Exercise for Lecture 11

Q.1.

(a). $p(z) = \prod_{c \in \{a,b,c\}} \pi_c^{\mathbb{1}(z=c)}$

(b). $P(x|z) = \prod_{c \in \{a,b,c\}} \mathcal{N}(x|\mu_c, \Sigma_c)^{\mathbb{1}(z=c)}$

(c). $P(x_i) = \pi_a \mathcal{N}(x_i|\mu_a, \Sigma_a) + \pi_b \mathcal{N}(x_i|\mu_b, \Sigma_b) + \pi_c \mathcal{N}(x_i|\mu_c, \Sigma_c)$

(d). $L(x|\mu, \Sigma, \pi) = \prod_{n=1}^{N} p(x_n)$

$= \prod_{n=1}^{N} \left[\pi_a \mathcal{N}(x_i|\mu_a, \Sigma_a) + \pi_b \mathcal{N}(x_i|\mu_b, \Sigma_b) + \pi_c \mathcal{N}(x_i|\mu_c, \Sigma_c) \right]$

(e). $l = \sum_{n=1}^{N} \log \left[\pi_a \mathcal{N}(x_i|\mu_a, \Sigma_a) + \pi_b \mathcal{N}(x_i|\mu_b, \Sigma_b) + \pi_c \mathcal{N}(x_i|\mu_c, \Sigma_c) \right]$

(f). $\gamma(a) = P(z=a|x) = \dfrac{\pi_a \mathcal{N}(x_i|\mu_a, \Sigma_a)}{P(x_i)}$ → (from ©)

$\gamma(b) = P(z=b|x) = \dfrac{\pi_b \mathcal{N}(x_i|\mu_b, \Sigma_b)}{P(x_i)}$

$\gamma(c) = P(z=c|x) = \dfrac{\pi_c \mathcal{N}(x_i|\mu_c, \Sigma_c)}{P(x_i)}$

from © ...

$P(x_i) = \pi_a \mathcal{N}(x_i|\mu_a, \Sigma_a) + \pi_b \mathcal{N}(x_i|\mu_b, \Sigma_b) + \pi_c \mathcal{N}(x_i|\mu_c, \Sigma_c)$

(g). Update equations for μ_a, μ_b, μ_c :

$$\mu_a = \sum_{n=1}^{N} \frac{\gamma(a)\, x_n}{\gamma(a)} ; \quad \mu_b = \sum_{n=1}^{N} \frac{\gamma(b)\, x_n}{\gamma(b)} ; \quad \mu_c = \sum_{n=1}^{N} \frac{\gamma(c)\, x_n}{\gamma(c)} .$$

(h). Update equations for $\Sigma_a, \Sigma_b, \Sigma_c$:

$$\Sigma_a = \sum_{n=1}^{N} \frac{\gamma(a)(x-\mu_a)(x-\mu_a)^T}{\sum_{n=1}^{N} \gamma(a)}$$

$$\Sigma_b = \sum_{n=1}^{N} \frac{\gamma(b)(x-\mu_b)(x-\mu_b)^T}{\sum_{n=1}^{N} \gamma(b)}$$

$$\Sigma_c = \sum_{n=1}^{N} \frac{\gamma(c)(x-\mu_c)(x-\mu_c)^T}{\sum_{n=1}^{N} \gamma(c)}$$

(i). Update equations for π_a, π_b, π_c :

$$\pi_a = \sum_{n=1}^{N} \frac{\gamma(a)}{N} ; \quad \pi_b = \sum_{n=1}^{N} \frac{\gamma(b)}{N} ; \quad \pi_c = \sum_{n=1}^{N} \frac{\gamma(c)}{N} .$$

(j). EM algorithm :

max Iter = 1000
for i = 1 to max Iter

① Pick initial values for $\mu_a, \mu_b, \mu_c, \Sigma_a, \Sigma_b, \Sigma_c, \pi_a, \pi_b, \pi_c$.

② Estep : Evaluate $\gamma_a, \gamma_b, \gamma_c$.

$$\gamma_a = \frac{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a)}{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a) + \pi_b \mathcal{N}(x \mid \mu_b, \Sigma_b) + \pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}$$

Similarly for γ_b and γ_c

③ Mstep : Evaluate $\mu_a, \mu_b, \mu_c, \Sigma_a, \Sigma_b, \Sigma_c, \pi_a, \pi_b, \pi_c$.

$$\mu(a) = \sum_{n=1}^{N} \frac{\gamma(a)\, x^n}{\gamma(a)} .$$ Similarly for μ_b and μ_c.

$$\Sigma(a) = \sum_{n=1}^{N} \frac{\gamma(a)(x-\mu_a)(x-\mu_a)^T}{\sum_{n=1}^{N}\gamma(a)}$$. Similarly for $\Sigma(b)$ & $\Sigma(c)$

$$\pi_a = \frac{\sum_{n=1}^{N}\gamma(a)}{N}$$. Similarly for π_b and π_c.

(4) check for convergence of either the log likelihood or the parameter values. If convergence criteria is not met then:

$$\theta^{old} \leftarrow \theta^{new}$$

and return to step (2)

(5) end for.

(j) Julia Code

(k). Julia Code

Q.2. $\log p(X|\theta) = \mathcal{L}(q,\theta) + KL(q\|p)$ where,

$$\mathcal{L}(q,\theta) = \sum_z q(z)\log\left\{\frac{p(X,z|\theta)}{q(z)}\right\}$$

$$KL(q\|p) = -\sum_z q(z)\log\left\{\frac{p(z|X,\theta)}{q(z)}\right\}$$

R.H.S =

$$\sum_z q(z)\log\left\{\frac{p(X,z|\theta)}{q(z)}\right\} - \sum_z q(z)\log\left\{\frac{p(z|X,\theta)}{q(z)}\right\}$$

$$= \sum_z q(z)\log\left\{\frac{p(X,z|\theta)}{\cancel{q(z)}}\cdot\frac{\cancel{q(z)}}{p(z|X,\theta)}\right\}$$

$$= \sum_z q(z) \log \left[\frac{P(X, Z \mid \theta)}{P(Z \mid X, \theta)} \right]$$

$$\because \log[P(X, Z \mid \theta)] = \log[(Z \mid X, \theta)] + \log P(X \mid \theta)]$$

$$= \sum_z q(z) \Big(\log \left[P(X, Z \mid \theta) \right] - \log \left[P(Z \mid X, \theta) \right] \Big)$$

$$= \sum_z q(z) \left(\cancel{\log(Z \mid X, \theta)} + \log P(X \mid \theta) - \cancel{\log(Z \mid X, \theta)} \right)$$

$$= \cancel{\sum_z q(z)}^{1} \log P(X \mid \theta)$$

$$= \log P(X \mid \theta)$$

$$= L.H.S.$$

Hence,

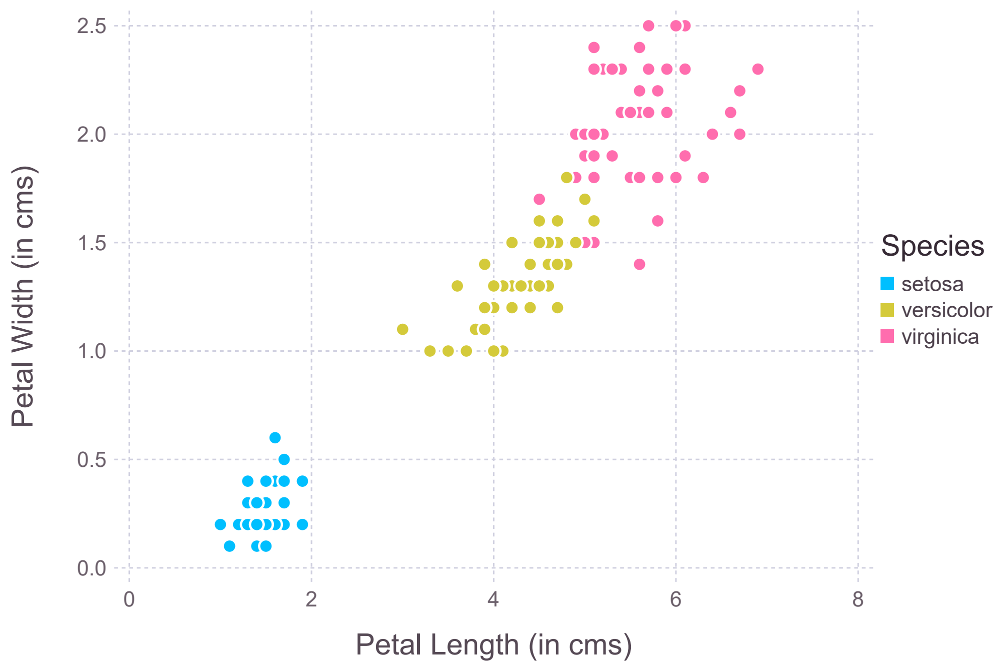
$$\log P(X \mid \theta) = \mathcal{L}(q, \theta) + KL(q \,\|\, p)$$

In [1]:
```
using RDatasets, Gadfly, Distributions;
```

In [2]:
```
data = dataset("datasets", "iris");
myplot = plot(data,x=:PetalLength,y=:PetalWidth, color=:Species,Geom.point,
    Guide.xlabel("Petal Length (in cms)"),
    Guide.ylabel("Petal Width (in cms)"), major_label_font_size=18pt,
    minor_label_font_size=14pt,
    key_title_font_size = 18pt,
    key_label_font_size = 14pt,
    major_label_color=colorant"black",
    minor_label_color=colorant"black",
    Coord.Cartesian(xmin=0, xmax=8))
```

```
WARNING: Method definition unix2zdt(Real) in module TimeZones at C:\Users\Amar\.julia\v0.6\TimeZones\src\conv
ersions.jl:122 overwritten in module RData at C:\Users\Amar\.julia\v0.6\RData\src\convert.jl:201.
WARNING: key_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: minor_label_color is not a recognized aesthetic. Ignoring.
WARNING: minor_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: major_label_color is not a recognized aesthetic. Ignoring.
WARNING: major_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: key_title_font_size is not a recognized aesthetic. Ignoring.
```

Out[2]:



In [12]:

```
## Coding EM
function E_step(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_c)
    numerator_a = zeros(size(x,1));
    numerator_b = zeros(size(x,1));
    numerator_c = zeros(size(x,1));
    denominator = zeros(size(x,1));
    post_x_a = zeros(size(x,1));
    post_x_b = zeros(size(x,1));
    post_x_c = zeros(size(x,1));
    for i=1:size(x,1)
        numerator_a[i] = pi_a.*pdf(MvNormal(mu_a,sigma_a),x[i,:]);
        numerator_b[i] = pi_b.*pdf(MvNormal(mu_b,sigma_b),x[i,:]);
        numerator_c[i] = pi_c.*pdf(MvNormal(mu_c,sigma_c),x[i,:]);
        denominator[i] = pi_a.*pdf(MvNormal(mu_a,sigma_a),x[i,:]) +
            pi_b.*pdf(MvNormal(mu_b,sigma_b),x[i,:]) +
            pi_c.*pdf(MvNormal(mu_c, sigma_c),x[i,:]);
        post_x_a[i] = numerator_a[i] ./denominator[i];
        post_x_b[i] = numerator_b[i] ./denominator[i];
        post_x_c[i] = numerator_c[i] ./denominator[i];

    end
    return post_x_a, post_x_b, post_x_c;
end

function M_step(x,post_x_a,post_x_b,post_x_c)

    mu_a = sum(post_x_a.*x,1)./sum(post_x_a); mu_a = Vector(mu_a[:]);
    mu_b = sum(post_x_b.*x,1)./sum(post_x_b); mu_b = Vector(mu_b[:]);
    mu_c = sum(post_x_c.*x,1)./sum(post_x_c); mu_c = Vector(mu_c[:]);
    sigma_a = round.((post_x_a.*(x.-mu_a'))'*(x.-mu_a')/sum(post_x_a),5);
    sigma_b = round.((post_x_b.*(x.-mu_b'))'*(x.-mu_b')/sum(post_x_b),5);
    sigma_c = round.((post_x_c.*(x.-mu_c'))'*(x.-mu_c')/sum(post_x_c),5);
    pi_a = sum(post_x_a)/size(x,1);
    pi_b = sum(post_x_b)/size(x,1);
    pi_c = sum(post_x_c)/size(x,1);

    return mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_c;
end

function EM(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_c)
    maxIter = 1000;
    for i=1:maxIter
```

```
        print(i,"\n");
        post_x_a,post_x_b,post_x_c =
            E_step(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_c);
#         print(post_x,"\n");

        nmu_a,nmu_b,nmu_c,nsigma_a,nsigma_b,nsigma_c,npi_a,npi_b,npi_c = M_step(x,post_x_a,post_x_b,post_x_c);
        print(nmu_a," ",nmu_b," ",nmu_c,"\n");
        print(nsigma_a," ",nsigma_b," ",nsigma_c,"\n");

        if(
                sum(abs.(mu_a-nmu_a))<0.001 &&
                sum(abs.(mu_b-nmu_b))<0.001 &&
                sum(abs.(mu_c-nmu_c))<0.001 &&
                sum(abs.(sigma_a-nsigma_a))<0.001 &&
                sum(abs.(sigma_b-nsigma_b))<0.001 &&
                sum(abs.(sigma_c-nsigma_c))<0.001
            )
            break;
        end;
        mu_a = nmu_a; mu_b = nmu_b; mu_c = nmu_c;
        sigma_a = nsigma_a; sigma_b = nsigma_b; sigma_c = nsigma_c;
        pi_a = npi_a; pi_b = npi_b; pi_c = npi_c;
    end
    return mu_a,mu_b,mu_c,sigma_b,sigma_b,sigma_c,pi_a,pi_b,pi_c;
end
```

Out[12]: EM (generic function with 1 method)

In [30]:
```
data = dataset("datasets","iris");
x = convert(Array,data[:,[:PetalLength,:PetalWidth]]);
mu_a=Vector([1, 2]);
mu_b=Vector([1, 4]);
mu_c=Vector([2, 5]);
sigma_a = [1.0 0.0; 0.0 1.0];
sigma_b = [1.0 0.0; 0.0 1.0];
sigma_c = [1.0 0.0; 0.0 1.0];
pi_a = 0.33; pi_b = 0.34; pi_c = 0.33;
mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_c = EM(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,pi_a,pi_b,pi_
```

```
1
[3.36544, 1.0148] [4.97396, 1.78084] [5.68281, 2.10156]
[2.798 1.13001; 1.13001 0.48648] [0.94748 0.40652; 0.40652 0.23879] [0.38634 0.0791; 0.0791 0.08671]
2
[3.26315, 0.960523] [5.04075, 1.79789] [5.59732, 2.09161]
[2.77323 1.09534; 1.09534 0.45299] [0.44509 0.21537; 0.21537 0.16805] [0.27263 0.03056; 0.03056 0.08125]
3
[3.15808, 0.909957] [5.00033, 1.78175] [5.5765, 2.08063]
[2.69546 1.04506; 1.04506 0.42149] [0.28919 0.16506; 0.16506 0.14451] [0.24042 0.02379; 0.02379 0.07782]
4
[3.06968, 0.868693] [4.93039, 1.75631] [5.56545, 2.06953]
[2.64465 1.01049; 1.01049 0.40041] [0.21907 0.14321; 0.14321 0.12933] [0.23518 0.02569; 0.02569 0.07739]
5
[2.98997, 0.832962] [4.86681, 1.72731] [5.57253, 2.06535]
[2.61608 0.98854; 0.98854 0.38665] [0.18386 0.12872; 0.12872 0.11806] [0.23503 0.02774; 0.02774 0.07909]
6
[2.91435, 0.800787] [4.81341, 1.6987] [5.60029, 2.06914]
[2.59411 0.97279; 0.97279 0.37722] [0.16468 0.11971; 0.11971 0.11095] [0.23105 0.02631; 0.02631 0.08095]
7
```

In [31]:

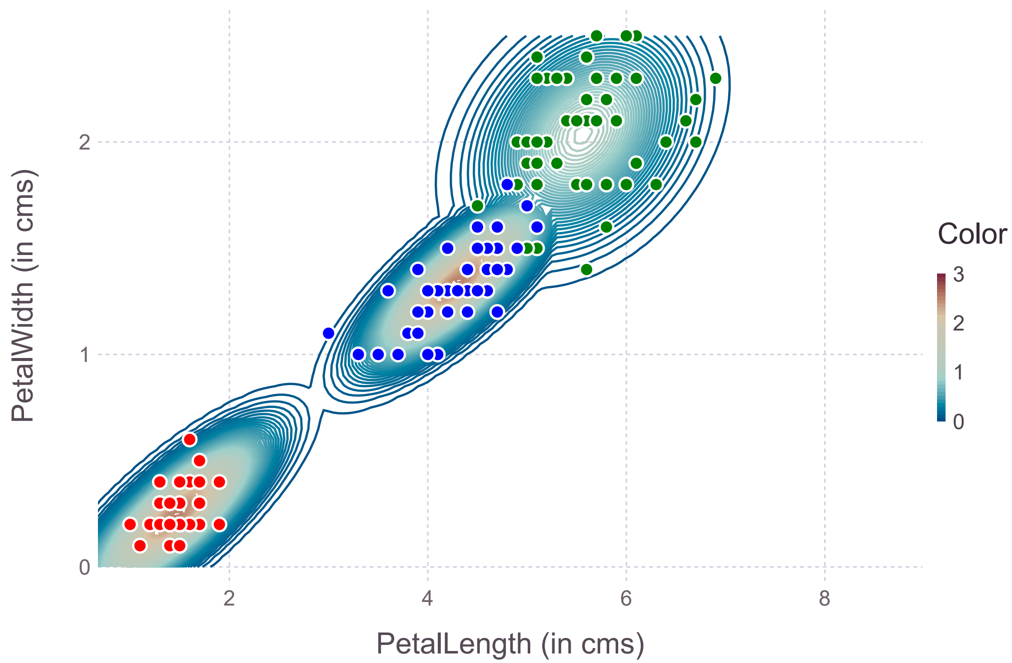
```
data_mat_a = data[find(data[:Species].=="setosa"),[:PetalLength,:PetalWidth]];
data_mat_b = data[find(data[:Species].=="versicolor"),[:PetalLength,:PetalWidth]];
data_mat_c = data[find(data[:Species].=="virginica"),[:PetalLength,:PetalWidth]];
nrows_a = size(data_mat_a,1);
nrows_b = size(data_mat_b,1);
nrows_c = size(data_mat_c,1);
#Estimate these using EM for MV Gaussian approach
mean_vec_a = mu_a;
mean_vec_b = mu_b;
mean_vec_c = mu_c;
cov_mat_a = sigma_a;
cov_mat_b = sigma_b;
cov_mat_c = sigma_c;

d_a = MvNormal(mean_vec_a,cov_mat_a);
d_b = MvNormal(mean_vec_b,cov_mat_b);
d_c = MvNormal(mean_vec_c,cov_mat_c);
```

In [32]:

```
a = collect(0:0.05:8);
b = collect(0:0.05:2.5);
pdf_mv = zeros(length(a),length(b));
for i=1:length(a)
    for j=1:length(b)
        pdf_mv[i,j] = maximum([pdf(d_a,[a[i],b[j]]),pdf(d_b,[a[i],b[j]]),pdf(d_c,[a[i],b[j]])]);
    end
end
myplot = plot(layer(x=data_mat_a[:,1],y=data_mat_a[:,2],
Geom.point,Theme(default_color=colorant"red")),layer(x=data_mat_b[:,1],y=data_mat_b[:,2],
Geom.point,Theme(default_color=colorant"blue")),layer(x=data_mat_c[:,1],y=data_mat_c[:,2],
Geom.point,Theme(default_color=colorant"green")),layer(z=pdf_mv,x=a,y=b, Geom.contour(levels=80)),
Coord.Cartesian(xmin=0, xmax=8,ymin=0,ymax=2.55),
major_label_font_size=18pt,
minor_label_font_size=14pt,
key_title_font_size = 18pt,
key_label_font_size = 14pt,
major_label_color=colorant"black",
minor_label_color=colorant"black",Guide.xlabel("PetalLength (in cms)"),Guide.ylabel("PetalWidth (in cms)"))
```

```
WARNING: key_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: minor_label_color is not a recognized aesthetic. Ignoring.
WARNING: minor_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: major_label_color is not a recognized aesthetic. Ignoring.
WARNING: major_label_font_size is not a recognized aesthetic. Ignoring.
WARNING: key_title_font_size is not a recognized aesthetic. Ignoring.
```

Out[32]:



In [ ]:
```
# Question 1 Part l)
# OBSERVATIONS:

# Changing the initializations gave us quite different results of the fit.
# We found the best fit with the above initialization in the code.
# This fit seems like a good fit as it separates the three species into
# three distinct distributions with little overlap.
```