Yesterday…

- The triplets: Data, model, associated parameters
- We constructed our very first likelihood function
- We maximised a likelihood function by hand using calculus
- We maximised a likelihood function in R using `optim()` or `optimize()`
- We did many examples… quite productive actually…

This morning

- Properties of MLE
- More examples
- Likelihood-Ratio test

Properties of MLE

- Asymptotically unbiased
  – We are getting what we want
- Low variance (efficient)
  – Better than other estimators

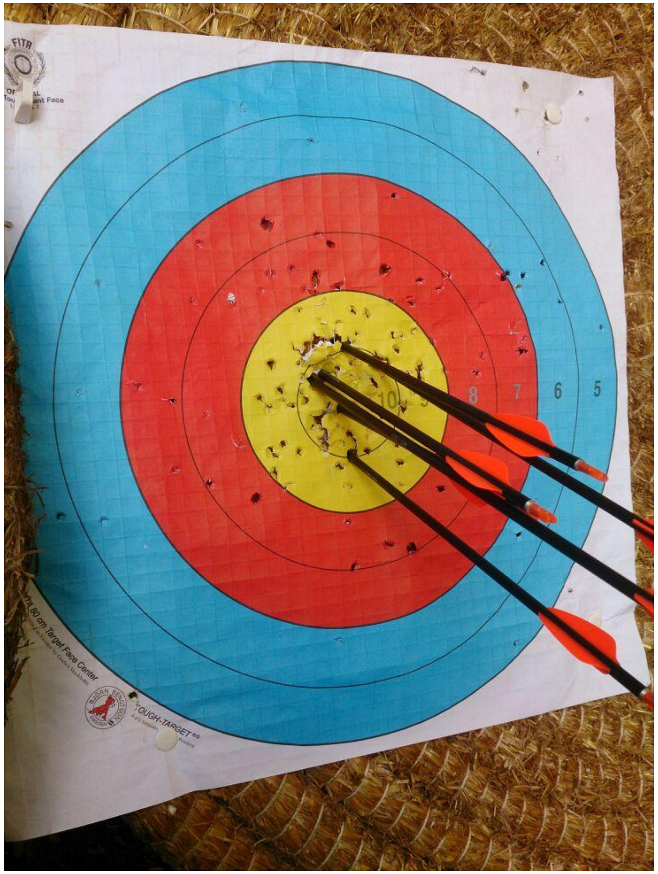


Photo credit: WP Luk

- Consistent: converges in probability to the true parameters
  - More samples, better estimate
- Asymptotically normal
  - If the model is true and we repeat the experiment for many times, the estimator is normally distributed
  - Central limit theorem??
  - Constructing confidence interval (more on this later)

Example: Logistic regression

- Binary response: dead or alive, yes or no, success or failure…
- Explanatory variable x is often called a risk factor (affect the risk/probability of "bad" outcome)
- Very useful in health science

#	State	Average cholesterol
1	Dead	5.0
2	Alive	4.4
3	Alive	3.4
4	Dead	3.7
5	Alive	3.6
6	Dad	4.7
…	…	…

- Recall: what’s the name of the r.v. that have only two outcomes?
- $Bernoulli(p)$. Logistic regression assumes that each individual a Bernoulli distribution with the “success” probability p.
- We try to associate p with our risk factor (linear predictor).
- $y_i \sim Bernoulli(p_i), where\ p_i = \eta^{-1}(a + bx_i)$

- What’s the form of the function η^{-1}?

- In logistic regression, $\eta^{-1}(x) = \frac{\exp(x)}{1+\exp(x)}$
- $expit(x)$. The inverse of $logit(p)$

- $\eta^{-1}(x)$ is bounded by zero and one (remember, it's the probability of success) regardless the value of x

- Let's construct the likelihood function

- Two parameters: a and b

$$L(a,b) = \prod_{i=1}^{n} f(y_i) = \prod_{i=1}^{n} [p_i^{y_i}(1-p_i)^{1-y_i}]$$

$$= \prod_{i=1}^{n} [expit(a+bx_i)^{y_i}(1-\exp(a+bx_i))^{1-y_i}]$$

- Take to log of the likelihood function

$$l(a,b) = \sum_{i=1}^{n} \{y_i \ln[\exp(a+bx_i)] + (1-y_i)\ln[1-expit(a+bx_i)]\}$$

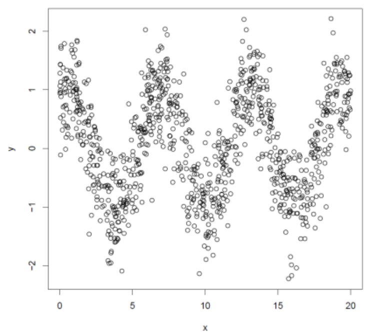
- It becomes a function of a and b only (with known y_i and x_i). We can maximise the (log-) likelihood function w.r.t. a and b.

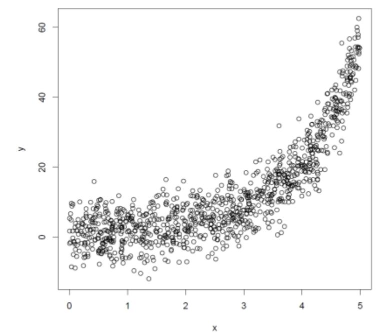
Non-standard regression

- The good thing of learning MLE is that you can write down your own models and likelihood functions
- Especially for non-standard cases where no “instant meals” are available

- $y_i = \mathrm{asin}(bx_i + c) + d + \epsilon_i$

- $y_i = \exp(mx_i + b) + \epsilon_i$





- e.g. rainfall model. Very likely to be non-linear.

- It may be possible to fit these models with built-in `glm()`, but obviously we can do it with MLE.

Likelihood-Ratio Test

- Model selection

- Let M1 and M2 be two models, and M1 is nested in M2. If M2 has $d2$ parameters and M1 has $d1$ parameters ($d2 > d1$), then $D = 2 * (\ln(L2) - \ln(L1))$ is approximately a chi-square distribution with $d2 - d1$ degrees of freedom.

- The procedure is as follows:
  - Fit M1 to the data, obtain the MLE and record down the log-likelihood value
  - Fit M2 to the data, obtain the MLE and record down the log-likelihood value
  - Compute $D = 2 * (\ln(L2) - \ln(L1))$ using their log-likelihood values
  - Look up χ^2_{d2-d1} table for critical value. Accept M1 as the simplified model if D is smaller than the critical value

- Rationale:
  - The larger the likelihood value the better the model
  - M2 fits the data better as it has more parameters than M1, therefore M2 has a larger log-likelihood value
  - M1 is a simplified model, with less explanatory power than M2, hence a smaller log-likelihood value
  - D measures the difference in ‘explanatory power’
  - If the parameters dropped by M1 are ‘unimportant’, then the explanatory power of M1 is close to M2, hence small value of D

Linear regression: test for intercept

- In `recapture.data`, we may think (biologically) that the intercept should be zero, because if a rabbit is captured "within zero days", then there should be no difference in body length

- We let M1 be a linear regression model without an intercept i.e. $y_i = bx_i + \varepsilon_i$ (Two parameters)

- We let M2 be the full linear regression model as before i.e. $y_i = a + bx_i + \varepsilon_i$ (Three parameters)

- Clearly M1 is a special case of M2 with intercept $a = 0$. We say M1 is nested in M2.

Log-likelihood function for M1

```
# THE LOG-LIKELIHOOD FUNCTION FOR M1 WITHOUT AN INTERCEPT
regression.no.intercept.log.likelihood<-function(parm, dat)
{
# DEFINE THE PARAMETERS
# NO INTERCEPT THIS TIME
?????
?????

# DEFINE THE DATA
# SAME AS BEFORE
x<-dat[,1]
y<-dat[,2]

# DEFINE THE ERROR TERM, NO INTERCEPT HERE
error.term<-?????

# REMEMBER THE NORMAL pdf?
density<-dnorm(error.term, mean=0, sd=sigma, log=T)

# THE LOG-LIKELIHOOD IS THE SUM OF
return(sum(density))
}
```

Log-likelihood function for M1

```
# THE LOG-LIKELIHOOD FUNCTION FOR M1 WITHOUT AN INTERCEPT
regression.no.intercept.log.likelihood<-function(parm, dat)
{
# DEFINE THE PARAMETERS
# NO INTERCEPT THIS TIME
b<-parm[1]
sigma<-parm[2]

# DEFINE THE DATA
# SAME AS BEFORE
x<-dat[,1]
y<-dat[,2]

# DEFINE THE ERROR TERM, NO INTERCEPT HERE
error.term<-(y-b*x)

# REMEMBER THE NORMAL pdf?
density<-dnorm(error.term, mean=0, sd=sigma, log=T)

# THE LOG-LIKELIHOOD IS THE SUM OF
return(sum(density))
}
```

Performing likelihood-ratio test

```
# PERFORMING LIKELIHOOD-RATIO TEST
M1<-optim(par=c(1,1), regression.no.intercept.log.likelihood,
          dat=recapture.data, method='L-BFGS-B',
          lower=c(-1000,0.0001), upper=c(1000,10000),
          control=list(fnscale=-1), hessian=T)
M2<-optim(par=c(1,1,1), regression.log.likelihood,
          dat=recapture.data, method='L-BFGS-B',
          lower=c(-1000,-1000,0.0001), upper=c(1000,1000,10000),
          control=list(fnscale=-1), hessian=T)

# THE TEST STATISTIC D
D<-2*(M2$value-M1$value)
D
```

```
[1] 3.047676
```

```
# CRITICAL VALUE
qchisq(0.95, df=1)
```

```
[1] 3.841459
```

We accept the hypothesis that the intercept is zero at $\alpha = 0.05$ (Same conclusion is drawn from `lm()` using anova table)

Model selection

- AIC is a tool to determine which of two models is better by weighting the improved fit of more complex models against their larger number of parameters.

- $AIC = -2\mathrm{l}(\hat{\theta}) + 2K$, where $\mathrm{l}(\hat{\theta})$ is the maximised log-likelihood and K is the number of parameters in the model

- Find the model with the lowest AIC value

Exercise: Non-constant variance regression

- Again, back to the `recapture.data`, we observe that the variance of the response is increasing with `day`.
- Can we incorporate non-constant variance in our regression?

- Not sure about the build-in `lm()` or `glm()`. Transformation of variables may help, but it is relatively simple MLE.

- The only trick is to write down the desired log-likelihood function in R.

- How about $\varepsilon_i \sim N\left(0, x_i^2 \sigma^2\right)$? Variance of residuals increases with days before recaptured?

Log-likelihood function: non-constant variance

```
# THE LOG-LIKELIHOOD FUNCTION FOR M1 WITHOUT AN INTERCEPT
regression.non.constant.var.log.likelihood<-function(parm, dat)
{
# DEFINE THE PARAMETERS
# NO CHANGE FROM M1
b<-parm[1]
sigma<-parm[2]

# DEFINE THE DATA
# SAME AS BEFORE
x<-dat[,1]
y<-dat[,2]

# DEFINE THE ERROR TERM, NO INTERCEPT HERE
error.term<-(y-b*x)

# REMEMBER THE NORMAL pdf
density<-dnorm(error.term, mean=0, sd=x*sigma, log=T)

# THE LOG-LIKELIHOOD IS THE SUM OF INDIVIDUAL DENSITIES
return(sum(density))
}
```

```
# MAXIMISE THE LOG-LIKELIHOOD
# HOW ABOUT CALLING IT M4?
M4<-optim(par=c(1,1), regression.non.constant.var.log.likelihood,
       dat=recapture.data, method='L-BFGS-B',
       lower=c(-1000,0.0001), upper=c(1000,10000),
       control=list(fnscale=-1))
M4
```

```
> M4
$par
[1] 3.483407 1.149874

$value
[1] -60.62583

$counts
function gradient
      25       25

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

This afternoon…

- Free ☺