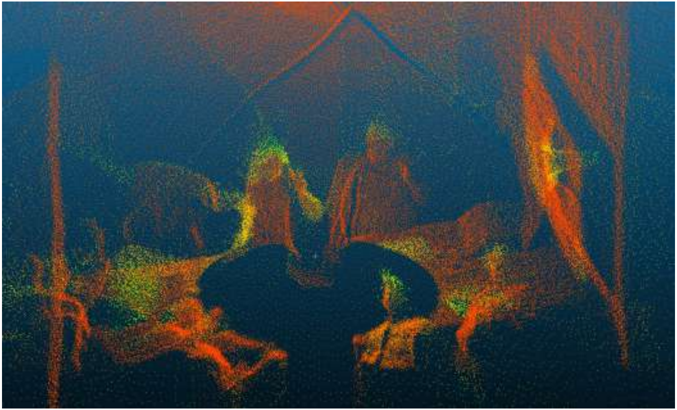


Fig. 12 Confusion matrix corresponding to feed-forward neural network classifier with the neighborhood matrix as input

Thanks!



Geohackweek, University of Washington 2018