TP 1 – AOS1

Introduction to Bayesian inference
Corrigé

1 Introduction

In this practical session, we will use the `numpy`, `scipy`, and `matplotlib` packages.

```
import numpy as np
import scipy as sp
import scipy.stats as spst
import matplotlib.pyplot as plt
import itertools
```

2 Maximum likelihood estimation

2.1 Random sample generation

First, we are interested in generating random samples according to some specific (user-defined) distribution.

(1) For the binomial and Poisson distributions, pick a particular parameter value, generate a sample of desired size, visualize the empirical distribution of the data (using `plt.bar`), and compare it to the actual distribution (using `distrib.pmf`).

Example with the binomial distribution with parameters $N = 100$ and $p = 0.4$, where we generate a sample of size $n = 1000$:

```
distrib = spst.binom(n=100, p=0.4)
x = distrib.rvs(size=1000)
t = np.arange(start=0, stop=100, step=1)
freq = [np.mean(x==i) for i in range(100)]

fig1, axs1 = plt.subplots(1, 2, sharex=True, tight_layout=True)
axs1[0].bar(x=range(0,100), height=freq)
axs1[1].plot(t, distrib.pmf(t))
```

(2) Do the same for the beta, gamma, exponential and Gaussian distributions. You may display the empirical distribution of the sample using using `plt.hist` and the actual (theoretical) distribution using using `distrib.pdf`.

Example with the Gaussian distribution:

```
distrib = spst.norm(loc=0, scale=1)
x = distrib.rvs(size=1000)
t = np.arange(start=-5, stop=5, step=0.1)

fig2, axs2 = plt.subplots(1, 2, sharex=True, tight_layout=True)
axs2[0].hist(x, range=(-5,5), bins=20)
axs2[1].plot(t, distrib.pdf(t))
```

2.2 Likelihood plot

(3) Program a function `loglike` which computes the log-likelihood of a parameter given a sample and a family of distributions; for instance, given a vector of values sampled according to $\boldsymbol{X} \sim \mathcal{N}(\mu, \sigma^2)$, we would compute the log-likelihood $\ln L(\mu = 0, \sigma = 2; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ by:

```
loglike(spst.norm, (0,2), x)
```

where `x` contains the data sample. Beware that for multivariate distributions, instances are by convention stored row-wise in `x`.

```
def loglike(distrib, params, sample):
    try:
        return np.sum(distrib.logpdf(sample, *params), axis=0)
    except AttributeError:
        return np.sum(distrib.logpmf(sample, *params), axis=0)
```

We have here a single function `loglike` which admits the distribution family, its parameters and the sample into account. This however requires to test for the nature (discrete or continuous) of the distribution.

(4) Write a script which plots the likelihood for a single parameter. Using some of the previous distributions, locate the maximum likelihood estimate of the parameter. What do you notice when you increase the size of the sample ?

The solution is straightforward in the case of a single parameter: for example, $\ln L(\mu, \sigma = 1; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is displayed for $\mu \in [-2; 2]$ by:

```
par = np.arange(-2.0, 2.0, 0.025)

ll = [loglike((i,1), spst.norm, x) for i in par]

fig3, ax3 = plt.subplots()
ax3.plot(par, ll)
ax3.set_title('$\log L(\mu,\sigma^2=1)$')
ax3.set_xlabel('$\mu$')
```

(5) Write a script which plots the level curves of the likelihood for a couple of parameter values. Using the Gaussian distribution, locate the maximum likelihood estimate of the parameter vector (μ, σ^2). What do you notice when you increase the size of the sample ?

In the case of two parameters, plotting the level curves is a bit trickier:

```
par1 = np.arange(-2.0, 2.0, 0.025)
par2 = np.arange(0.25,2,0.025)

ll = [[loglike((i,j), spst.norm, x) for i in par1] for j in par2]

fig4, ax4 = plt.subplots()
par1, par2 = np.meshgrid(par1, par2)
CS = ax4.contour(par1, par2, np.asarray(ll), levels=250)
ax4.clabel(CS, inline=1, fontsize=10)
ax4.set_title('$\log L(\mu,\sigma^2)$')
ax4.set_xlabel('$\mu$')
ax4.set_ylabel('$\sigma^2$')
```

With a high number of observations, we can see that the log-likelihood is sharper around the maximum-likelihood estimate. It illustrates the consistency of the ML estimator.

3 Bayesian updating using conjugate priors

3.1 Beta-binomial distribution

(6) Recall the expression of the beta distribution. What is its definition domain ? On which parameters does it depend ? What are the expectation, the mode, and the variance ? Which particular distribution can be retrieved as a special case of the beta distribution ?

The pdf of the beta distribution is defined by

$$\pi_\theta(t|\alpha,\beta) = \mathrm{beta}(t;\alpha,\beta) = \frac{t^{\alpha-1}(1-t)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}, \text{ with } \mathrm{B}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)},$$

and Γ is the Gamma function. It depends on two parameters $\alpha > 0$ and $\beta > 0$. We have

$$\mathbb{E}[\theta] = \frac{\alpha}{\alpha+\beta}, \quad \mathrm{Mode}(\theta) = \frac{\alpha-1}{\alpha+\beta-2} \text{ for } \alpha > 1, \beta > 1, \quad \mathrm{Var}(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

A special case, retrieved with $\alpha = \beta = 1$, is the uniform distribution on the $[0;1]$ interval.

(7) Write a script which plots the prior distribution given a set of chosen hyper-parameter values.

The pdf of the beta distribution can be represented using the following code.

```
distrib = spst.beta(a=0.5,b=0.5)

t = np.arange(start=0, stop=1, step=0.01)

fig1, ax1 = plt.subplots()
ax1.plot(t, distrib.pdf(t))
```

(8) Consider a random variable X following a binomial sampling distribution $\mathcal{B}(n,\theta)$, with n known and $\theta \sim \mathrm{beta}(\alpha,\beta)$. Compute the posterior distribution of θ given x, α and β.

The posterior distribution of θ is also a beta distribution:

$$L(\theta|x) = p_X(x|\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x};$$
$$\pi_\theta(t|x;\alpha,\beta) \propto L(t|x)\pi_\theta(t|\alpha,\beta) = \binom{n}{x}\frac{1}{\mathrm{B}(\alpha,\beta)}t^{x+\alpha-1}(1-t)^{n-x+\beta-1}.$$

It is common to interpret the hyper-parameters α and β of the prior as a number of "pseudo-observations" of X. In the binomial setting, x stands for the number of successes of the underlying independently repeated Bernoulli experiment, and its complement $n-x$ as the number of failures. If we properly compute the normalizing constant of the posterior distribution, we obtain

$$\pi_\theta(t|x;\alpha,\beta) = \frac{L(t|x)\pi_\theta(t|\alpha,\beta)}{\int_0^1 L(t|x)\pi_\theta(t|\alpha,\beta)dt} = \frac{t^{x+\alpha-1}(1-t)^{n-x+\beta-1}}{\mathrm{B}(\alpha+x,\beta+n-x)};$$

we recognize here the beta distribution with parameters $\alpha+x$ and $\beta+n-x$: then, α and β can therefore be thought of as numbers of virtual successes and failures — the more such virtual outcomes there are, the more the prior distribution will influence the inference. This also shows that the beta prior is conjugate to the binomial distribution.

9 Plot $\pi_\theta(\cdot)$, $L(\cdot|x)$ and $\pi_\theta(\cdot|x;\alpha,\beta)$.

The prior, sampling and posterior distributions can be plot using the following code.

```
nbin = 10
alph, beta = (5, 1)

sampl_dist = spst.binom(n=nbin, p=0.2)
prior_dist = spst.beta(a=alph, b=beta)

x = sampl_dist.rvs()
post_dist = spst.beta(a=alph+x, b=beta+nbin-x)

t = np.arange(start=0.01, stop=1, step=0.01)
ll = [loglike((nbin,p), spst.binom, x) for p in t]

fig, axs = plt.subplots(1, 3, sharex=True, tight_layout=True)
axs[0].plot(t, prior_dist.logpdf(t))
axs[1].plot(t, ll)
axs[2].plot(t, post_dist.logpdf(t))
```

10 Let us now assume that we have previously observed $X = x$ positive outcomes out of n outcomes. What is the predictive distribution for the number X_0 of positive outcomes out of n_0 new experiments, given $X = x$ out of n, α and β ?

The predictive distribution for X_0 is

$$\begin{aligned}
p_{X_0}(x_0|x;\alpha,\beta) &= \int_0^1 p_{X_0}(x_0|t)\pi_\theta(t|x;\alpha,\beta)dt,\\
&= \int_0^1 \binom{n_0}{x_0}t^{x_0}(1-t)^{n_0-x_0}\mathrm{beta}(\alpha+x,\beta+n-x)dt,\\
&= \binom{n_0}{x_0}\frac{1}{\mathrm{B}(\alpha+x,\beta+n-x)}\int_0^1 t^{\alpha+x+x_0-1}(1-t)^{\beta+n-x+n_0-x_0-1}dt,\\
&= \binom{n_0}{x_0}\frac{\mathrm{B}(\alpha+x+x_0,\beta+n-x+n_0-x_0)}{\mathrm{B}(\alpha+x,\beta+n-x)}.
\end{aligned}$$

3.2 Gamma-Poisson distribution

11 Recall the expression of the gamma distribution, its definition domain, the parameters on which it depends. Recall its expectation, mode, and variance.

The pdf of the gamma distribution with parameters $\alpha > 0$ and $\beta > 0$ is

$$\pi_\theta(t|\alpha, \beta) = \text{gamma}(\text{t}; \alpha, \beta) = \frac{\beta^\alpha t^{\alpha-1}\exp(-\beta t)}{\Gamma(\alpha)}, \text{ with } \Gamma(\alpha) = \int_0^{+\infty} t^{\alpha-1}\exp(-t)dt,$$

(again, Γ is the Gamma function). Note that the gamma distribution can also be expressed using parameters α and $\eta = 1/\beta$; generally, η is referred to as *scale* parameter, and β as *rate* parameter. The formulation using the former is more frequent, but using the latter makes computing the posterior distribution more obvious; we shall therefore use it. We have

$$\mathbb{E}[\theta] = \frac{\alpha}{\beta}, \quad \text{Mode}(\theta) = \frac{\alpha-1}{\beta} \text{ for } \alpha > 1, \quad \text{Var}(\theta) = \frac{\alpha}{\beta^2}.$$

Plot the prior distribution given a set of chosen hyper-parameter values.

The pdf of the beta distribution can be represented using the following code.

```
distrib = spst.gamma(a=9,scale=1/2)

t = np.arange(start=0, stop=15, step=0.1)

fig1, ax1 = plt.subplots()
ax1.plot(t, distrib.pdf(t))
```

Note that Python uses the scale parametrization.

12 Consider a random variable $X \sim \mathcal{P}(\theta)$, with $\theta \sim \text{gamma}(\alpha, \beta)$. What is the posterior distribution of θ given an iid sample $x_1, \dots, x_n$, α and β ?

Plot $\pi_\theta(\cdot)$, $L(\cdot|x_1, \dots, x_n)$ and $\pi_\theta(\cdot|x_1, \dots, x_n; \alpha, \beta)$, for various values of θ, α, β and n.

The posterior distribution of θ is also a gamma distribution. Indeed, assuming $x_i \geqslant 0$ for all $i = 1, \dots, n$,

$$L(\theta|x_1, \dots, x_n) = \prod_{i=1}^n \Pr(X = x_i|\theta) = \exp(-n\theta)\frac{\theta^{\sum_i x_i}}{\prod_i x_i!};$$

$$\pi_\theta(t|x_1, \dots, x_n; \alpha, \beta) = \frac{L(t|x_1, \dots, x_n)\pi_\theta(t|\alpha, \beta)}{\displaystyle\int_0^{+\infty} L(t|x_1, \dots, x_n)\pi_\theta(t|\alpha, \beta)} = \frac{\exp\left(-(n+\beta)t\right)\, t^{\alpha+\sum_i x_i - 1}}{\displaystyle\int_0^{+\infty} \exp\left(-(n+\beta)t\right)\, t^{\alpha+\sum_i x_i - 1} dt}.$$

Remark that

$$\begin{aligned}\int_0^{+\infty} \exp\left(-(n+\beta)t\right)\, t^{\alpha+\sum_i x_i-1} dt &= \frac{\Gamma(\alpha + \sum_i x_i)}{(\beta+n)^{\alpha+\sum_i x_i}} \int_0^{+\infty} \text{gamma}(t; \alpha + \sum_i x_i, \beta + n)dt \\ &= \frac{\Gamma(\alpha + \sum_i x_i)}{(\beta+n)^{\alpha+\sum_i x_i}};\end{aligned}$$

Hence, the gamma prior is conjugate for the Poisson distribution:

$$\pi_\theta(t|x_1, \dots, x_n; \alpha, \beta) = \text{gamma}\left(t; \alpha + \sum_i x_i, \beta + n\right).$$

The prior, sampling and posterior distributions can be plot using the same code as above.

13 Assume that an iid sample $x_1, \dots, x_n$ of realizations of $X \sim \mathcal{P}(\theta)$ has been observed. Show that the predictive distribution of a new outcome x_0 given the sample, α and β is a negative

binomial (or Pólya) distribution. At some point, you may want to make a change of integration variable, by replacing t with $z = (\beta + n + 1)t$.

Assuming $x_0 \geqslant 0$, we integrate with respect to θ:

$$\begin{aligned} p_{X_0}(x_0|x_1,\ldots,x_n;\alpha,\beta) &= \int_0^{+\infty} p_{X_0}(x_0|t)\pi_\theta(t|x_1,\ldots,x_n;\alpha,\beta)dt, \\ &= \frac{(\beta+n)^{\alpha+\sum_i x_i}}{\Gamma(\alpha+\sum_i x_i)\, x_0!} \int_0^{+\infty} \exp\left(-(n+\beta+1)t\right)\, t^{\alpha+\sum_i x_i + x_0 - 1} dt. \end{aligned}$$

Let us now make the change of variable:

$$z = (\beta+n+1)t \quad \Leftrightarrow \quad t = (\beta+n+1)^{-1} z \quad \Rightarrow \quad dt = (\beta+n+1)^{-1} dz;$$

since the bounds of the integral are not modified by this change of variable, this yields

$$p_{X_0}(x_0|x_1,\ldots,x_n;\alpha,\beta) = \frac{(\beta+n)^{\alpha+\sum_i x_i}}{\Gamma(\alpha+\sum_i x_i)\, x_0!} \int_0^{+\infty} \frac{\exp\left(-z\right)\, z^{\alpha+\sum_i x_i + x_0 - 1}}{(\beta+n+1)^{\alpha+\sum_i x_i + x_0}} dz.$$

The integral corresponds to a specific value for the Gamma function Γ: we finally obtain

$$\begin{aligned} p_{X_0}(x_0|x_1,\ldots,x_n;\alpha,\beta) &= \frac{\Gamma(\alpha+\sum_i x_i + x_0)}{\Gamma(\alpha+\sum_i x_i)\, x_0!} \frac{(\beta+n)^{\alpha+\sum_i x_i}}{(\beta+n+1)^{\alpha+\sum_i x_i + x_0}}, \\ &= \frac{\Gamma(\alpha+\sum_i x_i + x_0)}{\Gamma(\alpha+\sum_i x_i)\, x_0!} \left(\frac{\beta+n}{\beta+n+1}\right)^{\alpha+\sum_i x_i} \frac{1}{(\beta+n+1)^{x_0}}, \\ &= \frac{\Gamma(\alpha+\sum_i x_i + x_0)}{\Gamma(\alpha+\sum_i x_i)\, x_0!} \left(\frac{\beta+n}{\beta+n+1}\right)^{\alpha+\sum_i x_i} \left(1 - \frac{\beta+n}{\beta+n+1}\right)^{x_0}, \end{aligned}$$

which corresponds to the pdf of the "generalized" negative binomial distribution, when the numbers of successes and trials are not necessarily integers. Thus, we have

$$X_0|x_1,\ldots,x_n,\alpha,\beta \sim \operatorname{Neg\,Bin}\left(\alpha + \sum_i x_i, \frac{\beta+n}{\beta+n+1}\right).$$

3.3 Normal-gamma Gaussian distribution

We now consider a Gaussian random variable $X \sim \mathcal{N}(\mu, \lambda^{-1})$, where the Gaussian distribution is parameterized using the expectation μ and the *precision* $\lambda = 1/(\sigma^2)$.

Classically, a normal-gamma prior is used for parameters μ and λ:

$$\begin{aligned} \pi_\lambda(\ell|\alpha,\beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)} \ell^{\alpha-1} \exp(-\beta\ell), \qquad \text{for } \ell > 0; \\ \pi_{\mu|\lambda}(u|\nu,\lambda,\eta) &= (2\,\pi)^{-1/2} \lambda^{1/2} \exp\left(-\frac{\eta\lambda}{2}(t-\nu)^2\right), \qquad \text{for } t \in \mathbb{R}. \end{aligned}$$

The parameter η is called the *shrinkage* parameter of the normal prior.

(14) Compute the pdf of the normal-gamma prior, i.e. the joint pdf $\pi_{\mu,\lambda}(u,\ell)$.

To compute the pdf, we simply have to multiply the pdfs defined in both equations above:

$$\begin{aligned} \pi_{\mu,\lambda}(u,\ell|\nu,\eta,\alpha,\beta) &= \pi_{\mu|\lambda}(u|\nu,\ell,\eta) \cdot \pi_\lambda(\ell|\alpha,\beta), \\ &= (2\,\pi)^{-1/2} \ell^{1/2} \exp\left(-\frac{\eta\ell}{2}(t-\nu)^2\right) \frac{\beta^\alpha}{\Gamma(\alpha)} \ell^{\alpha-1} \exp\left(-\beta\ell\right), \\ &= (2\,\pi)^{-1/2} \frac{\beta^\alpha}{\Gamma(\alpha)} \ell^{\alpha-1/2} \exp\left(-\ell\left(\frac{\eta}{2}(t-\nu)^2 + \beta\right)\right). \end{aligned}$$

Display the contour plot of the normal-gamma prior for various values of α, β, ν and η.

The contour plot of the normal-gamma prior density can be obtained by

```
alph, beta = (1, 1)
eta, nu = (2, 0)

t1 = np.arange(start=-2.95, stop=3, step=0.05)
t2 = np.arange(start=0.05, stop=4, step=0.05)

prior = [[spst.norm.pdf(i,loc=nu,scale=np.sqrt(1/(eta*j)))*
          spst.gamma.pdf(j,a=alph,scale=1/beta) for i in t1] for j in t2]

t1, t2 = np.meshgrid(t1, t2)
fig2, ax2 = plt.subplots()
CS = ax2.contour(t1, t2, np.asarray(prior), levels=50)

ax2.clabel(CS, inline=1, fontsize=10)
ax2.set_title('$\pi_{\mu,\lambda}(u,\ell)$')
ax2.set_xlabel('$u$')
ax2.set_ylabel('$\ell$')
```

(15) Assume that we have observed an iid sample $x_1, \dots, x_n$ of realizations of a random variable $X \sim \mathcal{N}(\mu, \lambda^{-1})$. Recall the expression for the likelihood function $L(\mu, \lambda)$.

We have

$$L(\mu, \lambda | x_1, \dots, x_n) = \prod_{i=1}^n p_x(x_i; \mu, \lambda) = (2\pi)^{-n/2} \lambda^{n/2} \exp\left(-\frac{\lambda}{2} \sum_{i=1}^n (x_i - \mu)^2\right).$$

(16) Show that the posterior distribution for (μ, σ^2) given the sample, λ, α and β is a normal-gamma distribution. You may drop the computation of the denominator (normalization constant).

We have

$$p_{\mu,\lambda}(u, \ell | x_1, \dots, x_n; \nu, \eta, \alpha, \beta) \propto \ell^{1/2} \exp\left(-\frac{\ell}{2}\Big(\sum_i (x_i - t)^2 + \eta(t - \nu)^2\Big)\right) \ell^{\alpha + n/2 - 1} \exp(-\beta \ell).$$

We can remark that

$$\sum_{i=1}^n (x_i - t)^2 = \sum_{i=1}^n (x_i - \overline{x})^2 + n(\overline{x} - t)^2;$$

$$n(\overline{x} - t)^2 + \eta(t - \nu)^2 = (n + \eta)(t - \tilde{\nu})^2 + \frac{n\eta(\overline{x} - \nu)^2}{n + \eta}, \qquad \text{with } \tilde{\nu} = \frac{n\overline{x} + \eta\nu}{n + \eta};$$

$$\text{thus,} \quad \sum_i (x_i - t)^2 + \eta(t - \nu)^2 = \sum_{i=1}^n (x_i - \overline{x})^2 + (n + \eta)(t - \tilde{\nu})^2 + \frac{n\eta(\overline{x} - \nu)^2}{n + \eta}.$$

This finally gives

$$p_{\mu,\lambda}(u, \ell | x_1, \dots, x_n; \lambda, \alpha, \beta) \propto \mathcal{N}\left(u | \tilde{\nu}, ((n + \eta)\ell)^{-1}\right) \times \text{gamma}\left(\ell | \alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^n (x_i - \overline{x})^2 + \frac{n\eta(\overline{x} - \nu)^2}{2(n + \eta)}\right).$$

The normal-gamma prior is thus conjugate for the normal sampling distribution.

Display the prior, likelihood, and posterior contours, for various values of n.

The contour plots can be displayed using the following code.

```
alph, beta = (1, 1)
eta, nu = (2, 0)
mu, sig2, n = (1, 2, 2)
x = spst.norm.rvs(size=n, loc=mu, scale=np.sqrt(sig2))

t1 = np.arange(start=-4.95, stop=5, step=0.1)
t2 = np.arange(start=0.05, stop=4, step=0.05)

prior = [[spst.norm.pdf(i,loc=nu,scale=np.sqrt(1/(eta*j)))*
          spst.gamma.pdf(j,a=alph,scale=1/beta) for i in t1] for j in t2]

ll = [[loglike((i,np.sqrt(1/j)), spst.norm, x) for i in t1] for j in t2]

nutilde = (x.sum()+eta*nu)/(n+eta)
alphtilde = alph+n/2
betatilde = beta+n*x.var()/2+(n*eta*(x.mean()-nu)**2)/(2*(n+eta))
post = [[spst.norm.pdf(i,loc=nutilde,scale=np.sqrt(1/((n+eta)*j)))*
         spst.gamma.pdf(j,a=alphtilde,scale=1/(betatilde))
         for i in t1] for j in t2]

fig, ax = plt.subplots(3, 1, sharex=True, sharey=True, tight_layout=True,
                       figsize=(5,15))
t1_, t2_ = np.meshgrid(t1, t2)

CS = ax[0].contour(t1_, t2_, np.asarray(prior), levels=10)
ax[0].clabel(CS, inline=1, fontsize=10)
ax[0].set_title('$\pi_{\mu,\lambda}(u,\ell)$')
ax[0].set_xlabel('$u$')
ax[0].set_ylabel('$\ell$')

CS = ax[1].contour(t1_, t2_, np.asarray(np.exp(ll)), levels=10)
ax[1].clabel(CS, inline=1, fontsize=10)
ax[1].set_title('$L(\mu=u,\lambda=\ell)$')
ax[1].set_xlabel('$u$')
ax[1].set_ylabel('$\ell$')

CS = ax[2].contour(t1_, t2_, np.asarray(post), levels=10)
ax[2].clabel(CS, inline=1, fontsize=10)
ax[2].set_title('$\pi_{\mu,\lambda}(u,\ell|\cdot)$')
ax[2].set_xlabel('$\mu$')
ax[2].set_ylabel('$\lambda$')
```