Approximate Inference for Bayesian Models: Markov Chain Monte Carlo

Data 102 Fall 2022

Lecture 9

Weekly Outline

- So far: Bayesian models, conjugate priors, graphical models, rejection sampling

- **Today: Markov Chain Monte Carlo**
  - Markov Chains
  - Metropolis-Hastings
  - Gibbs sampling

- Next time: Prediction (Bayesian and frequentist)

Recap: Bayesian Inference

Approximate Bayesian Inference

θ: unknown (random) state of the world

x: data you observe

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} = \frac{p(x|\theta)\,p(\theta)}{\int_\theta p(x|\theta)p(\theta)d\theta}$$

posterior: what we want! our "belief" about θ after observing x.

likelihood

prior

we specify these when choosing our model.

Intractable if θ is high-dimensional

Recap: Sampling for Approximate Inference

- Computing exact posteriors can be hard (and often impossible)
- Idea: use samples to approximate
  - e.g., instead of computing $E_{\theta|x}[\theta|x]$, use mean of samples
  - e.g., instead of $var_{\theta|x}(\theta)$, use variance of samples
- Certain distributions are easy to sample from
  - Well-studied distributions: uniform, normal, beta, etc.
  - `np.random.beta(...)` or `scipy.stats.beta.rvs(...)`
- Getting samples for arbitrary posteriors is hard (but possible)
- Sampling methods
  - Rejection sampling
  - Markov Chain Monte Carlo (Metropolis-Hastings and Gibbs sampling)

Recap: Rejection Sampling

- Goal: generate samples from an unnormalized target distribution
  - Usually represents the posterior distribution over parameters that we care about
- Start by defining a proposal distribution
  - Should be easy to sample from (uniform, normal, etc.)
  - Must be ≥ the target distribution for all values of the parameter(s)
- Generate samples from the proposal
- Compute the acceptance probability as target ÷ proposal
- For each proposed sample, randomly make an accept/reject decision
  - In practice: generate a uniform random variable and seeing whether it's below the acceptance probability
- Easy to implement in most cases, but usually very inefficient

Markov Chains

- Sequence of random variables (usually indexed by “time”)
- Possible values are called “states” of the chain
- Each random variable only depends on the previous one (Markov property)
- The transition probabilities between states are known and stay the same over time

Markov Chain Key Ideas

- **Steady-state distribution**: for large values of t, how likely is z_t to be in each state?
  - Depends on transition probabilities
  - Only defined for aperiodic Markov chains where all states are reachable

- **Mixing time**: how long does it take the Markov chain to reach the steady state distribution?
  - Depends on how "well-connected" the transition probability graph is

Markov Chain Monte Carlo

- Observation: rejection sampling is wasteful because we reject many “bad” samples
- Idea: instead of generating all samples at once, can we use each “good” sample to help us get the next “good” sample?

- **Markov Chain Monte Carlo**: construct a *sequence* of samples
  - $\theta^{(1)} \rightarrow \theta^{(2)} \rightarrow \cdots \rightarrow \theta^{(t)} \rightarrow \cdots$
  - Each sample depends on the previous one
  - We get to specify how the transitions happen using the unnormalized target $q(\theta)$
  - We choose the transitions to theoretically guarantee that the steady state distribution is the true posterior
  - Therefore, for large values of t, $\theta^{(t)}$ is a sample from the true posterior
  - How do we set up the transitions?
    - Metropolis-Hastings
    - Gibbs sampling
    - and many more (beyond the scope of Data 102)

Metropolis-Hastings

- **Inputs**:
  - Unnormalized target distribution q(θ) *(usually numerator of the posterior)*
  - Proposal distribution V(θ' | θ) *(should be easy to sample from)*
  - Initial sample $\theta^{(0)}$
- **Outputs**:
  - Samples that approximate the normalized distribution q(θ) / ∫ q(θ) dθ
- **How it works**:
  - Repeat, starting with t=1:
    - Generate a proposal for $\theta^{(t)}$ using $V(\theta^{(t)} \mid \theta^{(t-1)})$
    - With probability $\min\left\{1, \frac{q(\theta')}{q(\theta)} \frac{V(\theta|\theta')}{V(\theta'|\theta)}\right\}$, accept the proposal and assign it to $\theta^{(t)}$
    - If the proposal is rejected, try again with a new proposal

MCMC: Practical Considerations

- Theory says: as $t \rightarrow \infty$, then $\theta^{(t)}$ is one sample from the true posterior
- In practice, we can't afford to wait that long for just one sample
- Practical MCMC
  - **Burn-in time** (almost always): throw away the first ~100s of samples
  - Option 1: take every sample after that
  - Option 2: take every k-th sample after that (throwing away the ones in between)
- If we take "small" steps, then we need to choose option 2
- If we take "big" steps (each new sample is allowed to be far from the previous one), then we can choose option 1

Frequentist and Bayesian Modeling

- Goal: estimate unknown (random/fixed for Bayesian/frequentist) from data
- What good are Bayesian models?
  - We can bring in domain knowledge with a **prior** distribution
  - Results depend on how much data we have (we’re less confident with less data)
  - We can model complex relationships between many different hidden variables
- How do we compute posterior distributions?
  - Simple models: conjugate priors speed up computation
  - Complex models: approximate inference
  - Rejection sampling
  - MCMC
  - Metropolis-Hastings
  - Gibbs sampling