Computational Inference Summary

Shravan Vasishth (vasishth@uni-potsdam.de)

June 9, 2014

Contents

Computational Inference Summary

Compiled by: Shravan Vasishth (vasishth@uni-potsdam.de)

Version dated: June 9, 2014

Very basic math

Product rule: $(uv)' = uv' + vu'$
Quotient role: $(u/v)' = (vu' - uv')/v^2$
$\Gamma(1/2) = \sqrt{\pi}$.
$\exp(-1) = 0.36788$.
Inverse of a matrix:
$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Solving quadratics

$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$

Taylor series expansions: The Delta Method

Let X be a random variable, and $g(\cdot)$ some function. Calculating the mean and variance of g(X) will involve integration. The Delta method allows us to approximate variance of g(X).
Let $E(X) = \mu$, and $Var(X) = \sigma^2$.
The Taylor series expansion of a function about a value a:

$$f(x) = f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2!} \tag{1}$$

Then,

$$f(x) \approx f(a) + f'(a)(x-a) \tag{2}$$

Let $a = \mu_x$, the mean of x. Then

$$y = f(x) \approx f(\mu_x) + f'(\mu_x)(x-\mu_x) \tag{3}$$

Since $y = f(x), E[y] = E[f(x)] = \mu$. Var(y)=Var(f(x))= $[f'(\mu)]^2 Var(X)$.
Example:
Suppose we have $X \sim t_3$, i.e., a Student's t-distribution with df=3:

$$f(x) = \frac{4\sqrt{3}}{\pi}\left(1 + \frac{x^2}{3}\right)^{-2} \tag{4}$$

[Note that for t_3, $\mu = 0$ and $\sigma^2 = 3/(3-2) = 3$.]
If we want to use importance sampling to sample from this distribution by using the normal as the importance density, we first need to know what the mean and variance of the Normal is going to be. We can obtain the mean and variance of the importance density by approximating the mean and variance of $\log f(x)$ (about 0) using the Delta method.

$$g(x) = \log f(x) = \log 4\sqrt{3} - \log \pi - 2\log\left(1 + \frac{x^2}{3}\right) \tag{5}$$

The first derivative is:

$$g'(x) = \log f'(x) = -\frac{2}{1+\frac{x^2}{3}} \times \frac{2x}{3} = \frac{4x}{3+x^2} \tag{6}$$

Equating the above to 0, we see that the MLE is x=0.
The variance: $g''(x) = \frac{8x^2-12-4x}{3+x^2} = -12/3 = -4$ (we plug in the MLE for x=0). So variance is 1/4.

```
> x<-seq(-10,10,by=0.01)
> plot(x,dt(x,df=3),ylim=c(0,0.9))
> lines(x,dnorm(x,mean=0,sd=1/2))
> ## testing:
> x<-rnorm(10000,mean=0,sd=1/2)
> w<-dt(x,df=3)/dnorm(x,mean=0,sd=1/2)
> mean(w*x)

[1] 0.019156

> ## prob. of x<2; seems OK
> mean(w*(x<2))

[1] 0.84609

> ## by comparison:
> pt(2,df=3)

[1] 0.93034

> ## importance samples given two uniform RVs, 0.9, and 0.45:
> (x1<-qnorm(0.9,mean=0,sd=1/2))

[1] 0.64078

> dt(x1,df=3)/dnorm(x1,mean=0,sd=1/2)

[1] 0.81021

> (x2<-qnorm(0.45,mean=0,sd=1/2))

[1] -0.062831

> dt(x2,df=3)/dnorm(x2,mean=0,sd=1/2)

[1] 0.46309
```

Monte Carlo Integration: Express integral as expectation

$$R = \int \frac{f(x)}{g(x)} g(x)\, dx = \int h(x) g(x)\, dx = E[h(X)] \tag{7}$$

$$\hat{R} = \frac{1}{n} \sum h(X) \quad \hat{R} \sim N(R, \frac{\sigma^2}{n}) \quad \sigma^2 = Var(h(X)) \tag{8}$$

$Var(h(X))$ is the usual definition of variance: $\frac{1}{n-1} \sum (h(X) - \hat{R})^2$.
SEs can be computed with $SE = \sigma/\sqrt{n}$, and so CIs can also be computed.

Example

$$R = \int_{-1}^{1} \exp(-x^2)\, dx \tag{9}$$

Let $g(x) = Unif(-1, 1)$ (g(x) needs to mimic f(x)). Then:

$$\hat{R} = \frac{1}{n} \frac{\sum \exp(-x^2)}{g(x)}\, dx = \frac{1}{n} \sum 2 \exp(-x^2) \tag{10}$$

Bayesian application: When unable to work with conjugate priors, the calculation of the expectation is a special case of MCI.

An important example of MC integration

Let $h(\theta, \phi)$ be some function where $\theta \sim Beta(5, 95)$, and $\phi \sim Beta(3, 1)$. The function h delivers the expected number of new infections for some disease, say. The user wants to determine

$$M = \int \int h(\theta, \phi) f(\theta) f(\phi)\, d\theta\, d\phi \tag{11}$$

If $Z_1, \dots, Z_{100}$ are the values generated by $h(\cdots)$, and if an alternative distribution is considered for $\theta \sim Unif(0, 0.10)$, explain how MC integration can be used to get an estimate of M with the new distribution (without recomputing h).
Here, note that $h(\theta, \phi)$ is a function of θ and ϕ. So, when we compute M, we are computing the joint expectation of θ and ϕ:

$$M = \int \int h(\theta, \phi) \mathbf{f}(\theta) \mathbf{f}(\phi)\, d\theta\, d\phi \tag{12}$$

If we want to replace $f(\theta)$ with $g(\theta)$, then we can eliminate $f(\theta)$ as follows:

$$M = \int \int h(\theta, \phi) \mathbf{f}(\theta) f(\phi)\ \frac{g(\theta)}{\mathbf{f}(\theta)}, d\theta\, d\phi \tag{13}$$

Therefore, given an estimate of M, you just have to do:

$$M \times \frac{g(\theta)}{f(\theta)} \tag{14}$$

In the above example, we would get:

$$\hat{M} = \sum_{i=1}^{100} Z_i \frac{10}{f(x_i)} \tag{15}$$

The numerator is 10 on the right side of the equation because $Unif(0, 0.10)$ has height 10.

Monte Carlo Hypothesis Tests

1. Generate n-1 test statistics under H_0.
2. Let $m = n\alpha$.
3. If T_{obs} is one of m largest $\{T_1, \dots, T_{n-1}, T_{obs}\}$, reject null.

Examples: Spatial stats, Chi-sq tests.

Randomization tests

1. No distributional assumptions
2. No random sampling assumption

Two-sample case

Sample from x,y together, without replacement, and randomly create new x and y.

```
> A<-c(233,291,312,250,246,197,268,224)
> B<-c(185,263,246,224,212,188,250,148)
> T_val<-abs(t.test(A,B)$statistic)
> ## randomly shuffle:
> n_sim<-1000
> T_samp<-rep(NA,n_sim)
> for(i in 1:n_sim){
 samp<-sample(c(A,B),replace=FALSE)
 T_samp[i]<-abs(t.test(samp[1:8],
                  samp[9:16])$statistic)}
> mean(T_samp>=T_val)
```

```
[1] 0.073
```

Other examples: Cholesterol and diet, outliers, ANOVA.

Single sample

Special assumptions:

1. Random samples
2. Symmetric distribution about mean of population

Steps:

1. If $H_0 : \mu = k$, then subtract k from data: $y = x - k$
2. Let observed mean be T_{obs}.
3. With p=0.5 change sign of each observation; compute T.
4. Significance test $mean(T >= T_{obs})$.

See Fisher Randomization Test p. 34.

Bootstrap

No distributional assumptions.

Two sample bootstrap hyp. testing

Note: sampling **with replacement**, cf. Randomization, which has sampling without replacement.
Null hypothesis: $H_0 : F_x = F_y = F$.

```
> T_val<-abs(t.test(A,B)$statistic)
> n_sim<-1000
> T_samp<-rep(NA,n_sim)
> for(i in 1:n_sim){
 samp<-sample(c(A,B),replace=TRUE)
 T_samp[i]<-abs(t.test(samp[1:8],
                  samp[9:16])$statistic)
 }
> mean(T_samp >= T_val)

[1] 0.058
```

Examples: Heart attack study, law school data

One sample bootstrap hyp. testing

Let $H_0 : \mu = k$. Transform data $y = x - mean(x) + k$. Then sample with replacement from transformed data y.

Parametric bootstrap

Procedure:

1. From data x, create estimates θ_1,θ_2 (etc.).
2. Get fitted distribution $f(\theta_1, \theta_2)$.
3. Create new bootstrap samples from this distribution.
4. Get estimates from bootstrap samples to obtain sampling distribution of parameters.

Non-parametric bootstrap

Create new bootstrap samples by sampling from x with replacement.

1. Estimate ECDF of data x.
2. Sample x_i^* from ECDF by sampling from replacement from x.
3. Estimate parameter of interest from x_i^*.
4. Find percentile CI.

Generating random variables

The inversion method

This method works when we can know the closed form of the pdf we want to simulate from and can derive the inverse of that function.
Steps:

```
Data: Given: iid sequence of $U_1, \ldots, U_n \sim Uniform(0,1)$
for $i$ in 1:n do
    Let $u_i = F(z)$;
    Store $z_i = F^{-1}(u_i)$ as a draw from $f(x)$;
end
```

Algorithm 1: Inversion sampling algorithm.

Example: let $f(x) = \frac{1}{40}(2x+3)$, with $0 < x < 5$. We have to draw a number from the uniform distribution and then solve for z, which amounts to finding the inverse function:

$$u = \int_0^z \frac{1}{40}(2x+3) \tag{16}$$

```
> u<-runif(1000,min=0,max=1)
> z<-(1/2) * (-3 + sqrt(160*u +9))
```

Example: Exponential distribution
Example: Binomial distribution
Example: Sampling from normal distribution
This method can't be used if we can't find the inverse, and it can't be used with multivariate distributions.

Box Muller method to generate RVs

The normal CDF $\Phi(\cdot)$ does not have closed form, so we can't use inversion sampling.
Take two RVs $u_1, u_2 \sim Unif(0,1)$, then compute

$$X_1 = \sqrt{-2ln(u_1)}\cos(2\pi u_2) \tag{17}$$

$$X_2 = \sqrt{-2ln(u_1)}\sin(2\pi u_2) \tag{18}$$

Then X_1, X_2 are N(0,1).

Generating chi-sq RVs using Box-Muller

If Y follows a standard normal distribution, then $X = Y^2 \sim \chi^2_1$.
If $Y_i, i = 1, \dots, n$ follow a standard normal distribution, then $X = \sum Y_i^2 \sim \chi^2_n$.
To generate χ^2_n:

1. Generate $Y_i, i = 1, \dots, n$ using Box-Muller method.
2. Then compute $X = \sum Y_i^2 \sim \chi^2_n$.

Generating Student's t-distribution using Box-Muller

1. Generate $Y \sim N(0,1)$ using Box-Muller method.
2. Generate $Z \sim \chi^2_n$ as above.
3. Then $X = \frac{Y}{\sqrt{Z/n}} \sim t_n$.

Generating F-distribution using Box-Muller

1. Generate $Y \sim \chi^2_m$ as above.
2. Generate $Z \sim \chi^2_n$ as above.
3. Then $X = \frac{Y/m}{Z/n} \sim F_{m,n}$.

The rejection method

If $F^{-1}(u)$ can't be computed, we sample from $f(x)$ as follows:

1. Sample a value z from a distribution $g(z)$ from which sampling is easy, and for which

$$mg(z) > f(z) \quad m \text{ a constant} \tag{19}$$

   $mg(z)$ is called an envelope function because it envelops $f(z)$.

2. Compute the ratio

$$R = \frac{f(z)}{mg(z)} \tag{20}$$

3. Sample $u \sim Unif(0,1)$.
4. If $R > u$, then z is treated as a draw from $f(x)$. Otherwise return to step 1.

For example, consider f(x) as above: $f(x) = \frac{1}{40}(2x+3)$, with $0 < x < 5$. The maximum height of $f(x)$ is 0.325 (why?). So we need an envelope function that exceeds 0.325. The uniform density $Unif(0,5)$ has maximum height 0.2, so if we multiply it by 2 we have maximum height 0.4, which is greater than 0.325.
In the first step, we sample a number x from a uniform distribution Unif(0,5). This serves to locate a point on the x-axis between 0 and 5 (the domain of x). The next step involves locating a point in the y direction once the x coordinate is fixed. If we draw a number u from Unif(0,1), then $mg(x)u = 2 * 0.2u$ is a number between 0 and $2 * 0.2$. If this number is less than f(x), that means that the y value falls within f(x), so we accept it, else reject. Checking whether $mg(x)u$ is less than $f(x)$ is the same as checking whether

$$R = f(x)/mg(z) > u \tag{21}$$

```
> #R program for rejection method of sampling
> ## From Lynch book, adapted by SV.
> count<-0
> k<-1
> accepted<-rep(NA,1000)
> rejected<-rep(NA,1000)
> while(k<1001)
 {
 z<-runif(1,min=0,max=5)
 r<-((1/40)*(2*z+3))/(2*.2)
 if(r>runif(1,min=0,max=1)) {
   accepted[k]<-z
   k<-k+1} else {
     rejected[k]<-z
   }
 count<-count+1
 }
```

```
> hist(accepted,freq=F,
      main="Example of rejection sampling")
> fn<-function(x){
   (1/40)*(2*x+3)
 }
> x<-seq(0,5,by=0.01)
> lines(x,fn(x))
```

```
> ## acceptance rate:
> table(is.na(rejected))[2]/
   sum(table(is.na(rejected)))
```

```
 TRUE
0.499
```

Rejection sampling can be used with multivariate distributions.
Some limitations of rejection sampling: finding an envelope function may be difficult; the acceptance rate would be low if the constant m is set too high and/or if the envelope function is too high relative to f(x), making the algorithm inefficient.

Rejection sampling: special case of truncated distributions

Example:
Find k in the pdf:

$$\int_2^\infty \frac{k}{\sqrt{2\pi}} \exp^{-\frac{1}{2}x^2}\, dx \tag{22}$$

This is a truncation of the standard normal. Since

```
> ## the area to the right of 2:
> (pval<-pnorm(2,lower.tail=F))
```

```
[1] 0.02275
```

$k = \frac{1}{0.0228} = 43.9558$.
For rejection sampling, sample from the full standard normal. Then:

$$f(x)/g(x) = \begin{cases} k, & x > 2, \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

So $supf(x)/g(x) = k$.
Normally, we would sample $u \sim Unif(0,1)$ and y from $g(\cdot)$, and accept if $u \leq f(y)/kg(y)$. But since

$$f(x)/kg(x) = \begin{cases} 1, & x > 2, \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

it will always be true that $u \leq f(x)/kg(x)$ if $y > 2$ and it will always be true that $u > f(x)/kg(x)$ if $y \leq 2$. So we don't need u.

1. Generate y from g(y)
2. if y>2, accept as sample.
3. else return to 1.

The acceptance probability will be large if c is small; so if the truncation region is large, sampling will be inefficient. So in this case, since c is large (43.9), sampling is inefficient. Cf. below.
Truncated example continued:
Given $u = 0.6$, use inversion sampling to sample from $\frac{k}{\sqrt{2\pi}} \exp^{-\frac{1}{2}x^2}$.
If u=0.60, this means that the upper bound z of the above distribution is at a location that is

```
> (prob<-0.60*pnorm(2,lower.tail=F))
```

```
[1] 0.01365
```

That means that the upper bound is at

```
> pnorm(2)
```

```
[1] 0.97725
```

```
> qnorm(pnorm(2)+prob)
```

```
[1] 2.3615
```

Hence, the sample given u=0.6 is 2.3615.
Truncated exponential (example continued)
Given a truncated exponential distribution, $f(y) = \exp(-0.5y)$, where $y > 2$, and given one random uniform draw 0.60, produce one random variable from the distribution of Y using the inversion method.
Steps:

1. We can work out $\int_0^2 1/2 \exp(-1/2y)\, dy = 0.6321$.
2. Therefore, $\int_2^\infty 1/2 \exp(-1/2y)\, dy = 1 - 0.6321 = 0.3678$.

3. 60% of 0.3678 is 0.22.

4. Therefore, we have to find z in $\int_0^z 1/2\exp(-1/2y)\,dy = 0.6321 + 0.22$.

5. Answer: $z = 3.83$.

You can generate a negatively correlated value by taking $u = 1 - 0.6 = 0.4$ and then repeating the above. Instead of 60% of 0.3678, take 40% of 0.3678, which is 0.14712. So, we have to find z in $\int_0^z 1/2\exp(-1/2y)\,dy = 0.6321 + 0.14712$. Answer is $z = 3.01$.

Now we use rejection sampling to sample from $\frac{k}{\sqrt{2\pi}}\exp^{-\frac{1}{2}x^2}$ using the above truncated exponential as envelope: $g(x) = 1/2\exp(-1/2x)$.

Note that we can find k_2 the normalizing constant for g(x):

$$\begin{aligned}\int_2^\infty k_2\frac{1}{2}\exp(-\frac{1}{2}y)\,dy =&1\\ =&k_2\left[-\exp(-\frac{1}{2}y)\right]_2^\infty \\ =&k_2\times 0.3678\end{aligned} \tag{25}$$

So $k_2 = 2.718$.

Now we have:

$$f(x) = \frac{k}{\sqrt{2\pi}}\exp^{-\frac{k_2}{2}x^2} \qquad g(x) = 1/2\exp(-1/2x) \tag{26}$$

Taking ratios:

$$\frac{f(x)}{g(x)} = \frac{2k}{k_2\sqrt{2\pi}}\exp(\frac{1}{2}x - \frac{1}{2}x^2) \tag{27}$$

To get supremum, take logs, ignoring terms not involving x, then differentiate and equate to 0. We get: $x = 1/4$. Replace x in the ratio with this number:

$$\frac{f(x)}{g(x)} = \frac{2k}{k_2\sqrt{2\pi}}\exp(-1/2\times 1/4 - 1/2\times 1/16) = 9.42 \tag{28}$$

So c=9.42, which is much more efficient than using the standard normal with c=43.9.

Why rejection sampling works

Let $R = \frac{f(y)}{mg(y)}$.
Algorithm: repeat n times:

1. Generate $Y \sim g(\cdot)$, and $u \sim Unif(0,1)$.
2. if $u \leq R$, accept $X = Y$; else return to 1.

Why does it work (Robert and Casella p. 52, explain this)? We need to show that:

$$P(Y \leq x \mid U \leq R) = P(X \leq x) \tag{29}$$

By the definition of conditional probability:

$$P(Y \leq x \mid U \leq R) = \frac{P(Y \leq x, U \leq R)}{P(U \leq R)} \tag{30}$$

Note that $P(Y \leq x, U \leq R) = \int_{-\infty}^x\int_0^R du g(y)\,dy$, and $P(U \leq R) = \int_{-\infty}^\infty\int_0^R du g(y)\,dy$ (Why the double integral? Because R depends on y).
This means that we can write:

$$\frac{P(Y \leq x, U \leq R)}{P(U \leq R)} = \frac{\int_{-\infty}^x\int_0^R du g(y)\,dy}{\int_{-\infty}^\infty\int_0^R du g(y)\,dy} \tag{31}$$

Also note that $\int_0^R du = R$. This allows us to write:

$$\frac{\int_{-\infty}^x\int_0^R du g(y)\,dy}{\int_{-\infty}^\infty\int_0^R du g(y)\,dy} = \frac{\int_{-\infty}^x Rg(y)\,dy}{\int_{-\infty}^\infty Rg(y)\,dy} \tag{32}$$

Expanding out $R = \frac{f(y)}{mg(y)}$:

$$\frac{\int_{-\infty}^x\int_0^R du g(y)\,dy}{\int_{-\infty}^\infty\int_0^R du g(y)\,dy} = \frac{\int_{-\infty}^x[\frac{f(y)}{mg(y)}]g(y)\,dy}{\int_{-\infty}^\infty[\frac{f(y)}{mg(y)}]g(y)\,dy} \tag{33}$$

m is a constant and cancels out in the numerator and denominator, and $g(y)$ also cancels out, giving us:

$$\frac{\int_{-\infty}^x[f(y)]\,dy}{\int_{-\infty}^\infty[f(y)]\,dy} = P(X \leq x) \tag{34}$$

Efficiency of the rejection method

Need to make c as small as possible because probability of rejection is $1 - \frac{1}{c}$.
Expected number of samples needed to generate an RC is c.

Sampling from multivariate distributions

Let $X \sim N_2(m, V)$. Let U be the Cholesky square root of V. Generate two independent normal RVs Z, then generate X by doing $m + U^T Z$. See p. 65 of notes.

Example of generating normal from Cauchy

to-do

Exercise 7.4: Beta(2,2) from Unif

c should be 1.5:

```
> x<-seq(0,1,by=0.01)
> plot(x,dbeta(x,2,2))
```

Derive this: (done)

Exercise 7.6.4

```
> library(MASS)
> Sigma<-matrix(c(1,0.5,0.5,1),ncol=2)
> x<-mvrnorm(100000,mu=c(0,1),Sigma=Sigma)
> plot(x[,1],x[,2])
> var(x)
```

```
        [,1]    [,2]
[1,] 0.99896 0.49966
[2,] 0.49966 0.99974
```

Exercise 7.7.5

Importance sampling exercise:

```
> x<-runif(10000,min=0,max=1)
> w<-dbeta(x,10,15)/dunif(x)
> mean(w*x)
```

```
[1] 0.40705
```

```
> mean(rbeta(10000,10,15))
```

```
[1] 0.39996
```

```
> x.unweighted<-sample(x,replace=TRUE,
                       prob=w)
> quantile(x.unweighted,c(0.05,0.95))
```

```
     5%     95%
0.24948 0.56399
```

```
> qbeta(c(0.05,0.95),10,15)
```

```
[1] 0.24639 0.56289
```

Latin Hypercube sampling

"We first divide the sample space of X into n regions of equal probability, and then sample one value at random from each region, to get $X_1, \ldots, X_n$. We then do likewise for Y to get $Y_1, \ldots, Y_n$. Finally, we randomly permute the order of the Ys before pairing the X and Y values, so that each X_i will be randomly paired with one value from the set of Ys. The idea is that by stratifying into regions of equal probability, we can obtain a small sample that is 'more representative' of the distribution of X and Y ."

Monotonicity:

"If we have some scalar function Y = h(X) and an LHS X (a vector), then it possible to prove that $1/n \sum g(Y)$ is an unbiased estimator of E[g(Y)], where $Y_i = h(X_i)$. Additionally, if h is monotone with respect to each element of X, and g is a monotone function of Y, then it also possible to prove that this estimator has smaller variance than the usual estimator based on a simple random sample of the same size. Note that even if these conditions do not hold, Latin hypercube may still be more efficient than simple Monte Carlo sampling."

Example: Exercise 7.9.4 Latin Hypercube:

```
> n<-100
> ## create strata:
> z<-seq(from=0,to=1-1/n,length=n)
> ## generate probs within each stratum:
> u1<-z+runif(n,0,1/n)
> ## inversion sampling to generate x2:
> x1<-qgamma(u1,3,4)
> ## inversion sampling to generate x2:
> u2<-z+runif(n,0,1/n)
> logx2<-qlnorm(u2,0,1)
> ## permute x2:
> logx2<-sample(logx2)
> plot(x1,logx2,type="p")
> ## cf. ``regular'' sampling:
> x1<-rgamma(100,3,4)
> x2<-rlnorm(100,0,1)
> plot(x1,x2)
```

M divisions, N variables, the maximum number of combinations is $(M!)^{N-1}$.

Antithetic variables

For generating correlated variables, used in conjunction with inversion method. To get m MC samples from $f(x)$:

1. Generate $U_1, \dots, U_n \sim Unif(0,1)$. Call this u1.
2. Create correlated values: $u2 \leftarrow 1 - u1$.
3. Generate each sample j using inversion sampling: $Y_j = f(F^{-1}(u1j))$, $Y_j' = f(F^{-1}(1-ui))$. j=m/2.
4. Mean is $\frac{1}{m/2}\sum((Y_j + Y_j')/2)$.
5. Variance is $\frac{\sigma^2}{n}(1+\rho)$ see page 79 of notes.

Profile likelihood

To compute profile log likelihood for a parameter θ_1, first find MLE for θ_2 treating θ_1 as constant. Then plug in MLE for θ_2 into full log likelihood to find profile log lik for θ_1.
Profile deviance for k parameter log lik:
$D_p(\theta_1*) = 2 \times (\ell(\hat{\theta}; x) - \ell_p(\theta_1*; x)) \sim \chi^2_k$
If null hypothesis is that $\theta = \theta_0$ and $\hat{\theta}$ is the MLE, then hypothesis test is:

$$D_p(\theta_1*) = 2 \times (\ell_p(\hat{\theta}; x) - \ell_p(\theta_0; x)) \sim \chi^2_k \tag{35}$$

Maximum likelihood estimation notes

Example 1: Binomial

Instead of calling the parameter θ I will call it p.

$$L(p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{36}$$

Log lik:

$$\ell(p) = \log \binom{n}{x} + x \log p + (n-x)\log(1-p) \tag{37}$$

Differentiating:

$$\ell'(p) = \frac{x}{p} - \frac{n-x}{1-p} = 0 \tag{38}$$

Taking the second partial derivative with respect to p:

$$\ell''(p) = -\frac{x}{p^2} - \frac{n-x}{(1-p)^2} \tag{39}$$

The quantity $-\ell''(p)$ is called **observed Fisher information**.

Taking expectations:

$$E(\ell''(p)) = E(-\frac{x}{p^2} - \frac{n-x}{(1-p)^2}) \tag{40}$$

Exploiting that fact the $E(x/n) = p$ and so $E(x) = E(n \times x/n) = np$, we get

$$E(\ell''(p)) = E(-\frac{x}{p^2} - \frac{n-x}{(1-p)^2}) = -\frac{np}{p^2} - \frac{n-np}{(1-p)^2} \underset{\substack{\uparrow \\ exercise}}{=} -\frac{n}{p(1-p)} \tag{41}$$

Next, we negate and invert the expectation:

$$-\frac{1}{E(\ell''(\theta))} = \frac{p(1-p)}{n} \tag{42}$$

Evaluating this at $\hat{p}$, the estimated value of the parameter, we get:

$$-\frac{1}{E(\ell''(\theta))} = \frac{\hat{p}(1-\hat{p})}{n} = \frac{1}{I(p)} \tag{43}$$

[Here, $I(p)$ is called **expected Fisher Information**.]

If we take the square root of the inverse Information Matrix

$$\sqrt{\frac{1}{I(p)}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{44}$$

we have the **estimated standard error**.

Another example using the normal distribution:

Example 2: Normal distribution

This example is based on Khuri (p. 309). Let $X_1, \dots, X_n$ be a sample of size n from $N(\mu, \sigma^2)$, both parameters of the normal unknown.

$$L(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp[-\frac{1}{2\sigma^2} \sum (x-\mu)^2] \tag{45}$$

Taking log likelihood:

$$\ell = -\frac{n}{2} \log \frac{1}{(2\pi\sigma^2)} - \frac{1}{2\sigma^2} \sum (x-\mu)^2 \tag{46}$$

Taking partial derivatives with respect to μ and σ^2 we have:

$$\frac{\partial \ell}{\partial \mu} = \frac{1}{\sigma^2} \sum (x-\mu) = 0 \Rightarrow n(\bar{x} - \mu) = 0 \tag{47}$$

$$\frac{\partial \ell}{\partial \sigma^2} = \frac{1}{2\sigma^4} \sum (x-\mu)^2 - \frac{n}{2\sigma^2} = 0 \Rightarrow \sum (x-\mu)^2 - n\sigma^2 = 0 \tag{48}$$

Simplifying we get the maximum likelihood estimates of μ and σ^2: $\hat{\mu} = \bar{x}$ and $\hat{\sigma}^2 = \frac{1}{n} \sum (x - \bar{x})^2$. Note that these are unique values.

We can verify that $\hat{\mu}$ and $\hat{\sigma}^2$ are the values of μ and σ^2 that maximize $L(x \mid \mu, \sigma^2)$. This can be done by taking the second order partial derivatives and finding out whether we are at a maxima or not. It is convenient to write the four partial derivatives in the above example as a matrix, and this matrix is called a Hessian matrix.

If this matrix is positive definite (i.e., if the determinant of the matrix is greater than 0), we are at a maximum.

The Hessian is also going to lead us to the information matrix as in the previous example: we just take the negative of the expectation of the Hessian, and invert it to get the variance covariance matrix.

(This is just like in the binomial example above, except that we have two parameters to worry about rather than one.)

Consider the Hessian matrix H of the second partial derivatives of the log likelihood ℓ.

$$H = \begin{pmatrix} \frac{\partial^2 \ell}{\partial \mu^2} & \frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} \\ \frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} & \frac{\partial^2 \ell}{\partial \sigma^4} \end{pmatrix} \tag{49}$$

Now, if we compute the second-order partial derivatives replacing μ with $\hat{\mu}$ and σ^2 with $\hat{\sigma}^2$ (i.e., the values that we claim are the MLEs of the respective parameters), we will get:

$$\frac{\partial^2 \ell}{\partial \mu^2} = -\frac{n}{\hat{\sigma}^2} \tag{50}$$

$$\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} = -\frac{1}{\hat{\sigma}^2} \sum (x - \hat{\mu}) = 0 \tag{51}$$

$$\frac{\partial^2 \ell}{\partial \sigma^4} = -\frac{n}{2\hat{\sigma}^2} \tag{52}$$

The determinant of the Hessian is $\frac{n^2}{2\hat{\sigma}^6} > 0$. Hence, $(\hat{\mu}, \hat{\sigma}^2)$ is a point of local maximum of ℓ. Since it's the only maximum (we established that when we took the first derivative), it must also be the absolute maximum.

As mentioned above, if we take the negation of the expectation of the Hessian, we get the Information Matrix, and if we invert the Information Matrix, we get the variance-covariance matrix.

Once we take the negation of the expectation, we get ($\theta = (\mu, \sigma^2)$):

$$I(\theta) = \begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{pmatrix} \tag{53}$$

Finally, if we take the inverse and evaluate it at the MLEs, we will get:

$$\frac{1}{I(\theta)} = \begin{pmatrix} \frac{\hat{\sigma}^2}{n} & 0 \\ 0 & \frac{2\hat{\sigma}^4}{n} \end{pmatrix} \tag{54}$$

And finally, if we take the square root of each element in the matrix, we get the estimated standard error of $\hat{\mu}$ to be $\frac{\hat{\sigma}}{\sqrt{n}}$, and the standard error of the $\hat{\sigma}^2$ to be $\hat{\sigma}^2 \sqrt{2/n}$.