Introduction to Big Data and Machine Learning
Dimensionality Reduction
Continuous Latent Variables

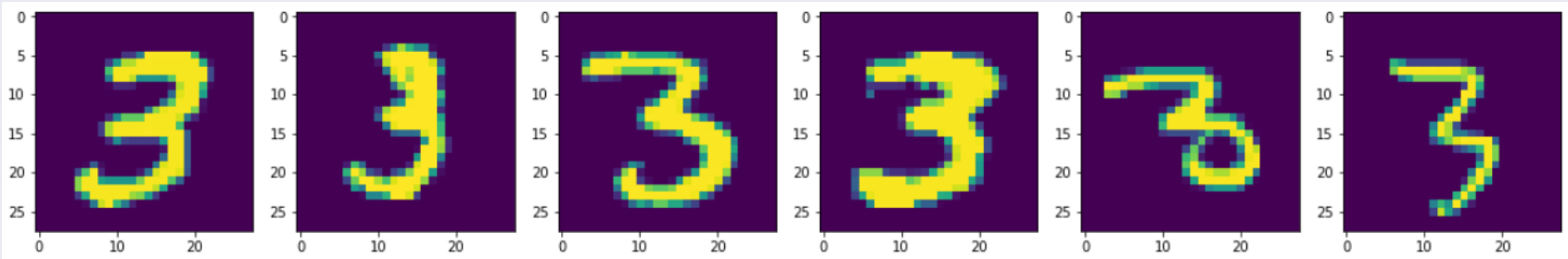
Dr. Mihail

October 8, 2019

Data Dimensionality

Idea

- Many datasets have the property that the data points all lie close to a manifold of much lower dimensionality than that of the original data space
- Consider MNIST digits



- They all lie in a 768-dimensional space, but are relatively close

Data Dimensionality

Idea

- Goal: "summarize" the ways in which the 3's (observed variables) vary with only a few continuous variables (latent variables)
- Nonprobabilistic Principal Component Analysis: express each observed variable as a projection on a lower dimensional subspace

Principal Component Analysis

Basics

- PCA is a technique widely used in dimensionality reduction, lossy data compression, feature extraction and data visualization
- Also known as the "Karhunen-Loève" transform
- There are two formulations of PCA that give rise to the same algorithm:
  1. An orthogonal projection of data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized
  2. Linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections

Maximum variance formulation

PCA derivation

- Consider a dataset of observations $\{x_n\}$ where $n = 1 \ldots N$ and x_n is a Euclidean variable with dimensionality D
- Goal: project the data onto a space with dimensionality $M < D$ while maximizing the variance of the projected data. We shall assume that M is given
- To start, we can imagine projecting on a space with $M = 1$.
- We define the direction of this 1-dimensional space with a $D-$dimensional vector u_1, such that u is a unit vector: $u_i^T u_i = 1$

Data Dimensionality

PCA derivation

- Each data point x_n is projected onto a scalar value $u_1^T x_n$.
- The mean of the projected data is $u_1^T \bar{x}$, where $\bar{x}$ is the data set mean given by:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{1}$$

and the variance of the projected data:

$$\frac{1}{N} \sum_{n=1}^{N} \{u_1^T x_n - u_1^T \bar{x}\}^2 = u_1^T S u_1 \tag{2}$$

where S is the covariance given by:

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T \tag{3}$$

PCA derivation

- We now maximize the projected variance $u_1^T S u_1$ with respect to u_1.
- Constrained maximization to prevent the naive solution $||u_1|| \rightarrow \infty$
- The appropriate constraint should be to maintain unity $||u_1^T u_1|| = 1$. To enforce, we introduce a Lagrange multiplier λ_1, and make solve unconstrained maximization of:

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1) \tag{4}$$

and setting the derivative of above to 0 w.r.t. u_1, we see that

$$S u_1 = \lambda_1 u_1 \tag{5}$$

which says that u_1 has to be an eigenvalue of S

Data Dimensionality

PCA derivation

- If we left-multiply by u_1^T and make use of $u_1^T u_1 = 1$, then the variance is given by:

$$u_1^T S u_1 = \lambda_1 \tag{6}$$

  and so the variance will be at a maximum when we set u_1 to the eigenvector with the largest eigenvalue λ_1
- This eigenvector is known as the principal component

Data Dimensionality

Summary

- PCA involves computing the mean $\bar{x}$ and the covariance matrix S of a dataset, and then finding the M eigenvectors of S corresponding to the largest eigenvalues

Data Dimensionality

Summary

- PCA involves computing the mean $\bar{x}$ and the covariance matrix S of a dataset, and then finding the M eigenvectors of S corresponding to the largest eigenvalues
- Potential concern: finding the eigenvectors and eigenvalues for a DxD matrix is $O(D^3)$.
- If we only need $M << D$ eigenvectors, there are other methods

Data Dimensionality

Minimum-error formulation of PCA

- Let the basis vectors u_i be a complete D-dimensional orthonormal set, where $i = 1 \ldots D$

Minimum-error formulation of PCA

- Let the basis vectors u_i be a complete D-dimensional orthonormal set, where $i = 1 \ldots D$
- Because this basis is complete, each data point can be represented as a linear combination of the basis vectors:

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i \tag{7}$$

  where the coefficients α_{ni} will be different for different data points
- Since the basis is orthonormal, this is a simple rotation, so the original D components $\{x_{n1}, \ldots, x_{nD}\}$ are replaced by an equivalent set $\{\alpha_{n1}, \ldots, \alpha_{nD}\}$
- Taking the inner product with u_j and making use of orthonormality, we obtain $\alpha_{nj} = x_n^T u_j$

Data Dimensionality

Minimum-error formulation of PCA

- Therefore we can now write each data point as follows:

$$x_n = \sum_{i=1}^{D} (x_n^T u_i) u_i \tag{8}$$

- Our goal is to reduce dimensionality, to an $M < D$, thus each point can be approximated by:

$$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i \textbf{??} \tag{9}$$

Minimum-error formulation of PCA

- $$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i$$

  where $\{z_{ni}\}$ depend on a particular data point, and $\{b_i\}$ are constants for all data points

- We are free to choose $\{u_i\}$, $\{z_{ni}\}$ and $\{b_i\}$ so as to minimize the distortion introduced by the reduction in dimensionality:

$$J = \frac{1}{N} \sum_{n=1}^{N} ||x_n - \tilde{x}_n||^2 \tag{10}$$

Minimum-error formulation of PCA

- Consider first $\{z_{ni}\}$. Substituting for $\tilde{x}_n$, setting the derivative wrt $z_{nj}$to zero we obtain:

$$z_{nj} = x_n^T u_j \tag{11}$$

- Similarly, setting the derivative of J with respect to b_i to zero, we obtain

$$b_j = \bar{x}^T u_j \tag{12}$$

where $j = M+1, \ldots, D$. If we substitute $z_{ni}$and b_i in Equation **??** we obtain:

$$x_n - \tilde{x}_n = \sum_{i=M+1}^{D} \{(x_n - \bar{x})^T u_i\} u_i \tag{13}$$

Minimum-error formulation of PCA

- We obtain a formulation of J, purely as a function of $\{u_i\}$:

$$J = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}(x_n^T u_i - \bar{x}^T u_i)^2 = \sum_{i=M+1}^{D} u_i^T S u_i \tag{14}$$

- The solution to the constrained minimization of J involves solving the eigenvalue problem:

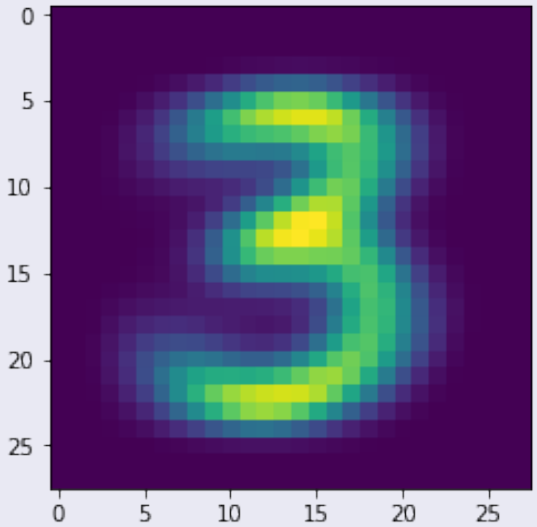
$$Su_i = \lambda_i u_i \tag{15}$$

where i=1, . . . , D and the eigenvectors are orthonormal

Data Dimensionality

PCA algorithm shown on MNIST

- Compute $\bar{x}$.



Data Dimensionality

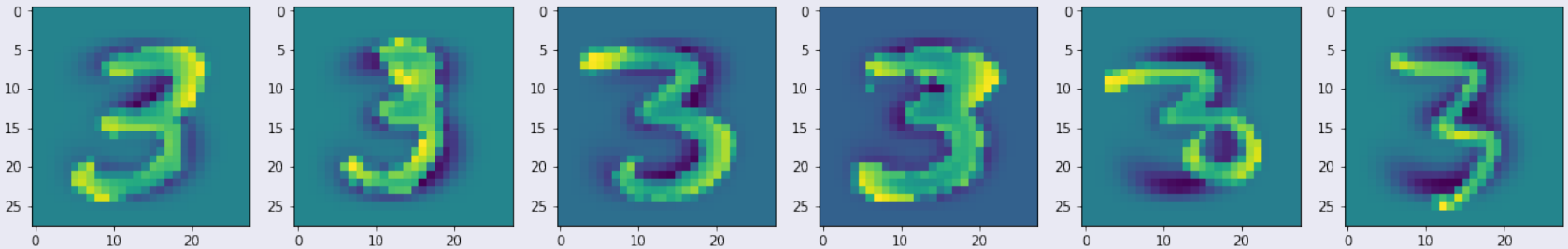
Code to finding $\bar{x}$

```
import scipy.io
mat = scipy.io.loadmat('mnist.mat')
import numpy as np
import matplotlib.pyplot as plt

X = mat['trainX'][:, :]
y = mat['trainY'][:, :][0]


threes = X[np.where(y==3)]
xbar = np.mean(threes, axis=0)
plt.subplots(1, 1)
plt.imshow(np.reshape(xbar, (28, 28)))
```

Data Dimensionality

PCA algorithm

- Subtract the mean from all x_n



```
xzeromean = threes - xbar
```

Data Dimensionality

Algorithm

- Compute the covariance matrix $x^T x$ and its eigendecomposition:

```
# Compute covariance matrix
cov_mat = xzeromean.T.dot(xzeromean) / (xzeromean.shape[0]-1)

# Compute eigenvalue decomposition
eigen_vals, eigen_vecs = np.linalg.eig(cov_mat)

# Arrange as pairs (tuples)
eig_pairs = [(eigen_vals[i], eigen_vecs[:,i]) for i in range(len(eig_vals))]

# Sort the (eigenvalue, eigenvector) tuples from high to low
eig_pairs.sort(key=lambda x: x[0], reverse=True)
```

Data Dimensionality

Project to subspace and reconstruct

```
fig, ax = plt.subplots(5, 9, figsize = (25, 15))
for digit in range(5):
  onethree = xzeromean[digit, :]
  ax[digit, 0].imshow(np.reshape(onethree+xbar, (28, 28)))
  ax[digit, 0].set_title('Original')
  for (basis_ix, basis) in enumerate([1, 2, 5, 10, 100, 200, 600, 28*28]):
    subspace = np.array([eig_pairs[i][1] for i in range(basis)]).T

    X_pca = np.dot( onethree, subspace)
    X_recon = np.dot(subspace , X_pca)  + xbar

    ax[digit, basis_ix+1].imshow(np.reshape(np.abs(X_recon), (28, 28)))
    ax[digit, basis_ix+1].set_title(str(basis)+' components')
    ax[digit, basis_ix+1].tick_params(labelbottom=False, labelleft=False)
```