Random Sampling

ISTA 410 / INFO 510 - Bayesian Modeling and Inference

University of Arizona School of Information

February 8, 2021

Outline

Last time:

- Normal model with known variance
- A little bit about priors

Next up:

- Approximating a posterior
- Random sampling
- Normal model, unknown variance

Non-conjugate priors

There's a problem on the homework that asks you to compute with a non-conjugate prior. Why?

- the conjugate prior's shape doesn't make sense for your modeling problem (we'll see this)
- you're working with a distribution or combination of distributions for which you can't find a conjugate prior

Non-conjugate priors

There's a problem on the homework that asks you to compute with a non-conjugate prior. Why?

- the conjugate prior's shape doesn't make sense for your modeling problem (we'll see this)
- you're working with a distribution or combination of distributions for which you can't find a conjugate prior

Most often we will have to use approximations:

- construct an approximate distribution that we can do exact calculations with
    - Laplace's normal approximation
    - expectation-maximization, expectation propagation, variational inference
- sample from the exact posterior and compute sample statistics to approximate parameters

Random sampling

Our goal is to go from a formula for $p(\theta|y)$ – which is straightforward to write down at least up to a normalizing constant – to various useful summaries such as histograms, expectations, and intervals.

- One approach: sample from $p(\theta|y)$
- If the posterior has the form of a known standard distribution, this is easy
- See HW0 problem 6: generate samples from $\mathrm{Beta}(4/3, 2)$ and make a histogram, quantiles, mean, SD

Random sampling

What if our posterior isn't a standard distribution? (cf. HW1 problem 4)

- What does a random number generator produce?
- How can we use that to produce what we need?

Uniform random numbers on $[0, 1]$

The starting point for any random sampling procedure is the PRNG (pseudo-random number generator), which generates a sequence of numbers from $[0, 1]$ that has the statistical properties of a Uniform distribution

- A lot of work has gone into producing good PRNGs
- In the old days, hardware and software limitations led to PRNGs with flaws such as short periodicity, autocorrelation
- These days, they're pretty good, and pseudo-randomness is generally enough for statistical work

Transforming a uniform random number

The simplest way to generate a random number from an arbitrary distribution is to use the CDF:

- Recall: CDF $=$ *cumulative distribution function*
- CDF for a random variable X is $F(x) = \Pr(X \leq x)$
- If we can compute F^{-1}, then:
    - Generate values z_i from the PRNG
    - Then $F^{-1}(z_i)$ has the desired probability distribution

Example 1

Example 1: how do you generate a normal random variate? Work backwards:

- First, remember that if $Z \sim N(0,1)$ then $\sigma Z + \mu \sim N(\mu, \sigma^2)$, so it's enough to generate standard normals
- Have the "error function":

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

- Inverse CDF can be written in terms of the error function as

$$F^{-1}(p) = \sqrt{2}\mathrm{erf}^{-1}(2p-1)$$

  and erf^{-1} is implemented in SciPy

Example 1

So we can do something like:

```
from scipy.special import erfinv
u = np.random.rand()
z = np.sqrt(2) * erfinv(2*u - 1)
```

and confirm that the generated z values have a standard normal distribution.

Example 1

So we can do something like:

```
from scipy.special import erfinv
u = np.random.rand()
z = np.sqrt(2) * erfinv(2*u - 1)
```

and confirm that the generated z values have a standard normal distribution.

Clearly we benefited here from the fact that this *is* a well-known distribution and the inverse CDF was easily available (obviously not a coincidence). What if it's not?

Building a CDF “by hand”

Some simple cases will allow us to find the inverse CDF by calculus, in which case you can just do that. But what if you have something less nice?

One option:

- Evaluate the PDF on a grid and normalize it numerically
- Numerically integrate to get a grid approximation of the CDF
- Compute the inverse CDF using this approximation

The last step, and numerical inversion of functions in general, is a broad field and we’ll just use a crude approach for now.

Building a CDF "by hand"

Let's say we're running the following model:

$$\begin{aligned} y_i &\sim t_3(\theta, 2) \\ \theta &\sim \mathrm{Normal}(10, 5) \end{aligned}$$

t_3 is the Student's t-distribution with 3 degrees of freedom.

This might be used if the observations are known to be more widely dispersed than a Normal likelihood allows. But then the Normal prior on θ isn't conjugate to the likelihood.

Say we make two observations $y_0 = 8.1, y_1 = 13.5$.

Building a CDF “by hand”

Bayes’ theorem tells us:

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

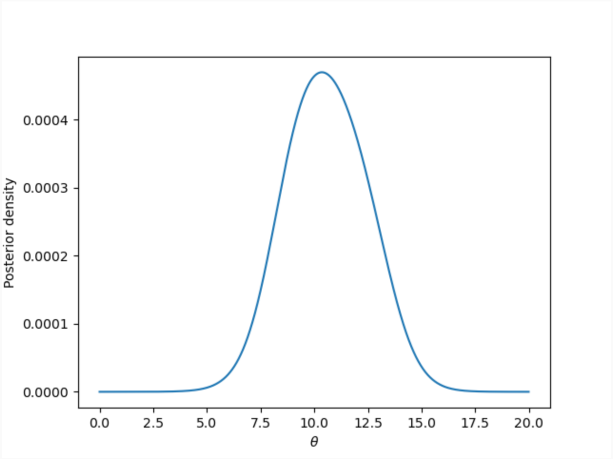
So, we can calculate the posterior density by evaluating:

```
from scipy.stats import norm, t
p = norm.pdf(theta, 10, 5) * t.pdf(8.1, 3, theta, 2) * \
    t.pdf(13.5, 3, theta, 2)
```

(You can also write down analytic formulas, but the PDF for the *t* distribution is a bit messy so I’ve left it off the slide)

Building a CDF “by hand”

Plotting the posterior PDF using the code before:



Normalize

Problem 1: not normalized

If we sum the y values, multiplied by the grid spacing, we get the Riemann sum for the integral (you could get fancier and use a trapezoid method or even Simpson's rule). So we can then scale to the correct value.

Once we have the normalized PDF evaluated on the grid, we can build the CDF.

Build the CDF values

The CDF is, by definition, just the integral of the PDF, so we can again use the Riemann sum approach to get the CDF's y values.

Then, here is a crude approach:

1. Generate a uniform random value u
2. Subtract u from the CDF and locate the minimum absolute value
3. This gives the grid point whose y-value is closest to u

Let's go test all of this...

Inference from the posterior

Now that we have a sample from the posterior, we can compute:

- expectations
- quantiles
- density estimates (e.g. kernel smoothing)

Normal model, unknown variance

Introduction to multi-parameter models

The known-variance assumption isn't necessarily particularly realistic. So instead, we can allow σ^2 to be an unknown parameter in our model.

New model (simple, improper priors):

$$\begin{aligned} y_i &\sim \mathrm{Normal}(\mu, \sigma^2) \\ p(\mu, \sigma^2) &\propto (\sigma^2)^{-1} \end{aligned}$$

This improper prior is derived from applying a uniform prior to $\mu, \log \sigma^2$ (more on this next time)

The joint posterior

As before we can get the joint posterior by simply multiplying this prior by the likelihood $N(\mu, \sigma^2)$ for our data $y_1, \ldots, y_n$.

$$
\begin{aligned}
p(\mu, \sigma^2 | y) &\propto \sigma^{-n-2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2\right) \\
&= \sigma^{-n-2} \exp\left(-\frac{1}{2\sigma^2} [(n-1)s^2 + n(\bar{y} - \mu)]\right)
\end{aligned}
$$

where $\bar{y}$ is the sample mean, s^2 the sample variance.

Interpreting the posterior

The posterior here shows that μ, σ are not independent.

In order to interpret this, it's easier to think about the distribution of μ conditional on σ^2. This is simple:

$$\mu | \sigma^2 \sim \mathrm{Normal}(\bar{y}, \sigma^2/n)$$

which is the same as the known variance case. To factor the joint posterior, then, we need $p(\sigma^2|y)$.

Marginal distribution of σ^2

The marginal posterior distribution of σ^2 is obtained by averaging the joint posterior over μ:

$$p(\sigma^2|y) \propto \int_{-\infty}^{\infty} \sigma^{-n-2} \exp\left(-\frac{1}{2\sigma^2}[(n-1)s^2 + n(\bar{y}-\mu)]\right) d\mu$$

Looks gnarly, but it's not – the exponential factors into the part dependent on s and the part dependent on μ. The part dependent on μ is a Gaussian integral, proportional to σ^{-1}. So, we get...

Marginal distribution of σ^2

The marginal posterior distribution of σ^2 is

$$p(\sigma^2|y) \propto \sigma^{n-3} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right)$$

which is a standard density on σ^2:

$$\sigma^2|y \sim \mathrm{Inv}{-}\chi^2(n-1, s^2)$$

Sampling from the joint posterior

It is easy to sample from the joint posterior distribution:

- draw a value of σ^2 from the posterior for σ^2 (a scaled inverse chi-squared distribution)
- draw a value of μ from the conditional posterior given your value of σ^2 (a normal distribution)

Sampling from the joint posterior

Let's draw and interpret some samples from the posterior for the basketball score variable.

Now, we take y_i to be the total score in a game, and specify the model:

$$y_i \sim \mathrm{Normal}(\mu, \sigma^2)$$
$$p(\mu, \sigma^2) \propto (\sigma^2)^{-1}$$

We'll start by plotting points in the (μ, σ) plane, then the normal distributions they represent

Sampling from the joint posterior

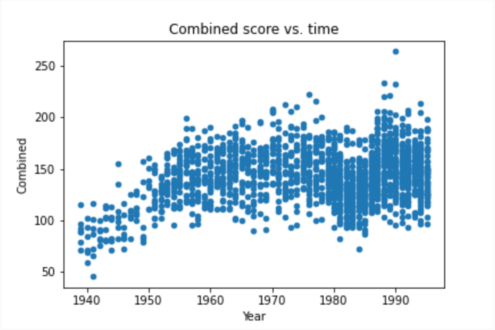
For the following plots:

- Restricted to year ≥ 1988 for consistency in the target distribution
- Subsampled: $n = 10, 40, 100$ to see how sample size affects uncertainty

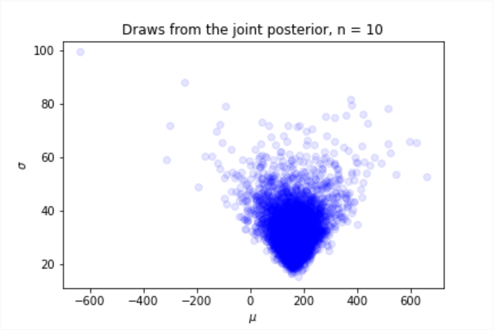
Why limit to data after 1988?

Time series

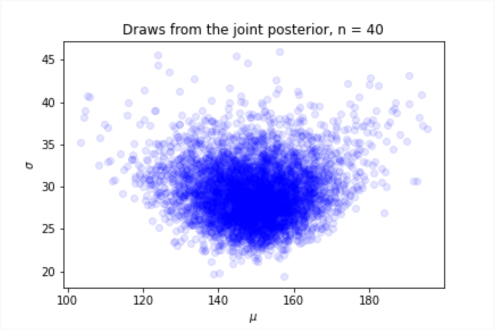
Plotting all of the combined scores against time:



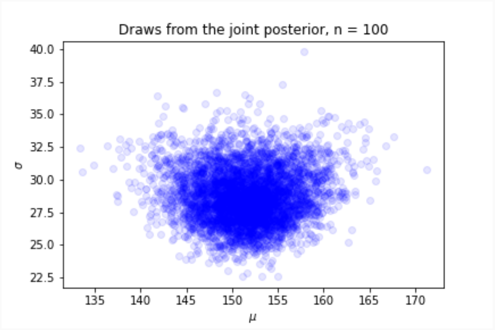
Sampling from the joint posterior



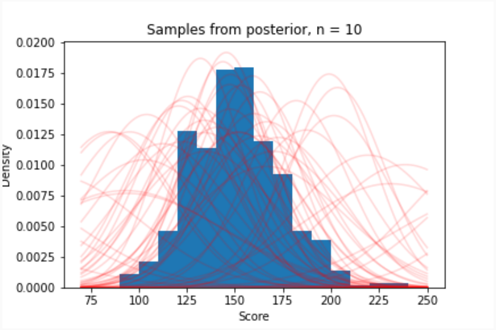
Sampling from the joint posterior



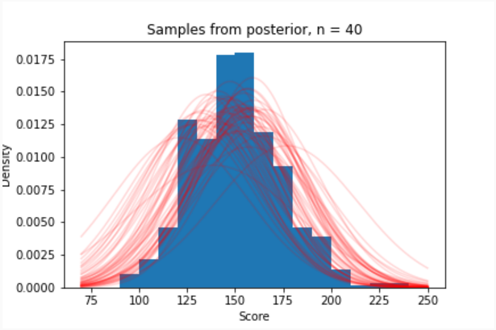
Sampling from the joint posterior



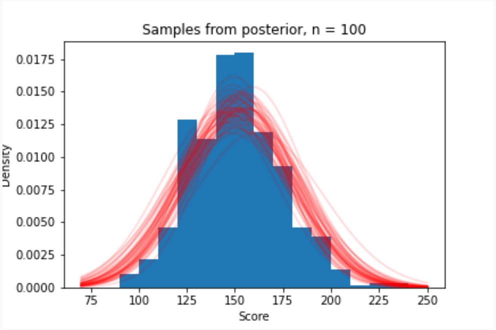
Visualizing uncertainty



Visualizing uncertainty



Visualizing uncertainty


Samples from posterior, n = 100
Density
Score