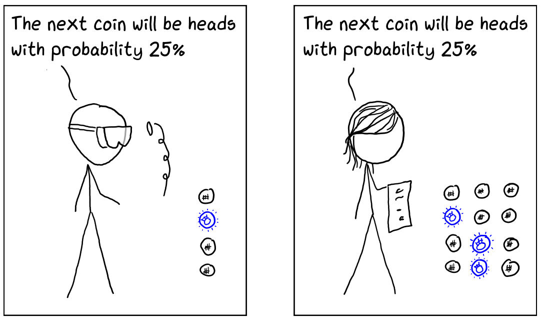
Part III
Inference

Inference is the business of drawing conclusions about the world based on data.

If one scientist tosses four coins and gets one head, he'll estimate that the next coin will be heads with probability 25%. If another scientist tosses 12 coins and gets three heads, she'll make the same estimate. But clearly there is a difference between the two—the second scientist should be more confident in her estimate.

exercise 1.1 page 1: after observing x heads from n tosses, the estimated probability of heads is $\hat{p} = x/n$

Everything that we learnt in part I was about making estimates. But an estimate on its own is *not* a sensible statement about the world—it would almost certainly be incorrect to say "the actual true probability of heads *is* 25%". To say something sensible, something that at least has the *form* of a valid statement about the world, we need a way to express not just the estimate but also our confidence in the estimate.[16]

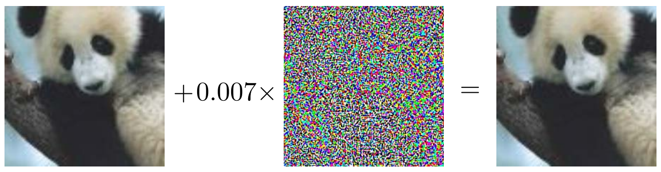

The next coin will be heads with probability 25%
The next coin will be heads with probability 25%

How should confidence be measured? Statisticians have spent the better part of the last century thinking about this, and their work has broadly speaking fallen into two schools of thought, Bayesianism and frequentism. In this part of the book, we will explore what these two schools have to say about inference.

* ✻ *

The machine learning community has by and large concentrated on estimation and has failed to grasp the challenge of inference (though there are exceptions—for example there is a strand of work at Cambridge bringing Bayesianism into deep learning.) The example below is a summary, of the sort you'll see all over blog posts and news stories, of some ingenious research into adversarial attacks against neural network image classifiers. The summary confuses 'probability estimate' with 'confidence'. If we get our concepts muddled right at the beginning when we're framing a problem, we have no hope of solving it.

Example 5.1 (The adversarial panda[17]).



`panda,` 57.7% confidence — `nematode,` 8.2% confidence — `gibbon,` 99.3% confidence

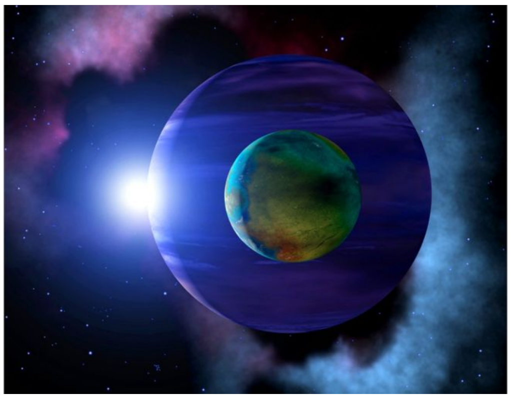
A neural network was trained to classify images. When shown the leftmost image, it reports "`panda, 57.7% confidence`". The center image is carefully chosen noise. By blending the

[16]On the other hand, if our goal is to make decisions rather than inferences, then there's nothing wrong with "I shall proceed as though the probability of heads is 25%" This has the form of a valid statement, because it's a statement about something we know, namely our own plans. Whether or not it's a good idea is another matter—and that depends on confidence.

original panda with noise at 0.7% opacity, we obtain the rightmost image, which the neural network interprets as “`gibbon, 99.3% confidence`”.

Scientists also struggle with inference:

Example 5.2 (Extract from BBC News[18]).
Signal may be from first ‘exomoon’



The work by [these astronomers] assigns a confidence level of four sigma to the signal from the distant planetary system. The confidence level describes how unlikely it is that an experimental result is simply down to chance. If you express it in terms of tossing a coin, it’s equivalent to tossing 15 heads in a row.

But [one of the astronomers] said this is not the best way to gauge the potential detection. He told BBC News: “We’re excited about it... statistically, formally, it’s a very high probability. But do we really trust the statistics? That’s something unquantifiable. Until we get the measurements from Hubble, it may as well be 50–50 in my mind.”

Formal statistics *is* the way to quantify uncertainty. If an astronomer doesn’t trust the statistics, then they are doing the wrong statistics—they haven’t framed their problem in a way that captures their uncertainty, nor how data can shift that uncertainty. It’s the job of a scientist to say “This is how empirical evidence can change my beliefs.” If you can’t do this, if you haven’t got the conceptual tools to handle empirical evidence, then you’re not a scientist.

[17]I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *ArXiv e-prints* (Dec. 2014). arXiv: `1412.6572 [stat.ML]`

[18]`http://www.bbc.co.uk/news/science-environment-40741545`

6. Bayesianism

Suppose, as usual, that we collected data x and we believe it comes from a probability model $\Pr_X(x|\theta)$, where X is a random variable representing possible outcomes of an experiment and θ is an unknown parameter about which we'd like to make inferences.

Bayesianism says we should represent our uncertainty about unknown parameters by using a probability distribution. Instead of "this is a parameter whose value I don't know", we say "this is a parameter whose actual value I don't know, but whose *distribution* I do know". Our starting point isn't "complete ignorance about θ", it's "my prior beliefs tell me about the distribution that θ comes from".

One view of probability is that it's a tool for describing random events, for example the number of particles emitted from a lump of radioactive matter (commonly modelled with a Poisson distribution). Bayesianism is a gigantic conceptual step away from this. Bayesianism says that not only should we use probability to describe random phenomena in nature, we should also use it to describe our subjective state of mind. Not only do we work with $\Pr_X(x \mid \theta)$, we also work with $\Pr_\Theta(\theta)$ and $\Pr_\Theta(\theta \mid X = x)$.

6.1. Finding the posterior distributions

The steps in a Bayesian analysis.

1. Let Θ denote the unknown parameter, treated as a random variable. Invent a distribution for it, $\Pr_\Theta(\theta)$. This is called the *prior distribution*. It must include *all* the unknown parameters for the problem.

   Bayesianism requires us to set down a prior belief about θ. If we don't have a prior belief, Bayesianism says, then there are no grounds for us to make inferences. An exam question will tell you which prior to use, but real life doesn't.

2. Write out $\Pr_X(x|\theta)$, the density for the observed data conditional on θ. Here X represents *all* the observed data, whether it be a single value or a collection.

3. Use Bayes's rule to find the distribution of $(\Theta \mid X = x)$. This step is called *applying the Bayes update*, and the resulting distribution is called the *posterior distribution*. We can apply Bayes's rule computationally or mathematically, as described in section 6.

4. Any conclusions we want to draw about the unknown parameters should be expressed as statements about the distribution of $(\Theta \mid X = x)$. For example, we could plot a histogram. Other standard readouts are described in section 6.2.

6.1.1. USING COMPUTATION

Section 3.6.2 described the computational approach to finding the posterior distribution. Translating that section to the variable names we're using here,

1. Take a random sample $(\theta_1, \ldots, \theta_m)$ from the prior distribution Θ
2. For each sampled value, compute a weight

$$w_i = \frac{\Pr_X(x \mid \theta_i)}{\sum_j \Pr_X(x) \mid \theta_j}$$

   (Care should be taken to avoid underflow in cases where all the $\Pr_X$ values are small.)
3. Probabilities and expected values for the posterior distribution can be approximated by

$$\mathbb{P}(\Theta \in A \mid X = x) \approx \sum_{i=1^m} w_i 1_{\theta_i \in A}, \qquad \mathbb{E}\big(h(\Theta) \mid X = x\big) \approx \sum_{i=1}^{m} w_i h(\theta_i).$$

In the examples below, we'll illustrate the posterior distribution using a weighted histogram: choose a set of bins, and for each bin compute $\mathbb{P}(\Theta \in \text{bin} \mid X = x)$, and draw a bar of this height.

Exercise 6.1.
I have a coin, which might be biased. I toss it $n = 10$ times and get $x = 9$ heads. Let Θ be the probability of heads, an unknown parameter. Using the prior distribution $\Theta \sim U[0,1]$, find the posterior distribution, and depict it with a histogram.

First, take a sample of values from the prior distribution:

```
θ_samp = numpy.random.uniform(size=10000)
```

Second, write out the density function for the observed data. We'll assume the data is drawn from a $\mathrm{Binom}(n,\theta)$ *random variable, which has*

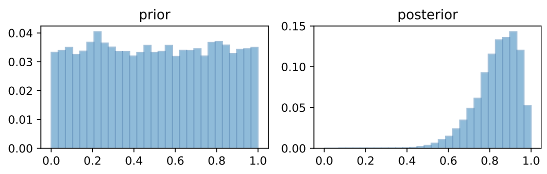
$$\Pr_X(x \mid \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}, \quad x \in \{0,1,\dots,n\}.$$

Use this to compute a weight for each sampled θ. There's no point including the constant term $\binom{n}{x}$ in prx *since it'll cancel out when we find weights.*

```
n,x = 10,9
prx = theta_samp**x * (1-theta_samp)**(n-x)
w = prx / numpy.sum(prx)
```

Finally, plot a weighted histogram to show the posterior distribution. Here is the output, together with a histogram of the prior distribution (weighted by $1/10000$ to make the two histograms comparable).

```
plt.hist(θ_samp, weights=w, bins=numpy.linspace(0,1,30))
```



∎

Exercise 6.2 (Multiple observations / avoiding underflow).
I toss a coin, which might be biased. I toss it $n = 10$ times and get $x_1 = 9$ heads. I do four more repeats of this experiment and get $[x_2, x_3, x_4, x_5] = [9, 10, 8, 10]$ heads. Using the prior distribution $\Theta \sim \mathrm{Uniform}[0,1]$, find the posterior distribution.

This is exactly the same as the previous exercise, except that prx *is different. The data density is*

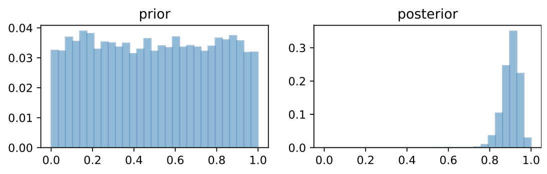
$$\begin{aligned} \Pr(x_1,\dots,x_5 \mid \theta) &= \Pr(x_1 \mid \theta) \times \cdots \Pr(x_5 \mid \theta) && \textit{assumimg successive trials are independent} \\ &= \textit{const} \times \theta^s (1-\theta)^{5n-s} && \textit{where } s = \sum_i x_i. \end{aligned}$$

To avoid underflow, we'll scale by a well-chosen factor before normalizing the weights.

```
θ_samp = numpy.random.uniform(size=10000)

n = 10
sx = numpy.sum([9,9,10,8,10])
logprx = sx * numpy.log(θ_samp) + (5*n-sx) * numpy.log(1-θ_samp)
w = numpy.exp(logprx - max(logprx))
w = w / numpy.sum(w)

plt.hist(θ_samp, weights=w, bins=numpy.linspace(0,1,30))
```



∎

Exercise 6.3 (Multiple unknowns).
We have a random sample $x = (x_1, \dots, x_n)$ drawn from $X \sim \text{Uniform}[a, a+b]$ where a and b are unknown parameters. Using $A \sim \text{Exp}(\lambda_0)$ and $B \sim \text{Exp}(\mu_0)$ as prior distributions, where $\lambda_0 = 0.2$ and $\mu_0 = 0.1$, find the posterior distribution of B, for the data

```
x = [2, 3, 2.1, 2.4, 3.14, 1.8]
```

There are two unknown parameters here. To apply Bayes's rule, our prior distribution needs to specify all unknown parameters, since $\text{Pr}_X(\textit{data} \mid \textit{params})$ *depends on them. Therefore we generate a random sample of* (A, B) *values. We'll assume* A *and* B *are independent. (If a question tells you about random variable distributions and doesn't explicitly say otherwise, you should take them to be independent, and state that you are doing so.)*

```
λ0,μ0 = 0.2, 0.1
samples = 100000
a_samp = numpy.random.exponential(scale=1/λ0, size=samples)
b_samp = numpy.random.exponential(scale=1/μ0, size=samples)
```

The density function for a single observation is

$$\text{Pr}_X(x \mid a, b) = \frac{1}{b} 1_{a \leq x \leq a+b}, \quad x \in \mathbb{R}.$$

(In any question where the bounds of a random variable are themselves being estimated, it's helpful to use indicator functions. That way, the algebra of indicator functions forces you to keep explicit track of bounds.) The density function for the entire dataset is

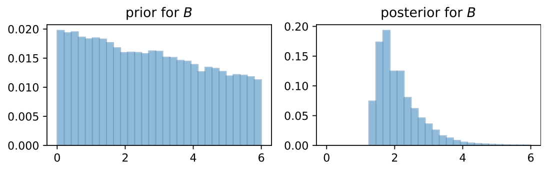
$$\begin{aligned}\text{Pr}(x_1, \dots, x_n \mid a, b) &= \frac{1}{b^n} 1_{a \leq x_1 \leq a+b} \times \cdots \times 1_{a \leq x_n \leq b} \\ &= \frac{1}{b^n} 1_{a \leq \min_i x_i} 1_{\max_i x_i \leq a+b}.\end{aligned}$$

For every prior sample (a_i, b_i)*, we need to compute* $\text{Pr}(x \mid a_i, b_i)$*. The code below uses numpy vectorized syntax: the* a_samp *and* b_samp *vectors line up, so we get a vector of* prx *values, one for each pair* (a_i, b_i).

```
x = [2, 3, 2.1, 2.4, 3.14, 1.8]
n = len(x)
minx,maxx = min(x), max(x)
prx = 1/b_samp**n * numpy.where((a_samp <= minx) & (maxx <= a_samp+b_samp), 1, 0)
w = prx / sum(prx)
```

This question asks for the posterior distribution of B*. Our code has actually produced a weighted posterior sample of* (A, B)*. If all we want to know about is* B*, just ignore* A*. (Formally, this is finding the marginal distribution of* B*—see section 3.4.) Here is the posterior histogram of* B*, together with the prior.*

```
plt.hist(b_samp, weights=w, bins=numpy.linspace(0,6,30))
```



∎

6.1.2. USING MATHS

Section 3.6.1 described how to apply Bayes's rule using maths. Translating into the variable names we're using here,

1. Write down the prior density $\Pr_\Theta(\theta)$. If there are multiple unknown parameters, this should be a joint density.
2. Write down the data density $\Pr_X(x \mid \theta)$.
3. Apply Bayes's rule to get the posterior density
$$\Pr_\Theta(\theta \mid X = x) = \kappa \Pr_\Theta(\theta) \Pr_X(x \mid \theta).$$
The constant κ can in principle be computed using the "probabilities sum to one" rule. Chances are, either the integral is unnecessary because we pattern-match and recognize the equation as the density of a standard random variable, in which case we should name that random variable; or the integral is intractable, in which case we should leave the answer with κ in.
4. If there are multiple unknowns and we're only interested on one of them, we should find the marginal density of the parameter we're interested in by integrating out the others. The others are called *nuisance parameters*.

marginalization: section 3.4 page 43

Mathematical solution to exercise 6.1:
The prior distribution is $\Theta \sim \text{Uniform}[0,1]$ *which has density* $\Pr_\Theta(\theta) = 1$*. For the data, as for our earlier solution,*
$$\Pr_X(x \mid \theta) = \textit{const} \times \theta^x (1-\theta)^{n-x}.$$
Applying Bayes's rule, the posterior density is
$$\Pr_\Theta(\theta \mid X = x) = \kappa \times 1 \times \theta^x (1-\theta)^{n-x}.$$
Hopefully this reminds you of the Beta distribution—if $Z \sim \text{Beta}(\alpha, \beta)$ *then it has density*
$$\Pr_Z(z) = \binom{\alpha+\beta-1}{\alpha-1} z^{\alpha-1}(1-z)^{\beta-1}$$
and so, using "densities sum to one", we conclude that $(\Theta \mid X = x) \sim \text{Beta}(x+1, n-x+1)$.
∎

Mathematical solution to exercise 6.3:
There are two unknown paramaters here. To apply Bayes's rule, our prior distribution needs to specify all unknown parameters, since $\Pr_X(\textit{data} \mid \textit{params})$ *depends on them. Therefore we need to write down a joint density for the pair* (A, B)*:*
$$\begin{aligned} \Pr_A(a) &= \lambda_0 e^{-\lambda_0 a} \\ \Pr_B(b) &= \mu_0 e^{-\mu_0 b} \\ \Pr_{A,B}(a,b) &= \lambda_0 \mu_0 e^{-\lambda_0 a - \mu_0 b} \quad \textit{assuming independence}. \end{aligned}$$
As before, the density function for the entire dataset is
$$\Pr(\textit{data} \mid a, b) = \frac{1}{b^n} 1_{a \le \min_i x_i} 1_{\max_i x_i \le b}.$$
Now use Bayes's rule to find the posterior density of (A, B) *given the data:*
$$\Pr_{A,B}(a, b \mid \textit{data}) = \kappa \frac{1}{b^n} e^{-\lambda_0 a - \mu_0 b} 1_{a \le \min_i x_i} 1_{a+b \ge \max_i x_i}.$$
We have gathered terms that don't involve (a, b) *into the constant* κ.

The question asks for the posterior distribution of B*. We've just worked out the posterior joint distribution of* (A, B)*, so the next step is to find the marginal for* B *by integrating out* A.
$$\begin{aligned} \Pr_B(b \mid \textit{data}) &= \int_a \Pr_{A,B}(a, b \mid \textit{data})\, da \\ &= \kappa \frac{e^{-\mu_0 b}}{b^n} \int_a e^{-\lambda_0 a} 1_{a \le \min_i x_i} 1_{a \ge \max_i x_i - b}\, da \\ &= \kappa \frac{e^{-\mu_0 b}}{\lambda_0 b^n} \left(e^{-\lambda_0 \max(\max_i x_i - b, 0)} - e^{-\lambda_0 \min_i x_i} \right) 1_{b \ge \max_i x_i - \min_i x_i}. \end{aligned}$$
The last step involved some heroic integration, and even so we're stuck—this isn't any standard distribution, and it looks like it'll be intractable to find the normalizing constant. (Problems with intractable integrals like this are where computational Bayes shows its strength!)
∎

Exercise 6.4 (Conjugate prior).
Here is an generalisation of exercise 6.1. That exercise used a uniform prior distribution, but now we will use a more expressive distribution.

I have a coin, which might be biased. I toss it n times and get x heads. Let θ be the probability of heads. Using the prior distribution $\Theta \sim \text{Beta}(\alpha_0, \beta_0)$ for given constants $\alpha_0 > 0$ and $\beta_0 > 0$, show that the posterior distribution is

$$(\Theta \mid x) \sim \text{Beta}(\alpha + x, \beta + n - x).$$

(The prior distribution was Beta, and we find that the posterior is another Beta but with different parameters. This is called a *conjugate prior*.)

* ✻ *

In well-chosen models, it shouldn't matter too much what the prior is. For example, in exercise 6.4, if n is very large then α and β have little influence. If we have too little data then the prior distribution will have a big impact on our answer. Bayesianism makes it easy to crank a handle and get out answers—but you should always stop and reflect whether your answers really reflect the data or whether they just reflect the assumptions you put in. This is usually easy to see from a maths formula, much harder to see in the output of a computation.

6.2. Readouts from posterior distributions

You're working on a dataset. You've invented a probability model to describe the data, with some unknown parameters that are crucial to your company's business. With just a few qualms you invent a reasonable prior distribution for the parameters, and you write code to sample from the posterior distribution.

Your boss excitedly and asks you "So, what values did you find for these parameters?"

"That's a category error" you reply sanctimoniously. "As a Bayesian, I consider parameters to be random variables. They are described with probability distributions. We can't pick out a single value to describe an entire distribution."

"Tell me a value, or you're fired," your boss says.

There are several standard summaries that Bayesians can report, to describe their posterior distributions. Suppose we're interested in the posterior distribution $(\Theta \mid \text{data})$, and either we've calculated this distribution with maths, or we've taken the computational approach and we have a collection of sample values θ_samp together with associated posterior weights w.

POSTERIOR POINT ESTIMATES *

If we really truly have to report a single value, here are some choices. If Θ is a simple $\mathbb{R}$-valued random variable, we could just report its mean or median. The posterior mean is

$$\text{posterior mean} = \mathbb{E}(\Theta \mid \text{data}) \approx \texttt{numpy.sum}(\theta\texttt{\_samp * w})$$

The posterior median is

$$\text{posterior median} = \hat{\theta} \text{ such that } \mathbb{P}(\Theta \leq \hat{\theta}) = 0.5.$$

```
i = numpy.argsort(θ_samp)
θ_samp,w = θ_samp[i], w[i]           # sort so θ is ascending
F = numpy.cumsum(w)                  # cumulative distribution
post_median = θ_samp[F<=0.5][-1]     # pick largest θ such that F ≤ 0.5
```

Another choice is to report the most likely θ. This is called the *maximum a posteriori* (MAP) estimate—a fancy name to give this estimate the facade of rigour.

$$\text{MAP estimate} = \arg\max_{\theta} \Pr_{\Theta}(\theta \mid \text{data}) \approx \theta\texttt{\_samp[numpy.argmax(w)]}$$

In some problems there is a natural *loss function* $L(\phi, \theta)$, which measures the price you pay if you report estimate ϕ and the true value is θ. Then you should report $\hat{\theta}$ to minimize the *expected posterior loss*,

$$\hat{\theta} = \arg\min_{\phi} \mathbb{E}\big(L(\phi, \Theta) \mid \text{data}\big).$$

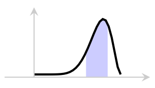
The expectation here is over values of Θ. How we implement this will depend on the problem—whether θ discrete or continuous, etc.

POSTERIOR CONFIDENCE INTERVALS

A 95% confidence interval for a random variable Y is an interval [lo, hi] such that $\mathbb{P}(Y \in [\text{lo}, \text{hi}]) = 0.95$. A good way to express our uncertainty about the unknown parameter is by giving a confidence interval for it, also called a *posterior confidence interval*: report an interval [lo, hi] such that

$$\mathbb{P}\big(\Theta \in [\text{lo}, \text{hi}] \;\big|\; \text{data}\big) = 0.95.$$

A common choice is to pick it so that



$$\mathbb{P}(\Theta < \text{lo}) = 0.025 \quad \text{and} \quad \mathbb{P}(\Theta > \text{hi}) = 0.025$$

though other choices are just as legitimate, for example the one-sided confidence interval $[-\infty, \text{hi}]$ with hi chosen so that $\mathbb{P}(\Theta > \text{hi}) = 0.05$. Computationally, if we have a collection of sample values θ_samp with weights w,

```
i = numpy.argsort(θ_samp)
θ_samp, w = θ_samp[i], w[i]
F = numpy.cumsum(w)
(lo,hi) = (θ_samp[F<0.025][−1], θ_samp[F>0.975][0])
```

POSTERIOR PREDICTIVE PROBABILITY *

When we're making inferences about an unknown parameter, only rarely do we actually care about the parameter itself—the parameter and indeed the entire probability model are fictions inside a data scientist's head. Rather, we want to make *predictions about outcomes* of future experiments.

For example: suppose we have a biased coin with unknown probability of heads θ, and we want to estimate the probability that the next two coin tosses will be heads, which is θ^2. We don't know θ, so we can't give an actual number for this probability. But suppose that from earlier experiments we've concluded that with probability 80% the coin is fair ($\theta = 1/2$), and with probability 20% the coin has heads on both sides ($\theta = 1$). Then a reasonable estimate for the probability of interest would be $0.8 \times (1/2)^2 + 0.2 \times 1^2 = 0.4$.

More generally, suppose we want to estimate the probability of some future event A, and that this probability depends on one or more unknown parameters; let's give this function a name, $\mathbb{P}(A \mid \theta) = h(\theta)$. As Bayesians we are representing our uncertainty about θ by treating it as a random variable $(\Theta \mid \text{data})$. So a reasonable estimate for the probability of A is the mean value of h applied to this random variable. We call this the *posterior predictive probability* of A. This whole process is written rather tersely as

$$\text{post.pred.prob.}(A) = \mathbb{E}\big[h(\Theta) \;\big|\; \text{data}\big] \equiv \mathbb{E}\big[\mathbb{P}(A \mid \Theta) \;\big|\; \text{data}\big]$$

Computationally it's easy to approximate, assuming we've managed to implement a function for h:

$$\text{post.pred.prob}(A) \approx \texttt{numpy.sum}\big(\texttt{h}(\theta\texttt{\_samp}) \texttt{ * w}\big).$$

Mathematically, if we have formulae for $h(\theta)$ and for the posterior density $\Pr_\Theta(\theta \mid \text{data})$, we can (if the calculus gods are friendly) calculate the posterior predictive probability by writing out the expectation as an integral:

$$\text{post.pred.prob}(A) = \mathbb{E}(h(\Theta) \mid \text{data}) = \int_\theta h(\theta) \Pr_\Theta(\theta \mid \text{data})\, d\theta.$$

Example 6.5. Suppose we have a biased coin, with unknown bias θ, and we've found a posterior distribution for this parameter using exercise 6.1. Find the predictive probability that the next two coin tosses will be heads.

The event {next two coin tosses are heads} has probability θ^2. *Exercise 6.1 gave us a sample of* θ_samp *together with posterior weights* w. *So the posterior predicitive probability that the next two coin tosses are heads is*

$$\texttt{numpy.sum}\big((\theta\texttt{\_samp ** 2) * w}\big).$$

∎

6.3. Bayesian model selection *

Your PhD supervisor has achieved eminence for the data she has collected and the probability model she has formulated to describe that data. A competing lab has just published a different model. Your supervisor asks you to figure which of the two models is a better fit for the data.

"That's a meaningless question" you reply sanctimoniously. "Speaking as a Bayesian, 'which model is better?' is uncertain, therefore like any other uncertainty it should be represented as a random variable. I can find a posterior distribution, if you tell me the prior."

"Just tell me which is better," your supervisor says. "You'll never make a scientific career if you can't draw conclusions."

A true Bayesian won't draw a conclusion about which model is better. But there is a nice way to write down the posterior distribution for the 'which model is better?' random variable, which is nearly as good.

Let the two models be $\mathcal{A}$ and $\mathcal{B}$, and let $m \in \{\mathcal{A}, \mathcal{B}\}$ denote which of the two models is correct. Treat this as a random variable: let $\Pr_M(\mathcal{A})$ be the prior probability that $\mathcal{A}$ is correct, and $\Pr_M(\mathcal{B}) = 1 - \Pr_M(\mathcal{A})$ be the prior probability that $\mathcal{B}$ is correct.

The two models may have unknown parameters. Write θ for the unknown parameters in model $\mathcal{A}$, and ϕ for the unknown parameters in $\mathcal{B}$, and assume we have set down prior distributions for them.

The first step of applying Bayes's rule is to write out the prior distribution. There are three unknowns here, M, Θ, and Φ, so we have to specify a prior joint distribution. We'll take an independent prior,

$$\Pr(m, \theta, \phi) = \Pr_M(m) \Pr_\Theta(\theta) \Pr_\Phi(\phi).$$

The second step is to write out the data density. Each of the two probability models, $\mathcal{A}$ and $\mathcal{B}$, proposes its own data density,

$$\Pr_X^{\mathcal{A}}(x \mid \theta) \quad \text{and} \quad \Pr_X^{\mathcal{B}}(x \mid \phi).$$

Let's combine these into one formula by

$$\Pr_X(x \mid \theta, \phi, m) = \begin{cases} \Pr_X^{\mathcal{A}}(x \mid \theta) & \text{if } m = \mathcal{A} \\ \Pr_X^{\mathcal{B}}(x \mid \phi) & \text{if } m = \mathcal{B}. \end{cases}$$

Now we can apply Bayes's rule:

$$\Pr_{M,\Theta,\Phi}(m, \theta, \phi \mid X = x) = \kappa \Pr_M(m) \Pr_\Theta(\theta) \Pr_\Phi(\phi) \Pr_X(x \mid \theta, \phi, m).$$

If we're trying to decide which model is better, we care about the posterior distribution of M, and Θ and Φ are nuisance parameters. So let's marginalize them out:

$$\Pr_M(m \mid X = x) = \int_{\theta,\phi} \Pr_{M,\Theta,\Phi}(m, \theta, \phi \mid X = x)\, d\theta\, d\phi$$

The random variable M takes only two values, $\mathcal{A}$ or $\mathcal{B}$, so let's write out $\Pr_M(m \mid X = x)$ explicitly for these two cases:

$$\begin{aligned}
\Pr_M(\mathcal{A} \mid X = x) &= \int_{\theta,\phi} \kappa \Pr_M(\mathcal{A}) \Pr_\Theta(\theta) \Pr_\Phi(\phi) \Pr^{\mathcal{A}}(x \mid \theta)\, d\theta d\phi \\
&= \kappa \Pr_M(\mathcal{A}) \int_\theta \Pr_\Theta(\theta) \Pr^{\mathcal{A}}(x \mid \theta)\, d\theta \\
&= \kappa \Pr_M(\mathcal{A})\, \text{ev}(\mathcal{A} \mid X = x) \\
\Pr_M(\mathcal{B} \mid X = x) &= \int_{\theta,\phi} \kappa \Pr_M(\mathcal{B}) \Pr_\Theta(\theta) \Pr_\Phi(\phi) \Pr^{\mathcal{B}}(x \mid \theta)\, d\theta d\phi \\
&= \kappa \Pr_M(\mathcal{B}) \int_\phi \Pr_\Phi(\phi) \Pr^{\mathcal{B}}(x \mid \phi)\, d\phi \\
&= \kappa \Pr_M(\mathcal{B})\, \text{ev}(\mathcal{B} \mid X = x)
\end{aligned}$$

The quantity $\text{ev}(m \mid X = x)$ is called the *evidence* for model m, and it's defined by these integrals above. We can find κ using the "densitites sum to one" rule, summing over two values, $\Pr_M(\mathcal{A} \mid X = x) + \Pr_M(\mathcal{B} \mid X = x) = 1$. This gives

$$\Pr_M(\mathcal{A} \mid X = x) = \frac{\Pr_M(\mathcal{A})\, \text{ev}(\mathcal{A} \mid X = x)}{\Pr_M(\mathcal{A})\, \text{ev}(\mathcal{A} \mid X = x) + \Pr_M(\mathcal{B})\, \text{ev}(\mathcal{B} \mid X = x)}$$

and similarly for $\Pr_M(\mathcal{B})$. In words, we start with a prior belief $\Pr_M(m)$ about which model is correct, and we update that belief in the light of the data, according to the evidence for each model. Evidence is a scale factor, used to reweight your prior belief about which model is correct. Don't try too hard to interpret the absolute value of evidence.

The true Bayesian would never say "Use Model $\mathcal{A}$ rather than Model $\mathcal{B}$", they would only say "In the light of the data, and given prior beliefs, here are the updated weights to use for each of the models." Their predictions about new observations would be based on posterior predictive probability, averaging over all unknown parameters including M. This is called *Bayesian model averaging*.

see page 69 for Bayesian posterior predictive probability

* ✻ *

Model selection includes choosing which prior to use. Bayesian methods can't help: strict Bayesianism insists that we invent a prior before we even look at the data. We can weasel out by declaring the parameters of the prior distribution to be *hyperparameters*, which is a fancy way of saying "parameter that I have no prior for", and use non-Bayesian model selection such as cross-validation to pick values for them.

Brian Ripley, an eminent data scientist, says "I think Bayesians are rarely Bayesian in their model choices"[19].

[19] Brian Ripley. *Selecting amongst large classes of models*. Lecture for a symposium in honour of John Nelder's 80th birthday, Imperial College. Mar. 2004. URL: `http://www.stats.ox.ac.uk/~ripley/Nelder80.pdf`.