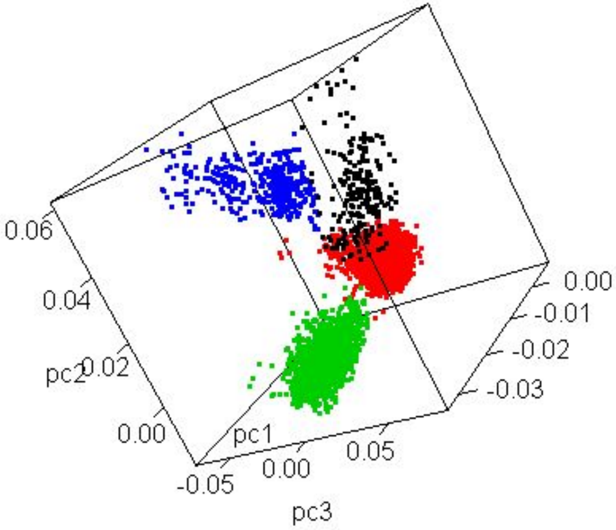
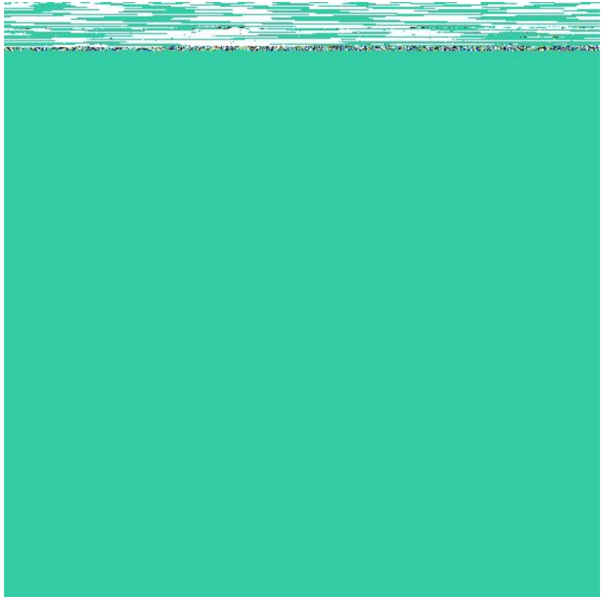
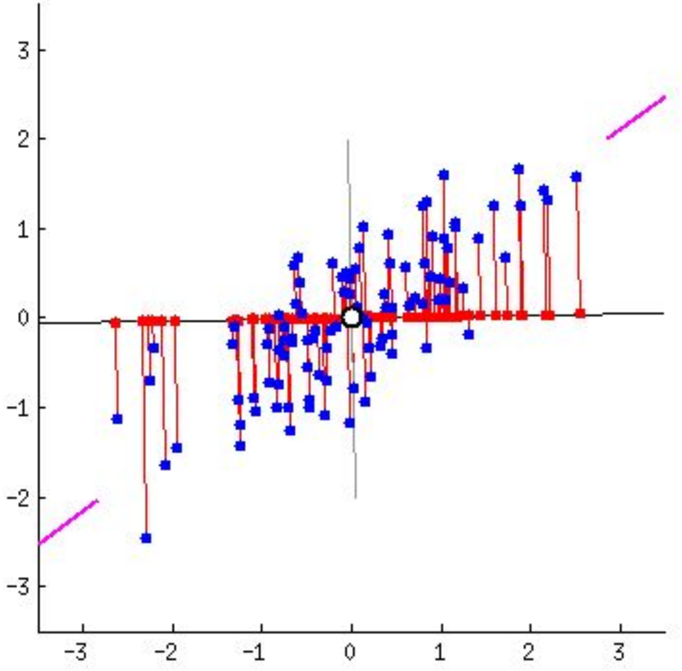
Dimensionality Reduction

Week 10 Discussion





A Review: PCA



- dimensionality reduction by projecting data onto the principal components
- principal components correspond to directions of highest variance

PCA Three Ways

1. Project data onto the leading left-singular vectors of the mean-centered X (using SVD)
2. Project Data onto the leading eigenvectors of the covariance matrix (using Eigendecomposition)
3. Project data to k-dimensions such that the projected data has the highest variance (solving the equation below)

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\}$$

Inspired by Seung Hee Lee's lecture notes

The Two Common Algorithms

Algorithm 1

- Mean center the data

compute the global mean $\vec{m}$

for $i=1$ n

$x_i = \vec{x}_i - \vec{m}$

end

- Take K-SVD of $\tilde{X}$

$\tilde{X}_K = U_k \Sigma_k V_k^T$

- Project data using U_k:

$\vec{x}_{proj} = U_k^T \vec{X}$

Algorithm 2

- Form the Sample Covariance matrix C

(compute global mean)

$$C = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i - \vec{m})(\vec{x}_i - m)^T$$

- Take top-k eigen val/ eigenvectors of C

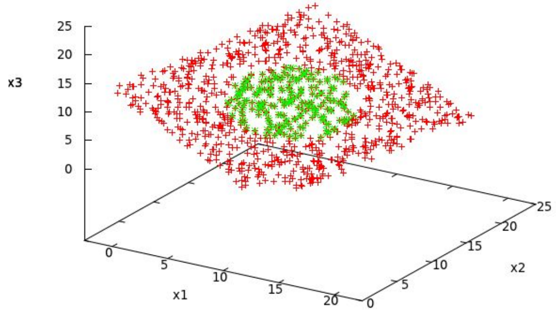
$$C_k = {}_d\overset{k}{U}_k \, {}_k\overset{k}{\Lambda}_k \, {}_k U_k^T$$

- project using U_k

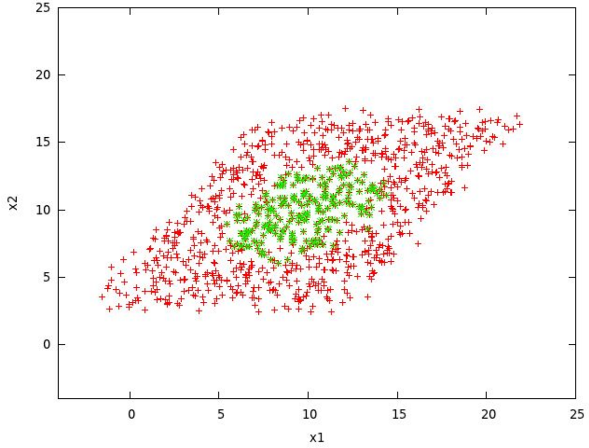
$\vec{X}_{proj} = U_k^T \vec{X}$

Notes by Seung Hee Lee (thank you!)

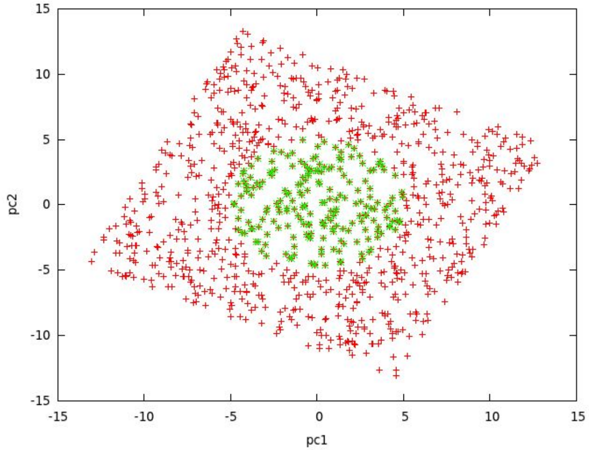
PCA for Visualization



2d map
(drop 1 dimension)



2d dimensionality
reduction with PCA

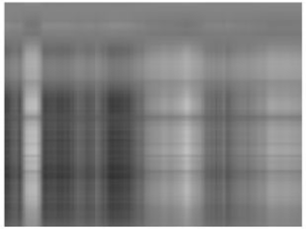


Dataset photos from http://nghiaho.com/?page_id=1030

A Real World Example: Image Compression

Original Image

An Image has 512x512 pixels,
each pixel represents a feature



(a) 1 principal component

(b) 5 principal component

(c) 9 principal component

(d) 13 principal component

(e) 17 principal component

(f) 21 principal component

(g) 25 principal component

(h) 29 principal component

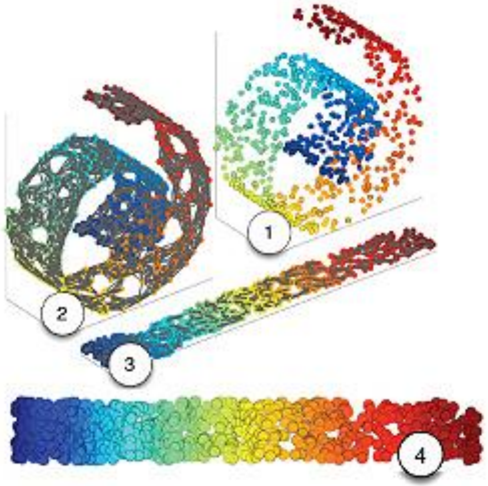
Example from https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples

Nonlinear Dimensionality Reduction

<u>Manifold Learning</u>

Finding distances between high-dimensional data and mapping to low-dimensional space

- Isomaps
- Laplacian Eigenmaps
- Self-Organizing Map (SOMs)



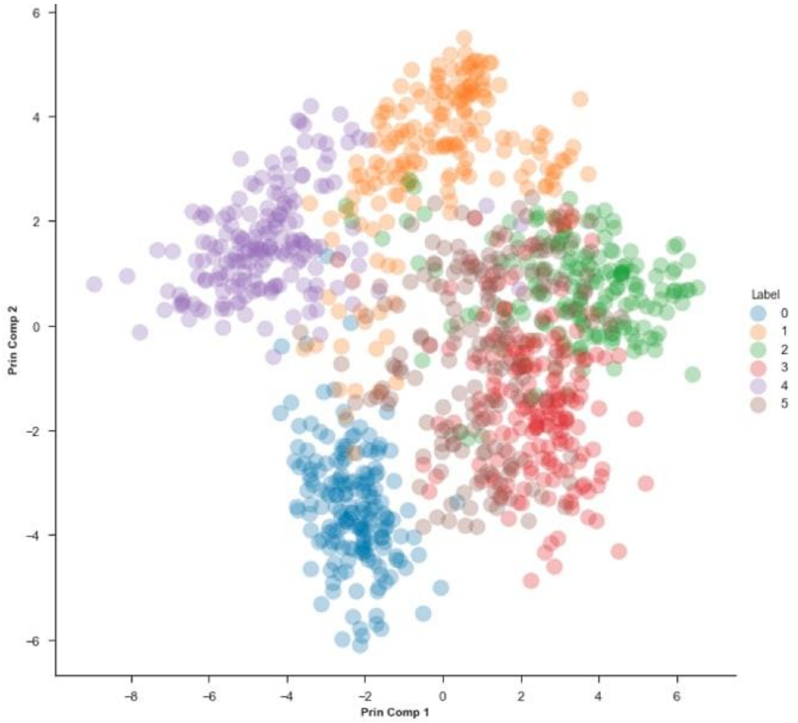
t-Distributed **S**tochastic **N**eighbor **E**mbedding

- A non-linear dimensionality reduction technique consisting of the following steps:

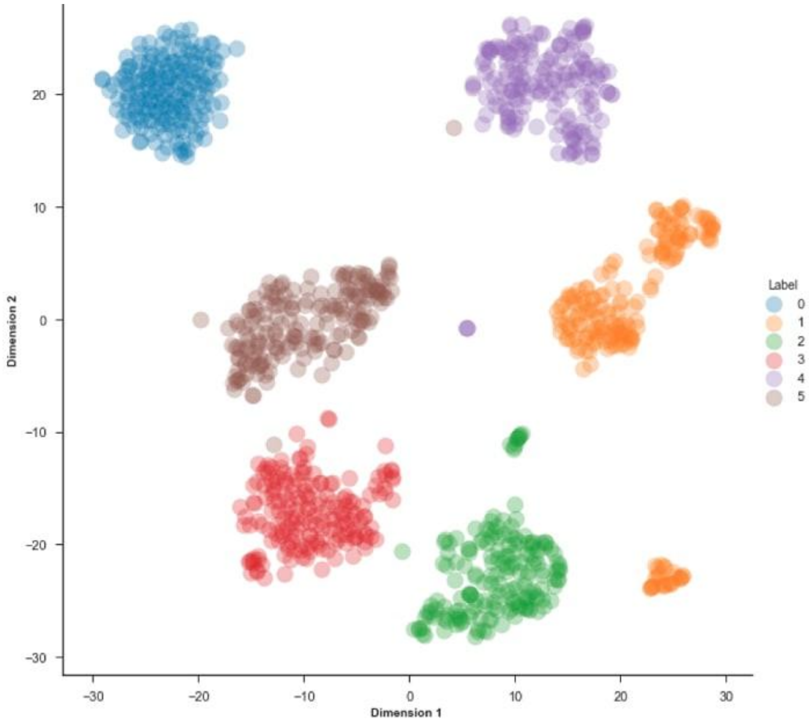
1. construct a probability distribution over pairs of high-dimensional objects in such a way that
    - similar objects have a high probability of being picked
    - dissimilar points have an extremely small probability of being picked
2. t-SNE defines a similar probability distribution over the points in the low-dimensional map, and minimizes the KL divergence between the two distributions with respect to the locations of the points in the map

Digits Dataset

(1797 8x8 images of handwritten digits)



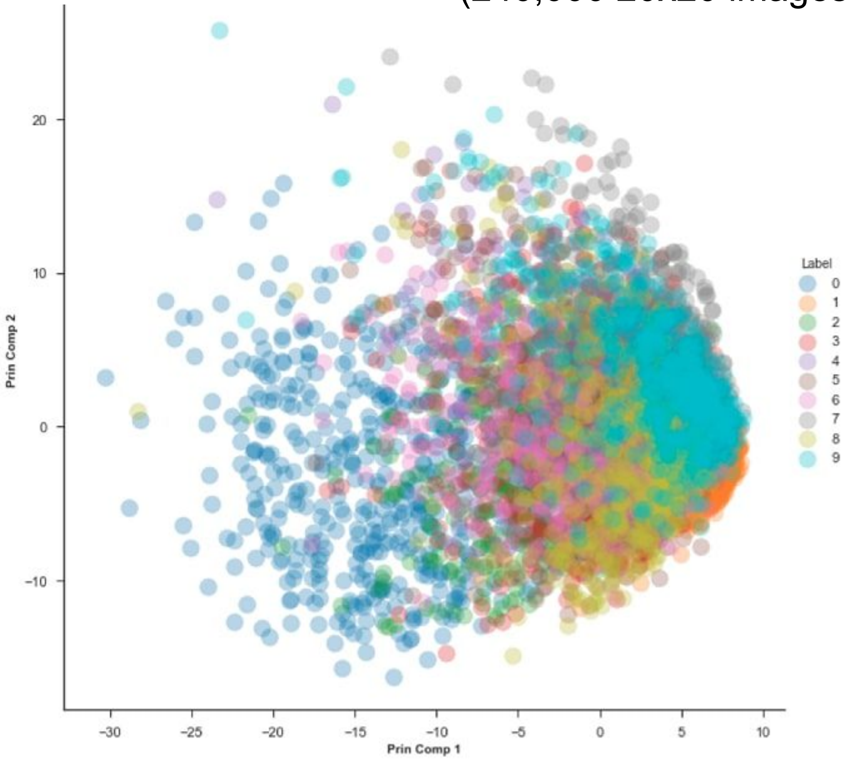
PCA



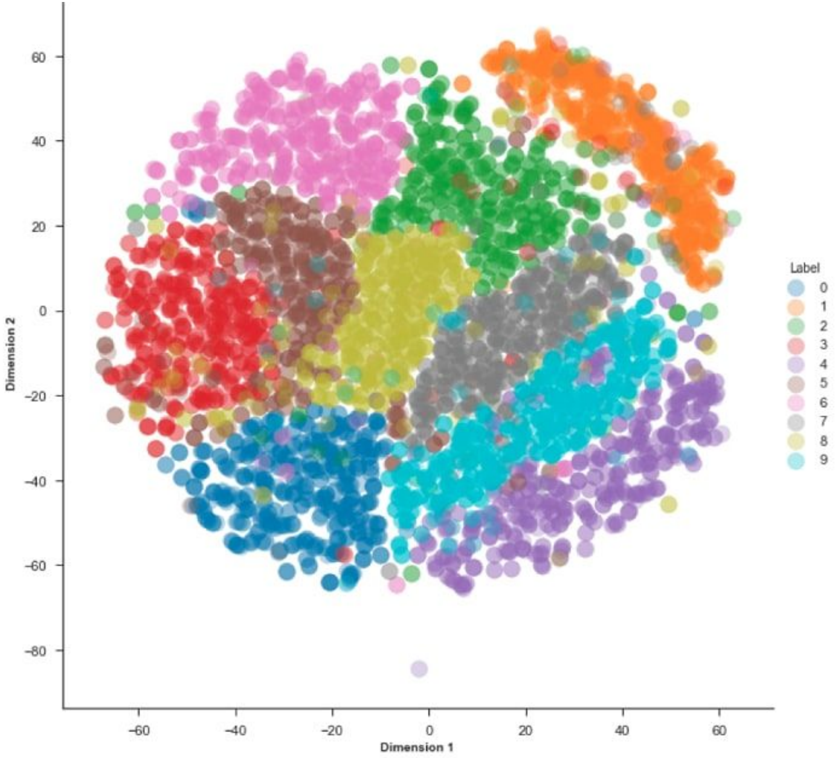
T-SNE

MNIST Dataset

(240,000 20x20 images of handwritten digits)



PCA



T-SNE