Normal model with variance known

Suppose that we have a sample $x_1, ..., x_n$ and we assume the following model

$$\left\{ N(\mu, \sigma^2); \mu \in \mathbb{R} \right\}$$

That is a normal distribution with mean μ, unknown and variance σ^2, known. We want to test

$$H_0 : \mu = \mu_0 \qquad \text{against} \qquad H_1 : \mu \neq \mu_0$$

The idea is to set up a test of hypothesis based on the Maximum Likelihood Estimator $\hat{\mu}$.

Likelihood Ratio Test (LRT)

By definition, the Likelihood Ratio Test (LRT) statistic is given by

$$\Lambda(x_1, ..., x_n) = \frac{\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{x_i - \hat{\mu}}{\sigma}\right)^2}}{\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \; e^{-\frac{1}{2}\left(\frac{x_i - \mu_0}{\sigma}\right)^2}}$$

Which simplifies to

$$\Lambda(x_1, ..., x_n) = exp\left(- \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(x_i - \hat{\mu}\right)^2 + \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(x_i - \mu_0\right)^2 \right)$$

Where $\hat{\mu}$ is the Maximum Likelihood estimator for μ and is equal to $\bar{x}$.

Now, it is interesting to note that twice the logarithm of $\Lambda(x_1, ..., x_n)$ gives us a convenient expression (see next slide)

Likelihood Ratio Test (LRT)

$$
\begin{aligned}
2ln\Big(\Lambda(x_1, \ldots, x_n)\Big) &= -\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(x_i - \bar{x}\Big)^2 + \frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(x_i^2 - \mu_0\Big)^2 \\
&= -\frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(x_i^2 - 2x_i\bar{x} + \bar{x}^2\Big) + \frac{1}{\sigma^2}\sum_{i=1}^{n}\Big(x_i^2 - 2x_i\mu_0 + \mu_0^2\Big) \\
&= -\frac{1}{\sigma^2}\Bigg(\sum_{i=1}^{n} x_i^2 - 2\bar{x}\sum_{i=1}^{n} x_i + \sum_{i=1}^{n}\bar{x}^2\Bigg) + \frac{1}{\sigma^2}\Bigg(\sum_{i=1}^{n} x_i^2 - 2\mu_0\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n}\mu_0^2\Bigg) \\
&= \frac{1}{\sigma^2}\Bigg(-\sum_{i=1}^{n} x_i^2 + 2\bar{x}\sum_{i=1}^{n} x_i - \sum_{i=1}^{n}\bar{x}^2 + \sum_{i=1}^{n} x_i^2 - 2\mu_0\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n}\mu_0^2\Bigg) \\
&= \frac{n}{\sigma^2}\Bigg(-\frac{1}{n}\sum_{i=1}^{n} x_i^2 + 2\bar{x}\frac{1}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}n\bar{x}^2 + \frac{1}{n}\sum_{i=1}^{n} x_i^2 - 2\mu_0\frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n}n\mu_0^2\Bigg) \\
&= \frac{n}{\sigma^2}\Bigg(2\bar{x}^2 - \bar{x}^2 - 2\mu_0\bar{x} + \mu_0^2\Bigg) \\
&= \frac{n}{\sigma^2}\Bigg(\bar{x} - 2\mu_0\bar{x} + \mu_0^2\Bigg) = n\Bigg(\frac{\bar{x} - \mu_0}{\sigma}\Bigg)^2
\end{aligned}
$$

Likelihood Ratio Test (LRT)

And by the CLT, we know that $\sqrt{n}(\frac{\bar{x}-\mu_0}{\sigma}) \sim N(0,1)$.

Since, by definition,

$$\sum_{i=1}^{n} Z_i^2 \sim \chi_n^2$$

It follows that

$$2ln(\Lambda(x_1,...,x_n)) \sim \chi_1^2$$

And therefore H_0 is rejected if $n\left(\frac{\bar{x}-\mu_0}{\sigma}\right)^2 > \chi^2_{1,1-\alpha}$.

Working example 1

Suppost that we get the following sample of size 40:

```
[1] 11.75  9.05  6.37 11.44 10.10 15.27  9.26 15.00 10.80 10.60 12.98 10.76
    13.69 13.99 10.19 14.10 13.79 13.36 14.69 13.72 10.77 11.12 15.66
    12.73 10.66  6.46 10.11  9.42 16.54 20.21 11.18 15.83  9.57 11.87 10.08
    10.68 13.52 15.39 14.93 11.63
```

The data are available here:
https://github.com/JRigh/Likelihood-Ratio-Tests/blob/main/data.csv

For what integer value(s) of μ_0 the LRT statistic is NOT rejected at a significance level of 5%? In other words, what are acceptable population mean for that dataset? Answer: 12 and 13 (see code below).

Working example 1 - R code

```r
# Set seed for reproducibility
set.seed(2023)

# Parameters
mu = 12 # true mean (unknown in practice)
n = 40; sigma <- 3; mu0 <- seq(5, 15, by = 1); alpha <- 0.05

# Generate artificial sample from a normal distribution
data <- rnorm(n, mean = mu, sd = sigma)  # True mean = 12

#  LRT statistic
lrt_statistic <- n * ((mean(data) - mu0)^2 / sigma^2)

# Calculate the critical value from the chi-squared distribution
critical_value <- qchisq(1 - alpha, df = 1)

# Perform the Likelihood Ratio Test
reject_null <- lrt_statistic > critical_value

# Print results
results = data.frame(mu0 = mu0,
                     lrt_statistic = lrt_statistic,
                decision = ifelse(lrt_statistic > critical_value, 'Yes', 'No'))
#    mu0 lrt_statistic decision
#...
# 6   10    22.1237569      Yes
# 7   11     6.7361255      Yes
# 8   12     0.2373829       No
# 9   13     2.6275293       No
# 10  14    13.9065645      Yes
# 11  15    34.0744886      Yes
```

Working example 1 - Python code

```
import numpy as np
import pandas as pd
from scipy.stats import chi2

# import the data
data = pd.read_csv("path/data.csv")

# Calculate LRT statistic for each mu0
lrt_statistic = n * ((np.mean(data['x']) - mu0)**2 / sigma**2)

# Calculate the critical value from the chi-squared distribution
critical_value = chi2.ppf(1 - alpha, df=1)

# Perform the Likelihood Ratio Test and make decisions
decisions = np.where(lrt_statistic > critical_value, "Yes", "No")

# Create a results data array
results = np.column_stack((mu0, lrt_statistic, decisions))
results

# ['10', '22.136480277777775', 'Yes'],
# ['11', '6.743146944444444', 'Yes'],
# ['12', '0.2387024999999998', 'No'],
# ['13', '2.623146944444445', 'No'],
# ['14', '13.89648027777778', 'Yes'],
```

Mixture models: introduction

Let Y be a random variable and y be any observed values of this random variable. Then Y obeys a finite mixture distribution if its density can be written as $f(y) = \lambda_1 f_1(y) + ... + \lambda_k f_k(y) = \sum_{j=1}^{k} \lambda_j f_j(y)$ provided that $\lambda_j > 0$ and $\sum_{j=1}^{k} \lambda_j = 1$.The weights λ_j are called the *mixing proportions* and $f_j(y)$ are called the *component densities*. Further, a k-component finite mixture model has the form:

$$f(y \mid \boldsymbol{\Psi}) = \sum_{j=1}^{k} \lambda_j f_j(y \mid \boldsymbol{\theta}_j)$$

In the case of a two-component mixture, that is with $k = 2$, we have the mixture parameter vector is $\boldsymbol{\Psi} = (\lambda_1, \lambda_2, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$; the number of components is k; the component density parameters are $\boldsymbol{\theta_1} = (\mu_1, \sigma_1^2)$ and $\boldsymbol{\theta_2} = (\mu_2, \sigma_2^2)$; the mixing proportions are λ_1 and $\lambda_2 = (1 - \lambda_1)$.

Detection of the number of components k - (1/2)

Among the many methods for determining the number of components in a mixture model, the Likelihood Ratio Test (LRT) is one commonly used in practice. A good explanation is given in McLachlan and Peel (2000). Roughly, the procedure proposes to test sequentially for a null hypothesis of a model with the smallest number of component k in the mixture against an alternative hypothesis of $k+1$ components, that is:

$$H_0 : k = k_0 \qquad \text{against} \qquad H_1 : k = k_0 + 1$$

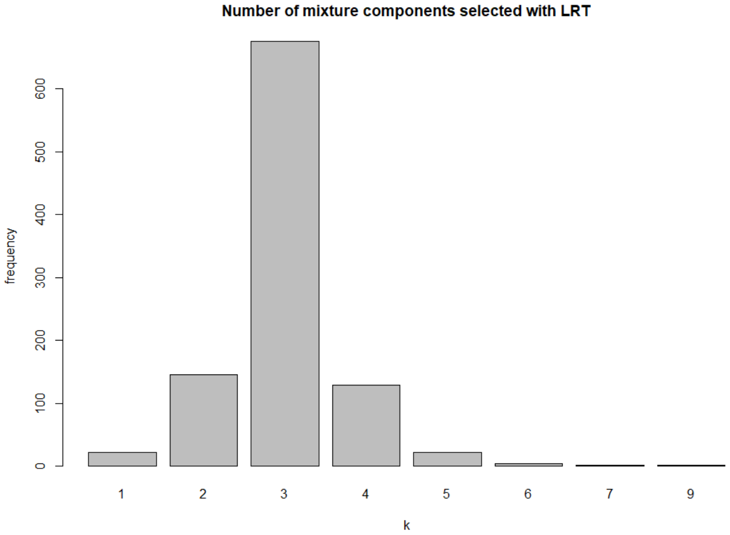
The test statistic considered is:

$$-2log(\Lambda) = 2\big(logL(\hat{\boldsymbol{\Psi}}_1) - logL(\hat{\boldsymbol{\Psi}}_0)\big) \ ,$$

Detection of the number of components k - (2/2)

with $\hat{\boldsymbol{\Psi}}_1$ and $\hat{\boldsymbol{\Psi}}_0$ being the maximum likelihood estimates for the model with the largest respectively the smallest number of components. A small value for Λ or similarly a large value for $-2log(\Lambda)$ will indicate strong evidence against the null hypothesis. The approximate distribution of the statistic under H_0 is not clearly identified, that is why we often resort to bootstrap in practice to take a decision with regards to the null hypothesis. A parametric bootstrap for this statistic is implemented in the mixtools package. This procedure is practically useful because it can be carried prior an EM or a bayesian analysis to have a guess about k, unlike information critera, introduced in the next section, that are used conditionally to a preliminary parameter estimation.

k as selected by the LRT



Detection of k - R code

```r
library(mixtools)

set.seed(2023)
data2 = rnormmix(n = 100, lambda = c(0.2, 0.3, 0.5), mu = c(1,5,8), sigma = c(1,1,1))

## we will first run it 1,000 times and record each time the number of components selected by the lrt
## hoping that one value for 'k' clearly stands out.
set.seed(2)
count.k = numeric(1000)
for(i in 1:1000)  {
  count.k[i]<- length(boot.comp(y=data2, max.comp=10, B=5,
                                sig=0.05, mix.type=c("normalmix"))$p.values)
}
count.k

sum(count.k==1)/1000  # percentage that LRT detect 3 components
sum(count.k==2)/1000  # percentage that LRT detect 3 components
sum(count.k==3)/1000  # percentage that LRT detect 3 components
sum(count.k==4)/1000  # percentage that LRT detect 4 components
sum(count.k==5)/1000  # percentage that LRT detect 5 components

# visualize the results
par(mfrow=c(1,1))
barplot(table(count.k), col="grey", xlab="k",
        ylab="frequency", main="Number of mixture components selected with LRT")

# from the barplot the LRT detect the presence of k=3 subpopulations in the mixture.
```

References

Bijma, F., Jonker M., Van der Vaart, A., An Introduction to Mathematical Statistics. Amsterdam University Press., 2016

R.V.Hogg and E.A.Tanis: Probability and Statistical Inference, Sixth Edition, Prentice Hall, Upper Saddle River, N.J., 2001.

McLachlan, G. and Peel, D., Finite mixture models. 0471006262, John Wiley & Sons., 2000

The R Project for Statistical Computing:
https://www.r-project.org/

Python:
https://www.python.org/

course notes