HW06- CS 189

1.
Who else did you work with on this homework? In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

I worked on this homework with Ehimare Okoyomon, Prashanth Ganeth, and Daniel Mockaitis. We worked by getting together throughout the week and communicating on facebook.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up}

Nicholas Lorio, 26089160

Question 9 Own Question.

How does CCA compare to PCA ? When to use which method? Does CCA cause dimensionality reduction?

In essence, PCA is a dimensionality reduction method. It is fundamentally based off of the eckart young theorem. CCA however, is a method which compares draws from two multivariate distributions. A single quantity is found for each, picked in such a way as to make them as correlated as possible. If the two data sets have no correlation, or very little correlation than there is not a strong linear relationship between the two data sets.

CCA can be used when we have two feature spaces and want to deduce relationships from one another. It can produce a model equation which relates two sets of variables.

CCA finds mutually orthogonal pairs of maximally correlated variables, until all the multivariate variables that can be predicted are exhausted. It does not reduce dimensions per say, however it does create a space where the correlated variables amongst the two features spaces are maximized.

https://stats.stackexchange.com/questions/65692/how-to-visualize-what-canonical-correlation-analysis-does-in-comparison-to-what

189 - HW06

Nicholas Klein
26089160

2.a. $\underset{n\times d}{A} = \underset{n\times n}{U}\ \underset{n\times d}{\Sigma}\ \underset{d\times d}{V^T} =$

For each singular vector V_j, $AV_j = \sum_{i=1}^{d} \sigma_i U_i V_i^T V_j$

as any vector v can be expressed as linear combination of the singular vectors plus a vector that is perpendicular to the V_i vector,

$$\Rightarrow AV = \sum_{i=1}^{d} \sigma_i U_i V_i^T V$$

$Ax = Bx$? (and: $A = B$ iff for all vectors x $Ax = Bx$

$e_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$ using standard basis vectors e_i $\forall i \in \mathbb{R}$, if $Ae_i = Be_i$ $\forall i \in \mathbb{R}$ then the ith col A = ith col B which is $A = B$ for each i.

$$\therefore A = \sum_{i=1}^{d} \sigma_i U_i V_i^T$$

2.b. ii) $AA^T = \sum_{i=1}^{d} \sigma_i U_i V_i^T \cdot \left(\sum_{i=1}^{d} \sigma_i U_i V_i^T\right)^T$

$= \sum \sigma_i U_i V_i^T \sum \sigma_i V_i U_i^T = \sum \sigma_i U_i V_i^T \sigma_i V_i U_i^T$

$= \sum \sigma_i^2 U_i V_i^T V_i U_i^T$

$\circledast$ $V_i^T V_j = 0$ unless $i = j \Rightarrow V_i^T V_j = 1$ (same for $U_i^T U_j$)

$AA^T = \sum \sigma_i^2 U_i U_i^T$

$AA^T U_j = \sum \sigma_i^2 U_i \underbrace{U_i^T U_j}_{} \overset{\circledast}{=} \sigma_j^2 U_j$

thus U_j is an associated eigenvector of U_j

2b. i) $A^TA = \sum \sigma_i v_i u_i^T \sum \sigma_i u_i v_i^T$

$$= \sum_{i=1} \sigma_i^2 v_i \underbrace{u_i^T u_i}_{\substack{1 \\ \text{or} \\ 0}} v_i^T = \sum \sigma_i^2 v_i v_i^T$$

$$A^T A v_j = \sum \sigma_i^2 v_i \underbrace{v_i^T v_j}_{\substack{1 \\ \text{or} \\ 0}} = \sigma_j^2 v_j$$

thus v_j is an associated eigenvector of σ_j^2

2.c. $\max u^T A v = \max u^T \sum_{i=1}^{d} \sigma_i u_i v_i^T v$

$= \max \sum^{d} \sigma_i u^T u_i v_i^T v$

u is $= \sum_{d} \alpha_i u_i$
v is $= \sum^{d} \beta_i v_i$

• the Cauchy Schwarz inequality holds

$$\leq \sum^{d} \sigma_i \|u\|_2 \|u_i\|_2 \|v_i^T\|_2 \|v\|_2$$

• u is orthogonal to any other u_j $i \neq i$, same for v
• we know $\|u\|_2 = \|v\|_2 = 1$ $u = a_i u_i$ $v = b_i v_i$ $\forall i$

$$\max_{\substack{\|u\|_2=1 \\ \|v\|_2=1}} u^T A v = \max_{a_i, b_i} \sum_{i=1}^{\infty} \sigma_i a_i b_i$$

• $a_i = b_i = 1$ for i & $= 0$ for else, • $u_i u_j = 1$, $v_i v_j = 1$ if $i=j$, else $= 0$

• σ_1 is the largest, $\therefore$ to maximize we set $a_1 = b_1 = 1$ &
$u = a_1 u_1$ $v = b_1 v_1$

$$\max_{\substack{\|u\|_2=1 \\ \|v\|_2=1}} u^T A v = \sigma_1(A)$$ as needed.

• Making u takes form $u = a_1 u_1$, $a_1 = 1$ thus
$u = u_1$, which is the first left singular vector.
• Same argument can be made for v.

2.2. $E[XX^T] = \Sigma_{XX}$ $\quad E[YY^T] = \Sigma_{YY}$ $\quad E[XY^T] = \Sigma_{XY}$

Find a^TX, b^TY s.t. $\text{Max}\ \rho(a^TX, b^TY)$

$= \text{Max}\ \text{Covar}(a^TX, b^TY) \Big/ \sqrt{\text{var}(a^TX)\ \text{var}(b^TY)}$

Mean centred :. $\text{covar}(X,Y) = E[(X - E[X])(Y - E[Y])^T] = E[XY^T]$

$\text{cov}(a^TX, b^TY) = E[a^TX(b^TY)^T] = E[a^TXY^Tb] = a^TE[XY^T]b$

$\text{var}(a^TX) = E[(a^TX)^2] = a^TE[XX^T]a$ $\qquad \text{var}(b^TY) = E[b^TYY^Tb]$
$= b^TE[YY^T]b$

$= \underset{a,b \in \mathbb{R}}{\text{Max}}\ a^T\Sigma_{XY}b \Big/ (a^T\Sigma_{XX}a)^{1/2}(b^T\Sigma_{YY}b)^{1/2}$

If we scale $(\alpha a^T, \beta b^T)$ $\underset{a,b \in \mathbb{R}}{\text{Max}}$ $\dfrac{\alpha a^T\Sigma_{XY}\beta b}{(\alpha^2 a^T\Sigma_{XX}a)^{1/2}(\beta^2 b^T\Sigma_{YY}b)^{1/2}}$

the constants cancel out

3.e. Σ_{xx} Σ_{yy} } assume full rank

$$\max_{a,b \in \mathbb{R}} \frac{a^T \Sigma_{xy} b}{(a^T \Sigma_{xx} a)^{1/2} (b^T \Sigma_{yy} b)^{1/2}} =$$

$\bar{X} = \Sigma_{xx}^{-1/2} X$

$\Sigma_{xx} = I$ $\Sigma_{yy} = I$ let

symmetric	invertible
$c = \Sigma_{xx}^{1/2} a$	$a = \Sigma_{xx}^{-1/2} c$
$d = \Sigma_{yy}^{1/2} b$	$b = \Sigma_{yy}^{-1/2} d$

$$\max (\Sigma_{xx}^{-1/2} c)^T \Sigma_{xy} \Sigma_{yy}^{-1/2} d \Big/ (c^T \Sigma_{xx}^{-1/2} \Sigma_{xx} \Sigma_{xx}^{-1/2} c)^{1/2} \cdot (d^T \Sigma_{yy}^{-1/2} \Sigma_{yy} \Sigma_{yy}^{-1/2} d)^{1/2}$$

$$\max \frac{c^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} d}{(c^T c)^{1/2} (d^T d)^{1/2}}$$

Scaling gives same max, as shown earlier.

$$\text{thus} = \max_{\substack{c: \|c\|_2 = 1 \\ d: \|d\|_2 = 1}} \frac{c^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} d}{1 \cdot 1}$$

2.F. Show ρ^2 is maximum eigenvalue of

$$\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{xy}^{T} \Sigma_{xx}^{-1/2}$$

$$\rho = \max_{\substack{c: \|c\|_2 = 1 \\ d: \|d\|_2 = 1}} c^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} d \qquad \left[\rho \text{ is max singular val of } \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \right]$$

(From part c)

$A^T A$ i^{th} eigenval σ_i^2 w/ eigenvector V_i (From part b)
$A A^T$ i^{th} eigenval σ_i^2 w/ eigenvector U_i

ρ^2 is maximal of ① & ②

① $\left(\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \right) \left(\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \right)^T$

$$= \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \Sigma_{yy}^{-T/2} \Sigma_{xy}^{T} \Sigma_{xx}^{-T/2} = \boxed{\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{xy}^{T} \Sigma_{xx}^{-1/2}}$$

② $\left(\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \right)^T \left(\Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \right)$

$$= \Sigma_{yy}^{-T/2} \Sigma_{xy}^{T} \Sigma_{xx}^{-T/2} \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2}$$

2.g. $c^* = \Sigma_{xx}^{1/2} a^*$

Part 2b gives that c^* is the eigenvector corresponding to eigenvalue of the Matrix $\left(\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}\right)\left(\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}\right)^T$

Similar to above $\left(\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T\Sigma_{xx}^{-1/2}\right)$

2.h $d^* = \Sigma_{yy}^{-1/2} b^*$ From 2.b. d^* is eigenvalue of

$\left(\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}\right)^T\left(\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}\right)$

~~eigenvalue of~~ $= \left(\Sigma_{yy}^{-1/2}\Sigma_{xy}^T\Sigma_{xx}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1/2}\right)$

2.I. X,Y uncorrelated then covar $(X,Y)=0$

& thus correlation $\rho(X,Y)=0$ & Maximizing over this

quantity tells us nothing & Max ρ is not useful, ∃ No relation to maximize over.

2.J. given X, Y^2 } share linear relation $Y^2 = \alpha X$

- as long as there is linearity / linear relation we can still run CCA. let $Z = Y^2$, Max $\rho(X,Z)$ would give our CCA.
- whole point of CCA is Maximizing correlation of linear combinations of X,Y w/ eachother.

2.k. CCA, why relevant?

- allows us to find linear combinations of X_i Y_i which have max correlation w/ each other.
- It is the generalised methodology for finding the relationship between two sets of variables.

- lets us infer information between two feature spaces
- Simultaneously finds dimension reduction for two feature spaces.

4. a.

$$Bias^2(\hat{\omega}_\lambda) = \left(E[\hat{\omega}_\lambda] - w^*\right)^2$$

$$E[\hat{\omega}_\lambda] = E\left[\frac{S_{xy}}{S_x^2 + \lambda}\right]$$

$$= E\left[\sum x_i(x_i w^* + z) \Big/ S_x^2 + \lambda\right]$$

$$= E\left[(w^* S_x^2 + S_x z) / (S_x^2 + \lambda)\right]$$

$$= E\left[\frac{w^* S_x^2}{S_x^2 + \lambda}\right] + E\left[\frac{S_x z}{S_x^2 + \lambda}\right] = \frac{S_x^2}{S_x^2 + \lambda} E[w^*] + 0$$

$$Bias^2 = \left(\frac{S_x^2}{S_x^2 + \lambda} E[w^*] - w^*\right)^2 = \left(w^*\left(\frac{S_x^2}{S_x^2 + \lambda} - 1\right)\right)^2$$

4. b. $Var(\hat{\omega}_\lambda) = E[(\hat{\omega}_\lambda - E[\hat{\omega}_\lambda])^2] = E[\hat{\omega}_\lambda^2] - E[\hat{\omega}_\lambda]^2$

let $r = S_x z / S_x^2 + \lambda$

$$= E\left[\left(\hat{\omega}_\lambda + r - E[\hat{\omega}_\lambda]\right)^2\right] = E[r^2] = E\left[\left(S_x z / (S_x^2 + \lambda)\right)^2\right]$$

$$= \frac{S_x^2}{(S_x^2 + \lambda)^2} E[z^2] = \frac{S_x^2}{(S_x^2 + \lambda)^2} \cdot Var(z) = \frac{S_x^2}{(S_x^2 + \lambda)^2}$$

4. c. $\lambda \to \infty$: $Var \to 0$, $Bias^2 \to (w^*)^2$

$\lambda \to 0$: $Var \to 1/S_x^2$ (increases), $Bias^2 \to 0$

$\exists$ a tradeoff as we increase our regularization param we get better var or bias respectively.

5. a. $w \sim N(\hat{w}_\beta, \Psi)$ $\quad \varepsilon \sim N(0, I)$
prior

$$Y = Xw + \varepsilon \qquad \varepsilon = Y - Xw$$

$$w = \Psi_w^{+1/2} v + \hat{w}_\beta \qquad v \sim N(0, I)$$

$$Y = X(\Psi^{+1/2} v + \hat{w}_\beta) + \varepsilon$$

$$\underbrace{Y - X\hat{w}_\beta = X\Psi^{+1/2} v + \varepsilon}$$

Follows form of statistical prob underlying OLS

$$\hat{v} = (\Psi^{+T/2} X^T X \Psi^{+1/2} + I)^{-1} \Psi^{+T/2} X^T (Y - X\hat{w}_\beta)$$

$$\hat{w} = \hat{w}_\beta + \Psi^{1/2} \hat{v} = \hat{w}_\beta + \Psi^{+1/2} (\Psi^{T/2} X^T X \Psi^{1/2} + I)^{-1} \Psi^{T/2} X^T (Y - X$$

$$= \hat{w}_\beta + (X^T X + \Psi^{-T/2} \Psi^{-1/2})^{-1} X^T (Y - X\hat{w}_\beta)$$

$$\hat{w} = \hat{w}_\beta + (X^T X + \Psi^{-1})^{-1} X^T (Y - X\hat{w}_\beta)$$

5.6. we would want to solve the minimization problem $\min \|Y - Xw\|_2^2 + \lambda\|w - \hat{w}_B\|_2^2$

where λ is our regularizing hyperparameter.

· this method generalizes ridge regression. we encode that the solution w should be close to some $\hat{w}_B$ (our Hospital B prior info).

This can be reduced to a std ridge regression format problem

$w' = w - w_B$ $\quad \min \|X(w' + \hat{w}_B) - Y\|_2^2$ s.t. $\|w'\|_2^2 \leq \beta^2$

$Y' = Y - X\hat{w}_B$ $\quad \min \|Xw' - (Y - X\hat{w}_B)\|_2^2$ s.t. $\|w'\|_2^2 \leq \beta^2$

$\min \|Xw' - Y'\|_2^2$ s.t. $\|w'\|_2^2 \leq \beta^2$

thus

$w' = (X^T X + \lambda I)^{-1} X^T Y' =$

$\hat{w} = \hat{w}_B + (X^T X + \lambda I)^{-1} X^T (Y - X\hat{w}_B)$

· we can see which λ works best w/ our given Data.

6. a. For ridge:

$$\hat{y}_{test} = x_{test}^T \sum_{i=1}^{d} v_i \beta_i u_i^T y = x_{test}^T V \, diag\left(\frac{\sigma_i}{\lambda + \sigma_i^2}\right) U^T y$$

$$x_{test}^T \sum \frac{\sigma_i}{\lambda + \sigma_i^2} v_i u_i^T y$$

$$V \hat{w}_{ridge} = V \, diag(\,) U^T y = \sum_{i=1}^{d} diag(\sigma_i / \lambda + \sigma_i^2) v_i u_i^T y$$

therefore $\beta_i = diag\left(\frac{\sigma_i}{\lambda + \sigma_i^2}\right)$

6. b. k-PCA-OLS

8. a. $\sum_{i=1}^{k} \sigma_i u_i v_i^T$

b. $1, x_i, y_i, x_i^2, x_i y_i, y_i^2$

c. • $k(x,y) = x^T y$ • $k(x,y) = (1 + x^T y)^p$

• $k(x,y) = k_1(x,y) - k_2(x,y)$ for valid kernels k_1, k_2

d. OLS, ridge, Weighted OLS

e. • Preprocess X using k CCA rand projections

• " " " PCA w/ k CCA components

• Add ridge penalty to OLS

F. • Use of a kernel

g. • training error • Validation error • Bias

h. • Variance • Validation error

i. • Validation error

$\hat{y}_{test} = \hat{X}_{test} V \hat{w}_{k\text{-}pca\text{-}ols}$

6 b. $\hat{y}_{test} = x_{test}^T \sum_{i=1}^{d} v_i \beta_i u_i^T y$

$\hat{w}_{k\text{-}pca\text{-}ols} =$

Proj $k < d$ principle components covar Matrix $X^T X$

$\beta_i = \frac{1}{\sigma_i}$

6.c. i) λ Ridge regression ii) PCA iii) OLS

7.a. $XX^T = U\Sigma V^T V \Sigma U^T = \underset{n\times n}{U}\,\underset{n\times d}{\Sigma^2}\,\underset{n\times n}{U^T}$

U From $XX^T = U\Sigma^2U^T$?

- Since XX^T is symmetric, we can use XX^T to find the columns for the original X.

7.b. $X \in \mathbb{R}^d$ $z_j = v_j^T x_{test}$

$$X x_{test} = \begin{pmatrix} \langle x_1, x_{test} \rangle \\ \vdots \\ \langle x_n, x_{test} \rangle \end{pmatrix}$$

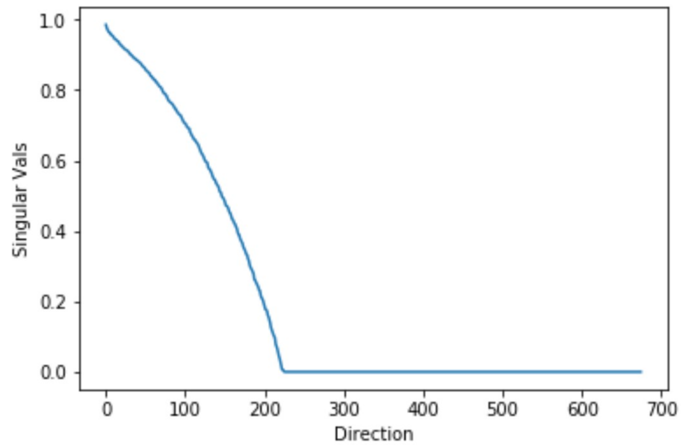
$X = U\Sigma V^T$ $V^T = (U\Sigma)^{-1} X$ v_i^T

$X_{ij} = \sum \sigma_i u_i v_i^T$, $v_j^T = (\sigma_j u_j)^{-1} X_j$

$z_j = ((U\Sigma)^{-1} X)_j\, x_{test}$

$$\boxed{z_j = \sigma_j^{-1} u_j^{-1} X_j\, x_{test}}$$

In [25]:
```
import os
import numpy as np
import cv2
import copy
import glob

import sys

from numpy.random import uniform

import pickle
from scipy.linalg import eig
from scipy.linalg import sqrtm
from numpy.linalg import inv
from numpy.linalg import svd
import numpy.linalg as LA
import matplotlib.pyplot as plt
import IPython
from sklearn.preprocessing import StandardScaler

def standardized(v):
        return (v/255.0) * 2.0 - 1.0

def flatten_and_standardize(data):
        result = []
        for d in data:
                d = d.flatten()
                d = standardized(d)
                result.append(d)
        return result

class Mooney(object):

        def __init__(self):
                self.lmbda = 1e-5

        def load_data(self):
                self.x_train = pickle.load(open('x_train.p','rb'))
                self.y_train = pickle.load(open('y_train.p','rb'))
                self.x_test = pickle.load(open('x_test.p','rb'))
                self.y_test = pickle.load(open('y_test.p','rb'))

        def compute_covariance_matrices(self):
                # USE STANDARD SCALAR TO DO MEAN SUBTRACTION
                ss_x = StandardScaler(with_std = False)
                ss_y = StandardScaler(with_std = False)

                num_data = len(self.x_train)

                x = self.x_train[0]
                y = self.y_train[0]

                x_f = x.flatten()
                y_f = y.flatten()

                x_f_dim = x_f.shape[0]
                y_f_dim = y_f.shape[0]

                self.x_dim = x_f_dim
                self.y_dim = y_f_dim

                self.C_xx = np.zeros([x_f_dim,x_f_dim])
```




```
(956, 675)

(0, 1)

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-25-fb8d34a92363> in <module>()
    269                 # COMPUTE REGRESSION
    270                 mooney.compute_projected_data_matrix(X_proj)
--> 271                 mooney.ridge_regression()
    272                 training_error = mooney.measure_error(mooney.X_ridge, mo
oney.Y_ridge)
    273                 test_error = mooney.measure_error(mooney.X_test_ridge, m
ooney.Y_test_ridge)

<ipython-input-25-fb8d34a92363> in ridge_regression(self)
    190                         print()
    191                         print(self.X_ridge.shape)
--> 192                         p_2 = np.matmul(self.X_ridge[:,i], self.Y_ridge)
    193                         ridge = np.matmul(p_1, p_2)
    194                         w_ridge.append(ridge)

ValueError: shapes (0,) and (956,675) not aligned: 0 (dim 0) != 956 (dim 0)
```

3.a

The covariance matrices for mean centered data sets are given by

$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n} x_i \cdot y_i^T = \frac{1}{n} \cdot X^T Y$

$Cov(X,X) = \frac{1}{n}\sum_{i=1}^{n} x_i \cdot x_i^T = \frac{1}{n} \cdot X^T X$

$Cov(Y,Y) = \frac{1}{n}\sum_{i=1}^{n} y_i \cdot y_i^T = \frac{1}{n} \cdot Y^T Y$