Bayesian Inference

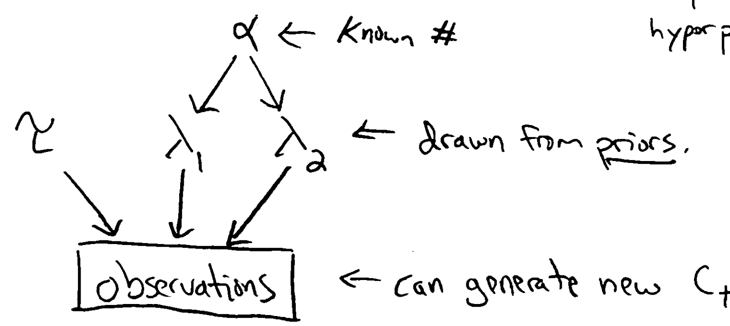
A statistical model for text-message data. (con't)

Recall: $C_t \equiv$ # texts received on day T.

$$C_t \sim \mathrm{Pois}(\lambda_t), \quad \lambda_t = \begin{cases} \lambda_1 & t < \tau \\ \lambda_2 & t \geq \tau \end{cases}$$

$$\tau \sim \mathrm{DiscreteUniform}(0, T), \quad \lambda_1, \lambda_2 \sim \mathrm{Exp}(\underset{\substack{\uparrow \\ \text{hyperparameter}}}{\alpha}), \quad \frac{1}{\alpha} = E[C_t]$$

flow chart:



We want to understand how the model relates to the data. What are typical values of $\lambda_1, \lambda_2, \tau$? Does $\lambda_1 = \lambda_2$?

$\Rightarrow$ Answer these w/ Bayesian Inference!

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})\, P(\text{model})}{P(\text{data})}$$

(Last time, we explored how synthetic data transformed $P(\text{model})$ into $P(\text{model}|\text{data})$, prior $\rightarrow$ posterior.)

- We want to understand the posterior as this gives us answers to our questions!

$$\Rightarrow P(\text{model}|\text{data}) = P(\lambda_1, \lambda_2, \tau | \{C_t\}) \propto \underbrace{\prod_{t=0}^{T} P(C_t | \lambda_1, \lambda_2, \tau) \cdot (\text{prior})}_{= f(\text{model}|\text{data})}$$

Posterior is prop. to. $f(M|D)$ because $P(\text{data}) = P(\{C_t\})$ is fixed, it's some unknown (unknowable?) constant!

$\Rightarrow$ How do we draw statistical models distributed according to the posterior $P(\text{model}|\text{data})$ when we only know $f(\text{model}|\text{data})$ and only at very few values?

Remember, the point of Bayesian Inference is to look at the posterior distribution. The spread of probability over the space of models encapsulates our knowledge and uncertainty about those models as told to us by the data

Often the space of models is enormous: 1000 parameters = 1000 dimension space

- A simple expression for the posterior is often not available.

- What "Bayesians" will do, instead of looking at an equation, is draw large numbers of samples (ie, models) from the posterior and look at the spread of the samples.

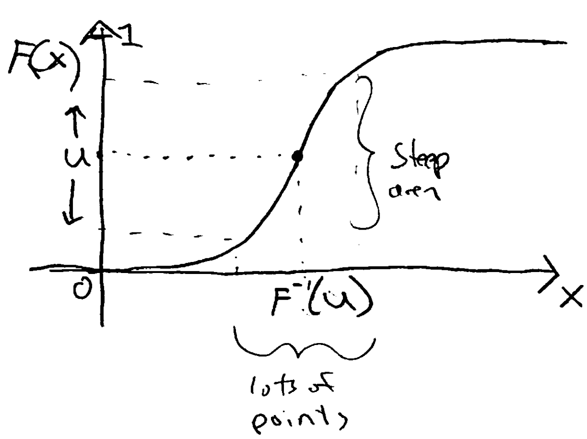
$\Rightarrow$ How do we draw samples from the posterior when we don't know the distribution

Sampling from a distribution

The computer gives us a pseudo random number generate for $u \sim U(0,1)$. There are many ways to transform u into some $X \sim Pr(\theta)$, e.g. $X \sim N(\mu, \sigma^2)$ (normal distribution).

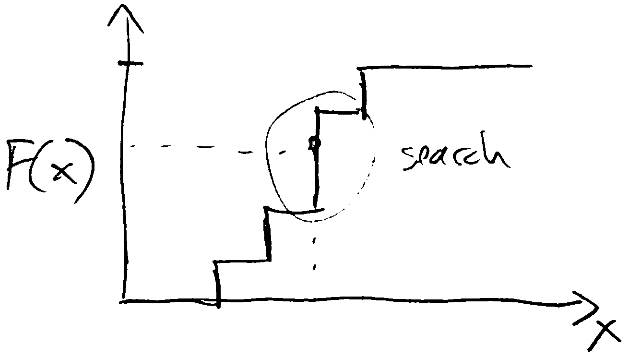
A nice, general-purpose method: Inverse CDF transform

CDF: $Pr(X < x) = F(x)$



$Pr(F^{-1}(u) \leq x)$ F is monotonic, non-negative
$= Pr(u \leq F(x))$
$= F(x)$ if u is uniformly distribution since $Pr(U \leq x) = x$ for uniform distr.

Even if $F^{-1}(u)$ is unknown can still draw values empirically (ECDF) using a bisection method. since ECDF is sorted



Can even work if ECDF is not normalized but you must know what it sums to!

$\Rightarrow$ Can't use this nice technique to sample from the posterior!

MCMC Markov Chain Monte Carlo $\Rightarrow$ the answer.

Intuition: We are going to wander around $f(M|D) = f(\theta)$ randomly, checking the height (value) of f as we do, and spending more time near higher areas of f and less time near lower areas.

As we wander we will keep a history of our positions $\theta_1 \to \theta_2 \to \cdots \to \theta_t \to \cdots \to \theta_T$ ("trace"). This, it turns out, will become our samples from the posterior.

Three things.

1. How do we do this?
2. How do we know it works? $\Big\{$ "Markov chain" is aperiodic, positive recurrent, irreducible $\Rightarrow$ it is ergodic and has a stationary distribution
3. Let's see it in action!

1. How $\Rightarrow$ algorithm.

1. Choose a random initial position θ_0 $(= \lambda_1^{(0)}, \lambda_2^{(0)}, \tau^{(0)})$
2. Pick a random nearby position θ' (jump kernel)
3. Compute $r = \dfrac{f(\theta')}{f(\theta_0)}$. Notice that $\dfrac{f(\theta')}{f(\theta_0)} = \dfrac{P(\theta')}{P(\theta_0)}$ (suppressing the "| data" notation)
4. If $r > 1$, then set $\theta_1 = \theta'$ (jump to nearby location)
   If not: set $\theta_1 = \begin{cases} \theta' & \text{w/ prob. } r \\ \theta_0 & \text{w/ prob } 1-r \end{cases}$ $\leftarrow$ jump any way.
5. Repeat from 2 using the new θ as the initial position until T times.

As we keep moving we are biased in favor of higher probability regions, and, unless something bad is happening, our samples will eventually follow the posterior distribution (convergence in distribution.)

Problems 1. Our initial location will likely be very far from typical regions of the posterior.

This is known as the "burn in" phase and typically we will throw out the beginning of our sample (but how much to throw away?).

2. The jump kernel introduces serial correlations, meaning sample i is correlated w/ other "nearby" samples $i-d \leq j \leq i+d$ for some (hopefully small) value of d.

We can fix this with thinning, where we keep only every n^{th} sample. (like every other sample)

⇒ The trace may still have problems, like it may not be mixing well and become trapped in part of the posterior landscape.

[See it in action → computer]

Now we have the posteriors $\left.\begin{array}{l}\Pr(\lambda_1 \mid \text{data}) \\ \Pr(\lambda_2 \mid \text{data}) \\ \Pr(\tau \mid \text{data})\end{array}\right\}$ "empirical" distributions from our 30k sample trace.

We can combine our samples into an average:

$$\overline{C}_t = \frac{1}{N_{trace}} \sum_{s=1}^{N_{trace}} \left(\lambda_1^{(s)} \underbrace{[t < \tau^{(s)}]} + \lambda_2^{(s)} [t \geq \tau^{(s)}] \right)$$

↑ size of trace = 30k

"Iverson Bracket": $[P] = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{if } P \text{ is false} \end{cases}$