Queens' College Cambridge

Data Science

Alistair O'Brien

Department of Computer Science

March 28, 2021

Contents

1 Probability Models

1.1 Specifying and Fitting Models

1.1.1 Parametric Models and Fitting Parameters

- A parametric probability model assumes a theoretical distribution (e.g. a Binomial, Normal, etc distribution). Theoretical distributions are parameterized by $\boldsymbol{\theta}$, the parameter vector.
- Conversely, a non-parametric model assumes no distribution.
- Estimating $\boldsymbol{\theta}$, denoted $\hat{\boldsymbol{\theta}}$ is referred to as *fitting the model*.

1.1.2 Maximum Likelihood Estimation

- The most popular method for learning parametric models.
- Consider a dataset $\langle x_i \rangle$ of a sample $\langle X_i \rangle$ of size n with a joint pmf $p_{X_1,\ldots,X_n}$. The joint pmf p depends on parameters $\boldsymbol{\theta}$, we emphasize this by denoting

$$p_{\boldsymbol{\theta}} = p_{X_1,\ldots,X_n}.$$

Definition 1.1.1. (**Likelihood Function**) For a dataset $\langle x_i \rangle$ with realizations $x_1, \ldots, x_n$ of a sample $\langle X_i \rangle$ of size n with joint pmf $p_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ is a vector of parameters, the likelihood function is

$$\mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) = p_{\boldsymbol{\theta}}(x_1, \ldots, x_n).$$

If the random sample consists of continuous random variables with a joint pdf $f_{\boldsymbol{\theta}}$, then the likelihood function is

$$\mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_n).$$

- The **log-likelihood function** is $\log \mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \boldsymbol{\theta})$.
- When the data are modelled as iid (independent and identical) samples, the likelihood factors into a product, so the log likelihood can be decomposed into a sum.

Definition 1.1.2. (**Maximum-likelihood estimator**) The **maximum likelihood estimator** (MLE) for the vector of parameters $\boldsymbol{\theta}$ is

$$\begin{aligned}
\text{MLE}[\boldsymbol{\theta}] &= \arg\max_{\boldsymbol{\theta}} \mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log \mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \boldsymbol{\theta})
\end{aligned}$$

since log is a monotonically increasing function.

- Note that $\text{MLE}[\boldsymbol{\theta}]$ is an estimator, and therefore should be a function $\delta(X_1, \ldots, X_n)$ where $\langle X_i \rangle$ is a sample of size n with a distribution F that is indexed by $\boldsymbol{\theta}$. (*Hence the MLE shouldn't contain any parameters, just random variables / realizations of the random variables*).

Example 1.1.1. (**MLE for Bernoulli Distribution**) For random sample $\langle X_i \rangle$ of size n s.t $X_1, \ldots, X_n \sim \text{Bern}(p)$. Recall that the pmf is

$$P(X = x) = p_X(x) = \begin{cases} p & x = 1 \\ 1 - p & x = 0 \end{cases}.$$

Hence the likelihood function is given by

$$\begin{aligned}
\mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid p) &= p_{(p)}(x_1, \ldots, x_n) \\
&= \prod_{i=1}^{n} p_X(x_i) \\
&= p^k (1-p)^{n-k}
\end{aligned}$$

where k are the number of realizations of the samples equal to 1. Hence the MLE of the parameter p is

$$\begin{aligned}
\text{MLE}[p] &= \arg\max_{p} \log \mathcal{L}_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid p) \\
&= \arg\max_{p} k \log p + (n-k) \log(1-p)
\end{aligned}$$

We compute the derivative (and second derivative) of log-likelihood function

$$\frac{\mathrm{d}\log \mathcal{L}_{X_1,\dots,X_n}(x_1,\dots,x_n \mid p)}{\mathrm{d}p} = \frac{k}{p} - \frac{n-k}{1-p}$$
$$\frac{\mathrm{d}^2\log \mathcal{L}_{X_1,\dots,X_n}(x_1,\dots,x_n \mid p)}{\mathrm{d}p^2} = -\frac{k}{p^2} + \frac{n-k}{(1-p)^2}$$

We have a stationary point at
$$p = \frac{k}{n}.$$
At this point, the second derivative is negative, hence the stationary point is maximum. Hence
$$\mathrm{MLE}\,[p] = \frac{k}{n}.$$

Example 1.1.2. (**MLE for Normal Distribution**) For random sample $\langle X_i \rangle$ of size n s.t. $X_1,\dots,X_n \sim \mathcal{N}(\mu,\sigma^2)$ with parameters $\mu \in \mathbb{R}, \sigma^2 > 0$. Recall that the pdf is

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

Hence the likelihood function is given by

$$\begin{aligned}\mathcal{L}_{X_1,\dots,X_n}(x_1,\dots,x_n \mid \mu,\sigma) &= f_{\mu,\sigma}(x_1,\dots,x_n) \\ &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]\end{aligned}$$

and the log-likelihood function is

$$\mathcal{L}_{X_1,\dots,X_n}(x_1,\dots,x_n \mid \mu,\sigma) = -\frac{n\log(2\pi)}{2} - n\log\sigma - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$$

So the MLE of the parameters μ and σ is

$$\begin{aligned}\mathrm{MLE}\,[(\mu,\sigma)] &= \arg\max_{\mu,\sigma} \log \mathcal{L}_{X_1,\dots,X_n}(x_1,\dots,x_n \mid \mu,\sigma) \\ &= \arg\max_{\mu,\sigma} -n\log\sigma - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}\end{aligned}$$

We compute the partial derivatives of the log-likelihood function,

$$\frac{\partial \log \mathcal{L}_{X_1,\ldots,X_n}(x_1,\ldots,x_n \mid \mu,\sigma)}{\partial \mu} = -\sum_{i=1}^{n} \frac{x_i-\mu}{\sigma^2}$$
$$\frac{\partial \log \mathcal{L}_{X_1,\ldots,X_n}(x_1,\ldots,x_n \mid \mu,\sigma)}{\partial \sigma} = -\frac{n}{\sigma} + \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{\sigma^3}$$

So the stationary points are at

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

To show these points maximize μ, σ, we would use the Hessian (see vector calculus notes). Hence

$$\text{MLE}[\mu] = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$\text{MLE}[\sigma] = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$$

- **Plug-in Principle**. Suppose we have an estimator $\hat{\theta} = \delta_\theta(X_1,\ldots,X_n)$ for the parameter θ and a statistic $t = \delta_t(\theta) = \delta'_t(X_1,\ldots,X_n)$. We have

$$\hat{t} = \delta_t(\hat{\theta}) = \delta_t(\delta_\theta(X_1,\ldots,X_n)).$$

1.1.3 Numerical Optimization with Scipy

- To find the minimum of a function $f : \mathbb{R}^n \to \mathbb{R}$,

```
import scipy.optimize

# f : ℝ^k → ℝ
def f(x):
    # The function to minimize. Input x is a length-k vector.
    return ...

x_0 = [...]
x̂ = scipy.optimize.fmin(f, x_0)
```

- To maximize f, simply minimize $-f$ e.g.

```
x̂ = scipy.optimize.fmin(lambda x: -f(x), x0)
```

- The initial value x_0 must be well-chosen to obtain a local minimum.
- Optimization will only obtain a local minimum (not necessarily a global minimum).
- Optimization is performed over $\mathbb{R}^k$. To optimize over a constrained domain $\mathcal{D}$, a mapping $T : \mathbb{R}^\ell \to \mathcal{D}$ must be used.
  - Minimizing over $\mathcal{D} = x \in \mathbb{R} : x > 0$. Minimize over $y \in \mathbb{R}$ with $T(y) = e^y$.
  - Minimizing over $\mathcal{D} = [0, 1]$. Minimize over $y \in \mathbb{R}$ with $T(y) = \frac{e^y}{1+e^y}$.
  - Minimizing over $\mathcal{D} = \{(x, y, z) \in \mathbb{R}^3 : x + y + z = 1\}$. Minimize over $(x, y) \in \mathbb{R}^2$ with $T(x, y) = (x, y, 1 - x - y)$.
  - Minimizing over $\mathcal{D} = \{(x, y, z) \in [0, 1]^3 : x + y + z = 1\}$. Minimize over $(x, y, z) \in \mathbb{R}^3$ with

$$T(x, y, z) = \left(\frac{e^x}{e^x + e^y + e^z}, \frac{e^y}{e^x + e^y + e^z}, \frac{e^z}{e^x + e^y + e^z} \right).$$

    This is known as the *softmax transformation*

```
def f(x):
    # The function to minimize. Input x ∈ D

def T(y):
    # The function representing the transformation T : R^ℓ → D
    return ...

x0 = [...] # y0 ∈ D
ŷ = scipy.optimize.fmin(lambda y: f(T(y)), T^{-1}(x0))
x̂ = T(ŷ)
```

1.1.4 Standard Distributions in Numpy

Discrete Distributions

Distribution	Notation	Probability Mass Function	Numpy
Bernoulli	$X \sim \text{Bern}(p)$	$p_X(x) = \begin{cases} p & x = 1 \\ 1-p & x = 0 \\ 0 & \text{otherwise} \end{cases}$	x = np.random.binomial(1, p)
Discrete Uniform	$X \sim U(a,b)$	$p_X(x) = \begin{cases} \frac{1}{b+1-a} & x \in \{a, \dots, b\} \\ 0 & \text{otherwise} \end{cases}$	x = np.random.randint(low=a, high=$b+1$)
Binomial	$X \sim B(n,p)$	$p_X(x) = \begin{cases} \frac{1}{b+1-a} & x \in \{a, \dots, b\} \\ 0 & \text{otherwise} \end{cases}$	x = np.random.binomial(n, p)
Geometric	$X \sim \text{Geo}(p)$	$P(X=x) = p_X(x) = p(1-p)^{x-1}$	x = np.random.geometric(p) − 1
Poisson	$X \sim \text{Poisson}(\lambda)$	$P(X=x) = p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}$	x = np.random.poisson(lam=λ)

Continuous Distributions

Distribution	Notation	Probability Density Function	Numpy
Uniform	$X \sim U[a,b]$	$f_X(x) = \begin{cases} \frac{1}{b-1} & x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$	x = np.random.random(low=a, high=b)
Exponential	$X \sim \text{Exp}(\lambda)$	$f_X(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$	x = np.random.exponential(scale=λ)
Normal	$X \sim \mathcal{N}(\mu, \sigma^2)$	$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$	x = np.random.normal(loc=μ, scale=σ)

Example 1.1.3. (**Deriving Likelihood Function from Python**) Consider

```
def rz(p, μ0, μ1, σ0, σ1):
    x = np.random.binomial(1, p)
    y = np.random.normal(loc=μ0, scale=σ0) if x == 0 \
        else np.random.normal(loc=μ1, scale=σ1)
    return np.power(10, y)
```

where $\boldsymbol{\theta} = (p, \mu_1, \mu_2, \sigma_1, \sigma_2) \in [0,1] \times \mathbb{R}^2 \times \mathbb{R}^2_{>0}$. Let X be a discrete random variable on $(\Omega, \mathcal{F}, P)$ s.t $X \sim \text{Bern}(p)$. Let Y be a continuous random variable on $(\Omega, \mathcal{F}, P)$ s.t $Y \sim \mathcal{N}(\mu_X, \sigma_X^2)$. Let us consider the pdf f_Y given $X = x$,

$$f_Y(y \mid X = x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(y-\mu_x)^2}{2\sigma_x^2}\right]$$

Recall that the mixed joint density function is given by

$$f_Y(x, y) = f_Y(y \mid X = x) \cdot P(X = x) = \begin{cases} \frac{1-p}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(y-\mu_0)^2}{2\sigma_0^2}\right] & x = 0 \\ \frac{p}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(y-\mu_1)^2}{2\sigma_1^2}\right] & x = 1 \end{cases},$$

hence

$$f_y(y) = \sum_{x \in \overrightarrow{X}(\Omega)} f_Y(x, y) = \frac{1-p}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(y-\mu_0)^2}{2\sigma_0^2}\right] + \frac{p}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(y-\mu_1)^2}{2\sigma_1^2}\right].$$

So by the definition of the likelihood function,

$$\mathcal{L}_Y(y \mid \boldsymbol{\theta}) = \frac{1-p}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(y-\mu_0)^2}{2\sigma_0^2}\right] + \frac{p}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(y-\mu_1)^2}{2\sigma_1^2}\right].$$

1.1.5 Unsupervised Learning

Definition 1.1.3. (**Parameterised Unsupervised Learning**) Unsupervised learning is the process:

1. Given a dataset $\langle x_i \rangle$ of the sample $\langle X_i \rangle$ of size n,
2. Select a distribution F indexed by parameters $\boldsymbol{\theta}$ that models X_i.
3. Fit the distribution using MLE on $\boldsymbol{\theta}$.

Sometimes referred to as **generative modelling**.

Example 1.1.4. Fit the distribution F with likelihood function

$$\mathcal{L}_Y(y \mid \boldsymbol{\theta}) = \frac{1-p}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(y-\mu_0)^2}{2\sigma_0^2}\right] + \frac{p}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(y-\mu_1)^2}{2\sigma_1^2}\right],$$

indexed by $\boldsymbol{\theta} = (p, \mu_0, \mu_1, \sigma_0, \sigma_1) \in [0,1] \times \mathbb{R}^2 \times \mathbb{R}^2_{>0}$ using the data mp_expenses.

Use numerical optimization by Scipy. Note that we have the constrained domain $[0,1] \times \mathbb{R}^2 \times \mathbb{R}^2_{>0}$, so let us apply the transformations

$$p = \frac{e^q}{1+e^q} \qquad\qquad \sigma_i = e^{\tau_i}$$

giving us the parameter vector $\boldsymbol{\theta'} = (q, \mu_0, \mu_1, \tau_0, \tau_1) \in \mathbb{R}^5$. So we have

```
def f(θ, y):
    p, μ0, μ1, σ0, σ1 = θ
    lik = (1−p) * Φ(y, loc=μ0, scale=σ0) + p * Φ(y, loc=μ1, scale=σ1)
    return np.log(lik)

def T(θ'):
    q, μ0, μ1, τ0, τ1 = θ'
    p = np.exp(q) / (1 + np.exp(q))
    σ0, σ1 = np.exp([τ0, τ1])
    return p, μ0, μ1, σ0, σ1

y = np.log10(mp_expenses)
θ'0 = [...]

# Maximize the sum of likelihoods
θ̂' = scipy.optimize.fmin(lambda θ': −np.sum(f(T(θ'), y)), θ'0)
θ̂ = T(θ̂')
```

where Φ is a vectorized Normal pdf.

1.1.6 Supervised Learning

Definition 1.1.4. (**Parameterised Supervised Learning**) Supervised learning is the process:

1. Given a labelled dataset $\langle(x_i, y_i)\rangle$ of the sample $\langle(X_i, Y_i)\rangle$ of size n, where x_i is the predictor variable and y_i is the label.

2. Select a distribution F_{Y_i} indexed by parameters $\boldsymbol{\theta}$ and X_i that models Y_i.

3. Fit the distribution using MLE on $\boldsymbol{\theta}$.

Example 1.1.5. (**Straight-line Fit**) Given labelled dataset $\langle(x_i, y_i)\rangle$ of the sample $\langle(X_i, Y_i)\rangle$ of size n s.t

$$(Y_i \mid X_i = x_i) \sim a + bx_i + \mathcal{N}(0, \sigma^2) = \mathcal{N}(a + bx_i, \sigma^2),$$

where $\boldsymbol{\theta} = (a, b) \in \mathbb{R}^2$ and σ is known.

Recall that the pdf is

$$f_Y(y \mid x, a, b) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y - a - bx)^2}{2\sigma^2}\right].$$

Hence the likelihood function is given by

$$\begin{aligned}
\mathcal{L}_{Y_1,\dots,Y_n}(y_1, \dots, y_n \mid x_1, \dots, x_n, a, b) &= \prod_{i=1}^{n} f_Y(y_i \mid x_i, a, b) \\
&= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y_i - a - bx_i)^2}{2\sigma^2}\right]
\end{aligned}$$

and the log-likelihood function is

$$\mathcal{L}_{Y_1,\dots,Y_n}(y_1, \dots, y_n \mid x_1, \dots, x_n, a, b) = -\frac{n \log 2\pi}{2} - n \log \sigma - \sum_{i=1}^{n} \frac{(y_i - a - bx_i)^2}{2\sigma^2}$$

So the MLE of a and b is

$$\begin{aligned}
\text{MLE}\left[(a, b)\right] &= \arg\max_{a,b} \mathcal{L}_{Y_1,\dots,Y_n}(y_1, \dots, y_n \mid x_1, \dots, x_n, a, b) \\
&= \arg\max_{a,b} \sum_{i=1}^{n} (y_i - a - bx_i)^2
\end{aligned}$$

Computing the partial derivatives of the log-likelihood function yields:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}_{Y_1,\dots,Y_n}(y_1, \dots, y_n \mid x_1, \dots, x_n, a, b)}{\partial a} &= -2 \sum_{i=1}^{n} (y_i - a - bx_i) \\
\frac{\partial \log \mathcal{L}_{Y_1,\dots,Y_n}(y_1, \dots, y_n \mid x_1, \dots, x_n, a, b)}{\partial b} &= -2 \sum_{i=1}^{n} (y_i - a - bx_i)x_i
\end{aligned}$$

So the stationary points are at:

$$a = \frac{1}{n}\sum_{i=1}^{n} y_i - bx_i = \overline{y} - b\overline{x}$$
$$b = \frac{\sum_i x_i y_i - a x_i}{\sum_i x_i^2}$$

Hence

$$\text{MLE}\left[a\right] = \overline{y} - \text{MLE}\left[b\right]\overline{x}$$
$$\text{MLE}\left[b\right] = \frac{\sum_i x_i y_i - a x_i}{\sum_i x_i^2}$$

Alternatively, using Scipy optimization, we have

```
x, y = [ ... ], [ ... ]
σ = ...

Φ = scipy.stats.norm.pdf

def f(y, x, θ):
    a,b = θ
    lik = Φ(y, loc=a+b * x, scale=σ)
    return np.log(lik)

θ₀ = [ ... ]
â,b̂ = scipy.optimize.fmin(lambda θ: -np.sum(f(y, x, θ)), θ₀)
```

1.2 Linear Regression

1.2.1 Linear Models

- A type of supervised learning.

Definition 1.2.1. (**Linear Model**) Given a labelled data set $\langle(\mathbf{x}_i, y_i)\rangle$ of the sample $\langle(\mathbf{X}_i, Y_i)\rangle$ of size n, Y_i is said to have a linear model if for all y_i,

$$y_i = \beta_0 x_{i0} + \cdots + \beta_m x_{im} + \varepsilon_i,$$

where ε_i is the error term and $\mathbf{x}_i = (x_{i0}, \ldots, x_{im})$, a vector of *features* and $\beta_j \in \mathbb{R}$ is the parameter weighting for the jth feature.

- We often write
$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$
where
$$\mathbf{y} = (y_i) \qquad \mathbf{X} = (\mathbf{x}_i^T) \qquad \boldsymbol{\beta} = (\beta_i) \qquad \boldsymbol{\varepsilon} = (\varepsilon_i)$$

- **Notation and Terminology**:
  - $\mathbf{y}$ is called the *response vector*, $\mathbf{X}$ is a matrix of *regressors* or *features*. This matrix may contain values that are (non-linear) functions of other features. The model is still linear since it's linear in $\boldsymbol{\beta}$.
  - Usually $\mathbf{x}_{i0} = 1$. The corresponding β_0 is the *intercept*.
  - $\boldsymbol{\beta}$ is a $(m+1)$-dimensional *parameter vector*. Sometimes called the *regression coefficients*. These elements β_j can be as the partial derivative
$$\forall i . \beta_j = \frac{\partial y_i}{\partial x_{ij}}.$$
  - $\boldsymbol{\varepsilon}$ is a vector of error, or noise, terms

- **Fitting a linear model**:
  1. Given a dataset $\langle(\mathbf{x}_i, y_i)\rangle$ from a sample $\langle(\mathbf{X}_i, Y_i)\rangle$ where Y_i is modelled by a linear model, that is to say $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.
  2. Estimate $\boldsymbol{\beta}$ such that the error term
$$\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta},$$
is minimized.
  3. Commonly we use mean square error to minimize $\boldsymbol{\beta}$
$$\text{MSE}[\boldsymbol{\varepsilon}] = \frac{1}{n}\sum_i \varepsilon_i^2.$$
That is
$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Theorem 1.2.1. Given a data set $\langle(\mathbf{x}_i, y_i)\rangle$ from the sample of $\langle(\mathbf{X}_i, Y_i)\rangle$ of size n with the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. Then

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Proof. We have

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^2 = \arg\min_{\beta} \underbrace{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} x_{ij}\beta_j\right)^2}_{S}$$

Taking the partial derivative of S with reject to β_j yields:

$$\frac{\partial S}{\partial \beta_j} = 2\sum_{i=1}^{n} \varepsilon_i \frac{\partial \varepsilon_i}{\partial \beta_j} = -2\sum_{i=1}^{n} x_{ij}\left(y_i - \sum_{k=1}^{m} x_{ik}\beta_k\right)$$

Note that S is minimized when $\partial_{\beta_j} S = 0$ for all $0 \leq j \leq m$, since S is convex. Hence

$$\begin{aligned} -2\sum_{i=1}^{n} x_{ij}\left(y_i - \sum_{k=1}^{m} x_{ik}\beta_k\right) &= 0 \\ \iff \sum_{k=1}^{m}\sum_{i=1}^{n}(x_{ij}x_{ik})\beta_k &= \sum_{i=1}^{n} x_{ij}y_i \\ \iff \sum_{k=1}^{m}(\mathbf{X}^T\mathbf{X})_{jk}\beta_k &= (\mathbf{X}^T\mathbf{y})_j \\ \iff (\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} &= \mathbf{X}^T\mathbf{y} \end{aligned}$$

Hence $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. □

Example 1.2.1. Fit the model

$$y_i = \alpha + \beta x_i + \varepsilon_i,$$

using computational methods. We have

```
x, y = [ ... ], [ ... ]
n = len(y)

X = np.column_stack([np.ones(n), x])
model = sklearn.linear_model.LinearRegression(fit_intercept=False)
model.fit(X, y)
```

```
α,β = model.coef_

# Sklearn always includes a one vector (without fit_intercept=False)
model = sklearn.linear_model.LinearRegression()
model.fit(np.column_stack([x]), y)
α,(β,) = model.intercept_, model.coef_
```

1.2.2 Feature Design

One-Hot Encoding

- **One-Hot Encoding**: A feature encoding technique where categorical variables (or *factors*) are converted into binary vectors $\mathbf{x}_i$ s.t $\exists! j. x_{ij} = 1 \wedge (\forall j' \neq j. x_{ij} = 0)$. Where $x_{ij} = 1$ implies that category j is observed. We denote this using the indicator function: $\mathbf{I}_{cat=j}$.

- **Notation**:

  – Suppose we have a labelled dataset $\langle(\mathbf{x}_i, y_i)\rangle$ from a sample $\langle(\mathbf{X}_i, Y_i)\rangle$ of size n, with categories $k_1, \ldots, k_\ell$. Then the linear model is given by
  $$\mathbf{y} = \sum_{1 \leq i \leq \ell} \mathbf{I}_{cat=i} \otimes (\mathbf{X}\boldsymbol{\beta}_i) + \boldsymbol{\varepsilon},$$
  where $\boldsymbol{\beta}_i$ is the parameter vector for category i and $\otimes$ is the tensor product (in this context, elementwise multiplication).

  – Alternatively, we write
  $$\mathbf{X} = \begin{bmatrix} \mathbf{I}_{cat=i} \otimes \mathbf{x}_{i0} & \cdots & \mathbf{I}_{cat=i} \otimes \mathbf{x}_{im} & \cdots \\ \downarrow & & \downarrow & \end{bmatrix}$$
  with $\boldsymbol{\beta}$ s.t
  $$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \downarrow \\ \vdots \\ \boldsymbol{\beta}_\ell \\ \downarrow \end{bmatrix}.$$
  This yields the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.

Example 1.2.2. Fit the model

$$\mathsf{Petal.Length} = \alpha_s + \beta_s \mathsf{Sepal.Length} + \boldsymbol{\varepsilon},$$

where α_s and β_s are species-dependent parameters. We have the species $\mathsf{setosa}, \mathsf{versicolor}, \mathsf{virginica}$.

```
species_classes = ["setosa", "versioclor", "virginica"]
species, SL, PL = ...

def X():
    features = []
    for s in species_classes:
        I_s = np.where(species == s, 1, 0)
        features.extend([I_s, I_s * SL])

    return np.column_stack(features)

# unique intercept for each species
model = sklearn.linear_model.LinearRegression(fit_intercept=False)
model.fit(X(), PL)
```

Non-Linear Response

- **Non-Linear Response**: A linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is said to have a non-linear response iff $\mathbf{x}_i$ is given by $\mathbf{x}_i = f(\mathbf{t})$ where $\mathbf{t}$ is some feature.

Periodic Patterns

- **Periodic Pattern**: A linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is said to have a periodic pattern iff $\mathbf{x}_i$ is a periodic function of $\mathbf{t}$. e.g. $\sin(2\pi\mathbf{t})$

- For example, a model for climate temperature may be:

$$\mathbf{T} = \alpha + \beta \sin(2\pi\mathbf{t} + \phi\mathbf{1}) + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta} = \begin{bmatrix}\alpha & \beta & \phi\end{bmatrix}^T$. Not linear, however, using the composite angle formulae, we have

$$\mathbf{T} = \alpha + \beta_1 \sin(2\pi\mathbf{t}) + \beta_2 \cos(2\pi\mathbf{t}),$$

where $\boldsymbol{\beta}' = \begin{bmatrix}\alpha & \beta_1 & \beta_2\end{bmatrix}^T$ with $\beta_1 = \beta\cos\phi$ and $\beta_2 = \beta\sin\phi$.

Secular Trend

- **Secular Trend**: A linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is said to have a secular trend if $\mathbf{y}$ is a time-series that is monotonically increasing.
- For example, a model for climate temperature (with global warming):

$$\mathbf{T} = \alpha + \beta_1 \sin(2\pi\mathbf{t}) + \beta_2 \cos(2\pi\mathbf{t}) + \gamma\mathbf{t},$$

  where $\gamma\mathbf{t}$ is secular trend.

Diagnosing a Linear Model

- To determine whether a linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ has the correct features, we can plot the error vector $\boldsymbol{\varepsilon}$.
- If $\boldsymbol{\varepsilon}$ varies with some feature $\mathbf{t}$ then we may rewrite our model as:

$$\mathbf{y} = \begin{bmatrix}\mathbf{X} & \mathbf{t}\end{bmatrix} \boldsymbol{\beta}' + \boldsymbol{\varepsilon}.$$

Feature Spaces

Definition 1.2.2. (**Linear Span**) Suppose $S = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subseteq V$ of a space V over K. The span of S is defined as

$$\text{span}(S) = \{(\lambda_k \cdot \mathbf{v}_k) : \lambda_k \in K\}.$$

Definition 1.2.3. (**Linearly Dependent**) Let V be a space over K. The vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in V$ are linearly dependent if

$$\exists \lambda_1, \dots, \lambda_n \in K. \sum_{k=1}^{n} \lambda_k \cdot \mathbf{v}_k = \mathbf{0} \implies \exists \lambda_k . \lambda_k \neq 0.$$

Definition 1.2.4. (**Linearly Independent**) Let V be a space over K, The vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in V$ are linearly independent if

$$\exists \lambda_1, \cdots, \lambda_n \in K. \sum_{k=1}^{n} \lambda_k \cdot \mathbf{v}_k = \mathbf{0} \implies \forall \lambda_k . \lambda_k = 0.$$

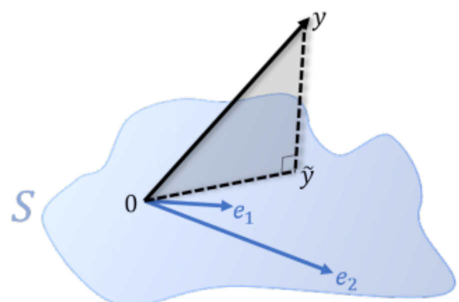
- We may determine whether the vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ are linearly independent using Numpy:

```
k = np.linalg.matrix_rank(np.column_stack([v_1,...,v_n]))
```

If $k = n \implies$ linear independence by definition of matrix rank (dimension of space spanned by columns)

- Given a feature space S spanned by $\{\mathbf{x}_{i0}, \ldots, \mathbf{x}_{im}\}$, the feature vectors, and any vector $\mathbf{x}$ (not necessarily a member of S), there exists a unique vector $\tilde{x} \in S$ s.t

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{y} \in S} \|\mathbf{x} - \mathbf{y}\|^2.$$



$\tilde{\mathbf{x}}$ is the *projection* of $\mathbf{x}$ onto S. Note that $(\mathbf{x} - \tilde{\mathbf{x}}) \cdot \mathbf{y} = 0$ for all $\mathbf{y} \in S$. Since $\tilde{\mathbf{x}} \in S$, there exists β_j s.t

$$\tilde{\mathbf{x}} = \sum_j \beta_j \mathbf{x}_{ij}.$$

Finding β_j is *least squares estimation*. If $\mathbf{x}_{ij}$ are independent, then unique solution.

Method 1.2.1. (**Interpreting Parameters**) To interpret parameters of a linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$:

- Inspect the relation between $\mathbf{y}$ and $\boldsymbol{\beta}$ for certain datapoints. (*Gain some intuition*)

- Ensure the feature vectors $\mathbf{x}_0, \ldots, \mathbf{x}_n$ are linearly independent. If not $\implies$ parameters are *non-identifiable* and features are *confounded*.

1.2.3 Linear Regression

Definition 1.2.5. (**Linear Regression**) Given a labelled sample $\langle(\mathbf{X}_i, Y_i)\rangle$ of size n, Y_i is said to be a linear regression if for all Y_i

$$(Y_i \mid \mathbf{X}_i) \sim \beta_0 X_{i0} + \cdots + \beta_m X_{im} + \mathcal{N}(0, \sigma^2),$$

where $\mathbf{X}_i = (X_{i0}, \ldots, X_{im})$ is the random variable feature vector and $\beta_j, \sigma \in \mathbb{R}$ are parameters. We may write this as

$$(\mathbf{Y} \mid \mathbf{X}) \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 I_n).$$

- **Note**: Not all linear models are linear regressions, since non-trivial assumptions:
  - **Linearity**: $\mathbb{E}\left[Y_i \mid \mathbf{x}_i\right] = \sum_j \beta_k x_{ij}$
  - **Normality**: $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$
  - **Independence of errors**: $\varepsilon_1, \ldots, \varepsilon_n$ are independent / uncorrelated $\mathrm{Cov}\left[\varepsilon_i, \varepsilon_j\right] = \mathbb{E}\left[\varepsilon_i \varepsilon_j\right] = 0$ for all $i \neq j$.
- Given the data set $\langle(\mathbf{x}_i, y_i)\rangle$, we may fit the regression using least squares estimation on the model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Theorem 1.2.2. The maximum likelihood estimator of $\boldsymbol{\beta}$ is the least squares estimate:

$$\mathrm{MLE}\left[\boldsymbol{\beta}\right] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y}.$$

Proof. Recall that the pdf f_Y is given by

$$f_Y(y \mid \mathbf{x}, \boldsymbol{\beta}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y - \sum_{j=1}^m x_j \beta_j)}{2\sigma^2}\right].$$

Hence the log-likelihood function is

$$\begin{aligned}\log \mathcal{L}_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n \mid \mathbf{X}, \boldsymbol{\beta}, \sigma) &= \sum_{i=1}^n \log f_{Y_i}(y_i \mid \mathbf{x}_i, \boldsymbol{\beta}, \sigma) \\ &= -\frac{n \log 2\pi}{2} - n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^n \left(y_i - \sum_{j=1}^m x_{ij}\beta_j\right)^2\end{aligned}$$

So the MLE of $\boldsymbol{\beta}$ is

$$\begin{aligned}
\text{MLE}\left[\boldsymbol{\beta}\right] &= \arg\max_{\boldsymbol{\beta}} \log \mathcal{L}_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n \mid \mathbf{X}, \boldsymbol{\beta}, \sigma) \\
&= \arg\max_{\boldsymbol{\beta}} -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} x_{ij}\beta_j\right)^2 \\
&= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} x_{ij}\beta_j\right)^2
\end{aligned}$$

By Theorem ??,

$$\text{MLE}\left[\boldsymbol{\beta}\right] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y}.$$

So we are done. □

- Note that

$$\begin{aligned}
\frac{\partial \log \mathcal{L}_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n \mid \mathbf{X}, \boldsymbol{\beta}, \sigma)}{\partial \sigma} &= -\frac{n}{\sigma} + \frac{1}{\sigma^3}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right) = 0 \\
\iff \sigma^2 &= \frac{1}{n}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)
\end{aligned}$$

So $\text{MLE}\left[\sigma\right] = \sqrt{\frac{1}{n}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)}$.

1.3 Computational Methods

1.3.1 Monte Carlo Integration

- Computational method for integrating multidimensional definite integral

$$I = \int_{\mathcal{D}} f(\mathbf{x})\,\mathrm{d}\mathbf{x}\,.$$

- Recall for a continuous random variable X on $(\Omega, \mathcal{F}, P)$, for all $g : \overrightarrow{X}(\Omega) \to \mathbb{R}$,

$$\mathbb{E}\left[g(X)\right] = \int_{\overrightarrow{X}(\Omega)} g(x) f_X(x)\,\mathrm{d}x\,.$$

So define $f(\mathbf{x}) = g(\mathbf{x}) f_X(\mathbf{x})$, then $I = \mathbb{E}\left[g(\mathbf{X})\right]$

- Construct a sample $\langle \mathbf{X}_i \rangle$ of size N, distributed over $\mathcal{D}$ with pdf f_X.
- The Monte Carlo estimator of I, denoted $\left\langle I^N \right\rangle$, is

$$\left\langle I^N \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \underbrace{\frac{f(\mathbf{X}_i)}{f_X(\mathbf{X_i})}}_{g(\mathbf{X}_i)} = \overline{Y_N},$$

where $Y_i = g(\mathbf{X}_i) = f(\mathbf{X}_i)/f_X(\mathbf{X}_i)$.

So by Strong Law of Large Numbers:

$$\lim_{N\to\infty} P\left(\left\langle I^N \right\rangle = I\right) = P(\overline{Y_N} - \mathbb{E}[Y]) = 1.$$

- Expectation:

$$\begin{aligned}
\mathbb{E}\left[\left\langle I^N \right\rangle\right] &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)}{f_X(\mathbf{X}_i}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\frac{f(\mathbf{X}_i)}{f_X(\mathbf{X}_i)}\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\int_{\mathbf{x}\in\overrightarrow{\mathbf{X}}(\Omega)} f(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int_{\mathcal{D}} f(\mathbf{x})\,\mathrm{d}\mathbf{x}
\end{aligned}$$

So unbiased estimator.

- Variance:

$$\begin{aligned}
\operatorname{Var}\left[\left\langle I^N \right\rangle\right] &= \operatorname{Var}\left[\frac{V}{N}\sum_{i=1}^{N} f(\mathbf{X}_i)\right] \\
&= \frac{1}{N^2}\operatorname{Var}\left[\sum_{i=1}^{N}\frac{f(\mathbf{X}_i)}{f_X(\mathbf{X}_i)}\right] && \text{Non-linearity of variance} \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\operatorname{Var}\left[Y_i\right] && \text{Variance of independent variables} \\
&= \frac{1}{N}\operatorname{Var}\left[Y\right] && \text{Identical distributions}
\end{aligned}$$

Uniform Monte Carlo

- Construct a sample $\langle \mathbf{X}_i \rangle$ of size N, distributed uniformly in $\mathcal{D}$. So

$$f_X(\mathbf{x}) = \begin{cases} \frac{1}{V} & \text{if } \mathbf{x} \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases}$$

  where V is the volume over $\mathcal{D}$: $V = \int_{\mathcal{D}} \mathrm{d}\mathbf{x}$

- The Monte Carlo estimator of I is

$$\left\langle I^N \right\rangle = \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{X}_i) = V\overline{Y_N},$$

  where $Y_i = f(\mathbf{X}_i)$.

- **Problem**: Thus standard deviation of $\left\langle I^N \right\rangle$ converges with rate $O(\sqrt{N})$.
- **Solutions**:
  - Use *stratified sampling*. Split the domain $\mathcal{D}$ to N subdomains $\mathcal{D}_1, \ldots, \mathcal{D}_N$. This converges with $O(N)$.
  - Use *importance sampling*.

Example 1.3.1. Compute the integral $I = \int_a^b f(x)\,\mathrm{d}x$ using computational methods.

```
N = ...
x = np.random.uniform(a, b, size=N)
⟨I^N⟩ = (b - a) * np.mean(f(x))
```

Importance Sampling

- Construct a sample $\left\langle \tilde{\mathbf{X}}_i \right\rangle$ of size N, distributed over $\mathcal{D}$ with pdf $f_{\tilde{X}}$. Define h such that

$$h(\mathbf{x}) = g(\mathbf{x}) \frac{f_X(\mathbf{x})}{f_{\tilde{X}}(\mathbf{x})}.$$

Then

$$\mathbb{E}\left[h(\tilde{\mathbf{X}})\right] = \mathbb{E}\left[g(\tilde{\mathbf{X}})\frac{f_X(\tilde{\mathbf{X}})}{f_{\tilde{X}}(\tilde{\mathbf{X}})}\right]$$
$$= \int_{\mathbf{x}\in\vec{\mathbf{X}}(\Omega)} g(\mathbf{x})\frac{f_X(\mathbf{x})}{f_{\tilde{X}}(\mathbf{x})} f_{\tilde{X}}(\mathbf{x})\,\mathrm{d}\mathbf{x}$$
$$= \int_{\mathcal{D}} g(\mathbf{x}) f_X(\mathbf{x})\,\mathrm{d}\mathbf{x} = I$$

- Hence the importance sampling Monte Carlo estimator is

$$\langle I^N \rangle = \frac{1}{N}\sum_{i=1}^{N} \underbrace{g(\tilde{\mathbf{X}}_i)\frac{f_X(\tilde{\mathbf{X}}_i)}{f_{\tilde{X}}(\tilde{\mathbf{X}}_i)}}_{Y_i} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\tilde{\mathbf{X}}_i)}{f_{\tilde{X}}(\tilde{\mathbf{X}}_i)}.$$

- Select $f_{\tilde{X}}$ s.t $\mathrm{Var}\left[Y\right]$ is minimized. This occurs if $Y_i = \kappa$ where κ is some constant (since variance of constant is zero). Hence

$$f_{\tilde{X}}(\tilde{x}) = \frac{1}{\kappa} g(\tilde{x}) f_X(\tilde{x})$$

Note that $\kappa = \int_{x\in\vec{X}(\Omega)} g(x) f_X(x)\,\mathrm{d}x = I$. So we approximate s.t $f_{\tilde{X}}(\tilde{x}) \propto g(\tilde{x}) f_X(\tilde{x}) = f(\tilde{x})$.

1.3.2 Estimating Probabilities

Computational Probability

- Consider estimating $P(X \in \mathcal{I})$ and we have a function to sample: `rx()`. Note that

$$\mathbb{E}\left[I_{X\in\mathcal{I}}\right] = P(X \in \mathcal{I}) = \int_{\vec{X}(\Omega)} I_{x\in\mathcal{I}} f_X(x)\,\mathrm{d}x\,.$$

- Using a Monte Carlo estimator, we have

$$\left\langle P(X \in \mathcal{I})^N \right\rangle = \frac{1}{N}\sum_{i=1}^{N} I_{X_i\in\mathcal{I}}.$$

```
N, I = ...
x = [rx() for _ in range(N)]
p = np.mean(np.where(x in I, 1, 0))
```

Computational Bayes

- Recall that Bayes' Law states:

$$\mathcal{L}_{X|Y}(x \mid Y = y) = \frac{\mathcal{L}_X(x)\mathcal{L}_{Y|X}(y \mid X = x)}{\mathcal{L}_Y(y)}.$$

- Suppose we have a function to sample: `rx()` and $\mathcal{L}_{Y|X}$:
  1. Generate a dataset $\langle x_i \rangle$ of size N (using `rx()`)
  2. Compute the weights (w_i):

     $$w_i = \frac{\mathcal{L}_{Y|X}(y \mid X = x_i)}{\mathcal{L}_Y(y)},$$

     where $\mathcal{L}_Y(y) = \sum_i \mathcal{L}_{Y|X}(y \mid X = x_i)$
  3. Compute

     $$\left\langle \mathbb{E}\left[h(X) \mid Y = y\right]^N \right\rangle = \frac{1}{N}\sum_{i=1}^{N} w_i h(x_i),$$

     using a Monte Carlo estimator.

Proof. We have

$$\begin{aligned}
\mathbb{E}\left[h(X) \mid Y = y\right] &= \int_{x \in \overrightarrow{X}(\Omega)} h(x) f_{X|Y}(x \mid Y = y)\,\mathrm{d}x \\
&= \int_{x \in \overrightarrow{X}(\Omega)} \underbrace{h(x)\kappa f_{Y|X}(y \mid X = x)}_{g(x)} f_X(x)\,\mathrm{d}x \\
&= \int_{x \in \overrightarrow{X}(\Omega)} g(x) f_X(x)\,\mathrm{d}x \\
\left\langle \mathbb{E}\left[h(X) \mid Y = y\right]^N \right\rangle &= \frac{1}{N}\sum_{i=1}^{N} g(x_i) && \text{Monte Carlo Estimator}
\end{aligned}$$

The constant κ is determined using

$$\begin{aligned}\kappa &= 1\Big/\int_{x\in\overrightarrow{X}(\Omega)} \underbrace{f_{Y|X}(y \mid X=x)}_{g(x)} f_X(x)\,\mathrm{d}x \\ &= 1/\mathbb{E}\left[g(X)\right] \\ \left\langle \kappa^N \right\rangle &= 1\Big/\frac{1}{N}\sum_{i=1}^{N} f_{Y|X}(y \mid X=x_i)\end{aligned}$$

So

$$\left\langle \mathbb{E}\left[h(X) \mid Y=y\right]^N \right\rangle = \frac{1}{N}\sum_{i=1}^{N} \kappa h(x_i)\mathcal{L}_{Y|X}(y \mid X=x) = \sum_{i=1}^{n} \frac{h(x_i)\mathcal{L}_{Y|X}(y \mid X=x)}{\sum_j \mathcal{L}_{Y|X}(y \mid X=x_j)}.$$

□

- Note that for estimating $P(X \in \mathcal{I} \mid Y=y)$, use $h(X) = I_{X\in\mathcal{I}}$.
- Useful to plot the distribution $(X \mid Y=y)$, using histogram.

$$\text{bar height} = P(X \in \mathcal{I} \mid Y=y) = \sum_{i=1}^{N} w_i I_{x_i in \mathcal{I}} = \sum_{x_i in \mathcal{I}} w_i.$$

In Python:

```
plt.hist(x, weights=w, density=True)
```

`density=True` ensures area $= 1$.

Example 1.3.2. Given $\Theta \sim U[0,1]$ and $(Y \mid \Theta=\theta) \sim B(n,\theta)$. Compute $(\Theta \mid Y=y)$.

```
θ_sample = np.random.uniform(0, 1, size=N)

w = θ_sample ** y * (1 - θ_sample) ** (n - y)
w /= np.sum(w) # rescale weights

# plot histogram density.
plt.hist(θ_sample, weights=w, density=True, bins=100)
```

1.4 Empirical Methods

- Empirical methods are based on datasets (observed).

1.4.1 The Empirical Distribution

Definition 1.4.1. (**Empirical Distribution**) Let $\langle X_i \rangle$ be a random sample of size n on $(\Omega, \mathcal{F}, P)$ with cdf F. The empirical distribution function is defined as

$$F_n^*(x) = \frac{n(X_i \leq x)}{n} = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \leq x}.$$

- Plotted:

```
x = [ ... ] # dataset
y = np.arange(1, len(x) + 1) / len(x) # 1/n, ...
plt.plot(np.sort(x), y, drawstyle="steps")
```

- By strong law of large numbers:

$$\lim_{n \to 0} F_n^*(x) = F(x).$$

Definition 1.4.2. (**Ordered Sample**) Let $\langle X_i \rangle$ be a random sample of size n. Let $\pi : [1, n] \to [1, n]$ be a permutation s.t $X_{\pi(i)} < X_{\pi(j)}$ for all $i < j$. We define the ordered sample to be $\langle X_{(i)} \rangle$ s.t $X_{(i)} = X_{\pi(i)}$

- Ordered samples reorder the sample s.t $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$. Useful for defining X^*.

Definition 1.4.3. (**Empirical Random Variable**) Let be a random ordered sample $\langle X_{(i)} \rangle$ of size n on $(\Omega, \mathcal{F}, P)$ with cdf F. The empirical random variable X^* is random variable defined as $X^* = X_{(K)}$ where $K \sim U[1, n]$. Hence the pmf is

$$P(X^* = x) = p_{X^*}(x) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i = x}.$$

- The cdf of X^* is F_n^*.
- X^* models taking values from the dataset $\langle x_i \rangle$ w/ uniform probability.
- The function `rx_star()` is given by

```
x = [ ... ] # dataset of size n
def rx_star():
    return np.random.choice(x, size=n) # generate dataset (size n)
```

- Expectation:

$$\begin{aligned}\mathbb{E}\left[X^*\right] &= \mathbb{E}\left[X_{(K)}\right] \\ &= \sum_{k=1}^{n} X_{(k)} P(K = k) \\ &= \frac{1}{n}\sum_{k=1}^{n} X_{(k)}\end{aligned}$$

- Variance:

$$\begin{aligned}\mathrm{Var}\left[X^*\right] &= \mathbb{E}\left[\left(X_{(K)} - \mathbb{E}\left[X^*\right]\right)^2\right] \\ &= \sum_{k=1}^{n} \left(X_{(k)} - \mu\right)^2 P(K = k) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left(X_{(k)} - \mu\right)^2\end{aligned}$$

- Suppose we have a random sample $\langle X_i \rangle$ of size N. Let X^* be the empirical random variable of $\langle X_i \rangle$.

  The Monte Carlo estimator of $I = \int f(x) f_X(x)\,\mathrm{d}x$ using $\langle X_i \rangle$ is given by

$$\left\langle I^N \right\rangle = \frac{1}{N}\sum_{i=1}^{N} f(X_i) = \mathbb{E}\left[f(X^*)\right].$$

- Empirical distribution used:
  - No knowledge about the sample (dataset) apart from the realizations $\implies$ cannot make any assumptions
  - No storage constraints.

1.4.2 KL Divergence

- In generative modelling, we have a dataset $\langle x_i \rangle$ from a random sample $\langle X_i \rangle$ of size n. We fit a parametric model $\mathcal{M}$ for X with likelihood $\mathcal{L}_X(x \mid \boldsymbol{\theta})$.

- **Problem**: Is $\mathcal{M}$ a "good fit" for the dataset. The KL diverge is a metric to measure whether $\mathcal{M}$ is a "good fit".

Definition 1.4.4. (**Kullback-Leibler Divergence**) Let $\mathcal{L}_{X^*}(\cdot)$ be the likelihood of the empirical distribution X^*, and $\mathcal{L}_X(\cdot \mid \boldsymbol{\theta})$ be the likelihood of the proposed model $\mathcal{M}$ with parameters $\boldsymbol{\theta}$. The KL divergence is

$$\mathrm{KL}\left(\mathcal{L}_{X^*}(\cdot) \,||\, \mathcal{L}_X(\cdot \mid \boldsymbol{\theta})\right) = \sum_{x \in \overrightarrow{X^*}(\Omega)} \mathcal{L}_{X^*}(x) \log \frac{\mathcal{L}_{X^*}(x)}{\mathcal{L}_X(x \mid \boldsymbol{\theta})}.$$

*This can be applied to **any** distribution, for continuous X^*, replace sum with integral*

- *Intuition*:
  - The log-likelihood of the dataset $\langle x_i \rangle$ is:

    $$\log \mathcal{L}_{(X_i)}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) = \sum_{i=1}^{n} \log \mathcal{L}_X(x_i \mid \boldsymbol{\theta}) = \sum_{x \in \overrightarrow{X^*}(\Omega)} N_x \log \mathcal{L}_X(x \mid \boldsymbol{\theta}),$$

    where $N_x = \#$ of occurrences of $x = n\mathcal{L}_{X^*}(x)$.
  - Above metric depends on n. Metric should be independent of dataset size:

    $$\frac{1}{n} \log \mathcal{L}_{(X_i)}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) = \sum_{x \in \overrightarrow{X^*}(\Omega)} \mathcal{L}_{X^*}(x) \log \mathcal{L}_X(x \mid \boldsymbol{\theta}).$$

  - Metric should have a reference point. Define reference point as best-possible fit to $\langle x_i \rangle$ (the empirical distribution). So

    $$\begin{aligned}\mathrm{KL}\left(\mathcal{L}_{X^*}(\cdot) \,||\, \mathcal{L}_X(\cdot \mid \boldsymbol{\theta})\right) &= \frac{1}{n} \log \mathcal{L}_{(X^*)}(x_1, \ldots, x_n) - \log \mathcal{L}_{(X_i)}(x_1, \ldots, x_n \mid \boldsymbol{\theta}) \\ &= \sum_{x \in \overrightarrow{X^*}(\Omega)} \mathcal{L}_{X^*}(x) \log \frac{\mathcal{L}_{X^*}(x)}{\mathcal{L}_{(X_i)}(x \mid \boldsymbol{\theta})}\end{aligned}$$

- **Properties**:

- $\overrightarrow{\mathrm{KL}}(P \,||\, Q) = [0, \infty)$. Recall that Jensen's Inequality states that for X on $(\Omega, \mathcal{F}, P)$. If g is convex, then

  $$E\left[g(X)\right] \geq g\left(\mathbb{E}\left[X\right]\right).$$

  Instantiating for a discrete random variable X with pmf p_X:

  $$\sum_{x \in \overrightarrow{X}(\Omega)} g(x) p_X(x) \geq g\left(\sum_{x \in \overrightarrow{X}(\Omega)} x p_X(x)\right).$$

  Since $-\log$ is concave, it follows that

  $$\begin{aligned} \mathrm{KL}\left(P \,||\, Q\right) &= -\sum_{x \in \overrightarrow{X}(\Omega)} P(x) \log \frac{Q(x)}{P(x)} \\ &\geq -\log\left(\sum_{x \in \overrightarrow{X}(\Omega)} P(x)\right) \\ &= -\log 1 = 0 \end{aligned}$$

- If $\mathrm{KL}(P \,||\, Q) = 0$, it follows that $P(x) = Q(x)$.
- If $\mathrm{KL}\left(P \,||\, Q\right) = \infty$, exists $x \in \overrightarrow{X}(\Omega)$ (in our dataset) s.t the proposed model $Q(x) = 0$ (proposed model says x shouldn't be in dataset) $\implies Q$ is a bad fit.

Example 1.4.1. (**KL Divergence of Gaussians**) Consider the continuous random variables X, Y on $(\Omega, \mathcal{F}, P)$ with $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$. We have

$$\mathcal{L}_X(x \mid \mu_X, \sigma_X^2) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left[-\frac{(x - \mu_X)^2}{2\sigma_X^2}\right]$$
$$\mathcal{L}_Y(y \mid \mu_Y, \sigma_Y^2) = \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left[-\frac{(y - \mu_Y)^2}{2\sigma_Y^2}\right]$$

By definition ??, we have

$$\begin{aligned} KL(\mathcal{L}_X(\cdot \mid \mu_X, \sigma_X^2) || \mathcal{L}_Y(\cdot \mid \mu_Y, \sigma_Y^2)) = &\int_{x \in \overrightarrow{X}(\Omega)} \mathcal{L}_X(x \mid \mu_X, \sigma_X^2) \log \mathcal{L}_X(x \mid \mu_X, \sigma_X^2) \,\mathrm{d}x \\ &- \int_{x \in \overrightarrow{X}(\Omega)} \mathcal{L}_X(x \mid \mu_X, \sigma_X^2) \log \mathcal{L}_Y(x \mid \mu_Y, \sigma_Y^2) \,\mathrm{d}x \end{aligned}$$

Let us first consider

$$
\begin{aligned}
&\int_{x\in\overrightarrow{X}(\Omega)} \mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\log \mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\,\mathrm{d}x \\
&\qquad = \int_{x\in\overrightarrow{X}(\Omega)} \frac{1}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\log\left(\frac{1}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\right)\mathrm{d}x \\
&\qquad = \frac{1}{\sqrt{2\pi}\sigma_X}\int_{x\in\overrightarrow{X}(\Omega)}\left(-\log\sqrt{2\pi}\sigma_X - \frac{(x-\mu_X)^2}{2\sigma_X^2}\right)\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x \\
&\qquad = -\log\sqrt{2\pi}\sigma_X \underbrace{\int_{x\in\overrightarrow{X}(\Omega)}\frac{1}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x}_{1} \\
&\qquad\qquad - \frac{1}{2\sigma_X^2}\underbrace{\int_{x\in\overrightarrow{X}(\Omega)}\frac{(x-\mu_X)^2}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x}_{\sigma_X^2} \\
&\qquad = -\frac{1}{2}\left(1+\log 2\pi\sigma_X^2\right)
\end{aligned}
$$

Similarly, we have

$$
\begin{aligned}
&\int_{x\in\overrightarrow{X}(\Omega)} \mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\log \mathcal{L}_Y(x \mid \mu_Y, \sigma_Y^2)\,\mathrm{d}x \\
&\qquad = \frac{1}{\sqrt{2\pi}\sigma_X}\int_{x\in\overrightarrow{X}(\Omega)}\left(-\log\sqrt{2\pi}\sigma_Y - \frac{(x-\mu_Y)^2}{2\sigma_Y^2}\right)\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x \\
&\qquad = -\frac{1}{2}\log 2\pi\sigma_Y^2 \underbrace{\int_{x\in\overrightarrow{X}(\Omega)}\frac{1}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x}_{1} \\
&\qquad\qquad - \frac{1}{2\sigma_Y^2}\int_{x\in\overrightarrow{X}(\Omega)}\frac{(x-\mu_Y)^2}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x
\end{aligned}
$$

Note that

$$
\begin{aligned}
&\int_{x\in\overrightarrow{X}(\Omega)}\frac{(x-\mu_Y)^2}{\sqrt{2\pi}\sigma_X}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right]\mathrm{d}x \\
&= \int_{x\in\overrightarrow{X}(\Omega)} x^2\mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\,\mathrm{d}x - 2\mu_Y\int_{x\in\overrightarrow{X}(\Omega)} x\mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\,\mathrm{d}x + \mu_Y^2\int_{x\in\overrightarrow{X}(\Omega)}\mathcal{L}_X(x \mid \mu_X, \sigma_X^2)\,\mathrm{d}x \\
&= \mathbb{E}\left[X^2\right] - 2\mu_Y\mu_X + \mu_Y^2
\end{aligned}
$$

Recall that $\sigma_X^2 = \mathbb{E}\left[X^2\right] - \underbrace{\left(\mathbb{E}\left[X\right]\right)^2}_{\mu_X^2}$. So we have

$$\begin{aligned}\int_{x \in \overrightarrow{X}(\Omega)} \frac{(x-\mu_Y)^2}{\sqrt{2\pi}\sigma_X} \exp\left[-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right] \mathrm{d}x &= \sigma_X^2 + \mu_X^2 - 2\mu_Y\mu_X + \mu_Y^2 \\ &= \sigma_X^2 + (\mu_X - \mu_Y)^2\end{aligned}$$

Hence

$$\int_{x \in \overrightarrow{X}(\Omega)} \mathcal{L}_X(x \mid \mu_X, \sigma_X^2) \log \mathcal{L}_Y(x \mid \mu_Y, \sigma_Y^2)\, \mathrm{d}x = -\frac{1}{2}\log 2\pi\sigma_Y^2 - \frac{\sigma_X^2 + (\mu_X - \mu_Y)^2}{2\sigma_Y^2}$$

So the KL divergence is given by

$$\begin{aligned}KL(\mathcal{L}_X(\cdot \mid \mu_X, \sigma_X^2) || \mathcal{L}_Y(\cdot \mid \mu_Y, \sigma_Y^2)) &= -\frac{1}{2}\left(1 + \log 2\pi\sigma_X^2\right) + \frac{1}{2}\log 2\pi\sigma_Y^2 + \frac{\sigma_X^2 + (\mu_X - \mu_Y)^2}{2\sigma_Y^2} \\ &= \log\frac{\sigma_Y}{\sigma_X} + \frac{\sigma_X^2 + (\mu_X - \mu_Y)^2}{2\sigma_Y^2} - \frac{1}{2}\end{aligned}$$

2 Inference

- *Inference* is the process of deducing properties of an unknown probably model using datasets.

2.1 Bayesianism

- **Philosophy**:

probability is the right way to describe uncertainity.

Definition 2.1.1. (**Bayesian Inference**) For some hypothesis H, dependent on the random sample $\langle X_i \rangle$ of size n, referred to as *evidence*, then

$$\mathcal{L}_{H|(X_i)}(H \mid X_1, \ldots, X_n) = \frac{\mathcal{L}_{(X_i)|H}(X_1, \ldots, X_n \mid H)\mathcal{L}_H(H)}{\mathcal{L}_{(X_i)}(X_1, \ldots, X_n)},$$

where

- $\mathcal{L}_H(H)$ is the *prior likelihood*.
- $\mathcal{L}_{H|(X_i)}(H \mid X_1, \ldots, X_n)$ is the posterior likelihood. The likelihood of the hypothesis given the evidence.
- $\mathcal{L}_{(X_i)|H}(X_1, \ldots, X_n \mid H)$ is the likelihood of the evidence, given the hypothesis H.
- $\mathcal{L}_{(X_i)}(X_1, \ldots, X_n)$ is the *marginal likelihood*.

Bayesian Inference is the process of computing $\mathcal{L}_{H|(X_i)}$, the posterior likelihood.

- Often denote $\{\Theta = \theta\}$ as H where Θ is a random variable.
- **Problem**: The prior distribution isn't specified by Bayesianism, therefore varies from one statistician to another.

2.1.1 Finding the Posterior

- Two methods for finding the posterior distribution, $(\Theta \mid \mathbf{X} = \mathbf{x})$:
  - **Computational**:
    * Method:
      1. Generate a dataset $\langle \theta_i \rangle$ of size N using the prior distribution of Θ.
      2. Compute the weights (w_i) given the dataset $\langle x_i \rangle$ of size m:
      $$w_i = \frac{\mathcal{L}_{(X_i)|\Theta}(\mathbf{x} \mid \Theta = \theta_i)}{\sum_j \mathcal{L}_{(X_i)|\Theta}(\mathbf{x} \mid \Theta = \theta_j)}.$$
      3. Approximate $(\Theta \mid \mathbf{X} = \mathbf{x})$ using
      $$\left\langle P(\Theta \in \mathcal{I} \mid \mathbf{X} = \mathbf{x})^N \right\rangle = \frac{1}{N} \sum_{i=1}^{N} w_i I_{\theta_i \in \mathcal{I}}.$$

      See section ??
    * **Problem**: For large m, $\mathcal{L}_{(X_i)|\Theta}(x_1, \dots, x_m \mid \Theta = \theta_i)$ will be infinitesimally small $\implies$ underflow.
    * **Solution**: Use log likelihood function $\log \mathcal{L}_{(X_i)|\Theta}$:
    $$\log \mathcal{L}_{(X_i)|\Theta}\left(x_1, \dots, x_m \mid \Theta = \theta\right) = \sum_{i=1}^{m} \log \mathcal{L}_{X_i|\Theta}(x_i \mid \Theta = \theta).$$
  - **Mathematical**:
    * Method:
      1. Let Θ be a random variables on $(\Omega, \mathcal{F}, P)$ and $\langle X_i \rangle$ be a random sample on $(\Omega, \mathcal{F}, P)$ of size m. Determine the prior likelihood $\mathcal{L}_\Theta$.
      2. Determine the posterior likelihood using Bayes' Theorem:
      $$\mathcal{L}_{\Theta|(X_i)}(\theta \mid \mathbf{X} = \mathbf{x}) = \kappa \mathcal{L}_{(X_i)|\Theta}(\mathbf{x} \mid \Theta = \theta) \mathcal{L}_\Theta(\theta),$$

      where (for a continuous random variable Θ)
      $$\kappa = \frac{1}{\mathcal{L}_{(X_i)}(\mathbf{x})} = 1 \Big/ \int_{\theta \in \overrightarrow{\Theta}(\Omega)} \mathcal{L}_{(X_i)|\Theta}(\mathbf{x} \mid \Theta = \theta) \mathcal{L}_\Theta(\theta) \, \mathrm{d}\theta \,.$$

3. Reason about $(\Theta \mid \mathbf{X} = \mathbf{x})$ using $\mathcal{L}_{\Theta|(X_i)}(\theta \mid \mathbf{X} = \mathbf{x})$.

* **Problem**: Mathematical Bayes' can result in intractable integrals.
* **Solution**: Use Computational Bayes'.

Example 2.1.1. For $\Theta \sim U[0,1]$. For the sample $\langle X_i \rangle$ of size n s.t $(X_i \mid \Theta = \theta) \sim \text{Bernoulli}(\theta)$. Determine $(\Theta \mid \mathbf{X})$.

We have

$$\mathcal{L}_{\Theta}(\theta) = 1 \qquad \mathcal{L}_{X_i|\Theta}(x \mid \theta) = \theta$$

Hence by Bayes' Theorem

$$\begin{aligned}\mathcal{L}_{\Theta|(X_i)}(\theta \mid \mathbf{x}) &= \kappa \mathcal{L}_{(X_i)|\Theta}(\mathbf{x} \mid \theta)\mathcal{L}_{\Theta}(\theta) \\ &= \kappa \theta^S (1-\theta)^{n-S}\end{aligned}$$

where $S = \sum_{i=1}^n I_{X_i=1}$. By inspection, we have

$$(\Theta \mid \mathbf{X}) \sim \text{Beta}(S+1, n-S+1),$$

with

$$\kappa = \frac{\Gamma(n+2)}{\Gamma(S+1)\Gamma(n-S+1)}.$$

Example 2.1.2. For $\boldsymbol{\Theta} = (\Theta_1, \Theta_2) \sim (\text{Exp}(\lambda_1), \text{Exp}(\lambda_2))$ with the sample $\langle X_i \rangle$ of size n s.t $(X_i \mid \boldsymbol{\Theta} = (\theta_1, \theta_2)) \sim U[\theta_1, \theta_1 + \theta_2]$. Determine $(\boldsymbol{\Theta} \mid \mathbf{X})$ (both computationally and mathemeatically).

We have

$$\mathcal{L}_{\Theta_i}(\theta_i) = \lambda_i e^{-\lambda_i \theta_i}$$

So the joint likelihood is given by

$$\mathcal{L}_{\boldsymbol{\Theta}}(\theta = (\theta_1, \theta_2)) = \prod_{i=1}^{2} L_{\Theta_i}(\theta_i) = \left(\prod_{i=1}^{2} \lambda_i\right) \exp\left[-\sum_i \lambda_i \theta_i\right].$$

We note that

$$\mathcal{L}_{X|\boldsymbol{\Theta}}(x \mid \theta_1, \theta_2) = \frac{1}{b} I_{\theta_1 \le x \le \theta_1 + \theta_2}.$$

Hence

$$\mathcal{L}_{(X_i)|\mathbf{\Theta}}(\mathbf{x} \mid \theta_1, \theta_2) = \prod_{i=1}^{n} I_{\theta_1 \le x_i \le \theta_1 + \theta_2}$$
$$= \frac{1}{\theta_2^n} I_{\theta_1 \le \min_i x_i} I_{\max_i x_i \le \theta_1 + \theta_2}$$

So by Bayes' Theorem, we have

$$\mathcal{L}_{\mathbf{\Theta}|(X_i)}(\theta_1, \theta_2 \mid \mathbf{x}) = \kappa \mathcal{L}_{(X_i)|\mathbf{\Theta}}(\mathbf{x} \mid \theta_1, \theta_2) \mathcal{L}_{\mathbf{\Theta}}(\theta_1, \theta_2)$$
$$= \frac{\kappa}{\theta_2^n} I_{\theta_1 \le \min_i x_i} I_{\max_i x_i \le \theta_1 + \theta_2} \lambda_1 \lambda_2 \exp\left[-\lambda_1 \theta_1 - \lambda_2 \theta_2\right]$$

Computationally, we have

```
N, λ = ..., [...]
θ_sample = np.array([np.random.exponential(scale=1/λ[i], size=N) for i in range(len(λ))])

x = [ ... ]
n = len(x)

max_x, min_x = max(x), min(x)
w = 1 / θ_sample[1] ** n \
    * np.where(θ_sample[0] <= min_x & max_x <= np.sum(θ_sample), 1, 0)
w /= np.sum(w)
```

Example 2.1.3. (**Model Comparison**) Consider K parameteric models $\mathcal{M}_1, \dots, \mathcal{M}_K$. Let the discrete random variable M model the correct (or optimal model) s.t $\overrightarrow{M}(\Omega) = \{\mathcal{M}_1, \dots, \mathcal{M}_K\}$. Let $\mathbf{\Theta}_j$ denote the random variable modelling the model $\mathcal{M}_j$'s parameters, with the priors $\mathcal{L}_M(\cdot)$ ad $\mathcal{L}_{\mathbf{\Theta}_j}(\cdot)$.

Let $\langle X_i \rangle$ be a random sample of size n. For the model $\mathcal{M}_j$, we have the likelihood $\mathcal{L}^{\mathcal{M}_j}_{X|\mathbf{\Theta}_j}(x \mid \boldsymbol{\theta}_j)$. Hence our combined prior is given by

$$\mathcal{L}_{M,(\mathbf{\Theta}_j)}(m, \boldsymbol{\theta}_1 \dots, \boldsymbol{\theta}_K) = \mathcal{L}_M(m) \prod_{j=1}^{K} \mathcal{L}_{\mathbf{\Theta}_j}(\boldsymbol{\theta}_j).$$

We note that

$$\mathcal{L}_{X|M,(\mathbf{\Theta}_j)}(x \mid m, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K) = \sum_{j=1}^{K} I_{m=\mathcal{M}_j} \mathcal{L}^{\mathcal{M}_j}_{X|\mathbf{\Theta}_j}(x \mid \boldsymbol{\theta}_j).$$

So our evidence is given by

$$\begin{aligned}\mathcal{L}_{(X_i)|M,(\boldsymbol{\Theta}_j)}(\mathbf{x} \mid m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) &= \prod_{i=1}^{n} \mathcal{L}_{X_i|M,(\boldsymbol{\Theta}_j)}(x_i \mid m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \\ &= \prod_{i=1}^{n} \sum_{j=1}^{K} I_{m=\mathcal{M}_j} \mathcal{L}_{X|\boldsymbol{\Theta}_j}^{\mathcal{M}_j}(x_i \mid \boldsymbol{\theta}_j)\end{aligned}$$

So by Bayes' Theorem, we have

$$\begin{aligned}\mathcal{L}_{M,(\boldsymbol{\Theta}_j)|(X_i)}(m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \mid \mathbf{x}) &= \kappa \mathcal{L}_{(X_i)|M,(\boldsymbol{\Theta}_j)}(\mathbf{x} \mid m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \mathcal{L}_{M,(\boldsymbol{\Theta}_j)}(m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \\ &= \kappa \left(\prod_{i=1}^{n} \sum_{j=1}^{K} I_{m=\mathcal{M}_j} \mathcal{L}_{X|\boldsymbol{\Theta}_j}^{\mathcal{M}_j}(x_i \mid \boldsymbol{\theta}_j) \right) \left(\mathcal{L}_M(m) \prod_{j=1}^{K} \mathcal{L}_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta}_j) \right)\end{aligned}$$

Let us consider the marginal distribution $\mathcal{L}_{M|(X_i)}(m \mid \mathbf{x})$. So we have

$$\begin{aligned}\mathcal{L}_{M|(X_i)}(m \mid \mathbf{x}) &= \int_{\prod_{j=1}^{K} \overrightarrow{\boldsymbol{\Theta}_j}(\Omega)} \mathcal{L}_{M,(\boldsymbol{\Theta}_j)|(X_i)}(m, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \mid \mathbf{x}) \, \mathrm{d}\boldsymbol{\theta}_1 \cdots \mathrm{d}\boldsymbol{\theta}_K \\ &= \kappa \mathcal{L}_M(m) \int_{\prod_{j=1}^{K} \overrightarrow{\boldsymbol{\Theta}_j}(\Omega)} \prod_{i=1}^{n} \sum_{j=1}^{K} I_{m=\mathcal{M}_j} \mathcal{L}_{X|\boldsymbol{\Theta}_j}^{\mathcal{M}_j}(x_i \mid \boldsymbol{\theta}_j) \prod_{j=1}^{K} \mathcal{L}_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta}_j) \, \mathrm{d}\boldsymbol{\theta}_1 \cdots \mathrm{d}\boldsymbol{\theta}_K\end{aligned}$$

Without loss of generality, let $m = \mathcal{M}_\alpha$, so we have

$$\begin{aligned}\mathcal{L}_{M|(X_i)}(\mathcal{M}_\alpha \mid \mathbf{x}) &= \kappa \mathcal{L}_M(\mathcal{M}_\alpha) \int_{\prod_{j=1}^{K} \overrightarrow{\boldsymbol{\Theta}_j}(\Omega)} \prod_{i=1}^{n} \mathcal{L}_{X|\boldsymbol{\Theta}_\alpha}^{\mathcal{M}_\alpha}(x_i \mid \boldsymbol{\theta}_\alpha) \prod_{j=1}^{K} \mathcal{L}_{\boldsymbol{\Theta}_j}(\boldsymbol{\theta}_j) \, \mathrm{d}\boldsymbol{\theta}_1 \cdots \mathrm{d}\boldsymbol{\theta}_K \\ &= \kappa \mathcal{L}_M(\mathcal{M}_\alpha) \int_{\overrightarrow{\boldsymbol{\Theta}_1}(\Omega)} \mathcal{L}_{\boldsymbol{\Theta}_1}(\boldsymbol{\theta}_1) \, \mathrm{d}\boldsymbol{\theta}_1 \cdots \int_{\overrightarrow{\boldsymbol{\Theta}_\alpha}(\Omega)} \prod_{i=1}^{n} \mathcal{L}_{X|\boldsymbol{\Theta}_\alpha}^{\mathcal{M}_\alpha}(x_i \mid \boldsymbol{\theta}_\alpha) \mathcal{L}_{\boldsymbol{\Theta}_\alpha}(\boldsymbol{\theta}_\alpha) \, \mathrm{d}\boldsymbol{\theta}_\alpha \\ &\qquad \cdots \int_{\overrightarrow{\boldsymbol{\Theta}_n}(\Omega)} \mathcal{L}_{\boldsymbol{\Theta}_n}(\boldsymbol{\theta}_n) \, \mathrm{d}\boldsymbol{\theta}_n \\ &= \kappa \mathcal{L}_M(\mathcal{M}_\alpha) \int_{\overrightarrow{\boldsymbol{\Theta}_\alpha}(\Omega)} \prod_{i=1}^{n} \mathcal{L}_{X|\boldsymbol{\Theta}_\alpha}^{\mathcal{M}_\alpha}(x_i \mid \boldsymbol{\theta}_\alpha) \mathcal{L}_{\boldsymbol{\Theta}_\alpha}(\boldsymbol{\theta}_\alpha) \, \mathrm{d}\boldsymbol{\theta}_\alpha\end{aligned}$$

2.1.2 Interpreting the Posterior Distribution

- Desire to summarize posterior distribution.

Posterior Point Estimates

- **Maximum a posteriori**: The *maximum a posteriori* (MAP) estimates Θ, denoted $\hat{\theta}_{MAP}$ as the *mode* of the posterior distribution $(\Theta \mid \mathbf{X} = \mathbf{x})$:
$$\hat{\theta}_{MAP} = \arg\max_{\theta} \mathcal{L}_{\Theta|(X_i)}(\theta \mid \mathbf{X} = \mathbf{x}).$$

  For computational Bayes':

```
def θ̂_MAP(θ_sample, w):
    return θ_sample[np.argmax(w)]
```

- **Mean and Median**: Estimates Θ using the *mean* or *median* of the posterior distribution $(\Theta \mid \mathbf{X} = \mathbf{x})$
$$\hat{\theta}_{\mu} = \mathbb{E}\left[\Theta \mid \mathbf{x} = \mathbf{x}\right]$$
$$\hat{\theta}_{median} = \max_{\theta} \left\{ \theta \in \mathbb{R} : P(X \leq \theta) \geq \frac{1}{2} \wedge P(X \geq \theta) \geq \frac{1}{2} \right\}$$

  For computational Bayes':

```
def θ̂_μ(θ_sample, w):
    return np.sum(w * θ_sample)

def θ̂_median(θ_sample, w):
    θ_sample, w = np.sort(θ_sample), np.sort(w)
    F_Θ|(X_i) = np.cumsum(w)
    return θ_sample[F_Θ|(X_i) <= 0.5][-1]
```

- **Expected Posterior Loss**: Define a loss function: $L(\phi, \theta)$ which determines the loss if ϕ is the estimate and θ is the true value. The estimate $\hat{\theta}$ is
$$\hat{\theta} = \arg\min_{\phi} \mathbb{E}\left[L(\phi, \Theta) \mid \mathbf{X} = \mathbf{x}\right].$$

Example 2.1.4. (**Bayesian Linear Models**) For the labelled sample $\langle(\mathbf{X}_i, Y_i)\rangle$ of size n with $(Y_i \mid \mathbf{X}_i = \mathbf{x}_i, \boldsymbol{\beta}, \Sigma^2 = \sigma^2) \sim \mathcal{N}(\boldsymbol{\beta}^T \mathbf{x}_i, \sigma^2)$, where Σ^2 and $\boldsymbol{\beta}$ have the prior distributions:

$$\Sigma^2 \sim \text{InverseGamma}(a, b) \qquad (\boldsymbol{\beta} \mid \sigma^2) \sim \mathcal{N}_m(\boldsymbol{\mu}^0_{\boldsymbol{\beta}}, \sigma^2 \boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})$$

where $a, b > 0$ and $\mathbf{\Lambda}_{\boldsymbol{\beta}}^0$ is a positive symmetric matrix. Hence

$$\mathcal{L}_{\Sigma^2}(\sigma^2) = \frac{b^a}{\Gamma(a)}(\sigma^2)^{a-1}\exp\left[-\frac{b}{\sigma^2}\right]$$
$$\mathcal{L}_{\boldsymbol{\beta}|\Sigma^2}(\boldsymbol{\beta} \mid \sigma^2) = (2\pi\sigma^2)^{-\frac{m}{2}}(\det \mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-\frac{m}{2}}\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0)\right]$$

Find the MAP estimate for $\boldsymbol{\beta}$ given $\boldsymbol{\mu}_{\boldsymbol{\beta}}^0 = \mathbf{0}$ and $\mathbf{\Lambda}_{\boldsymbol{\beta}}^0 = \mathbf{I}_m$.

We note that $(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2) \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n)$. Hence

$$\mathcal{L}_{(Y_i)|(X_i),\boldsymbol{\beta},\Sigma^2}(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}}\exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right]$$

and

$$\mathcal{L}_{\boldsymbol{\beta},\Sigma^2}(\boldsymbol{\beta}, \sigma^2) = \mathcal{L}_{\boldsymbol{\beta}|\Sigma^2}(\boldsymbol{\beta} \mid \sigma^2)\mathcal{L}_{\Sigma^2}(\sigma^2).$$

Hence, by Bayes' Theorem we have

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{\beta},\Sigma^2|((X_i,Y_i))}(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}, \mathbf{X}) &= \kappa\mathcal{L}_{(Y_i)|(X_i),\boldsymbol{\beta},\Sigma^2}(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2)\mathcal{L}_{\boldsymbol{\beta},\Sigma^2}(\boldsymbol{\beta}, \sigma^2) \\
&= \kappa\left(2\pi\sigma^2\right)^{-\frac{n}{2}}\exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right] \\
&\quad \times (2\pi\sigma^2)^{-\frac{m}{2}}(\det \mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-\frac{m}{2}}\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0)\right] \\
&\quad \times \frac{b^a}{\Gamma(a)}(\sigma^2)^{a-1}\exp\left[-\frac{b}{\sigma^2}\right] \\
&= \kappa'(\sigma^2)^{-\frac{n}{2}-\frac{m}{2}+a-1}\exp\left[-\frac{A}{2\sigma^2}\right]
\end{aligned}$$

where

$$\begin{aligned}
A &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^0) + 2b \\
&= \mathbf{y}^T\mathbf{y} - \mathbf{y}\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\
&\quad + \boldsymbol{\beta}^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}^0 - (\boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{\beta} + (\boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}^0 \\
&\quad + 2b \\
&= \boldsymbol{\beta}^T\left[\mathbf{X}^T\mathbf{X} + (\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\right]\boldsymbol{\beta} - \boldsymbol{\beta}^T\left[\mathbf{X}^T\mathbf{y} + \left(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0\right)^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}^0\right] - \left[\mathbf{y}^T\mathbf{X} + \left(\boldsymbol{\mu}_{\boldsymbol{\beta}}^0\right)^T\left(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0\right)^{-1}\right]\boldsymbol{\beta} \\
&\quad + \left[(\boldsymbol{\mu}_{\boldsymbol{\beta}}^0)^T(\mathbf{\Lambda}_{\boldsymbol{\beta}}^0)^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}^0 + 2b + \mathbf{y}^T\mathbf{y}\right]
\end{aligned}$$

Let us define

$$\begin{aligned}\boldsymbol{\Lambda}_{\boldsymbol{\beta}} &= \left[\mathbf{X}^T\mathbf{X} + (\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})^{-1}\right]^{-1} \\ \boldsymbol{\mu}_{\boldsymbol{\beta}} &= \boldsymbol{\Lambda}_{\boldsymbol{\beta}}\left[\mathbf{X}^T\mathbf{y} + \left(\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}}\right)^{-1}\boldsymbol{\mu}^0_{\boldsymbol{\beta}}\right]\end{aligned}$$

So we have

$$\begin{aligned}A &= \boldsymbol{\beta}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta} + \left[(\boldsymbol{\mu}^0_{\boldsymbol{\beta}})^T(\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})^{-1}\boldsymbol{\mu}^0_{\boldsymbol{\beta}} + 2b + \mathbf{y}^T\mathbf{y}\right] \\ &= (\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}})^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}}) - \boldsymbol{\mu}_{\boldsymbol{\beta}}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} + \left[(\boldsymbol{\mu}^0_{\boldsymbol{\beta}})^T(\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})^{-1}\boldsymbol{\mu}^0_{\boldsymbol{\beta}} + 2b + \mathbf{y}^T\mathbf{y}\right]\end{aligned}$$

Hence

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\beta},\Sigma^2|((X_i,Y_i))}(\boldsymbol{\beta},\sigma^2 \mid \mathbf{y},\mathbf{X}) = \kappa'(\sigma^2)^{-\frac{m}{2}} \exp\left[-\frac{(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}})^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{\beta}})}{2\sigma^2}\right] \\ \times (\sigma^2)^{-\frac{n}{2}+a-1} \exp\left[-\frac{(\boldsymbol{\mu}^0_{\boldsymbol{\beta}})^T(\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})^{-1}\boldsymbol{\mu}^0_{\boldsymbol{\beta}} + 2b + \mathbf{y}^T\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}}{2\sigma^2}\right]\end{aligned}$$

We also note that

$$\mathcal{L}_{\boldsymbol{\beta},\Sigma^2|((X_i,Y_i))}(\boldsymbol{\beta},\sigma^2 \mid \mathbf{y},\mathbf{X}) = \mathcal{L}_{\boldsymbol{\beta}|\Sigma^2,((X_i,Y_i))}(\boldsymbol{\beta} \mid \sigma^2,\mathbf{y},\mathbf{X})\mathcal{L}_{\Sigma^2|((X_i,Y_i))}(\sigma^2 \mid \mathbf{y},\mathbf{X}).$$

Hence

$$\begin{aligned}&(\boldsymbol{\beta} \mid \sigma^2,\mathbf{y},\mathbf{X}) \sim \mathcal{N}_m(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \sigma^2\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}}) \\ &\left(\Sigma^2 \mid \mathbf{y},\mathbf{X}\right) \sim \text{InverseGamma}\left(-\frac{n}{2} + a, \frac{(\boldsymbol{\mu}^0_{\boldsymbol{\beta}})^T(\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}})^{-1}\boldsymbol{\mu}^0_{\boldsymbol{\beta}} + 2b + \mathbf{y}^T\mathbf{y} - \boldsymbol{\mu}_{\boldsymbol{\beta}}^T\boldsymbol{\Lambda}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}}}{2}\right)\end{aligned}$$

Note that the MAP estimate for $\boldsymbol{\beta}$ given $\boldsymbol{\mu}^0_{\boldsymbol{\beta}} = \mathbf{0}$ and $\boldsymbol{\Lambda}^0_{\boldsymbol{\beta}} = \mathbf{I}_m$ is the MLE / Least sequares estimate

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{MAP} &= \arg\max_{\boldsymbol{\beta}} \mathcal{L}_{\boldsymbol{\beta}|\Sigma^2,((X_i,Y_i))}(\boldsymbol{\beta} \mid \sigma^2,\mathbf{y},\mathbf{X}) \\ &= \arg\min_{\boldsymbol{\beta}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y}\end{aligned}$$

Posterior Confidence Intervals

Definition 2.1.2. (**Confidence Interval**) Let $\langle X_i \rangle$ be a random sample on $(\Omega, \mathcal{F}, P)$ of size n and $t : \langle X_i \rangle \to \mathbb{R}$ be a statistic.

A confidence interval for the statistic t with confidence level $\gamma \in [0, 1]$ is the interval $[l(\mathbf{X}), h(\mathbf{X})]$ such that

$$P(l(\mathbf{X}) \leq t(\mathbf{X}) \leq h(\mathbf{X})) = \gamma.$$

- To determine $[l(\mathbf{X}), h(\mathbf{X})]$:
  - Find the cumulative distribution $F_{t(\mathbf{X})}$ of $t(\mathbf{X})$.
  - Determine it's inverse, denoted $F^{-1}_{t(\mathbf{X})}$.
  - Since

$$\begin{aligned} & P(l(\mathbf{X}) \leq t(\mathbf{X}) \leq h(\mathbf{X})) = 1 - \alpha \\ \iff & P(t(\mathbf{X}) < l(\mathbf{X})) = \alpha/2 \qquad P(t(\mathbf{X}) > h(\mathbf{X})) = \alpha/2 \\ \iff & l(\mathbf{X}) = F^{-1}_{t(\mathbf{x})}(\alpha/2) \qquad h(\mathbf{X}) = F^{-1}_{t(\mathbf{X})}(1 - \alpha/2) \end{aligned}$$

    where α is the significance level s.t $\gamma = 1 - \alpha$.

- For computational methods:

```
# p is the likelihood vector for X
F_X = np.cumsum(p)
l, h = x_sample[F_X < α/2][-1], x_sample[F_X > 1 - α/2][0]
```

  or

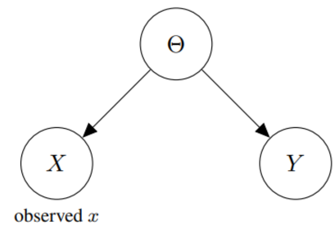
```
l, h = np.quantile(x_sample, [α/2, 1 - α/2])
```

- To interpret the posterior distribution $(\Theta \mid \mathbf{X} = \mathbf{x})$, we may compute a a confidence interval e.g. 95 %:

$$P(l \leq \Theta \leq h \mid \mathbf{X} = \mathbf{x}) = 0.95.$$

2.1.3 Posterior Predictive Distribution

- Suppose $\langle Y_i \rangle$ is a sample on $(\Omega, \mathcal{F}, P)$ of size M with causal diagram:



- We have, by the Law of Iterated Expectation,

$$\begin{aligned}\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X} = \mathbf{x}\right] &= \int_{\theta \in \overrightarrow{\Theta}(\Omega)} \mathbb{E}(\mathbf{Y} \mid \Theta = \theta, \mathbf{X} = \mathbf{x}) \mathcal{L}_{\Theta}(\theta \mid \mathbf{X} = \mathbf{x}) \, \mathrm{d}\theta \\ &= \int_{\theta \in \overrightarrow{\Theta}(\Omega)} \mathbb{E}(\mathbf{Y} \mid \Theta = \theta) \mathcal{L}_{\Theta}(\theta \mid \mathbf{X} = \mathbf{x}) \, \mathrm{d}\theta\end{aligned}$$

Definition 2.1.3. (**Posterior Predictive Distribution**) The posterior predictive distribution is the distribution of $\langle Y_i \rangle$ marginalized over the posterior:

$$\mathcal{L}_{(Y_i)|(X_i)}(y \mid \mathbf{x}) = \int_{\theta \in \overrightarrow{\Theta}(\Omega)} \mathcal{L}_{(Y_i)|\Theta}(\mathbf{Y} \mid \Theta = \theta) \mathcal{L}_{\Theta}(\theta \mid \mathbf{X} = \mathbf{x}) \, \mathrm{d}\theta \, .$$

Example 2.1.5. For the sample $\langle X_i \rangle$ of size n with $(X_i \mid \Theta) \sim \text{Bernoulli}(\theta)$ where $\Theta \sim U[0,1]$. Given that $(Y \mid \Theta) \sim \text{Bernoulli}(\theta)$, determine the predictive posterior distribution $(Y \mid \mathbf{X})$.

From example ?? we have $(\Theta \mid \mathbf{X}) \sim \text{Beta}(S+1, n-S+1)$ where $S = \sum_{i=1}^{n} I_{X_i=1}$. Hence by definition ?? we have

$$\begin{aligned}\mathcal{L}_{Y|(X_i)}(y \mid \mathbf{x}) &= \int_0^1 (\theta I_{y=1} + (1-\theta) I_{y=0}) \frac{\Gamma(n+2)}{\Gamma(S+1)\Gamma(n-S+1)} \theta^S (1-\theta)^{n-S} \, \mathrm{d}\theta \\ &= \frac{\Gamma(n+2)}{\Gamma(S+1)\Gamma(n-S+1)} \left[I_{y=1} \int_0^1 \theta^{S+1} (1-\theta)^{n-S} \, \mathrm{d}\theta + I_{y=0} \int_0^1 \theta^S (1-\theta)^{n-S+1} \, \mathrm{d}\theta \right]\end{aligned}$$

Define

$$I(x,y) = \int_0^1 \theta^x (1-\theta)^y \, \mathrm{d}\theta \, .$$

Integrating by parts and determining $I(x,0)$ yields

$$I(x,y) = \frac{x!y!}{(x+y+1)!} = \frac{\Gamma(x+1)\Gamma(y+1)}{\Gamma(x+y+2)} .$$

Hence

$$\begin{aligned}\mathcal{L}_{Y|(X_i)}(y \mid \mathbf{x}) &= \frac{\Gamma(n+2)}{\Gamma(S+1)\Gamma(n-S+1)}\left[I_{y=1}I(S+1, n-S) + I_{y=0}I(S, n-S+1)\right] \\ &= \frac{\Gamma(n+2)}{\Gamma(S+1)\Gamma(n-S+1)}\left[I_{y=1}\frac{\Gamma(S+2)\Gamma(n-S+1)}{\Gamma(n+3)} + I_{y=0}\frac{\Gamma(S+1)\Gamma(n-S+2)}{\Gamma(n+3)}\right]\end{aligned}$$

Recall that $\Gamma(z+1) = z\Gamma(z)$, hence

$$\mathcal{L}_{Y|(X_i)}(y \mid \mathbf{x}) = \frac{1}{n+2}\begin{cases} S+1 & \text{if y = 1} \\ n-S+1 & \text{otherwise}\end{cases}$$

S the predictive posterior distribution is $(Y \mid \mathbf{X}) \sim \text{Bernoulli}\left(\dfrac{S+1}{n+2}\right)$.

2.2 Frequentism

- **Philosophy**:

Frequentist definition of probability holds

Definition 2.2.1. (**Frequentist Definition of Probability**) The probability measure P on $(\Omega, \mathcal{F}, P)$ of some event $A \in \mathcal{F}$ is

$$P(A) = \lim_{n \to \infty} \frac{n(A)}{n},$$

where $n(A)$ is the number of trials where A occurs and n is the number of trials.

- Determine parameters $\boldsymbol{\theta}$ of model using resampling. Reason about them using probability (or confidence intervals)

2.2.1 Parametric and Non-Parametric Resampling

- **Parametric Resampling**: Let $\langle x_i \rangle$ be a dataset from $\langle X_i \rangle$ of size n. We have parametric model $\mathcal{L}_X(x \mid \boldsymbol{\theta})$:

  1. Define a *test statistic* $t : \langle X_i \rangle \to \mathbb{R}$ on the sample $\langle X_i \rangle$.

2. Fit the model, determine MLE[$\boldsymbol{\theta}$].
3. Define the synthetic sample of size n $\langle X_i^* \rangle$ generated by $\mathcal{L}_X(\cdot \mid \text{MLE}[\boldsymbol{\theta}])$.
4. Generate N synthetic datasets $\langle x_{ij}^* \rangle$. This is *parametric resampling*.
5. Compute $t\left(\langle x_{ij}^* \rangle\right)$ for all $1 \leq j \leq N$.

Example 2.2.1. (**Distribution and Confidence Interval of** MLE[μ]) For the dataset $\langle x_i \rangle$ of the sample $\langle X_i \rangle$ of size n distributed by $X_i \sim \mathcal{N}(\mu, \sigma^2)$ where μ, σ^2 are unknown parameters. Determine the distribution and the 95% confidence interval of MLE[μ] using parametric resampling.

```
x = [ ... ]
n = len(x)

# Define our test statistic MLE[μ]
def μ̂(x): return np.mean(x)

# Fit the model
MLE[μ] = μ̂(x)
MLE[σ] = np.sqrt(np.mean((x − μ̂(x)) ** 2))

# Define synthetic sample
def x*():
    return np.random.normal(loc=MLE[μ], scale=MLE[σ], size=n)

# Generate synthetic datasets and compute test statistic
N = ...
μ̂_sample = [μ̂(x*()) for _ in range(N)]

# Plot distributon of MLE[μ]
M = ...
plt.hist(μ̂_sample, bins=M)

# Determine confidence interval
l,h = np.quantile(μ̂_sample, [.025, 0.975])
```

Example 2.2.2. (**Comparing Groups**) For two random samples $\langle X_i \rangle$ and $\langle Y_i \rangle$ of size m and n respectively s.t $X_i \sim \mathcal{N}(\mu, \sigma^2)$ and $Y_i \sim \mathcal{N}(\mu + \delta, \sigma^2)$ with parameters $\mu, \delta \in \mathbb{R}$ and $\sigma^2 > 0$. Determine the distribution of MLE[δ]

From supervision 1, we note that

$$\text{MLE}[\mu] = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{MLE}[\delta] = \frac{1}{n}\sum_{i=1}^{n} y_i - \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{MLE}[\sigma] = \sqrt{\frac{1}{m+n}\left(\sum_{i=1}^{m}(x_i-\mu)^2 + \sum_{i=1}^{n}(y_i-\mu-\delta)^2\right)}$$

So we have

```
x,y = [ ... ], [ ... ]
m, n = ...

# Define test statistic
def t(x, y):
    return np.mean(y) - np.mean(x)

# Fit model
MLE[μ] = np.mean(x)
MLE[δ] = t(x, y)
MLE[σ] = np.sqrt(
    (np.sum(x - MLE[μ]) ** 2) + (np.sum(y - MLE[μ] - MLE[δ]) ** 2)
    / (m + n))

# Define synthetic sample
def xy*():
    return (np.random.normal(loc=MLE[μ], scale=MLE[σ], size=m),
            np.random.normal(loc=MLE[μ] + MLE[δ], scale=MLE[σ], size=n))

# Generate synthetic datasets and compute t
N = ...
t_sample = [t(*xy*()) for _ in range(N)]

# Plot distributon of t
M = ...
plt.hist(t_sample, bins=M)
```

- **Problem**: Model $\mathcal{L}_X(x \mid \boldsymbol{\theta})$ can't always be determined.
- **Solution**: Use empirical distribution (best fit for the given dataset). This is *non-parametric resampling*
- **Non-parametric Resampling**: Let $\langle x_i \rangle$ be a dataset from $\langle X_i \rangle$ of size n.

  1. Define a *test statistic* $t : \langle X_i \rangle \to \mathbb{R}$ on the sample $\langle X_i \rangle$.

2. Define synthetic sample of size n $\langle X_i^* \rangle$ using empirical distribution X^*. See ??
3. Generate N synthetic datasets $\left\langle x_{ij}^* \right\rangle$.
4. Compute $t\left(\left\langle x_i^j \right\rangle\right)$ for all $1 \leq j \leq N$.

Example 2.2.3. (**Comparing Groups**) For two random samples $\langle X_i \rangle$ and $\langle Y_i \rangle$ of size m and n respectively. Determine the distribution of $t = \overline{Y} - \overline{X}$ using non-parameteric resampling.

```
x, y = [...], [...]
m, n = len(x), len(y)

# Define test statistic
def t(x, y):
    return np.mean(y) - np.mean(x)

# Define synthetic sample
def xy*():
    return (np.random.choice(x, size=m),
            np.random.choice(y, size=n))

# Generate synthetic datasets
N = ...
t_sample = [t(*xy*()) for _ in range(N)]
```

Example 2.2.4. (**Coin Toss**) For a random sample $\langle X_i \rangle$ of size n where $X_i \sim \text{Bernoulli}(p)$ models the result of the ith coin toss. Find a 95% confidence interval of MLE $[p]$ using non-parameteric resampling.

```
x = [ ... ]
n = len(x)

# Define test statistic
def p̂(x):
    return np.mean(x)

# Define synthetic sample
def x*():
    return np.random.choice(x, size=n)

# Generate synthetic datasets and compute p̂
N = ...
p̂_sample = [p̂(x*()) for _ in range(N)]

# Compute confidence interval
l,h = np.quantile(p̂_sample, [.025, .975])
```

2.2.2 Hypothesis Testing

- Desire to "prove" statements based on samples $\langle X_i \rangle$ within a confidence interval $\gamma = 1 - \alpha$.
- **Hypothesis Testing**: Let $\langle x_i \rangle$ be a dataset from $\langle X_i \rangle$ of size n.
  1. Define a *test statistic* $t : \langle X_i \rangle \to \mathbb{R}$ on the sample $\langle X_i \rangle$.
  2. Define the null and alternate hypothesizes H_0, H_1 using t.
  3. Generate N synthetic datasets $\left\langle x_{ij}^* \right\rangle$ using resampling assuming H_0 holds.
  4. Compute $t\left(\left\langle x_i^j \right\rangle\right)$ for all $1 \leq j \leq N$ and the observed t, denoted $t\left(\langle x_i \rangle\right)$. Plotting it's distribution using a Histogram.
  5. Determine p-value: the probability of a more *extreme* value than $t\left(\langle x_i \rangle\right)$.
  6. If $p \leq \alpha$, the significance level, then reject H_0.
- Two types of tests:
  - **One-sided**: $H_0 : \theta = \theta_0$, and $H_1 : \theta > \theta_0$ (or $\theta < \theta_0$). α corresponds to the critical value t_c s.t $P(t(\mathbf{X}) \geq t_c) = \alpha$:

```
def reject_H0(t_dataset, t, α):
    p1, p2 = np.mean(t_dataset >= t), np.mean(t_dataset <= t)
    return min(p1, p2) < α
```

  - **Two-sided**: $H_0 : \theta = \theta_0$ and $H_1 : \theta \neq \theta_0$. α corresponds to the critical value t_c s.t $2P(t(\mathbf{X}) \geq t_c) = \alpha$:

```
def reject_H0(t_dataset, t, α):
    p1, p2 = np.mean(t_dataset >= t), np.mean(t_dataset <= t)
    return 2 * min(p1, p2) < α
```

- The significance level α is the probability of rejecting the null hypothesis given it is true.

Example 2.2.5. (**Sign Test**) Let $\langle (X_i, Y_i) \rangle$ be a random sample of size n. Define $Z_i = X_i - Y_i$. Let $W = \sum_{i=1}^{n} I_{Z_i > 0}$ with $W \sim B(n, p)$. Define H_0 and H_1 s.t

$$H_0 : p = \frac{1}{2} \qquad\qquad H_1 : p \neq \frac{1}{2}$$

So we have the following test:

```
x, y = [...], [...]
n = len(x)

# Define test statistic
def p(w):
    return np.mean(w)

# Define synthetic sample assuming H_0
def w*():
    return np.random.binomial(n=n, p=1/2)

# Compute observed p
p_observed = p(np.where(x - y > 0, 1, 0))

# Generate synthetic datasets and compute p
N = ...
p_dataset = [p(w*()) for _ in range(N)]

# Plot distribution
plt.hist(p_dataset, bins=...)
plt.axvline(p_observed)

# Determine p-value
def reject_H_0(t_dataset, t, α):
    p_1, p_2 = np.mean(t_dataset >= t), np.mean(t_dataset <= t)
    return min(p_1, p_2) < α

# Reject H_0?
α = .05
print(f"Reject H_0 w/ α={α}? {reject_H_0(p_dataset, p_observed, α)}")
```

Example 2.2.6. (**Equality of** μ) For the two random samples $\langle X_i \rangle$ and $\langle Y_i \rangle$ of sizes m and n, respectively s.t $X_i \sim \mathcal{N}(\mu_X, \sigma^2)$ and $Y_i \sim \mathcal{N}(\mu_Y, \sigma^2)$.

Let us define

$$H_0 : \mu_X = \mu_Y \qquad H_1 : \mu_X \neq \mu_Y$$

with the test statistic

$$t = (\text{MLE}[\mu_X] - \text{MLE}[\mu])^2 + (\text{MLE}[\mu_Y] - \text{MLE}[\mu])^2,$$

where $X_i, Y_i \sim \mathcal{N}(\mu, \sigma^2)$, assuming H_0 holds.

So we have

```
x, y = [ ... ], [ ... ]
m, n = len(x), len(y)
σ = ...

# Define test statistic
def t(x, y):
    μ̂ = np.mean(np.concatenate([x, y]))
    return (np.mean(x) - μ̂) ** 2 + (np.mean(y) - μ̂) ** 2
```

```

# Define synthetic sample
def xy*():
    μ̂ = np.mean(np.concatenate([x, y]))
    return (np.random.normal(loc=μ̂, scale=σ, size=m),
            np.random.normal(loc=μ̂, scale=σ, size=n))

# Compute observed t
t_observed = t(x, y)

# Generate synthetic datasets and compute t
N = ...
t_dataset = [t(*xy*()) for _ in range(N)]

# Plot distribution
plt.hist(t_dataset, bins=...)
plt.axvline(t_observed)

# Determine p-value
def reject_H0(t_dataset, t, α):
    p1, p2 = np.mean(t_dataset >= t), np.mean(t_dataset <= t)
    return min(p1, p2) < α

# Reject H0?
α = .05
print(f"Reject H0 w/ α={α}? {reject_H0(t_dataset, t_observed, α)}")
```

3 Markov Chains

3.1 Causal Diagrams

Definition 3.1.1. A causal diagram is a directed acyclic graph $G = (V, E, P)$ where V is the set of random variables, P is the set of (hyper)parameters and $E \subseteq (V \cup P)^2$ is the edges or dependencies between variables.

$$\begin{tikzcd}
p \arrow[d, dash, dotted] & & \\
K \arrow[d] & \mu \arrow[dl, dash, dotted] & \sigma \arrow[dll, dash, dotted] \\
Y \arrow[d] & & \\
X & &
\end{tikzcd}$$

- Causal diagrams capture dependencies and are useful for describing probability models.

$$\begin{tikzcd}
A \arrow[dr] & & B \arrow[dl] \\
& X &
\end{tikzcd}$$

$$\mathcal{L}_{X,Y,Z}(x, y, z) = \mathcal{L}_X(x)\mathcal{L}_{Y|X}(y \mid x)\mathcal{L}_{Z|Y}(z, y)$$

$$\begin{tikzcd}
& \Theta \arrow[dl] \arrow[dr] & \\
X & & Y
\end{tikzcd}$$

$$\mathcal{L}_{\Theta,X,Y}(\theta, x, y) = \mathcal{L}_\Theta(\theta)\mathcal{L}_{X|\Theta}(x \mid \theta)\mathcal{L}_{Y|\Theta}(y \mid \theta)$$

$$\begin{tikzcd}
M \arrow[d] \arrow[dr] & H \arrow[d] \\
X & Y
\end{tikzcd}$$

$$\mathcal{L}_{M,H,X,Y}(m, h, x, y) = \mathcal{L}_M(m)\mathcal{L}_H(h)\mathcal{L}_{X|M}(x \mid m)\mathcal{L}_{Y|M,H}(y \mid m, h)$$

3.1.1 Causal Diagram Analysis

- **Method**:
  - Use the law of total probability to include the *parents* into the expression
  - Use Bayes' Theorem to calculate probability conditional on the *children*
  - Use independence for simplification
- **Examples**:

$$\begin{tikzcd} K \arrow[d] \\ X \end{tikzcd}$$

$$\mathcal{L}_X(x) = \sum_{k \in \vec{K}(\Omega)} \mathcal{L}_{X|K}(x \mid k)\mathcal{L}_K(k)$$

Total Probability

$$\begin{tikzcd} K \arrow[d] \\ X \end{tikzcd}$$

$$\mathcal{L}_{K|X}(k \mid x) = \kappa \mathcal{L}_K(k)\mathcal{L}_{X|K}(x \mid k)$$

Bayes' Theorem

$$\begin{tikzcd} A \arrow[dr] & & B \arrow[dl] \\ & X & \end{tikzcd}$$

$$\mathcal{L}_{X|A}(x \mid a) = \sum_{b \in \vec{B}(\Omega)} \mathcal{L}_{X|A,B}(x \mid a, b)\mathcal{L}_B(b)$$

Total Probability with Baggage $A = a$

$$\begin{tikzcd} M \arrow[d] \arrow[dr] & H \arrow[d] \\ X & Y \end{tikzcd}$$

$$\begin{aligned} &\mathcal{L}_{H|X,Y}(h \mid x, y) \\ &= \sum_{m \in \vec{M}(\Omega)} \mathcal{L}_{H|M,X,Y}(h \mid m, x, y)\mathcal{L}_{M|X,Y}(m \mid x, y) \\ &= \kappa \sum_{m \in \vec{M}(\Omega)} \mathcal{L}_H(h)\mathcal{L}_M(m)\mathcal{L}_{X|M}(x \mid m)\mathcal{L}_{Y|H,M}(y \mid h, m) \end{aligned}$$

Both

3.2 Markov Chains

- $\mathcal{S}$ is the state space (a countable set).
- Working on probability space $(\Omega, \mathcal{F}, P)$ with random variables $X : \Omega \to \mathcal{S}$.

Definition 3.2.1. (**Markov Chain**) $(X_i)_{n\geq 0}$ is a *Markov Chain* with initial distribution λ and transition matrix P, denoted (λ, P), if for all $n \geq 0$, $s_0, \ldots, s_{n+1} \in \mathcal{S}$:

1. $P(X_0 = s_0) = \lambda_{s_0}$
2. $P(X_{n+1} = s_{n+1} \mid X_0 = s_0, \ldots, X_n = s_n) = P(X_{n+1} = s_{n+1} \mid X_n = s_n) = p_{s_n s_{n+1}}$

- (2) introduces *memorylessness* property.
- Useful to reason about Markov Chains using causal diagram. $(X_i)_{n\geq 0}$ is a Markov Chain (λ, P) if $(X_i)_{n\geq 0}$ has the following causal diagram:

$$X_0 \longrightarrow X_1 \longrightarrow X_2 \longrightarrow \cdots .$$

- P is a *stochastic matrix* (each row of P is a distribution over $\mathcal{S}$):

$$\forall s, t \in \mathcal{S}. p_{st} \geq 0 \qquad\qquad \forall s \in \mathcal{S}. \sum_{t \in \mathcal{S}} p_{st} = 1.$$

Theorem 3.2.1. $(X_i)_{n\geq 0}$ with (λ, P) is a Markov Chain iff

$$\forall n \geq 0, s_0, \ldots, s_n \in \mathcal{S}. P(X_0 = s_0, \ldots, X_n = s_n) = \lambda_{s_0} p_{s_0 s_1} \cdots p_{s_{n-1} s_n}.$$

Proof. ($\implies$) Let us assume $(X_i)_{n\geq 0}$ with (λ, P) is a Markov Chain. Let $n \geq 0, s_0, \ldots, s_n \in \mathcal{S}$ be arbitrary.

$$\begin{aligned} P(X_0 = s_0, \ldots, X_n = s_n) &= P(X_n = s_n \mid X_0 = s_0, \ldots, X_{n-1} = s_{n-1}) P(X_0 = s_0, \ldots, X_{n-1} = s_{n-1}) \\ &= P(X_0 = s_0) P(X_1 = s_1 \mid X_0 = s_0) \cdots P(X_n = s_n \mid X_{n-1} = s_{n-1}) \\ &= \lambda_{s_0} p_{s_0 s_1} \cdots p_{s_{n-1} s_n} \end{aligned}$$

($\impliedby$) Let $n \geq 0, s_0, \ldots, s_n \in \mathcal{S}$ be arbitrary. Assume $P(X_0 = s_0, \ldots, X_n = s_n) = \lambda_{s_0} p_{s_0 s_1} \cdots p_{s_{n-1} s_n}$. Then

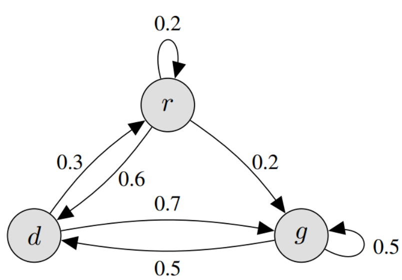
$$\begin{aligned} P(X_0 = s_0) = &\sum_{s_1, \ldots, s_n \in \mathcal{S}} P(X_0 = s_0, \ldots, X_n = s_n) = \lambda_{s_0} \sum_{s_1, \ldots, s_n \in \mathcal{S}} p_{s_0 s_1} \cdots p_{s_{n-1} s_n} \\ &= \lambda_{s_0} \sum_{s_1 \in \mathcal{S}} p_{s_0 s_1} \sum_{s_2 \in \mathcal{S}} p_{s_1 s_2} \cdots \sum_{s_n \in \mathcal{S}} p_{s_{n-1} s_n} \\ &= \lambda_{s_0} \end{aligned}$$

and

$$P(X_n = s_n \mid X_0 = s_0, \dots, X_{n-1} = s_{n-1}) = \frac{P(X_0 = s_0, \dots, X_n = s_n)}{P(X_0 = s_0, \dots, X_{n-1} = s_{n-1})} = p_{s_{n-1}s_n}$$

□

Example 3.2.1. (**Weather Model**) For a Markov chain with *state space diagram* and transition matrix:



$$P = \begin{array}{c c} & \begin{array}{ccc} r & d & g \end{array} \\ \begin{array}{c} r \\ d \\ g \end{array} & \begin{pmatrix} 0.2 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0 & 0.5 & 0.5 \end{pmatrix} \end{array}$$

where the states r, g, d correspond to rain, grey and drizzle respectively. We have the following implementation:

```
S = ["r", "g", "d"]
P = np.array([
    [.2, .6, .2],
    [.3, 0, .7],
    [0, .5, .5]
])
# Assert P is stochastic
assert all(P.sum(axis = 1) == 1)

def weather(x0):
    i = S.index(x0)
    while True:
        yield S[i]
        i = np.random.choice(len(S), p=P[i])
```

- **Sliding window technique**:
  - For a causal diagram:

$$\begin{tikzcd} X_0 \arrow[rr, bend left] & X_1 \arrow[r] \arrow[rr, bend left] & X_2 \arrow[r] \arrow[rr, bend left] & X_3 \arrow[r] \arrow[rr, bend left] & X_4 \arrow[r] \arrow[rr, bend left] & X_5 \arrow[r] & \cdots \end{tikzcd}$$

- Define a "sliding window" $Y_n = (X_n, X_{n+1})$, which yields the causal diagram:

$$Y_0 \longrightarrow Y_1 \longrightarrow Y_2 \longrightarrow \cdots .$$

which is a Markov Chain $(Y_i)_{n\geq 0}$.

3.2.1 The n-step Transition Matrix

Theorem 3.2.2. For Markov Chain $(X_i)_{n\geq 0}$ with (λ, P):

$$P(X_n = s_n) = \sum_{s_0,\ldots,s_{n-1}\in\mathcal{S}} \lambda_{s_0} p_{s_0 s_1} \cdots p_{s_{n-1} s_n} = (\lambda P^n)_{s_n}.$$

Proof.

$$\begin{aligned}
P(X_n = s_n) &= \sum_{s_0,\ldots,s_{n-1}\in\mathcal{S}} P(X_0 = s_0, \ldots, X_n = s_n) \\
&= \sum_{s_0,\ldots,s_{n-1}\in\mathcal{S}} \lambda_{s_0} p_{s_0 s_1} \cdots p_{s_{n-1} s_n} \\
&= \sum_{s_0} \lambda_{s_0} \left(\sum_{s_{n-1}} \cdots \left(\sum_{s_2} \left(\sum_{s_1} p_{s_0 s_1} p_{s_1 s_2} \right) p_{s_2 s_3} \right) \cdots p_{s_{n-1} s_n} \right) \\
&= \sum_{s_0} \lambda_{s_0} p^{(n)}_{s_0 s_n} = (\lambda P^n)_{s_n}
\end{aligned}$$

□

- Numpy calculations with n-step transition matrices:

```
λ, P = np.array([ ... ]), ...
assert all(P.sum(axis = 1) == 1)

# Compute P(X_n = s_n | X_0 = s_0)
np.linalg.matrix_power(P, n)[s_0, s_n]
```

Example 3.2.2. (**Hidden Markov Model (Particle Filter)**) A *Hidden Markov Model* is defined by the causal diagram:

$$\begin{tikzcd}
X_0 \arrow[r] \arrow[d] & X_1 \arrow[r] \arrow[d] & X_2 \arrow[r] \arrow[d] & X_3 \arrow[r] \arrow[d] & \cdots \\
Y_0 & Y_1 & Y_2 & Y_3 &
\end{tikzcd}$$

where $(X_i)_{n\geq 0}$ is a Markov Chain w/ (λ, P), state-space $\overrightarrow{X}(\Omega) = Q$ and Y_i is a random variable on $(\Omega, \mathcal{F}, P)$ defined on $\overrightarrow{Y}(\Omega) = O$, the emission or observation space, with emission matrix $E = (e)$, where $e_x(y) = p_{Y|X}(y \mid x)$.

Suppose we have the dataset $\langle y_i \rangle$ from the random sample $\langle Y_i \rangle$ of size n. For $n \geq 0$, let us define $\pi_n(\cdot)$ as

$$\pi_n(x) = p_{X_n|(Y_i)}\left(x \mid Y_0 = y_1, \ldots, Y_n = y_n\right) = p_{X_n|(Y_i)}(x \mid y_1, \ldots, y_n).$$

For $n = 0$, we have

$$\begin{aligned}
\pi_0(x_0) &= p_{X_0|Y_0}(x_0 \mid y_0) \\
&= \kappa p_{Y_0|X_0}(y_0 \mid x_0) p_{X_0}(x_0) && \text{(Bayes' Theorem)} \\
&= \kappa p_{Y_0|X_0}(y_0 \mid x_0) \lambda_{x_0}
\end{aligned}$$

For $n \geq 1$, we have

$$\begin{aligned}
\pi_n(x_n) &= p_{X_n|(Y_i)}\left(x_n \mid y_0, \ldots, y_n\right) \\
&= \kappa' p_{Y_n|(Y_i),X_n}(y_n \mid y_0, \ldots, y_{n-1}, x_n) \cdot p_{X_n|(Y_i)}(x_n \mid y_0, \ldots, y_{n-1}) \\
&= \kappa' p_{Y_n|X_n}(y_n \mid x_n) \cdot p_{X_n|(Y_i)}(x_n \mid y_0, \ldots, y_{n-1}) \\
&= \kappa' e_{x_n}(y_n) \cdot p_{X_n|(Y_i)}(x_n \mid y_0, \ldots, y_{n-1})
\end{aligned}$$

where $e_x(y) = p_{Y_i|X_i}(y \mid x)$. By the law of total probability, we have

$$\begin{aligned}
p_{X_n|(Y_i)}(x_n \mid y_0, \ldots, y_{n-1}) &= \sum_{x_{n-1} \in Q} p_{X_n|(Y_i),X_{n-1}}(x_n \mid y_0, \ldots, y_{n-1}, x_{n-1}) \\
&\qquad\qquad \cdot p_{X_{n-1}|(Y_i)}(x_{n-1} \mid y_0, \ldots, y_{n-1}) \\
&= \sum_{x_{n-1} \in Q} p_{X_n|X_{n-1}}(x_n \mid x_{n-1}) p_{X_{n-1}|(Y_i)}(x_{n-1} \mid y_0, \ldots, y_{n-1})
\end{aligned}$$

by definition of a Markov chain. So we have

$$\begin{aligned}
\pi_0(x_0) &= \kappa e_{x_0}(y_0) \lambda_{x_0} \\
\pi_n(x_n) &= \kappa \sum_{x_{n-1} \in Q} p_{x_{n-1} x_n} \pi_{n-1}(x_{n-1}) e_{x_n}(y_n)
\end{aligned}$$

Supposing we have functions $e(y, x) = e_x(y)$ and $p(i, j) = p_{ij}$, we have the following Python code for computing the vector $\pi = (\pi_n)$ gives dataset $\mathbf{y} = [\ y_0,\ ...,\ y_n\ ]$.

```
Q, λ = ...

def π(y):
    n = len(y)
    assert(n > 0)

    if (n == 1):
        Π = [e(y[0], i) * λ for i in Q]
    else:
        Π_{n−1} = π(y[:−1])
        Π = [np.sum([
            p(j, i) * Π_{n−1}[j] * e(y[−1], i) for j in Q
        ]) for i in Q]

    return Π / np.sum(Π)
```

3.2.2 Class Structure and Irreducibility

Definition 3.2.2. State s is said to lead to t if

$$\exists n \geq 0 . P(X_n = t \mid X_0 = s) > 0.$$

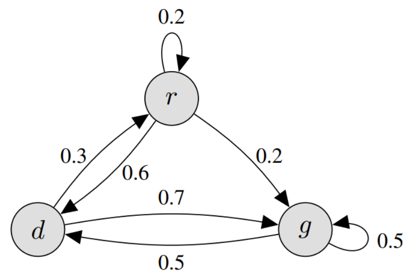
Denoted $s \to t$.

- We write $s \leftrightarrow t \iff s \to t \wedge t \to s$.
- Note that $s \to t \iff \exists n \geq 0 . p_{st}^{(n)} > 0$.
- $\leftrightarrow$ is an equivalent relation on $\mathcal{S}$, thus partitions $\mathcal{S}$ into *classes* $C_1, \ldots, C_m$. IMAGE

Definition 3.2.3. Closed Class A class C is said to be closed if:

$$\forall s \in C, t \in \mathcal{S} . s \to t \implies t \in C.$$

- A class that is not closed is said to be *open*.
- The Markov Chain $(X_i)_{n \geq 0}$ is said to be **irreducible** if $\mathcal{S}$ is the only class. (The state-space diagram is connected)



3.2.3 Hitting Probabilities

- Let $(X_i)_{n\geq 0}$ be a Markov Chain (λ, P).

Definition 3.2.4. (**Hitting Time**) The hitting time of the set of states $A \subseteq \mathcal{S}$ is the random variable $H^A : \Omega \to \mathbb{N} \cup \{\infty\}$ s.t

$$H^A(\omega) = \min \{n \geq 0 : X_n(\omega) \in A\},$$

where $\min \emptyset = \infty$.

Definition 3.2.5. (**Hitting Probability**) The hitting probability is the probability that $(X_n)_{n\geq 0}$ hits A given $X_0 = s$:

$$h_s^A = P(H^A < \infty \mid X_0 = s).$$

Theorem 3.2.3. The vector of hitting probabilities h^A satisfies:

$$h_s^A = \begin{cases} 1 & s \in A \\ \sum_{t\in\mathcal{S}} p_{st} h_t^A & s \notin A \end{cases}.$$

Proof. Let $s \in \mathcal{S}$ be arbitrary Two cases:

- Case $X_0 = s \in A$: Then $H^A = 0$. Hence $h_s^A = 1$.
- Case $X_0 = s \notin A$: Then $H^A \geq 1$. So

$$\begin{aligned} h_s^A = P(H^A < \infty \mid X_0 = s) &= \sum_{t\in\mathcal{S}} P(H^A < \infty, X_1 = t \mid X_0 = s) \\ &= \sum_{t\in\mathcal{S}} P(H^A < \infty \mid X_1 = t, X_0 = s) P(X_1 = t \mid X_0 = s) \end{aligned}$$

Note that $P(H^A < \infty \mid X_1 = t, X_0 = s) = P(H^A < \infty \mid X_1 = t) = h_t^A$ by the Markov Property (a sub-chain conditional on $X_m = s$ is a chain). Hence

$$h_s^A = \sum_{t \in \mathcal{S}} P(X_1 = t \mid X_0 = s) h_t^A = \sum_{t \in \mathcal{S}} p_{st} h_t^A$$

□

- Computationally, this consists of solving the set of equations

$$\begin{aligned} h_s^A &= 1 & s &\in A \\ h_s^A &= \sum_{t \in \mathcal{S}} p_{st} h_t^A & s &\notin A \end{aligned}$$

```
P = np.array([ ... ])
n = len(P)

assert all(P.sum(axis = 1) == 1)

A = [ ... ]
# Solve Ph = h or (P - I_n)h = 0 (except for s ∈ A)
B = P - np.eye(n)
x = np.zeros(n)

# Set h_s^A = 1 for all s ∈ A
for s in A:
    B[s, :] = np.zeros(n)
    B[s, s] = 1
    x[s] = 1

# Compute h s.t Bh = x
h, *_ = np.linalg.lstsq(B, x)
```

3.2.4 Stationary Distributions

Definition 3.2.6. (**Stationary Distribution**) A distribution λ is said to be stationary if $\lambda P = \lambda$.

- Hence for a stationary Markov Chain $(X_i)_{n \geq 0}$ with (λ, P), there exists μ s.t:
$$\forall s \in \mathcal{S}, n \geq 0. P(X_n = s) = \mu_s = (\mu P^n)_s.$$

- Computationally, this consists of solving:

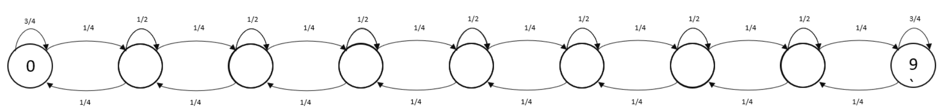
$$\begin{aligned}(P - I_n)^T \lambda &= \mathbf{0} \\ \mathbf{1}^T \lambda &= 1\end{aligned}$$

```
P = np.array([ ... ])
n = len(P)

# Compute (P - I_n)^T and 1^T and concatenate into matrix A
A = np.concatenate([P - np.eye(n).transpose(), np.ones((1, n)) ])
x = np.concatenate([ np.zeros(n), [1] ])

π, *_ = np.linalg.lstsq(A, x)
```

Example 3.2.3. Suppose we have the Markov chain $(X_n)_{n\geq 0}$ w/ state-space $\overrightarrow{X}(\Omega) = Q = \{0, 1, \ldots, 9\}$ initial distribution $\lambda = (1, 0, \ldots, 0)$, and causal diagram:



So we have the following transition matrix P such that

$$p_{00} = p_{99} = \frac{3}{4}, \qquad p_{ii} = \frac{1}{2}, 1 \leq i \leq 8$$

$$p_{i(i+1)} = \frac{1}{4}, 0 \leq i \leq 8 \qquad p_{i(i-1)} = \frac{1}{4}, 1 \leq i \leq 9$$

and $p_{ij} = 0$, otherwise.

We wish to compute the stationary distribution π of $(X_n)_{n\geq 0}$. Giving us the following Python code for calculating π:

```
P = np.array([
    [.75, .25, .0, .0, .0, .0, .0, .0, .0],
    [.25, .5, .25, .0, .0, .0, .0, .0, .0],
    [.0, .25, .5, .25, .0, .0, .0, .0, .0],
    [.0, .0, .25, .5, .25, .0, .0, .0, .0],
    [.0, .0, .0, .25, .5, .25, .0, .0, .0],
    [.0, .0, .0, .0, .25, .5, .25, .0, .0],
    [.0, .0, .0, .0, .0, .25, .5, .25, .0],
    [.0, .0, .0, .0, .0, .0, .25, .5, .25],
    [.0, .0, .0, .0, .0, .0, .0, .25, .75]
])

A = np.concatenate([(P - np.eye(9)).transpose(), np.ones((1,9)) ])
x = np.concatenate([np.zeros(9), [1]])

π, *_ = np.linalg.lstsq(A, x)
```

Theorem 3.2.4. If P is irreducible, then there is a unique stationary distribution μ.

Definition 3.2.7. (**Detailed Balance**) A Markov Chain $(X_i)_{n\geq 0}$ with (λ, P) is said to be in detailed balance if there exists μ s.t

$$\forall t, s \in \mathcal{S}. \mu_s p_{st} = \mu_t p_{ts}.$$

Theorem 3.2.5. If P and μ are in detailed balance, then μ is stationary for P, that is

$$\mu = \mu P.$$

Proof. $(\mu P)_s = \sum_{t\in\mathcal{S}} \mu_t p_{ts} = \sum_{t\in\mathcal{S}} \mu_s p_{st} = \mu_s$. □

3.2.5 Limit Theorems

- Define the first return to state s as $T_s = \min\{n \geq 1 : X_n = s\}$ with $\min\emptyset = \infty$.
- $V_s(n) = \sum_{t=0}^{n-1} I_{X_t=s}$. The number of visits to state s before time n.
- $V_s^t = V_s(T_t)$. The number of visits to s before first return to t.
- $\gamma_s^t = \mathbb{E}\left[V_s^t \mid X_0 = t\right]$. The mean number of visits to s between successive visits to t.

Theorem 3.2.6. If P is irreducible with stationary distribution μ and $(X_i)_{n\geq 0}$ is a Markov Chain (λ, P), then

$$P\left(\lim_{n\to\infty} \frac{V_s(n)}{n} = \mu_s\right) = 1.$$

Definition 3.2.8. (**Aperiodic**) A state s is said to be aperiodic if there exists $n \geq 1$ s.t

$$\forall N \geq n. p_{ss}^{(N)} > 0.$$

Theorem 3.2.7. For irreducible P and aperiodic state s, then for all states are aperiodic.

Proof. Let $t, r \in \mathcal{S}$ be arbitrary. Since P is irreducible, then there exists $n_t, n_r > 0$ s.t $p_{ts}^{(n_t)}, p_{sr}^{(n_r)} > 0$. Then

$$\forall N \geq n_s . p_{tr}^{(n_t+N+n_r)} = \sum_{i,j \in \mathcal{S}} p_{ti}^{(n_t)} p_{ij}^{(N)} p_{jr}^{(n_r)} \geq p_{ts}^{(n_t)} p_{ss}^{(N)} p_{sr}^{(n_r)} > 0.$$

Instantiating the above for $t = r = i$ yields $p_{ii}^{(N')} > 0$ for all $N' \geq n_t + n_s + n_r$. Hence all states are aperiodic.

□

Theorem 3.2.8. For irreducible and aperiodic P with stationary distribution μ. Then for any Markov Chain $(X_i)_{n \geq 0}$ with (λ, P)

$$\forall s, t \in \mathcal{S}. \lim_{n \to \infty} P(X_n = t) = p_{st}^{(n)} = \mu_s.$$