**36-705: Intermediate Statistics** **Fall 2017**

# Lecture 32: November 27

*Lecturer: Siva Balakrishnan*

Today we will wrap up our discussion of MCMC and then begin to talk about the bootstrap. These two ideas were developed around the same time (in the mid-70s, though MCMC really took off in the 90s), and gained popularity for roughly the same reason. This is also true of the Expectation-Maximization (EM) algorithm (and other iterative methods for computing point estimates). Their popularity coincides with the growth in computing power. Very abstractly, all of these methods (MCMC, the Bootstrap and EM) replace difficult analytic problems by "brute-force" computing.

These methods are not always successful and often cannot replace thinking critically about the application at hand, but do highlight a broad trend in statistics, where better computational tools have strongly influenced the methods of choice for a wide variety of problems. This is still the case and the interface between algorithmic ideas and statistics continues to blossom.

## 32.1 The Metropolis-Hastings algorithm

Recall, that our broad goal is to draw samples from a distribution $f$ (say).

Choose $X_0$ arbitrarily. For each subsequent index $i$ we follow the algorithm given below:

1. Sample a proposal $y \sim q(y|X_i = x)$ from a "proposal distribution" $q$.
2. Evaluate the ratio:
$$r = \min\left\{\frac{f(y)q(x|y)}{f(x)q(y|x)}, 1\right\}.$$
3. Accept the new sample $Y$ with probability $r$, and reject it otherwise. Alternatively, think of sampling $u \sim U[0,1]$ and accept if $u \leq r$ and reject otherwise.

**Some basic intuition:** We will understand formally why this works, but for now consider the case when the proposal is symmetric, i.e. $q(y|x) = q(x|y)$. In this case, we accept a new sample with probability:

$$r = \min\left\{\frac{f(y)}{f(x)}, 1\right\}.$$

Now, let us think about what it means to sample from $f$, roughly we want to draw more samples where $f$ is high, and fewer samples where $f$ is low. Our rule above basically says, always accept a sample if the density is higher at the proposed point (like hill-climbing) and if the density is lower at the proposed point you accept it with a smaller probability. This sampling rule is effectively biased to accept samples from regions where the density is high.

Three tasks remain: we need to decide how to choose a proposal distribution, we need to show that this algorithm does what we set out to, i.e. roughly generates samples from $f$, and we need to understand why this is useful in sampling from the posterior distribution for example.

### 32.1.1 Choosing a proposal distribution

This one is mostly an art, i.e. you try to pick a proposal distribution that somehow approximates the shape of the distribution you care about ($f$).

Often what we do is to choose:

$$q(Y|X = x) \sim N(x, \sigma^2),$$

so we sample a proposal around our current data point, and try to tune the tuning parameter $\sigma$ (by trying to maintain a reasonable acceptance ratio while still enforcing that we explore most of the space).

### 32.1.2 Sampling posteriors

Our goal in the beginning of last lecture was to sample from the posterior distribution:

$$\pi(\theta|X_1, \ldots, X_n) = \frac{\pi(\theta)\mathcal{L}(\theta; X_1, \ldots, X_n)}{\int \pi(\theta)\mathcal{L}(\theta; X_1, \ldots, X_n)d\theta}.$$

More generally, suppose we have a distribution that we know up to the normalizing constant, i.e. we can compute $g(x)$ but we want to sample from $f$ which is given by:

$$f(x) = \frac{g(x)}{\int g(x)dx},$$

and the denominator can be difficult to compute.

**The key point:** The Metropolis Hastings algorithm, only interacts with $f$ through ratios of the form

$$\frac{f(x)}{f(y)} = \frac{g(x)}{g(y)},$$

which are easy to compute. When you take the ratio, the normalizing constant disappears.

### 32.1.3 Limiting distributions

Now, we go back to the Metropolis Hastings algorithm. We need to show that this algorithm is constructing a Markov chain and the limiting distribution of this Markov chain is $f$.

The first part is easy: each subsequent sample $X_{i+1}$ only depends on $X_i$ and $q$ and does not depend on any of the prior $X_1, \ldots, X_{i-1}$ (conditional on $X_i$) so the samples $X_1, \ldots, X_n$ form a Markov chain.

Let us first understand the transition probabilities of our Markov chain. In order to transition from $x$ to $y$ we need to sample $y$ from the proposal and then need to accept this proposal. This happens with probability:

$$T(x, y) = q(y|x) \min \left\{ \frac{f(y)q(x|y)}{f(x)q(y|x)}, 1 \right\}.$$

Using this we can see that detailed balance is satisfied and $f$ is the limiting distribution if:

$$f(x)q(y|x) \min \left\{ \frac{f(y)q(x|y)}{f(x)q(y|x)}, 1 \right\} \stackrel{?}{=} f(y)q(x|y) \min \left\{ \frac{f(x)q(y|x)}{f(y)q(x|y)}, 1 \right\}.$$

This is easy to check by some case analysis. For instance, suppose $f(x)q(y|x) \geq f(y)q(x|y)$, then this reduces to:

$$f(x)q(y|x)\frac{f(y)q(x|y)}{f(x)q(y|x)} \stackrel{?}{=} f(y)q(x|y),$$

which is clearly true. We can similarly check this is true in the case when $f(x)q(y|x) < f(y)q(x|y)$. From this we can conclude that $f$ is the limiting distribution of the Markov chain we have constructed.

### 32.1.4 Some caution

While MCMC is a really nice trick in order to generate samples from something close to the posterior, there is an important caveat that I have ignored. For a Markov chain, the limiting distribution is its "asymptotic distribution", i.e. it is the distribution you are getting samples from asymptotically (as $n \to \infty$).

The hope is usually that for small (finite) values of $n$ the distribution is close to the limiting distribution. This is called mixing or rapid mixing. Unfortunately, however, in many cases we do not know if the Markov chain mixes rapidly (this depends in a complicated fashion on the proposal and the unknown density $f$).

To a large extent, MCMC is a sensible heuristic, and some caution/care is required in applying it to difficult problems.

## 32.2 The Bootstrap

At a high-level the bootstrap is a way to try to estimate the variability (think variance or confidence intervals) of a point estimate, but to do so in a way that avoids difficult analytic calculations.

If our estimator is a simple average, then under some mild conditions we know that the sample variance and the CLT can be used to construct confidence intervals. If the estimator is the MLE (which asymptotically behaves somewhat like an average), then we can use the Fisher information, and MLE asymptotics to construct Wald-intervals.

The second case is already quite challenging: depending on the model, the Fisher information can be quite hard to compute analytically. Furthermore, often we want to estimate the variance of, or derive confidence intervals for an arbitrary (non MLE, non-average) statistic, and the bootstrap gives a way to do this in many cases without resorting to tedious calculations.

## 32.3 Bootstrap samples

We have discussed this before when we discussed plug-in estimators: given samples $X_1, \ldots, X_n \sim P$ we can write the empirical CDF as:

$$\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i \leq x),$$

and the corresponding empirical distribution as:

$$P_n(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_i \in A).$$

We can also imagine drawing *bootstrap samples* by drawing samples from $P_n$. We denote these as:

$$X_1^*, \ldots, X_n^* \sim P_n.$$

Drawing from the empirical distribution is the same as drawing from the distribution that puts mass $1/n$ at each observed sample, i.e. it is the same as drawing from the uniform distribution on the given samples. Equivalently, you can imagine drawing from the given samples (uniformly) with replacement.

## 32.4 Bootstrap variance estimate

To understand the idea, let us first consider the Monte-Carlo variance estimate. Suppose we had an estimator $\widehat{\theta}_n = g(X_1, \ldots, X_n)$ (this could be a complicated function), where $X_1, \ldots, X_n \sim P$ and we want to estimate $\text{Var}_P(\widehat{\theta}_n)$.

Supposing that we knew $P$ we could try to compute the variance analytically: this might be difficult. The Monte-Carlo variance estimate would be to instead draw $B$ samples of size $n$ from $P$, i.e. we draw, $\{X_{11}, \ldots, X_{1n}\}, \ldots, \{X_{B1}, \ldots, X_{Bn}\} \sim P$, to compute our estimator on each of these samples, i.e. compute $\widehat{\theta}_n^{(1)}, \ldots, \widehat{\theta}_n^{(B)}$ and then use the sample variance, i.e.

$$\widehat{\sigma}_n^2 = \frac{1}{B}\sum_{i=1}^{B}\left(\widehat{\theta}_n^{(i)}\right)^2 - \left(\frac{1}{B}\sum_{i=1}^{B}\widehat{\theta}_n^{(i)}\right)^2.$$

By the LLN we have that $\widehat{\sigma}^2 \xrightarrow{p} \text{Var}_P(\widehat{\theta}_n)$. Unfortunately, we typically do not know $P$.

By now, you have already guessed the idea behind the bootstrap. The idea is to replace $P$ in the above procedure by the empirical distribution $P_n$. We'll reason about this more carefully in the next lecture. For now, here is the algorithm:

Bootstrap Variance Estimator

1. Draw a bootstrap sample $X_1^*, \ldots, X_n^* \sim P_n$. Compute $\widehat{\theta}_n^* = g(X_1^*, \ldots, X_n^*)$.
2. Repeat the previous step, $B$ times, yielding estimators $\widehat{\theta}_{n,1}^*, \ldots, \widehat{\theta}_{n,B}^*$.
3. Compute:
$$\widehat{s}^2 = \frac{1}{B}\sum_{j=1}^{B}(\widehat{\theta}_{n,j}^* - \overline{\theta})^2,$$
where $\overline{\theta} = \frac{1}{B}\sum_{j=1}^{B}\widehat{\theta}_{n,j}^*$.
4. Output $\widehat{s}^2$.

## 32.5 Bootstrap Confidence Intervals

The bootstrap can also be used to obtain confidence intervals. If your estimator has a normal limit then you could just use a Wald interval with the bootstrap variance estimate, i.e. $C_n = [\widehat{\theta}_n - \widehat{s}z_{\alpha/2}, \widehat{\theta}_n + \widehat{s}z_{\alpha/2}]$.

It is often more accurate to use the distribution of the bootstrap estimates itself to construct the bootstrap confidence interval.

### 32.5.1 Hypothetical confidence interval

Suppose we knew the distribution of our estimator, in particular suppose we knew the distribution of $\sqrt{n}(\widehat{\theta}_n - \theta)$. Let us denote the distribution by $G$ and denote its $\alpha/2$ and $1-\alpha/2$ quantiles by $g_{\alpha/2}$ and $g_{1-\alpha/2}$.

Then a $1-\alpha$ confidence interval would be:

$$C_n = \left[\widehat{\theta}_n - \frac{g_{1-\alpha/2}}{\sqrt{n}}, \widehat{\theta}_n - \frac{g_{\alpha/2}}{\sqrt{n}}\right].$$

This might seem a little strange, but this is probably because you are used to confidence intervals based on the normal distribution which has symmetric quantiles. To verify this,

$$\begin{aligned}\mathbb{P}(\theta \in C_n) &= \mathbb{P}\left(g_{\alpha/2} \leq \sqrt{n}(\widehat{\theta}_n - \theta) \leq g_{1-\alpha/2}\right) \\ &= 1 - \alpha/2 - \alpha/2 = 1 - \alpha.\end{aligned}$$

Again the point is that we do not know the distribution $G$ above so we try to approximate this using the bootstrap.

### 32.5.2 Bootstrap confidence interval algorithm

Bootstrap Confidence Interval

1. Draw a bootstrap sample $X_1^*, \ldots, X_n^* \sim P_n$. Compute $\widehat{\theta}_n^* = g(X_1^*, \ldots, X_n^*)$.
2. Repeat the previous step, $B$ times, yielding estimators $\widehat{\theta}_{n,1}^*, \ldots, \widehat{\theta}_{n,B}^*$.
3. Let
$$\widehat{G}(t) = \frac{1}{B}\sum_{j=1}^{B} I\left(\sqrt{n}(\widehat{\theta}_{n,j}^* - \widehat{\theta}_n\right) \leq t).$$
4. Let
$$C_n = \left[\widehat{\theta}_n - \frac{g_{1-\alpha/2}}{\sqrt{n}},\ \widehat{\theta}_n - \frac{g_{\alpha/2}}{\sqrt{n}}\right]$$
where $g_{\alpha/2} = \widehat{G}^{-1}(\alpha/2)$ and $g_{1-\alpha/2} = \widehat{G}^{-1}(1-\alpha/2)$.
5. Output $C_n$.

## 32.6 Variants

There are many many many papers that have been written about the bootstrap. Particularly, there are lots of variants – the block bootstrap for time-series, the residual bootstrap or the wild bootstrap for regression, the parametric bootstrap for parametric models, the smooth bootstrap and ideas related to sub-sampling to avoid certain regularity conditions, the less computationally intensive but less general Jackknife and so on.