Approximate Bayesian computation

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

November 29, 2021

Outline

Last week:

- Measurement error
- Missing data – types of missingness

Today:

- Approximate methods for computing the posterior

Recap

Approximating the posterior

Recall the point of computation in Bayesian modeling:

- Applying Bayes' theorem gives us an un-normalized posterior distribution:

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

- If we could calculate the normalizing constant,

$$p(y) = \int p(y|\theta)p(\theta)d\theta$$

  then we could describe the posterior distribution and make inferences directly from that
- In practice this integral is not usually tractable, analytically or numerically
- We still want to approximate expectations, variances, and other summary statistics for parameters of interest

Approximating the posterior

Previous approximation methods:

- Grid approximation
  - Compute the unnormalized $p(\theta|y)$ on a number of grid points θ_i – use this to normalize, sample, etc.
  - Hopeless for high-dimensional θ
- Quadratic approximation (Laplace approximation)
  - Find a posterior mode $\hat{\theta}$ via gradient descent
  - Fit a quadratic to $\log p(\theta|y)$ at $\hat{\theta}$ for a normal approximation to the posterior
- Markov chain Monte Carlo
  - Approximate quantities of interest using sample statistics, sampling from the exact posterior

Weaknesses of MCMC

- MCMC is really pretty good
  - Modern methods (e.g. NUTS) explore parameter space very efficiently most of the time
  - We get to sample from the exact posterior distribution – no approximation required
  - Automatically implemented by probabilistic programming languages
- For some models, it is too computationally expensive
  - Gaussian processes
  - Hidden Markov models
  - Bayesian neural networks
  - Any sufficiently complex model, or very large data set
- Sensitive to tuning parameters, can struggle with some models

Approximate Bayesian computation

Other methods of approximating the posterior fall under the umbrella of *approximate Bayesian computation*

- Expectation-maximization (EM) algorithm
  - Saw this for HMM fitting
  - Can be applied more generally
- Variational inference
  - Have used this as an initialization step for MCMC
  - Also applicable more generally
- Batched methods: expectation propagation, etc.

Expectation-maximization algorithm

EM algorithms

The Baum-Welch algorithm we saw in fitting HMM parameters is an example of a much wider class of algorithms called *expectation-maximization* algorithms.

These are applicable when the observed data depends on hidden/latent state variables as well as model parameters. Roughly, the idea is:

- Expectation step: compute the distribution of hidden state variables, given current model parameters
- Maximization step: compute the model parameters that maximize (log) likelihood given the state parameters from the expectation step

Repeat until done – score model by total log-likelihood of the data.

Section 13.4-13.6 in BDA has another presentation of EM algorithms in a different context.

EM algorithms

Formally:

- θ: model parameters
- X: hidden variables
- Y: observations
- $L(\theta|X, Y)$: likelihood function

1. E-step: compute $Q(\theta|\hat{\theta}) = E_{X|Y,\hat{\theta}}[\log L(\theta|X, Y)]$
2. M-step: compute $\theta^{\text{new}} = \arg\max_{\theta} Q(\theta|\hat{\theta})$

BW algorithm as EM

Recall the Baum-Welch algorithm has two steps:

- Perform smoothing to estimate the distribution of each X_t, given current transition/observation matrix values
- Update parameter values by counting transitions/observations given distributions of X_t

Although we don't explicitly calculate expectations of log-likelihoods, the smoothing step is an E step and the update step is an M step.

Gaussian HMM

The most common alternative to the multinomial distribution for HMMs is the Gaussian (normal) distribution. In this model:

- X_t still evolves according to a Markov chain with transition matrix A
- $\mathcal{O}_t \sim \mathrm{MVNormal}(\mu_{X_t}, \Sigma_{X_t})$
- Result: the observation distributions are Gaussian mixtures

EM for Gaussian HMM

To fit the Gaussian HMM, we only need to make the following modifications to the M step:

- Replace B_{ij} with $\boldsymbol{\mu}_i, \Sigma_i$
- Replace the update of B_{ij} with a maximum-likelihood estimate for a Gaussian, weighted by the estimated state probabilities (from smoothing):

$$\boldsymbol{\mu}_i^{\text{new}} = \frac{\sum_t P(X_t = i)\boldsymbol{y}_t}{\sum_t P(X_t = i)}$$

$$\Sigma_i^{\text{new}} = \frac{\sum_t P(X_t = i)(\boldsymbol{y}_t - \boldsymbol{\mu}_i^{\text{new}})(\boldsymbol{y}_t - \boldsymbol{\mu}_i^{\text{new}})^T}{\sum_t P(X_t = i)}$$

where y_i are the observations.

Variational inference

Variational inference

Variational inference refers to a suite of methods that attempt to find the best-fitting member of an approximation family

- Choose a family of approximating distributions $q(\theta)$ – e.g. multivariate Gaussian, mixture of Gaussians, etc.
- Minimize some measure of divergence between $q(\theta)$ and $p(\theta|y)$
    - Classically and most commonly, this is the KL divergence:

$$D_{KL}(q, p) = \int q(x) \log \frac{q(x)}{p(x)} dx$$

- Take the minimizer to be the final approximation of the posterior, and draw inferences from that

Variational inference

Basic approach of variational inference:

- Start with a family of approximating distributions $q(\theta|\phi)$ dependent on some parameters ϕ. Common approach is to take components of θ to be independent,

$$q(\theta|\phi) = \prod_j q_j(\theta_j|\phi_j)$$

- Begin with guesses of the parameters ϕ.
- Cycle through the q_j; update the parameters ϕ_j so that $\log g_j(\theta_j|\phi_j)$ is set to the average of $\log p(\theta|y)$, averaged over the remaining $J-1$ approximating distributions $q_i(\theta_i|\phi_i)$ with their current values of ϕ_i
    - q must be chosen so that this averaging is practical – can be specific to the model, if using a conditionally conjugate model, or can work with, e.g., normal distributions

Variational inference

- Similar in spirit to Gibbs sampling
  - Update one parameter at a time, holding all others fixed
  - Difference: interested in averages over other parameters rather than sampling from the conditional density
  - Exploit conditional conjugacy or similar properties so that expectations can be computed analytically
- Example in BDA3 Section 13.7:
  - Application to the 8 schools problem
  - Chooses model-specific approximating distributions
  - Shows that the update step improves the KL divergence

Model checking and followups

- The end result of variational Bayes, $q(\theta)$, is a fully generative model – a probability distribution for the parameters θ
- Not just a MAP estimate, as one might get from gradient descent
- As a result we get estimates of uncertainty
- Can do our usual model checking and evaluation:
  - Draw a sample (θ^s) from the fitted $q(\theta)$
  - Use to generate posterior predictions for a predictive check
  - Use to compute log pointwise predictive density for WAIC, LOO

Implementation in PyMC: ADVI

Although you can put a lot of work into specifying and programming a custom VI algorithm, PyMC also has a built-in approach

- ADVI – automatic differentiation variational inference
- Uses Gaussian approximating family
- Computes necessary derivatives using automatic differentiation (also used under the hood in the MCMC sampling)

Using ADVI

- Using as an initialization method to MCMC: pass `init='advi'` when you call `pm.sample()`
  - We've seen this before once or twice, e.g. in the bike share GP example
- To fit the model itself: replace `pm.sample()` and use ADVI directly

Let's see this in action.

Model checking and followups

Can use variational methods to provide a starting point for sampling methods:

- Get a better approximation to the target distribution by using the variational approximation $q(\theta)$ as a proposal distribution for importance sampling
  - Potential problem – variational fit often less variable than target distribution
  - Distribution of importance ratios can have a long tail
  - Ran into this problem when using importance sampling in PSIS-LOO
- Use draws from $q(\theta)$ as a starting point for a particle filtering algorithm
- Use statistics and draws from $q(\theta)$ to choose tuning parameters and starting points for MCMC

EM as VI

Expectation-maximization is a special case of variational inference:

- VI has $J(=\dim\theta)$ steps – update one conditional distribution q_j at a time, averaging over all others
- EM has two steps: alternately estimate a parameter ϕ and average over the others γ
- EM is VI, if:
  - parameters are partitioned into two components, ϕ and γ
  - approximating distribution for ϕ is a point mass
  - approximating distribution for γ is unconstrained, so that $q(\gamma) = p(\gamma|\phi, y)$.

A couple other methods

Next steps: OPVI

Limitation of classical variational inference:

- Generally only works with KL divergence as an objective function
- Needs sufficiently tractable approximating family, to compute conditional expectations

New method: operator variational inference (OPVI)

- More general – can use other objective functions
- Does not require any analytical properties for the approximating family; uses Monte Carlo sampling to estimate necessary expectations

Expectation propagation

EP is

- Another iterative algorithm approximating the posterior by a best-fit distribution from a specific family of $q(\theta)$s
- Different objective function and computation
- Especially useful for batching and parallelizing computation – can be much faster for large or hierarchically structured models/data
- No guarantee of convergence!

Additional reading

For more mathematical details (in case you find yourself having to implement any of this):

- Overview of several methods (EM, VI, EP): BDA chapter 13
- Expectation propagation: Vehtari et al., "Expectation propagation as a way of life" (`https://arxiv.org/abs/1412.4869`)
- OPVI: Ranganath et al., "Operator variational inference" (`https://arxiv.org/pdf/1610.09033.pdf`)

Summary

Today:

- Survey of some approximate computational methods

Next time:

- Bayesian neural networks