Model Parameter Estimation by Maximum Likelihood

Example I: The biased coin

Consider a data sequence $D = (x_1, x_2, \ldots, x_n)$ of bits $x_i \in \{0, 1\}$ which we belive are generated independently at random with the same probability. Call θ the **unknown** probabiliy of 1. The probability of the sequence D under this **model** is

$$P(D|\theta) = \prod_{i=1}^{n} \theta^{x_i} \; (1-\theta)^{1-x_i}$$

If D is observed (ie fixed), we study $P(D|\theta)$ as a function of θ. We call it the **likelihood**.

To **estimate** the **true parameter** θ of the model from which the data was generated we use the method of Maximum Likelihood choosing $\widehat{\theta} = \operatorname{argmax} P(D|\theta)$. For this parameter, the observed data have the highest probability. Equivalent we maximize the log–likelihood

$$\ln P(D|\theta) = \sum_{i=1}^{n} (x_i \ln \theta + (1-x_i) \ln(1-\theta)) = n_1 \ln \theta + (n - n_1) \ln(1-\theta)$$

Differentiating gives

$$\frac{d \ln P(D|\theta)}{d\theta} = 0 \qquad \longrightarrow \qquad \widehat{\theta} = \frac{n_1}{n} \ .$$

Example II: Gaussian density

The density of a <u>one dimensional Gaussian</u> random variable with *mean* $E(X) = \mu$ and variance $\sigma^2 = E(X-\mu)^2$ is given by

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\sigma^2}} .$$

The goal is to estimate μ, σ^2 from a set of data $D = (x_1, x_2, \ldots, x_n)$. Each data is assumed to be drawn independently from $p(x|\mu, \sigma^2)$. Maximizing the Likelihood is equivalent to *minimizing*

$$-\ln p(D|\mu, \sigma^2) = \frac{1}{2} \sum_{i=1}^{N} \left\{ \frac{(x_i - \mu)^2}{\sigma^2} + \ln(2\pi\sigma^2) \right\}$$

Minimization with respect to μ and σ^2 leads to the *Maximum Likelihood Estimates*

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{N} x_i$$

$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \widehat{\mu})^2$$

Example III: Gaussian noise and Linear Regression

Observe a set of input–ouput data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ with x = input, y = target values. Try to fit a linear function $y = w_0 + w_1 x$ to the data. We represent this as a probabilistic model and assume that n observations are generated as

$$y_i = w_0 + w_1 x_i + \text{ noise}_i$$

for $i = 1, \ldots, n$. For independent Gaussian noise of variance σ^2 we can write

$$p(y, x|\mathbf{w}) = p(y|x, \mathbf{w})p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - w_0 - w_1 x)^2}{2\sigma^2}} p(x)$$

The unknown parameters are $\mathbf{w} = (w_0, w_1)$ and σ^2.

Hence, the negative **log–likelihood** is

$$-\ln P(D|\mathbf{w}, \sigma^2) = \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_i)^2$$

and ML estimation of w_0 and w_1 becomes equivalent to *Least Squares* fitting!

Generalised linear models

Assume data generated as $y_i = f(x_i) + \nu_i$ for $i = 1, \ldots, N$, with $f(\cdot)$ unknown, ν_i i.i.d. $\sim \mathcal{N}(0, \sigma^2)$.

Polynomial regression:

$$f_{\mathbf{w}}(x) = \sum_{j=0}^{K} w_j x^j$$

allowing for different orders K. The **likelihood** is

$$p(D|\mathbf{w}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\sum_{i=1}^{N} \frac{(y_i - f(x_i))^2}{2\sigma^2}\right]$$

Exponential families

ML estimates look simple (analytically computable) for models from the so–called (*regular*†) **exponential families** which in their **canonical representation** are written as

$$p(x|\boldsymbol{\theta}) = f(x)\ \exp[\boldsymbol{\psi}(\boldsymbol{\theta}) \cdot \boldsymbol{\phi}(x) + g(\boldsymbol{\theta})]\ .$$

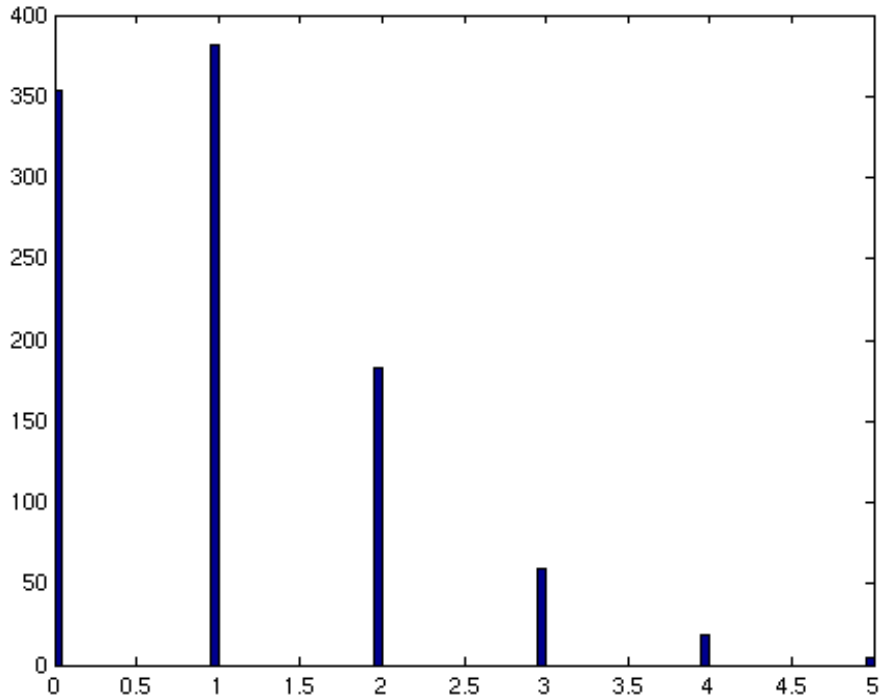
For a Gaussian, take $\boldsymbol{\psi}(\boldsymbol{\theta}) = (\mu/\sigma^2, 1/2\sigma^2)$ and $\boldsymbol{\phi}(x) = (x, -x^2)$.

(† regular means that the range of the data x is independent of the parameter θ).

Another exponential family: Poisson distributions

$$p(n|\theta) = e^{-\theta}\frac{\theta^n}{n!}$$

for $n = 0, 1, 2, \ldots$. This shows the distribution for $\theta = 1$.



Example: Multinomial family

Let $\mathbf{n} = (n_1, \ldots, n_K)$, with $n_j \in N$ and $\sum_j n_j = n$, we define the Multimomial family as

$$P(\mathbf{n}|\boldsymbol{\theta}) = \frac{n!}{\prod_{j=1}^{K} n_j!} \prod_{j=1}^{K} \theta_j^{n_j}$$

where $\sum_{j=1}^{K} \theta_j = 1$. Useful for **histogramme** data (counts, e.g. in *Bag of words* model).

<u>Sufficiency:</u> Let $p(x|\theta)$ be a parametric familiy. A statistics $T(\mathbf{x})$ of the sample $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is called **sufficient** if the conditional probability

$$p(\mathbf{x}|T(\mathbf{x}) = t, \theta)$$

is independent of θ. Thus $T(\mathbf{x})$ incorporates all relevant information of the parameter $\mathbf{x}$!

For exponential families, $\mathbf{T}(\mathbf{x}) = \sum_{i=1}^{n} \boldsymbol{\phi}(x_i)$ is a sufficient statistics.

Properties of Estimators

- Parameter estimates $\widehat{\theta}(D)$ are random variables with respect to the random drawing of the data. The *bias* of an estimator is defined as $E_D(\widehat{\theta}) - \theta$ and its *variance* as $E_D\left(\widehat{\theta} - E_D(\widehat{\theta})\right)^2$, where the expectation E_D is over datasets which are drawn at random from a distribution with *true* parameter θ.

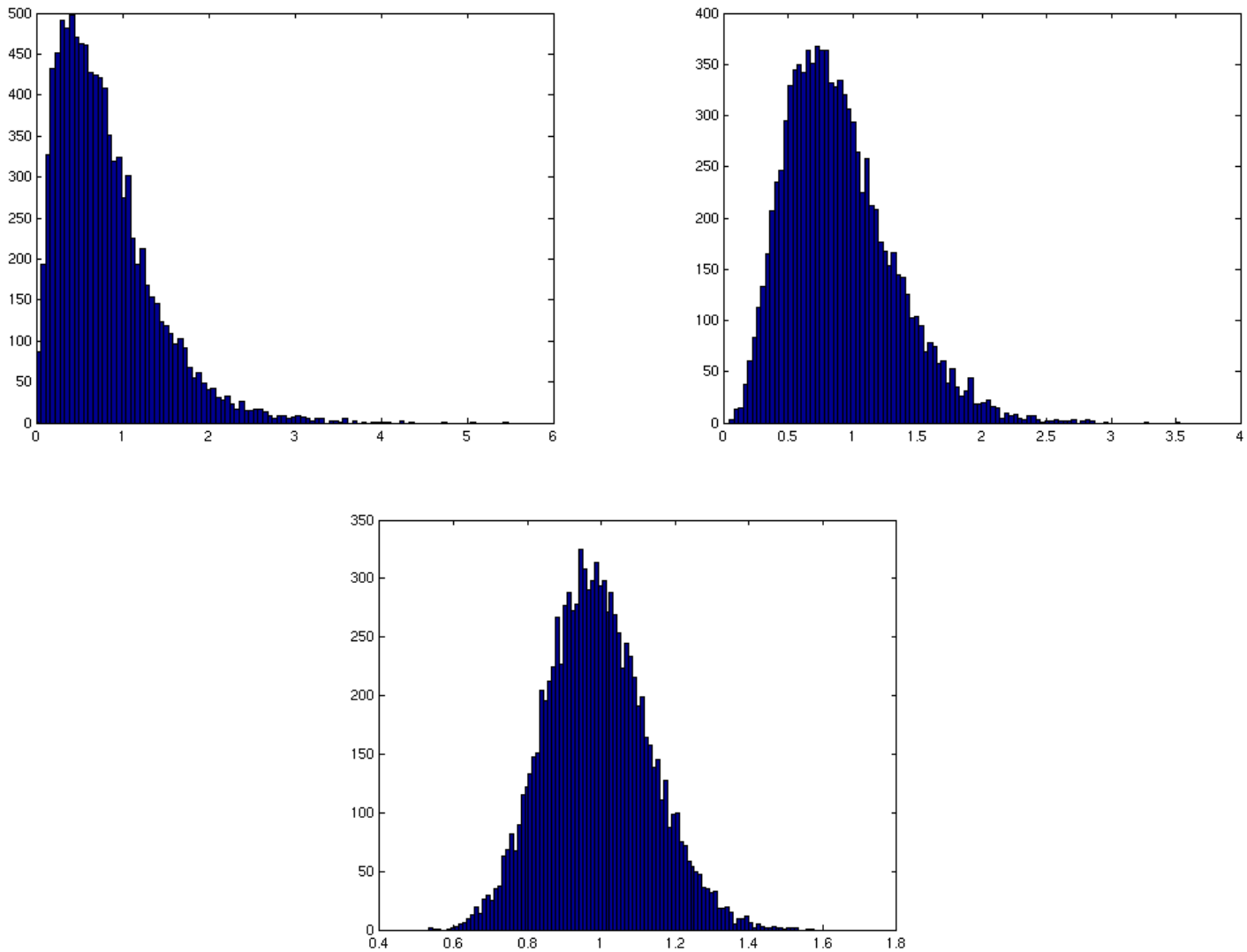
- "Good" estimators should become asymptotically *consistent*, i.e. the estimates should converge to the *true* parameters as $N \to \infty$. This means that bias and variance must go to 0 as $N \to \infty$.

- ML estimators are consistent under rather general circumstances. Note that

$$-\frac{1}{n} \ln P(D|\theta) = -\frac{1}{n} \sum_i \ln p(x_i|\theta) \to -E_D \ln p(x|\theta)$$

  Hence, minimizing $-\frac{1}{n} \ln P(D|\theta)$ becomes asymptotically equivalent of minimizing $KL(p_{\text{true}}, p_\theta)$!

ML estimation of the variance (10.000 repetitions) for $n = 5, 10, 100$



Efficiency & Rao–Cramér inequality

This limits the speed at which the estimate $\widehat{\boldsymbol{\theta}}$ approaches the true parameter θ on average. For a single (scalar) parameter

$$\mathrm{Var}(\widehat{\theta}) \geq \frac{(\partial_\theta E(\widehat{\theta}))^2}{nJ(\theta)}$$

with $J(\theta) = E_\theta \left[\frac{d \ln p(x|\theta)}{d\theta}\right]^2$.

Generalization to a k dimensional vector of parameters: For any real vector $(z_1, \ldots, z_k)$ (we specialise to **unbiased** estimators $E(\widehat{\theta}) = \theta$ for simpicity)

$$E\left(\sum_i z_i(\widehat{\theta}_i - \theta_i)\right)^2 \geq \frac{1}{n}\sum_{ij} z_i z_j (J^{-1}(\boldsymbol{\theta}))_{ij} \ , \tag{6}$$

with the **Fisher Information** matrix

$$J_{ij}(\theta) = \int dx \ p(x|\boldsymbol{\theta})\partial_i \ln p(x|\boldsymbol{\theta})\partial_j \ln p(x|\boldsymbol{\theta}) \ .$$

For $z_i \geq 0$, we can interprete the left hand side as a squared weighted average of the individual error components $\widehat{\theta}_i - \theta_i$. Estimators which fulfill these relations with an **equality**, are called **efficient**. Under weak assumptions, ML estimators are asymptotically efficient.

One can show that (under some technical conditions)

$$\widehat{\theta}_{ML} \sim \mathcal{N}\left(\theta, \frac{1}{n} J^{-1}(\theta)\right)$$

for $n \to \infty$. To use this result for the computation of error bars, we can use the approximation

$$J_{ij}(\boldsymbol{\theta}) \approx -\frac{1}{n} \partial_i \partial_j \sum_i \ln p(x_i | \widehat{\boldsymbol{\theta}}_{ML})$$

Note: A different representation of the Fisher Information is

$$J_{ij}(\boldsymbol{\theta}) = -\int dx \; p(x|\boldsymbol{\theta}) \partial_i \partial_j \ln p(x|\boldsymbol{\theta}) \; .$$

In the case, where the family $p(x|\theta)$ **does not contain the true distribution** $p(x)$ one has a similar result

$$\widehat{\theta}_{ML} \sim \mathcal{N}\left(\theta_0, \frac{1}{n} J^{-1} K J^{-1}\right)$$

for $n \to \infty$. where

$$J_{ij} = -\int dx\, p(x) \partial_i \partial_j \ln p(x|\theta_0) \ .$$

and

$$K_{ij} = \mathrm{COV}_p[\nabla \ln p(x|\theta_0)] \ .$$

with $\theta_0 = \arg\min D(p, p(\cdot|\theta))$ gives the model closest (in relative entropy) to the true distribution p.

S. Amari has developed a differential geometric (Information geometry) approach to estimation. Here, one defines a **metric** in parameter space by

$$||d\theta||^2 \propto \sum_{ij} d\theta_i J_{ij}(\theta) d\theta_j = d\boldsymbol{\theta}^T \mathbf{J}(\theta) d\boldsymbol{\theta}. \tag{7}$$

which reflects how well neighbouring distributions can be distinguished by an estimation based on random data. Assuming that the probability distribution of efficient estimators is Gaussian (at large n) with a covariance given by (6), the probability density that a point close to the true value θ will be the estimate for θ, depends only on the distance $||d\theta||$.

Online Learning

As a learning algorithm, one can use e.g. a gradient descent algorithm and iterate

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \eta \; \nabla_\theta \sum_{k=1}^{n} \ln p(x_k|\boldsymbol{\theta})$$

until convergence. This requires storage of all previous data.

Goal of online learning: Calculate new estimate only based on the new data point x_{n+1}, the old estimate $\widehat{\boldsymbol{\theta}}(n)$ (and possibly a set of other auxiliary quantities which have to be updated at each time step, but are much smaller in number than the entire set of previous training data).

Popular idea:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \eta(n) \; \nabla_\theta \ln p(x_{n+1}|\boldsymbol{\theta}(n))$$

If the algorithm should converge asymptotically, the learning rate $\eta(n)$ must be decreased during learning. A schedule $\eta \propto 1/n$ yields the fastest rate of convergence, but the prefactor must be chosen with

care, in order to avoid that the algorithm gets stuck away from the optimal parameter.

Natural gradient learning

S. Amari: Replace scalar learning rate $\eta(n)$ by a tensor. This is derived from the natural **distance** $||\Delta\theta||$ which reflects distances between probability distributions and is invariant against transformations of the parameters. A simple Euklidian distance will not satisfy this condition.

In the **natural gradient** algorithm the update is defined by a minimization of the training energy under the condition that $||\Delta\theta||^2$ is kept fixed. Solving the constrained variational problem for small $\Delta\theta$ yields

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \gamma_n \mathbf{J}^{-1}(\boldsymbol{\theta}(n)) \nabla_\theta \ln p(x_{t+1}|\boldsymbol{\theta}(n)).$$

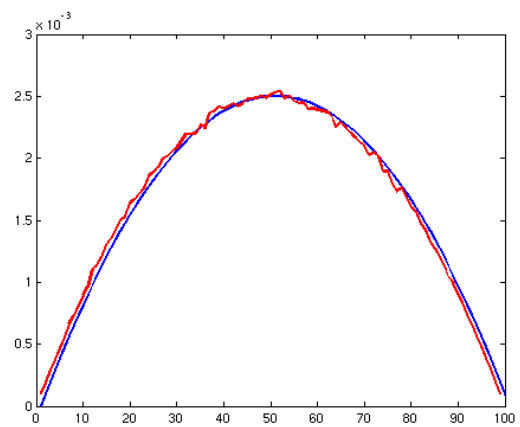
The differential operator $\mathbf{J}^{-1}(\boldsymbol{\theta}(n))\nabla_\theta$ is termed natural gradient. For the choice $\gamma_n = \frac{1}{n}$, one can show that the online algorithm yields *asymptotically efficient* estimation.

Example: Fisher Information

Bernoulli random variables

$p(x|\theta) = \theta^x (1-\theta)^{1-x}$ has $J(\theta) = \frac{1}{\theta(1-\theta)}$

$E(\widehat{\theta} - \theta)^2$ and $\frac{1}{J(\theta)n}$ as a function of θ



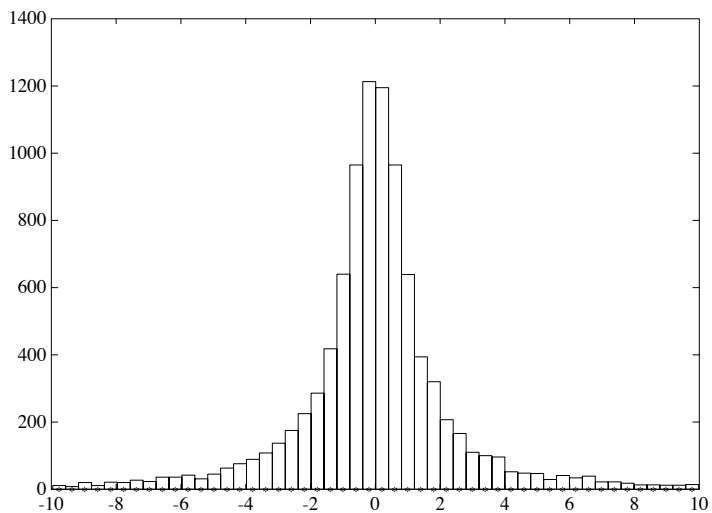
Cauchy density

$p(x|\theta) = \frac{1}{\pi(1+(x-\theta)^2)}$ has $J(\theta) = \pi/8$.

Estimating a Cauchy Density

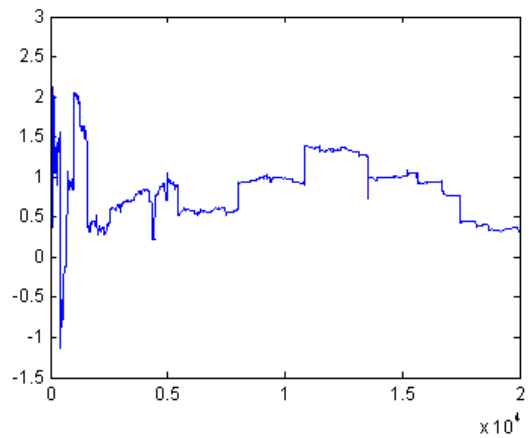
We consider the family of Cauchy densities given by

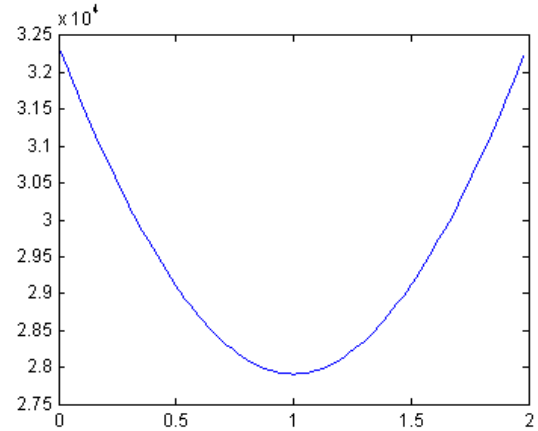
$$p(x|\theta) = \frac{1}{\pi(1 + (x - \theta)^2)} \ .$$



with location parameter θ.

Naive estimate $\widehat{\theta} = \frac{1}{n}\sum_i x_i$ negative log–likelihood $-\ln p(D|\theta)$.
(true $\theta = 1$).





Natural gradient $\theta_{n+1} = \theta_n + \frac{4(x_{n+1}-\theta_n)}{n(1+(x_{n+1}-\theta_n)^2)}$

Prediction θ_n (single run) Average error (10.000 runs) vs $1/n$.