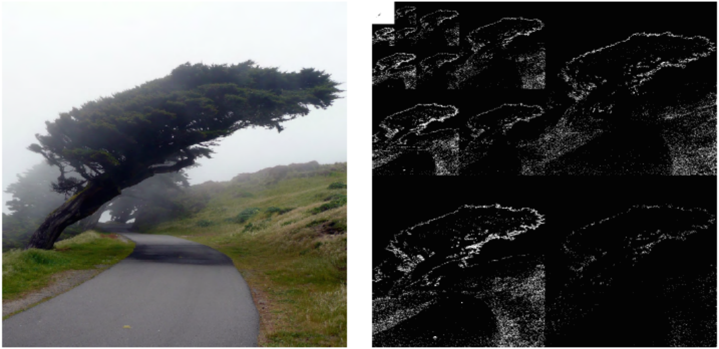
Dealing with Multidimensional Data

- Ubiquitous problem of processing multidimensional signals or data
- Images, Speech Signals, Sensor Data, Medical Readings etc.
- Often we would like to *preprocess* the data so that we can work effectively in a low dimensional space, remove everything that is irrelevant or remove everything that all the instances have in common
- Compress our data and find compact representations of their variability

Other approaches, PCA and Sparse Coding

- PCA models linear variations in high-dimensional data
- Compute the linear projections with the highest variance based on eigenanalysis of the data covariance matrix
- Sparse Coding: find *sparse* representations of data in overcomplete bases

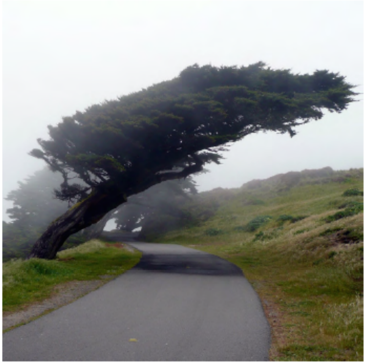
Sparse Representations of Images



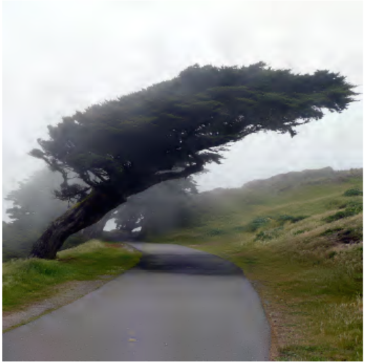
(a) (b)

- Oftentimes natural signals are *sparse* in a suitable basis

Reconstruction from Largest Wavelet Coefficients



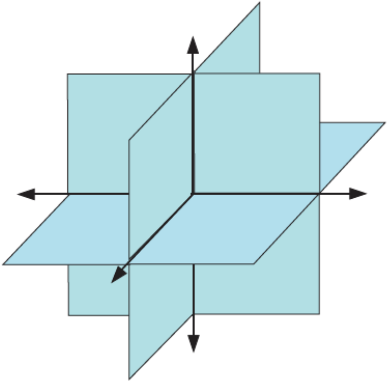
(a)



(b)

- Reconstruction obtained with the 10 percent largest wavelet coefficients

Geometry of Sparse Coding

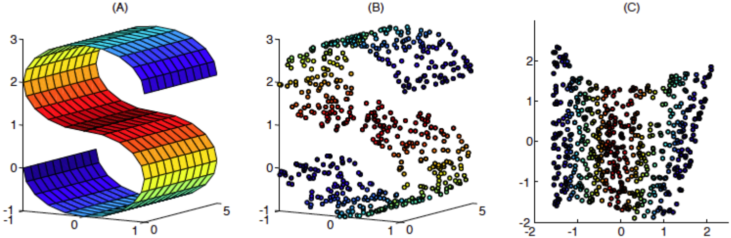


- The 2-sparse signals or vectors in three dimensional space

Locally Linear Embedding

- Algorithm for non-linear dimensionality reduction
- Developed due to the inherent limitations in linear methods

Locally Linear Embedding Illustration



- A non-linear manifold embedded in 3-dimensional space

Manifold (Differentiable)

- *Manifold*: Collection of points that are connected to one another in a smooth fashion such that the neighborhood of each point looks like the neighborhood of an m dimensional cartesian space, m is the dimension of the manifold
- Examples: $\mathbb{R}^n$, surface of a sphere, a torus, higher dimensional spheres, collection of all matrices whose elements are infinitely differentiable functions
- Not manifolds: spaces with sharp edges or kinks, isolated points

Locally Linear Embedding

Box 2

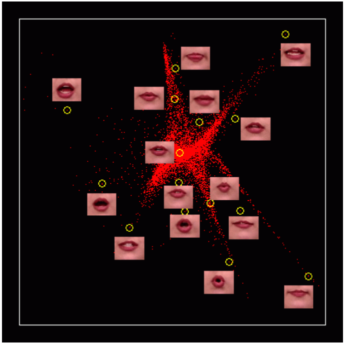
- Find the neighbors of each data instance, x_i
- Find a weight matrix W to reconstruct each data point from its neighbors
- Find the vectors y_i best reconstructed by the weights

$$\sum ||x_i - \sum_j W_{ij} x_j||^2 \tag{1}$$

$$\sum ||y_i - \sum_j W_{ij} y_j||^2 \tag{2}$$

Locally Linear Embedding (example)

- The LLE approach can be used to find the underlying structure in a set of natural images [https://www.cs.nyu.edu/ roweis/lle/]



Readings

- Read this paper for reference
- Saul, Lawrence K., and Sam T. Roweis. ”An introduction to locally linear embedding.” unpublished. Available at: http://www. cs. toronto. edu/ roweis/lle/publications. html (2000).

LLE in Python

- LLE is implemented in `sklearn` as `LocallyLinearEmbedding` or `locally_linear_embedding`
- Find the `roll.txt` csv dataset on the moodle, the datapoints are given in the first three columns
- Attempt to analyze the data using PCA (`sklearn.learn.decomposition`) and plot the transformed results (use the 4th column as color of the data points)
- Then attempt to apply the `LocallyLinearEmbedding` methods to the same dataset
- Remember you can use the pandas library for loading in csv files