Lectures 11: Bootstrap I.

error propagation for nonlinear functions of fit parameters
\( \chi^2 \) distribution \quad \text{Maximum Likelihood parameter errors?}

How accurately are the fitted parameters determined? As Bayesians, we would instead say, what is their posterior distribution?

Taylor series:

\[
-\frac{1}{2} \chi^2(b) \approx -\frac{1}{2} \chi_{\text{min}}^2 - \frac{1}{2} (b - b_0)^T \left[ \frac{1}{2} \frac{\partial^2 \chi^2}{\partial b \partial b} \right] (b - b_0)
\]

So, while exploring the \( \chi^2 \) surface to find its minimum, we must also calculate the Hessian (2\textsuperscript{nd} derivative) matrix at the minimum.

Then

\[
P(b | \{y_i\}) \propto \exp \left[ -\frac{1}{2} (b - b_0)^T \Sigma_b^{-1} (b - b_0) \right] P(b)
\]

with

\[
\Sigma_b = \left[ \frac{1}{2} \frac{\partial^2 \chi^2}{\partial b \partial b} \right]^{-1}
\]

Notice that if (i) the Taylor series converges rapidly and (ii) the prior is uniform, then the posterior distribution of the \( b \)'s is multivariate Normal.
Linearized error propagation

What is the uncertainty in quantities other than the fitted coefficients:

Method 1: Linearized propagation of errors

\( b_0 \) is the MLE parameters estimate

\( b_1 \equiv b - b_0 \) is the RV as the parameters fluctuate

\[
\begin{align*}
    f & \equiv f(b) = f(b_0) + \nabla f \ b_1 + \cdots \\
    \langle f \rangle & \approx \langle f(b_0) \rangle + \nabla f \ \langle b_1 \rangle = f(b_0) \\
    \langle f^2 \rangle - \langle f \rangle^2 & \approx 2f(b_0)(\nabla f \ \langle b_1 \rangle) + \langle (\nabla f \ b_1)^2 \rangle \\
    & = \nabla f \ \langle b_1 b_1^T \rangle \nabla f^T \\
    & = \nabla f \ \Sigma \ \nabla f^T
\end{align*}
\]
from discussion on correlated normal variables

**Multivariate Normal Distributions**

Generalizes Normal (Gaussian) to M-dimensions
Like 1-d Gaussian, completely defined by its mean and (co-)variance
Mean is a M-vector, covariance is a M x M matrix

\[
N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{M/2} \det(\Sigma)^{1/2}} \exp\left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right]
\]

The mean and covariance of r.v.’s from this distribution are*

\[
\mu = \langle x \rangle \quad \Sigma = \langle (x - \mu)(x - \mu)^T \rangle
\]

In the one-dimensional case \( \sigma \) is the standard deviation, which can be visualized as “error bars” around the mean.

In more than one dimension \( \Sigma \) can be visualized as an error ellipsoid around the mean in a similar way.

\[
1 = (x - \mu)^T \Sigma^{-1} (x - \mu)
\]
from discussion on correlated normal variables

Question: What is the generalization of

$$
\chi^2 = \sum_i \left( \frac{x_i - \mu_i}{\sigma_i} \right)^2, \quad x_i \sim N(\mu_i, \sigma_i)
$$

to the case where the $x_i$'s are normal, but not independent? I.e., $x$ comes from a multivariate Normal distribution?

$$
N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{M/2} \det(\Sigma)^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right]
$$

The mean and covariance of r.v.'s from this distribution are

$$
\mu = \langle x \rangle \quad \Sigma = \langle (x - \mu)(x - \mu)^T \rangle
$$

In the one-dimensional case $\sigma$ is the standard deviation, which can be visualized as “error bars” around the mean.

In more than one dimension $\Sigma$ can be visualized as an error ellipsoid around the mean in a similar way.

$$
1 = (x - \mu)^T \Sigma^{-1} (x - \mu)
$$
from general definition of covariance matrix

The covariance matrix is a more general idea than just for multivariate Normal. You can compute the covariances of any set of random variables. It's the generalization to M-dimensions of the (centered) second moment Var.

$$\text{Cov} (x, y) = \langle (x - \bar{x})(y - \bar{y}) \rangle$$

For multiple r.v.'s, all the possible covariances form a (symmetric) matrix:

$$C = C_{ij} = \text{Cov} (x_i, x_j) = \langle (x_i - \bar{x}_i)(x_j - \bar{x}_j) \rangle$$

Notice that the diagonal elements are the variances of the individual variables.

The variance of any linear combination of r.v.'s is a quadratic form in $C$ :

$$\text{Var} \left( \sum \alpha_i x_i \right) = \left\langle \sum_i \alpha_i (x_i - \bar{x}_i) \sum_j \alpha_j (x_j - \bar{x}_j) \right\rangle$$

$$= \sum_{ij} \alpha_i \langle (x_i - \bar{x}_i)(x_j - \bar{x}_j) \rangle \alpha_j$$

$$= \alpha^T C \alpha$$

This also shows that $C$ is positive definite, so it can still be visualized as an ellipsoid in the space of the r.v.'s., where the directions are the different linear combinations.
from general definition of covariance matrix

The covariance matrix is closely related to the linear correlation matrix.

$$r_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} \quad \text{more often seen written out as}$$

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

When the null hypothesis is that X and Y are independent r.v.’s, then r is useful as a p-value statistic (“test for correlation”), because

1. For large numbers of data points N, it is normally distributed,
   $$r \sim N(0, N^{-1/2})$$
   so $$r\sqrt{N}$$ is a normal t-value

2. Even with small numbers of data points, if the underlying distribution is multivariate normal, there is a simple form for the p-value (comes from a Student t distribution).
In our example, if we are interested in the area of the “hump”,

\[
\begin{align*}
\text{bfit} &= \begin{pmatrix}
1.1235 & 1.5210 & 0.6582 & 3.2654 & 1.4832 \\
\end{pmatrix} \\
\text{covar} &= \begin{pmatrix}
0.1349 & 0.2224 & 0.0068 & -0.0309 & 0.0135 \\
0.2224 & 0.6918 & 0.0052 & -0.1598 & 0.1585 \\
0.0068 & 0.0052 & 0.0049 & 0.0016 & -0.0094 \\
-0.0309 & -0.1598 & 0.0016 & 0.0746 & -0.0444 \\
0.0135 & 0.1585 & -0.0094 & -0.0444 & 0.0948 \\
\end{pmatrix}
\end{align*}
\]

\[
f = b_3 b_5
\]

\[
\nabla f = (0, 0, b_5, 0, b_3)
\]

\[
\nabla f \Sigma \nabla f^T = b_5^2 \Sigma_{33} + 2b_3 b_5 \Sigma_{35} + b_3^2 \Sigma_{55} = 0.0336
\]

\[
\sqrt{0.0336} = 0.18
\]

So \[b_3 b_5 = 0.98 \pm 0.18\].

Is it normally distributed?

Absolutely not! A function of normals is not normal (although, if they are all narrow, it might be close).
Sampling the posterior histogram

Method 2: Sample from the posterior distribution

1. Generate a large number of (vector) $b$'s

   $$b \sim \text{MVNormal}(b_0, \Sigma_b)$$

2. Compute your $f(b)$ separately for each $b$

3. Histogram

   ![Histogram Graph]

Note again that $b$ is typically (close to) m.v. normal because of the CLT, but your (nonlinear) $f$ may not, in general, be anything even close to normal!
Sampling the posterior histogram

Our example:

```matlab
bees = mvnrnd(bfit, covar, 10000);
humps = bees(:,3).*bees(:,5);
hist(humps,30);
std(humps)
```

`std = 0.1833`

Does it matter that I use the full covar, not just the 2x2 piece for parameters 3 and 5?
comparison of linear propagation and posterior sampling:

Compare linear propagation of errors to sampling the posterior

- Note that even with lots of data, so that the distribution of the b's really → multivariate normal, a derived quantity might be very non-Normal.
  - In this case, sampling the posterior is a good idea!

- For example, the ratio of two normals of zero mean is Cauchy
  - which is very non-Normal!.

- So, sampling the posterior is a more powerful method than linear propagation of errors.
  - even when optimistically (or in ignorance) assuming multivariate Gaussian for the fitted parameters

- In fact, sampling the posterior distribution of large Bayesian models whose parameters are not at all Gaussian is, under the name MCMC, the most powerful technique in modern computational statistics.
Method 3: Bootstrap resampling of the data

- We applied some end-to-end process to a data set and got a number $f$ out.
- The data set was drawn from a population of repetitions of the identical experiment
  - which we don’t get to see, unfortunately
  - we see only a sample of the population
- We’d like to draw new data sets from the population, reapply the process, and see the distribution of answers
  - this would tell us how accurate the original answer, on average, was
  - but we can’t: we don’t have access to the population

- However, the data set itself is an estimate of the population pdf!
  - in fact, it’s the only estimate we’ve got!
- So we draw from the data set— with replacement — many “fake” data sets of equal size, and carry out the proposed program
  - does this sound crazy? for a long time many people thought so!
  - Bootstrap theorem [glossing over technical assumptions]: The distribution of any resampled quantity around its full-data-set value estimates (naively: “asymptotically has the same histogram as”) the distribution of the data set value around the population value.
Let’s try a simple example where we can see the “hidden” side of things, too.

Visible side (sample):

Hidden side (population):

These happen to be drawn from a Gamma distribution.

Statistic we are interested in happens to be (it could be anything):

\[
\frac{\text{mean of distribution}}{\text{median of distribution}}\]

\[
\text{samedean} = \text{median(sample)} \\
\text{sammcan} = \text{mean(sample)} \\
\text{samstatistic} = \text{sammcan}/\text{samedean}
\]

\[
\text{samedean} = 2.6505 \\
\text{sammcan} = 2.9112 \\
\text{samstatistic} = 1.0984
\]

\[
\text{themedean} = \text{median(bigsample)} \\
\text{themean} = \text{mean(bigsample)} \\
\text{thestatistic} = \text{themean}/\text{themedean}
\]

\[
\text{themedean} = 2.6730 \\
\text{themean} = 2.9997 \\
\text{thestatistic} = 1.1222
\]

How accurate is this?
**Gamma distribution:**

\[ x \sim \text{Gamma}(\alpha, \beta), \quad \alpha > 0, \beta > 0 \]

\[ p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x > 0 \]

\[ \text{Mean}\{\text{Gamma}(\alpha, \beta)\} = \frac{\alpha}{\beta} \]

\[ \text{Var}\{\text{Gamma}(\alpha, \beta)\} = \frac{\alpha}{\beta^2} \]

When \( \alpha \geq 1 \) there is a single mode at \( x = (\alpha - 1)/\beta \)
To estimate the accuracy of our statistic, we bootstrap:

```matlab
ndata = 100;
nboot = 100000;
vals = zeros(nboot,1);
for j=1:nboot,
    choose = randsample(ndata,ndata,true);
    vals(j) = mean(sample(choose)) / median(sample(choose));
end
hist(vals,100)
```

Things to notice:
The mean of resamplings does **not** improve the original estimate! (Same data!)
The distribution around the mean is not identical to that of the population. But it is close and would become identical asymptotically for large `ndata` (not `nboot`).