CS 189 Introduction to Machine Learning
Fall 2017 · HW5

This homework is due **Friday, September 29 at 10pm.**

1 Getting Started

You may typeset your homework in latex or submit neatly handwritten and scanned solutions. Please make sure to start each question on a new page, as grading (with Gradescope) is much easier that way! Deliverables:

1. Submit a PDF of your writeup to assignment on Gradescope, "HW[n] Write-Up"
2. Submit all code needed to reproduce your results, "HW[n] Code".
3. Submit your test set evaluation results, "HW[n] Test Set".

After you've submitted your homework, be sure to watch out for the self-grade form.

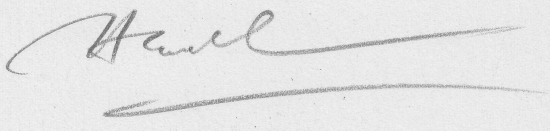
(a) Before you start your homework, write down your team. Who else did you work with on this homework? List names and email addresses. In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

None

(b) Please copy the following statement and sign next to it:

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.

I certify that all sols are entirely in my words & that I have not looks at another student's solutions. I have credited all external sources in this writeup



Problem # 2.

(a) $y = f(x) = (-|x| - 1)^2$, i.e $y = \begin{cases} (x-1)^2 & \text{if } x \geq 0 \\ (x+1)^2 & \text{if } x < 0 \end{cases}$

This function is convex.

$x \in (-\infty, 0)$: $y = (x+1)^2$ $y'' = 2 > 0$ → convex

$x \in [0, +\infty)$: $y = (x-1)^2$ $y'' = 2 > 0$ → convex

⇒ y is convex $\forall\ x \in R$

although the Hessian does not exist @ $x = 0$

It is not differentiable at $x = 0$.

$$\lim_{x \to 0^+} y' = \lim_{x \to 0^+} (x-1)^{2\prime} = \lim_{x \to 0^+} 2x - 2 = -2$$

$$\lim_{x \to 0^-} y' = \lim_{x \to 0^-} (x+1)^{2\prime} = \lim_{x \to 0^-} 2x + 2 = 2$$

$$\lim_{x \to 0^+} y' \neq \lim_{x \to 0^-} y'$$

(b) $f(x) = \|Ax - y\|^2 = (Ax - y)^T (Ax - y)$

$$= x^T A^T A x - x^T A^T y - y A x - y^2$$

$$\frac{\partial^2 x^T A^T y}{\partial x_i^2} = \frac{\partial^2 x^T A^T y}{\partial x_i \partial x_j} = 0$$ ← linear in x; ← second order derivative

$$\frac{\partial^2 y A x}{\partial x_i^2} = \frac{\partial^2 y A x}{\partial x_i \partial x_j} = 0$$

$$\frac{\partial^2 y^2}{\partial x_i^2} = \frac{\partial^2 y^2}{\partial x_i \partial x_j} = 0$$ constant y^2.

$\Rightarrow$ Hessian $\quad \nabla^2 f(x) = \nabla^2 x^T A^T A x$

$$x^T A^T A x = \sum_{i,j,k} x_i A_{ij}^T A_{jk} x_k = \sum_{i,j,k} x_i A_{ji} A_{jk} x_k$$

$$\frac{\partial^2 x^T A^T A x}{\partial x_i^2} = 2 \sum_j A_{ij}^T A_{ji} \quad \text{(if } i \neq k \text{, the derivative} = 0\text{)}$$

$$\frac{\partial^2 x^T A^T A x}{\partial x_i \partial x_k} = 2 \sum_{\substack{j \\ i \neq k}} A_{ij}^T A_{jk}$$

$$\Rightarrow \quad \nabla^2 f(x) = \begin{bmatrix} 2\sum A_{1j} A_{j1} & & 2\sum A_{ij}^T A_{jk} \\ & 2\sum A_{2j} A_{j2} & \\ 2\sum_{\substack{j \\ i \neq k}} A_{ij}^T A_{jk} & & \\ & & 2 A_{nj} A_{jn} \end{bmatrix} = 2 A^T A$$

$A^T A$ is PSD because:

$$z^T A^T A z = \sum z_i a_i^T \sum a_i z_i$$

but $\quad \sum z_i a_i^T = \left(\sum a_i z_i \right)^T$

$$\Rightarrow \quad z^T A^T A z = \left\| \sum z_i a_i^T \right\|_2^2 \geq 0 \quad \forall z$$

where a_i is the i-column of A.

(c) Take x' arbitrary, consider:

$$(1-\lambda)x_0 + \lambda x'$$

x_0 is local minimum, $\Rightarrow \forall x : |x_0 - x| \leq \varepsilon$

i.e $x_0 - \varepsilon \leq x \leq x_0 + \varepsilon$

we have: $f(x) \geq f(x_0)$

we can choose λ such that:

$$x_0 - \varepsilon \leq (1-\lambda)x_0 + \lambda x' \leq x_0 + \varepsilon \quad (*)$$

intuitively we can just choose λ to be very small value

or we can solve for λ

$$\frac{-\varepsilon}{x' - x_0} \leq \lambda \leq \frac{\varepsilon}{x' - x_0} \quad \text{if } x' > x_0 \text{ or}$$

($x' - x_0$: positive; negative)

$$\frac{\varepsilon}{x' - \varepsilon_0} \leq \lambda \leq \frac{-\varepsilon}{x' - x_0} \quad \text{if } x' < x_0$$

($x' - \varepsilon_0$: negative; $x' - x_0$: positive)

$\Rightarrow$ we can always choose $\lambda \in [0,1]$ satisfying $(*)$

$$\Rightarrow f(x_0) \leq f\left((1-\lambda)x_0 + \lambda x'\right) \leq (1-\lambda) f(x_0) + \lambda f(x')$$

(f convex)

$$f(x_0) \leq f(x_0) - \lambda f(x_0) + \lambda f(x')$$

$$f(x_0) \leq f(x') \quad \forall x'$$

This is the condition for global maximum $\rightarrow$ satisfying

d.) (i) $h(x) = f(x) + g(x)$

$$\lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2)$$

$$\lambda g(x_1) + (1-\lambda) g(x_2) \geq g(\lambda x_1 + (1-\lambda) x_2)$$

Adding side by side

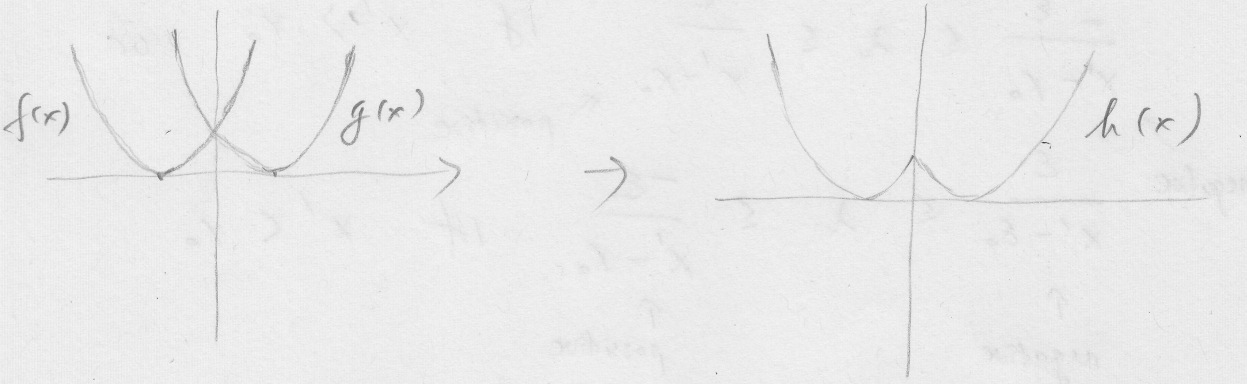
$$\lambda h(x_1) + (1-\lambda) h(x_2) \geq h(\lambda x_1 + (1-\lambda) x_2)$$

$\Rightarrow$ h convex

(ii) $h(x) = \min(f(x), g(x))$

h is not convex. Counter example:

$$f(x) = (x+1)^2 \qquad g(x) = (x-1)^2$$



Take $x_1 = -1$, $x_2 = 1$, $\lambda = 0.5$

$\Rightarrow$ $f(x_1) = 0$ $\quad g(x_1) = 4$

$\Rightarrow$ $h(x_1) = 0$

$f(x_2) = 4$ $\quad g(x_2) = 0$

$\Rightarrow$ $h(x_2) = 0$

$m = \lambda x_1 + (1-\lambda) x_2 = 0$

$f(m) = g(m) = h(m) = 1$.

But $\underbrace{\lambda h(x_1)}_{0} + \underbrace{(1-\lambda) h(x_2)}_{0} \not\geq h(\underbrace{\lambda x_1 + (1-\lambda) x_2}_{1})$ (NO)

(iii) $h(x) = \max\{f(x), g(x)\}$

$\lambda f(x_1) + (1-\lambda) f(x_2) \geq f(\lambda x_1 + (1-\lambda) x_2)$

$\lambda g(x_1) + (1-\lambda) g(x_2) \geq g(\lambda x_1 + (1-\lambda) x_2)$

Assume $f(\lambda x_1 + (1-\lambda) x_2) \geq g(\lambda x_1 + (1-\lambda) x_2)$

$\Rightarrow h(\lambda x_1 + (1-\lambda) x_2) = f(\lambda x_1 + (1-\lambda) x_2)$

$\leq \underbrace{\lambda f(x_1)}_{\leq \lambda \max(f(x_1), g(x_1)) = \lambda h(x_1)} + \underbrace{(1-\lambda) f(x_2)}_{\leq (1-\lambda)\max(f(x_2), g(x_2)) = (1-\lambda) h(x_2)}$

$\Rightarrow h(\lambda x_1 + (1-\lambda) x_2) \leq \lambda h(x_1) + (1-\lambda) h(x_2)$

$\rightarrow h$ convex

(iv) $h(x) = f(g(x))$

let $f(x) = g(x) = e^{-x}$

$f''_{xx} = g''_{xx} = e^{-x} > 0 \Rightarrow f, g$ convex

$h = f(g(x)) = e^{-e^{-x}}$

$h'_x = +e^{-x}e^{-e^{-x}} = e^{-x-e^{-x}}$

$h''_{xx} = \underbrace{(-1+e^{-x})}_{<0} e^{-x-e^{-x}} < 0$

since $e^{-x} = 1/e^x < 1$

$\Rightarrow$ h is concave

Problem # 3

(a) $$A = U\Sigma V^T = [u_1\ u_2\ \dots\ u_d] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_d \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_d^T \end{bmatrix}$$

$$= [u_1\sigma_1\ u_2\sigma_2\ \dots\ u_d\sigma_d] \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_d^T \end{bmatrix} = u_1\sigma_1 v_1^T + \dots + u_d\sigma_d v_d^T$$

$$= \sum_1^d u_i \sigma_i v_i^T$$

(b) $$A^TA = \left(\sum_1^d \sigma_i u_i v_i^T\right)^T \sum_1^d \sigma_j u_j v_j^T$$

$$= \sum_1^d \sigma_i v_i u_i^T \sum \sigma_j u_j v_j^T$$

$$= \sum_1^d \sigma_i^2 v_i v_i^T \qquad \left(u_i^T u_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \right)$$

$\Rightarrow$ eigenvalue σ_i^2

$$A^TA\, v_i = \sum_{j=1}^d \sigma_j^2 v_j \underbrace{v_j^T v_i}_{1 \text{ for } i=j} = \sigma_i^2 v_i$$

↙ column

$\Rightarrow$ v_i is eigenvector assoc. w/ σ_i^2

$$AA^T = \sum_1^d \sigma_i u_i v_i^T \left(\sum_1^d \sigma_j u_j v_j^T\right)^T$$

$$= \sum_1^d \sigma_i u_i v_i^T \sum_1^d \sigma_j v_j u_j^T$$

$$= \sum_1^d \sigma_i^2 u_i u_i^T \qquad \left(v_i^T v_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \right)$$

$\Rightarrow$ eigenvalue σ_i^2

$$AA^T u_i = \sum_{j=1}^{d} \sigma_j^2 u_j \underbrace{u_j^T u_i}_{1 \text{ for } i=j} = \sigma_i^2 u_i$$

$\Rightarrow$ u_i is eigenvector assoc. w/ σ_i^2

(c) Consider $u^T A v$ where u, v are arbitrary vectors. We can write

$$u = \alpha_1 u_1 + \alpha_2 u_2 + \dots + \alpha_d u_d$$

$$v = \beta_1 v_1 + \beta_2 v_2 + \dots + \beta_d v_d$$

where u_i, v_i are column vectors of U & V

$$\Rightarrow u^T A v = \alpha_1\beta_1 u_1^T A v_1 + \alpha_2\beta_2 u_2^T A v_2 + \dots + \alpha_d\beta_d u_d^T A v_d$$

$$+ \alpha_1\beta_2 u_1^T A u_2 + \dots$$

$$+ \alpha_2\beta_3 u_2^T A v_3 + \dots$$

$$+ \dots$$

$$= \sum_{i=1}^{d} \alpha_i\beta_i u_i^T A v_i + \sum_{\substack{i=1 \\ i\neq j}}^{d}\sum_{j=1}^{d} \alpha_i\beta_j u_i^T A v_j$$

But from (a): $A = \sum_{k=1}^{d} \sigma_k u_k v_k^T$

$$\Rightarrow \sum_{k=1}^{d}\sum_{\substack{i=1 \\ i=j}}^{d}\sum_{j=1}^{d} \alpha_i\beta_j u_i^T \sigma_k u_k v_k^T v_j = 0$$

Since: if $k \neq i$, $k \neq j$: $u_i^T u_k = v_k^T v_j = 0$

if $k = i \Rightarrow k \neq j$: $v_k^T v_j = 0$

if $k = j \Rightarrow k \neq i$: $u_k^T u_i = 0$

there is no case $k = i = j$ because $i \neq j$

$$\Rightarrow \quad u^T A u = \sum_1^d \alpha_i \beta_i \, u_i^T A v_i$$

$$= \sum_{k=1}^{d} \sum_{i=1}^{d} \alpha_i \beta_i \, u_i^T u_k \sigma_k v_k^T v_j$$

$$= \sum_{i=1}^{d} \alpha_i \beta_i \underbrace{u_i^T u_i}_{1} \sigma_i \underbrace{v_i^T v_i}_{1} \qquad \text{since } u_i^T u_k = v_k^T v_i = 0 \text{ if } i \neq k$$

$$= \sum_{i=1}^{d} \alpha_i \beta_i \sigma_i$$

we know that $u = U \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix}$

$$\|u\| = 1 \Rightarrow \|u\|^2 = 1$$

$$\Rightarrow [\alpha_1 \; \alpha_2 \cdots \alpha_d] \underbrace{U^T U}_{I} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{bmatrix} = 1$$

$$\Rightarrow \sum \alpha_i^2 = 1$$

Similarly we have $\sum \beta_i^2 = 1$

if $\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}$

$$\Rightarrow \|\alpha\| = 1 \qquad \|\beta\| = 1$$

but

$$\underbrace{\alpha^T \beta}_{\sum_1^d \alpha_i \beta_i} \leq \|\alpha\|_2 \, \|\beta\|_2 = 1$$

$$\sum_1^d \alpha_i \beta_i \leq 1$$

Now return to $u^T A u = \sum_{i=1}^{d} \alpha_i \beta_i \sigma_i \leq \sigma_{max} \sum_{i=1}^{d} \alpha_i \beta_i \leq \sigma_{max}$

thus $u^T A v \leq \sigma_{max}$ for arbitrary u, v w/ $\|u\| = \|v\| = 1$

$\Rightarrow$ $\max u^T A v = \sigma_{max} = \sigma_1(A)$

"=" happen iff $\sum_1^d \alpha_i \beta_i = 1$ (1)

and $\sum_1^d \alpha_i \beta_i = \|\alpha\|_2 \|\beta\|_2$.

$= \text{sqrt}\left(\sum_{i=1}^d \alpha_i^2 \sum_{j=1}^d \beta_j^2 \right)$ (2)

and $\alpha_i \beta_i \sigma_i = \alpha_i \beta_i \sigma_{max} \quad \forall i = \overline{1, d}$ (3)

the (3) condition happens only if

$\alpha_i \beta_i = 0$ for $i \neq 1$

$\alpha_i \beta_i = 1$ for $i = 1$

this also satisfies the conditions (1) & (2)

$\Rightarrow$ $\alpha_1 = \beta_1 = 1$

$\alpha_i = \beta_i = 0 \quad \forall i \neq 1$

$\Rightarrow$ $u = u_1 \quad v = v_1$

which are the left & right singular vectors

(d)

$$\rho = \max_{a,b \in \mathbb{R}^d} \rho(a^T X, b^T Y) = \max \frac{E[a^T X Y^T b]}{\sqrt{E[a^T X X^T a]\, E[b^T Y Y^T b]}}$$

Σ_{xy}

$$= \max_{a,b \in \mathbb{R}^d} \frac{a^T E[XY^T] b}{\sqrt{a^T E[XX^T] a \; b^T E[YY^T] b}}$$

$E[XX^T] = \Sigma_{xx}$, $E[YY^T] = \Sigma_{yy}$

$$= \max_{a,b \in \mathbb{R}^d} \frac{a^T \Sigma_{xy} b}{\sqrt{a^T \Sigma_{xx} a \; b^T \Sigma_{yy} b}}$$

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

replace a by αa, $b = \beta b$

$$\max \frac{\alpha a^T \Sigma_{xy} \beta b}{(\alpha a^T \Sigma_{xx} \alpha a)^{1/2} (\beta b^T \Sigma_{yy} \beta b)^{1/2}}$$

$$= \max \frac{\alpha\beta a^T \Sigma_{xy} b}{\alpha (a^T \Sigma_{xx} a)^{1/2} \beta (b^T \Sigma_{yy} b)^{1/2}} = \rho$$

$\Rightarrow$ if (a^*, b^*) is a maximizer, then $(\alpha a^*, \beta b^*)$ is also a maximizer for ρ

(e) i) $$\rho_{max} = \frac{a^{*T} \Sigma_{xy} b^*}{(a^{*T} \Sigma_{xx} a^*)^{1/2} (b^{*T} \Sigma_{yy} b^*)^{1/2}}$$

let $u = \Sigma_{xx}^{1/2} a \qquad v = \Sigma_{yy}^{1/2} b$

$$u^T u = a^T \Sigma_{xx} a \qquad v^T v = b^T \Sigma_{yy} b$$

$$\rho = \max \frac{a^T \Sigma_{xy} b}{(a^T \Sigma_{xx} a)^{1/2} (b^T \Sigma_{yy} b)^{1/2}} = \max \frac{a^T \Sigma_{xx}^{1/2} \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \Sigma_{yy}^{1/2} b}{(u^T u)^{1/2} (v^T v)^{1/2}}$$

(u^T over $a^T \Sigma_{xx}^{1/2}$, v over $\Sigma_{yy}^{1/2} b$)

$$= \max \frac{u^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} v}{(u^T u)^{1/2} (v^T v)^{1/2}}$$

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

$$u' = \frac{u}{(u^T u)^{1/2}} \qquad v' = \frac{v}{(v^T v)^{1/2}}$$

$$\Rightarrow \|u'\| = 1 \qquad \|v'\| = 1$$

we have

$$\rho = \max u'^T A v'$$

exactly problem (c)

$\Rightarrow \rho^2$ is the max eigen value of AA^T (or ρ is the max singular value)

$$\uparrow$$

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

ii) From (b) and (c)

u' or u is the maximal eigenvector of AA^T

↗

$\Sigma_{xx}^{1/2} a^*$ with a replaced by maximizer a^*

iii) From (b) and (c)

v' or v is the maximal eigenvector of $A^T A$

↗ ↗

$\Sigma_{YY}^{1/2} b^*$ $\Sigma_{YY}^{-1/2} \Sigma_{XY}^T \Sigma_{XX}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1/2}$

(f) when X and Y are uncorrelated

$$\Sigma_{XY} = 0$$

the problem becomes:

$$\rho = \max 0$$

⇒ we cannot defined matrix A and decompos A into SVD

⇒ "vanilla" CCA useless

If X and Y^2 are correlated → calculate

$$\Sigma_{XX} \quad \Sigma_{Y^2Y^2} \quad \Sigma_{XY^2}$$

→ proceed CCA with the modified variance matrix

when we take sample, we should square Y^2

when we get output, we take square root.

Problem # 4

a.)

$$\Sigma_{xx} = \frac{1}{n}\sum_{i=1}^{n}(X^{(i)}-\mu_{xi})(X^{(i)T}-\mu_{xi}) = E[X_s X_s^T]$$

$$\Sigma_{yy} = \frac{1}{n}\sum_{i=1}^{n}(Y^{(i)}-\mu_{yi})(Y^{(i)T}-\mu_{yi}) = E[Y_s Y_s^T]$$

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(X^{(i)}-\mu_{xi})(Y^{(i)T}-\mu_{yi}) = E[X_s Y_s^T]$$

where : n is # of sample

$X^{(i)}$ is input image i-th
$Y^{(i)}$ is output image i-th
} training

μ_{xi}, μ_{yi} are mean value of flattened $X^{(i)}$ and $Y^{(i)}$ respectively.

subscrip "s" means standardized

See appendix for coding

(b) See appendix for plot & code

(c) see appendix for image

(d) see appendix for plot and code.

(e) see appendix for image

Problem # 6

Define $\|A\|_2^2 = \sqrt{\sum_{i,j}^{n} |a_{i,j}|^2}$ $\quad A \in R^{n \times n}$

prove 1.) $\|A\|_2^2 = tr(AA^T)$

2.) Trace of a matrix is equal sum of eigen values

3.) From (1) & (2) $\Rightarrow$ $\|A\|_2^2 = \sum \sigma_i^2$

Solution

1.) $AA^T = \sum_{j=1}^{n} a_{ij} a_{jk}^T$ ← element (i,k)

$= \sum_{j=1}^{n} a_{ij} a_{kj}$

at diagonal $i = k$ $\Rightarrow$ $tr(AA^T) = \sum_{k=j=1}^{n} \sum^{n} a_{kj} a_{kj}$

$= \sum_{k,j}^{n} a_{kj}^2$

$\Rightarrow$ $tr(AA^T) = \sum_{i,j}^{n} a_{ij}^2 = \|A\|_2^2$.

2.) From definition of determinant

$\det(A - \lambda I) = (-1)^n (\lambda - a_1)(\lambda - \lambda_2) \dots (\lambda - \lambda_n)$

$= \prod_{1}^{n} \lambda_i - \lambda \sum_{1}^{n} \lambda_i + \dots$

↑ degree 1

trace of A is sum of diagonal that has first order

(since the i-th element on diagonal is $a_{ii} - \lambda$)

$\Rightarrow$ $tr(A) = \sum \lambda_i$

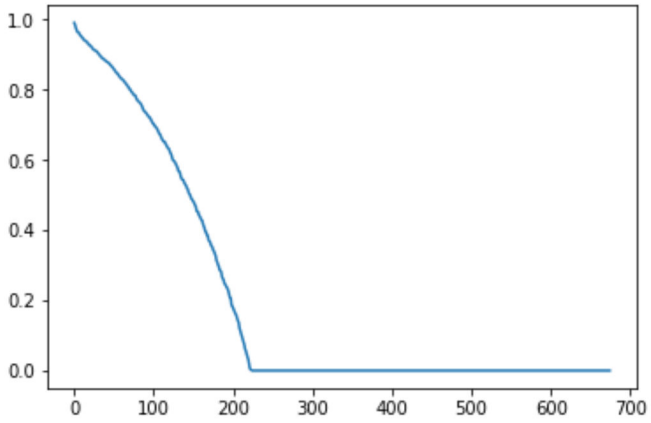
3.) From (1) and (2)

$$\Rightarrow \quad \|A\|_2 = tr(AA^T)$$

$$= \sum_{1}^{n} \sigma_i^2$$

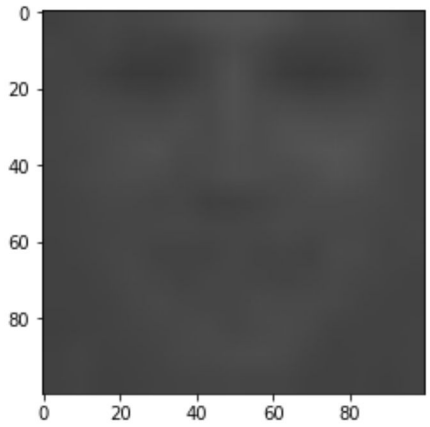
where σ_i^2 are the eigenvalues of AA^T

In [3]:
```
import pickle
import os
from scipy.linalg import eig
from scipy.linalg import sqrtm
from scipy.linalg import inv
import numpy as np
from numpy.linalg import svd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
import cv2
import random

def plot_image(vector):
    vector = ((vector + 1.0) / 2.0) * 255.0
    vector = np.reshape(vector, (15, 15, 3))
    p = vector.astype("uint8")
    p = cv2.resize(p, (100, 100))
    count = 0
    plt.imshow(p)
    plt.show()

DATA_DIR = "hw05-data"
d = 15 * 15 * 3

x_train = [((i / 255) * 2.0 - 1.0).flatten().reshape(d, 1) for i in pickle.load( op
en(os.path.join(DATA_DIR,"x_train.p"), "rb") )]
y_train = [((i / 255) * 2.0 - 1.0).flatten().reshape(d, 1) for i in pickle.load( op
en(os.path.join(DATA_DIR,"y_train.p"), "rb") )]
x_test = [((i / 255) * 2.0 - 1.0).flatten().reshape(d, 1) for i in pickle.load( ope
n(os.path.join(DATA_DIR,"x_test.p"), "rb") )]
y_test = [((i / 255) * 2.0 - 1.0).flatten().reshape(d, 1) for i in pickle.load( ope
n(os.path.join(DATA_DIR,"y_test.p"), "rb") )]

n = len(x_train)
Sxx = np.zeros( (d, d), "float" )
Syy = np.zeros( (d, d), "float" )
Sxy = np.zeros( (d, d), "float" )

for i in range(n):
    x_mean = np.mean(x_train[i])
    y_mean = np.mean(y_train[i])
    Sxx += np.matmul( (x_train[i] - x_mean), (x_train[i] - x_mean).T)
    Syy += np.matmul( (y_train[i] - y_mean), (y_train[i] - y_mean).T)
    Sxy += np.matmul( (x_train[i] - x_mean), (y_train[i] - y_mean).T)

Sxx /= n
Syy /= n
Sxy /= n

I = np.identity( d, "float")
lamb = 0.00001

M = inv(sqrtm(Sxx + lamb * I)).dot(Sxy).dot(inv(sqrtm(Syy + lamb * I)))
U, s, V = svd(M)
plt.figure()
plt.plot(s)
plt.show()
```



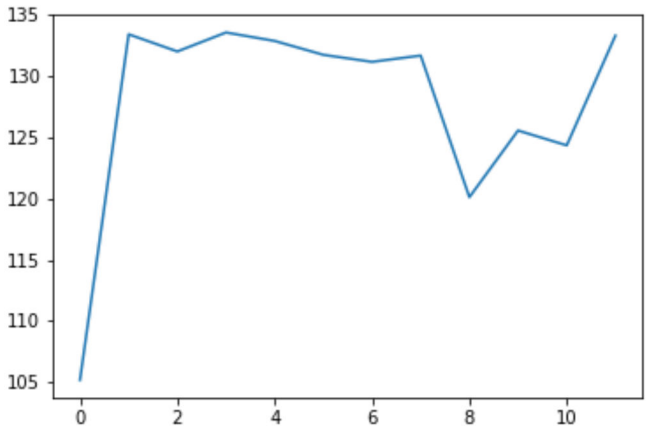
In [46]:
```
plot_image(U[:, 0] + np.mean(x_train[0]))
```

```
/home/taflab/.local/lib/python3.5/site-packages/ipykernel_launcher.py:16: Comple
xWarning: Casting complex values to real discards the imaginary part
  app.launch_new_instance()
```



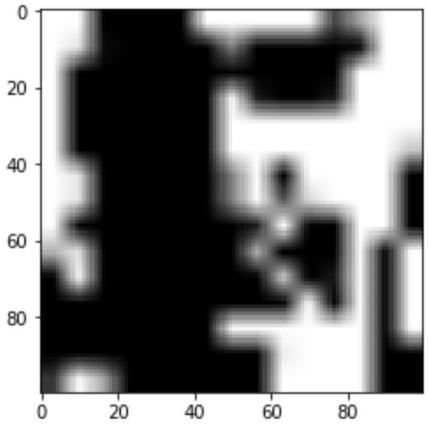
In [47]:
```
k = [0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 650]
Pk = U[:, k]

X = np.concatenate(tuple([i.T for i in x_train]))
Y = np.concatenate(tuple([i.T for i in y_train]))
XPk = np.matmul(X, Pk)

I = np.identity(XPk.shape[1])
w = np.matmul( inv( np.matmul(XPk.T, XPk) + lamb * I), np.matmul(XPk.T, Y) )

X_test = np.concatenate(tuple([i.T for i in x_test]))
Y_test = np.concatenate(tuple([i.T for i in y_test]))

error = np.zeros(len(k), "float")
for i in range(len(k)):
    Pi = Pk[:, i].reshape(675, 1)
    wi = w[i, :].reshape(1,675)
    Y_estimate = X_test.dot(Pi).dot(wi)
    error[i] = np.sum((Y_test - Y_estimate) ** 2) / len(Y_diff)

plt.figure()
plt.plot(error)
plt.show()
```

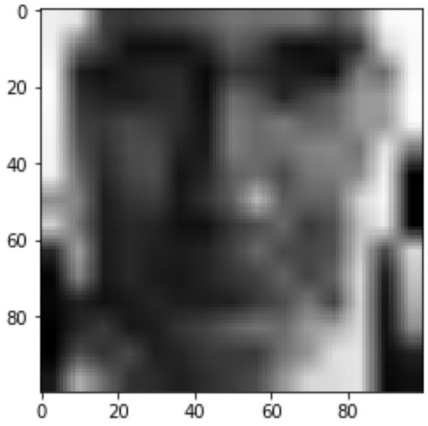
```
/home/taflab/.local/lib/python3.5/site-packages/ipykernel_launcher.py:19: Comple
xWarning: Casting complex values to real discards the imaginary part
```



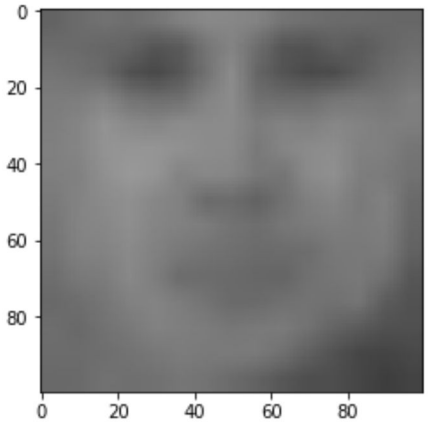
In [52]:
```
random.seed(5)
lst = [random.randint(0, 254) for _ in range(4)]
print(lst)

for i in lst:
    print("Image " + str(i))
    X = x_test[i].T
    plot_image(X)
    Y = y_test[i].T
    plot_image(Y)
    Y_predicted = X.dot(Pk).dot(w)
#     Y_predicted = X.dot(Pk[:, 3].reshape(675, 1)).dot(w[3, :].reshape(1, 675))
    plot_image(Y_predicted)
```

```
[159, 65, 189, 91]
Image 159
```

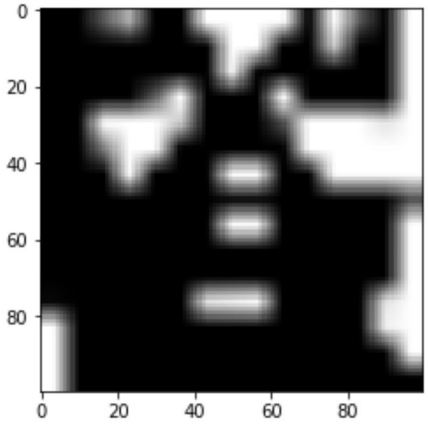


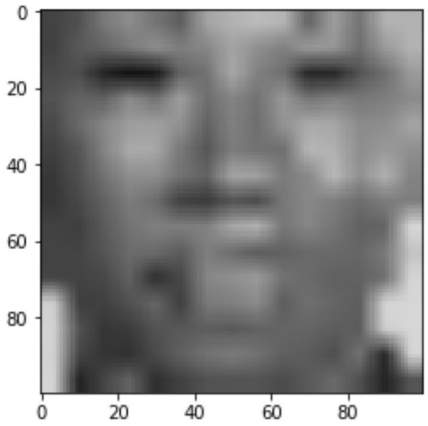


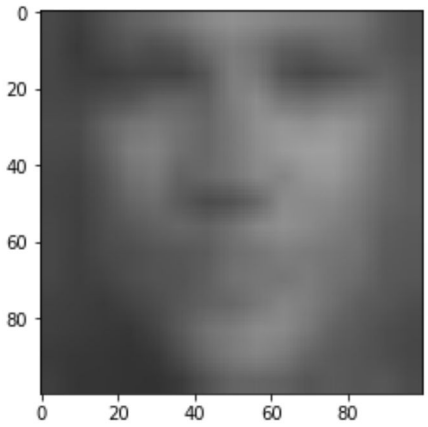

```
/home/taflab/.local/lib/python3.5/site-packages/ipykernel_launcher.py:16: Comple
xWarning: Casting complex values to real discards the imaginary part
  app.launch_new_instance()
```



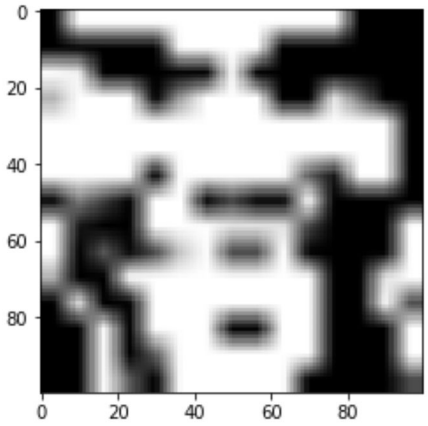

```
Image 65
```

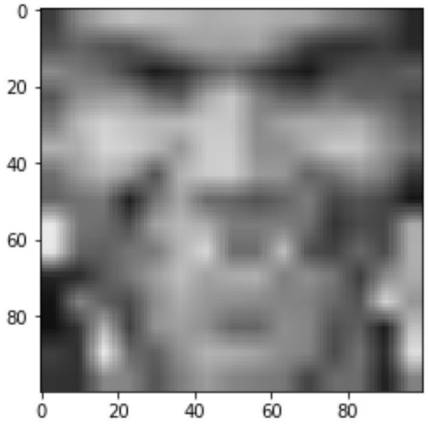

0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80

```
Image 189
```


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80

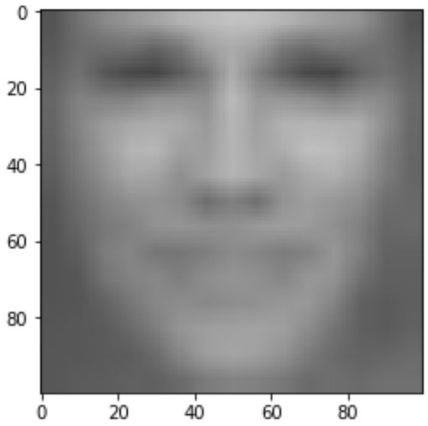
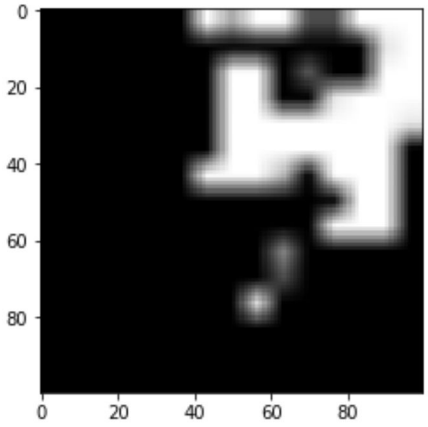
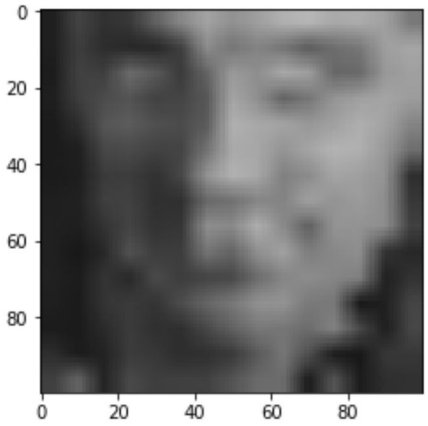
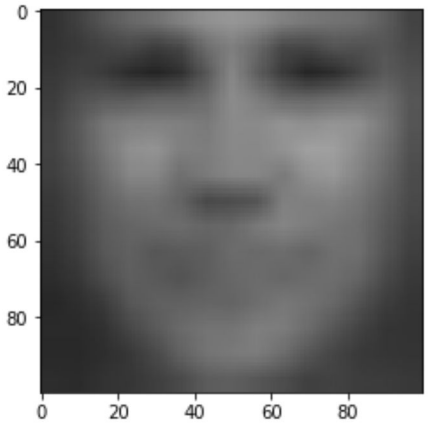

0
20
40
60
80
0
20
40
60
80

Image 91


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80


0
20
40
60
80
0
20
40
60
80