STA410/2102 (Statistical Computation) Lecture Notes, Fall 2007

by Jeffrey S. Rosenthal, University of Toronto

(Last updated: December 5, 2007.)

Note: I will update these notes regularly (on-line). However, they are just rough, point-form notes, with no guarantee of completeness or accuracy. They should in no way be regarded as a substitute for attending the lectures, doing the homework exercises, or reading the reference books.

- Introduction to course, handout, references, R, prerequisites, etc.
- "Statistical Computation" v. "Computational Techniques in Statistics".
- Grading: Final Exam 50%, Term Test (Oct 23) 30%, Homework 20%.
- Course web page (updates etc.): **probability.ca/sta410**.
- Homework #0: worth one BONUS point. Due next class at 6:10 SHARP!
  - Run R (either on utstat, or cquest, or home computer, or laptop, or ...).
  - Use R to add up the digits of your student number.
  - Use R to compute "2∧52 + 1 - 2∧52", and "2∧53 + 1 - 2∧53".
  - Use R to verify at least one other (simple) computation from the first lecture.
  - Print out R's output, together with your NAME and STUDENT NUMBER, and bring it to class next week (by 6:10 – no late assignments will be accepted!).
- How many undergrads? In Statistics specialist program? Statistics major? Actuarial Science specialist? major? Math? Computer Science? Physics/Chemistry? Economics? Management? Life Sciences? Engineering? Other?
- How many grad students? In statistics? biostat? CS? other?
- How many neither undergrad nor grad?
- INTRODUCTION TO R:
  - "3+4", "2∧10", "1.3∧6.2", "sin(5)", "log(17)", "exp(9)", etc.

- "x = c(3,4,7,2.3)" ... "x" ... "sum(x)" ... "mean(x)" ... "sd(x)" ...
- > s = 0 ... > for (i in 1:5) ... + s = s + i∧2 ... > s ... [1] 55 ...
- x = 1:5 ... sum(x∧2) ...
- pnorm(-1) ... rpois(5,14) ...
- "plot(x)" ... "plot(x, type='b')" ... "hist(x)" ... pie(x∧2) ...

• COMPUTER ARITHMETIC:

- > 2∧10 ... [1] 1024 ... > 2∧100 [1] 1.267651e+30 > 2∧1000 ... [1] 1.071509e+301 ... > 2∧10000 ... [1] Inf ...
- > 2∧(-1000) ... [1] 9.332636e-302 ... > 2∧(-10000) ... [1] 0 ...
- > 2∧10 + 1 - 2∧10 ... [1] 1 ... > 2∧100 + 1 - 2∧100 ... [1] 0 ... Why??

• DOUBLE PRECISION FLOATING POINT NUMBERS:

- (There's also single precision, integer, etc., but we'll stick to double precision flaoting point.)
- Computers store numbers in binary (base 2), usually in the following format:

$$(-1)^s\, 2^{e-1023}\, 1.m_1 m_2 \ldots m_{52} \quad \text{(base 2)}$$

$$= \; (-1)^s\, 2^{e-1023}\, (1 + \sum_{i=1}^{52} m_i\, 2^{-i})\,,$$

where:
—— the "sign" $s = 0$ or 1 (1 bit);
—— the "written exponent" e is between 0 and $(2^{11} - 1) - 1 = 2046$ (11 bits);
—— (So, the "true exponent" equals $e - 1023$, and is between -1023 and 1023.)
—— the "mantissa" consists of 52 bits m_i, each 0 or 1 (52 bits).
∗ (64 bits [i.e., 8 eight-bit "bytes"] total, where each "bit" is 0 or 1.)
—— (single precision: 1 + 8 + 23 = 32 bits)

• For example, $5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 101$ (base 2) $= (-1)^0\, 2^{1025-1023}\, 1.01$.

- And, $-14.75 = -1110.11$ (base 2) $= (-1)^1\, 2^{1026-1023}\, 1.11011$.

- (Also have a few special values, like Inf, $-$Inf, NaN, ... ; the special case $e = 2^{11} - 1$ is reserved for these; hence that final "-1" in the written exponent's range.)

- (Special underflow trick: when $e = 0$, then the leading digit "1" is omitted, allowing for even smaller values to be represented. [Further details at beginning of next class.])

  – > 2∧1023 ... [1] 8.988466e+307 ... > 2∧1024 ... [1] Inf ...

  – > 2∧(-1023) ... [1] 1.112537e-308 ... > 2∧(-1024) ... [1] 5.562685e-309 ... > 2∧(-1025) ... [1] 2.781342e-309 ...

  – > 2∧(-1074) ... [1] 4.940656e-324 ... > 2∧(-1075) ... [1] 0 ...

- To <u>multiply</u> two floating point numbers (goes from left to right):
  —— Add their true exponents.
  —— Use long multiplication (base 2) on mantissas, with carrying, discarding (or rounding) any bits lower-order than 2^{-52}.
  —— "Normalise", i.e. do a final shift of the exponent.
  —— Works pretty much as expected, e.g.

$$7 \times 3 \;=\; (1.11 \times 2^2) \times (1.1 \times 2^1) \;=\; (1.11 \times 1.1) \times 2^{2+1} \;=\; 10.101 \times 2^3$$

$$=\; 1.0101 \times 2^4 \;=\; (1 + (1/4) + (1/16)) \times 16 \;=\; 16 + 4 + 1 \;=\; 21\,.$$

- To <u>add</u> two floating point numbers (goes from left to right):
  —— Adjust them to have the same (larger) true exponent, dropping lower-order bits.
  —— Add their mantissas, with carrying.
  —— "Normalise", i.e. do a final shift of the exponent.
  —— Can lead to dropping of important quantities.

  – For example,

$$32 + 3 \;=\; 1.0 \times 2^5 + 1.1 \times 2^1 \;=\; 1.0 \times 2^5 + 0.00011 \times 2^5$$

$$=\; 1.00011 \times 10^5 \;=\; 32 + 2 + 1 \;=\; 35\,.$$

  – > 2∧10 + 1 - 2∧10 ... [1] 1 ... > 2∧52 + 1 - 2∧52 ... [1] 1 ... > 2∧53 + 1 - 2∧53 ... [1] 0 ... > 2∧53 - 2∧53 + 1 ... [1] 1 ...

– Why? Well,

$$2^{53} + 1 \;=\; 1.0 \times 2^{53} + 1.0 \times 2^{0} \;=\; 1.0 \times 2^{53} + 0.00\ldots01 \times 2^{53}$$

$$=\; 1.00\ldots01 \times 2^{53} \;=\; 1.0 \times 2^{53}$$

(lower order bit gets dropped!).

– > 1 + 2∧(-52) - 1 . . . [1] 2.220446e-16 . . . > 1 + 2∧(-53) - 1 . . . [1] 0 . . .

- MUST SOMETIMES BE CAREFUL WITH COMPUTATIONS!
- e.g. suppose want $P[X = 200]$, where $X \sim$ Poisson(100). (Guesses?)
  - > exp(-100) * 100∧200 / prod(1:200) . . . [1] NaN
  - > exp(-100) . . . [1] 3.720076e-44 . . . > exp(-100) * 100∧200 . . . [1] Inf . . .
  - > exp(-100) * 100∧100 / prod(1:200) * 100∧100 . . . [1] 0 . . . > exp(-100) * 100∧100 / prod(1:200) . . . [1] 0 . . .
  - > 100∧200 . . . [1] Inf . . . > prod(1:200) . . . [1] Inf . . .
  - > exp(-100) * 100∧100 / prod(1:100) . . . [1] 0.039861 . . . > exp(-100) * 100∧100 / prod(1:100) * 100∧100 . . . [1] 3.9861e+198 . . . > exp(-100) * 100∧100 / prod(1:100) * 100∧100 / prod(101:200) . . . [1] 4.716971e-19 . . .
  - > dpois(200,100) . . . [1] 4.716971e-19 . . . > dpois(201,100) . . . [1] 2.346752e-19 . . . > ppois(199, 100, lower.tail=FALSE) . . . [1] 9.34315e-19 . . .
- Alternatively, can compute $\log \mathbf{P}[X = 200] = -100 + 200\log(100) - \sum(\log(1:200)) = -42.19795$, and then $\exp(-42.19795) = 4.716971\text{e-}19$.
- CONCLUSION: Computer arithmetic mostly works well, but not always!
  - Important to understand <u>when</u> and <u>why</u> it fails.
- INTRODUCTION TO MONTE CARLO – REAL/FAKE RANDOMNESS:
- Divide into about eight groups of students: A,B,C,. . .
- Each group:

- Take sheet of paper.
- Write group name at top.
- Flip a coin once.
- If it's heads, you're a "true randomness" group: flip a coin 100 times, NEATLY writing '1' for heads and '0' for tails.
- If it's tails, you're a "fake randomness" group: put the coin away, and MAKE UP 100 FAKE COIN FLIPS, again NEATLY writing '1' for heads and '0' for tails.

- Can the professor tell which are fake? (to be continued!)

——————— **END WEEK #1**———————

[Collect Homework #0 assignments.]

[Assign Homework #1.]

[Offer extra course handouts as needed.]

[Note: I posted lecture notes on web (probability.ca/sta410). Did anyone look at them?]

[Question: How many are running R on cquest? utstat? home? laptop? other? none?]

[Question: Did anyone look at any of the reference books?]

Summary of Previous Class:

∗ Intro to course, R, etc.

∗ R's Failures, e.g. $2^{100} + 1 - 2^{100}$.

∗ Double precision floating point numbers:

—— $(-1)^s \, 2^{e-1023} \, 1.m_1 m_2 \ldots m_{52}$ (base 2)

∗ Multiplication and addition of floating point numbers.

∗ Experiment with real versus fake randomness (continue today).

- CLARIFICATION: true exponent is between -1023 and 1023. However, if it's -1023 (i.e. $e = 0$), then m_1 becomes units digit (since mantissa might not begin with 1):

$$(-1)^s \, 2^{-1023} \, m_1.m_2 m_3 \ldots m_{52} \text{ (base 2)} \;=\; (-1)^s \, 2^{-1023} \, (\sum_{i=1}^{52} m_i \, 2^{-i+1}) \, .$$

So, the smallest possible power of two is

$$(-1)^0\, 2^{-1023}\, 0.000\ldots001 \text{ (base 2)} \;=\; 2^{-1023}\, 2^{-51} \;=\; 2^{-1023-51} \;=\; 2^{-1074}\,.$$

- Computations near 2^{-1074} are quite fragile:
  - > 2∧(-1074) ... [1] 4.940656e-324 ... > 2∧(-1074)*1.4 ... [1] 4.940656e-324 ... > 2∧(-1074)*1.5 ... [1] 9.881313e-324 ... > 2∧(-1074)*2 ... [1] 9.881313e-324 ...
- REAL/FAKE RANDOMNESS EXPERIMENT, CONTINUED:
  - How to distinguish between real and fake randomness?
  - Read them out? See if they "sound" random??
  - Record various "summary statistics" of the 100 flips: number of heads, number of changes, length of longest sequence, number of three-in-a-row ["coinstats.pdf"].
  - Formally: "heads" $= \#\{i : X_i = 1\} = \text{sum}(X)$. "changes" $= \#\{i : X_i \neq X_{i-1}\}$. "longest" $= \max\{m : \exists\, i \text{ s.t. } X_i = \ldots = X_{i+m-1}\}$. "three" $= \#\{i : X_i = X_{i+1} = X_{i+2}\}$.
  - Which are "typical" values??
- To compute typical values, we do a Monte Carlo simulation.
  - Run file "Rcoins", using command: `source('Rcoins')`
  - Get sequence of the values of the each statistic for repeated, randomised sequences.
  - Consider each statistic's mean and standard deviation.
  - Also, get histograms of the values of the statistics.
  - Using this, which of the sequences have "reasonable" values of the statistics?
- Our computer program was an example of a *Monte Carlo simulation*.
  - We performed random simulation from some probability distribution, to determine typical values of various statistics under that particular model (i.e., true coin flips).

  - We then compared the output to the observed values (from the student groups), to judge whether the model was accurate.

- There are many other uses of Monte Carlo, too.
- Example: Suppose want to estimate $\mathbf{E}[Z^4 \cos(Z)]$, where $Z \sim \text{Normal}(0, 1)$.
  - Monte Carlo solution: replicate a large number $z_1, \ldots, z_n$ of Normal(0,1) random variables, and let $x_i = z_i^4 \cos(z_i)$.
  - Their mean $\overline{x} \equiv n^{-1} \sum_{i=1}^n x_i$ is an (unbiased) estimate of $\mathbf{E}[X] \equiv \mathbf{E}[Z^4 \cos(Z)]$.
  - What about this estimate's uncertainty?
  - Well, can estimate standard deviation of $\overline{x}$ by "standard error":

$$se \;=\; \text{sd}(\overline{x}) \;=\; n^{-1/2}\,\text{sd}(x) \;=\; n^{-1/2}\,\sqrt{\text{var}(x)} \;=\; n^{-1/2}\,\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}\;.$$

  - Homework #1!

- We'll see more about Monte Carlo later on.
- NUMERICAL OPTIMISATION:
  - e.g. maximum likelihood estimator (MLE): find θ to <u>maximise</u> $L(x_1, \ldots, x_n \,|\, \theta)$.
  - e.g. regression: find β_1 and β_2 to <u>minimise</u>, say, $\sum_{i=1}^n \big(y_i - (\beta_1 + \beta_2 x_i)\big)^2$.
  - How to maximise (or minimise)?
  - Can use R functions "optimise", "nlm", "optim", ...
  - But how do they work??
- EXAMPLE: $f(x) = x^2 \exp(-(x + \sin(x)))$.
  - How to maximise f, over all $x > 0$?
  - Want to solve for $f'(x) = 0$, but how?
  - Well, if $f'(x) = 0$, then $h(x) \equiv f'(x) + x = x$.
  - So, how to solve for $h(x) = x$?

- FIXED-POINT ITERATION TO SOLVE FOR $h(x) = x$, e.g. $f'(x) + x = x$:
  - Choose some initial guess x_0.
  - Then let $x_1 = h(x_0)$, and $x_2 = h(x_1)$, and so on.
  - In general, $x_{n+1} = h(x_n)$.
  - If it converges, i.e. $\{x_n\} \to x$, and h continuous, then $h(x) = x$.
- BACK TO EXAMPLE: maximise $f(x) = x^2 \exp(-(x + \sin(x)))$ over $x > 0$.
  - Here $f'(x) = 2 * x * \exp(-(x + \sin(x))) + x^2 * \exp(-(x + \sin(x))) * (-1 - \cos(x))$.
  - Then $h(x) = f'(x) + x$.
  - If, say, start at 1, then $h(1) = 1.072901$, $h(h(1)) = 1.136110$, $h(h(h(1))) = 1.192873$.
  - Will it ever converge?
  - Yes it will! (Run file "Riter".) Converges to 4.165672.
  - Works well for any starting value in $(0, 8)$.
  - But $h(0) = 0$, so *that* doesn't go anywhere (local min).
  - Also, if start at 9 or larger, then converges to another fixed point, namely 10.06609.
  - Actually, $f(4.17) \doteq 0.63$ while $f(10.07) \doteq 0.007$, so the point we "wanted" was 4.17, not 10.07. How to know?
  - 10.07 is tiny additional local max: "curve(f, 8, 12)".
- So, in this example, iterative algorithm works, but is globally *unstable*.
- LOCAL STABILITY: if x_n is close to fixed point r, then

  $e_{n+1} \equiv x_{n+1} - r = h(x_n) - r = h(r + (x_n - r)) - r \approx h(r) + h'(r)\,(x_n - r) - r = h'(r)\,(e_n)$,

  so that $e_n \approx e_0\,(h'(r))^n$.
  - Geometric convergence <u>if</u> $|h'(r)| < 1$. <u>Pretty</u> good ...

- Comment: why not just use "plot" or "curve", and just "eyeball" the result?
  - Plots can "miss" features (off of plot's domain, too narrow, ...).
  - Want procedure to be *automated* for greater efficiency.
  - In higher dimensions, can't "eyeball" so easily.
  - And more.

——————— **END WEEK #2**———————

[Reminder: Homework #1 due Oct 2 at 6:10 pm sharp. In Question 5, delete "use enough replications to make the standard error less than 0.05".]

[Contacting me: I don't have regular weekly "office hours" since usually no one comes! But you can e-mail or find me any time, e.g. after class, or arrange a meeting by e-mail.]

[Perhaps office hour just before HW#1 is due: this Friday, Sept 28 and/or next Monday, Oct 1, from 2:30 – 3:30?]

Summary of Previous Class:
∗ Discussion of floating-point arithmetic near 2^{-1074}.
∗ Real/Fake randomness experiment (cont'd).
—— Summary statistics.
—— Monte Carlo simulation to estimate mean, sd, histogram.
—— Not too successful, but ...
∗ Monte Carlo to estimate, say, $\mathbf{E}[Z^4 \cos(Z)]$.
—— Mean, standard error.
—— Finish it on HW #1.
∗ Optimisation.
—— Used for MLE, regression, etc.
∗ Iterative method: $h(x) = f'(x) + x$, $x_{n+1} = h(x_n)$.
—— Works well in example of $f(x) = x^2 \exp(-(x + \sin(x)))$.
—— Locally stable (local geometric convergence) if $|h'(r)| < 1$.
∗ Just plotting f not good enough ...

- MORE ABOUT "ROUNDING" IN R:

- > 2∧53 + 1 - 2∧53 ... [1] 0 ... > 2∧53 + 1.1 - 2∧53 ... [1] 2 ... > 2∧54 + 1.1 - 2∧54 ... [1] 0 ... > (2∧53 + 2) + 1 - (2∧53 + 2) ... [1] 2 ... > (2∧53 + 2) + 0.9 - (2∧53 + 2) ... [1] 0 ... > (2∧53 + 4) + 1 - (2∧53 + 4) ... [1] 0 ...

- BACK TO OPTIMISATION:

- Simple MLE example:

  - $X_1, \ldots, X_n \sim N(\theta, 1)$, with θ unknown.

  - Observe some values $x_1, \ldots, x_n$.

  - Likelihood function:

$$L(x_1, \ldots, x_n \,|\, \theta) \;=\; \prod_{i=1}^{n} N(\theta, 1;\, x_i)$$

$$= \prod_{i=1}^{n} (2\pi)^{-1/2} \exp(-(x_i - \theta)^2/2) \;=\; (2\pi)^{-n/2} \exp(-\sum_{i=1}^{n} (x_i - \theta)^2/2)\,.$$

  - Want to find the value $\hat{\theta}$ for θ that maximises $L(x_1, \ldots, x_n \,|\, \theta)$.

  - (Actually, we know: maximised when $\theta = \overline{x}$. But suppose we didn't.)

  - Log-likelihood function:

$$\ell(x_1, \ldots, x_n \,|\, \theta) \;=\; \log L(x_1, \ldots, x_n \,|\, \theta) \;=\; C - \sum_{i=1}^{n} (x_i - \theta)^2/2\,.$$

  - Suppose two observations: $x_1 = 1$, and $x_2 = 2$. (And, can ignore "C".)

  - f = function(theta) { - (1-theta)∧2/2 - (2-theta)∧2/2 }

  - fp = function(theta) { + (1-theta) + (2-theta) }

  - h = function(theta) { fp(theta) + theta} = 3 - theta.

  - Iterations don't converge – they oscillate back and forth about 1.5.

  - Indeed, here $h'(\theta) \equiv -1$.

- Suppose instead have density $\propto \exp(-(x_i - \theta)^4)$.

- $\ell(x_1, \ldots, x_n \,|\, \theta) \;=\; C - \sum_{i=1}^{n} (x_i - \theta)^4$.
- Again suppose two observations: $x_1 = 1$, and $x_2 = 2$.
- f = function(theta) { - (1-theta)∧4 - (2-theta)∧4 }
- fp = function(theta) { + 4*(1-theta)∧3 + 4*(2-theta)∧3 }
- h = function(theta) { fp(theta) + theta}
- Even worse – iterations oscillate to $\pm\infty$, unless start at 1.5. (File "Riter2".)
- Indeed, here $h'(\theta) = O(\theta^3)$.

- BISECTION ALGORITHM TO SOLVE FOR $g(x) = 0$, e.g. $f'(x) = 0$:
  - Assume that g is a continuous function.
  - Start with interval $[a, b]$ with $g(a)\, g(b) < 0$, i.e. either $g(a) < 0 < g(b)$ or $g(a) > 0 > g(b)$. (So, a and b must "bracket" a root.)
  - Then, let $c = (a + b)/2$ (midpoint).
  - If $g(a)\, g(c) < 0$, replace b by c, otherwise (if $g(b_0)\, g(c_1) < 0$), replace a by c. [If $g(c) = 0$ then done.]
  - Guaranteed to always have a root within the interval $[a, b]$.
  - Once interval $[a, b]$ sufficiently small, then a and b are (both) sufficiently close to a root. (Error is bounded by length of interval – good!)
  - And, length of interval $[a, b]$ multiplied by $1/2$ at each iteration – good.
  - (Do graphical example on the blackboard ... more on next HW.)
- NEWTON-RAPHSON ALGORITHM TO SOLVE FOR $g(x) = 0$, e.g. $f'(x) = 0$:
  - WANT to find r such that $g(r) = 0$.
  - Have some guess x_0, and know $g(x_0)$ and $g'(x_0)$.
  - How to find r?

- Well, if g were linear about x_0, i.e. $g(x) = g(x_0) + (x - x_0)\, g'(x_0)$, then could solve for r such that $g(r) = 0$, by: $0 = g(r) = g(x_0) + (r - x_0)\, g'(x_0)$, i.e. $r = x_0 - g(x_0)/g'(x_0)$.

- Use this for iterative algorithm: $x_{n+1} = x_n - g(x_n)/g'(x_n)$, for $n = 0, 1, 2, \ldots$.

• BACK TO SECOND MLE EXAMPLE:

- g = function(theta) { + 4*(1-theta)∧3 + 4*(2-theta)∧3 }

- gp = function(theta) { - 12*(1-theta)∧2 - 12*(2-theta)∧2 }

- If $x_0 = 4$, then $x_1 = 4 - g(4)/g'(4) \doteq 3.103$, $x_2 = 2.474$, $x_3 = 2.014$, $x_4 = 1.676$, $x_5 = 1.513$, $x_6 = 1.500$, ...

- Converges very quickly to 1.5. (File "Rnewt".)

• LOCAL STABILITY OF NEWTON-RAPHSON?

- Well, <u>if</u> x_n close to r, then recalling that $x_{n+1} = x_n - g(x_n)/g'(x_n)$,

$$\begin{aligned} 0 \;=\; g(r) \;&\approx\; g(x_n) + (r - x_n)\, g'(x_n) + \frac{1}{2}(r - x_n)^2\, g''(x_n) \\ &=\; g(x_n) + r\, g'(x_n) - x_n\, g'(x_n) + \frac{1}{2}(r - x_n)^2\, g''(x_n) \\ &=\; r\, g'(x_n) - x_{n+1}\, g'(x_n) + \frac{1}{2}(r - x_n)^2\, g''(x_n) \\ &=\; -e_{n+1}\, g'(x_n) + \frac{1}{2}(e_n)^2\, g''(x_n)\,, \end{aligned}$$

  so $e_{n+1} \approx (e_n)^2[\frac{1}{2}g''(x_n)/g'(x_n)] \approx (e_n)^2[\frac{1}{2}g''(r)/g'(r)]$.

- Quadratic convergence – <u>better</u> than geometric!

• MUST BE CAREFUL!

- Might not converge at all.

- Or, might *appear* to converge, but not really, e.g. $g(x) = (1/x) - 1$, and $x_0 = 0.000001$, then $x_1 \doteq 0.000002$, and $x_2 \doteq 0.000004$; <u>appears</u> to be settling down to $x = 0$, but true root is at $x = 1$.

- File "Rnewt2".

- SECANT METHOD:
  - If it's too hard to compute $g'(x_n)$, then instead *approximate* this by $g'(x_n) \approx (g(x_n) - g(x_{n-1})) \,/\, (x_n - x_{n-1})$.
  - So, update becomes: $x_{n+1} = x_n - g(x_n)\,(x_n - x_{n-1})/(g(x_n) - g(x_{n-1}))$.
  - Need *two* starting values: x_0 and x_1.
  - Also, if $x_{n+1} = x_n$, then x_{n+2} will be undefined, so algorithm should terminate (and output x_{n+1}) at that point.
  - Asymptotically just as fast as Newton-Raphson, but even more unstable in early stages.
  - Note that if $g(x_n)$ and $g(x_{n-1})$ have opposite signs, then x_{n+1} will always be between x_n and x_{n-1}. [Show graph.]
- FALSE POSITION ("REGULA FALSI", "SAFE BISECTION") METHOD:
  - Combine Secant Method with Bisection Method.
  - Start with interval $[a, b]$ bracketing a root, as in Bisection Method.
  - Then let $c = b - g(b)\,(b - a)/(g(b) - g(a))$, as in Secant Method.
  - Then, as in Bisection Method, keep c, together with whichever of b and a has corresponding function value of opposite sign. [Or, if $g(c) = 0$, then done.]
  - Like Bisection Method, guarantees that the latest two points will always bracket a root.
  - But like Secant Method, "tries" to have faster convergence.

——————— **END WEEK #3**———————

[Collect HW #1; assign HW #2; reminder of in-class test on Oct 23.]

Summary of Previous Class:

$*$ MLE examples.

—— Fixed-Point Iteration Method doesn't always work.

$*$ Bisection Algorithm to solve $g(x) = 0$.

—— "Guaranteed" to converge, with geometric factor 1/2.

—— Clear error bound.

$*$ Newton-Raphson Algorithm.

—— Given x_n, let x_{n+1} be root of linear approximation.

—— Quadratic LOCAL convergence, but no GLOBAL guarantee.

—— Requires computing $g'(x_n)$.

$*$ Secant Method: Replace derivative with secant approximation.

—— Still quadratic local convergence; still no global guarantee.

$*$ False Position (Safe Bisection):

—— Combine best of Bisection and Secant methods.

—— Use Secant Method formula to find new "c".

—— Combine c with either a or b, as in Bisection Method.

- Safe Bisection can still be slow, e.g. $g(x) = x^{20} - 0.1$, with $x_0 = 0$ and $x_1 = 2$.
- Can improve this with ILLINOIS METHOD: Like safe bisection method, except that if the same endpoint (b, say) is used twice in a row, then replace $g(b)$ by $g(b)/2$ in the x_{n+1} formula. May help speed up convergence, e.g. above example.
- CONCLUSION: There are *many* optimisation algorithms out there.
  - Good software packages do optimisation very carefully, and <u>combine</u> different methods as needed.
- WHAT ABOUT OPTIMISING MULTIVARIATE FUNCTIONS $f : \mathbf{R}^d \to \mathbf{R}$?
  - Want to solve for $g_i(\mathbf{x}) \equiv \frac{\partial}{\partial x_i} f(\mathbf{x}) = 0$, for $i = 1, 2, \ldots, d$.
- MULTI-DIMENSIONAL NEWTON'S METHOD:
  - Proceed as in one-dim Newton's method:

$$0 \;=\; g_i(\mathbf{r}) \;\approx\; g_i(\mathbf{x}) + \sum_j (\frac{\partial g_i}{\partial x_j})(\mathbf{x})\,(r_j - x_j) \;=\; g_i(\mathbf{x}) + \sum_j H_{ij}(r_j - x_j)\,,$$

    where we write the derivative matrix as:

$$\{H_{ij}\} \;=\; \left(\, \tfrac{\partial g_i}{\partial x_j} \,\right) \;=\; \left(\tfrac{\partial^2 f}{\partial x_i\, \partial x_j} \right).$$

– Or, in "column vector" form,

$$0 \;\approx\; \mathbf{g}(\mathbf{x}) + \mathbf{H}(\mathbf{x})\,(\mathbf{r} - \mathbf{x})\,.$$

So, solving for $\mathbf{r}$ gives $\mathbf{r} \;\approx\; \mathbf{x} - \mathbf{H}^{-1}(\mathbf{x})\,\mathbf{g}(\mathbf{x}) \;=\; \mathbf{x} - (\mathbf{H}^{-1}\mathbf{g})(\mathbf{x})$.

– So, algorithm has updates of the form:

$$\mathbf{x}_{n+1} \;=\; \mathbf{x}_n - \mathbf{H}^{-1}(\mathbf{x}_n)\,\mathbf{g}(\mathbf{x}_n) \;=\; \mathbf{x}_n - (\mathbf{H}^{-1}\mathbf{g})(\mathbf{x}_n)\,.$$

– Harder to implement (matrix inverse, vector multiplication), but still has quadratic local convergence.

- ANOTHER METHOD: GRADIENT [STEEPEST] ASCENT/DESCENT:

  – Idea: at $\mathbf{x}$, f is increasing fastest in the direction of

$$\nabla f(\mathbf{x}) \;\equiv\; \left(\, \tfrac{\partial f}{\partial x_i}(\mathbf{x}) \,\right) \;\equiv\; \left(\, g_i(\mathbf{x}) \,\right)\,.$$

  – So, to maximise f, use update $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha\,\nabla f(\mathbf{x}_n)$, for small $\alpha > 0$.

  – (Or, to minimise, take $\alpha < 0$.)

  – Might "zig-zag" a lot, but <u>hopefully</u> will eventually find a (local) maximum.

- EXAMPLE: maximise $f(x1, x2) = -(x1 - 2)^2 - (x2 - 3)^4$.

  – If $\mathbf{x}_0 = (4, 4)$, then converges well. (file "Rsteep")

  – But if $\mathbf{x}_0 = (10, 10)$, then diverges. (again, file "Rsteep")

  – What to do?

- Use RETRACTION: if $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$, then the point $\mathbf{x}_{n+1}$ is REJECTED, and we instead replace $\mathbf{x}_{n+1}$ by $(\mathbf{x}_n + \mathbf{x}_{n+1})/2$ and try again. (Opposite if minimising.)

  – Fixes the problem. (file "Rsteep2")

- MANY OTHER ALGORITHMS, e.g. "simplex method" (see Gray, pp. 121–122) ...

  – Could spend an entire course ... entire PhD ... entire research career ...

- Point is: computers can optimise ...

- NON-LINEAR REGRESSION:
  - Computers can maximise/minimise "any" function. (e.g. R's "nlm")
  - So, can do "any" form of regression!
  - e.g. suppose we believe $Y = X^{\beta} + \text{error}$, with β unknown.
  - Then we observe some pairs (x_1, y_1), ... , (x_n, y_n).
  - Least squares: find value $\hat{\beta}$ for β to minimise $\sum_{i=1}^{n}(y_i - x_i^{\beta})^2$.
  - (Equivalent to MLE if errors are Normal$(0, \sigma^2)$.)
  - Can do this numerically [HW #2].

- Or, perhaps $Y = X^{\beta_1} + \beta_2\, e^X + \beta_3\, \sin(\beta_4 X) + \ldots$. Or perhaps ...

- SO WHERE DOES IT END??
  - Could use a model to determine a predictor function, $f(x)$.
  - Then, could measure "residual sum of squares", $RSS = \sum_i (y_i - f(x_i))^2$.
  - Does smallest RSS correspond to best model?

- PERHAPS NOT!
  - Given observations (x_1, y_1), ... , (x_n, y_n), with distinct $\{x_i\}$, consider:

$$f(x) \;=\; \sum_i y_i \, \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} \, .$$

  - Then f is a polynomial function, of degree $n - 1$.
  - Also, $f(x_i) = y_i$ for all i. So, $RSS = 0$.
  - Conclusion: Can find polynomial $\beta_{n-1}x^{n-1} + \beta_{n-2}x^{n-2} + \ldots + \beta_1 x + \beta_0$ which fits the n pairs *exactly*, i.e. $RSS = 0$.
  - But is that really a "good fit" of the data?

- Maybe not, since it's "cheating", i.e. using (x_i, y_i) as part of the fitting in order to predict y_i.
- What does "good fit" really MEAN, anyway?

- OVERFITTING:
  - Want our function to show the "underlying relationship" between X and Y, but not the "idiosyncracies" of the data we happen to observe.
  - Real goal: to PREDICT future values of Y, from future values of X.
  - How to test that??

——————— **END WEEK #4**———————

[Reminder: in-class test on Oct 23, room T.B.A., no aids, bring student card.]

[Reminder: homework #2 due Oct 16, by 6:10 p.m. sharp. hw2rev.pdf: (i) in Q1, defn of $f'(x)$ is missing two closing brackets; anyway $f(x)$ and $f'(x)$ are already defined in file "Riter"; also algorithm should terminate whenever $x_{n+1} = x_n$; (ii) in Q1(b), if one of the initial values equals 0, then replace it by 1 or 2; (iii) in Q3, use the "Illinois Method" modification to make sure the algorithm converges.]

[Web page: updated hw2.pdf ... R directory ... regrading policy ...]

[Office hour Friday (Oct 12) 2:30–3:30? Monday (Oct 15) 2:30–3:30? (Study buddies??)]

Summary of Previous Class:
∗ Illinois Method: modification of False Position method, where whenever the same interval endpoint is re-used, its function value is divided by two (for purposes of computing the next "c"). Important!
∗ Multi-dimensional Newton's method: $\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{H}^{-1}(\mathbf{x}_n)\,\mathbf{g}(\mathbf{x}_n)$.
∗ Steepest Ascent: $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha\,\nabla f(\mathbf{x}_n)$, for small $\alpha > 0$.
—— With retraction.
—— For descent (minimisation), take $\alpha < 0$.
∗ Regression: can use any model, then minimise $RSS = \sum_i (y_i - f(x_i))^2$.
∗ Can even fit a (degree $n-1$) polynomial f so $RSS = 0$.
∗ But is that really best? Overfitting? Prediction?

- TRAINING VERSUS TESTING DATA ("holdout method"):
  - Can divide up data (randomly) into training data (x_1, y_1), ... , (x_t, y_t), and testing data (x_{t+1}, y_{t+1}), ... , (x_n, y_n).
  - Fit values of the β_i (or whatever) using (x_1, y_1), ... , (x_t, y_t). Use these to obtain a predictor function, $f(x)$.
  - Then measure the ACCURACY of the predictor function by considering the "test data squared error", $\sum_{i=t+1}^{n}(y_i - f(x_i))^2$.
  - If a different model gives a smaller t.d.s.e., it's a "better fit".
  - Good method ... commonly used in CS/AI ...
  - However, need to have lots of data; some is "wasted" ...
- Alternative: CROSS-VALIDATION:
  - More precisely, "leave-one-out cross-validation (LOOCV)".
  - For each i, find a predictor function $f_{-i}(x)$ using only the OTHER data, i.e. (x_1, y_1), ... , (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), ... , (x_n, y_n).
  - Then predict y_i by $f_{-i}(x_i)$. (So, no "cheating".)
  - Then measure the accuracy by the "cross-validation sum of squares", $CVSS = \sum_{i=1}^{n}(y_i - f_{-i}(x_i))^2$.
  - NOTE: the final FIT is still done using the original predictor function $f(x)$, which minimises RSS for the model chosen; the functions $f_{-i}(x)$ are used ONLY for selecting which model gives the best predictions.
  - Observation: cross-validation can take lots of computer time, since fits of the data must be made n different times!
- A SIMPLE EXAMPLE:
  - Suppose observe three pairs: (2,5), (4,25), (6,35).
  - Consider three possible models: (a) $Y = \delta\, X + \text{error}$; (b) $Y = \beta_1 + \beta_2 X + \text{error}$;

(c) $Y = X^\gamma + \text{error}$.

- Which model is best? worst?
- Try fitting directly: (a) $\hat{\delta} = 5.71$, gives $RSS = 46.43$; (b) $\hat{\beta}_1 = -8.33$, $\hat{\beta}_2 = 7.5$, gives $RSS = 16.67$; (c) $\hat{\gamma} = 2.03$, gives $RSS = 78.99$.
- So does that really mean (b) is best, (a) is second, and (c) is worst??

- TRY CROSS-VALIDATION! (file "Rcross")
  - For (a), $f_{-1}(x) = 5.96\,x$, $f_{-2}(x) = 5.5\,x$, $f_{-3}(x) = 5.5\,x$. Get $CVSS = (5 - 2 * 5.96)^2 + (25 - 4 * 5.5)^2 + (35 - 6 * 5.5)^2 = 60.9$.
  - For (b), $f_{-1}(x) = 5 + 5x$, $f_{-2}(x) = -10 + 7.5\,x$, $f_{-3}(x) = -15 + 10x$. Get $CVSS = (5 - 2 * 5 - 5)^2 + (25 - 4 * 7.5 - (-10))^2 + (35 - 6 * 10 - (-15))^2 = 225$.
  - For (c), $f_{-1}(x) = x^{2.028}$, $f_{-2}(x) = x^{1.985}$, $f_{-3}(x) = x^{2.322}$. Get $CVSS = (5 - 2^{2.0278})^2 + (25 - 4^{1.985})^2 + (35 - 6^{2.322})^2 = 934.33$.
  - So, in this case, although (b) is the best direct fit of the data, (a) has smaller CVSS and thus is (supposedly!) better for future predictions. (And (c) is worst, by both measures.)

- VARIANCE OF ESTIMATORS:
  - Suppose want to estimate some parameter θ of a distribution, given a sample $x_1, x_2, \ldots, x_n$.
  - Suppose have some estimator, $\hat{\theta}(x_1, x_2, \ldots, x_n)$.
  - What is the uncertainty (variance) of the estimator?

- Example: suppose $\theta = \text{mean}(X_i)$, and $\hat{\theta} = \overline{x} \equiv \frac{1}{n}\sum_{i=1}^n x_i$.
  - Then $\text{Var}(\hat{\theta}) = \text{Var}(\overline{X}) = \frac{1}{n}\,\text{Var}(X_i)$.
  - And, can estimate this by $\frac{1}{n}\left(\frac{1}{n-1}\sum_{i=1}^n (x_i - \overline{x})^2\right)$.

- But what about other estimators?
  - median, quantile, "trimmed mean" (mean of middle $n - 2g$ order statistics), ...

- How to estimate their variances?

- THE JACKKNIFE:
  - Let $\hat{\theta}_{-i} = \hat{\theta}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, the same estimator except omitting the data point x_i (a bit like cross-validation).
  - Then let $\hat{\theta}_\bullet = \frac{1}{n}\sum_{i=1}^n \hat{\theta}_{-i}$.

- Jackknife estimator of variance: $\widehat{\mathrm{Var}}(\hat{\theta}) \equiv \frac{n-1}{n}\sum_{i=1}^n (\hat{\theta}_{-i} - \hat{\theta}_\bullet)^2$.

- e.g. if $\hat{\theta} = \overline{x}$, then $\hat{\theta}_{-i} = \frac{1}{n-1}\sum_{j\neq i} x_j$, and

$$\hat{\theta}_\bullet = \frac{1}{n}\sum_{i=1}^n \hat{\theta}_{-i} = \frac{1}{n(n-1)}\sum_{i=1}^n\sum_{j\neq i} x_j = \frac{1}{n(n-1)}\,(n-1)\sum_j x_j = \frac{1}{n}\sum_j x_j = \overline{x}\,.$$

  - Can be computed(!) that $\widehat{\mathrm{Var}}(\hat{\theta}) = \frac{1}{n(n-1)}\sum_{i=1}^n (x_i - \overline{x})^2$, the usual unbiased estimate of $\mathrm{Var}(\overline{x})$.

- Example: data = 10 fixed samples from Uniform[0,1].
  - jackknife estimate of variance = true sample variance = 0.00865. (file "Rjack")

- More generally, have Theorem (Math Stat?): if $\hat{\theta}(X_1, \ldots, X_n) = \frac{1}{n}\sum_{i=1}^n \phi(X_i)$ for some (finite-expectation) function ϕ, then $\mathbf{E}[\widehat{\mathrm{Var}}(\hat{\theta})] = \mathrm{Var}(\hat{\theta})$, i.e. $\widehat{\mathrm{Var}}(\hat{\theta})$ is an unbiased estimator of the variance of $\hat{\theta}$.

- For other estimators, the jackknife estimate of variance isn't always so accurate! (Efron & Stein 1981: if symmetric, on average it OVERESTIMATES var.)
  - e.g. median: $\hat{\theta} = \mathrm{median}(x_1, \ldots, x_n)$. data = 100 samples from Uniform[0,1]. Then true var around 0.002, and jackknife estimate of variance averages about 0.005, somewhat too high. (file "Rjack2")
  - e.g. truncated mean: $\hat{\theta} = \frac{1}{n-2g}\sum_{i=g+1}^{n-g} x_{(i)}$ for some $0 < g < n/2$, where $x_{(i)}$ is i'th order statistic. Take n=30, g=5, and data = 30 samples from N(0,1). Then both true var and jackknife estimate about 0.038, so quite good. [Check: when $g = 0$ it's very close.] (file "Rjack3")
  - e.g. max: $\hat{\theta} = \max(x_1, \ldots, x_n)$. [MLE for θ in Uniform$[0, \theta]$ dist.] data = 100

samples from Uniform[0,1]. Then true var around 0.0001, and jackknife estimate of variance averages about 0.00015, a little too high. (file "Rjack4")

——————— **END WEEK #5**———————

[Collect HW#2, and return HW#1. Reminder of regrading policy.]

[In-class test next week (Oct 23), in Mining Building (170 College Street) room 128. No aids. Bring student card. Photos might be taken.]

[Office hours: Monday Oct 22, 2:30-4.]

Summary of Previous Class:

∗ How to compare different models?

∗ Training versus testing data:

—— Fit to training data $(x_1, y_1), \ldots, (x_t, y_t)$.

—— Then test on testing data $(x_{t+1}, y_{t+1}), \ldots, (x_n, y_n)$.

∗ Cross-validation:

—— For each i, fit to all data except (x_i, y_i), then test on (x_i, y_i).

—— File "Rcross" (now in colour!).

∗ In both cases, final fit is to all data!

—— The training/testing or cross-validation is just to compare models.

∗ Estimating Variance of an Estimator (when true distribution completely unknown).

∗ Jackknife:

—— $\hat{\theta}_{-i} = \hat{\theta}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

—— $\hat{\theta}_\bullet = \frac{1}{n} \sum_{i=1}^n \hat{\theta}_{-i}$.

—— $\widehat{\mathrm{Var}}(\hat{\theta}) = \frac{n-1}{n} \sum_{i=1}^n (\hat{\theta}_{-i} - \hat{\theta}_\bullet)^2$.

∗ Tried on various examples (files "Rjack", "Rjack2", ...).

—— Lots of variability in individual estimates.

—— On average, sometimes very close, sometimes somewhat too high.

- Another approach: THE BOOTSTRAP.
  - Want to estimate $\mathrm{Var}(\hat{\theta}) = \mathbf{E}[(\hat{\theta} - \mathbf{E}(\hat{\theta}))^2]$.
  - However, the distribution of X_i, written $\mathcal{L}(X_i)$, is unknown.
  - Idea: find an estimate $\hat{F}_n$ of the unknown distribution $\mathcal{L}(X_i)$.

- Then compute $\mathrm{Var}(\hat{\theta})$ by assuming that $X_1, \ldots, X_n$ are i.i.d. $\sim \hat{F}_n$.

- EXAMPLE (parametric): suppose $X_1, \ldots, X_n \sim \text{Exponential}(\lambda)$, with λ unknown.
  - Parameter to be estimated: $\theta = \mathbf{E}(X_i) = 1/\lambda$.
  - Estimator: $\hat{\theta} = \overline{X}$.
  - Observe: $x_1, \ldots, x_n$.
  - Then could estimate λ by MLE $\equiv \hat{\lambda} = 1/\overline{x}$.
  - Hence, estimate the law of X_i by $\text{Exponential}(\hat{\lambda}) = \text{Exponential}(1/\overline{x})$.
  - Now, if $\hat{\theta}$ were the mean of n different i.i.d. $\text{Exponential}(\hat{\lambda})$ random variables, then its variance would be $\frac{1}{n}(\hat{\lambda})^{-2} = \frac{1}{n}(\overline{x})^2$.
  - So, estimate $\mathrm{Var}(\hat{\theta})$ by $\frac{1}{n}(\overline{x})^2$.
  - ("parametric bootstrap estimate of variance")

- But what if don't have a parametric model?

- Can estimate $\mathcal{L}(X_i)$ by the empirical distribution, $\hat{F}_n \equiv \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$.
  - [This is the distribution that gives mass $\frac{1}{n}$ to each point $x_1, \ldots, x_n$.]
  - [That is, we estimate $\mathbf{P}[X \le x]$ by $\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^n I(x_i \le x)$.]
  - Then estimate $\mathbf{E}(\hat{\theta})$, and then $\mathrm{Var}(\hat{\theta})$, by computing them according to $\hat{F}_n$.
  - ("nonparametric bootstrap estimate of variance")

- EXAMPLE: observe $x_1, \ldots, x_n$, from unknown distribution.
  - Parameter to be estimated: $\theta = \mathbf{E}(X_i)$. Estimator: $\hat{\theta} = \overline{X}$.
  - Then $\mathbf{E}_{\hat{F}_n}(X_1) = \frac{1}{n}\sum_{i=1}^n x_i = \overline{x}$.
  - So, $\mathrm{Var}_{\hat{F}_n}(\hat{\theta}) = \mathrm{Var}_{\hat{F}_n}(\overline{X}) = \frac{1}{n}\mathrm{Var}_{\hat{F}_n}(X_1) = \frac{1}{n}\mathbf{E}_{\hat{F}_n}[(X_1 - \overline{x})^2] = \frac{1}{n^2}\sum_{i=1}^n (x_i - \overline{x})^2$.
  - So, can estimate $\mathrm{Var}(\hat{\theta})$ by $\frac{1}{n^2}\sum_{i=1}^n (x_i - \overline{x})^2$.
  - (Similar to usual unbiased estimator $\frac{1}{n(n-1)}\sum_{i=1}^n (x_i - \overline{x})^2 \ldots$ but slightly biased.)

– ("Exact nonparametric bootstrap estimator of variance", i.e. no "resampling".)

- BUT WHAT IF CAN'T COMPUTE $\mathrm{Var}_{\hat{F}_n}(\hat{\theta})$? (Usual case.)
  - Idea: Estimate $\mathrm{Var}_{\hat{F}_n}(\hat{\theta})$ by Monte Carlo simulation!
  - First, generate a sample $x_1^{*(1)}, x_2^{*(1)}, \ldots x_n^{*(1)} \sim \hat{F}_n$ (i.i.d.).
  - (So, each $x_i^{*(1)}$ is equal to some x_j, where j is chosen uniformly in $\{1, 2, \ldots, n\}$, sampled with repetition. "resampling")
  - Then can compute $\hat{\theta}(x^{*(1)}) \equiv \hat{\theta}(x_1^{*(1)}, x_2^{*(1)}, \ldots x_n^{*(1)})$.
  - Repeat this entire procedure B times, generating samples $\mathbf{x}^{*(1)}, \mathbf{x}^{*(2)}, \ldots, \mathbf{x}^{*(B)}$, each consisting of n i.i.d. samples from $\hat{F}_n$.
  - Then can estimate $\mathbf{E}(\hat{\theta})$ by $\theta^* = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}(\mathbf{x}^{*(b)})$.
  - Then can estimate $\mathrm{Var}(\hat{\theta})$ by $\mathrm{Var}^*(\hat{\theta}) = \frac{1}{B-1}\sum_{b=1}^{B} \left(\hat{\theta}(\mathbf{x}^{*(b)}) - \theta^*\right)^2$, the usual unbiased estimate of the variance of $\hat{\theta}$ according to the distribution $\hat{F}_n$.
  - ("Bootstrap resampling estimate of variance", a.k.a. "the bootstrap".)
- BOOTSTRAP (RESAMPLING) FOR THE PREVIOUS EXAMPLES (B = 1000):
  - mean (with 10 fixed observations): bootstrap estimate of variance around 0.008, quite close to true value of 0.00865. (file "Rboot")
  - median: bootstrap estimate of variance around 0.002, quite close to true value (also around 0.002), though somewhat variable. (file "Rboot2")
  - truncated mean (n=30, g=5): bootstrap estimate of variance varies between about 0.0027 and 0.005, reasonably close to true value (around 0.004). (file "Rboot3")
  - max: bootstrap estimate of variance somewhat unstable, but usually fairly close to true value (around 0.0001), though sometimes much too small. (file "Rboot4")
- So, bootstrap (resampling) fairly robust and accurate (but computationally intensive).
- WHAT ABOUT BIAS ESTIMATION?

- e.g. $\hat{\theta} = \max(x_1, \ldots, x_n)$, MLE for θ in Uniform$[0, \theta]$ dist.
- data = 100 samples from Uniform[0,1], i.e. true $\theta = 1$.
- Biased estimator: bias $= \mathbf{E}[\hat{\theta} - \theta] < 0$, since always $\hat{\theta} < \theta$.
- How much bias??

- Jackknife estimator of bias: $\hat{b}(\hat{\theta}) \equiv (n-1)(\hat{\theta}_\bullet - \hat{\theta})$.
  - Theorem (Math Stat?): if $\mathbf{E}(\hat{\theta}) = \theta + \frac{a(\theta)}{n}$, then $\mathbf{E}[\hat{b}(\hat{\theta})] = \mathbf{E}[\hat{\theta} - \theta]$ = true bias, i.e. it's an unbiased estimator of the bias (!).
  - More generally, if $\mathbf{E}(\hat{\theta}) = \theta + \frac{a(\theta)}{n} + O(1/n^2)$, then $\mathbf{E}[\hat{b}(\hat{\theta})] = \mathbf{E}[\hat{\theta} - \theta] + O(1/n^2)$, i.e. unbiased to leading order.

- Bootstrap (resampling) estimator of bias:

$$b^*(\hat{\theta}) \;=\; \theta^* - \hat{\theta} \;\equiv\; \Big(\frac{1}{B}\sum_{b=1}^{B} \hat{\theta}(\mathbf{x}^{*(i)})\Big) - \hat{\theta}\,.$$

- For above "max" example ($\theta = 1$, $n = 100$):
  - Actual bias (from simulations) is around -0.01, and average jackknife bias estimate of bias is also around -0.01, quite good (at least on average). (file "Rjack5")
  - Bootstrap estimate of bias ranges from -0.002 to -0.009, so not bad, but somewhat unstable, and also too small (in magnitude) on average. (file "Rboot5")
  - Can also compute the exact bias numerically: -0.00990099, very close to -0.01. (see bottom of file "Rboot5")
  - (Actually, "max" is known to be a challenging estimator for the bootstrap, since often the resampled data has the same max.)

- NOTE: jackknife and bootstrap also used for many other things besides variance and bias estimation, not covered here ...

[Final reminder: In-class test next week in Mining Building, room 128!!]

——————— END WEEK #6———————

[IN-CLASS TEST!!]

——————— **END WEEK #7**———————

[Return in-class test & hw#2 ... check grades list ... NO FRIVILOUS REGRADES!]

[Assign hw#3. ... Final exam 7–10pm on Wed Dec 12, NR25.]

[Another on-line reference, by Galin Jones, now available on web page ... Books useful?]

[Keep break to 15 minutes!]

- (NONPARAMETRIC) DENSITY ESTIMATION:
- Suppose observe $x_1, \ldots, x_n$.
- Want to estimate the density, $f(x)$, describing their probabilities.
- (Nonparametric, i.e. the density is not assumed to be from a specific family.)
- Running example: 100 fixed data points (in file "Rdens").
- One approach: a HISTOGRAM.
  - For some $h > 0$, break up range into intervals $[jh,\, (j+1)h)$.
  - Interval containing x is $\big[\text{floor}(x/h)\, h,\ (\text{floor}(x/h)+1)\, h\big)$.
  - Would expect about $n\, h\, f(x)$ points in this interval.
  - So, use approximation:

$$\hat{f}(x) \ = \ \frac{1}{nh} \#\{i : \text{floor}(x/h)\, h \le x_i < (\text{floor}(x/h)+1)\, h\}\,.$$

    (file "Rdens")

  - (Could instead assign this value to interval MIDPOINTS, connected by lines.)
- Another approach: CENTERED INTERVALS.
  - For some $h > 0$, consider interval $[x-h,\, x+h]$.
  - Would expect about $2\, n\, h\, f(x)$ points in this interval.

– So, use approximation:

$$\hat{f}(x) \;=\; \frac{1}{2nh}\#\{i : |x - x_i| < h\}\,.$$

(file "Rdens")

- Another (more general) approach: KERNEL DENSITY ESTIMATOR.
  - For some $h > 0$, and some probability density function ("kernel") K, pretend each point x_i came from its own density K, scaled by h and centered at x_i, i.e. $\frac{1}{h}\,K\left(\frac{x-x_i}{h}\right)$.
  - So, use approximation:

$$\hat{f}(x) \;=\; \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)\,.$$

  - (Centered intervals correspond to $K(\cdot) = \text{Uniform}[-1, 1]$.)
  - Often take K to be density of Normal$(0, 1)$. (file "Rdens")
- Each of these approaches has a parameter h, the BANDWIDTH.
  - How does changing h affect the estimate? (file "Rdens")
- What is "optimal" bandwidth?
  - Ideally, want $h \to 0$ but $nh \to \infty$ (then get perfect estimate).
  - But what to do if n fixed? (Usual situation.)
- One approach: try to (approximately) minimise the Mean Integrated Squared Error,

$$MISE \;=\; \mathbf{E}\left[\int_{-\infty}^{\infty}\left(\hat{f}(x) - f(x)\right)^2 dx\right]\,,$$

where the expectation is taken over $x_1, \ldots, x_n \sim f$.

- [In running example, Summed Squared Error is about 2.30 for R's version, 1.77 for kernel dens ($h = 0.2$), 2.09 for centered invervals ($h = 0.2$).]
- Theorem: Kernel Density Estimator's MISE is minimised, to within $O(1/nh,\ h^4)$, at:

$$h \;=\; h_{opt} \;=\; \left(\frac{\int_{-\infty}^{\infty} K(x)^2 dx}{n\,\text{Var}(K)^2\,\int_{-\infty}^{\infty} f''(x)^2 dx}\right)^{1/5}\,. \qquad (*)$$

- (Good approximation if h small but nh large; usual case.)
- Theorem not too useful, since $f'(x)$ unknown.
- But does show that $h_{opt} = O(n^{-1/5})$.

- RULE OF THUMB approximation [based on a Normal approximation in $(*)$] is: $h_{opt} \approx (4/3n)^{1/5}\,\mathrm{sd}(x_1, \ldots, x_n)$.
  - In running example, gives $h \doteq 0.592$. (Gives summed error of 2.79, worse than 1.77 with $h = 0.2$.)
  - Alternative: first use this approximation to estimate $\int_{-\infty}^{\infty} f'(x)^2 dx$, then use result to estimate h_{opt} from $(*)$. Many options ...
  - (Or can just "eyeball" the result ... useful ... not automated though ...)
- Much more known about density estimation, but we'll end here ...
- NUMERICAL INTEGRATION:
- How to compute expected values from complicated densities?
- (Motivation: Bayesian statistics ... next.)
- EXAMPLE: Suppose X and Y are two random variables with joint density given by $f_{X,Y}(x,y) = C\,g(x,y)$ for $0 \le x, y \le 1$ (with $f_{X,Y}(x,y) = 0$ for other x, y), for appropriate constant C, where

$$g(x,y) \;=\; x^2\; y^3\; \sin(xy)\; \cos(\,\sqrt{xy}\,)\; \exp(x^2 + y)\,.$$

  - What is, say, $\mathbf{E}(X)$??
  - Well, here

$$C \;=\; \left(\int_0^1 \int_0^1 g(x,y)\,dx\,dy \right)^{-1}\,.$$

  - Then

$$\mathbf{E}(X) \;=\; \int_0^1 \int_0^1 x\, f_{X,Y}(x,y)\,dx\,dy \;=\; \frac{\int_0^1 \int_0^1 x\, g(x,y)\,dx\,dy}{\int_0^1 \int_0^1 g(x,y)\,dx\,dy}\,.$$

- How to compute this??

- No analytic solution ...
- Instead, use numerical integration ...

- Break up region $[0,1] \times [0,1]$ into grid:
  - Let M be a large integer.
  - Let $A_{ij} = [(i-1)/M,\ i/M) \times [(j-1)/M,\ j/M)$, for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, M$ (so, M^2 sub-regions).
  - Then $[0,1] \times [0,1] = \bigcup_{i=1}^{M} \bigcup_{j=1}^{M} A_{ij}$.
  - And, for large M, g is *approximately* constant over A_{ij}.
  - So, e.g.

$$\int_0^1 \int_0^1 g(x,y)\,dx\,dy = \sum_{i=1}^{M}\sum_{j=1}^{M} \int\int_{A_{ij}} g(x,y)\,dx\,dy \approx \sum_{i=1}^{M}\sum_{j=1}^{M} g(i/M,\, j/M)\,(1/M^2)\,.$$

    ("top-right sum")

- In above, can instead replace $g(i/M,\, j/M)$ by:
  - $g((i-1)/M,\, (j-1)/M)$ ("bottom-left sum");
  - $g((i-\frac{1}{2})/M,\, (j-\frac{1}{2})/M)$ ("midpoint sum");
  - $\frac{1}{4}\Big(g((i-1)/M,\, (j-1)/M) + g((i-1)/M,\, j/M) + g(i/M,\, (j-1)/M) + g(i/M,\, j/M)\Big)$ ("corner-averaged sum; equivalent to Trapezoidal Rule with $m = 1$");
  - Trapezoidal Rule (polynomial interpolation of degree m);
  - Simpson's Rule (replace g by quadratic function which agrees at middle and endpoints);
  - $\inf_{\substack{(i-1)/M \le x \le i/M \\ (j-1)/M \le y \le j/M}} g(x,y)$ ("lower sum"; always less than true integral);
  - $\sup_{\substack{(i-1)/M \le x \le i/M \\ (j-1)/M \le y \le j/M}} g(x,y)$ ("upper sum"; always more than true integral); etc.
  - (Differences not too large if grid size small.)

- Similarly for $\int_0^1 \int_0^1 x\, g(x,y)\,dx\,dy$, etc.

  - So, that way, can compute expected values!
  - (Homework #3!)

- Choice of M? Depends on function? Tricky issue!
- For, say, $\int_0^5 \int_0^5 \cdots$, need $(5M)^2$ sub-regions instead.
  - Not much more difficult ...
- For e.g. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots$, need to "cut off" at $\int_a^b \int_a^b \cdots$ for appropriate a and b.
  - e.g. $a = -1000$ and $b = 1000$ (?).
  - a and b should depend on function ... how?
  - Requires $((b-a)M)^2$ sub-regions – could be too many.
  - Another tricky issue!
- Higher dimensional integrals, e.g. $\int_0^1 \int_0^1 \int_0^1 \int_0^1 \cdots$?
  - Similar, but have d loops, so M^d sub-regions.
  - more time-consuming – exponentially slow. ("curse of dimensionality")
- Other solutions (including Monte Carlo, MCMC) coming later!

——————— END WEEK #8———————

[Reminder: HW #3 due next Tuesday, 6:10 p.m.]

[File "Rhw3Q1data" updated on Nov. 2.]

[Office hours 2:30 – 3:30 on Friday Nov 9 and Monday Nov 12?]

Summary of Previous Class:

∗ Density estimation:

—— (scaled) histogram

—— centered intervals

—— kernel density estimator

—— (centered intervals = kernel density estimator with K = Unif[-1,1])

$*$ Choice of bandwidth h:

—— Ideal: $h \to 0$ but $nh \to \infty$.

—— measure of fit: MISE (but requires knowledge of true dens)

—— RULE OF THUMB: $h \approx 1.06\, n^{-1/5}\, sd(x_1, \ldots, x_n)$.

$*$ Numerical integration (to e.g. compute expected values):

—— Break up region into M^d small sub-regions (squares/cubes)

—— Approximate integrand by e.g. constant on each sub-region

—— top-right / bottom-left / midpoint / corner-averaged / lower / upper / Trapezoidal / Simpson's ...

$*$ On larger regions, need more sub-regions.

$*$ On infinite regions, need some sort of "cut-off".

—— e.g. if cut off each coordinate at $[a, b]$ then need $((b-a)M)^d$ sub-regions!

- Another approach: MONTE CARLO INTEGRATION:
- EXAMPLE: Want to compute $\int_0^1 \int_0^1 g(x,y)\, dx\, dy$.
  - Regard this as $\mathbf{E}[g(X,Y)]$, where X, Y i.i.d. $\sim$ Uniform$[0,1]$.
- Then do Monte Carlo:
  - Sample $X_1, \ldots, X_M, Y_1, \ldots, Y_M \sim$ Uniform$[0,1]$.
  - Estimate $\int_0^1 \int_0^1 g(x,y)\, dx\, dy$ by $\frac{1}{M} \sum_{i=1}^M g(X_i, Y_i)$.
  - Since $\{g(X_i, Y_i)\}$ i.i.d., get usual estimate of standard error:

$$se \;=\; M^{-1/2}\, sd\big(g(X_1,Y_1), \ldots, g(X_M,Y_M)\big)\,.$$

  - e.g. $g(x,y) = \cos(\,\sqrt{xy}\,)$. (file "RMCint") Easy!
  - Get about 0.88 ± 0.003 ... Mathematica gives 0.879544.
- What about $\int_0^5 \int_0^4 g(x,y)\, dy\, dx$?
  - One way to think about this is,

$$\int_0^5 \int_0^4 g(x,y)\, dy\, dx \;=\; \int_0^5 \int_0^4 5 \cdot 4 \cdot g(x,y)\, (1/4) dy\, (1/5) dx \;=\; \mathbf{E}(5 \cdot 4 \cdot g(X,Y))\,,$$

    where $X \sim$ Uniform$[0,5]$ and $Y \sim$ Uniform$[0,4]$.

- So, instead let $X_i \sim \text{Uniform}[0,5]$, and $Y_i \sim \text{Uniform}[0,4]$.
- Estimate becomes $\frac{1}{M}\sum_{i=1}^{M} 5 \cdot 4 \cdot g(X_i, Y_i)$. (file "RMCint2")
- Standard error: $se = M^{-1/2}\ sd\big(5 \cdot 4 \cdot g(X_1, Y_1),\ \ldots,\ 5 \cdot 4 \cdot g(X_M, Y_M)\big)$.
- Get about -4.1 ± 0.4 ... [with $M = 10^6$, get -4.1 ± 0.01 ... if M gets much larger, R has problems] ... Mathematica gives -4.11692.

- MORE GENERALLY, whenever can write integral as an expected value, can do Monte Carlo integration.
- e.g. $\int_0^1 \int_0^\infty h(x,y)\, dy\, dx$, where $h(x,y) = e^{-y^2}\ \cos(\sqrt{xy}\,)$.
  - Can't do usual "uniform" Monte Carlo.
  - Instead, write this as $\int_0^1 \int_0^\infty (e^y\, h(x,y))\, e^{-y}\, dy\, dx$.
  - This is the same as $\mathbf{E}[e^Y\, h(X,Y)]$, where $X \sim \text{Uniform}[0,1]$ and $Y \sim \text{Exponential}(1)$.
  - So, estimate it by $\frac{1}{M}\sum_{i=1}^{M} e^{Y_i} h(X_i, Y_i)$, where $X_i \sim \text{Uniform}[0,1]$ and $Y_i \sim \text{Exponential}(1)$ (i.i.d.).
  - File "RMCint3" ... get about 0.75 ± 0.01 ... Mathematica gives 0.767211.
- Alternatively, could write integral as $\int_0^1 \int_0^\infty (\frac{1}{5}\, e^{5y}\, h(x,y))\, (5\, e^{-5y})\, dy\, dx$.
  - Then, estimate it by $\frac{1}{M}\sum_{i=1}^{M} \frac{1}{5} e^{5Y_i} h(X_i, Y_i)$, where $X_i \sim \text{Uniform}[0,1]$ and $Y_i \sim \text{Exponential}(5)$ (i.i.d.).
  - File "RMCint4" ... get about 0.75 ± 0.04 ... larger standard error ...
  - If replace 5 by 1/5, get about 0.75 ± 0.05 ... even worse.
  - So which choice is best??
  - Whichever one minimises the standard deviation! ($\lambda \approx 1.5$?)
- Monte Carlo integration's standard deviation goes down like $M^{-1/2}$.
  - "Pretty good" convergence, but not great.
  - And, depends on the standard deviation of each individual observation.

– More coming!

- SO WHO CARES about integration, anyway?
- Motivation: BAYESIAN STATISTICS:
  - Have unknown parameter(s) θ, and a model for how the distribution of the data Y depends on θ: $\mathcal{L}(Y \,|\, \theta)$.
  - Have a <u>prior</u> distribution, representing our "initial" (subjective?) probabilities for θ: $\mathcal{L}(\theta)$.
  - Combining these gives a full joint distribution for θ and Y, i.e. $\mathcal{L}(\theta, Y)$.
  - Then <u>posterior</u> distribution of θ, $\pi(\theta)$, is then the <u>conditional</u> distribution of θ, <u>conditioned</u> on the observed data y, i.e. $\pi(\theta) = \mathcal{L}(\theta \,|\, Y = y)$.
- SIMPLE EXAMPLE: WEIGHT OF MOOSE:
  - Model: Average moose weighs θ kilograms (kg), with θ unknown.
  - Our best guess for prior distribution of θ is (say) $N(500, 100^2)$.
  - An individual moose's weight, in kg, has distribution (say) $N(\theta, 50^2)$.
  - Going to weigh J mooses, and record weights $Y_1, Y_2, \ldots, Y_J$.
- So, joint density of $(\theta, Y_1, Y_2, \ldots, Y_J)$ is:

$$f(\theta, y_1, \ldots, y_J) \;=\; f(\theta)\; f(y_1, \ldots, y_J \,|\, \theta) \;=\; N(500, 100^2;\, \theta) \prod_{i=1}^{J} N(\theta, 50^2;\, y_i)$$

$$= \frac{1}{\sqrt{2\pi} \times 100} \exp\left[-(\theta - 500)^2/(2 \times 100^2)\right] \prod_{i=1}^{J} \Big(\frac{1}{\sqrt{2\pi} \times 50} \exp\left[-(y_i - \theta)^2/(2 \times 50^2)\right] \Big)$$

$$= (2\pi)^{-(J+1)/2} 100^{-1} 50^{-J/2} \exp\Big[-(\theta - 500)^2/(2 \times 100^2)) - \big(\sum_i (y_i - \theta)^2 \big)/(2 \times 50^2) \Big] .$$

  - ("joint density = prior times likelihood")
- Then posterior density of θ equals this density conditioned on the observed $\{y_i\}$, which is proportional to simply treating the $\{y_i\}$ as <u>constants</u>, so it is equal to:

$$C_1 \exp\Big[-\theta^2 \Big(\frac{1}{2 \times 100^2} + \frac{J}{2 \times 50^2} \Big) + \theta \Big(\frac{500}{100^2} - \sum_i \frac{y_i}{50^2} \Big) + C_2 \Big]$$

$$= C_3 \exp\Big[-(\theta - m)^2/2\sigma^2 \Big],$$

where

$$\sigma^2 = \frac{1}{\frac{1}{100^2} + \frac{J}{50^2}},$$

and

$$m = \Big(\frac{500}{100^2} + \sum_i \frac{y_i}{50^2} \Big) \sigma^2 = \frac{\frac{500}{100^2} + \frac{\sum_i y_i}{50^2}}{\frac{1}{100^2} + \frac{J}{50^2}}.$$

  - ("posterior = prior times likelihood, normalised")
  - (Check: if $y_i \equiv 500$, then $m = 500$.)

- Since this must be a probability density, it must work out that $C_3 = (2\pi\sigma^2)^{-1/2}$, so the posterior distribution for θ must be $N(m, \sigma^2)$. Phew!
  - This posterior distribution represents our probabilities for θ, given the observed data $y_1, y_2, \ldots, y_n$.
  - Then we can compute the posterior mean, posterior variance, posterior probabilities, etc., for θ.
  - e.g. posterior mean of θ = mean of $N(m, \sigma^2) = m$, etc.
  - Don't need any numerical integration.
- But that was an extremely simple example!
- Less simple example: VARIANCE COMPONENTS MODEL:
  - A lake has some pollution concentration μ (unknown).
  - Have K different extractions from the lake.
  - Extraction i has pollution concentration θ_i (unknown).
  - Observations: $Y_{i1}, \ldots, Y_{iJ}$ are J different measurements of θ_i $(1 \le i \le K)$.
  - Want to estimate $\mu, \theta_1, \ldots, \theta_K$, and the variances V between extractions and W between measurements.

- Statistical model: assume

$$\theta_i \sim N(\mu, V); \qquad (1 \le i \le K).$$

and

$$Y_{ij} \sim N(\theta_i, W); \qquad (1 \le i \le K; \quad 1 \le j \le J).$$

  - This defines a probability model for $V, W, \mu, \theta_1, \ldots, \theta_K, Y_{ij}$:

$$\begin{array}{ccccc} & & \mu & & \\ & \swarrow & \downarrow & \searrow & \\ & \theta_1 & \ldots\ldots & \theta_K & \qquad \theta_i \sim N(\mu, V) \\ \swarrow & & & & \searrow \\ Y_{11}, \ldots, Y_{1J} & & \ldots\ldots & & Y_{K1}, \ldots, Y_{KJ} \qquad Y_{ij} \sim N(\theta_i, W) \end{array}$$

- Prior distributions ("conjugate"):

$$V \sim IG(a_1, b_1); \qquad W \sim IG(a_2, b_2); \qquad \mu \sim N(a_3, b_3)\,,$$

  where a_i, b_i known constants, and $IG(a, b)$ is "inverse gamma" distribution, with density $\frac{b^a}{\Gamma(a)}\, e^{-b/x}\, x^{-a-1}$ for $x > 0$.

- Then for $V, W > 0$, joint density is:

$$\begin{aligned} & f(V, W, \mu, \theta_1, \ldots, \theta_K, Y_{11}, Y_{12}, \ldots, Y_{KJ}) \\ = \; & C \left(e^{-b_1/V} V^{-a_1-1}\right) \left(e^{-b_2/W} W^{-a_2-1}\right) \left(e^{-(\mu-a_3)^2/2b_3}\right) \times \\ & \quad \times \left(\prod_{i=1}^{K} V^{-1/2} e^{-(\theta_i-\mu)^2/2V}\right) \left(\prod_{i=1}^{K}\prod_{j=1}^{J} W^{-1/2} e^{-(Y_{ij}-\theta_i)^2/2W}\right) \\ = \; & C e^{-b_1/V} V^{-a_1-1} e^{-b_2/W} W^{-a_2-1} e^{-(\mu-a_3)^2/2b_3} V^{-K/2} W^{-JK/2} \times \\ & \quad \times \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V - \sum_{i=1}^{K}\sum_{j=1}^{J} (Y_{ij} - \theta_i)^2/2W\right]. \end{aligned}$$

——————— END WEEK #9———————

[Collect HW#3. Assign HW#4.]

Summary of Previous Class:

∗ Monte Carlo integration:

—— Express an integral as some expected value.

—— (e.g. $X \sim \text{Uniform}[0,1]$, $Y \sim \text{Exponential}(5)$, ...)

—— Then do usual Monte Carlo estimate of mean.

—— Get standard error too.

—— Quite easy ... but *se* can be large.

∗ Bayesian statistics:

—— Have model $\mathcal{L}(Y \,|\, \theta)$, and prior $\mathcal{L}(\theta)$.

—— Gives joint distribution $\mathcal{L}(\theta,\, Y)$.

—— Then posterior $= \pi(\cdot) = \mathcal{L}(\theta \,|\, Y)$.

—— Moose example: $\pi(\cdot) = N(m, v)$.

—— Variance components (lake) model: joint density

$$= Ce^{-b_1/V}V^{-a_1-1}e^{-b_2/W}W^{-a_2-1}e^{-(\mu-a_3)^2/2b_3}V^{-K/2}W^{-JK/2} \times$$
$$\times \exp\left[-\sum_{i=1}^{K}(\theta_i-\mu)^2/2V - \sum_{i=1}^{K}\sum_{j=1}^{J}(Y_{ij}-\theta_i)^2/2W\right].$$

∗ Posterior distribution $\pi(\cdot)$ is then the law of $(V, W, \mu, \theta_1, \ldots, \theta_K)$, conditional on the observed data Y_{ij}. (Equivalent to treating the Y_{ij} as constants.)

∗ (Density function in file "Rvarcomp" ... including on log scale ...)

∗ Computations difficult! What to do?

- One solution: Markov chain Monte Carlo (MCMC).
- Needs lots of random numbers (like regular Monte Carlo).
  - How do computers generate them?
- PSEUDO-RANDOM NUMBERS:
- Goal: generate an i.i.d. sequence $U_1, U_2, U_3, \ldots \sim \text{Uniform}[0,1]$.
- One method: LINEAR CONGRUENTIAL GENERATOR.
  - Choose (large) positive integers m, a, and b.

  - Start with a "seed" value, k_0. (e.g., current time in milliseconds)
  - Then, recursively, $k_n = (ak_{n-1} + b) \bmod m$, i.e. k_n = remainder when $ak_{n-1} + b$ is divided by m.
  - So, $0 \leq k_n \leq m - 1$.
  - Then let $U_n = k_n/m$.
  - Then $\{U_n\}$ will "seem" to be approximately i.i.d. $\sim$ Uniform$[0, 1]$. (file "Rrng")
  - How well does it work? (HW#4!)
- Choice of m, a, and b?
- Many issues:
  - need m large (so many possible values);
  - need a large enough that no obvious "pattern" between U_{n-1} and U_n.
  - need b to avoid short "cycles", to avoid repetition.
  - many statistical tests, to try to see which choices provide good randomness, avoid correlations, etc. (e.g. "diehard tests")
  - One common "good" choice: $m = 2^{32}$, $a = 69,069$, $b = 23,606,797$.
  - (More generally, can use "bitwise exclusive-or" between different pseudorandom binary tuples to increase randomness ... "generalised feedback shift register (GFSR)", "Mersenne-Twister", ...)
  - (R implementation: see "?.Random.seed" ...)
- Not "really" random, just "pseudorandom" ...
  - Can cause problems!
  - Will fail certain statistical tests ...
  - Some implementations also use external randomness, e.g. current temperature of computer's CPU.

  - Or the randomness of *quantum mechanics*, e.g. `www.fourmilab.ch/hotbits`.
  - But for most purposes, standard pseudorandom numbers are pretty good ...

- What about OTHER DISTRIBUTIONS?
- Once we have $U_1, U_2, \ldots$ i.i.d. $\sim$ Uniform$[0,1]$ (at least approximately), how do we generate other distributions?
- With transformations, using "change-of-variable" theorem!
- e.g. to make $X \sim \text{Uniform}[L, R]$, set $X = (R-L)U_1 + L$.
- e.g. to make $X \sim \text{Bernoulli}(p)$, set

$$X = \begin{cases} 1, & U_1 \le p \\ 0, & U_1 > p \end{cases}$$

- e.g. to make $Y \sim \text{Binomial}(n, p)$, either set $Y = X_1 + \ldots + X_n$ where

$$X_i \;=\; \begin{cases} 1, & U_i \le p \\ 0, & U_i > p \end{cases},$$

  or set

$$Y \;=\; \max\left\{ j \,:\, \sum_{k=0}^{j-1} \binom{n}{k} p^k (1-p)^{n-k} \;\le\; U_1 \right\}.$$

- More generally, to make $\mathbf{P}(Y = x_i) = p_i$ for any $x_1 < x_2 < x_3 < \ldots$, where $\sum_i p_i = 1$, simply set

$$Y \;=\; \max\{x_j \,;\, \sum_{k=0}^{j-1} p(x_k) \;\le\; U_1\}\,.$$

- e.g. to make $Z \sim \text{Exponential}(1)$, set $Z = -\log(U_1)$.
  - (So, $\mathbf{P}(Z > x) = \mathbf{P}(-\log(U_1) > x) = \mathbf{P}(\log(U_1) < -x) = \mathbf{P}(U_1 < e^{-x}) = e^{-x}$.)
  - (Then to make $W \sim \text{Exponential}(\lambda)$, set $W = Z/\lambda$.)
- What about normal dist.? By multidimensional change-of-variable theorem, if

$$X \;=\; \sqrt{2\,\log(1/U_1)}\;\cos(2\pi U_2)\,,$$

$$Y \;=\; \sqrt{2\,\log(1/U_1)}\;\sin(2\pi U_2)\,,$$

then $X, Y \sim N(0,1)$ (independent!).

- Another approach: "INVERSE CDF METHOD":
  - Suppose want $\mathbf{P}(X \le x) = F(x)$. ("CDF")
  - For $0 < t < 1$, set $F^{-1}(t) \;=\; \min\{x\,;\; F(x) \ge t\}$. ("inverse CDF")
  - Then set $X = F^{-1}(U_1)$.
  - Then $X \le x$ if and only if $F(x) \le U_1$.
  - So, $\mathbf{P}(X \le x) \;=\; \mathbf{P}\big(F(x) \le U_1\big) \;=\; F(x)$.
- etc.
- So, generating (pseudo)random numbers is easy. How to use this?
- Monte Carlo algorithms/integration (already discussed).
- MARKOV CHAIN MONTE CARLO (MCMC)!
- Suppose have complicated, high-dimensional density π.
- Want samples $X_1, X_2, \ldots \sim \pi$. (Then can do Monte Carlo.)
- Idea: define a Markov chain (random process) $X_1, X_2, \ldots$, so for large n, $X_n \approx \pi$.
- METROPOLIS ALGORITHM (1953):
  - Choose some initial value X_0 (perhaps multi-dimensional, perhaps random).
  - Then, given X_{n-1}, choose a proposal move $Y_n \sim MVN(X_{n-1},\, I)$ (say).
  - Let $\alpha_n = \pi(Y_n)\,/\,\pi(X_{n-1})$, and $U_n \sim \text{Uniform}[0,1]$.
  - Then, if $U_n < \alpha_n$, set $X_n = Y_n$ ("accept"), otherwise set $X_n = X_{n-1}$ ("reject").
  - THEOREM: $\lim_{n\to\infty} \mathbf{P}(X_n \in A) = \int_A \pi(x)\,dx$, and $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} h(X_i) = \mathbf{E}_\pi(h) \equiv \int h(x)\,\pi(x)\,dx$.
  - (Chain is "reversible" and "irreducible" and "aperiodic" ... STA447/2106 ...)

- So, for large n, have $X_n \approx \pi$. ("rwm.html" Java applet)
- Note: only need to compute $\pi(Y_n) \,/\, \pi(X_{n-1})$, so multiplicative constants cancel.

- HOW IS THIS USEFUL?
- Can estimate, say, $\mathbf{E}_\pi(h) \equiv \int h(x)\,\pi(x)\,dx$ by:

$$\frac{1}{M-B} \sum_{i=B+1}^{M} h(X_i)\,,$$

where B ("burn-in") chosen large enough so $X_B \approx \pi$, and M chosen large enough to get good Monte Carlo estimates.

- EXAMPLE: $\pi(x_1, x_2) = C\,|\cos(\sqrt{x_1\,x_2}\,)|\; I(0 \le x_1 \le 5,\, 0 \le x_2 \le 4)$.
  - Want to compute $\mathbf{E}_\pi(h)$, where $h(x, y) = e^{x_1} + (x_2)^2$.
  - Then $\pi(Y) \,/\, \pi(X) = C\,|\cos(\sqrt{y_1 y_2}\,)| \,/\, |\,C\,\cos(\sqrt{x_1 x_2}\,)| = |\cos(\sqrt{y_1 y_2}\,)| \,/\, |\cos(\sqrt{x_1 x_2}\,)|$, so don't need to know "C".
  - Metropolis algorithm ... works ... gets between about 34 and 44 ... but large uncertainty ... (file "Rmet") (Mathematica gets 38.7044)

- OPTIMAL SCALING:
  - Can change proposal distribution to $Y_n \sim MVN(X_n,\, \sigma^2 I)$ for any $\sigma > 0$.
  - Which is best?
  - If σ too small, then chain won't move much.
  - If σ too large, then will usually reject proposals, so chain still won't move much.
  - Optimal: need σ "just right" to avoid both extremes. ("Goldilocks Principle")
  - Can experiment ... ("rwm.html" applet, file "Rmet") ...
  - Some theory ... limited ... active area of research ...

- What about standard error, i.e. uncertainty??
  - Estimates seem to have large fluctuations ... more than the usual (iid) standard error suggests ...

- In fact: true standard error is actually larger than in iid case, due to correlations ...
  - Estimator $= \frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)$. Variance of estimator??
  - Let $\overline{h}(x) = h(x) - \mathbf{E}_\pi(h)$, so $\mathbf{E}_\pi(\overline{h}) = 0$.
  - And, assume B large enough that $X_i \approx \pi$ for $i > B$.
  - Then, for large $M - B$, variance of estimator is:

$$\approx \mathbf{E}_\pi\Big[\Big(\Big(\frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)\Big) - \mathbf{E}_\pi(h)\Big)^2\Big] \approx \mathbf{E}_\pi\Big[\Big(\frac{1}{M-B}\sum_{i=B+1}^{M} \overline{h}(X_i)\Big)^2\Big]$$

$$= \frac{1}{M-B}\Big(\mathbf{E}_\pi(\overline{h}^2) + 2\,\mathbf{E}_\pi(\overline{h}(X_i)\overline{h}(X_{i+1})) + 2\,\mathbf{E}_\pi(\overline{h}(X_i)\overline{h}(X_{i+2})) + \dots\Big)$$

$$= \frac{1}{M-B}\,\mathbf{E}_\pi(\overline{h}^2)\Big(1 + 2\,\mathrm{Corr}(\overline{h}(X_i)\overline{h}(X_{i+1})) + 2\,\mathrm{Corr}(\overline{h}(X_i)\overline{h}(X_{i+2})) + \dots\Big)$$

$$= \frac{1}{M-B}\,\mathrm{Var}_\pi(h)\Big(1 + 2\,\mathrm{Corr}(h(X_i)h(X_{i+1})) + 2\,\mathrm{Corr}(h(X_i)h(X_{i+2})) + \dots\Big)$$

$$\equiv \frac{1}{M-B}\,\mathrm{Var}_\pi(h)\big(\text{varfact}\big) = (\text{iid variance})\,(\text{varfact})\,.$$

——————— END WEEK #10———————

[Return HW#3. (Clarify "nonparametric" density estimation.)]

Summary of Previous Class:

∗ Pseudo-random numbers

∗ Linear Congruential Generator:

—— $k_n = (ak_{n-1} + b) \bmod m$

—— $U_n = k_n/m$

∗ random? cycle length? passes statistical tests?

∗ Other distributions:

—— generated from $\{U_n\}$ by transformations (should know this!) ...

∗ Metropolis Algorithm:

—— Given X_{n-1}, propose $Y_n \sim MVN(X_{n-1}, \sigma^2 I)$

—— Then accept if $U_n < \pi(Y_n)/\pi(X_{n-1})$, else reject.

∗ Then $\mathbf{P}(X_n \in A) \to \int_A \pi(x)\,dx$.

—— How quickly? Choice of σ? Choice of M? Burn-in B? etc.

$*$ example: Java applet ("rwm.html")

$*$ example: $g(x) = |\cos(\sqrt{x_1 x_2})|$ ("Rmet")

$*$ Calculation: $\mathrm{Var}_\pi(\frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i)) \approx \frac{1}{M-B}\mathrm{Var}_\pi(h)(\mathrm{varfact})$, where

$$\mathrm{varfact} \;=\; 1 + 2\,\mathrm{Corr}(h(X_i)h(X_{i+1})) + 2\,\mathrm{Corr}(h(X_i)h(X_{i+2})) + \ldots$$

$*$ THEREFORE, true standard error $\approx$ (iid standard error) $\sqrt{\mathrm{varfact}}$. (file "Rmet")

$*$ So, to make standard error small, need $M - B >>$ varfact.

- DIAGNOSING CONVERGENCE:
  - Previous analysis ("varfact", etc.) assumes the chain has approximately converged within B steps.
  - But convergence can be very problematic!
  - e.g. multimodal distribution: $\pi = \frac{1}{2}\,N(0,1) + \frac{1}{2}\,N(10,1)$. (file "Rmet2")
  - If know, then can fix problem, e.g. set $\sigma = 4$ or $\sigma = 10$. (file "Rmet2")
  - But how to tell??
  - Some theory ... complicated ... active area of research ... instead ...
- Idea: run many different copies of the chain, started from different initial values (from "overdispersed starting distribution" ...).
  - Then, if the different chains seem "similar", then we've hopefully converged ... otherwise we haven't.
- Details:
  - Do J different runs, $\{X_i^{(j)}\}_{i=0}^{M}$, for $1 \le j \le J$, each started from the same "overdispersed starting distribution".
  - Let h be functional of interest.
  - Let $\overline{h}_j = \frac{1}{M-B}\sum_{i=B+1}^{M} h(X_i^{(j)})$ be estimate from j^{th} run, for $1 \le j \le J$.
- "INTUITIVE DIAGNOSTIC" APPROACH:
  - Let $ESTSD = \mathrm{sd}(\overline{h}_1, \ldots, \overline{h}_J)$. ("estimator standard error")

- If $ESTSD$ small, then get approximately the same answer from different starting values, so hopefully estimate is accurate.
- But if $ESTSD$ large, this suggests B or M too small. (file "Rdiag")

- "SIMPLE DIAGNOSTIC" APPROACH:
  - Let $VAR_j = \text{var}(h(X_{B+1}^{(j)}), \ldots, h(X_M^{(j)}))$. ("$j^{\text{th}}$ within-chain variance")
  - Let $WITH = \text{mean}(VAR_1, \ldots, VAR_J)$. ("mean within-chain variance")
  - Let $INTER = \text{var}(h(X_{B+1}^{(1)}), \ldots, h(X_{B+1}^{(J)}))$ ("inter-chain variance")
- Idea: <u>if</u> B, M large enough, then:
  - $\{X_{B+1}^{(j)}\}_{j=1}^{J} \approx$ iid π, so $INTER \approx v \equiv \text{Var}_\pi(h)$.
  - Also, $VAR_j \approx \text{Var}_\pi(h) = v$. [since VAR_j uses usual (iid) estimator of var]
  - So, $WITH \approx v$.
  - So, $WITH \approx INTER$.
- BY CONTRAST, if B, M <u>not</u> large enough, then usually $WITH << v$ and $INTER >> v$, so $WITH << INTER$.
- <u>Diagnostic</u>: B, M large enough if $INTER/WITH \approx 1$; not if $INTER/WITH >> 1$.
  - Sometimes shows convergence. (file "Rdiag2")
  - Sometimes shows <u>lack</u> of convergence. (file "Rdiag3")
- Of course, won't <u>notice</u> slow convergence unless start different chains in "different regions" ... "overdispersed starting distribution" ... "premature diagnosis of convergence" ...
- "GELMAN-RUBIN" APPROACH (1992):
  - Replace $INTER$ by $BET = (M-B)\,\text{var}(\overline{h}_1, \ldots, \overline{h}_J)$. ("between-chain variance")
  - Uses all of $\{X_i^{(j)}\}_{i=B+1}^{M}$, not just $X_M^{(j)}$ – good.
  - <u>if</u> $X_{B+1} \approx \pi$, then:

- Var($\overline{h}_j$) $\approx$ (iid variance) (varfact) = $\frac{v}{M-B}$ (varfact).
- So, $BET \approx (M-B)\frac{v}{M-B}$ (varfact) = v(varfact).

- iid case (varfact = 1): $BET \approx WITH$. (file "Rgel")
- stationary case ($X_0 \sim \pi$): $\frac{BET}{WITH} \approx$ varfact. (file "Rgel2")
- fast-converging case ($X_B \approx \pi$): $\frac{BET}{WITH} \approx$ varfact. (file "Rgel3")
- slow-converging case: $\frac{BET}{WITH} >>$ varfact. (file "Rgel4")
- So, diagnostic is: run good if $\frac{BET}{WITH}$ is "small enough".
- How small is small enough??
- One option: require $\frac{BET}{WITH} < 1 + 0.44(M-B)$, i.e. $M - B > 2.3(\frac{BET}{WITH} - 1)$.
  - (Comes from requiring "$\sqrt{R} < 1.2$" in their paper ...)
  - Roughly: $M - B > 2.3$(varfact $- 1$) ... similar to previous conclusion ...
- For fast-mixing chains, this will be satisfied after, say, $M = 1000$. (file "Rgel3")
- But for slow-mixing chains, it often won't. (good) (file "Rgel4")
  - Remark: for slow-mixing chains, "varfact" gets underestimated, since each chain is "stuck" in its own small region of the state space ...
- Overall, convergence diagnostics are important, but problematic ...
- OTHER VERSIONS OF MCMC (BESIDES METROPOLIS ALGORITHM):
- METROPOLIS-WITHIN-GIBBS ALGORITHM:
  - Propose to move just one coordinate at a time, leaving all the other coordinates fixed (since changing all coordinates at once may be difficult or limiting).
  - Can choose coordinates in sequence (systematic-scan, file "Rmwg"), or uniformly at random (random-scan, file "Rmwg2").
  - Can be more efficient/successful in hard problems. (HW#4)

——————— **END WEEK #11**———————

[Reminders: HW#4 due Dec 4, 6:10pm. Exam Dec 12, 7–10pm, NR25 (William Doo Auditorium, 45 Willcocks Street, basement), no aids allowed. **Exam Notes:** Be sure to explain all of your answers in detail. You do not need to simplify complicated arithmetic expressions. And, you do not need to write an R program unless the question explicitly asks you to.]

[Office hours: 2:30–3:30 this Friday (Nov 30) and Monday (Dec 3).]

Summary of Previous Class:

∗ Diagnosing convergence

—— Intuitive Diagnostic

—— Simple Diagnostic

—— Gelman-Rubin Diagnostic

∗ Metropolis-within-Gibbs algorithm ("Rmwg", "Rmwg2")

- METROPOLIS-HASTINGS ALGORITHM:
  - Previous Metropolis algorithm works provided proposal distribution is symmetric (i.e., same prob of proposing y from x, as of proposing x from y), e.g. $MVN(X_{n-1},\, \sigma\, I)$.
  - But what if it isn't?
  - Write $q(x, y)$ for the density of the proposal Y_n, given that $X_{n-1} = x$.
  - Then, in Metropolis algorithm, replace $\alpha_n = \frac{\pi(Y_n)}{\pi(X_{n-1})}$ by $\alpha_n = \frac{\pi(Y_n)\, q(Y_n, X_{n-1})}{\pi(X_{n-1})\, q(X_{n-1}, Y_n)}$.
  - Everything else remains the same.
  - Intuition: if $q(x, y) >> q(y, x)$, then Metropolis chain would spend too much time at y and not enough at x, so need to accept fewer moves $x \to y$.
- EXAMPLE: again $\pi(x_1, x_2) = C\, |\cos(\sqrt{x_1\, x_2})|\; I(0 \le x_1 \le 5, 0 \le x_2 \le 4)$, and $h(x_1, x_2) = e^{x_1} + (x_2)^2$.
  - Proposal distribution: $(Y_n \,|\, X_{n-1}) \sim MVN(X_{n-1},\; \sigma^2\, (1 + |X_{n-1}|^2)^2\, I)$.
  - (Intuition: larger proposal variance if farther from center.)

- So, $q(x, y) = C\,(1 + |x|^2)^{-2}\,\exp(-|y - x|^2 \,/\, 2\,\sigma^2(1 + |x|^2)^2)$.
- So, can run Metropolis-Hastings algorithm for this example. (file "RMH")
- Usually get between 34 and 43, with claimed standard error ≈ 2. (Recall: Mathematica gets 38.7044.)

- INDEPENDENCE SAMPLER:
  - Proposals $\{Y_n\}$ are i.i.d. from some fixed distribution (say, $Y_n \sim MVN(0, I)$).
  - Then $q(x, y) = q(y)$, depends only on y.
  - So, replace α_n by $\alpha_n = \frac{\pi(Y_n)\,q(X_{n-1})}{\pi(X_{n-1})\,q(Y_n)}$.
  - Special case: if $q(y) \equiv \pi(y)$, i.e. propose <u>exactly</u> from target density π, then $\alpha_n \equiv 1$, i.e. make great proposals, and always accept them!

- METROPOLIS-HASTINGS-WITHIN-GIBBS:
  - From $X_{n-1} = x$, propose just <u>one</u> coordinate i at a time (either systematic-scan or random-scan), from some non-symmetric density $q_i(x, y)$.
  - Then $\alpha_n = \frac{\pi(Y_n)\,q_i(Y_n, X_{n-1})}{\pi(X_{n-1})\,q_i(X_{n-1}, Y_n)}$ as usual.

- GIBBS SAMPLER:
  - Version of Metropolis-Hastings-within-Gibbs.
  - Proposal distribution for i^{th} coordinate is equal to the conditional distribution of that coordinate (according to π), conditional on the values of all the other coordinates.
  - Then $q_i(x, y) = C(x^{(-i)})\,\pi(y)$, where $x^{(-i)}$ means all coordinates <u>except</u> i^{th} one, and $C(x^{(-i)})$ is the appropriate normalising constant (which depends on $x^{(-i)}$).
  - (Then we'll have $x^{(-i)} = y^{(-i)}$, and $C(x^{(-i)}) = C(y^{(-i)})$.)
  - Then $\alpha_n = \frac{\pi(Y_n)\,q_i(Y_n, X_{n-1})}{\pi(X_{n-1})\,q_i(X_{n-1}, Y_n)} = \frac{\pi(Y_n)\,C(Y_n^{(-i)})\,\pi(X_{n-1})}{\pi(X_{n-1})\,C(X_{n-1}^{(-i)})\,\pi(Y_n)} = 1$.
  - So, always accept.

- EXAMPLE: Variance Components Model
  - Update of μ (say) should be from conditional density of μ, conditional on current values of all the other coordinates: $\mathcal{L}(\mu \,|\, V, W, \theta_1, \ldots, \theta_K, Y_{11}, \ldots, Y_{JK})$.
  - This conditional density is proportional to the full joint density, but with everything except μ treated as constant.
  - Full joint density is:

$$= \; Ce^{-b_1/V} V^{-a_1-1} e^{-b_2/W} W^{-a_2-1} e^{-(\mu-a_3)^2/2b_3} V^{-K/2} W^{-JK/2} \; \times$$

$$\times \; \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V - \sum_{i=1}^{K}\sum_{j=1}^{J} (Y_{ij} - \theta_i)^2/2W \right] .$$

  - So, conditional density of μ is

$$C_2 \, e^{-(\mu-a_3)^2/2b_3} \; \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V \right] .$$

  - This equals

$$C_3 \, \exp\Big(-\mu^2 (\frac{1}{2b_3} + \frac{K}{2V}) + \mu (\frac{a_3}{b_3} + \frac{1}{V} \sum_{i=1}^{K} \theta_i) \Big) .$$

  - This is $N(m, v)$, where $1/2v = \frac{1}{2b_3} + \frac{K}{2V}$ and $m/v = \frac{a_3}{b_3} + \frac{1}{V} \sum_{i=1}^{K} \theta_i$.
  - Solve: $v = b_3 V/(V + Kb_3)$, and $m = (a_3 V + b_3 \sum_{i=1}^{K} \theta_i) \,/\, (V + Kb_3)$.
  - So, in Gibbs Sampler, each time μ is updated, we sample it from $N(m, v)$ for this m and v (and always accept).
- Similarly, conditional distribution for V is:

$$C_4 e^{-b_1/V} V^{-a_1-1} V^{-K/2} \exp\left[-\sum_{i=1}^{K} (\theta_i - \mu)^2/2V \right] , \qquad V > 0 .$$

  - Recall that "$IG(r, s)$" has density $\frac{s^r}{\Gamma(r)} \, e^{-s/x} \, x^{-r-1}$ for $x > 0$.
  - So, conditional distribution for V equals $IG(a_1 + K/2, \; b_1 + \frac{1}{2} \sum_{i=1}^{K} (\theta_i - \mu)^2)$.

- And, conditional dist. for W equals $IG(a_2 + KJ/2,\ b_2 + \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{J}(Y_{ij} - \theta_i)^2)$.
- And, for θ_i $(1 \le i \le K)$, conditional distribution (exercise: verify this!) is:

$$N\Big(\frac{V\sum_{j=1}^{J} Y_{ij} + W\,\mu}{J\,V + W},\ \frac{V\,W}{J\,V + W}\Big)\,.$$

- So, in this case, the systematic-scan Gibbs sampler proceeds by:
  - Update V from its conditional distribution $IG(\ldots, \ldots)$.
  - Update W from its conditional distribution $IG(\ldots, \ldots)$.
  - Update μ from its conditional distribution $N(\ldots, \ldots)$.
  - Update θ_i from its conditional distribution $N(\ldots, \ldots)$, for $i = 1, 2, \ldots, K$.
  - Repeat all of the above M times.
  - Not on homework ... but do try it!
- What OTHER tricks are there? We have already seen one ...
- IMPORTANCE SAMPLING:
- Suppose want to evaluate $I = \int h(x)\,\pi(x)\,dx$.
  - Can't sample iid $\sim \pi$, so can't do classical Monte Carlo.
  - Instead, re-write this as $I = \int h(x)\,\frac{\pi(x)}{f(x)}\,f(x)\,dx$, where f is easily sampled from.
  - Then $I = \mathbf{E}\left(h(X)\,\frac{\pi(X)}{f(X)}\right)$, where $X \sim f$.
  - Can then do classical (iid) Monte Carlo integration, get standard errors etc.
  - (Like what we did before with e.g. $X \sim$ Exponential(1), i.e. $f(x) = e^{-x}$.)
  - However, the function $h(x)\,\frac{\pi(x)}{f(x)}$ may be highly variable, leading to large standard error and poor estimates.
- REJECTION SAMPLER (another way to iid sample from π):
  - Suppose there is some other easily-sampled density f, and known $K > 0$, such that $K\,f(x) \ge \pi(x)$ for all x.

- (So, need to know normalising constant for π.)
- Sample $X \sim f$, and $U \sim \text{Uniform}[0,1]$.
- If $U \le \pi(X)/Kf(X)$, then accept X (as a draw from π).
- Otherwise, reject X and start over again!

- Conditional on accepting, we have

$$\mathbf{P}\Big(X \le y \,|\, U \le \pi(X)/Kf(X)\Big) \;=\; \frac{\mathbf{P}\big(X \le y,\;\; U \le \pi(X)/Kf(X)\big)}{\mathbf{P}\big(U \le \pi(X)/Kf(X)\big)}$$

$$= \frac{\int_{-\infty}^{y} f(x)\,(\pi(x)/Kf(x))\,dx}{\int_{-\infty}^{\infty} f(x)\,(\pi(x)/Kf(x))\,dx} \;=\; \frac{\int_{-\infty}^{y} \pi(x)\,dx}{\int_{-\infty}^{\infty} \pi(x)\,dx} \;=\; \int_{-\infty}^{y} \pi(x)\,dx\,.$$

  - So, conditional on accepting, $X \sim \pi$. Good! iid!
  - However, probability of accepting may be very small, in which case get very few samples. (Plus need normalising constant.)

- Example: $\pi = N(0,1)$, i.e. $\pi(x) = (2\pi)^{-1/2}\exp(-x^2/2)$.
  - Want: $\mathbf{E}_\pi(X^4)$, i.e. $h(x) = x^4$.
  - Let f be double-exponential distribution, i.e. $f(x) = \frac{1}{2}\exp(-|x|)$.

- If $K = 8$, then:
  - For $|x| \le 2$, $Kf(x) \ge 8\,\frac{1}{2}\,\exp(-2) \ge (2\pi)^{-1/2} \ge \pi(x)$.
  - For $|x| \ge 2$, $Kf(x) = 8\,\frac{1}{2}\,\exp(-|x|) \ge 8\,\frac{1}{2}\,\exp(-x^2/2) \ge (2\pi)^{-1/2}\exp(-x^2/2) = \pi(x)$.

- So, can apply rejection sampler with this f and K. (file "Rrej")

END WEEK #12

[Collect HW#4 ... discuss hw4clar.pdf etc. ...]

[Final exam Dec 12, 7–10pm, NR25 ... office hours afternoon of Dec 11?]

[Usefulness of the reference books?]

[Last class ... 47 posted R programs ... 703 e-mails ...]

Summary of Previous Class:

∗ Metropolis-Hastings algorithm ("RMH")

∗ Independence Sampler

∗ Metropolis-Hastings-within-Gibbs

∗ Gibbs sampler

—— Example: variance components model

—— How to compute conditional distributions

∗ Importance Sampling

∗ Rejection Sampler ("Rrej")

- EM ALGORITHM:
- Suppose want to maximise a likelihood $\mathcal{L}(\theta \,|\, y)$, for an unknown parameter θ, given observations y.
- Suppose it is <u>easy</u> to maximise $\mathcal{L}(\theta \,|\, y, z)$.
- However, we do <u>not</u> observe z. ("missing data") ("data augmentation")
- Iterative algorithm:
  - Initial guess $\hat{\theta}$ for θ.
  - Then, compute $\hat{z} = \mathbf{E}[z \,|\, y,\ \theta = \hat{\theta}]$. ("E step")
  - Then, consider $Q_{\hat{z}}(\theta) \equiv \mathcal{L}(\theta \,|\, y,\ z = \hat{z})$.
  - Find $\hat{\theta}$ to maximise $Q_{\hat{z}}$. ("M step")
  - Repeat!
- EXAMPLE #1: $y, z \sim$ Exponential(λ), i.i.d.

- Suppose observe y, but not z.
- What is MLE for λ?
- If knew y and z, then MLE would be $\hat{\lambda} = \frac{1}{\text{sample mean}} = \frac{1}{\frac{y+z}{2}} = \frac{2}{y+z}$.
- But we don't know z!

- EM algorithm proceeds by:
  - Start with initial guess $\hat{\lambda}$ for λ.
  - E-step: Let $\hat{z} = \mathbf{E}[z \,|\, y, \hat{\lambda}] = 1/\hat{\lambda}$.
  - M-step: Let $\hat{\lambda}$ be MLE for λ given y and $z = \hat{z}$, i.e. $\hat{\lambda} = \frac{1}{\frac{y+\hat{z}}{2}} = \frac{2}{y+\hat{z}}$.
  - Repeat!
- Numerical example: $y = 10$, start with $\hat{\lambda} = 2$. (file "REM")
  - $\hat{\lambda}$ converges to $0.1 = 1/y$ (usual MLE for Exponential(λ)).
  - $\hat{z}$ converges to $10 = y = 1/\hat{\lambda}$.
  - Not too surprising ...
- EXAMPLE #2: N individuals. $R_i \sim$ Poisson(λ) (iid).
  - Let $n_j = \#\{i : R_i = j\}$.
  - Observe n_j for $j \geq 2$, plus observe $n_* \equiv n_0 + n_1$, but do not observe n_0 or n_1.
  - (e.g. only get informed when $R_i \geq 2$)
- What is MLE for λ??
  - Well, if knew all the n_i, then MLE would be $\hat{\lambda} = \frac{1}{N} \sum_{j=1}^{\infty} j\, n_j$. (True since full MLE equals sample mean.)
  - But we don't know n_0 or n_1, just their sum $n_* = n_0 + n_1$. What to do?
- Use EM algorithm, with missing data $z = n_1$.
  - Initial guess $\hat{\lambda}$.

- E-step: set $\hat{z} = \mathbf{E}[z \,|\, \hat{\lambda}, n_*, n_2, n_3, \ldots] = \frac{\hat{\lambda}}{1+\hat{\lambda}} n_*$. (True since $\mathbf{P}(R_i = 1 \,|\, R_i = 0 \text{ or } 1) = \frac{\mathbf{P}(R_i=1)}{\mathbf{P}(R_i=0)+\mathbf{P}(R_i=1)} = \frac{e^{-\lambda}\lambda^1/1!}{e^{-\lambda}\lambda^0/0!+e^{-\lambda}\lambda^1/1!} = \frac{\lambda}{1+\lambda}$.)
- M-step: set $\hat{\lambda} = \frac{1}{N}\big(z + \sum_{j=2}^{\infty} j\, n_j\big)$. (True since full MLE equals sample mean.)
- Repeat!

- Run on data simulated from Poisson(2.7). (file "REM2")
  - $\hat{\lambda}$ converges to about 2.786; close to "true" value of 2.7.
  - $\hat{z}$ converges to about 179.5, out of $n_* = 244$.
  - Makes sense since $(\lambda/(1+\lambda))n_* = 2.7/3.7 * 244 \doteq 178.05$.

- WHAT ABOUT CONVERGENCE?
  - Not hard to see that each iteration can only <u>increase</u> $\mathcal{L}(y \,|\, \theta = \hat{\theta})$.
  - So, under differentiability conditions, at least should converge to a <u>critical</u> point of the likelihood (e.g. Jones, Thm 3.4.5.)
  - However, <u>not</u> guaranteed to converge to MLE in general.
  - Even if it converges, the convergence is <u>slow</u> at the end (usually just "geometric" [a.k.a. "linear"], i.e. $e_{n+1} = re_n$ for some $r < 1$).

- EXAMPLE #3: N individuals. For each individual i:
  - $C_i \sim$ Bernoulli(ξ). ("censoring indicator")
  - $R_i \sim$ Poisson(λ). (all independent)
  - Observe $Y_i = R_i\, I(C_i = 0)$, i.e. observe count R_i, or <u>zero</u> if it was censored.
  - Let $n_j = \#\{i : Y_i = j\}$. (observed counts)
  - So, don't know how many of n_0 were from $C_i = 1$ [censored], and how many from $C_i = R_i = 0$ [genuine zeroes].

- What is MLE of (ξ, λ)??

- Well, if knew $z \equiv \#\{i : C_i = 1\}$, then would have

$$\mathcal{L}(\xi, \lambda \,|\, \{n_j\}, z) \;=\; C\, \xi^z (1-\xi)^{N-z}\, (e^{-\lambda})^{n_0 - z} \prod_{j=1}^{\infty} \Big(\frac{\lambda^j}{j!} e^{-\lambda} \Big)^{n_j},$$

  and MLE would be empirical means, i.e. $\hat{\xi} = z/N$, and $\hat{\lambda} = \frac{1}{N-z} \sum_{j=1}^{\infty} j\, n_j$.

- So, use EM algorithm! (Here $y = \{n_j\}$.)

- Start with initial guesses $\hat{\xi}$ and $\hat{\lambda}$.

- E-step: let $\hat{z} = \mathbf{E}[z \,|\, \{n_j\}, \hat{\xi}, \hat{\lambda}] = \frac{\hat{\xi}}{\hat{\xi} + (1-\hat{\xi}) e^{-\hat{\lambda}}} n_0$. (True since $\mathbf{P}(C_i = 1 \,|\, Y_i = 0) = \frac{\hat{\xi}}{\hat{\xi} + (1-\hat{\xi}) e^{-\hat{\lambda}}}$.)

- M-step: let $\hat{\xi}$ and $\hat{\lambda}$ be the MLE if this were the true z: $\hat{\xi} = \hat{z}/N$, and $\hat{\lambda} = \frac{1}{N-\hat{z}} \sum_{j=1}^{\infty} j\, n_j$.

- Numerical example (from Thisted p. 188): $n_0 = 3062$, $n_1 = 587$, $n_2 = 284$, $n_3 = 103$, $n_4 = 33$, $n_5 = 4$, $n_6 = 2$. ($N = 4075$) (file "REM3")

  - $\hat{\xi}$ converges to about 0.615; $\hat{\lambda}$ converges to about 1.038.

  - (But convergence slow at end ...)

- THE END!

  - Good luck on exam, second semester, etc.
