Part 2: Markov chain Monte Carlo (MCMC) methods

- Indirect sampling methods
- Markov chains
- MCMC methods - Metropolis-Hastings and Gibbs Sampler
- The multivariate normal model
Lecture 2.6: Convergence criteria

As we have already seen, the validity of the MCMC approach relies on the fact that we can confidently assume that the realizations of the MC are indeed samples from the target distribution,

There are 2 ways to approach this problem:

- **Theoretical (probability) perspective:** measuring the distance and establishing theoretical bounds (hard or even impossible to calculate)

- **Statistical perspective:** analyzing the properties of the observed output from the chain.

  It must be used with caution because it is based on the empirical properties of a single chain or a set of chains; hence, it cannot ever guarantee convergence is reached with certainty.
When you run a Metropolis - Hastings

and wait for the convergence
Convergence diagnostic

- An MCMC algorithm has converged at iteration $T$ when its output can be safely thought to arise from the true stationary distribution of the Markov chain for all $t > T$.
- If the MCMC fails to converge to the target distribution, the resulting estimates will be biased and unreliable.
- Eye-balling by plotting traces of the individual parameters.
- There has been much effort in developing MCMC convergence diagnostics.
- **Note:** Passing an MCMC diagnostic does not guarantee that a chain is stationary! You can only check for signs of non-convergence.
Quick checks of convergence

- A **trace plot** of the sequence $\theta^{(t)}$’s against $t$ is a first empirical check of convergence.
- In cases of strong attraction from a local mode, the chain can behave as if it was simulated from the neighborhood of this mode and appear to have converged, when in fact, it has not.
- It is common to run a few parallel chains from different starting values until convergence and check that they settle around common values.
**Autocorrelation**

- How quickly the sampled values move around the parameter space is called the speed of **mixing**.
- A Markov chain with a high autocorrelation moves around the parameter space slowly, taking it longer to achieve stationarity.
- For a sequence \( \{\theta^{(1)}, \ldots, \theta^{(T)}\} \), the lag-\( t \) autocorrelation function estimates the correlation between elements of the sequence that are \( t \) steps apart:

\[
\text{acf}_t(\theta) = \frac{1}{T-t} \sum_{j=1}^{T-t} (\theta(j) - \bar{\theta})(\theta(j+t) - \bar{\theta}) \\
\frac{1}{T-1} \sum_{j=1}^{T} (\theta(j) - \bar{\theta})^2
\]

- In Gibbs sampling, we do not have much control over the correlation of the Markov chain, but with the Metropolis algorithm the correlation can be adjusted by selecting an optimal proposal distribution.
- Alternatively, modified versions of the Metropolis algorithm can be constructed that adaptively change the proposal distributions.
- High autocorrelations within chains indicate slow mixing and slow convergence.
- We are looking for the autocorrelation to drop precipitously over time.
- It might be necessary to increase the thinning interval to achieve a less highly correlated sample.
Convergence diagnostic statistics

- Geweke
- Gelman and Rubin
- Heidelberger and Welch
- Raftery and Lewis

See the R package CODA (Convergence Diagnostics and Output Analysis).
Geweke (1992) proposed a convergence diagnostic for Markov chains based on a test for equality of the means of the first and last part of a Markov chain (by default the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke’s statistic has an asymptotically standard normal distribution.
Gelman and Rubin (1992) propose a general approach to monitoring convergence of MCMC output in which $m > 1$ parallel chains are run.

- Use different starting values that are overdispersed relative to the posterior distribution.
- Convergence is diagnosed when the chains have “forgotten” their initial values, and the output from all chains is indistinguishable.
- The diagnostic is applied to a single variable from the chain. It is based on a comparison of within-chain and between-chain variances (similar to a classical analysis of variance) and assesses if they are different.
- Values of $\hat{R}$ near 1 suggest convergence.
• Given $M$ parallel chains for $\theta$, each of length $T$. The between-sequence variance, $B$, and the within-sequence variance $W$ are computed:

$$B = \frac{T}{M - 1} \sum_{j=1}^{M} (\bar{\theta}_j - \bar{\theta}.)^2$$

$$W = \frac{1}{M} \sum_{j=1}^{M} s_j^2 \quad \text{where} \quad s_j^2 = \frac{1}{T - 1} \sum_{k=1}^{T} (\theta_{jk} - \bar{\theta}_j)^2$$

• An estimate of the variance that is unbiased under stationarity is given by

$$\hat{Var}(\theta) = \frac{T - 1}{T} W + \frac{1}{T} B$$

• The estimated potential scale reduction or shrink factor can be used to monitor convergence of the Markov chain:

$$\hat{R} = \sqrt{\frac{\hat{Var}(\theta)}{W}}$$
• As the simulation convergence, $\hat{R}$ declines to 1, meaning that the parallel Markov chains are essentially overlapping.

• As $T \to \infty$, both $\widehat{Var}(\theta)$ and $W$ approach $Var(\theta)$, but from opposite directions, therefore $\hat{R} \to 1$.

• Note that when $B$ is large, which corresponds to the means being quite different between chains and thus the chain not having converged, $\hat{R}$ will exceed one.

• If the shrink factor is high, the chains should be run longer.

• A plot of $\hat{R}$ versus the iteration number can help in choosing the burnin time.
The convergence test uses the Cramer-von-Mises statistic to test the null hypothesis that the sampled values come from a stationary distribution.

- The test is successively applied, firstly to the whole chain, then after discarding the first 10%, 20%, of the chain until either the null hypothesis is accepted, or 50% of the chain has been discarded.
- The latter outcome constitutes “failure” of the stationarity test and indicates that a longer MCMC run is needed.
- If the stationarity test is passed, the number of iterations to keep and the number to discard (burn-in) are reported.
Raftery-Lewis
Calculates the number of iterations required to estimate the quantile \( q \) to within an accuracy of \( \pm r \) with probability \( p \).

\[
\text{raftery.diag(MCMC.chain1, q=0.95, r=0.01, s=0.95, converge.eps=0.001)}
\]

Quantile (q) = 0.95
Accuracy (r) = +/- 0.01
Probability (s) = 0.95

<table>
<thead>
<tr>
<th>Burn-in (M)</th>
<th>Total (N)</th>
<th>Lower bound (Nmin)</th>
<th>Dependence factor (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1889</td>
<td>1825</td>
<td>1.04</td>
</tr>
<tr>
<td>2</td>
<td>1845</td>
<td>1825</td>
<td>1.01</td>
</tr>
</tbody>
</table>

Separate calculations are performed for each variable within each chain. If the number of iterations in data is too small, an error message is printed indicating the minimum length of pilot run.
• $N_{\text{min}} =$ minimum number of iterations needed to estimate the specified quantile to the desired precision at the specified probability.

• $N =$ total number of iterations that should be run for each variable.

• $M =$ the number of initial iterations to discard as burn-in.

• The “dependence factor”, $I = \frac{N}{N_{\text{min}}}$, reveals mixing problems:
  - $I$ close to 1 indicates good mixing.
  - Large values of $I$ ($I > 5$) indicate high autocorrelations and likely convergence failure.
Summary

- Diagnostics cannot guarantee that chain has converged
- Can indicate that it has not converged

Solutions?

- Run longer and thin output
- Reparametrize model. Models that are overparametrized lead to high posterior correlations among the parameters and a dramatic slow down of the movement of the MCMC sampler through the parameter space.
- “Block” correlated variables together
- Integrate out variables
- Add auxiliary variables (Slice-sampler; data augmentation)
- Use “Rao-Blackwellization” in estimation
Rao-Blackwellization

Monte Carlo estimates are noisy! Can do better by using the idea of “Rao-Blackwellization”

- \( E_{MC}[g(\theta_1)|Y] = \frac{1}{T} \sum_t g(\theta_1^{(t)}) \)
- Iterated Expectations:

\[
E[g(\theta_1)|Y] = E_{\theta_2} E_{\theta_1|\theta_2}[g(\theta_1)|Y, \theta_2] = E_{\theta_2}[\tilde{g}(\theta_2)|Y] \\
\approx \frac{1}{T} \sum_t \tilde{g}(\theta_2^{(t)}) \equiv E_{RB}[g(\theta_1)|Y]
\]

- Motivated by the classical Rao Blackwell Theorem that says that taking any estimator and conditioning on a sufficient statistic will reduce its mean squared error.
- In particular, RB estimate has reduced variance over MCMC average (Liu et al 1996)
Example: Density estimation in the Normal model

Rather than use a histogram estimate of the density, use a Rao-Blackwell estimate.

- We know the posterior density of $\mu$ given $\sigma^2$, that is $p(\mu|\sigma^2, y)$
- Use MC average over draws $\sigma^2(1), \ldots, \sigma^2(T)$ to integrate out $\sigma^2$ to obtain the marginal

$$
\hat{p}(\mu|y) = \frac{1}{T} \sum_{t} p(\mu|\sigma^2(t), y)
$$