EPSE 581C: Bayesian Methods

Ed Kroc

University of British Columbia

ed.kroc@ubc.ca

October 28, 2019

Regression model building and validation

"All models are wrong, but some are useful." - George Box

- What constitutes a "useful" model depends on your inferential goals, but one should always:
  - Examine your model **residuals**
  - Consider if important covariates could be missing
  - Consider interactions and higher order terms, if suggested by the residuals
  - Default to the simplest model that "works"
  - In the Bayesian approach, consider how sensitive your model is to your specification of the priors.

Regression model building and validation

CAUTION:

- Just because you see no signs of model misfit, that does *not* mean that your model is correctly specified.
- Many ways for model misfit to hide; residuals diagnostics are the best way to search, but far from perfect.
- In particular, you can only try to fix the model with the variables that you have measured; in general, missing important covariates will necessarily compromise your model.

Bayesian prediction

- Just as in the classical frequentist framework, we may often want to *predict* a new value for the random phenomenon in question, rather than just infer global properties about that random variable (e.g. its theoretical mean).
- Classically, in the frequentist paradigm, we construct *prediction intervals*: similar to confidence intervals, but must account for extra uncertainty since we want to predict a *new observation*, not just an *average* observation.

Posterior predictive distributions

- The *posterior predictive distribution* is the distribution of a *new* data point, conditional on the already observed data (and our model):

$$
\begin{aligned}
f(\tilde{y} \mid \boldsymbol{y}) &= \frac{f(\tilde{y}, \boldsymbol{y})}{f(\boldsymbol{y})} \\
&= \frac{\int f(\tilde{y}, \boldsymbol{y}, \theta)\ d\theta}{f(\boldsymbol{y})} \\
&= \frac{\int f(\tilde{y}, \boldsymbol{y} \mid \theta)\pi(\theta)\ d\theta}{f(\boldsymbol{y})} \\
&= \frac{\int f(\tilde{y} \mid \theta) \cdot f(\boldsymbol{y} \mid \theta)\pi(\theta) d\theta}{f(\boldsymbol{y})} \\
&= \int f(\tilde{y} \mid \theta) \cdot f(\theta \mid \boldsymbol{y})\ d\theta
\end{aligned}
$$

Posterior predictive distributions

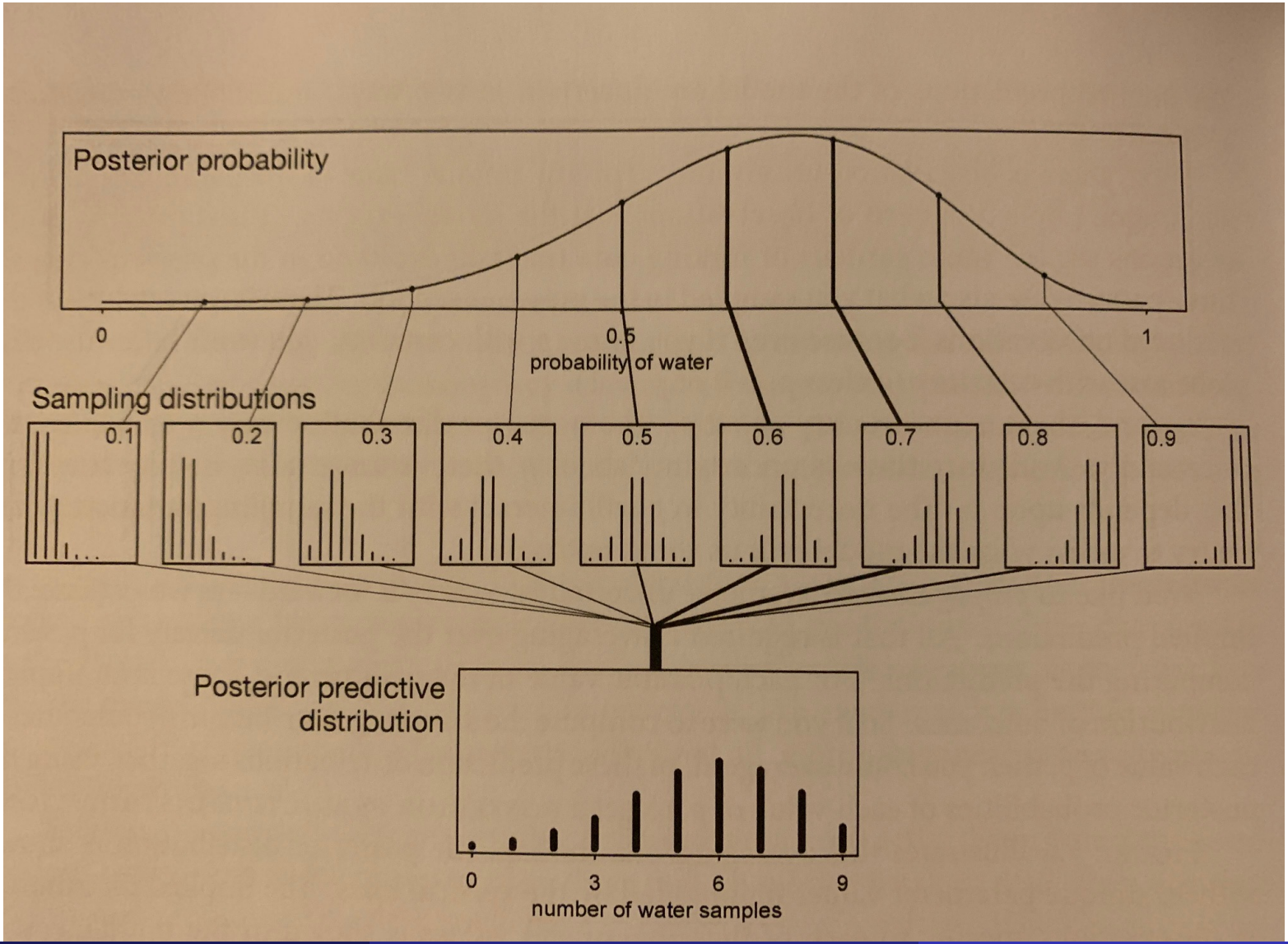
- The *posterior predictive distribution* is the distribution of a *new* data point, conditional on the already observed data (and our model):

$$f(\tilde{y} \mid \boldsymbol{y}) = \int f(\tilde{y} \mid \theta) \cdot f(\theta \mid \boldsymbol{y}) \, d\theta$$

- Notice: this integrates the *likelihood* of observing the new data point, given a particular value for the model's parameter(s), weighted by the *posterior probability* of those particular parameter value(s).
- So: use the data to tell us which models are more or less appropriate (via posterior probability), then consider how likely it would be that our new data point came from *any one* of these models: take a weighted average of all possibilities (i.e. integrate).
- Yields predictions that account for sampling and model uncertainty.

Posterior predictive distributions

From McElreath, p. 66:



Computational issues

All of Bayesian inference is based on the *posterior distribution*, but for complex, real-world situations, the posterior is often analytically untractable. Several potential problems:

- How to calculate the *normalizing factor* in the denominator of Bayes' theorem?
- How to calculate the *expectation* (e.g. mean, standard deviation) of a complicated posterior density?
- How to calculate the *mode* of anything?
- How to do all this flexibly; i.e. without having to restrict ourselves to very special classes of data and priors (e.g. conjugate priors)?

Computational issues

There are *many* techniques for addressing these issues; we will only briefly survey some of the most common/important:

- Numerical integration (deterministic)
- EM algorithm (deterministic)
- Monte Carlo sampling (stochastic)
- Markov chain Monte Carlo methods (stochastic)
  - Metropolis-Hastings algorithm
  - Gibbs sampler

Numerical integration

Numerical integration techniques actually pre-date integration itself!
I.e. how do we approximate an area under a curve?

- (finite) Riemann sum
- Quadratic approximation
- Laplace approximation
- etc.

All based on the idea that you can approximate an area of a complicated region by splitting it up into a bunch of simple regions and then summing up all their individual areas.

The EM algorithm

- In statistics, one of the most commonly employed methods to calculate (approximate) the *maximum* (or minimum) of a function is the *expectation-maximization (EM) algorithm*
- The algorithm iterates these two steps to find the (local) maximum (minimum) of a probability density function:

(1) Calculate the *expectation* of the posterior predictive distribution, call it

$$z_1 = \mathbb{E}(\tilde{y} \mid \boldsymbol{y}) = \int \tilde{y} \cdot f(\tilde{y} \mid \boldsymbol{y}) \, d\tilde{y}$$

(2) Augment the dataset to $\{\boldsymbol{y}, z_1\}$ and then find the new posterior density for this augmented dataset. This new posterior has a corresponding posterior predictive density, so we can return to step (1) and define z_2 as before. Iterating this procedure, z_n will converge to the *maximum* (mode) of the original posterior density.

Monte Carlo sampling

- Rather than the deterministic methods already discussed, there is a whole class of *stochastic* (i.e. random sampling based) methods for calculation/approximation.
- Most of these techniques fall under the general term of *Monte Carlo sampling*, loosely defined as any technique where you use a theoretical probability sample to approximate a target quantity.
- For example, if we want to calculate the *posterior mean*, then we can simply generate a bunch of random draws from the posterior distribution and compute their sample mean. Do this for a large enough sample size, and the approximation will be very good. (Why?)
- Could approximate *posterior standard deviation* the same way, using the sample standard deviation of a bunch of random draws from the posterior.

Markov chain Monte Carlo (MCMC) methods

- Markov chain Monte Carlo (MCMC) methods simply extend this basic idea by relaxing the requirement that each draw comes from the same distribution.
- For example, it is often very difficult to work directly with the posterior distribution! So how could we "sample" from it?
- Instead, we generate a random sample from a *sequence* of simpler probability distributions, chosen so that in the long run, these distributions converge to the target (posterior) distribution. This sequence of distributions forms a *Markov chain*.

Markov chain Monte Carlo (MCMC) methods

You could easily spend an entire term talking about these mathematical ideas (see MATH 303, 545), but just to fix terminology:

- A *Markov chain* is a random sequence (stochastic process), $X_1, X_2, \ldots$, that obeys the *Markov property*:

$$\Pr(X_{t+1} \mid X_t, \ldots, X_2, X_1) = \Pr(X_{t+1} \mid X_t), \text{ for all time points } t.$$

- Put another way, given the present value of the Markov chain, its future value is independent of the past.

Markov chain Monte Carlo (MCMC) methods

- MCMC methods all setup a sequence of random variables $X_1, X_2, \ldots$ obeying the Markov property so that this random sequence converges to what we want (e.g. the posterior distribution).
- Thus, we can "sample from the posterior" by sampling sequentially from the Markov chain.
- Eventually, our samples will be close enough to the target distribution, so we can approximate whatever we want by the usual sample analogues (i.e. we can use Monte Carlo approximation).‘
- Important question: *How long is long enough?*
- Important note: Regardless of that answer, notice that we would *never* want to use the early sample draws from an MCMC method, as there is not enough time for the Markov chain to converge. This generates what is called a *burn-in* period/window.

Metropolis-Hastings algorithm

- The Metropolis-Hastings algorithm is one of the most common MCMC methods around.
- Let $\pi(x)$ be our target density (e.g. joint posterior density), and let x_0 be an arbitrary starting value. Iteratively, proceed as follows:
  - Simulate a candidate value y from the distribution given by $\Pr(X_n \mid X_{n-1} = x_{n-1})$.
  - Define the *acceptance ratio*

$$\alpha(y \mid x_{n-1}) = \min\left\{\frac{\pi(y)\Pr(X_n = x_{n-1} \mid X_{n-1} = y)}{\pi(x_{n-1})\Pr(X_n = y \mid X_{n-1} = x_{n-1})},\ 1\right\}$$

  - Simulate $u \sim \mathit{Unif}(0,1)$. If $u \leqslant \alpha(y \mid x_{n-1})$, then the next state of the Markov chain is equal to y (i.e. we set $X_n = y$); otherwise, the Markov chain stays in the same state (i.e. we set $X_n = x_{n-1}$).

Metropolis-Hastings algorithm

- The MH algorithm is extremely flexible and can be used to approximately simulate data from all kinds of complicated distribution using only simple probability distributions.
- Notice a key feature of MH: although the acceptance ratio α depends on the target (posterior) density), it does *not* depend on the *normalizing constant*, because it is a ratio of the target density at two different values. Greatly simplifies computation!
  - Use MH to simulate normal data using a uniform distribution: `https://bookdown.org/rdpeng/advstatcomp/metropolis-hastings.html`
  - Use MH to simulate exponential data using a normal distribution: `https://stephens999.github.io/fiveMinuteStats/MH-examples1.html`

Gibbs sampler

Gibbs sampling is a particularly effecient kind of MH procedure, especially for high dimensional problems:

- Given a target density $\pi(y_1, \ldots, y_p)$, define the *full conditional* densities

$$\pi(y_i \mid y_{-i}) = \pi(y_i \mid y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_p).$$

- Our Markov chain will now sample from each of these full conditional densities cyclically at each time step.
- Also, rather than worrying about the accept/reject stage, we will always accept and move onto the next full conditional density.

Can be shown to very computationally efficient. Lots of ways to make it even better too.

Weak Law of Large Numbers

But why the heck do all these stochastic approximations even work?

Weak Law of Large Numbers

Let $X_1, X_2, \ldots$ be a sequence of independent and identically distributed (i.i.d.) random variables, each having finite mean $\mathbb{E}(X_i) = \mu$ and finite variance $\text{Var}(X_i) = \sigma^2$. Then for any $\varepsilon > 0$,

$$\lim_{n \to \infty} \Pr\left(\left| \frac{X_1 + \cdots + X_n}{n} - \mu \right| > \varepsilon \right) = 0.$$

We say that the sample mean of the X_i's *converge in probability*, or *converge in measure*, to the population (true) mean of X_i. That is, the sample averages, $\overline{X}_n$, converge in probability to the expectation.

In statistics, we say that $\overline{X}_n$ is a *consistent estimator* of μ.

Weak Law of Large Numbers

- This law holds in greater generality. In particular, it still holds even if $\text{Var}(X_i) = \sigma_i^2$ are all different, as long as $\sigma_i^2 \leqslant C$ for all i and some C finite.
- Note that the WLLN does *not* apply to random variables with infinite mean or variance, like Cauchy random variables.
- The WLLN law justifies the intuition that after a large number of i.i.d experiments, the sample average of the r.v. should be close, to the true expectation.
- The WLLN (or its generalization for Markov chains) guarantees that our sample estimates will converge to the target value, given enough time.

Visualizing the WLLN

http://digitalfirst.bfwpub.com/stats_applet/stats_applet_10_prob.html

Weak Law of Large Numbers

- The WLLN does *not* say that $|\overline{X}_n - \mu| > \varepsilon$ cannot happen upon further experimentation. In fact, the WLLN leaves open the possibility that we observe such a discrepancy infinitely often upon further experimentation.
- Thankfully, there is a Strong Law of Large Numbers that says this (usually) cannot happen. The SLLN says that for any $\varepsilon > 0$ and for all n large enough, $\Pr(|\overline{X}_n - \mu| > \varepsilon) = 0$.
- For Markov chains, this is the difference between *weak* and *strong mixing*. Lots of fascinating and important math here, and the real reason there are so many different approximation techniques that have been developed in practice. Not all techniques work equally well all of the time.

Weak vs. Strong Law of Large Numbers

Weak Law of Large Numbers

Let $X_1, X_2, \ldots$ be a sequence of independent and identically distributed (i.i.d.) random variables, each having finite mean $\mathbb{E}(X_i) = \mu$ and finite variance $\text{Var}(X_i) = \sigma^2$. Let $\bar{X}_n$ denote the sample average of $X_1, \ldots, X_n$. Then for any $\varepsilon > 0$,

$$\lim_{n \to \infty} \Pr\left(\left|\bar{X}_n - \mu\right| > \varepsilon\right) = 0.$$

Strong Law of Large Numbers

Under the same conditions as the WLLN, for any $\varepsilon > 0$,

$$\Pr\left(\lim_{n \to \infty} \left|\bar{X}_n - \mu\right| > \varepsilon\right) = 0.$$

We say that $\bar{X}$ converges *almost surely* or *almost everywhere* to μ.

Weak vs. Strong Law of Large Numbers

- There are certain random variables for which the WLLN holds, but the SLLN fails. This is rare, but explains the terminology "weak" vs. "strong".
- The proof of the SLLN requires notions from measure theory.
- For most well-behaved random variables, an i.i.d. sample will obey both the WLLN and the SLLN.
- Note: the Central Limit Theorem is often used to create measures of uncertainty about our sample estimates (e.g. confidence intervals), but it is the Law of Large Numbers that assures us that our sample estimates are "good" to begin with. In general, we have the following logical implications:

$$\text{CLT} \longrightarrow \text{SLLN} \longrightarrow \text{WLLN}$$