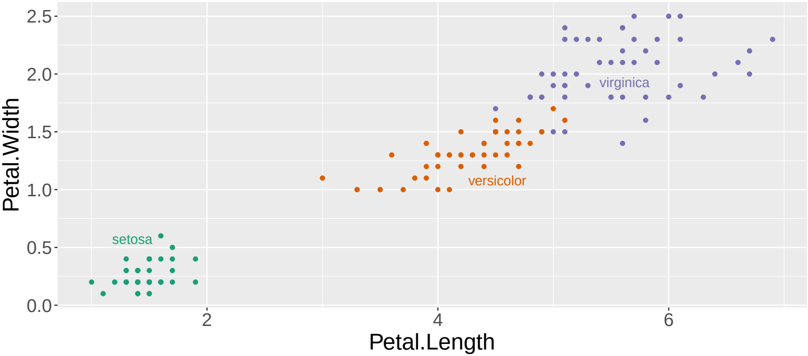
Gaussian mixture models

Toby Dylan Hocking

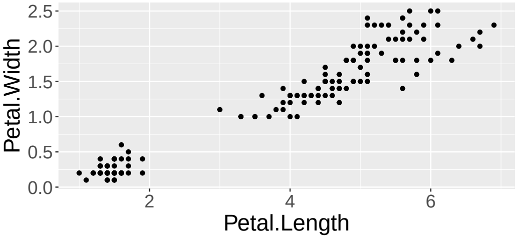
Visualize iris data with labels



Visualize iris data without labels

- Let $X = [x_1 \cdots x_n]^\intercal \in \mathbb{R}^{n \times p}$ be the data matrix (input for clustering), where $x_i \in \mathbb{R}^p$ is the input vector for observation i.
- Example iris $n = 150$ observations, $p = 2$ dimensions.

```
##      Petal.Width Petal.Length
## [1,]         0.2          1.4
## [2,]         0.2          1.4
## [3,]         0.2          1.3
## [4,]         0.2          1.5
```



Gaussian mixture model parameters and EM algorithm

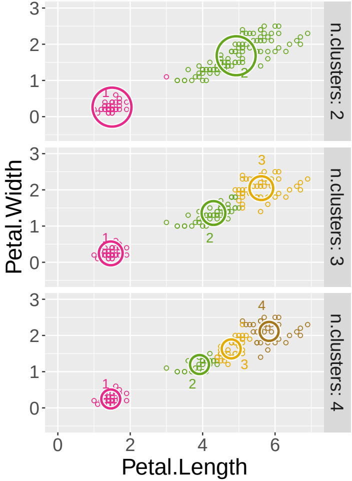
Need to fix number of clusters K, then for every $k \in \{1, \dots, K\}$ we have cluster-specific parameters $\theta_k = [\mu_k, S_k, \pi_k]$ which are updated during M step,

- mean vector $\mu_k \in \mathbb{R}^p$,
- covariance matrix $S_k \in \mathbb{R}^{p \times p}$, (must be symmetric, positive definite, next slides show optional additional constraints)
- prior weight $\pi_k \in [0, 1]$ (sum over all clusters k must equal one).

During E step we compute the probability matrix $T \in [0, 1]^{n \times K}$, where each row i sums to 1 and each entry T_{ik} is probability that data i is in cluster k.

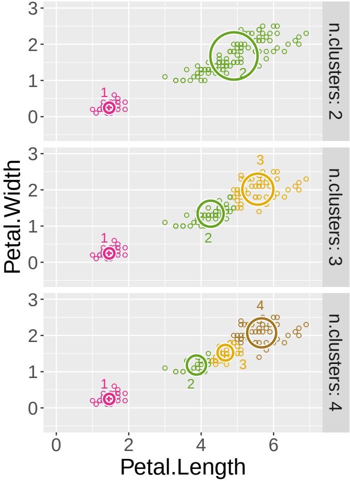
spherical, equal volume

```
##            c1     c1     c2     c2     c3     c3
## width  0.1077 0.0000 0.1077 0.0000 0.1077 0.0000
## length 0.0000 0.1077 0.0000 0.1077 0.0000 0.1077
```



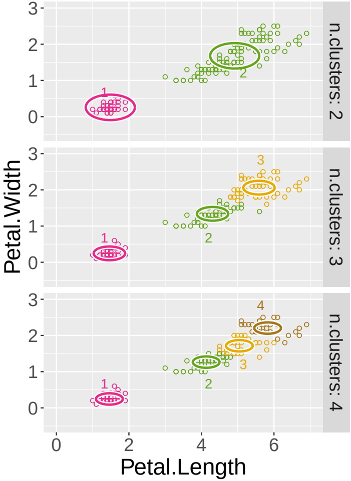
spherical, unequal volume

```
##               c1     c1     c2     c2     c3     c3
## width  0.0202 0.0000 0.1298 0.0000 0.1837 0.0000
## length 0.0000 0.0202 0.0000 0.1298 0.0000 0.1837
```



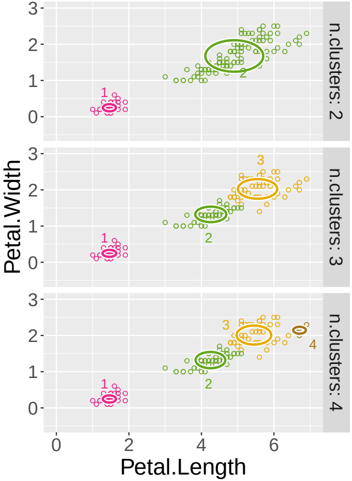
diagonal, equal volume and shape

```
##            c1     c1    c2     c2    c3     c3
## width  0.036 0.0000 0.036 0.0000 0.036 0.0000
## length 0.000 0.1878 0.000 0.1878 0.000 0.1878
```



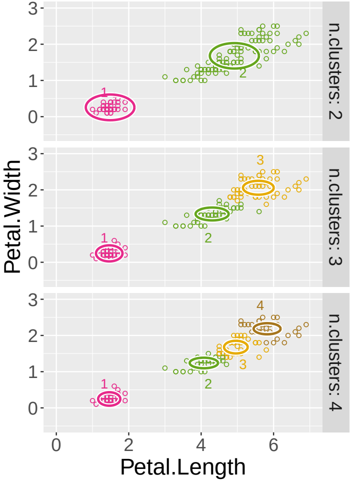
diagonal, varying volume, equal shape

```
##             c1     c1     c2     c2     c3     c3
## width  0.0091 0.0000 0.0457 0.0000 0.0732 0.0000
## length 0.0000 0.0367 0.0000 0.1837 0.0000 0.2944
```



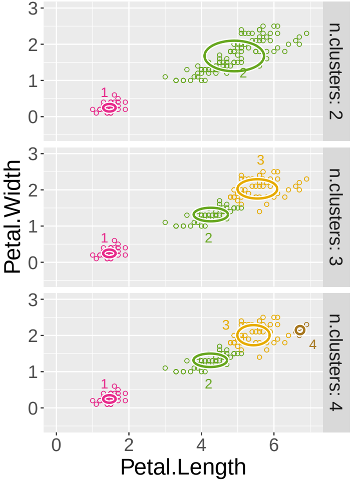
diagonal, equal volume, varying shape

```
##              c1     c1     c2     c2     c3     c3
## width  0.0494 0.0000 0.0317 0.0000 0.0368 0.0000
## length 0.0000 0.1341 0.0000 0.2089 0.0000 0.1802
```



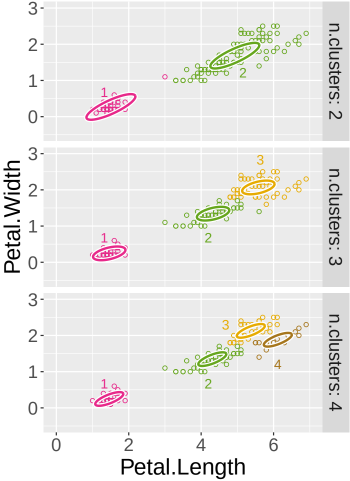
diagonal, varying volume and shape

```
##              c1     c1     c2     c2     c3     c3
## width  0.0109 0.0000 0.0352 0.0000 0.0709 0.0000
## length 0.0000 0.0296 0.0000 0.2243 0.0000 0.3008
```



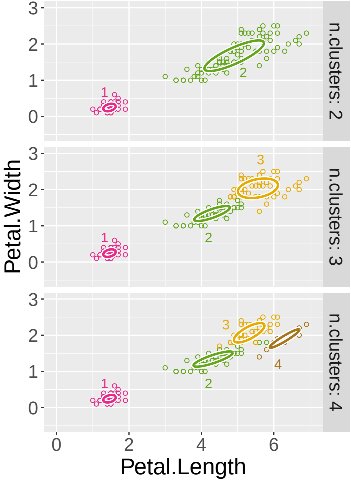
ellipsoidal, equal volume, shape, and orientation

```
##             c1     c1     c2     c2     c3     c3
## width  0.0358 0.0425 0.0358 0.0425 0.0358 0.0425
## length 0.0425 0.2005 0.0425 0.2005 0.0425 0.2005
```

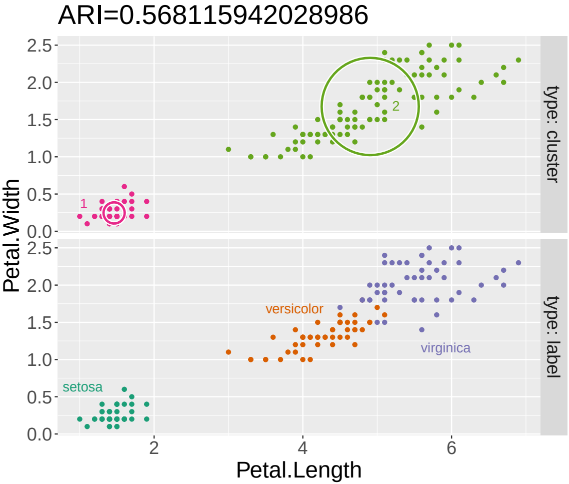


ellipsoidal, varying volume, shape, and orientation

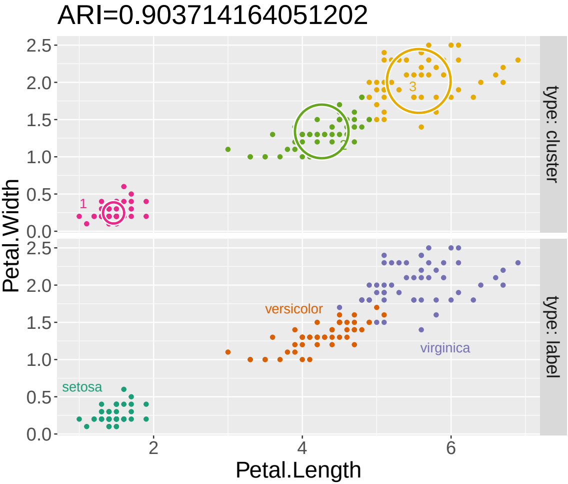
```
##               c1     c1     c2     c2     c3     c3
## width  0.0109 0.0059 0.0428 0.0813 0.0727 0.0482
## length 0.0059 0.0296 0.0813 0.2438 0.0482 0.3065
```



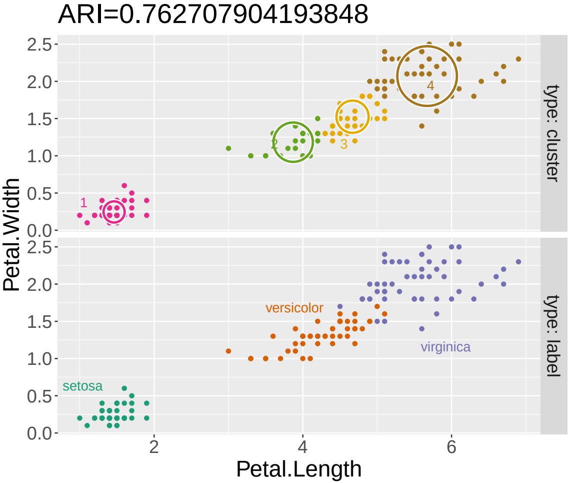
Compare two clusters to labels



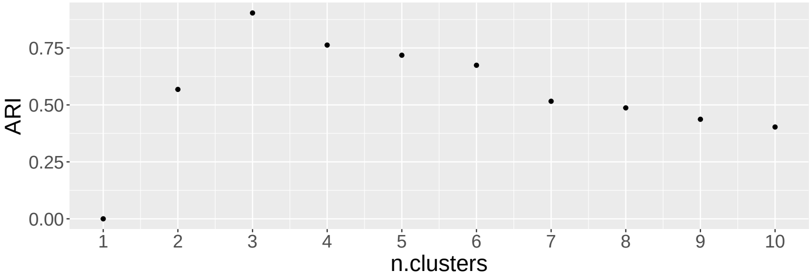
Compare three clusters to labels



Compare four clusters to labels

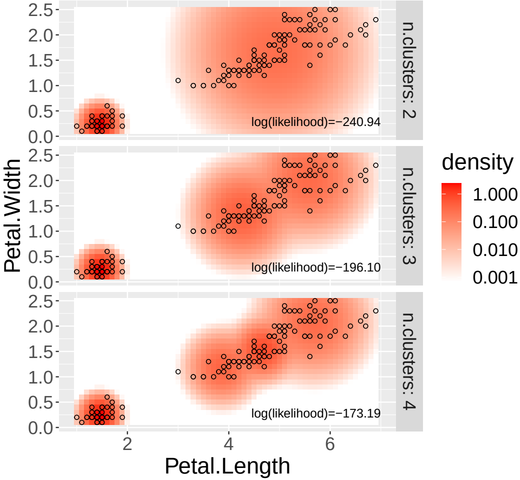


Compute ARI for several clusterings



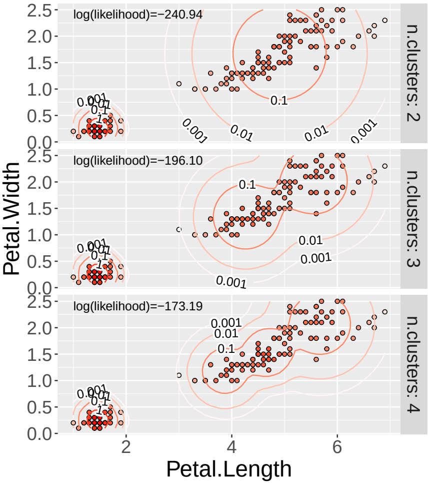
- Which K is best? Clear peak at 3 clusters, which makes sense since there are three species in these data.

Visualization of log likelihood

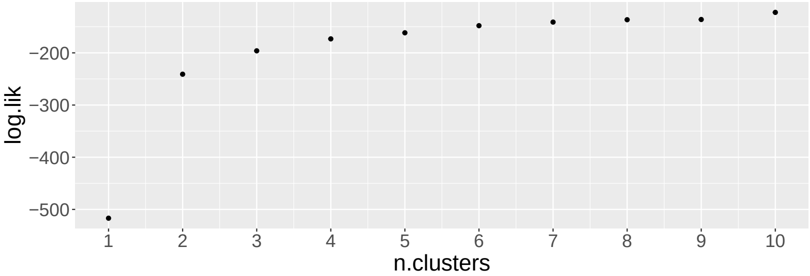


- Darker red means larger density value from learned model.
- The total redness in the data points represents the log likelihood, which is what the EM algorithm attempts to maximize.

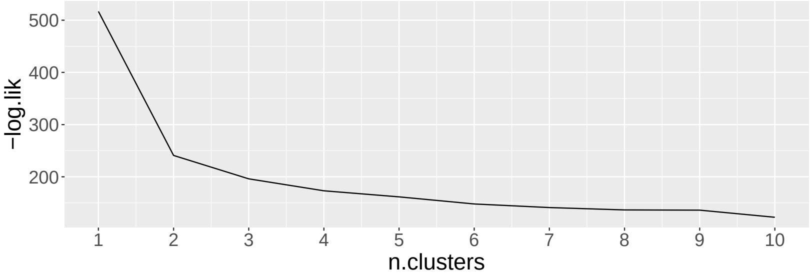
Visualize density using level curves



Compute log likelihood for several clusterings

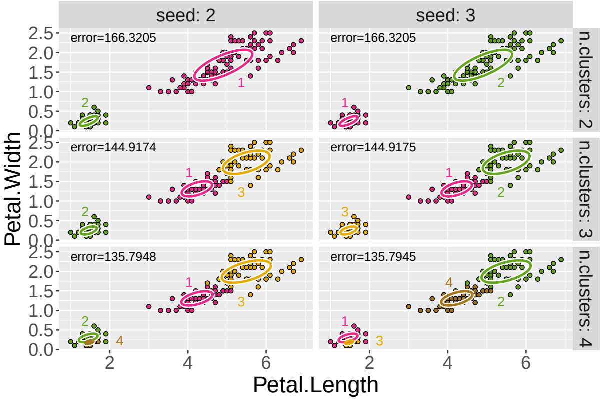


Model selection via error curve analysis (negative log likelihood)



- These error values can be computed using only the input data (labels/outputs are not required).
- In general, for any problem/data set, making this plot and then locating the “kink in the curve” is a good rule of thumb for selecting the number of clusters.

Visualize clusters using two random seeds



- Different seeds used for initial assignment based on K-means.
- EM solution quality depends on random seed (not much variation in these simple data though).

EM algo update rules

Let $f(x, \mu, S)$ be the (multivariate) normal density for a feature vector $x \in \mathbb{R}^p$, a mean vector $\mu \in \mathbb{R}^p$, and a covariance matrix $S \in \mathbb{R}^{p \times p}$.

In the E step we update the probability matrix,

$$T_{ik} \leftarrow \frac{\pi_k f(x_i, \mu_k, S_k)}{\sum_{k'=1}^{K} \pi_{k'} f(x_i, \mu_{k'}, S_{k'})}$$

.

In the M step we update the cluster parameters,

- $\pi_k \leftarrow \frac{1}{n} \sum_{i=1}^{n} T_{i,k}$,
- $\mu_k \leftarrow \frac{\sum_{i=1}^{n} T_{i,k} x_i}{\sum_{i=1}^{n} T_{i,k}}$,
- $S_k \leftarrow \frac{\sum_{i=1}^{n} T_{i,k}(x_i - \mu_k)(x_i - \mu_k)^\intercal}{\sum_{i=1}^{n} T_{i,k}}$ (no constraints).

Where do these update rules come from?

The goal of the algorithm is to find model parameters that maximize the log likelihood, $\log L(x, \theta, T)$.

$$\max_{\theta} \log L(x, \theta, T)$$

$$\max_{T} \log L(x, \theta, T)$$

- Use gradient condition to derive T, θ which maximize the likelihood given the data and other parameters.
- Covariance constraints affect the θ update rule. For example diagonal covariance update,
- $S_k \leftarrow \frac{\sum_{i=1}^{n} \text{Diag}[T_{i,k}(x_i - \mu_k)(x_i - \mu_k)^\intercal]}{\sum_{i=1}^{n} T_{i,k}}$ (diagonal constraint).
- j-th entry/feature of S_k is $\sum_{i=1}^{n} T_{i,k} D_{i,j}^2$ where $D \in \mathbb{R}^{n \times p}$ is the difference matrix $X - \mu_k$. (avoids matrix multiplication, linear rather than quadratic time in feature dimension p)

Numerical issues (underflow)

To avoid numerical issues in EM algorithm we need to

- Use the log density with max probability trick, to avoid non-finite probability values in E step.

$$T_{ik} \leftarrow \frac{\pi_k f(x_i, \mu_k, S_k)}{\sum_{k'=1}^{K} \pi_{k'} f(x_i, \mu_{k'}, S_{k'})}$$

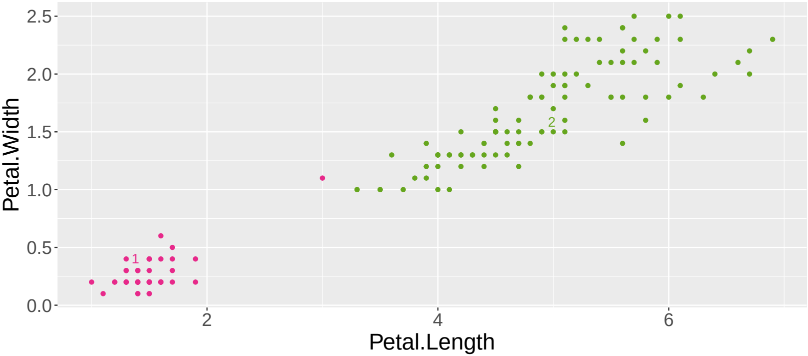
$$\log T_{ik} \leftarrow \log \pi_k + \log f(x_i, \mu_{k'}, S_{k'}) - Z$$

$$Z = M + \log[e^M \sum_{k'=1}^{K} \pi_{k'} e^{\log f(x_i, \mu_{k'}, S_{k'}) - M}]$$

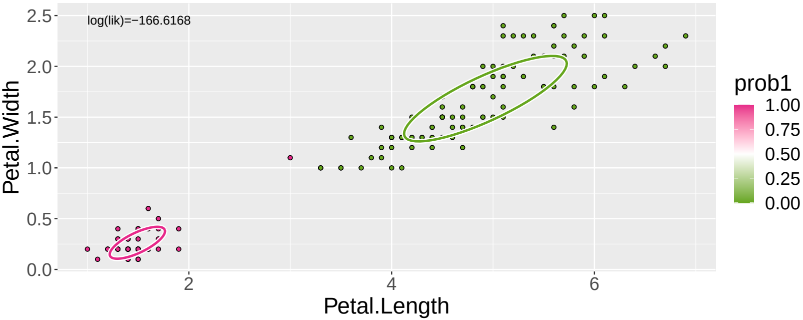
$$M = \max_{k' \in \{1, \dots, K\}} \log f(x_i, \mu_{k'}, S_{k'})$$

- Add a small number, $\lambda = 10^{-6}$ to the diagonal of the covariance matrix to avoid zero variance in M step, $S_k \leftarrow \frac{\sum_{i=1}^{n} T_{i,k}(x_i - \mu_k)(x_i - \mu_k)^\intercal}{\sum_{i=1}^{n} T_{i,k}} + \lambda I_p$.

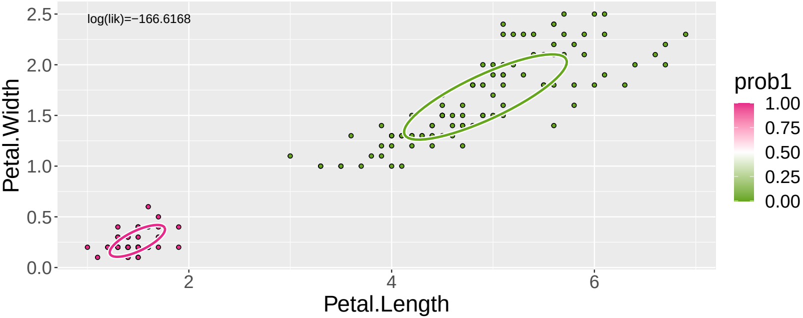
EM algo starting with K-means assignments



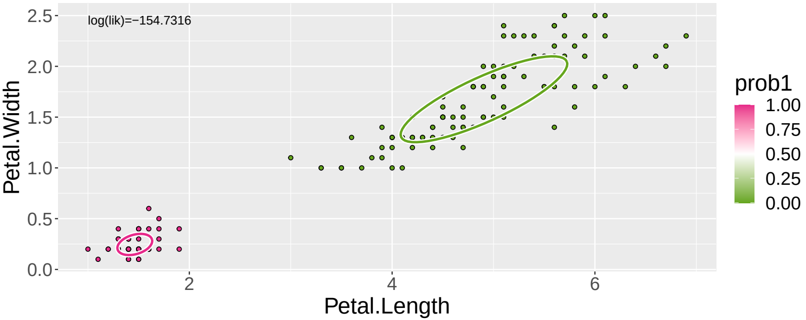
Compute weights, means, covariance matrices



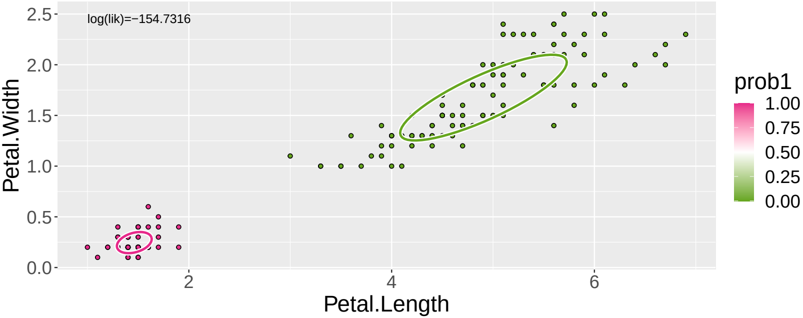
Cluster probabilities updated



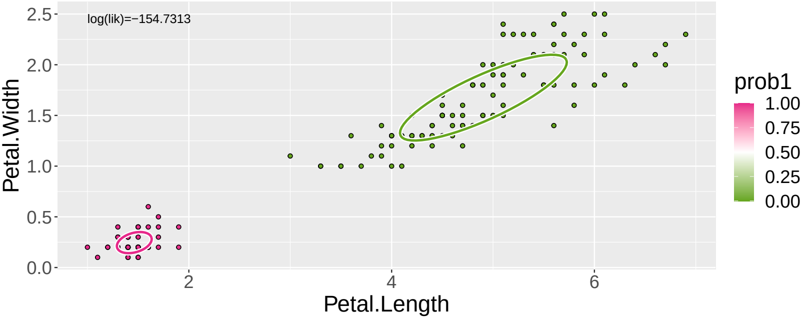
Compute new cluster parameters



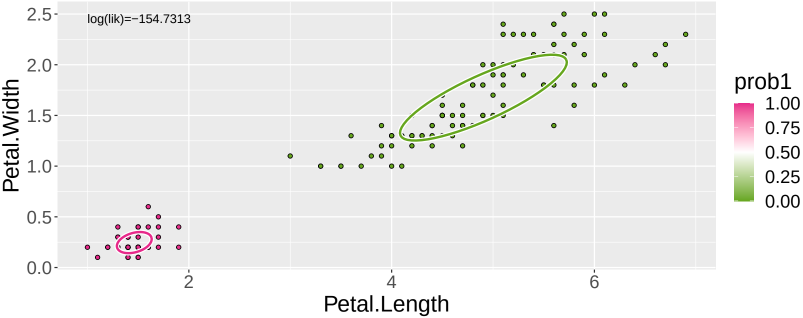
Compute new cluster/data probabilities



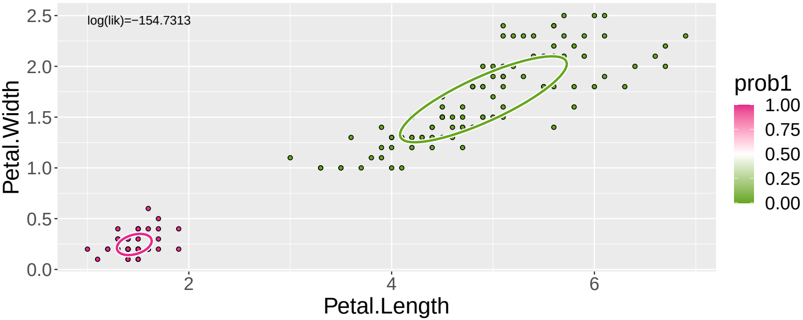
Compute cluster parameters iteration 3



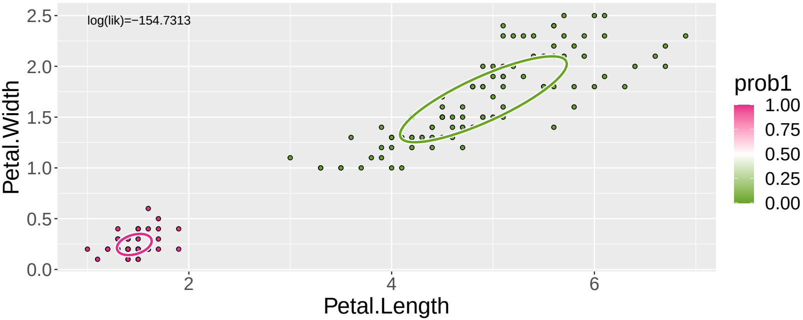
Compute probabilities iteration 3



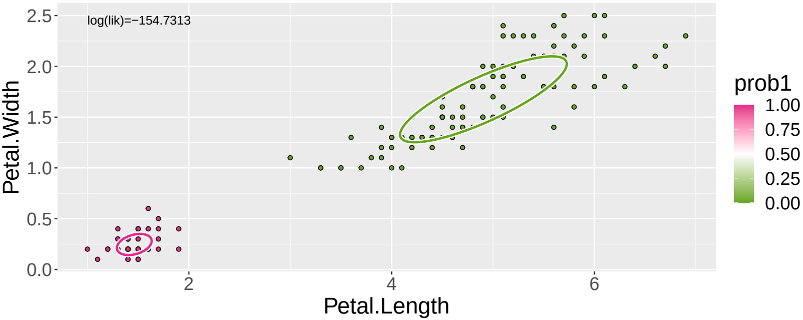
Compute cluster parameters iteration 4



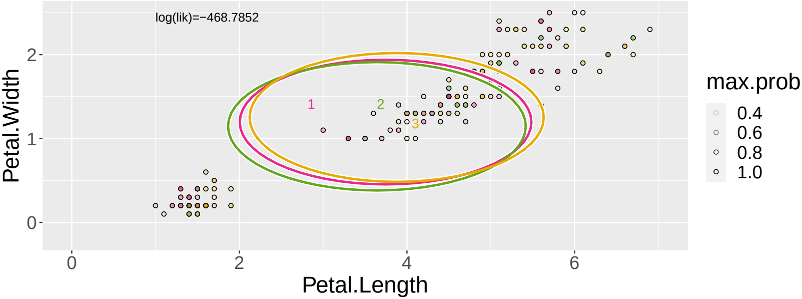
Compute probabilities iteration 4



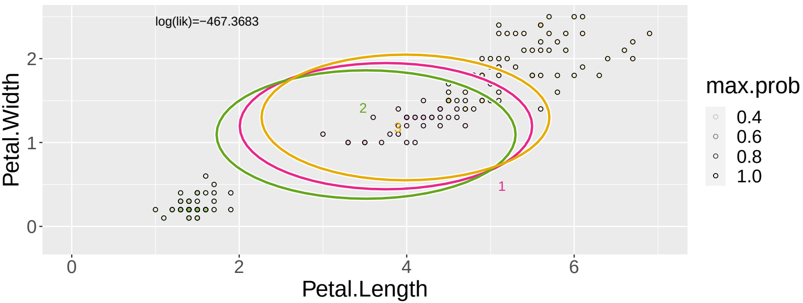
Compute cluster parameters iteration 5 (no change = stop)



Three clusters, diagonal constraint, random initialization

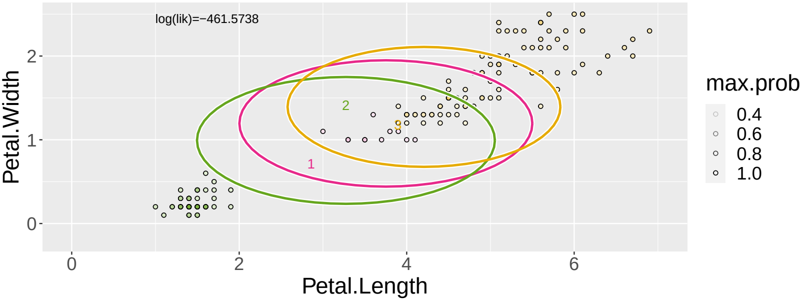


iteration 2

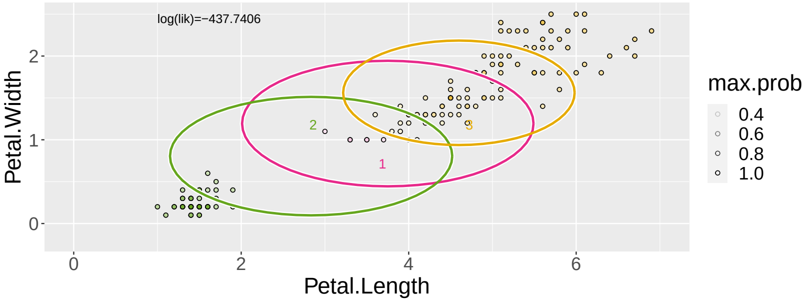


log(lik)=−467.3683
2
3
1
Petal.Width
0
1
2
Petal.Length
0
2
4
6
max.prob
0.4
0.6
0.8
1.0

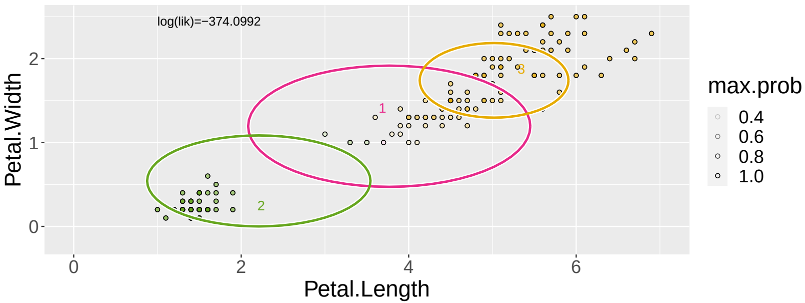
iteration 3


log(lik)=−461.5738
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

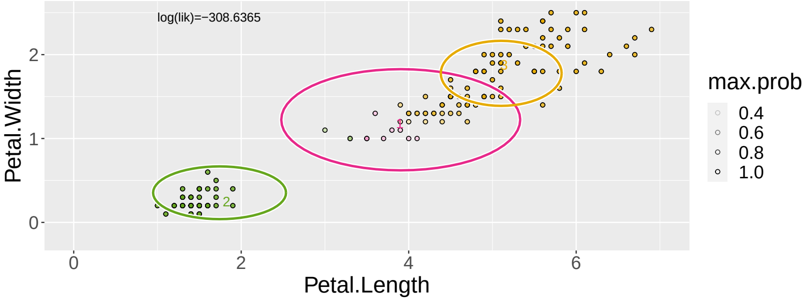
iteration 4


log(lik)=−437.7406
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

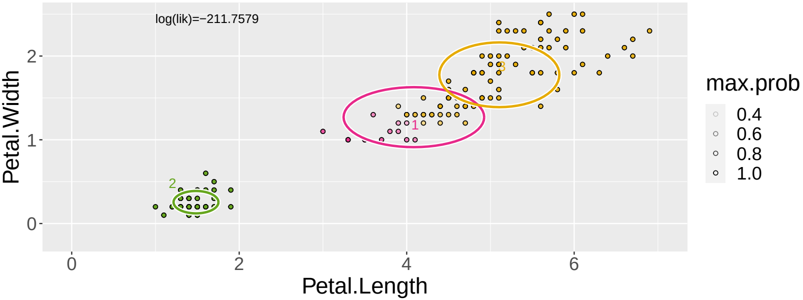
iteration 5


log(lik)=−374.0992
1
2
3
Petal.Width
0
1
2
Petal.Length
0
2
4
6
max.prob
0.4
0.6
0.8
1.0

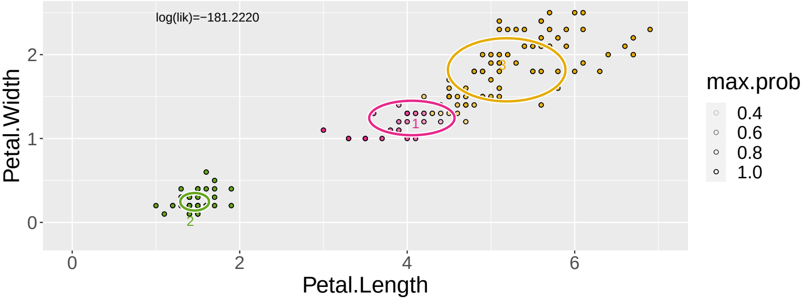
iteration 6


log(lik)=−308.6365
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

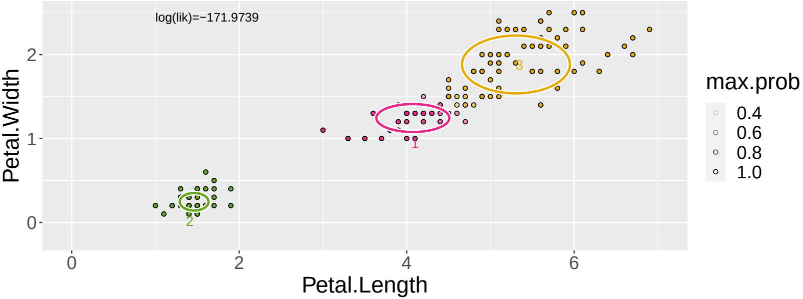
iteration 7


log(lik)=−211.7579
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

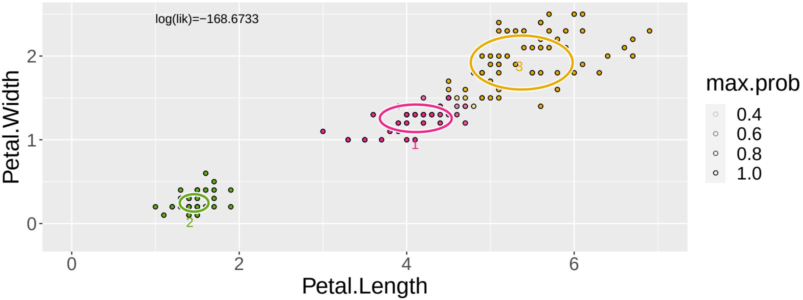
iteration 8


log(lik)=−181.2220
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

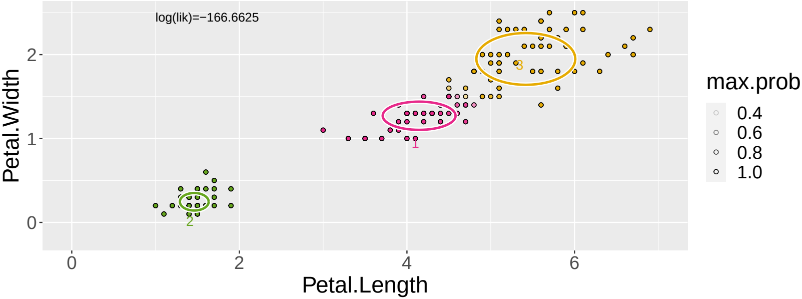
iteration 9


log(lik)=−171.9739
Petal.Width
Petal.Length
0
1
2
0
2
4
6
1
2
3
max.prob
0.4
0.6
0.8
1.0

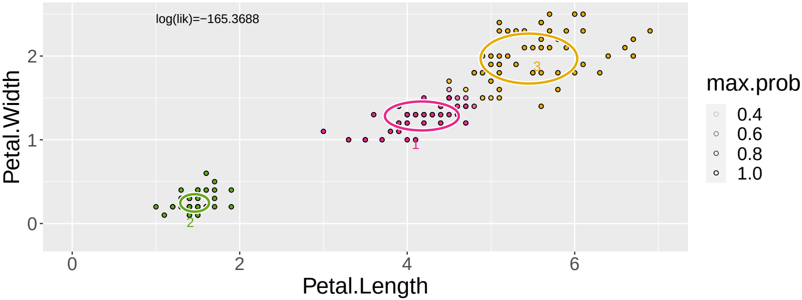
iteration 10



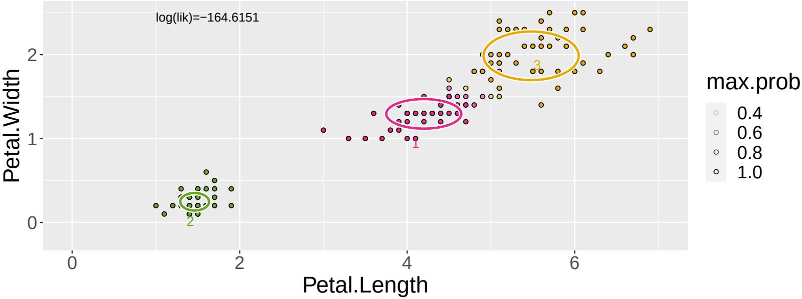
iteration 11



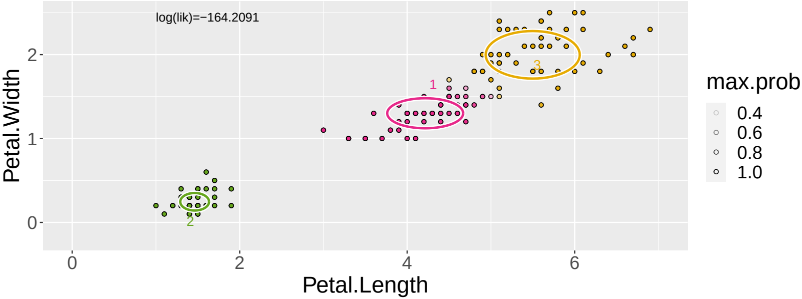
iteration 12



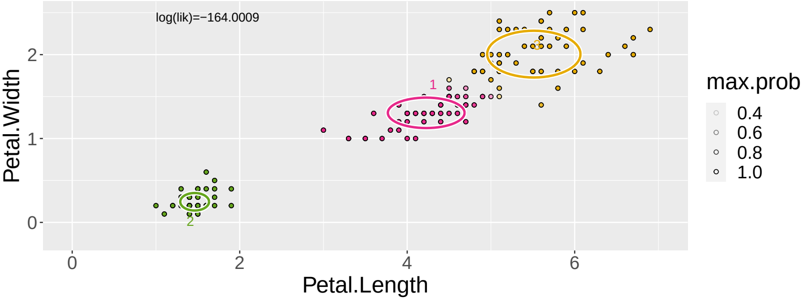
iteration 13



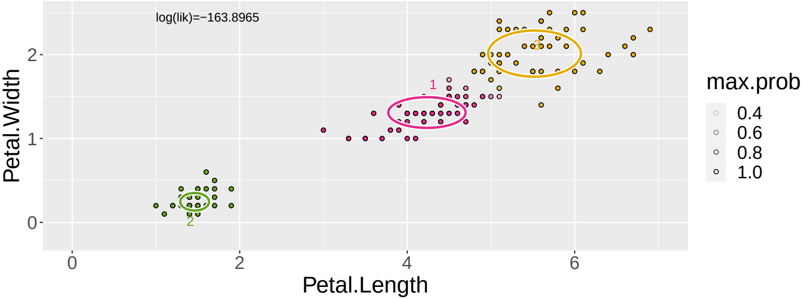
iteration 14



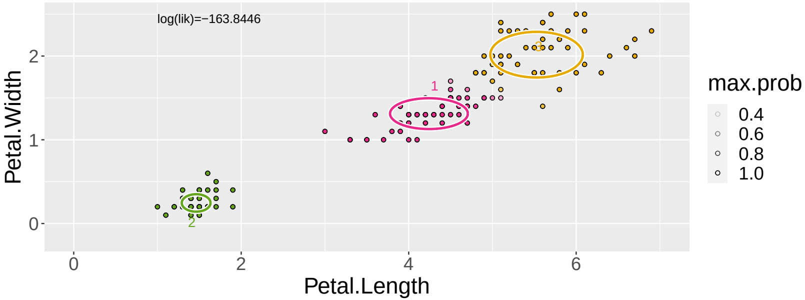
iteration 15



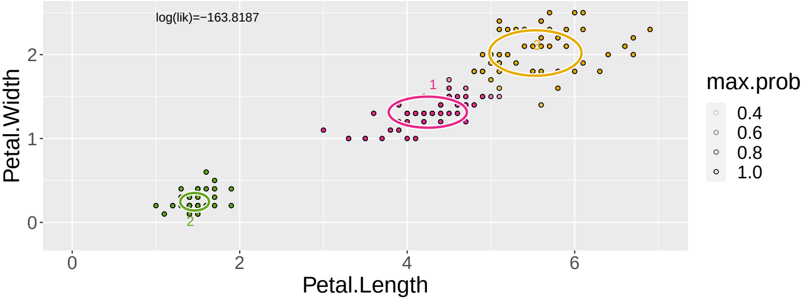
iteration 16



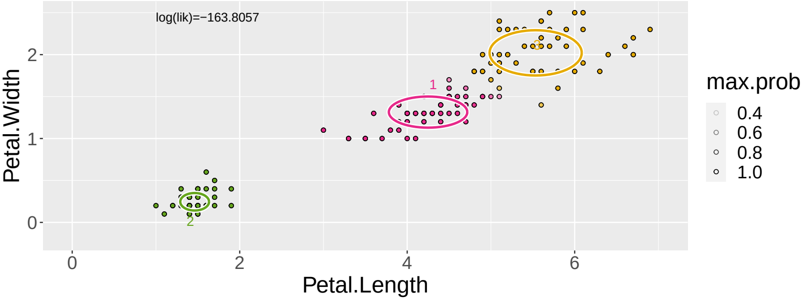
iteration 17



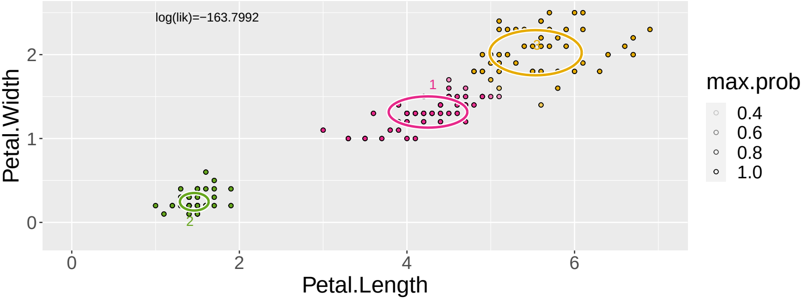
iteration 18



iteration 19



iteration 20



Possible Exam Questions

- How many real number parameters in an unconstrained gaussian mixture model for data with $p = 5$ features?
- What hyper-parameter is common to K-means and Gaussian mixtures? (A hyper-parameter is a model choice that must be fixed before running the learning/EM algorithm)
- What hyper-parameter is unique to Gaussian mixtures?
- What cluster parameter is common to K-means and Gaussian mixtures?
- What cluster parameters are present in Gaussian mixtures but not in K-means?

Possible exam questions 2

- We say K-means uses hard assignment and Gaussian mixtures uses soft assignment – what values are used in the probability/assignment matrix in each case?
- The K-means and Gaussian mixtures have similar learning algorithms. What are the main steps in common and what is the difference?
- In K-means we compute the squared error, and in Gaussian mixtures we compute the negative log likelihood – these values INCREASE or DECREASE as the number of clusters increase? These values INCREASE or DECREASE as the number of iterations of the learning algorithm increases?