Stats 102B - Week 6, Lecture 1

Miles Chen, PhD

Department of Statistics

Week 6 Monday

Section 1

Classification - Bayes Classifier

Classification

Chapter 5 in A First Course in Machine Learning

We have N training objects, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, each with dimension D. Each object is associated with a class label t_n.

There are C classes, and $t_n = 1, 2, \ldots, C$.

The goal is to predict t_{new} for an unseen object $\mathbf{x}_{new}$.

While the class labels are integers, it is important to treat them as categories. Class label 2 is not twice as big as class label 1. Also, it is not meaningful to predict a class label 2.5.

The Bayes Classifier and other classification methods

The Bayes classifier uses Bayes' rule to find the probability that an observation belongs to a certain class.

The Bayes classifier is a probabilistic classifier: it returns probabilities (e.g. the model says observation 1 belongs to class A with probability 0.19, class B with probability 0.80, Class C with probability 0.01). Logistic regression is another example of a probabilistic classifier.

On the other hand, non-probabilistic classifiers use "hard" rules and designate a class without an associated probability (e.g. our model says observation 1 is class B). Non-probabilistic classifiers include K-nearest-neighbors and Support Vector Machines.

While it is an application of Bayes' rule, it is not considered to be "Bayesian statistics" (which involves treating parameters as random variables - covered in 102C).

Review: Bayes Rule

Bayes' theorem states:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

The numerator is equal to $\Pr(B \cap A)$

And by the law of total probability, the denominator $(\Pr(B))$ can be decomposed to: $\sum_n \Pr(B|A_n)\Pr(A_n)$

Note that denominator is equal to the numerator summed across all possible values of A.

The Bayes Classifier: Our goal

We want:

$$\Pr(T_{new} = c | \mathbf{x}_{new}, \mathbf{X}, \mathbf{t})$$

We want to know the probabiltity that the class label T_{new} of some new observation is equal to some particular class label c, given:

- the values in the input vector of the new observation: $\mathbf{x}_{new}$
- and all the values in the trainings data $\mathbf{X}$
- and all the labels in the trainings data $\mathbf{t}$

The Bayes Classifier returns a probability

The result is a probability, so the following rules apply:

- The probability must be between 0 and 1

$$0 \leq \Pr(T_{new} = c | \mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) \leq 1$$

- the probabilities across all possible outcome classes must sum to 1

$$\sum_{c=1}^{C} \Pr(T_{new} = c | \mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) = 1$$

(e.g. If there are three classes (A, B, C) the model might say observation 1 belongs to class A with probability 0.19, class B with probability 0.80, Class C with probability 0.01, the sum of which add to 1)

The Bayes Classifier uses Bayes Rule:

We apply Bayes' rule: $\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$

$$\Pr(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) = \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t})\Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})}$$

- Posterior: $\Pr(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t})$ is the probability that we want to find
- Likelihood: $\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t})$ is the probability of observing the observed values in $\mathbf{x}_{new}$ if we assumed the new observation belonged to a specific class c. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)
- Prior: $\Pr(T_{new} = c|\mathbf{X}, \mathbf{t})$ is the probability that some new observation belongs to class c before we know anything else about it. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)
- Marginal: $\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})$ is the probability of observing the values in $\mathbf{x}_{new}$ regardless of class label. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)

Expanding the denominator

We apply the law of total probability and can take advantage of the fact that the probabilities across all possible classes must sum to 1. This means that the denominator must be equal to the sum of the numerator across all possible classes (i.e. all possible values of c).

$$
\begin{aligned}
\Pr(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) &= \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})} \\
&= \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\text{sum of numerator across all possible classes}} \\
&= \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\sum_{c'=1}^{C} \Pr(\mathbf{x}_{new}|T_{new} = c', \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c'|\mathbf{X}, \mathbf{t})}
\end{aligned}
$$

Section 2

A silly and simple example of the Bayes Classifier

A silly and simple example

Imagine you decide to purchase pouches that are filled with red and blue marbles. You go online and order 25 pouches of marbles from some arbitrary company.

When they arrive, we see that the pouches are filled at three different factories, each with a different filling proportion of red and blue marbles.

In our shipment, 24 pouches are labeled with their factory: factory A ($n = 12$), factory B ($n = 6$), and factory C ($n = 6$). The last one is not labeled, and we'll try to guess which factory it came from.

We assume the bags are filled via some Bernouli/Binomial process.

training data
test

Summary of our training data:

After tallying up all of our data across all pouces, we have the following summary stats for each factory:

- For the 12 pouches that came from Factory A, 60% of the marbles we observed are red, and 40% are blue.
- For the 6 pouches that came from Factory B, 50% of the marbles we observed are red, and 50% are blue.
- For the 6 pouches that came from Factory C, 40% of the marbles we observed are red, and 60% are blue.

Keep in mind, we don't work at the factories and don't know the exact filling proportions. The different pouches that come from the same factory exhibit variation, but do not cause concern about assuming the marble selection is a result of a binomial process. For example, one pouch from factory A has 20 marbles total and 14 are red ($p = 0.70$), while another pouch has 20 marbles and 11 are red ($p = 0.55$). These are all results with high likelihoods and we need not worry about the Binomial assumption.

Our summary stats above are simply maximum likelihood estimates of the filling proportions based on all the training data we observed.

Our test case: Likelihood

We open the bag and there are 20 marbles and 10 of them are red ($p = 0.5$). We wish to know which factory produced this bag.

The Likelihood: $\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t})$ is the probability of observing the observed data in $\mathbf{x}_{new}$ if we assumed the new observation belonged to a specific class c. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)

The Binomial probability: $\binom{n}{x}p^x(1-p)^{n-x}$

Based on our training data:

- The likelihood it came from Factory A is: $\binom{20}{10}0.6^{10}0.4^{10} = 0.1171416$
- The likelihood it came from Factory B is: $\binom{20}{10}0.5^{10}0.5^{10} = 0.1761971$
- The likelihood it came from Factory C is: $\binom{20}{10}0.4^{10}0.6^{10} = 0.1171416$

Our test case: Prior probabilities

Prior to opening the 25th bag, what is probability that it came from a particular factory?

The Prior probability: $\Pr(T_{new} = c|\mathbf{X}, \mathbf{t})$ is the probability that some new observation belongs to class c before we know anything else about it. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)

Based on our training data, it seems reasonable to assume it came from Factory A with probability 0.5 because 12 of the 24 bags in our training data were from Factory A. We apply the same reasoning to all factories

To summarize:

- It came from Factory A with probability 0.5 (12/24). $\Pr(T_{new} = A|\mathbf{X}, \mathbf{t}) = 0.5$
- It came from Factory B with probability 0.25 (6/24). $\Pr(T_{new} = B|\mathbf{X}, \mathbf{t}) = 0.25$
- It came from Factory C with probabiltiy 0.25 (6/24). $\Pr(T_{new} = C|\mathbf{X}, \mathbf{t}) = 0.25$

The numerators for all classes:

The numerator is the product of the likelihood and the prior probabilities for each class.

$$\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})$$

- For Class A: the likelihood of observing 10 red if bag came from A × prob a random bag comes from A
  - 0.1171416 * 0.5 = 0.0585708
- For Class B: the likelihood of observing 10 red if bag came from B × prob a random bag comes from B
  - 0.1761971 * 0.25 = 0.04404927
- For Class C: the likelihood of observing 10 red if bag came from C × prob a random bag comes from C
  - 0.1171416 * 0.25 = 0.0292854

The denominator

In the denominator, we have the marginal probability.

- The Marginal probability: $\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})$ is the probability of observing the values in $\mathbf{x}_{new}$ regardless of class label. (We calculate our probabilties based on the training data $\mathbf{X}, \mathbf{t}$)

We apply the law of total probability and we find the Marginal probability by summing up the numerator across all possible classes.

$$\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t}) = \sum_{c'=1}^{C} \Pr(\mathbf{x}_{new}|T_{new} = c', \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c'|\mathbf{X}, \mathbf{t})$$

For our example, the denominator is:

$0.0585708 + 0.04404927 + 0.0292854 = 0.1319055$

The posterior

Now we can calculate the probability of each class:

$$\Pr(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) = \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})}$$

$$\begin{aligned}\Pr(\text{class} = A|\text{10 red out of 20}, \text{training data}) &= \frac{\Pr(\text{10 red}|\text{class A has p=0.6}) \Pr(\text{Factory A})}{\Pr(\text{getting 10 red regardless of factory})} \\ &= \frac{\binom{20}{10} 0.6^{10} 0.4^{10} \times 12/24}{\text{sum of numerators across all classes}} \\ &\approx \frac{0.0585708}{0.1319055} \approx 0.444\end{aligned}$$

The posterior

$$\begin{aligned}
\Pr(\text{class} = B | \text{10 red out of 20, training data}) &= \frac{\Pr(\text{10 red}|\text{class B has p=0.5}) \Pr(\text{Factory B})}{\Pr(\text{getting 10 red regardless of factory})} \\
&= \frac{\binom{20}{10} 0.5^{10} 0.5^{10} \times 6/24}{\text{sum of numerators across all classes}} \\
&\approx \frac{0.04404927}{0.1319055} \approx 0.334
\end{aligned}$$

$$\begin{aligned}
\Pr(\text{class} = C | \text{10 red out of 20, training data}) &= \frac{\Pr(\text{10 red}|\text{class C has p=0.4}) \Pr(\text{Factory C})}{\Pr(\text{getting 10 red regardless of factory})} \\
&= \frac{\binom{20}{10} 0.4^{10} 0.6^{10} \times 6/24}{\text{sum of numerators across all classes}} \\
&\approx \frac{0.0585708}{0.1319055} \approx 0.222
\end{aligned}$$

Bayes Classifier

Keep in mind our test case is 0.5 red.

If we were to base our classification on this only this information, we would pick Factory B, because in our training data, the proportion of red marbles from Factory B is 50% red, while Factory A has an estimated proportion of 60% and Factory C 40%.

However, with the Bayes classifier, we take into account the probabilities prior to looking at the data. Specifically in our case, we saw that half the bags were from Factory A, while only one-fourth of the bags came from Factory B.

The likelihood calculations show that getting 50% red in a bag of 20 from a factory where the true proportion is 0.6 is not an unlikely occurrence.

So when we do our final calculations which combine the likelihood and prior probabilities, Factory A ends up the most probable class.

Some strong assumptions

Keep in mind the assumptions we make:

1. After tallying all the data in our training cases, we assumed our MLE estimates are the proportions of Red marbles at the Factory. (This assumption can be relaxed if we take a fully Bayesian approach and model our factory proportion with a probability distribution - but that is beyond the scope of 102B.)
2. We said that because 12 of 24 bags in the training data were from Factory A, the test case has a 50% probability of coming from Factory A. This assumes that our sample is representative of the class proportions in the population.

Regarding the second assumption, someone else may make entirely different assumptions. For example, someone may have knowledge of the manufacturing process and know that all three factories produce equal quantities of bags. Therefore, they say the probability that a random bag came from Factory A is 1/3 (and that we should ignore the class proportions observed in our training data).

The situation and your domain knowledge will affect how you model your prior probabilities.

Same example, different test case

Let's continue the same example, but just use a different test outcome. This time in our test data, we have 20 marbles and only 6 are red. The likelihoods are based on the binomial distribution and the observed data. The prior probabilities are based on the fact that out of 24 bags, 12 came from A, 6 from B, and 6 from C.

Let's calculate our numerators for all three classes:

$$\Pr(\text{6 red}|\text{class A has p=0.6})\Pr(\text{Factory A}) = \binom{20}{6}0.6^6 0.4^{14} \times 0.5 = 0.002427175$$

$$\Pr(\text{6 red}|\text{class B has p=0.5})\Pr(\text{Factory B}) = \binom{20}{6}0.5^6 0.5^{14} \times 0.25 = 0.009241104$$

$$\Pr(\text{6 red}|\text{class C has p=0.4})\Pr(\text{Factory C}) = \binom{20}{6}0.4^6 0.6^{14} \times 0.25 = 0.03110292$$

$$\text{denominator} = \text{sum of numerators} = 0.002427175 + 0.009241104 + 0.03110292 = 0.0427712$$

Same example, different test case

$$\Pr(\text{class} = A|\text{6 red out of 20, training data}) = \frac{\Pr(\text{6 red}|\text{class A has p=0.6})\Pr(\text{Factory A})}{\Pr(\text{getting 6 red regardless of factory})}$$

$$\approx \frac{0.002427175}{0.0427712} \approx 0.0567$$

$$\Pr(\text{class} = B|\text{6 red out of 20, training data}) = \frac{\Pr(\text{6 red}|\text{class B has p=0.5})\Pr(\text{Factory B})}{\Pr(\text{getting 6 red regardless of factory})}$$

$$\approx \frac{0.009241104}{0.0427712} \approx 0.2161$$

$$\Pr(\text{class} = C|\text{6 red out of 20, training data}) = \frac{\Pr(\text{6 red}|\text{class C has p=0.4})\Pr(\text{Factory C})}{\Pr(\text{getting 6 red regardless of factory})}$$

$$\approx \frac{0.03110292}{0.0427712} \approx 0.7272$$

Same example, different test case

With this test case, we can see that the likelihood of getting only 6 red out of 20 marbles ($\hat{p} = 0.3$) is very low if the bag came from Factor A (where the proportion is 0.6 red). So even though we assume 50% of bags come from Factory A, once we see that only 6 marbles are Red, it makes Factory A very improbable.

On the other hand, getting only 6 red is not unlikely if it came from Factory C (where the proportion is 0.4 red). So despite our prior probability that only 1/4 of bags come from Factory C, the observed data makes Factory C the most probable class assignment.

Section 3

Mixtures of Multivariate Gaussians

The Multivariate Gaussian (Normal) Distribution

If $\mathbf{x} = [x_1, x_2, \ldots, x_D]^T$ comes from a multivariate Gaussian distribution in D dimensions, the PDF is:

$$\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

where $\boldsymbol{\mu}$ is a vector of means of the population (the same size as $\mathbf{x}$), the *d*th element tells us the mean of x_d. The variance $\boldsymbol{\Sigma}$ is a $D \times D$ variance-covariance matrix of the x-variables.

If the off-diagonal elements in $\boldsymbol{\Sigma}$ are 0, then the x-variables are independent and the multivariate PDF can be factored into a product of univariate Gaussian PDFs.

Simulate data

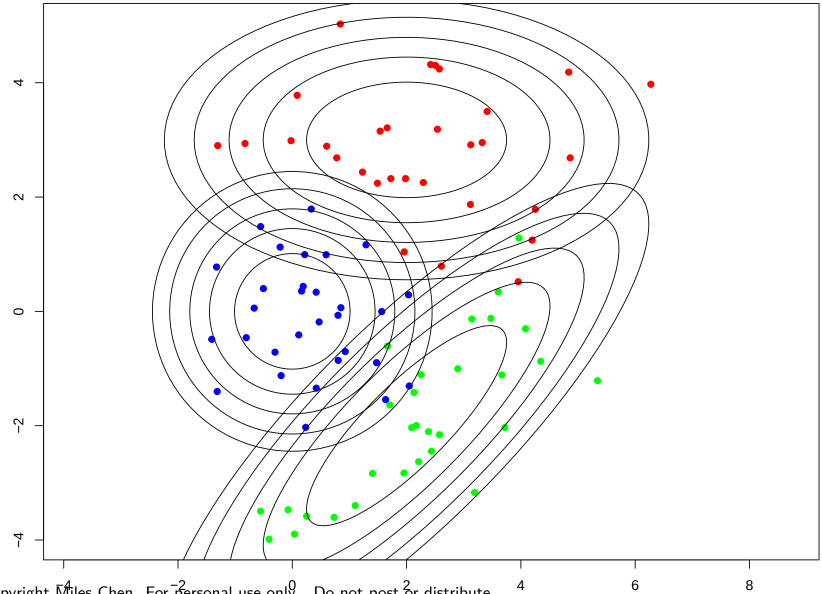
We'll generate data from 3 classes.

```r
library(mvtnorm) # to generate multivariate normal data
set.seed(150)

mu_A = c(2, 3)
sigma_A <- matrix(c(3, 0, 0, 1), nrow = 2) # x1 and x2 are independent
Xa <- rmvnorm(30, mu_A, sigma_A)

mu_B = c(0, 0)
sigma_B <- matrix(c(1, 0, 0, 1), nrow = 2) # x1 and x2 are independent
Xb <- rmvnorm(30, mu_B, sigma_B)

mu_C = c(2,-2)
sigma_C <- matrix(c(3, 2.5, 2.5, 3), nrow = 2) # x1 and x2 are not independent
Xc <- rmvnorm(30, mu_C, sigma_C)

X  <- rbind(Xa, Xb, Xc)
```

Code to plot the data with M.V. Normal Contour lines (ellipses)

```r
library(ellipse) # draw ellipses
par(mar = c(2, 2, 0, 2))
plot(Xa, col = 'red', xlim = c(min(X[, 1]), max(X[, 1])), xlab = "X1",
     ylim =  c(min(X[, 2]), max(X[, 2])), ylab = "X2", asp = 1, pch = 19)
points(Xb, col = 'blue', pch = 19)
points(Xc, col = 'green', pch = 19)
parameters <- list(list(mu_A, sigma_A), list(mu_B, sigma_B), list(mu_C, sigma_C))
for (l in parameters) {
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.4))
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.65))
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.8))
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.9))
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.95))
}
```

Simulated data



Pretending we don't know the values used to generate our data

We generated the data ourselves, so we know the population mean and population variance-covariance matrices.

We'll have to pretend that we don't know these values and we will estimate properties of the respective distributions using the training data.

Estimates of distribution parameters from the training data

With our labeled training data, we can estimate the mean and variance-covariance matrix of the distributions of the different classes.

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{x}_n$$

The MLE estimates of the mean vector for a particular class is simply the vector of sample means of the observations that belong to that class.

MLE estimates of distribution parameters from the training data

In R code this is as simple as:

```r
xbar_a <- colMeans(Xa); print(xbar_a)
```

```
## [1] 2.267457 2.821878
```

```r
xbar_b <- colMeans(Xb); print(xbar_b)
```

```
## [1]  0.3093744 -0.1093422
```

```r
xbar_c <- colMeans(Xc); print(xbar_c)
```

```
## [1]  2.250030 -1.919359
```

Estimating the variance

For the variance matrix, the MLE estimate is:

$$\widehat{\Sigma}_{\mathsf{MLE}} = \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{x}_n - \boldsymbol{\mu}_c)(\mathbf{x}_n - \boldsymbol{\mu}_c)^T$$

The MLE estimate of the variance can also be calculated as:

$$\widehat{\Sigma}_{\mathsf{MLE}} = \frac{1}{N_c} (\mathbf{X}_c - \boldsymbol{\mu}_c)^T (\mathbf{X}_c - \boldsymbol{\mu}_c)$$

Where $\mathbf{X}_c$ is the X matrix for all observations in class c.

Estimating the variance

However, the MLE estimate is biased. The unbiased estimator of the population variance is the sample variance:

$$\widehat{\Sigma} = \frac{1}{N_c - 1}(\mathbf{X}_c - \boldsymbol{\mu}_c)^T(\mathbf{X}_c - \boldsymbol{\mu}_c)$$

Variance estimates

```r
N_a <- dim(Xa)[1]
var_a = var(Xa); print(var_a)
```

```
##            [,1]       [,2]
## [1,]  2.9856216 -0.1880399
## [2,] -0.1880399  1.1980093
```

```r
N_b <- dim(Xb)[1]
var_b = var(Xb); print(var_b)
```

```
##            [,1]       [,2]
## [1,]  0.8893693 -0.1119304
## [2,] -0.1119304  0.9436161
```

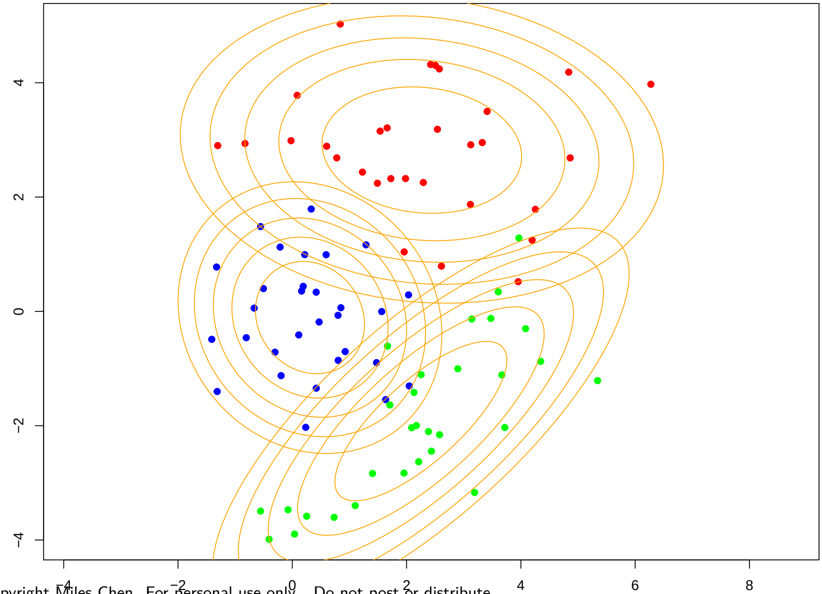
```r
N_c <- dim(Xc)[1]
var_c = var(Xc); print(var_c)
```

```
##          [,1]     [,2]
## [1,] 2.218390 1.568431
## [2,] 1.568431 1.903400
```

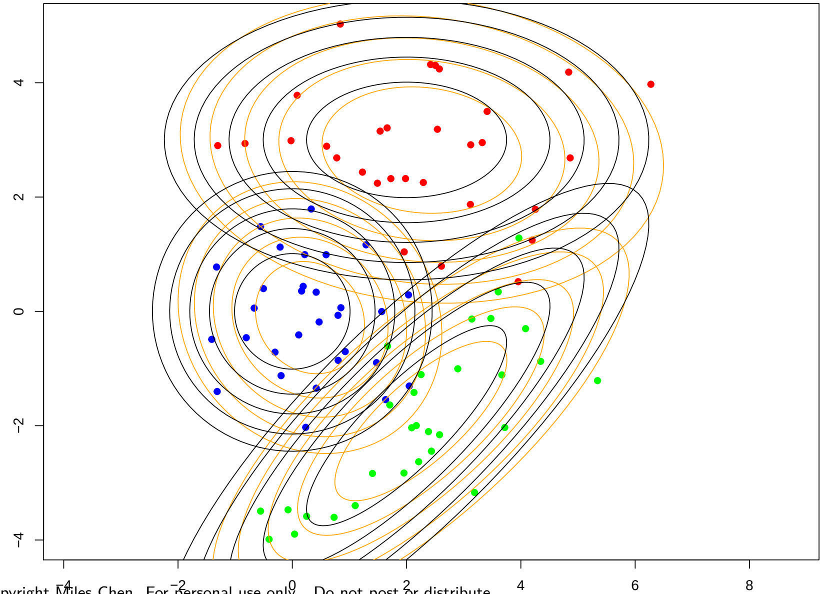
Plot of data with estimated contour lines

```r
par(mar = c(2, 2, 0, 2))
plot(Xa, col = 'red', xlim = c(min(X[, 1]), max(X[, 1])), xlab = "X1",
     ylim =  c(min(X[, 2]), max(X[, 2])), ylab = "X2", asp = 1, pch = 19)
points(Xb, col = 'blue', pch = 19)
points(Xc, col = 'green', pch = 19)
parameter_est <- list(list(xbar_a, var_a), list(xbar_b, var_b), list(xbar_c, var_c))
for (l in parameter_est) {
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.4), col = "orange")
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.65), col = "orange")
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.8), col = "orange")
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.9), col = "orange")
  lines(ellipse(l[[2]], centre = l[[1]], level = 0.95), col = "orange")
}
```

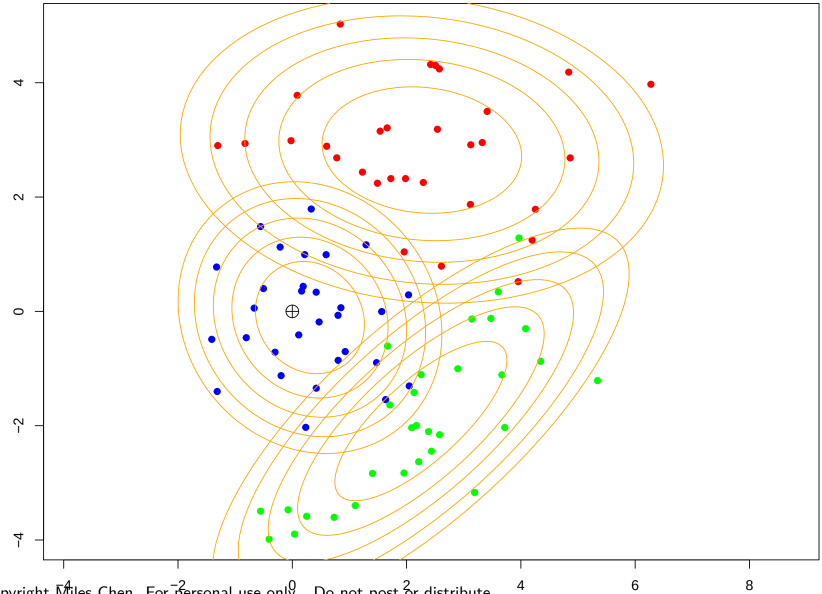
Plot of data with estimated contour lines



Plot with both “known” contours and estimated contours



Test Case



Bayes Classifier:

$$\Pr(T_{new} = c|\mathbf{x}_{new}, \mathbf{X}, \mathbf{t}) = \frac{\Pr(\mathbf{x}_{new}|T_{new} = c, \mathbf{X}, \mathbf{t}) \Pr(T_{new} = c|\mathbf{X}, \mathbf{t})}{\Pr(\mathbf{x}_{new}|\mathbf{X}, \mathbf{t})}$$

$$\text{Posterior Probability} = \frac{\text{Likelihood} \times \text{Prior Probability}}{\text{Marginal Probability}}$$