EECS 545 - Homework 2

Ding Ding

March 21, 2016

EECS 545

1) Information Theory

a). $I(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$

$= \sum_{y\in Y}\sum_{x\in X} p(x,y)\left(\log\frac{p(x,y)}{p(x)} - \log p(y)\right)$

$= \sum_{y\in Y}\sum_{x\in X} p(x,y)\left(\log p(y|x) - \log p(y)\right)$

$= -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(y) - \left(-\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(y|x)\right)$

$= H(Y) - \left(-\sum_{x\in X} p(x)\sum_{y\in Y} p(y|x)\log p(y|x)\right)$

$= H(Y) - \left(-\sum_{x\in X} p(x)\sum_{y\in Y} H(Y|X=x)\right) = H(Y) - H(Y|X)$

Similarly

$I(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\left(\log\frac{p(x,y)}{p(y)} - \log p(x)\right)$

$= \sum_{y\in Y}\sum_{x\in X} p(x,y)\left(\log p(x|y) - \log p(x)\right)$

$= -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x) - \left(-\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x|y)\right)$

$= H(X) - H(X|Y)$

b). since $X = f(Y)$, $Y = f^{-1}(X)$

we have $p(X|Y) = p(f(Y)|Y) = 1$

$p(Y|X) = p(f^{-1}(X)|X) = 1$

$\Rightarrow$ $H(Y|X) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(y|x) = 0$

$H(X|Y) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x|y) = 0$

$\Rightarrow$ due from (a) $I(X,Y) = H(Y) = H(X)$

c) $D_{KL}(\hat{p}\|q) = \sum_{x\in X}\hat{p}(x)\log\frac{\hat{p}(x)}{q(x)} = \sum_{x\in X}\hat{p}(x)\log\hat{p}(x) - \sum_{x\in X}\hat{p}(x)\log q(x) = -H(p) + H(p,q) \propto H(p,q)$

$H(p,q) = -\sum_{x\in X}\hat{p}(x)\log q(x) = -\sum_{x\in X}\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\{x=x_i\}}\log(q(x|\theta)) = -\sum_{x\in X}\frac{1}{N}\sum_{i=1}^{N}\log(q(x_i|\theta)) \propto -\sum_{i=1}^{N}\log(q(x_i|\theta))$

$\min_\theta D_{KL}(\hat{p}\|q) \Leftrightarrow \min_\theta H(p,q) \Leftrightarrow \min_\theta -\sum_{i=1}^{N}\log(q(x_i|\theta)) \Leftrightarrow \max_\theta \sum_{i=1}^{N}\log(q(x_i|\theta))$

$\Leftrightarrow \max_\theta \prod_{i=1}^{N} q(x_i|\theta)$

$\therefore$ the minimum Kullback - Leibler divergence can be obtained by maximum likelihood estimate θ_{ML} given the data D.

d). For continuous variable

we have $H(x) = -\int p(x) \ln p(x) dx$, due from the question, it has three constraint:

$$s.t \begin{cases} \int p(x)\, dx = 1 \\ \int x p(x)\, dx = \mu \\ \int (x-\mu)^2 p(x)\, dx = \sigma^2 \end{cases}$$

Use Lagrange multipliers to max $H(x)$:

$$\mathcal{L} = -\int p(x) \ln p(x) dx + \lambda_1 \left(\int p(x) dx - 1 \right) + \lambda_2 \left(\int x p(x) dx - \mu \right) + \lambda_3 \left(\int (x-\mu)^2 p(x) dx - \sigma^2 \right)$$

$$\frac{d\mathcal{L}}{dp(x)} = -\ln p(x) - 1 + \lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2 = 0$$

$$\Rightarrow \quad p(x) = e^{(-1+\lambda_1+\lambda_2 x+\lambda_3 (x-\mu)^2)}$$

$$\text{with} \begin{cases} \int e^{(-1+\lambda_1+\lambda_2 x+\lambda_3 (x-\mu)^2)} dx = 1 \\ \int x e^{(-1+\lambda_1+\lambda_2 x+\lambda_3 (x-\mu)^2)} dx = \mu \\ \int (x-\mu)^2 e^{(-1+\lambda_1+\lambda_2 x+\lambda_3 (x-\mu)^2)} dx = \sigma^2 \end{cases} \Rightarrow \quad p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$\therefore$ $p \sim N(\mu, \sigma^2)$ is the solution of max $H(x)$

$\Leftrightarrow$ $H(q) \le H(p)$ for q be any probability density with mean μ and variance σ^2

2) Dirichlet Maximum Likelihood

a) $D(p|\alpha) = \frac{\Gamma(\sum_{k=1}^m \alpha_k)}{\prod_{k=1}^m \Gamma(\alpha_k)} \prod_{k=1}^m p_k^{\alpha_k - 1} = \exp\left(\log \frac{\Gamma(\sum_{k=1}^m \alpha_k)}{\prod_{k=1}^m \Gamma(\alpha_k)} \prod_{k=1}^m p_k^{\alpha_k - 1}\right)$

$= \exp(\log\Gamma(\sum_{k=1}^m \alpha_k) - \sum_{k=1}^m \log\Gamma(\alpha_k) + \sum_{k=1}^m \log p_k^{\alpha_k - 1})$

$= \exp(\sum_{k=1}^m (\alpha_k - 1)\log p_k + \log\Gamma(\sum_{k=1}^m \alpha_k) - \sum_{k=1}^m \log\Gamma(\alpha_k))$

let $\eta(\alpha) = \begin{pmatrix} \alpha_1 - 1 \\ \vdots \\ \alpha_m - 1 \end{pmatrix}$ $T(p) = \begin{pmatrix} \log p_1 \\ \vdots \\ \log p_m \end{pmatrix}$, then $\sum_{k=1}^m (\alpha_k - 1)\log p_k = \eta(\alpha)^T T(p)$

let $A(\alpha) = -\log\Gamma(\sum_{k=1}^m \alpha_k) + \sum_{k=1}^m \log\Gamma(\alpha_k)$

$\Rightarrow D(p|\alpha) = \exp(\eta(\alpha)^T T(p) - A(\alpha))$

b). The Dirichlet log-likelihood function $F(\alpha) = \log P(D|\alpha) = \sum_{j=1}^N \log D(p^j|\alpha)$

$\therefore F(\alpha) = \sum_{j=1}^N (\sum_{k=1}^m (\alpha_k - 1)\log p_k^j - A(\alpha))$

$= \sum_{k=1}^m (\alpha_k - 1) \sum_{j=1}^N \log p_k^j - NA(\alpha)$

$= N\sum_{k=1}^m (\alpha_k - 1) \frac{1}{N}\sum_{j=1}^N \log p_k^j + N(\log\Gamma(\sum_{k=1}^m \alpha_k) - \sum_{k=1}^m \log\Gamma(\alpha_k))$

$= N(\sum_{k=1}^m (\alpha_k - 1)\hat{t}_k + (\log\Gamma(\sum_{k=1}^m \alpha_k) - \sum_{k=1}^m \log\Gamma(\alpha_k)))$ where $\hat{t}_k = \frac{1}{N}\sum_{j=1}^N \log p_k^j$

c). $\frac{\partial F}{\partial \alpha_k} = N(\hat{t}_k + \frac{\partial \log\Gamma(\sum_{k=1}^m \alpha_k)}{\partial \sum_{k=1}^m \alpha_k} \cdot 1 - \frac{\partial \log\Gamma(\alpha_k)}{\partial \alpha_k})$

$= N(\hat{t}_k + \Psi(\sum_{k=1}^m \alpha_k) - \Psi(\alpha_k))$

$\nabla F = (\frac{\partial F}{\partial \alpha_1} \cdots \frac{\partial F}{\partial \alpha_m})$

d)

$\frac{\partial F}{\partial \alpha_k^2} = N(\Psi'(\sum_{k=1}^m \alpha_k) - \Psi'(\alpha_k))$

$\frac{\partial F}{\partial \alpha_k \partial \alpha_j} = N\Psi'(\sum_{k=1}^m \alpha_k)$

$\Rightarrow q_{kk} = -N\Psi'(\alpha_k)$ for $\forall k = 1, \dots m$.

$c = N\Psi'(\sum_{k=1}^m \alpha_k)$

we have $\nabla_\alpha^2 F(\alpha) = (\frac{\partial F}{\partial \alpha_k \partial \alpha_j})_{k,j \in (1,\dots m)} = Q + c11^T$

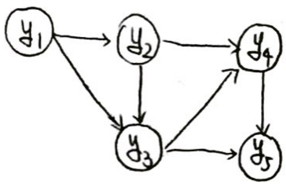
e) $\alpha^{new} = \alpha^{old} - [H_F(\alpha^{old})]^{-1} \nabla F(\alpha^{old})$

$= \alpha^{old} - [Q + c11^T]^{-1} \cdot \nabla F(\alpha^{old})$

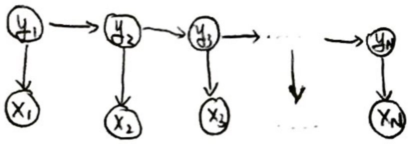
$= \alpha^{old} - \left(Q^{-1} - \frac{Q^{-1} c 1 1^T Q^{-1}}{1 + 1^T Q^{-1} c 1} \right) \cdot \nabla F \alpha^{old}$

3) Graphical Models

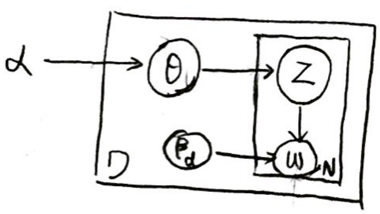
a) i) $P(y_1, y_2, y_3, y_4, y_5) = P(y_1) P(y_2 | y_1) \prod_{k=3}^{5} P(y_k | y_{k-1}, y_{k-2})$



ii) $P(x_1, \ldots, x_N, y_1, \ldots, y_N) = P(y_1) \prod_{k=2}^{N} P(y_k | y_{k-1}) \prod_{k=1}^{N} P(x_k | y_k)$



b).



c) ① $P(r, \theta, \phi, z) = \prod_{i=1}^{M} P(r_i) P(\theta_i | r_i) \prod_{j=1}^{N} P(\phi_{ij}) P(z_{ij} | \phi_{ij})$

② $P(z, w) = \prod_{i=1}^{M} P(z_i) \prod_{j=1}^{N} P(w_{ij} | z_i)$

2.f

```
### from __future__ import division
import numpy as np
from scipy.special import gammaln, polygamma
from matplotlib import pyplot as plt
# Generate the data according to the specification in the homework description
N = 1000
m = 5
alpha = np.array([10, 5, 15, 20, 50])
P = np.random.dirichlet(alpha, N)
t=((np.log(P)).sum(axis=0))/N
#loglikelihood with true parameter
F=N*(np.dot((alpha-1),t)+(gammaln(np.sum(alpha))-np.sum(gammaln(alpha))))
print F
#Newton method
alpha0=np.array([1,1,1,1,1])
F1=[0,1]
i=0
while np.abs(F1[i+1]-F1[i]) >=10**(-4):
    G=N*(t+polygamma(0,np.sum(alpha0))-polygamma(0,alpha0))
    #Hessian
    Q=np.diag((-N)*polygamma(1,alpha0))
    b=np.ones((5,5))
    bt=np.ones((5,1))
    c=N*polygamma(1,np.sum(alpha0))
    H=np.linalg.inv(Q)-(c*(np.linalg.inv(Q).dot(b).dot(np.linalg.inv(Q))))/(1+(c*(np.dot(np.dot(bt.T,np.linalg.inv(Q)),bt))))

    #update
    alpha0=alpha0-np.dot(H,G)
    F1=np.append(F1,N*(np.dot((alpha0-1),t)+(gammaln(np.sum(alpha0))-np.sum(gammaln(alpha0)))))
    i=i+1
F2=F1[2:]
M=np.arange(i)
plt.plot(M,F2,"-g",(0,i),(F,F),"-b")
plt.show()
```

The plot of the log-likelihood as a function of iteration number:

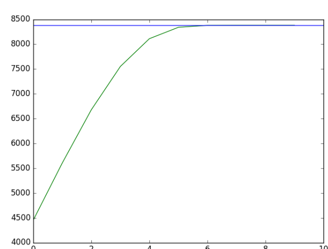


Figure 1: Iteration VS log-likelihood

The estimated model parameters are:

$alpha = (9.7913496, 4.7818326, 14.56371231, 19.34694717, 48.46071097)$

4 Clustering

a

```
from __future__ import division
from scipy.ndimage import imread
import numpy as np
from matplotlib import pyplot as plt
# Load the mandrill image as an NxNx3 array. Values range from 0.0 to 255.0.
mandrill = imread('mandrill.png', mode='RGB').astype(float)
N = int(mandrill.shape[0])

M = 2
k = 64

# Store each MxM block of the image as a row vector of X
X = np.zeros((N**2//M**2, 3*M**2))
for i in range(N//M):
    for j in range(N//M):
        X[i*N//M+j,:] = mandrill[i*M:(i+1)*M,j*M:(j+1)*M,:].reshape(3*M**2)
plt.imshow(mandrill/255)
plt.show()
import copy
import random
mu=np.array(random.sample(X,64))
J1=0
J2=1
J=0
while np.abs(J1-J2)>0:
    J1=copy.copy(J2)
    r=np.zeros((X[:,1].size,k))
    error=np.zeros(X[:,1].size)
    for i in range(X[:,1].size):
        a=np.argmin((np.sum((X[i,:]-mu)**2,1)))
        r[i,a]=1
        error[i]=np.linalg.norm(X[i,:]-mu[a,:])**2
    mu=np.dot(r.T,X)/(np.sum(r,0).reshape(64,1))
    J2=np.sum(error)
    J=np.append(J,J2)
J
M=np.arange(J.size)
plt.plot(M,J)
plt.show()
xnew=np.zeros(X.shape)
for i in range(X[:,1].size):
    j=r[i]
    xnew[i,:]=mu[np.argmax(r[i,:]),:]
print xnew
mandrill1=np.zeros(512*512*3).reshape(512,512,3)
for i in range(256):
    for j in range(256):
        mandrill1[i*2:(i+1)*2,j*2:(j+1)*2,:]=xnew[i*256+j,:].reshape(2,2,3)
plt.imshow(mandrill1/255)
plt.show()
d=mandrill-mandrill1
plt.imshow((d+128)/255)
```

```
plt.show()
error=np.sum(np.abs(d))/(3*N*N*255)
```

The plot of the k-means objective function value versus iteration number:

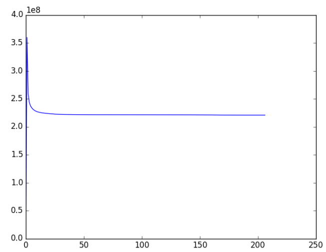


Figure 2: Iteration VS K-means objective function

The original picture and the compressed picture shown as follow:

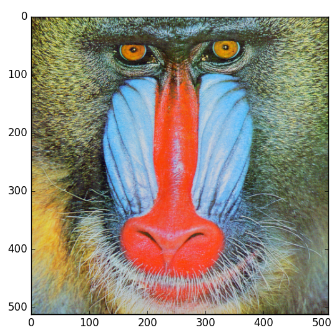


Figure 3: Original picture

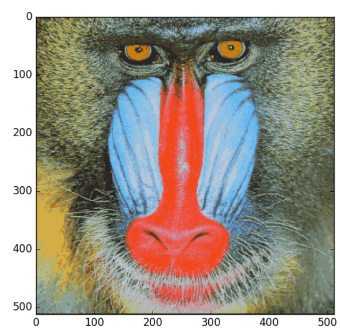


Figure 4: Compressed Picture

As show, the pure color has been well preserved which means those large area with single color are regions that are best preserved. For those area that change from one color to another, they are not well preserved.

The picture of the difference is

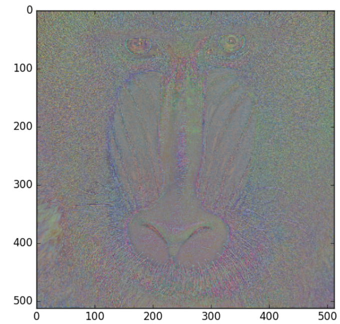


Figure 5: Difference of the two image

With M=2,K=64,N=512,The compress ratio is:

$$\frac{24 * M^2 * K + \frac{N^2}{M^2} * log_2 K}{24 * N^2} = 0.06347$$

The relative mean absolute error of the compressed image:0.050024957996999374

b The number of the bits per pixel need to store the compressed image is:

$$\frac{24 * M^2 * K + \frac{N^2}{M^2} * log_2 K}{N^2}$$

The compressed ratio of bpp is:

$$\frac{24 * M^2 * K + \frac{N^2}{M^2} * log_2 K}{24 * N^2}$$

5) EM Algorithm for Mixed Linear Regression

a). $f(y \mid X; \theta) = \sum_{k=1}^{K} \pi_k \phi(y; w_k^T x + b_k, \sigma_k^2)$ $\Rightarrow$ $f(\underline{y} \mid \underline{x}; \theta) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \phi(y_i; w_k^T x_i + b_k, \sigma_k^2)$.

$\log f(\underline{y} \mid \underline{x}; \theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \phi(y_i; w_k^T x_i + b_k, \sigma_k^2)$

b). $f(y \mid x, z=k; \theta) = \phi(y; w_k^T x + b_k, \sigma_k^2)$ $\quad f(y, z \mid x; \theta) = \pi_z \phi(y; w_z^T x + b_z, \sigma_z^2)$

set $\Delta_{ik} = \mathbb{1}_{\{z_i = k\}}$

$\log(f(\underline{y}, \underline{z} \mid \underline{x}; \theta)) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{1}_{\{z_i = k\}} \log \pi_{z_i} \phi(y_i; w_{z_i}^T x_i + b_{z_i}, \sigma_{z_i}^2) = \sum_{k=1}^{K} \sum_{i=1}^{N} \Delta_{ik} \log \pi_k \phi(y_i; w_k^T x_i + b_k, \sigma_k^2)$

c). $Q(\theta, \theta^{old}) = E_Z[\log f(\underline{y}, \underline{z} \mid \underline{x}; \theta) \mid \underline{y}, \underline{x}; \theta^{old}]$

let $r_{ik} = P(z_i = k \mid y_i, x_i, \theta^{old})$

$$r_{ik} = \frac{f(z_i, y_i \mid x_i, \theta^{old})}{f(y_i \mid x_i, \theta^{old})} = \frac{\pi_k^{old} \phi(y_i, w_k^{T\,old} x_i + b_k^{old}, \sigma_k^{2\,old})}{\sum_{k=1}^{K} \pi_k^{old} \phi(y_i, w_k^{T\,old} x_i + b_k^{old}, \sigma_k^{2\,old})}$$

$$Q(\theta, \theta^{old}) = \sum_{i=1}^{N} \sum_{k=1}^{K} E_q[\mathbb{1}_{\{z_i = k\}} \log \pi_k \phi(y, w_k^T x + b_k, \sigma_k^2)]$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \pi_k \phi(y, w_k^T x + b_k, \sigma_k^2)$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \log \phi(y, w_k^T x + b_k, \sigma_k^2)$$

d). $\theta^{new} = \arg\max_\theta Q(\theta, \theta^{old})$ s.t $\sum_{k=1}^{K} \pi_k = 1$, $\forall \pi_k > 0$ for $k = 1, \cdots K$.

let $\mathcal{L} = Q(\theta, \theta^{old}) + \lambda(\sum_{k=1}^{K} \pi_k - 1) + \sum_{k=1}^{K} \alpha_k \pi_k$ $\quad \alpha_k = 0, \forall k = 1, \cdots k$

$\frac{\partial \mathcal{L}}{\partial \pi_k} = \sum_{i=1}^{N} \frac{r_{ik}}{\pi_k} + \lambda = 0$ $\Rightarrow$ $\pi_k = \frac{\sum_{i=1}^{N} r_{ik}}{-\lambda}$

$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{k=1}^{K} \pi_k - 1 = 0$ $\Rightarrow$ $\sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} / -\lambda - 1 = 0$ $\Rightarrow$ $\lambda = -N$ $\rightarrow$ $\pi_k = \frac{\sum_{i=1}^{N} r_{ik}}{N}$

$$\frac{\partial \mathcal{L}}{\partial w_k} = \sum_{i=1}^{N} r_{ik} \frac{\partial(\log \phi(y, w_k^T x + b_k, \sigma_k^2))}{\partial w_k} = \sum_{i=1}^{N} r_{ik} \frac{\partial\left(\log \frac{1}{\sqrt{2\pi}\sigma_k} + \frac{(y_i - (w_k^T x_i + b_k))^2}{2\sigma_k}\right)}{\partial w_k} = \sum_{i=1}^{N} r_{ik} \frac{2(y_i - (w_k^T x_i + b_k))}{2\sigma_k}(-x_i) = 0$$

$$\frac{\partial \mathcal{L}}{\partial b_k} = \sum_{i=1}^{N} r_{ik} \frac{\partial\left(\log \frac{1}{\sqrt{2\pi}\sigma_k} + \frac{(y_i - (w_k^T x_i + b_k))^2}{2\sigma_k}\right)}{\partial b_k} = \sum_{i=1}^{N} r_{ik} \frac{y_i - (w_k^T x_i + b_k)}{\sigma_k}(-1) = 0$$

set $\tilde{w}_k = \begin{bmatrix} w_k \\ b_k \end{bmatrix}$, $\tilde{X} = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} = [X \ 1]$, $y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$

then, we can combine $\frac{\partial \mathcal{L}}{\partial w_k}$ & $\frac{\partial \mathcal{L}}{\partial b_k}$ get $\sum_{i=1}^{N} r_{ik}(y_i - \tilde{w}_k^T \tilde{x}_i)\tilde{x}_i = 0$

use closed form to represent the whole question, we can set the objective function as

$\sum_{i=1}^{N} -r_{ik}(y_i - \tilde{w}_k^T \tilde{x}_i)^2$ can be write as $(\tilde{X}\tilde{w}_k - y)^T R (\tilde{X}\tilde{w}_k - y)$ where $R = diag(r_{1k}, r_{2k}, \cdots r_{Nk})$

$$\therefore \sum_{i=1}^{N} r_{ik}(y_i - \tilde{w}_k^T \tilde{x}_i)\,\tilde{x}_i = 2X^TRXw - 2X^TRy = 0$$

$$\Rightarrow \quad \tilde{w}_k = (X^TRX)^{-1}X^TRy$$

$$\frac{\partial f}{\partial \sigma_k^2} = \frac{\partial \sum_{i=1}^{N} r_{ik}\left(\log\frac{1}{\sqrt{2\pi\sigma_k^2}} + \left(-\frac{1}{2\sigma_k}(y_i - \tilde{w}_k^T \tilde{x}_i)^2\right)\right)}{\partial \sigma_k^2}$$

$$= \sum_{i=1}^{N} r_{ik} + \frac{1}{\sigma_k^2}\sum_{i=1}^{N}(y_i - \tilde{w}_k^T \tilde{x}_i)^2 r_{ik} = 0$$

$$\Rightarrow \quad \sigma_k^2 = \frac{1}{\sum_{i=1}^{N} r_{ik}} \sum_{i=1}^{N}(y_i - \tilde{w}_k^T \tilde{x}_i)^2 r_{ik}$$

$$= \frac{1}{\sum_{i=1}^{N} r_{ik}} (\tilde{X}\tilde{w}_k - y)^T R (\tilde{X}\tilde{w}_k - y)$$

5.e EM Algorithm for Mixed Linear Regression

```
from __future__ import division
import numpy as np
from matplotlib import pyplot as plt

# Generate the data according to the specification in the homework description
N = 500
x = np.random.rand(N).reshape(500,1)
pi0 = np.array([0.7, 0.3])
w0 = np.array([-2, 1]).reshape(1,2)
b0 = np.array([0.5, -0.5]).reshape(1,2)
sigma0 = np.array([.4, .3])

y = np.zeros_like(x)
for i in range(N):
    k = 0 if np.random.rand() < pi0[0] else 1
    y[i] = w0[0,k]*x[i] + b0[0,k] + np.random.randn()*sigma0[k]
plt.scatter(x, y, c='r', marker='x')
plt.show()
#rebuilt the data
N=500
Intercept=np.ones((y.size,1))
x1=np.concatenate((x,Intercept),axis=1)
#w1=np.array([1,-1]).reshape(1,2)
#b1=np.array([0,0]).reshape(1,2)
wt1=np.array([1,0]).reshape(2,1)
wt2=np.array([-1,0]).reshape(2,1)
pi1=np.array([0.5,0.5])
#wt=np.concatenate((w1,b1))
sigma1=np.array([np.std(y),np.std(y)])
from scipy.stats import norm
R1=np.zeros(N)
R2=np.zeros(N)
for i in range(N):
    R1[i]=pi1[0]*(norm.pdf(y[i],np.dot(x1[i,:].reshape(1,2),wt1),sigma1[0]))/(pi1[0]*(norm.pdf(y[i],np.
     dot(x1[i,:].reshape(1,2),wt1),sigma1[0]))+pi1[1]*(norm.pdf(y[i],np.dot(x1[i,:].reshape(1,2),wt2),
     sigma1[1])))
    R2[i]=1-R1[i]
#maxlikelihood
def lg(pi1,w1,w2,x1,sigma):
    a=0
    for i in range(N):
        a=a+np.log(pi1[0]*(norm.pdf(y[i],np.dot(x1[i,:].reshape(1,2),w1),sigma[0]))+pi1[1]*(norm.pdf(y[i
     ],np.dot(x1[i,:].reshape(1,2),w2),sigma[1])))
    return a
#update
def pi(R1,R2,N):
    b=np.zeros(2)
    b[0]=np.sum(R1)/N
    b[1]=np.sum(R2)/N
    return b

def w(R1,R2,x1,y):
    R11=np.diag(R1)
```

```
    R22=np.diag(R2)
    w11=np.dot(np.dot(np.dot(np.linalg.inv(np.dot(np.dot(x1.T,R11),x1)),x1.T),R11),y)
    w12=np.dot(np.dot(np.dot(np.linalg.inv(np.dot(np.dot(x1.T,R22),x1)),x1.T),R22),y)
    return w11,w12

def s(R1,R2,y,w1,w2,x1):
    s=np.zeros(2)
    R11=np.diag(R1)
    R22=np.diag(R2)
    s[0]=np.sqrt((x1.dot(w1)-y).T.dot(R11).dot(x1.dot(w1)-y)/np.sum(R1))
    s[1]=np.sqrt((x1.dot(w2)-y).T.dot(R22).dot(x1.dot(w2)-y)/np.sum(R2))
    return s
j=0
lg0=lg(pi1,wt1,wt2,x1,sigma1)
pi1=pi(R1,R2,N)
wt1,wt2=w(R1,R2,x1,y)
sigma1=s(R1,R2,y,wt1,wt2,x1)
lg1=np.append(lg0,lg(pi1,wt1,wt2,x1,sigma1))
while np.abs(lg1[j+1]-lg1[j])>10**(-4):
    for i in range(N):
        R1[i]=pi1[0]*(norm.pdf(y[i],np.dot(x1[i,:].reshape(1,2),wt1),sigma1[0]))/(pi1[0]*(norm.pdf(y[i],
    np.dot(x1[i,:].reshape(1,2),wt1),sigma1[0]))+pi1[1]*(norm.pdf(y[i],np.dot(x1[i,:].reshape(1,2),wt2),
    sigma1[1])))
        R2[i]=1-R1[i]
    pi1=pi(R1,R2,N)
    wt1,wt2=w(R1,R2,x1,y)
    sigma1=s(R1,R2,y,wt1,wt2,x1)
    lg1=np.append(lg1,lg(pi1,wt1,wt2,x1,sigma1))
    j=j+1
plt.plot(M,lg1)
plt.show()
print pi1,wt1,wt2,sigma1
plt.scatter(x, y, c='r', marker='x')
plt.scatter(x,np.dot(x1,wt1))
plt.scatter(x,np.dot(x1,wt2))
plt.show()
```

The plot of the log-likelihood as a function of iteration number:

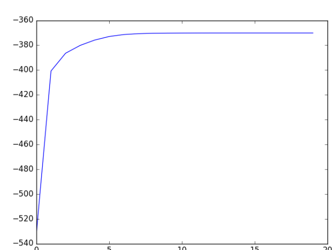


Figure 6: Iteration VS log-likelihood

The estimated model parameters:

$\pi = [0.69393268, 0.30606732]$

$w = [-1.95295001, 1.00135633]$

$b = [0.50580164, -0.50474549]$

$\sigma = [0.39743446, 0.30613885]$

The plot show the data and estimated line together:

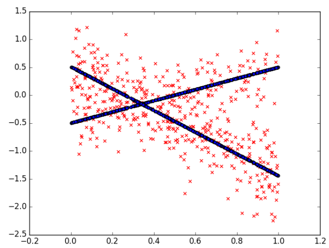


Figure 7: data&Estimated line