Detecting Location in the San Francisco Bay from Measured Dissolved Oxygen Content

Natalie Kim Gable
Stanford University
ngable@stanford.edu

***Abstract*—Dissolved oxygen, how much oxygen is dissolved in water and available to aquatic fauna and flora, is indicative of marine ecosystem health and water quality. In this project, we use sensor data collected from 22 different locations around the San Francisco Bay. We first use supervised learning, on raw data and data transformed using the Discrete Fourier Transform and the Discrete Wavelet Transform, to predict the location of the sensor, based off the dissolved oxygen content. We next perform dimensionality reduction via PCA and Fisher's LDA and clustering, to understand different patterns in the sensor data. The supervised learning task leads to low prediction accuracy, but clustering methods identify key patterns in daily dissolved oxygen content.**

I. Introduction

Environmental engineers and ecologists measure dissolved oxygen content to keep track of ecosystem health and water quality. Places with little dissolved oxygen are called "dead zones" as they are unable to support marine life. Dissolved oxygen is tracked using sensors and there is an abundance of this sensor data from the San Francisco Bay Area. There have been historically few applications of machine learning approaches to this type of environmental data and we are interested to see if these approaches can be leveraged to help with environmental and ecosystem surveys.

II. Dataset

This dataset was provided by a friend and colleague at the San Francisco Estuary Institute. It contains dissolved oxygen sensor measurements from 22 different locations around the San Francisco Bay. Dissolved oxygen is measured in milligrams of oxygen per liter of water and measurements are taken every 15 minutes. In this project, we use data collected from January 1, 2018 to December 31, 2018. We divide the dataset into 24-hour chunks, leaving us with 96 measurements per chunk. Missing data is handled by the following process: if the entire day is missing data, drop the observation, otherwise replace the missing values with the day's average value.

After cleaning the data, we are left with 6372 days, each day being a vector in $\mathbb{R}^{96}$, measured at 22 different locations. The breakdown by location are shown in the following table.

For our supervised learning tasks, we randomly divide the dataset into an 80-20 train-test split. In Figure 1, we can see a random sample of 5 different days from our dataset.

Location	Count
Rio Vista at Decker Island	319
San Mateo Bridge	340
Mossdale	278
Mallard Island	275
Martinez	273
Grizzly Bay At Suisun Slough Near Avon	341
Delta Cross Channel	343
Sacramento River at Freeport	358
Sacramento River at Mallard Island	174
Pond A8 Outlet	347
Coyote Creek at Alviso Slough	363
Newark Slough	267
Cache Slough	240
Liberty Cut	349
Deep Water Shipping Channel	48
Cache Slough at Liberty Island	335
San Joaquin River at Jersey Point	323
Mowry Slough	208
Prisoner Point	272
Alviso Slough	323
Guadalupe Slough	236
Dumbarton Bridge	360

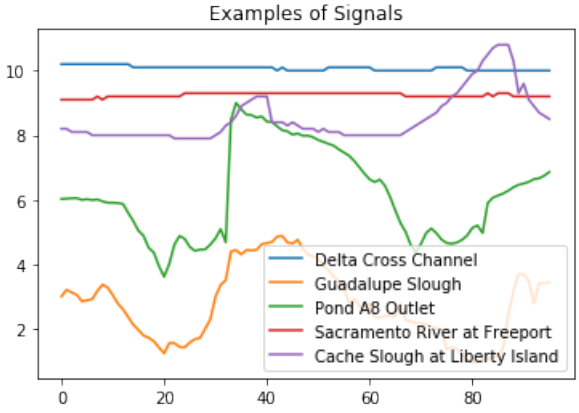

Fig. 1. Example signals from the dataset, randomly sampled.

III. Features and Transforms

For supervised learning, we use raw data as well as features generated from applying the Discrete Fourier Transform and the Discrete Wavelet Transform, as described below.

A. Discrete Fourier Transform

For these features, we pick out the magnitude of the DFT coefficients and shift them to be centered at zero frequency. We are curious to see if the different sites have distinct frequency spectra. Figure 2 shows samples of these DFT features. We

notice, in Figure 2, that there doesn't seem to be much difference in the frequency spectra between samples.

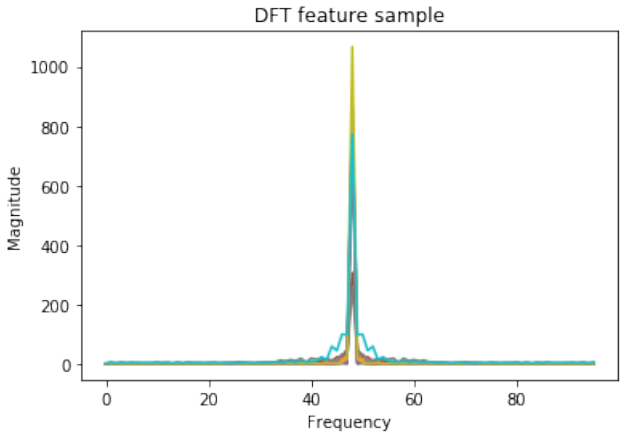

Fig. 2. Example DFT magnitudes from the dataset, randomly sampled.

B. Discrete Wavelet Transform

For the wavelet features, we use the Discrete Wavelet Transform to separate out 4 different sub-bands. We use the Daubechies order 5 wavelet as the mother wavelet. On these sub-bands, we pull out 12 features: zero-crossings, percentiles, mean, median, variance, standard deviation, root mean squared, and entropy. For each sample, we then get 48 DWT features, 12 per sub-band. We decide to use the DWT to generate features, since we see that the DWT features seem to carry much of the same frequency information. We try out these DWT features, since they provide us both rich information about the frequency domain and time domain characteristics of the signals.

IV. Supervised Learning

We set our goal to be predicting the location of the sensor (the 22 locations correspond to the labels 0 through 21 in our dataset). In order to do so, we use the input signals, as well at the features from the DFT and DWT as the inputs to our models. We describe our approaches below.

A. Simple Models

We begin by evaluating the performance of simple models. In particular, we use K-nearest neighbors, linear discriminant analysis, and quadratic discriminant analysis to predict the labels.

In K-nearest neighbors, we evaluated the prediction accuracy for the values of $k = 1, 3, 5, 10$. We present the test accuracy in the table below.

	k = 1	k = 3	k = 7	k = 10
Raw data	0.576	0.520	0.491	0.465
DFT features	0.0463	0.0533	0.0486	0.0580
DWT features	0.0439	0.0494	0.0478	0.0447

We can see that we get low accuracy, especially for the DFT and DWT features. This bad performance, for the DFT and DWT features, is akin to randomly guessing the correct label. The highest performing model was with the raw data, instead of the transformed features. This indicates that there is more information about the signals in the time domain than the frequency domain. Determining similarity between the raw signals is a better predictor that similarity between the frequency domains of signals.

We then try to run our data on LDA and QDA and also get bad performance. We present the findings in the table belows:

For LDA:

Features	Accuracy
Raw data	0.242
DFT features	0.0517
DWT features	0.0416

For QDA:

Features	Accuracy
Raw data	0.306
DFT features	0.0510
DWT features	0.0463

From these simple models, we do not get high prediction accuracy. The highest accuracy we get is from the simplest model: running k-nearest neighbors on the raw sensor data. We are curious to see if using a neural network could help us extract features that we might not otherwise be able to produce manually. This work is presented in the next section.

B. Convolutional Neural Network

Since we are getting bad performance on the simpler models, even with the DFT and DWT features, we believe we could get better performance using a neural net model. The power in neural networks is being able to extract features from the input data that would otherwise be tricky to pick out by hand. The architecture of the neural net we implemented was:

1) Convolutional layer (in channels = 1, out channels = 3)
2) ReLU activation
3) Convolutional layer (in channels = 3, out channels = 2)
4) ReLU activation
5) Flatten
6) Fully connected layer
7) Batch normalization
8) ReLU activation
9) Fully connected layer
10) Batch normalization
11) ReLU activation
12) Dropout layer
13) Fully connected layer
14) Batch normalization
15) ReLU activation
16) Dropout layer
17) Fully connected layer with 22 outputs, for 22 classes

Since this is a multi-class classification problem, we use a cross entropy loss function. We also add L2 regularization of the weights. Our final loss function is then:

$$\text{Loss} = -\sum_{c=0}^{22} y^{(i)} \log p_{i,c} + \|w\|^2$$

where w are the weights of the neural network. We train our model over 13000 iterations, with a learning rate of 0.001 and a regularization weight of 0.01. We implement Adam optimization, with a batch size of 64 samples.

With this neural network trained on our training data, we find that we get 0.633 training accuracy and 0.441 test accuracy. The curve of the accuracies over training time is shown in Figure 3.

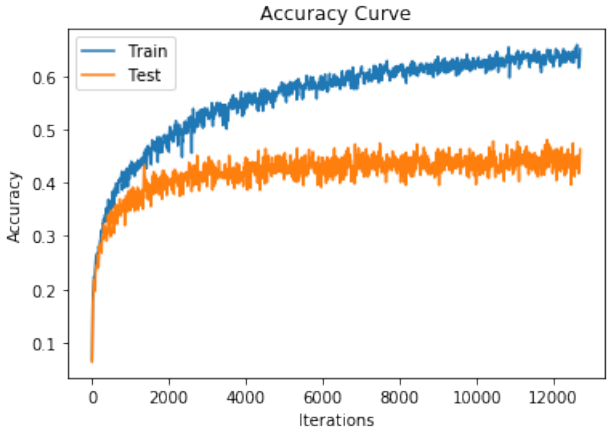

Fig. 3. Train and test accuracy, during training.

Although we achieved higher accuracy than we were able to get with LDA or QDA, we were still unable to beat the simplest model of k-nearest neighbors with $k = 1$ on the raw sensor data. This indicates that there is a problem with our approach: we believe we have formulated the wrong problem, and that it is difficult to predict locations as the labels. Instead of a supervised learning approach, we want to instead look for underlying structure in our data and try to draw information from approaching the dataset through unsupervised methods.

V. Dimensionality Reduction

What we first want to do, to understand some of the patterns in the dataset, is to project our data into two dimensions. This makes it easier for us to visualize possible trends. We do this by projected our dataset first onto the top two principle components, and then onto the top two eigenvalues from Fisher's LDA.

A. Principle Components Analysis

In PCA, we are able to extract the vectors that capture the most variance in the dataset. Since PCA is an unsupervised method, we do not have to take into account the labels when applying it to our data. For our purposes, during PCA, we extract the top two eigenvectors of the covariance matrix and use these as the axes along which we plot our datapoints. This process allows us to take our datapoints in 96 dimensions and project them into 2 dimensions that we can visualize. Figure 4 shows this PCA scatter plot, with the 22 different classes in different colors.

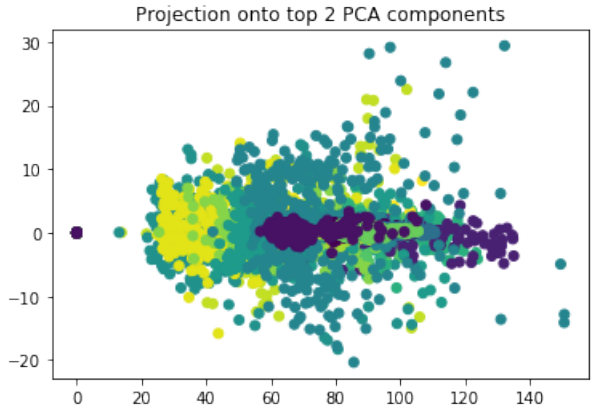

Fig. 4. Scatter plot of dataset along top 2 principle components.

In Figure 4 we can notice that although there's not clear distinction between groups, there does seem to be clustering within the different locations. For example, the location highlighted in yellow seems to be concentrated in a cluster.

We also look at the top principle components, to see what types of patterns are being pulled out by the eigendecomposition of the covariance matrix. A plot of the top 5 eigenvectors is shown in Figure 5.

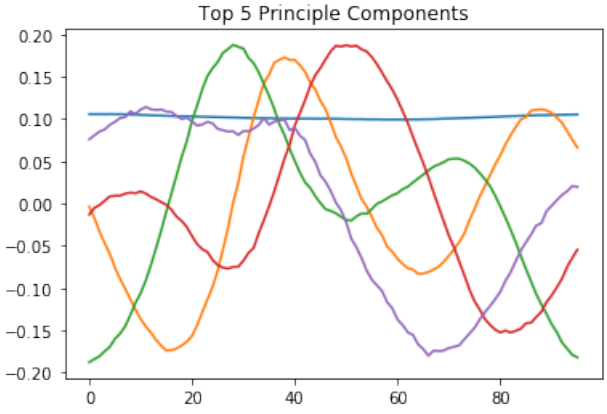

Fig. 5. Top 5 principle components.

We see that the top 5 principle components show these different patterns that exist in the dissolves oxygen content over the course of the day. For example we get a pattern where we peak early in the day, and then a smaller peak later on (shown in green). We can also look at the example of a very constant dissolved oxygen content throughout the day (shown in blue). We can interpret these top 5 principle components as the common dissolved oxygen content patterns that approximate much many of the points in our dataset.

B. *Fisher's LDA*

Fisher's LDA is another approach to dimensionality reduction. However, unlike PCA, Fisher's LDA takes into account the different labels. This makes Fisher's LDA a supervised algorithm, different from PCA and the other types of clustering described in the following sections. In Fisher's LDA, we calculate the scatter within labels and between labels, and find the vectors that maximize between class scatter and minimize within class scatter. We find the top two generalized eigenvectors of the between class scatter matrix and the in class scatter matrix and project our datapoints onto these axes. This plot is shown in Figure 6.

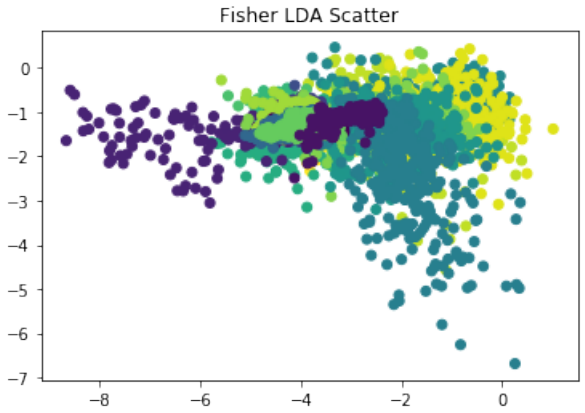

Fig. 6. Scatter plot of dataset along top 2 axes from Fisher's LDA.

In this scatter plot, we are able to see that the location represented by the purple points becomes more distinguishable from the rest of the datapoints. Again, we are able to see some grouping between locations, but there's not distinct clustering between the 22 different labels.

VI. Other Clustering

We also decide to try out some other clustering methods, k-means and agglomerative clustering to see how they compare to the clusters of the labels that exist in the dataset.

We run the clustering via these two different methods and specify 22 clusters. We then plot the PCA and Fisher's LDA scatter plots with these new clusters to compare how these clusters look with the actual labels in our data.

The scatter plots with k-means clustering are shown in Figure 7 and Figure 8. The scatter plots with agglomerative clustering are shown in Figure 9 and Figure 10.

We see that in the Fisher's LDA scatter plots with both k-means and agglomerative clustering, we can see that we define the points that show up on the lefthand side of the plot as a cluster, as it consistent with our Fisher's LDA plot of the actual labels.

VII. Discussion and Future Work

We are able to conclude the following:

- Our dataset does not contain important information in the frequency domain. Both DFT features and DWT features

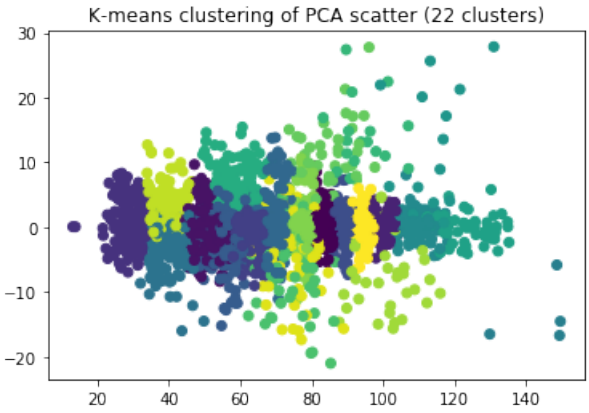

Fig. 7. PCA scatter plot with K-means clusters.

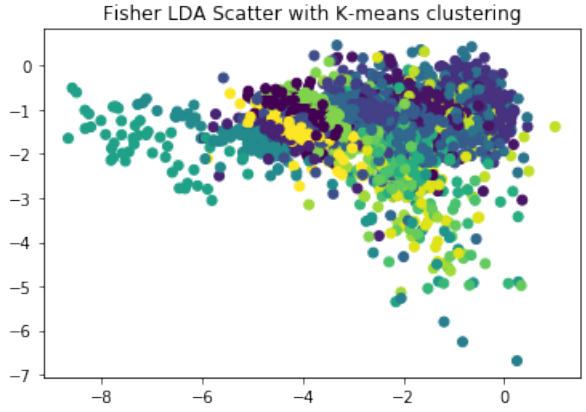

Fig. 8. Fisher's LDA scatter plot with K-means clusters.

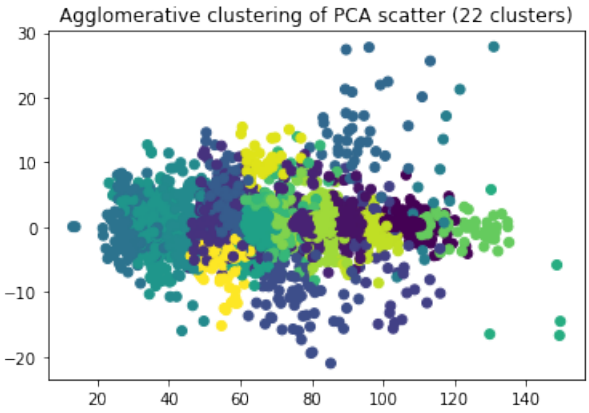

Fig. 9. PCA scatter plot with agglomerative clusters.

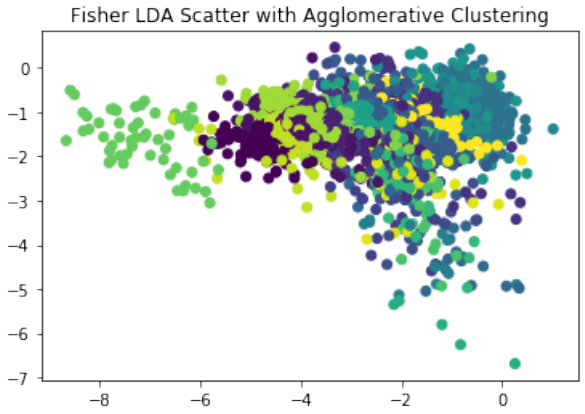

Fig. 10. Fisher's LDA scatter plot with agglomerative clusters.

performed poorly in the supervised learning tasks (on the order or randomly guessing labels).

- The most accurate supervised learning algorithm for this dataset is the simplest k-nearest neighbors with $k = 1$ on the raw data, with an accuracy of 57.6%. While this accuracy is better than randomly guessing, it's still not satisfactory.
- Working with the raw sensor data led to more accuracy than embedding the data via transforms manually, or through convolutional layers of a neural network. This indicates that the time-series nature of the data is significant in identifying the signal.
- When performing dimensionality reduction, we are able to separate out a few different clusters of data points. However, there is not clear distinction between the locations.
- In finding the top principle components, we can identify different patterns in daily dissolved oxygen content, that capture trends in the dataset.
- Performing clustering on the data via k-means and agglomerative clustering allows us to correctly separate out some groups that exist within the labels.

While this project has no conclusive results, or models that are able to correctly predict locations of the sensors, we can give the following recommendations for future work with this dataset:

- Predicting the 22 different locations does not work with the supervised learning methods tried in this dataset. We believe this has to do with the labels themselves. Instead of predicting the labels that are the locations, it might be best to predict certain types of ecosystems, that have distinct patterns of dissolved oxygen content.
- From the PCA decomposition, we were able to identify different dissolved oxygen content patterns. Future work could include clustering the data into fewer clusters and identify common trends in the daily dissolved oxygen content. With more domain knowledge and expertise, one could make assumptions about the productivity or health of the ecosystem.
- The sensors that measure dissolved oxygen content often are bundled with other sensors that measure quantities like turbidity or light. These other metrics could be used in conjunction with dissolved oxygen data for improved models.

VIII. Code

All code for this project can be found at: https://github.com/natgable/dissolved-oxygen-clustering

IX. Acknowledgments

Thanks to my friend and colleague, Sienna White, for providing the dataset.

X. References

References

[1] Taspinar, A. (2019, April 05). A guide for using the Wavelet Transform in Machine Learning. Retrieved November 19, 2020, from http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/

[2] Verma, A. (2020, June 30). PyTorch [Tabular] -Multiclass Classification. Retrieved November 19, 2020, from https://towardsdatascience.com/pytorch-tabular-multiclass-classification-9f8211a123ab