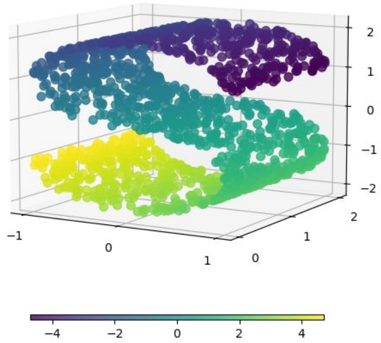
Unsupervised Learning Using Optimal Transport Based Embeddings

By Ryan Bui

Introduction to Manifold Learning

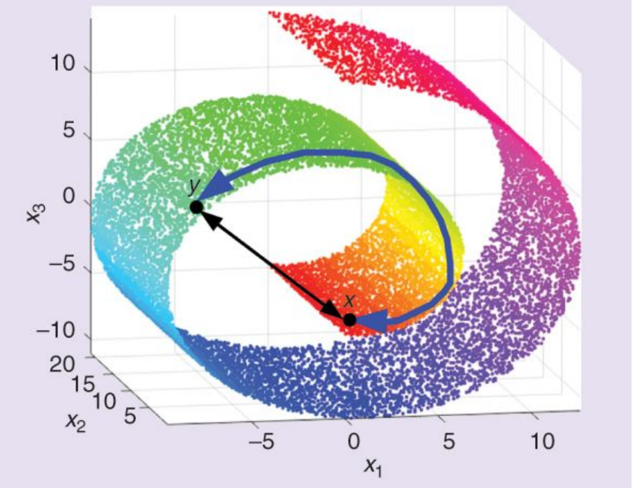
- Manifold Learning helps us visualize high-dimensional data by lowering the dimension of the data it allows us to visualize the data.
- PCA is similar in that it focuses on capturing linear structures within the data. While manifold learning techniques aim to capture nonlinear structures
- Manifold learning techniques aim to preserve the distance between points when mapping high-dimensional data onto lower-dimensional representations.

Original S-curve samples



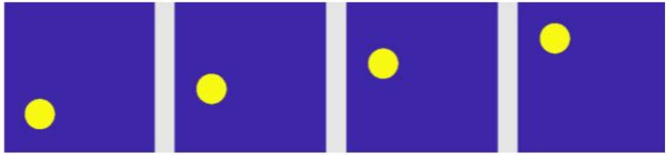
Why Optimal Transport?

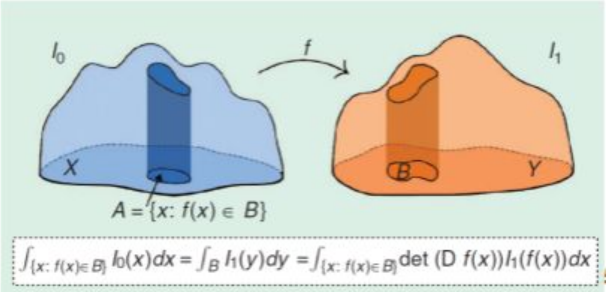
- When using manifold learning techniques such as Isomap or MDS (Multi-Dimensional Scaling) they both calculate distances between points.
- Isomap seeks a lower-dimensional embedding which maintains geodesic distances between all points.
- In the case of MDS it can actually use distance matrices created from different types of distances.



Why Optimal Transport?

- Wasserstein distance deals with distributions. It measures the minimum amount of work needed to transform one distribution into another.
- Optimal transport methods are valuable when the data has a strong distributional component, such as in image processing and NLP (natural language processing)




I_0
f
I_1
X
B
Y
$A = \{x: f(x) \in B\}$
$\int_{\{x:\, f(x)\in B\}} I_0(x)dx = \int_B I_1(y)dy = \int_{\{x:\, f(x)\in B\}} \det(D\, f(x)) I_1(f(x))dx$

The Goal

The goal of this project to provide a comprehensive evaluation of optimal transport-based techniques in real-world scenarios by applying it to a variety of datasets.

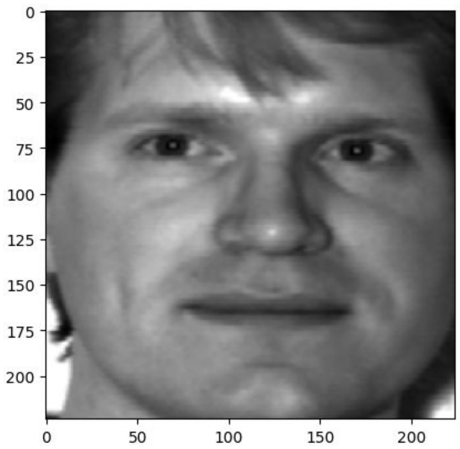
We would like to compare our method against already existing methods and get a deeper understanding of the strengths and weaknesses of using Wasserstein distance.

The Data: Yale Face Dataset

- The database contains 165 single light source images of 15 subjects. One for each of the following facial expressions or configurations: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.
- **Origin**: This is a public dataset provided by Yale.
- **Size**: The total size of the compressed dataset is about 12mb

Data Preprocessing

- For this dataset I needed to transform the image into a numpy array of pixel values and make sure to grayscale the image.
- I also decided to use a package called MTCNN (Multi-Task Cascaded Convolutional Neural Networks) which detects faces and facial landmarks on images.
- This is used to make the image more about the face rather than the background
- Additional preprocessing for Wassmap will be talked about later in the presentation.

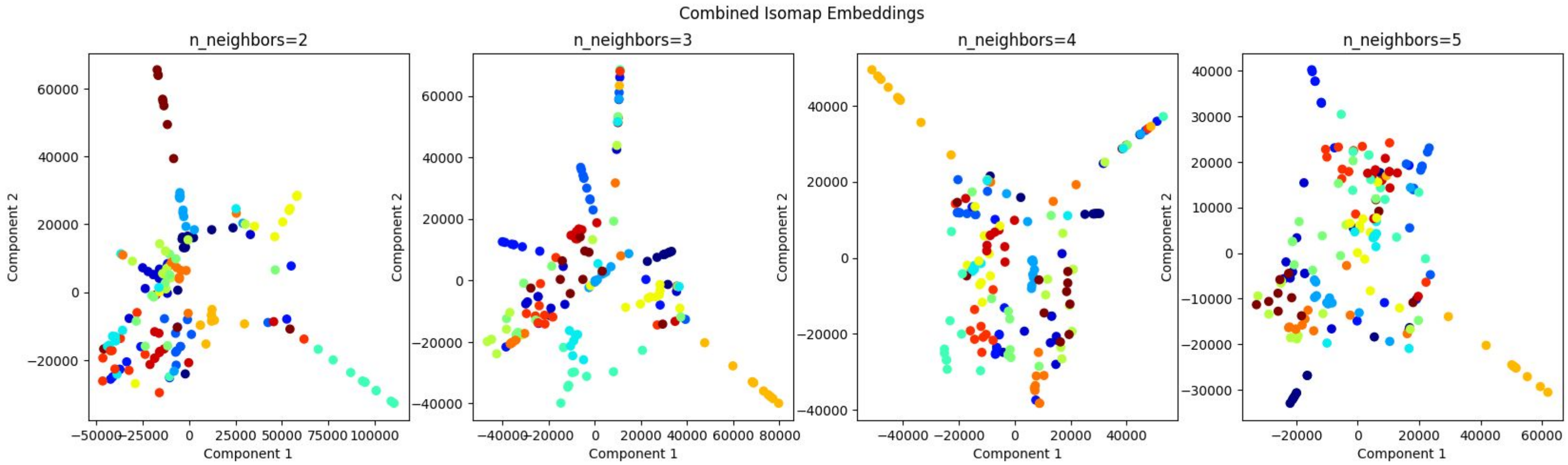


Original Image

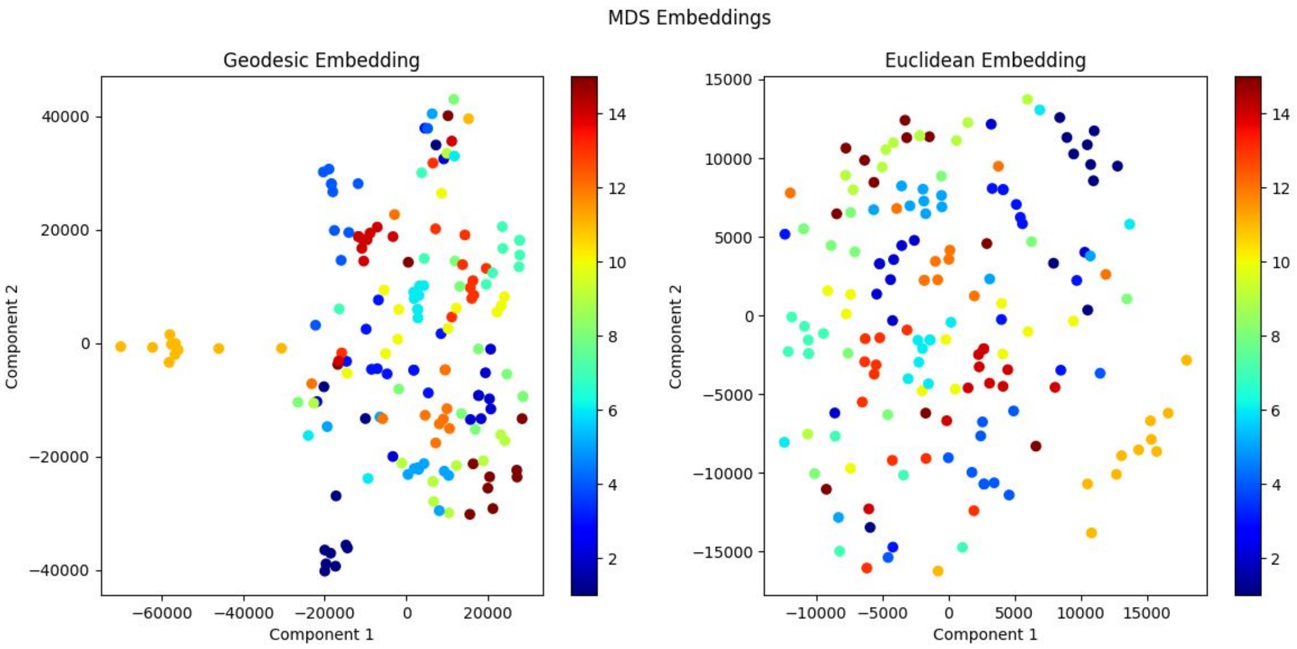
The Embeddings

- We will use 7 different embeddings:
    - Isomap
    - MDS (Multi-dimensional Scaling)
    - LLE (Locally Linear Embedding)
    - t-SNE
    - LTSA (Local Tangent Space Alignment)
    - Diffusion Map

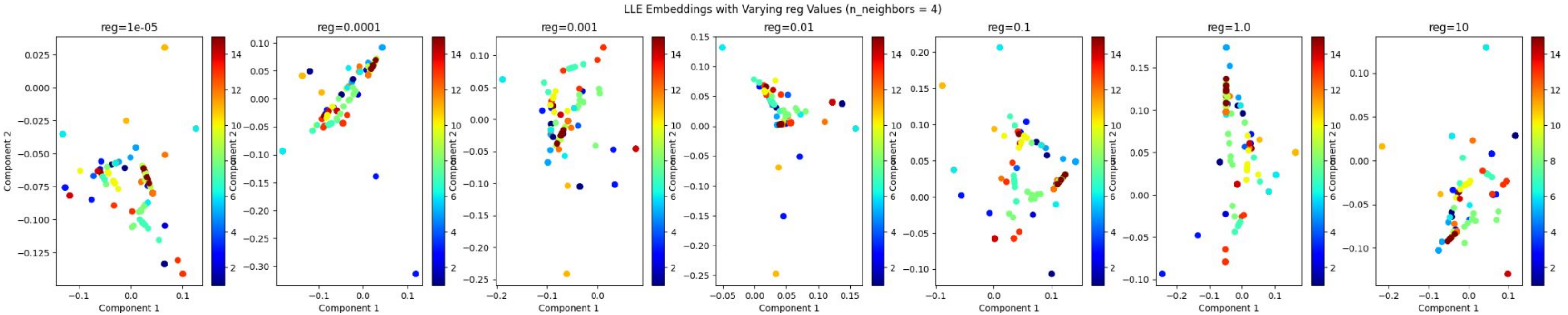
Isomap Embeddings

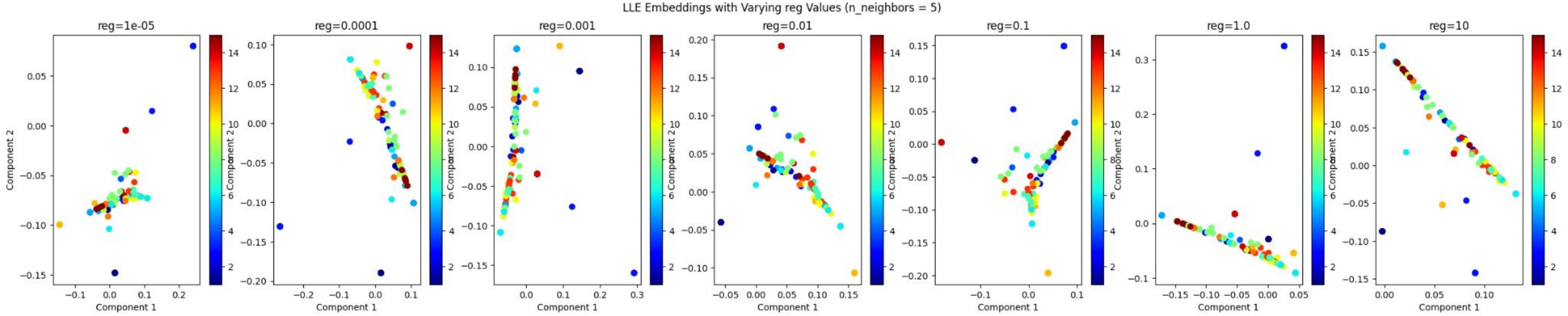


MDS Embeddings



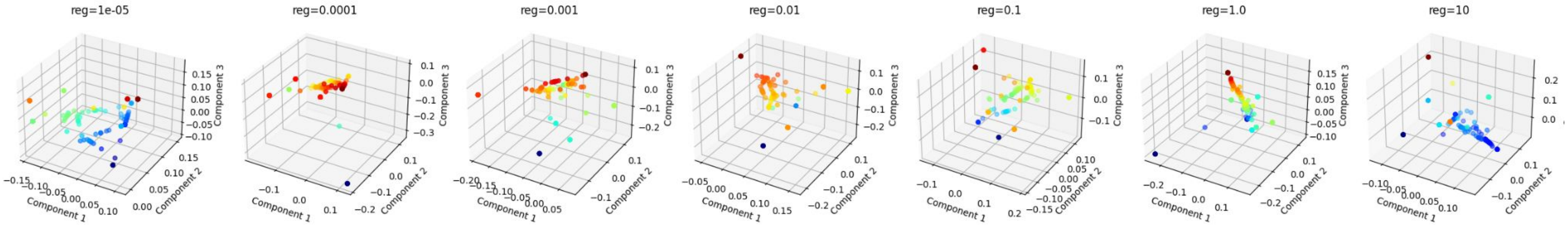
Locally Linear Embeddings (LLE)



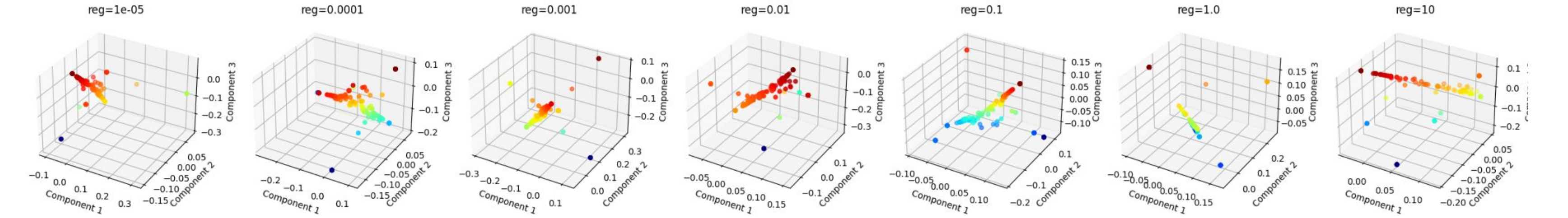


Locally Linear Embeddings (LLE)

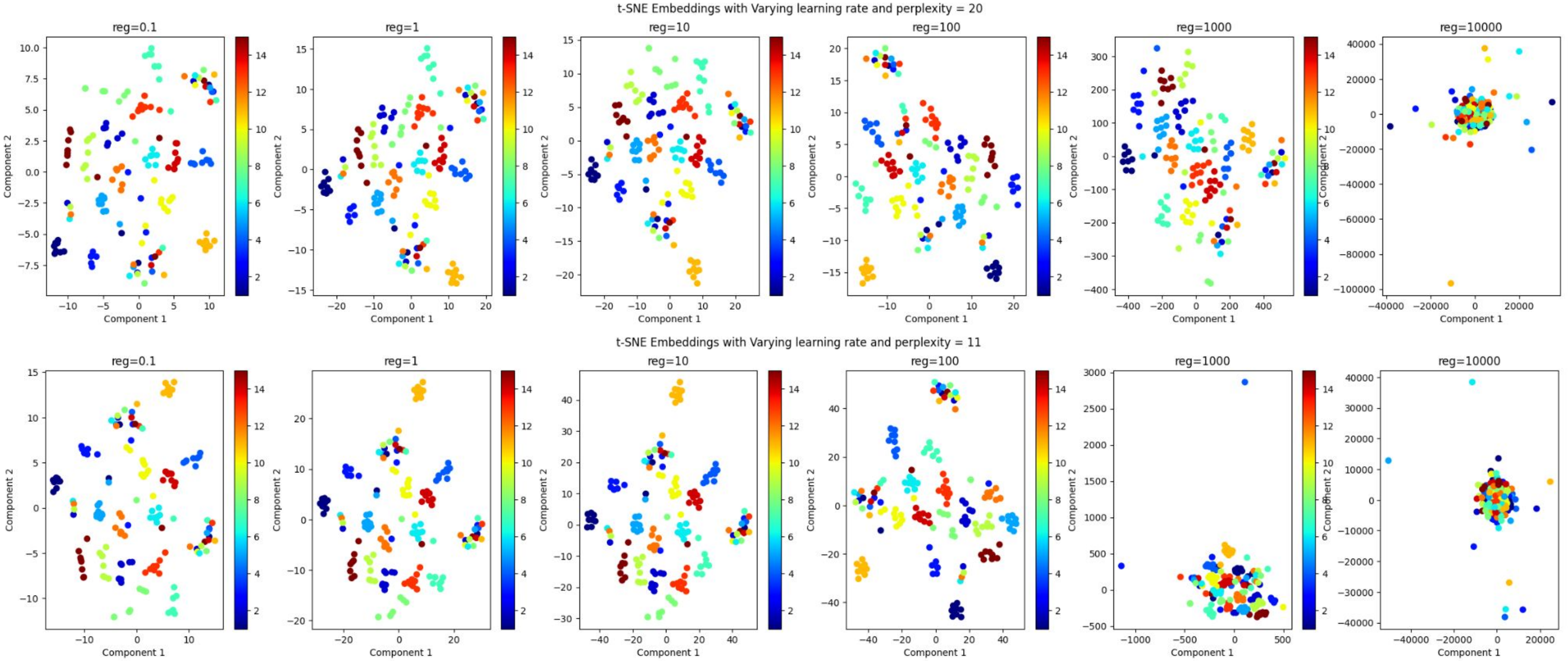
LLE Embeddings with Varying reg Values (n_neighbors = 4)



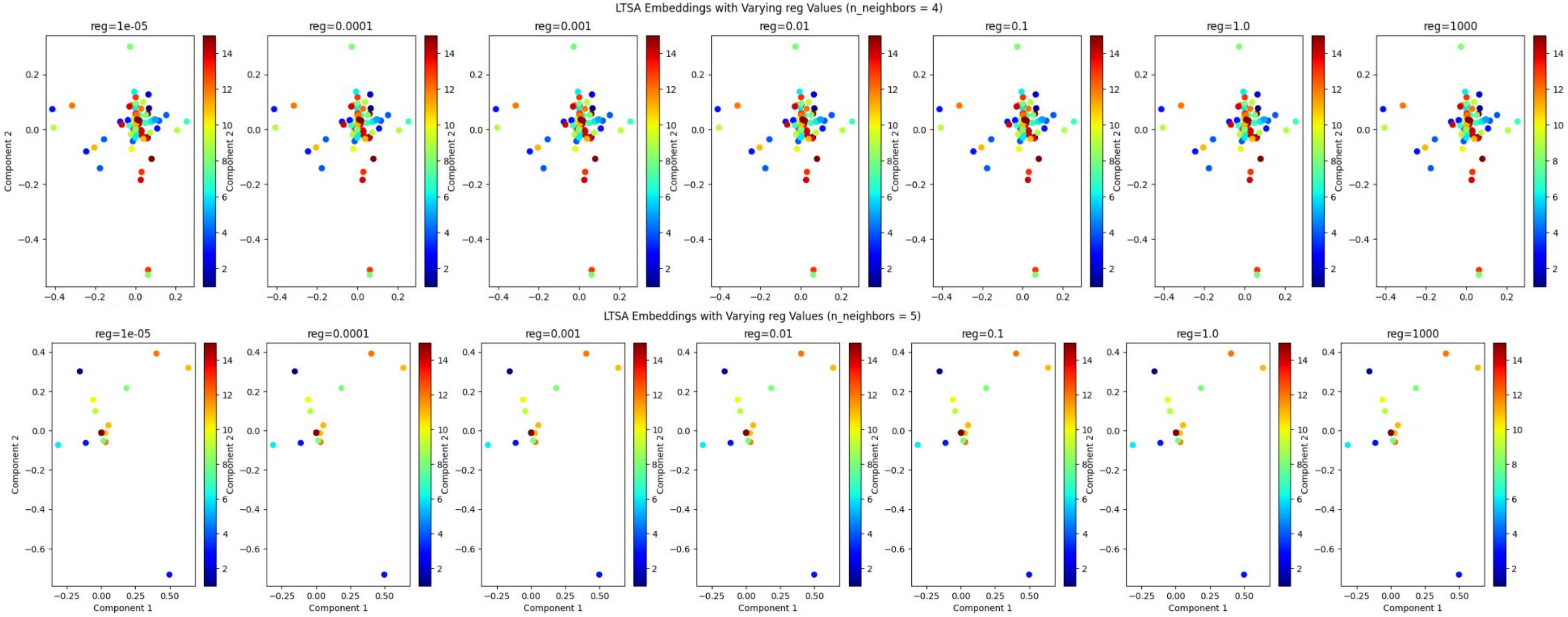
LLE Embeddings with Varying reg Values (n_neighbors = 5)



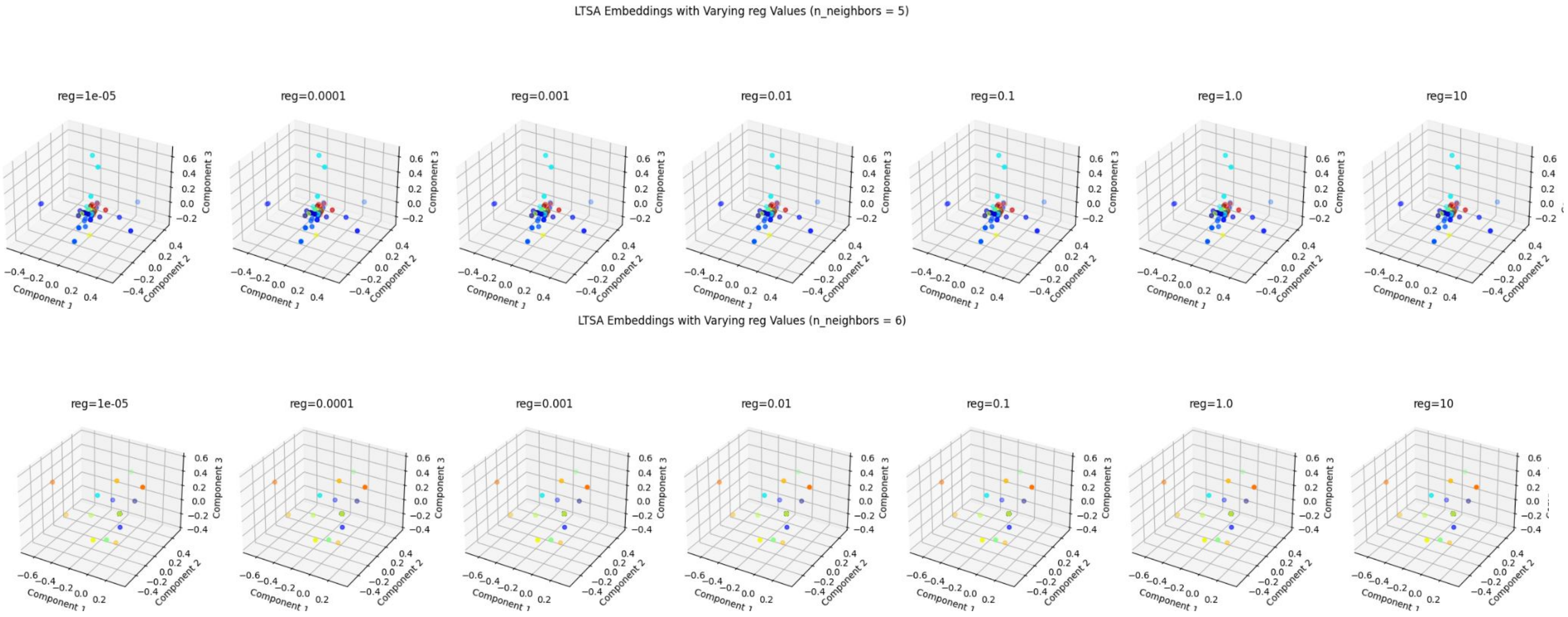
t-SNE Embeddings



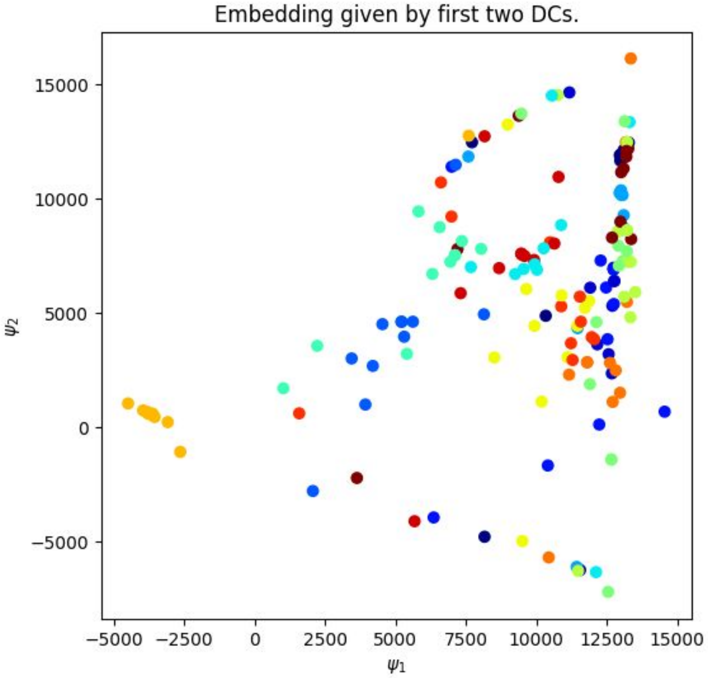
Local Tangent Space Alignment (LTSA) Embeddings

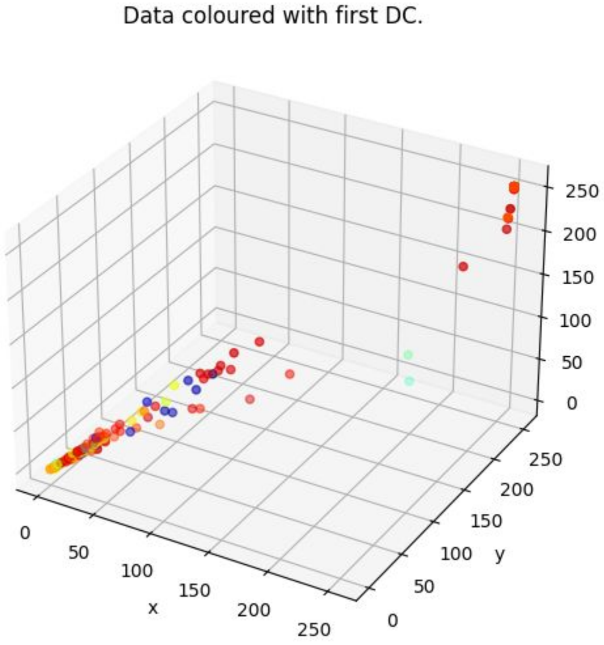


Local Tangent Space Alignment (LTSA) Embeddings



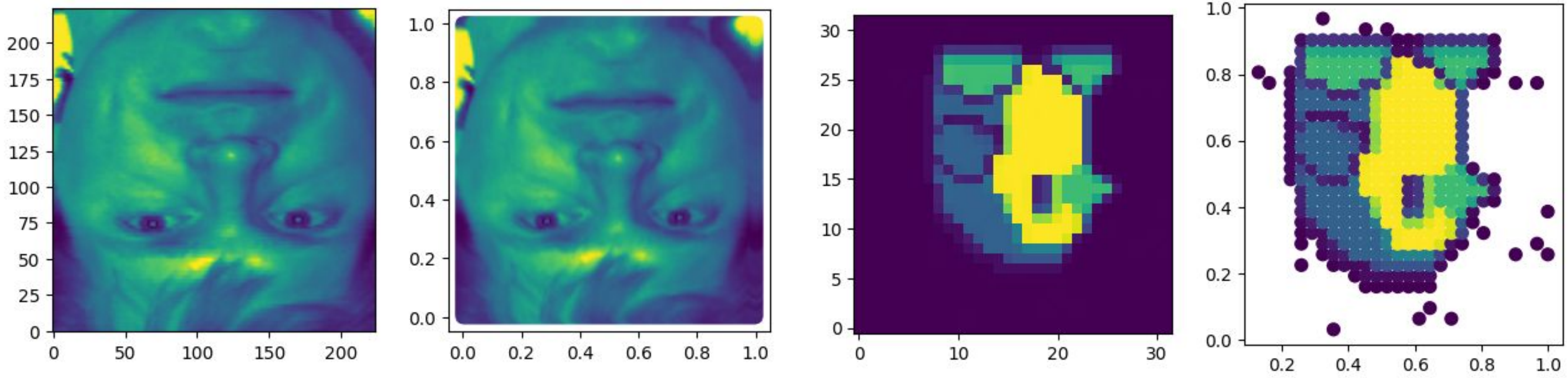
Diffusion Map Embeddings



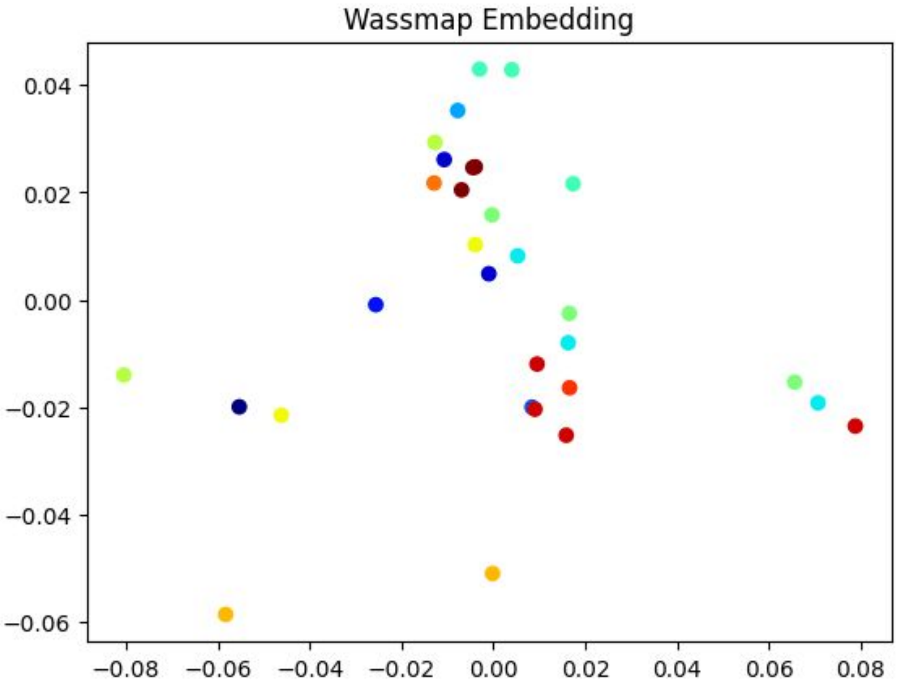


Preprocessing for Wassmap

- The only thing we need to do before using our Wassmap function is to convert all of images to a point cloud format this is necessary for our functions to run.



Wassmap Embedding

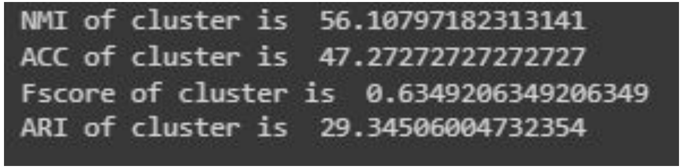


Problems using Wassmap

- Calculating Wasserstein distances is very computationally expensive
- Had to downsample my images to be 64x64
- Lowered my sample size to only 30 image
- In the future I plan to use some updated code to make the distance calculation faster and also run this code on a machine that can handle the higher resolution images

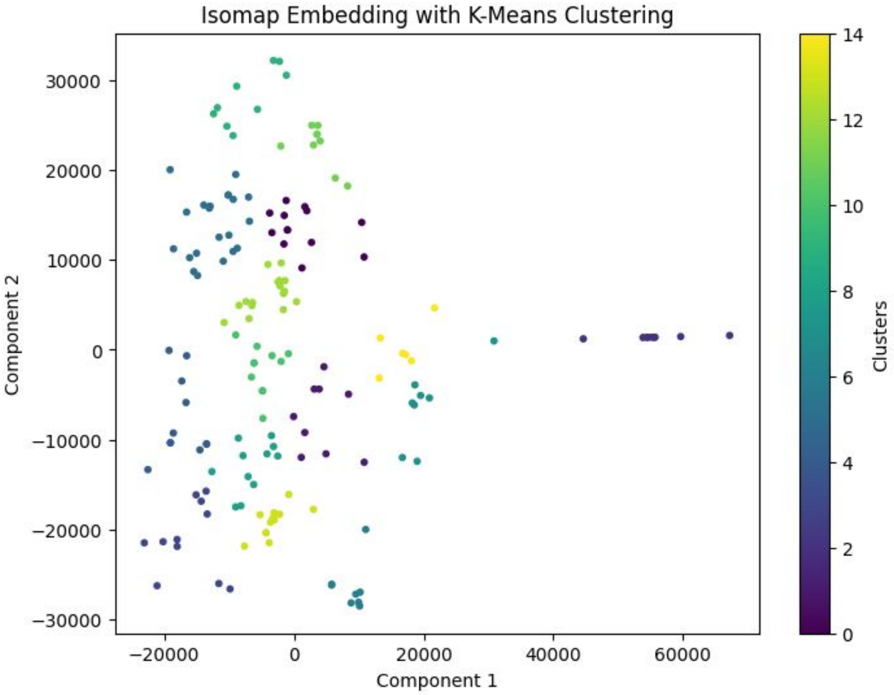
Performance

- In order to get some performance metrics from our embeddings we will be applying them to different clustering algorithms and see how they perform
- We will use 4 different types of performance metrics
    - Normalized Mutual Information (NMI)
    - Accuracy
    - F1 Score
    - Adjusted Rand Index (ARI)
- As of right now I have only tested the Isomap embedding on the k-means clustering algorithm.

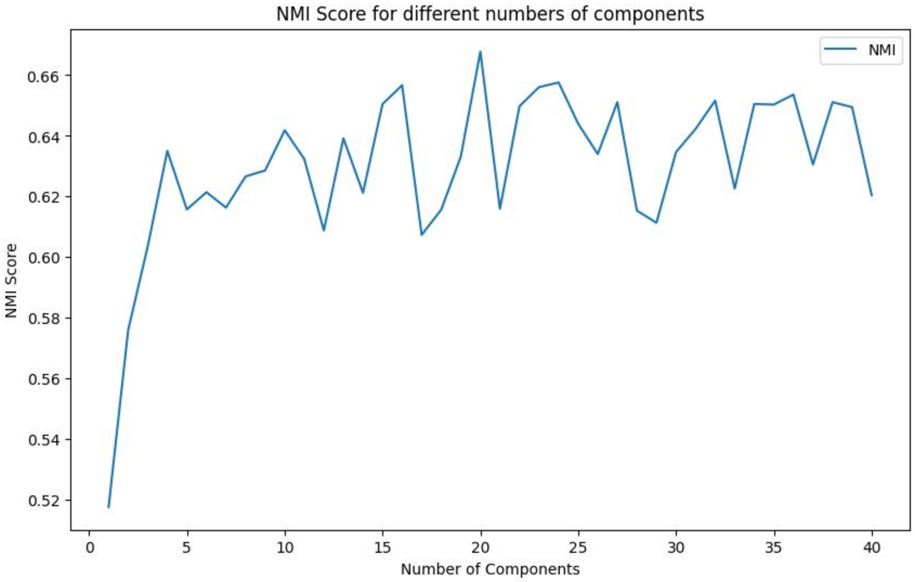


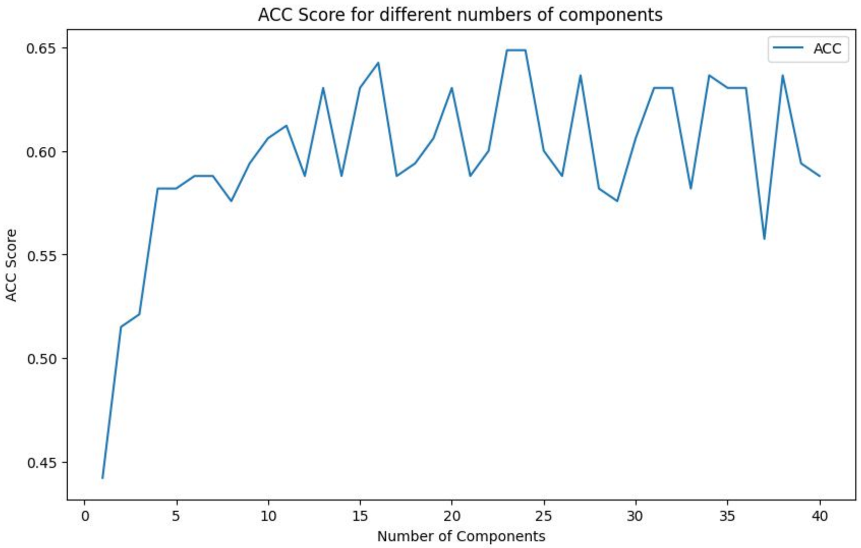
```
NMI of cluster is  56.10797182313141
ACC of cluster is  47.27272727272727
Fscore of cluster is  0.6349206349206349
ARI of cluster is  29.34506004732354
```

Performance (Isomap)

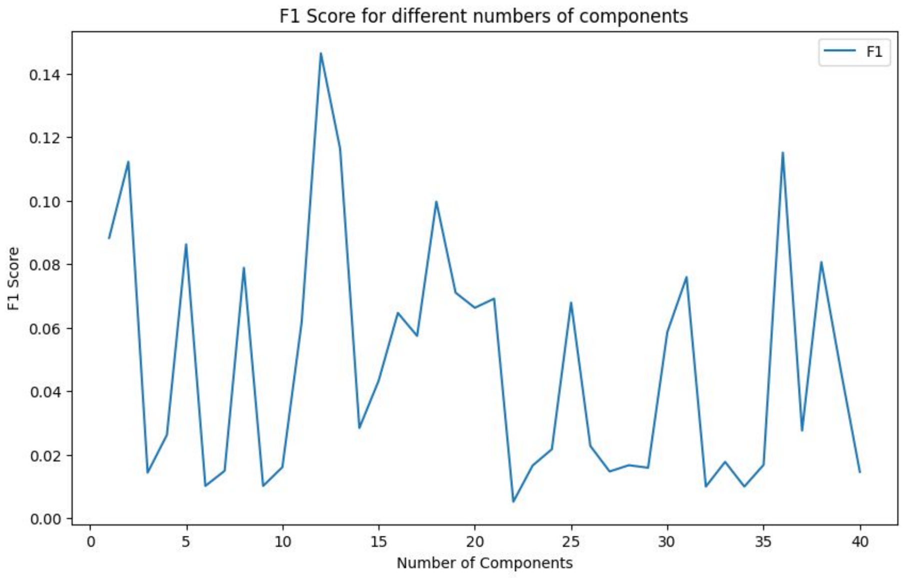


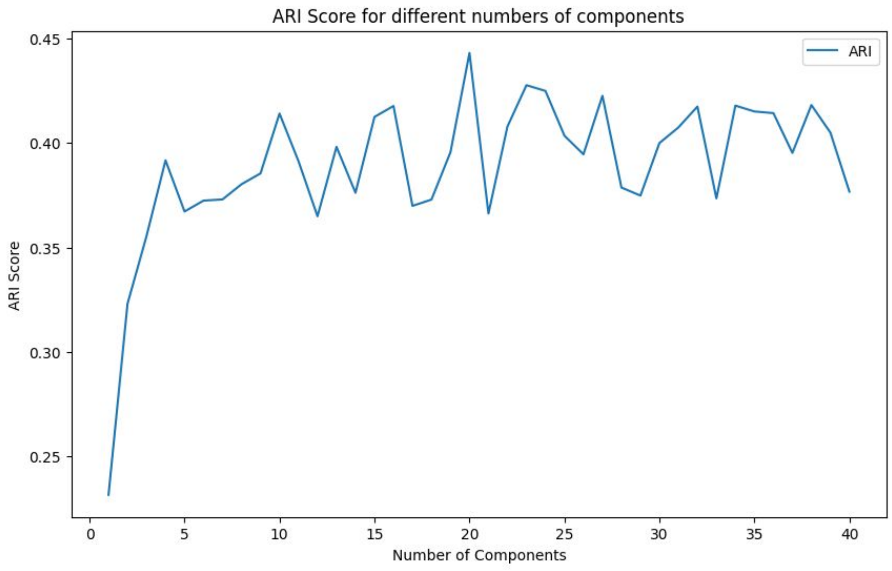
Performance (Isomap)





Performance (Isomap)

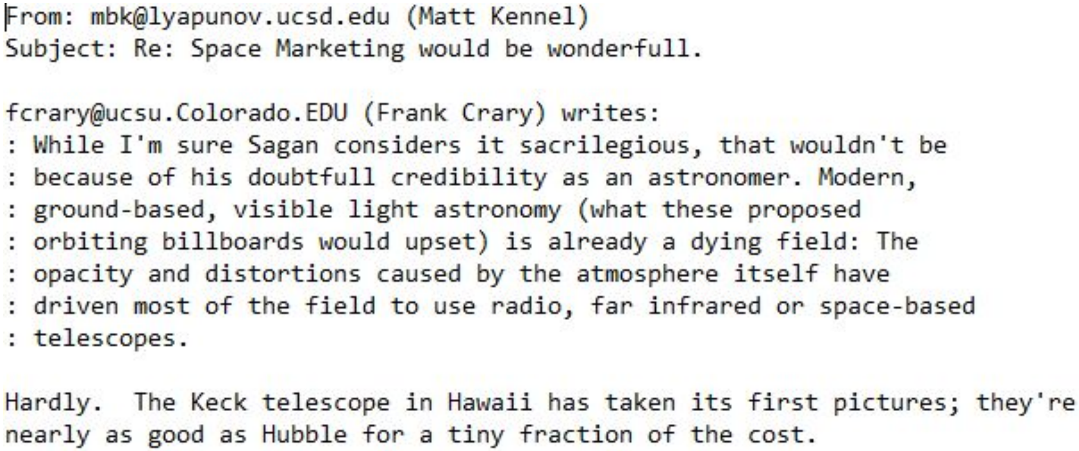




The Data: Newsgroup Dataset

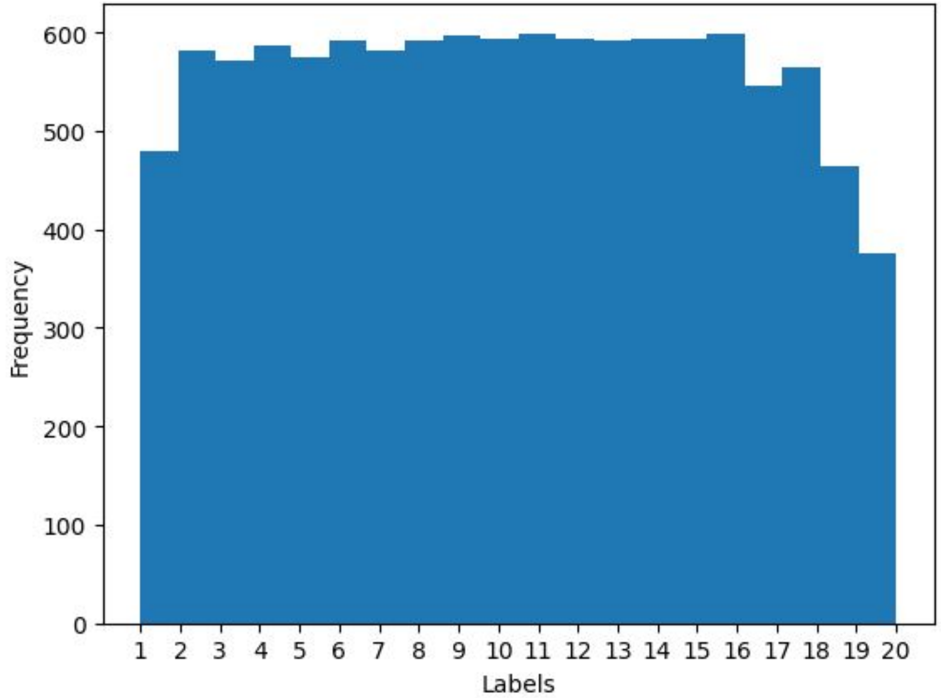
- Type: The 20 Newsgroups data set is a collection of approximately 11,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.
- This data does come with a .data file which can be read in as a pandas dataframe where the first column represents the document ID the second column represents the word ID and the 3 column represents how many times that word was used in that particular document.
- This data also comes with a .label file which represents the genre of each document.
- Origin: It was originally collected by Ken Lang.
- Size: The total size of the compressed database is about 14mb

The Data: Newsgroup Dataset



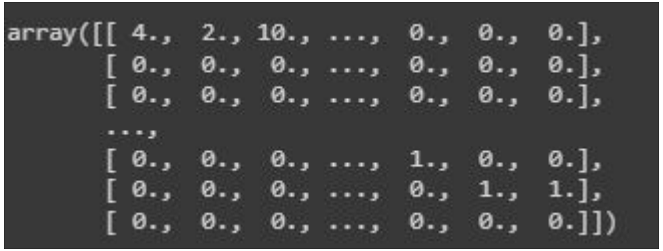
```
From: mbk@lyapunov.ucsd.edu (Matt Kennel)
Subject: Re: Space Marketing would be wonderfull.

fcrary@ucsu.Colorado.EDU (Frank Crary) writes:
: While I'm sure Sagan considers it sacrilegious, that wouldn't be
: because of his doubtfull credibility as an astronomer. Modern,
: ground-based, visible light astronomy (what these proposed
: orbiting billboards would upset) is already a dying field: The
: opacity and distortions caused by the atmosphere itself have
: driven most of the field to use radio, far infrared or space-based
: telescopes.

Hardly.  The Keck telescope in Hawaii has taken its first pictures; they're
nearly as good as Hubble for a tiny fraction of the cost.
```

The Data: Newsgroup Dataset



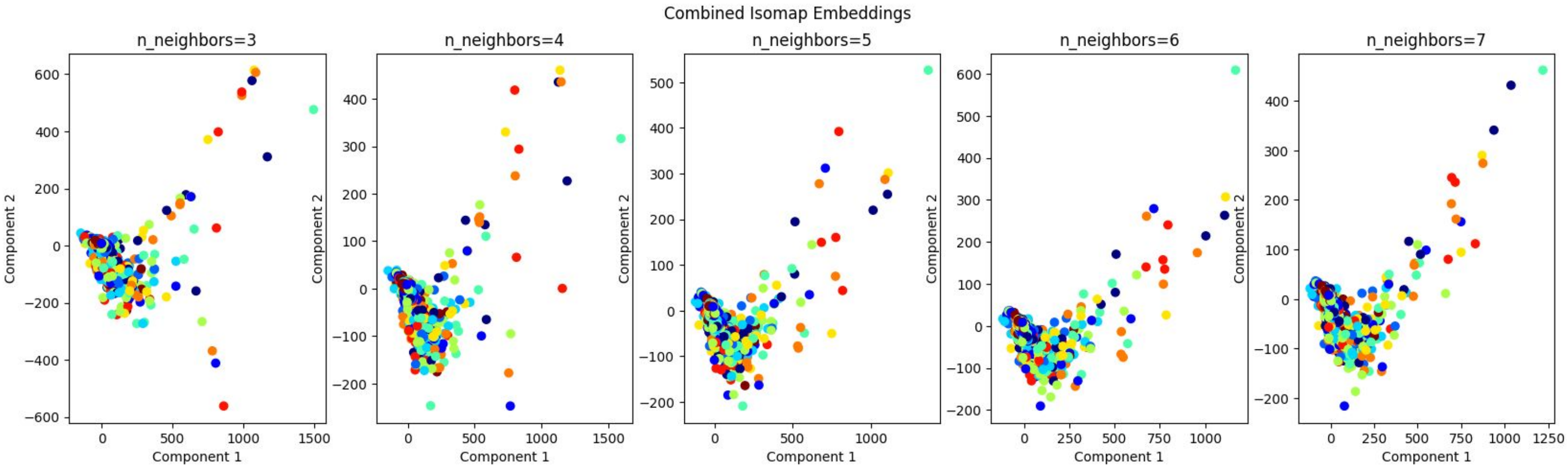
Data Preprocessing (Newsgroup)

- For our embeddings I transformed the given data set into pandas dataframe where each row represents a document and each column represents a word and the values for the columns represent the frequency of the word in the particular document.
- Because our matrix is so large I downsample my data to be only 5500 data points instead of the original 11629 in order to speed up the creation of the embeddings.



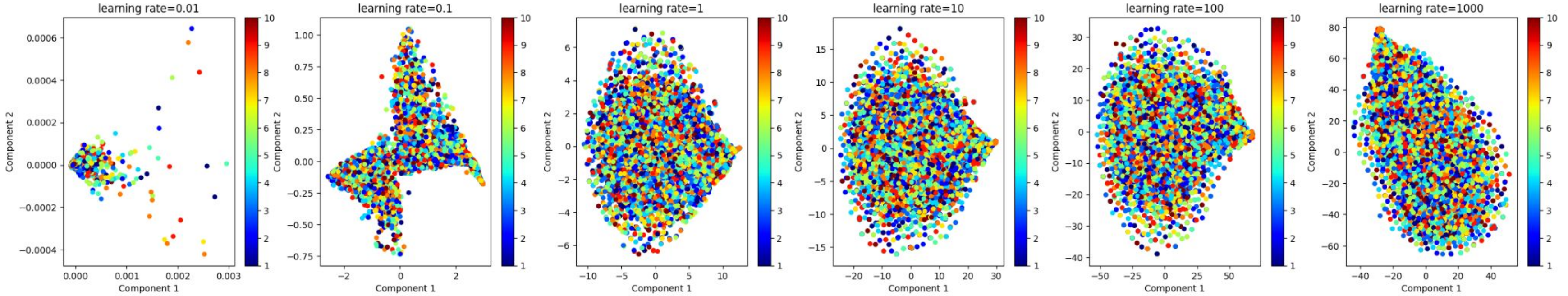
```
array([[ 4.,  2., 10., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       ...,
       [ 0.,  0.,  0., ...,  1.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  1.,  1.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.]])
```

Isomap Embedding

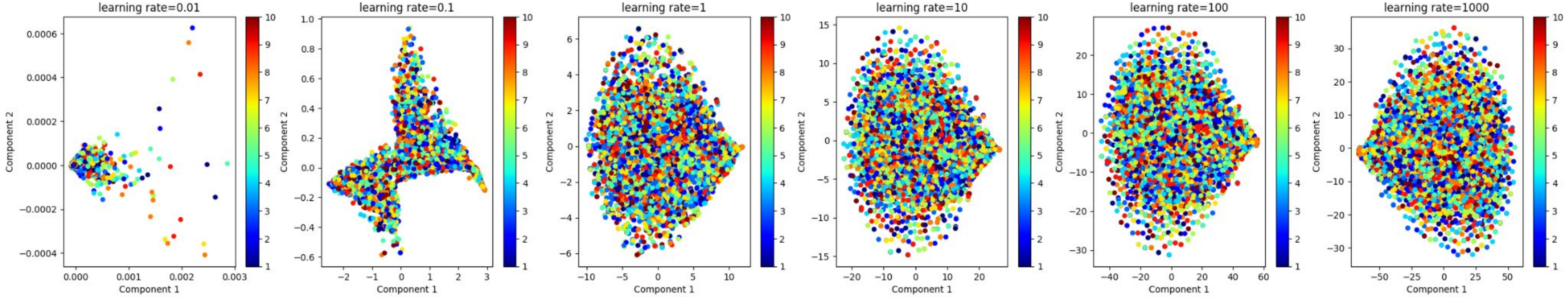


t-SNE Embeddings

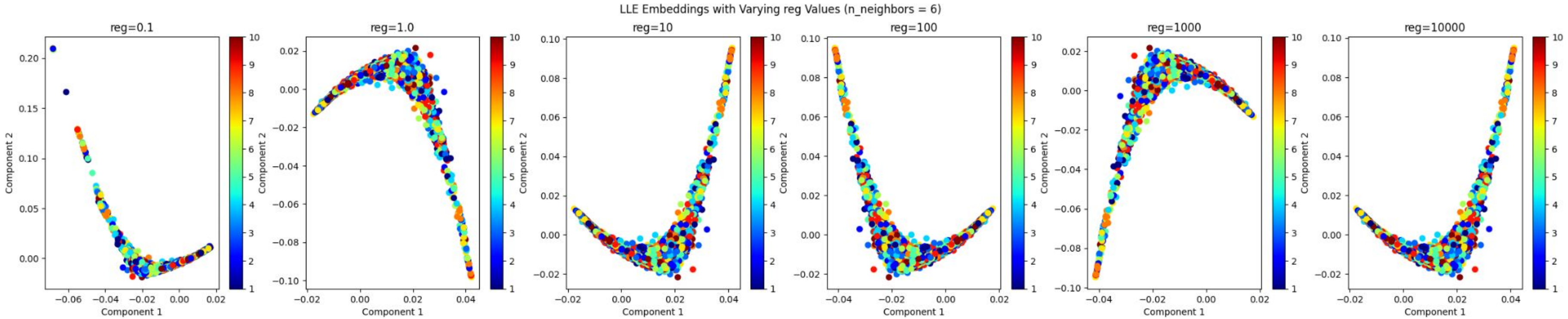
t-SNE Embeddings with Varying learning rate and perplexity = 30



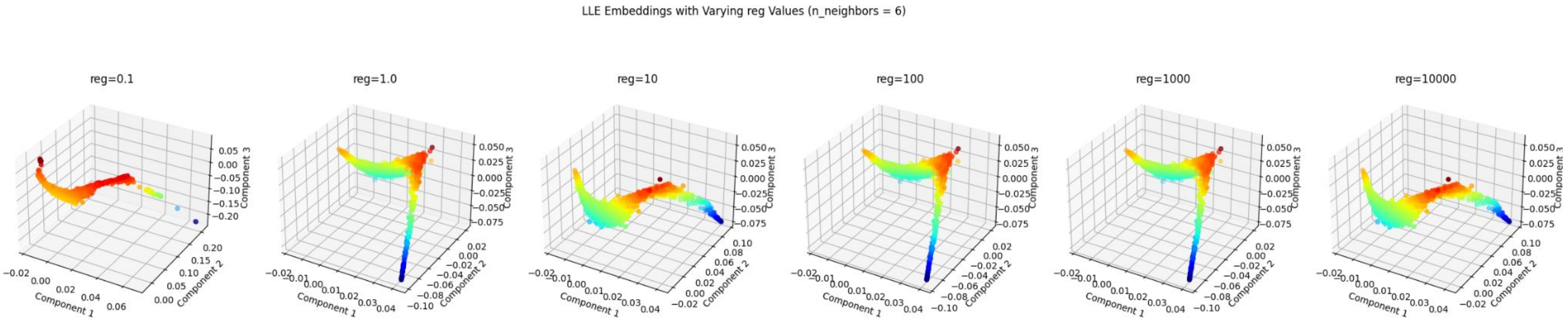
t-SNE Embeddings with Varying learning rate and perplexity = 40



Locally Linear Embeddings (LLE)



Locally Linear Embeddings (LLE)



Future Work

- More data exploration
- Scale up functions
- Finishing up embeddings and testing out different parameters
- Run Wassmap on updated code in order to increase sample size and run the code on a better machine for higher resolution images
- Get performance metrics for all embeddings
- Perform this on a 3rd dataset

References

https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database

http://qwone.com/~jason/20Newsgroups/

https://scikit-learn.org/stable/modules/manifold.html#

https://pydiffmap.readthedocs.io/en/master/reference/diffusion_map.html