Statistical Methods
(summer term 2024)

Maximum Likelihood Techniques

(based on original lectures by Prof. Dr. N. Christlieb and Dr. Hans-G. Ludwig)

Dr Yiannis Tsapras

ZAH – Heidelberg

Overview

- Maximum Likelihood estimation (MLE) is a method to estimate the parameters of a statistical model by maximizing the likelihood function, which measures how well the model matches the data

- MLE has two realms of application:
  - describe data and fit parameters
  - find estimator for property of population underlying a given sample

- Earliest techniques developed by Gauss, Legendre; modern formulation by R.A. Fisher (1912)

- The two fields of application are closely linked. One relies mostly on numerical techniques, while the other uses analytical techniques

Overview

- Maximum Likelihood estimation is a parametric method, which assumes that the data can be described by a specific functional form (a model) that depends on a set of parameters
  - This model must be a generative model, which describes how data is generated in terms of a probabilistic process, including the parameters that govern the distribution of the data
  - More formally, a generative model is a statistical model of the joint probability distribution $P(X, Y)$ on a given observable variable X and target variable Y, and it can be used to "generate" random instances of an observation x
- Objective of MLE: identify the parameter values that best describe the given data
- In simpler terms: MLE tries to find the optimal way to fit a distribution to the data
- The best fitting parameters could/should provide insights into the properties of the data

Maximum Likelihood Estimation: Basic idea

- A Probability Density Function (PDF) and a *likelihood function* are related but distinct concepts:
  - A PDF $p(x)$ is a function of x and indicates the probability density of observing a particular data point from the population (the parameters of the distribution are fixed)
  - A likelihood function $\mathcal{L}(\theta)$ takes the observed data as given and indicates the likelihood of the parameters θ given this data (the data are fixed)

- For continuous PDFs, the likelihood is often a high-dimensional probability density with respect to the data x_i, not with respect to the parameters

- To illustrate the difference, consider a random scalar variable X with a given PDF $p(x)$
  - Let's say that $p(x)$ is a univariate Gaussian distribution with two parameters, μ and σ, which are fixed
  - A sample of X is taken: $x_1, x_2, \ldots, x_n$, assuming all x_i are mutually independent for simplicity
  - The joint probability density of these observations is the product of their individual probabilities $\mathcal{L} = p(x_1) \times p(x_2) \times \ldots \times p(x_n)$

Maximum Likelihood Estimation: Basic idea

- Once we have taken the sample $x_1, x_2, \ldots, x_n$, we know the values
- What we do not know for the specific sample are the values for the parameters μ and σ, which will likely differ from the original population values due to sampling variability
- To estimate their values, we need to adjust these parameters until some optimization criterion is satisfied
- In this example, the joint probability density expressed as a function of μ and σ (with the data given) is the *likelihood function*:

$$\mathcal{L}(\mu, \sigma | x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} p(x_i | \mu, \sigma)$$

Maximum Likelihood Estimation: Basic idea

- The central idea of MLE is to find the parameter θ that maximizes the likelihood function $\mathcal{L}(\theta \mid x_1, x_2, \ldots, x_n)$ for the given data set

- A useful transformation is to take the natural logarithm of the likelihood function to obtain the *log-likelihood* function, $\ln \mathcal{L}$. Since $\ln$ is a monotonous function the maximum is the same

$$\ln \mathcal{L} = \sum_{i=1}^{n} \ln p(x_i|\theta)$$

This is computationally simpler and numerically more stable than multiplying probabilities directly

- To find the maximum of the log-likelihood, we set the derivative with respect to θ to zero: $\frac{\partial \ln \mathcal{L}}{\partial \theta}\big|_{\theta=\tilde{\theta}} = 0$

- $\tilde{\theta}$ is called the maximum likelihood estimate of the true value of θ

The standard case: normal distribution

- Consider x_i drawn from a normally distributed population, so that for each $x \in \{x_i\}$

$$p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

so that the log-likelihood function is

$$\ln \mathcal{L}(\mu,\sigma|\{x_i\}) = -n\ln\sigma - \frac{n}{2}\ln 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

(work it out $\rightarrow$blackboard)

- Maximizing $\ln\mathcal{L}$ has a close connection to the least-squares technique
  - In the least-squares approach, we minimize $\sum_{i=1}^{n}(x_i-\mu)^2$, and its equivalent here is maximizing the term $-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$ in the log-likelihood function

Example 1: Plot the log likelihood for population parameters and some others

- Draw 10 samples of size 3 each from an univariate normal distribution with $N(\mu = 2.0, \sigma = 0.5)$
- Compute the log-likelihood function for a range of μ values which should include the true μ of the population
- Repeat the process for 10 samples of size 30
- Plot all 20 log-likelihood functions on top of each other!
- Is the maximum of the log-likelihood functions always at μ?
- How does the shape change with increasing sample size?
  Try it out, then $\rightarrow$day06_example1.ipynb

Example 2: Derive an estimator for μ and σ of a Normal distribution

$\rightarrow$ blackboard

Example 3: Derive an estimator for expectation value λ of a Poisson distribution

- Let us now consider the Poisson distribution

- Since the variance and expectation value of a Poisson distribution is the same (denoted by λ), there are at least two possible ways to compute it from a sample of counts $r_1, r_2, \ldots, r_n$:

$$\tilde{\lambda} = \frac{1}{n}\sum_{i=1}^{n} r_i \quad \text{or perhaps} \quad \tilde{\lambda} = \frac{1}{n-1}\sum_{i=1}^{n}(r_i - \overline{r})^2 \ .$$

- Derive the ML estimator of λ. The likelihood function is given by:

$$\mathcal{L}(\lambda \,|\, \{r_i\}) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{r_i}}{r_i!}$$

Worked example: Fitting a straight line to data

- We have been using MLE to estimate parameters of probability distributions. Now, we will see how MLE can be applied to fit a model to data

- Consider the data points $(x_i, y_i \pm s_i)$ for $i = 1, \ldots, N$:
  - x_i and s_i are considered known and are not modeled
  - y_i are the observed data points with associated uncertainties s_i

- Our goal is to fit these data points with a linear model: $y_i = b + mx_i$

- Measurement of y_i is subject to noise; each y_i deviates from its true value due to a random offset drawn from a Gaussian distribution with standard deviation s_i:

$$y_i \sim N(y_{i,\mathrm{true}}, s_i^2) = N(b + mx_i, s_i^2)$$

- Implicit in this model is the assumption of statistical independence of the data points

- Parameters: We wish to determine the best-fitting values of b (intercept) and m (slope) that describe the data

Worked example: Fitting a straight line to data

- Given the model $y_i = b + mx_i + \epsilon_i$ where $\epsilon_i \sim N(0, s_i^2)$, we can write the log-likelihood function as:

$$\ln \mathcal{L}(b, m) = -\sum_{i=1}^{n} \ln s_i - \frac{n}{2} \ln 2\pi - \sum_{i=1}^{n} \frac{(y_i - b - mx_i)^2}{2s_i^2}$$

  - Here, $\mu = b + mx_i$ represents the expected value of y_i.
    Partial derivatives with respect to b and m, s_i are constants here

$$\frac{\partial \ln \mathcal{L}(b, m)}{\partial b} = \sum_{i=1}^{n} \frac{(y_i - b - mx_i)}{s_i^2} \tag{1}$$

$$\frac{\partial \ln \mathcal{L}(b, m)}{\partial m} = \sum_{i=1}^{n} \frac{(y_i - b - mx_i)x_i}{s_i^2}$$

Worked example: Fitting a straight line to data

- Setting these partial derivatives to zero yields the maximum likelihood estimates, resulting in the following $linear$ system of equations:

$$\begin{pmatrix} \sum_i s_i^{-2} & \sum_i x_i s_i^{-2} \\ \sum_i x_i s_i^{-2} & \sum_i x_i^2 s_i^{-2} \end{pmatrix} \begin{pmatrix} b \\ m \end{pmatrix} = \begin{pmatrix} \sum_i y_i s_i^{-2} \\ \sum_i y_i x_i s_i^{-2} \end{pmatrix}$$

- In the context of least-squares estimation, these equations are called the $normal\ equations$

- The square matrix on the LHS is called the $normal\ equation\ matrix$ $\mathbf{N}$.

- It is a symmetric $K \times K$ matrix, where K is the number of parameters
  - In our case, $K = 2$ for parameters b and m.

- The normal equation matrix encapsulates the weighted contributions of each data point to the estimation process
  - Each element in the matrix is a sum of weighted values, where the weights are inversely proportional to the variance s_i^2 of the observations

Worked example: Fitting a straight line to data

- Maximum likelihood estimation (MLE) always reduces to weighted least squares if the following conditions are met:
  - the data model is linear in the parameters. This means the model can be written as: $\theta_1 f_1 + \theta_2 f_2 + \ldots + \theta_K f_K$, where θ_i are the parameters and f_i are known arbitrary functions of auxiliary data
  - independent, known Gaussian errors are assumed. This implies that the errors in the measurements are uncorrelated and have a normal distribution with a known variance

- In such cases, MLE and least squares provide equivalent parameter estimates

- The normal equations derived from the MLE process can be solved using standard linear algebra procedures to obtain the best fitting parameters $\tilde{b}$ and $\tilde{m}$.
  - In R, you can use the `solve()` function to solve the system of normal equations.
  - To ensure clarity and manageability of your matrices, it's helpful to assign names to the columns and rows using `rownames()` and `colnames()` functions.

Sample R code for solving *normal equations*

```
# Example R code for solving normal equations
N <- matrix(c(sum(1/s^2), sum(x/s^2), sum(x/s^2),
              sum(x^2/s^2)), nrow=2, ncol=2)
y <- c(sum(y/s^2), sum(y*x/s^2))

# Assigning names to rows and columns
rownames(N) <- colnames(N) <- c("b", "m")

# Solving the normal equations
params <- solve(N, y)
b_est <- params["b"]
m_est <- params["m"]

cat("Estimated parameters:\n")
cat("b =", b_est, "\n")
cat("m =", m_est, "\n")
```

Interlude: correlation and regression

- There is a close relationship between correlation analysis and fitting straight lines by the least squares method. Consider n data points (x_i, y_i) and the following abbreviations:

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \qquad \sigma_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2$$

$$\mathrm{Cov}[x, y] = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

How is the function `cov` defined in `R`?

- Here, $\overline{x}$ and $\overline{y}$ are the arithmetic means of x_i and y_i, respectively

- It turns out that the estimated slope $\tilde{m}$, intercept $\tilde{b}$ of a fitted straight line, and the correlation coefficient r between x_i and y_i can be written as:

$$\tilde{m} = \frac{\mathrm{Cov}[x, y]}{\sigma_x^2} \qquad \tilde{b} = \overline{y} - \tilde{m}\,\overline{x} \qquad r = \frac{\mathrm{Cov}[x, y]}{\sigma_x \sigma_y} = \tilde{m}\,\frac{\sigma_x}{\sigma_y}$$

How good is the Maximum Likelihood fit?

- After determining the best fitting parameters, the next question is: how good is the fit?

- Intuitively, the goodness of fit is related to the deviations between data points y_i and the fitted line

- If the model of the data is correct and linear then the *sum of weighted residuals*

$$Q = \sum_{i=1}^{N} (y_i - \tilde{b} - \tilde{m}\, x_i)^2 s_i^{-2}$$

  is distributed as a χ^2-distribution with $N - K$ *degrees of freedom*, where N is the number of data points and K the number of parameters fitted

- Q is often referred to as the "χ^2 of the fit"

- The χ^2 distribution with k degrees of freedom, denoted by χ^2_k, has the following properties

$$\mathrm{E}\left[\chi^2_k\right] = k \quad \text{and} \quad \mathrm{Var}\left[\chi^2_k\right] = 2k$$

How good is the Maximum Likelihood fit?

- The fit is considered "good" if Q is around $(N-K) \pm \sqrt{2(N-K)}$
  - If Q is much larger than $N-K$, the fit may be poor, indicating that the model may not accurately describe the data
  - If Q is much smaller than $N-K$, the model might be overfitting the data

- Statistical fluctuations – like always – are possible, so some deviation from the expected range is normal

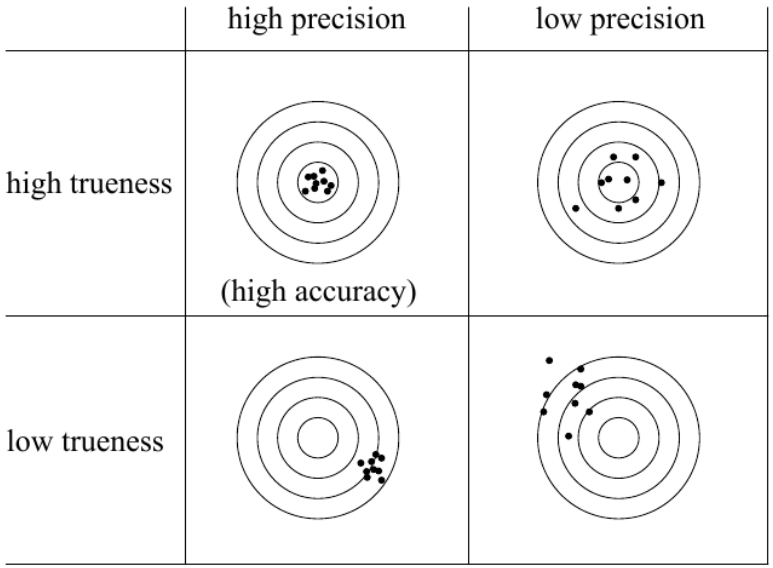
- Sometimes the "reduced" $\chi^2_{\mathrm{red}} \equiv \frac{Q}{N-K}$ is used
  - rough rule of thumb for an acceptable fit: $\chi^2_{\mathrm{red}} \approx 1$

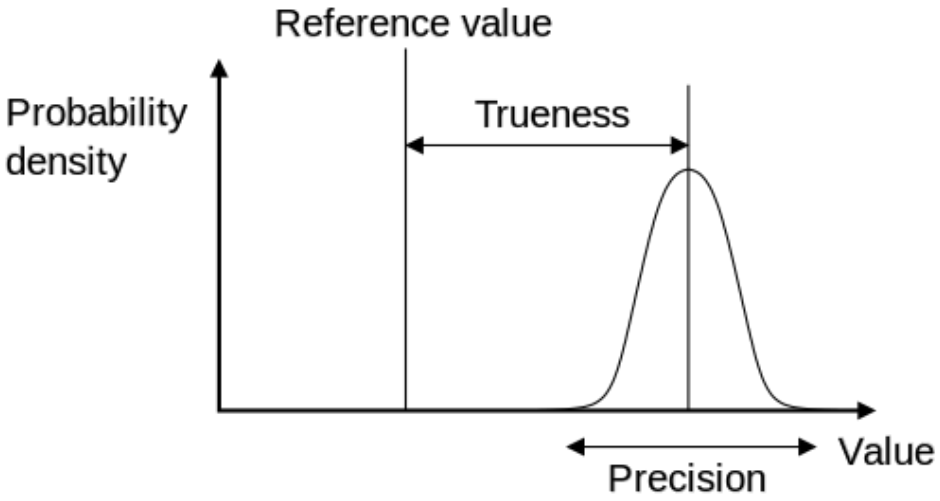
- Another important question is: how precise are the fitted parameters? This is a separate issue...
  - If the error bars are large the fit can be good but the parameters may not be precise
  - Conversely, if the error bars are small because uncertainties have been underestimated (or the data model is incorrect), the fitted parameters can be very precise but the fit can be very poor

Interlude on precision, trueness, accuracy

- In Physics, quantities without error bars have little value, as they lack information about their reliability and uncertainty

- ISO definitions:
  - **Precision**: The closeness of repeated measurements to each other
  - **Trueness**: The closeness of the mean of the measurement results to the actual (true) value
  - **Accuracy**: The closeness of a measurement to the true value, combining both precision and trueness





What is the precision of the estimated parameters in MLE?

- The precision of estimated parameters in MLE is related to the "sharpness" or curvature of the peak of the likelihood function.

- For the given case (linear model with Gaussian errors), the covariance matrix of the vector $\boldsymbol{\theta}$ of fitted parameters is given by:

$$\mathrm{Cov}\left[\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\right] \equiv \begin{pmatrix} \mathrm{Cov}\left[\tilde{b}, \tilde{b}\right] & \mathrm{Cov}\left[\tilde{b}, \tilde{m}\right] \\ \mathrm{Cov}\left[\tilde{m}, \tilde{b}\right] & \mathrm{Cov}[\tilde{m}, \tilde{m}] \end{pmatrix} = \left(-\frac{\partial^2 \ln \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathrm{T}}}\bigg|_{\tilde{\boldsymbol{\theta}}} \right)^{-1}$$

- This equation shows that the covariance of the fitted parameters is given by the negative inverse of the Hessian matrix (matrix of second derivatives) of the log-likelihood function

- The negative Hessian (non-inverted matrix) is known as the *information matrix* or *Fisher information*
  - The Fisher information matrix provides a measure of the *amount of information* that the data contains about the parameters. A high Fisher information value indicates that the parameter is estimated with greater precision

What is the precision of the estimated parameters in MLE?

- In the linear regression problem, the normal equations matrix $\mathbf{N}$ is the information matrix. This means that:

$$\mathrm{Cov}\left[(\tilde{b},\tilde{m})^{\mathrm{T}},(\tilde{b},\tilde{m})\right]=\mathbf{N}^{-1}$$

- The normal equations matrix $\mathbf{N}$ is formed as follows:

$$\mathbf{N}=\begin{pmatrix}\sum_i s_i^{-2} & \sum_i x_i s_i^{-2}\\ \sum_i x_i s_i^{-2} & \sum_i x_i^2 s_i^{-2}\end{pmatrix}$$

- The inverse of this matrix, $\mathbf{N}^{-1}$, provides the covariance matrix of the estimated parameters $(\tilde{b},\tilde{m})$.

- The diagonal elements of $\mathbf{N}^{-1}$ give the variances of $\tilde{b}$ and $\tilde{m}$:

$$\mathrm{Cov}\left[\tilde{b},\tilde{b}\right]=[\mathbf{N}^{-1}]_{11},\quad \mathrm{Cov}[\tilde{m},\tilde{m}]=[\mathbf{N}^{-1}]_{22}$$

  - These can be used to specify confidence limits on a particular parameter

What is the precision of the estimated parameters in MLE?

- The off-diagonal elements of $\mathbf{N}^{-1}$ provide the covariances between $\tilde{b}$ and $\tilde{m}$:

$$\mathrm{Cov}\left[\tilde{b}, \tilde{m}\right] = \mathrm{Cov}\left[\tilde{m}, \tilde{b}\right] = [\mathbf{N}^{-1}]_{12}$$

  - These covariances can be used to calculate the correlation coefficients
  - In R, you can use the function `cov2cor()` to convert a covariance matrix to a correlation matrix

- In the present case of weighted linear regression the covariance matrix of the parameters depends only on x_i, s_i but *not* on the data y_i. This means it can be calculated *even in the absence of real data*, giving an a-priori estimate of the precision of a parameter that can be obtained in a given set-up. This is extremely useful when is comes to the *planning of experiments*

- Title of a book of R.A. Fisher: *The Design of Experiments* (1935)
  (also introduces the concept of the *null hypothesis* → later)

→ example Blackboard and `day06_linefit_demo.ipynb`

More general MLE

- Previously, we discussed the simple case of MLE
  - linear in parameters
  - Gaussian errors (albeit $heteroscedastic$ errors)
  - independent errors
  - However, MLE can handle more complex situations:
    - ⋆ Errors in both x_i and y_i that need to be fitted
    - ⋆ Correlated errors between y_i and x_i
    - ⋆ Non-linear models for the data

- MLE more general than being restrictic to Normal PDF
  (have already seen Poisson example)

- Key points to consider:
  - The statements previously made are approximately correct in the limit of large sample sizes or not very non-linear data models
  - This might sound worrisome, but MLE generally behaves well
  - Monte Carlo experiments can be conducted to verify analytical results

The Cramér-Rao bound (Rao 1945, Cramer 1946)

The Cramér-Rao inequality states: The variance of any unbiased estimator is at least as high as the inverse of the Fisher information. Formally: for any estimator (function) $\psi(\boldsymbol{\theta})$ of $\boldsymbol{\theta}$ the following inequality holds:

$$\sigma_\psi^2 \geq \frac{\partial \psi}{\partial \boldsymbol{\theta}}^{\mathrm{T}} \mathrm{E}\left[\left(-\frac{\partial^2 \ln \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathrm{T}}}\right)^{-1}\right] \frac{\partial \psi}{\partial \boldsymbol{\theta}}$$

- Example: take $\psi(\boldsymbol{\theta}) = \theta_1 \quad \Rightarrow \quad \sigma_{\theta_1}^2 \geq \mathrm{E}\left[\left(-\frac{\partial^2 \ln \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathrm{T}}}\right)^{-1}\right]_{(1,1)}$

- While the Cramér-Rao theorem is a statement about the likelihood function and not directly ML estimation, it shows that ML estimators can be nearly optimal as long as the expectation of the Hessian can be replaced by its value at the estimated parameters

- This means: while an ML estimator may be biased, any unbiased estimator (if it exists) cannot have a smaller variance (is more efficient) than the ML estimator

Practical approach for non-linear MLE

- We have seen that MLE might give biased estimates
- However, MLE becomes unbiased for large sample sizes (N vs $N-1$)
- Often (not too small samples, not too non-linear data models) MLE approaches the Cramér-Rao bound and we can (like in the linear case) approximate

$$\mathrm{Cov}\left[\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\right] \approx \left(-\frac{\partial^2 \ln \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathrm{T}}}\bigg|_{\tilde{\boldsymbol{\theta}}} \right)^{-1} \qquad \text{(mnemonic version!)}$$

Summary of $\mathcal{L}(\theta)$ and generalized MLE

- **Consistency**: for large sample sizes N, the MLE converges to the true parameter value. This means that as the number of data points increases, the MLE becomes more accurate.

- **Invariance**: If the MLE of θ is $\tilde{\theta}$ and $\Psi(\theta)$ is some arbitrary transformation of the parameters, then $\Psi(\tilde{\theta})$ is the MLE of $\tilde{\Psi}(\theta)$. This property ensures that transformations of MLE estimates are still MLE estimates

- **Large-sample efficiency**: For large samplie sizes N, the MLE is at least as accurate as any other estimator. This means that in large samples, MLE provides the smallest possible variance among unbiased estimators

- **Near-optimality**: While the MLE may be biased, any unbiased estimator (if it exists) cannot have a smaller variance (is more efficient) than the MLE. This property is linked to the Cramér-Rao bound

- **Sufficiency**: Under certain conditions, the likelihood function $\mathcal{L}(\theta \mid x_i)$ summarizes all the information there is about θ in the data (assuming the data model is correct!) – thus no need to keep the x_i (this is the data reduction principle)

Summary of $\mathcal{L}(\theta)$ and generalized MLE

- **Likelihood ratio**: asymptotically, $2 \ln \left[\mathcal{L}(\tilde{\boldsymbol{\theta}}) / \mathcal{L}(\boldsymbol{\theta}_{\text{true}}) \right] \sim \chi^2_k$. This means that the ratio of the likelihoods of the estimated parameters to the true parameters follows a chi-squared distribution with k degrees of freedom in large samples

- **Covariance**: $\operatorname{Cov}\left[\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}^{\mathrm{T}} \right] \geq \left(-\frac{\partial^2 \ln \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathrm{T}}} \Big|_{\tilde{\boldsymbol{\theta}}} \right)^{-1}$. This is approximately the Cramér-Rao bound, indicating that the covariance matrix of the MLE estimates is bounded by the inverse of the Fisher information matrix

Asymptotic behavior of the likelihood and confidence regions

- For large sample sizes, the ratio of the likelihoods evaluated at the estimated parameters and the true parameters follows a chi-square distribution: $2\ln\left[\mathcal{L}(\tilde{\boldsymbol{\theta}})/\mathcal{L}(\boldsymbol{\theta}_{\text{true}})\right] \sim \chi^2_k$, where k is the number of parameters $k = \dim(\boldsymbol{\theta})$

- This property allows us to construct *confidence intervals* and *confidence regions*
  - confidence interval: interval of "trust" for a single parameter, irrespective of the value of the other parameters ($\rightarrow$ marginalization)
    - ⋆ This provides a range of values for a single parameter that is likely to contain the true parameter value with a certain probability (e.g., 68% for 1σ confidence interval).
  - confidence region: (sub)region of "trust" of parameter space considering several (perhaps all) parameters simultaneously
    - ⋆ This extends the idea of a confidence interval to multiple parameters, creating a region in parameter space where the true parameter values are likely to be found.

Asymptotic behavior of the likelihood and confidence regions

- The difference between the estimated parameters and the true parameters, $\Delta\boldsymbol{\theta} \equiv \tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\mathrm{true}}$, follows a multi-variate normal distribution:

$$\mathrm{P}(\Delta\boldsymbol{\theta}) = \mathrm{const} \times \exp\left(-\frac{1}{2}\Delta\chi^2\right) = \mathrm{const} \times \exp\left(-\frac{1}{2}\Delta\boldsymbol{\theta}^{\mathrm{T}}\mathrm{Cov}\left[\boldsymbol{\theta}, \boldsymbol{\theta}^{\mathrm{T}}\right]^{-1}\Delta\boldsymbol{\theta}\right)$$

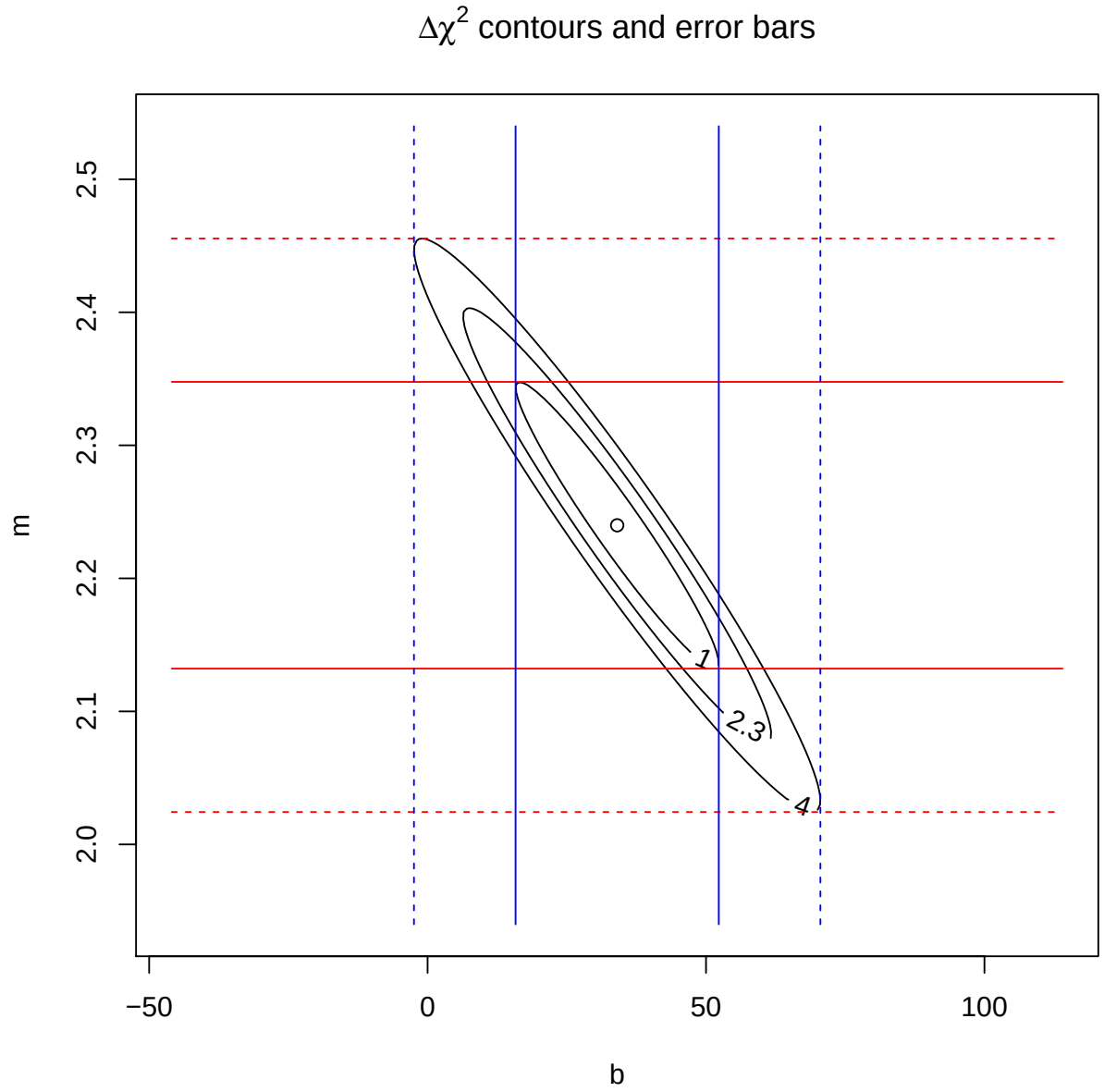
  where $\Delta\chi^2$ is related to the covariance matrix of the paramerers

  - This relation shows that $\Delta\chi^2$-contours correspond to confidence regions in the parameter space

- in practice, to calculate confidence intervals or regions, you need to evaluate the likelihood function over a grid of parameter values and identify the regions where the likelihood is within a certain threshold of its maximum value

- for example, in the case of straight-line fitting, you could vary m and b and calculate χ^2 for each combination, and then plot the contours of constant $\Delta\chi^2$

Example: likelihood ratio for straight line fitting



In linear regression, the change in χ^2, $\Delta\chi^2$, is a quadratic form near the Maximim Likelihood estimate. In the non-linear case only approximately (to leading order).

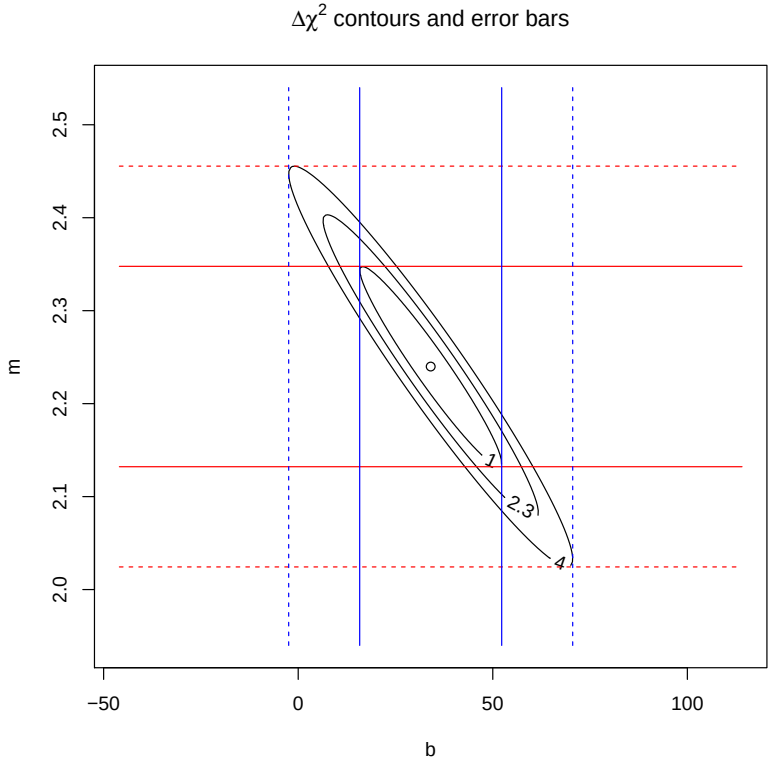
$$\Delta\chi^2(\Delta\boldsymbol{\theta}) = \Delta\boldsymbol{\theta}^{\mathrm{T}}\,\mathbf{N}\,\Delta\boldsymbol{\theta}$$

where $\Delta\boldsymbol{\theta}$ represents the difference between the estimated parameters and their true values

$$\Delta\boldsymbol{\theta} = \begin{pmatrix} b - b_{\mathrm{true}} \\ m - m_{\mathrm{true}} \end{pmatrix}$$

The contours in the $\Delta\chi^2$ plot represent regions of equal likelihood

$\Delta\chi^2$ and confidence levels



$\Delta\chi^2$ as a Function of Confidence Level and Degrees of Freedom						
	ν					
p	1	2	3	4	5	6
68.3%	1.00	2.30	3.53	4.72	5.89	7.04
90%	2.71	4.61	6.25	7.78	9.24	10.6
95.4%	4.00	6.17	8.02	9.70	11.3	12.8
99%	6.63	9.21	11.3	13.3	15.1	16.8
99.73%	9.00	11.8	14.2	16.3	18.2	20.1
99.99%	15.1	18.4	21.1	23.5	25.7	27.8

(from *Numerical Recipes*)

R: `qchisq(p,`ν`)`

- The solid lines in the $\Delta\chi^2$ plot represent the $\pm 1\sigma$ confidence interval for single parameter, and the dashed lines represent the $\pm 2\sigma$ interval
  - For two parameters, the $\Delta\chi^2$ value corresponding to a 68% confidence region is 2.3, not 1. This is because the confidence region is a *joint probability* over two parameters
  - The table on the right shows critical values of $\Delta\chi^2$ for different confidence levels and degrees of freedom

$\Delta\chi^2$ and confidence levels

- To determine the confidence region for a set of parameters:
  - Compute the χ^2 value at the best-fit parameters
  - Identify the contour where $\Delta\chi^2$ reaches the critical value for your desired confidence level
  - The region inside this contour represents the confidence region

- In R you can use the function `qchisq(p, ν)` to obtain the critical value of χ^2 for a given confidence level p and degrees of freedom ν