Review of the Final

BTRY/STSCI 4520

Reminders/Comments

This is a final exam that means

- You should work on your own

- We will answer clarification questions but not de-bug your code
- No extensions

but it is open book

- You may use any written resource, but only packages that we specifically refer to.
- We are happy to explain what error messages mean, but you should isolate the line of code that generates them.

On the final

- Structured like a homework; 3 questions, 6 parts each
- Sub-parts are structured as (i): do the coding, (ii) and (iii): plot or comment on results.
- Happy to tell you what code should do, written responses are up to you.

Question 1: Control Functionals

Reminder: control variates

- Monte Carlo estimate if $X_i \sim f(\cdot)$

$$Eg(X) = \int g(x)f(x)dx \approx \frac{1}{N}\sum g(X_i)$$

- If we know $Eh(X) = 0$ and $\mathrm{cor}(g(X), h(X)) \neq 0$ then

$$\frac{1}{N}\sum g(X_i) - \alpha\frac{1}{N}\sum h(X_i)$$

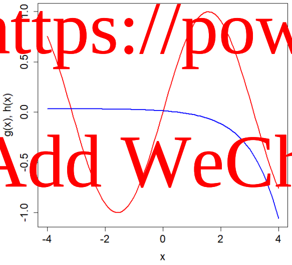
  has smaller variance if $\alpha = \mathrm{cov}(g(X), h(X))/\mathrm{var}(h(X))$.

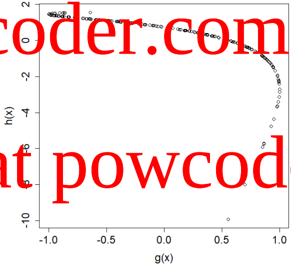
- α? Estimate from values of $g(X_i)$ and $h(X_i)$.

Example on Final

Part a: $X \sim N(0,1)$, $g(x) = \sin(x + 0.1)$

Use $h(x) = -e^x + e^{1/2}$ because $E\left[-e^X\right] = -e^{1/2}$.





Comparisons

Parts b/c with antithetic sampling

Note: for $X \sim N(0,1)$ $(X, -X)$ are antithetic.

We want to look at

1. Vanilla: $\frac{1}{N}\sum g(X_i)$
2. Control Variates: $\frac{1}{N}\sum g(X_i) - \alpha\frac{1}{N}\sum h(X_i)$
3. Antithetic $\frac{1}{N}\sum(g(X_i) + g(X_i))/2$
4. Both: $\frac{1}{N}\sum(g(X_i) + g(-X_i))/2 - \alpha\frac{1}{N}\sum(h(X_i) + h(-X_i))/2$

For both, calculate alpha from values of $(g(X_i) + g(-X_i))/2$ and $(h(X_i) + h(-X_i))/2$

Question about relative improvement are for you to think about. Also bonus on using $g(x) = \sin(x)$ or $h(x) = x$.

Control Functionals

Part d taken from Oates, Girolami, Chopin, 2016

- Idea, if we can make $h_N(x)$ closer to $g(x)$ as $N \to \infty$ we will do better faster.
- Use $X_1, \ldots, X_{N/2}$ to get a better $h_N(x)$, $X_{N/2+1}, \ldots, X_N$ for control variate integral.
- *How?* Nonparametric smoothing!

$$h_N(x) = \sum_{j=1}^{N/2} d_j k_j(x)$$

d_j make $h_N(x)$ approximate $g(x)$ but we know

$$\int k_j(x) f(x) dx = 0$$

Control Functional Details

Need $Ek_j(X) = 0$: modify a kernel function

$$k_j(x) = \phi'(x - X_j; s) + \phi(x - X_j; s) f'(x)/f(x)$$

where $\phi(z; s)$ is normal density with standard deviation s (`dnorm(z,sd=s)`)

$$\phi(z; s) = \frac{1}{\sqrt{2\pi} s} e^{-z^2/2s^2}.$$

Why?

$$\begin{aligned}
\int k_j(x) f(x) dx &= \int (\phi'(x - X_j; s) f(x) dx + \phi(x - X_j; s) f'(x)) dx \\
&= \int \frac{d}{dx} (\phi(x - X_j; s) f(x)) dx \\
&= \phi(\infty - X_j; s) f(\infty) - \phi(-\infty - X_j; s) f(-\infty) = 0
\end{aligned}$$

Control Functional Implementation

1. Calculate matrix to store for $1, \ldots, N/2$

$$K_{ij} = k_j(X_i) = -\phi'(X_i - X_j; s) + \phi(X_i - X_j; s) f'(X_i)/f(X_i)$$

   remember $\phi(z; s)$ is $N(0, s)$, $f(x)$ is $N(0, 1)$.

2. Obtain d_j by regressing $g(X_i)$ on the matrix K_{ij}

   - You may use the function `lm`
   - If d_j is `NA`, set it to zero.

3. Plug in $X_{N/2+1}, \ldots, X_N$ to

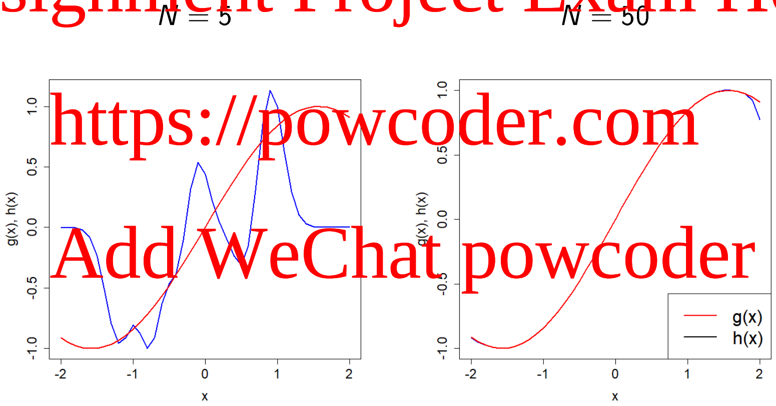
$$h_N(x) = \sum_{l=N/2+1}^{N} d_j k_j(X_l)$$

   (note that the X_j used to define k_j stay the same.

k_j: sort of kernel/sort of basis. No smoothing penalty because $g(X_i)$ has no error.

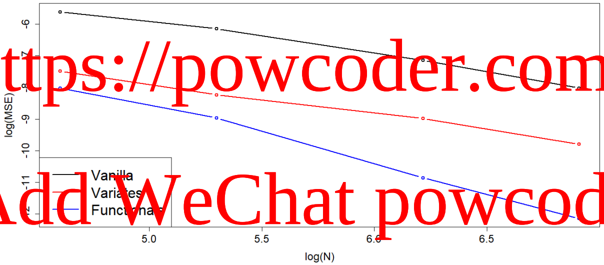
Example

Using $g(x) = \sin(x)$



Convergence Rates

Parts e/f: simulation based on control functionals Example is based on $g(x) = \sin(x)$, control variate $h(x) = x$ (more effective than parts a-c). Note [illegible] changes with N.



End of Part f:

$$\sqrt{MSE} \approx CN^{\alpha} \rightarrow \log(\sqrt{MSE}) \approx \alpha \log(N) + log(C)$$

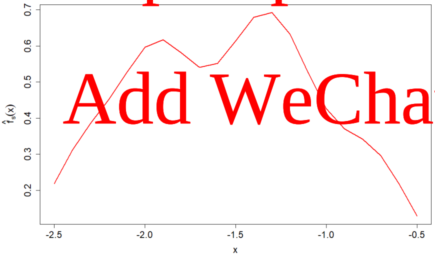
looks like a linear model.

Question 2: Kernel Density Estimates

Kernel Density Estimation, given $X_1, \ldots, X_n$:

$$\hat{f}_s(x) = \frac{1}{n}\sum_{i=1}^{n}\phi(x - X_i; s)$$

($\phi(z; s)$ is still $N(0,1)$ density).



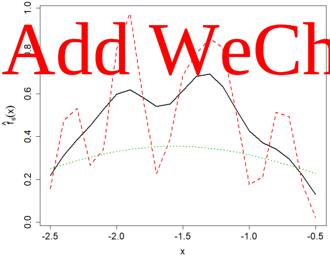
- Efficient calculation
- Choice of s
- Simulation

Cross Validation

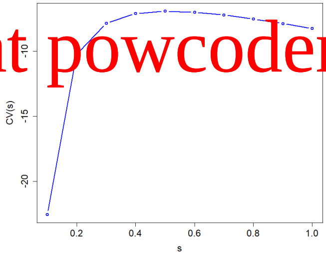
Part b to choose s, we want to make the density on a *new* point as high as possible.

- $\hat{f}_s^{-i}(X_i)$: density with bandwidth s evaluated at X_i estimated *without* using X_i.
- s small $\Rightarrow$ peaks will miss new points, s large $\rightarrow$ low density everywhere.
- Use $\sum \log \hat{f}_s^{-i}(X_i)$ as in log likelihood.

Possible Densities



CV Scores



CV can be calculated efficiently

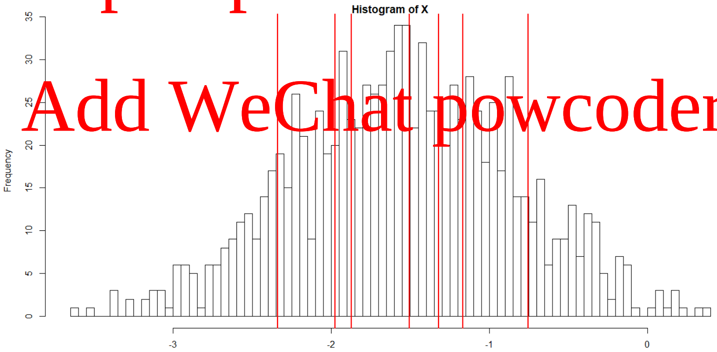
Simulation

Part c Form of density is like

$$Z|k \sim N(X_k, s), k \sim \text{Unif}(1, \ldots, n)$$

1. Choose an observation at random
2. Simulate from $N(X_k, s)$

Can also be vectorized effectively



Hellinger Distance

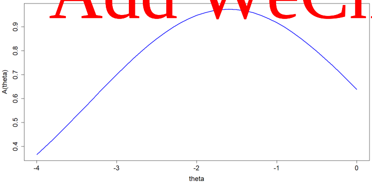
Part d Measure of distance between densities

$$\int (\sqrt{g(x)} - \sqrt{f(x)})^2 dx = 2 - 2\int \sqrt{g(x)}\sqrt{f(x)} dx = 2 - 2A(g, f)$$

or use *affinity*

$$A(g, f) = \int \sqrt{g(x)}\sqrt{f(x)} dx = \int \sqrt{\frac{g(x)}{f(x)}} f(x) dx \approx \frac{1}{N}\sum \sqrt{\frac{g(Z_j)}{f(Z_j)}}$$

Statistical use: find θ to maximize affinity between $\hat{f}(x)$ and $f(x; \theta)$



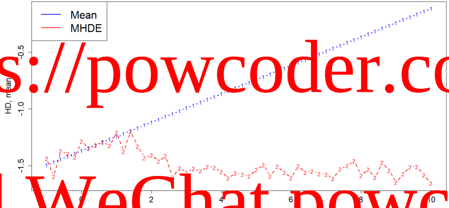
Here we have $\hat{f}(x)$ from KDE and $f(x; \theta)$ is $N(\theta, 1)$:

$$A(\theta) = \frac{1}{N}\sum \sqrt{\frac{\phi(Z_j - \theta, 1)}{\hat{f}(Z_j)}}$$

Optimization and Robustness

To find $\max_\theta A(\theta)$ use `optimize` function between -10 and 10.

Part e Reason for Hellinger is *robustness* – adding a large value to data doesn't change estimate much.



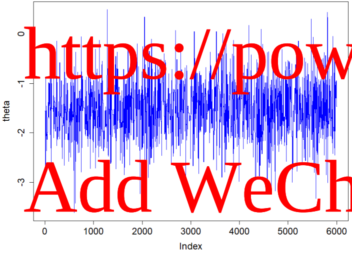
- Add O to data (8 points instead of 7)
- Don't change s with value of O (yes this is cheating).
- Do simulate new Monte Carlo points with each value.

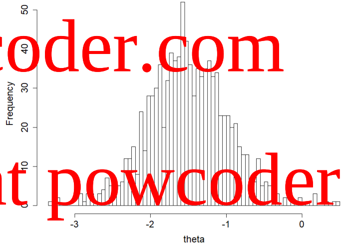
Hellinger's Posterior

Hooker and Vidyashankar, 2014 suggested finding a posterior based on

$$P(\theta) = e^{2nA(\theta)}\pi(\theta)$$

Here we let $\pi(\theta)$ be $N(0, 10)$ and sample from $P(\theta)$ by MCMC





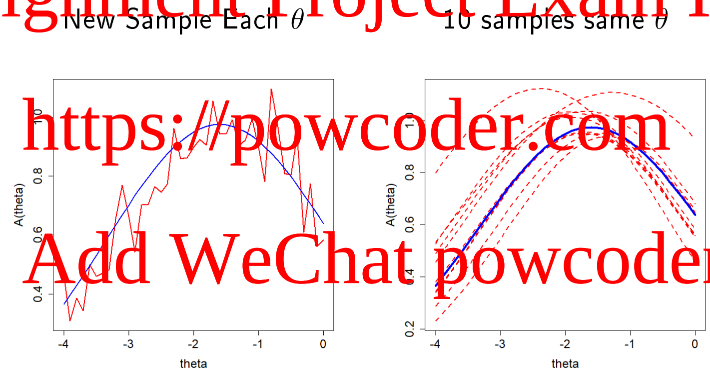
For this question:

- Keep same MC samples with $A(\theta)$ for each θ.
- Experiment to get acceptance to around 30%; decide on thinning from visual inspection (don't work too hard).

Stochastic Objective Functions

Monte Carlo integration $\Rightarrow$ evaluation of $A(\theta)$ random.

Using same samples makes problem smooth, still random.

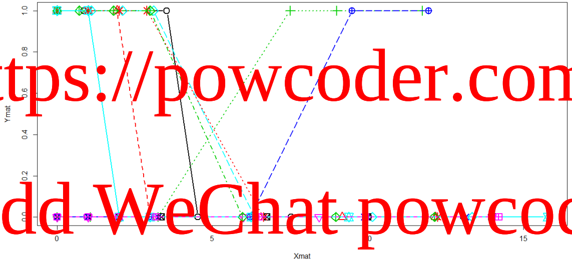


In MCMC, stochastic posteriors are ok, but decrease acceptance rate.

Mixed Effects Logistic Models

Example data

- 12 Subjects, each measured 7 times
- Record toenail health – time from treatment



Logistic model (can use `plogis`)

$$P(Y_{ij} = 1|Z_j) = \frac{e^{\beta 0+\beta_1 X_{ij}+Z_j}}{1+e^{\beta 0+\beta_1 X_{ij}+Z_j}}$$

X_{ij} = time of visit, Z_j = effect of subject j.

Generative Model

Don't get to see subject effects Z_j: model for a new data set is

1. Generate new $Z_1, \ldots, Z_{12}$ from $N(0, \sigma^2)$
2. Then generate Y_{ij} from $P(Y_{ij} = 1|Z_j)$

Since we don't see Z_j probability of observed data is

$$P(Y_{i1}, \ldots, Y_{ij}) = \int P(Y_{i1}, \ldots, Y_{ij}|z)\phi(z;\sigma)dz$$

Where we can write

$$\begin{aligned}
P(Y_1, \ldots, Y_{7j}|z) &= \prod_{i=1}^{7} P(Y_{ij}|z) \\
&= \prod_{i=1}^{7} P(Y_{ij} = 1|z)^{Y_{ij}}(1 - P(Y_{ij} = 1|z))^{1-Y_{ij}} \\
&= \prod_{i=1}^{7} [Y_{ij}P(Y_{ij} = 1|z) + (1 - Y_{ij})(1 - P(Y_{ij} = 1|z))]
\end{aligned}$$

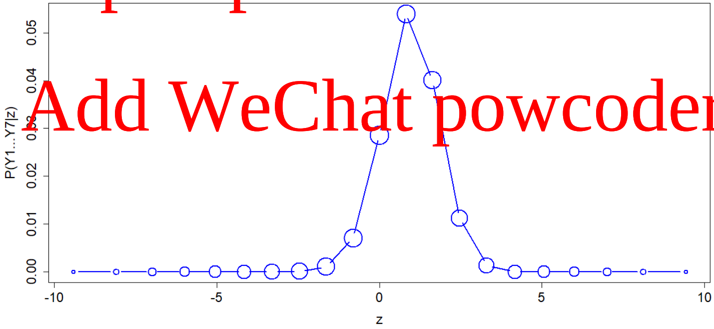
Use most convenient form, or `dbinom`.

Gauss Hermite Approximation

Part a Package ecoreg function `GaussHermite(21)` produces z_q, w_q in columns `Points` and `Weights` so that

$$\int P(Y_{i1},\ldots,Y_{ij}|z)\phi(z;\sigma)dz \approx \sum w_q P(Y_{i1},\ldots,Y_{ij}|\sigma z_q)$$

One subject at z_q; circle size $= w_q^{0.05}$.



log Likelihood

Parameter vector $\theta = (\beta_0, \beta_1, \sigma)$ has *negative* log likelihood

$$\ell(\theta) = -\sum_j \log P(Y_{1j}, \ldots, Y_{7j}|\theta)$$

(Note that θ changes $P(Y_{1j}, \ldots, Y_{7j})$ above – dropped to make notation cleaner).

Write function `logistic.nll` to evaluate:

- Obtain $P(Y_{ij}|z_q, X_{ij}, \theta)$ from `plogis` (incl. σ change to z_q).
- Evaluate $P(Y_{1j}, \ldots, Y_{7j}|z_q, X_{\cdot j}, \theta)$ from formulas or `dbinom`.
- Obtain

$$P(Y_{1j}, \ldots, Y_{7j}|X_{1j}, \ldots, X_{7j}, \theta) \approx \sum w_q P(Y_{1j}, \ldots, Y_{7j}|z_q, X_{\cdot j}, \theta)$$

- negative log likelihood is minus sum of logs.

Maximizing and Alternatives

Maximum likelihood estimator minimizes negative log likelihood.

Use

```
optim(par=theta,fn=logistic.nll,data=toenail)
```


to get minimum.

Part b Monte Carlo alternative

Replace Gauss-Hermite approximation with

$$\int P(Y_{i1}, \ldots, Y_{ij}|z)\phi(z, \sigma)dz \approx \frac{1}{N}\sum_k P(Y_{i1}, \ldots, Y_{ij}|Z_k)$$

with $Z_1, \ldots, Z_N \sim N(0, \sigma^2)$.

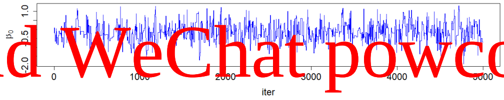
Possible to code so minimal changes from Part a.

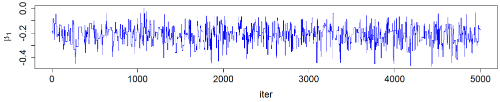
Part c: MCMC

- Fix $\sigma = 1.2$ (MCMC techniques for variances are more fiddly).
- Set a prior $\beta_0 \sim N(0, 10)$, $\beta_1 \sim N(0, 10)$.
- Easiest to work with log posterior = log likelihood + log prior (remember `log=TRUE` in density functions)

Run MCMC

- Start $\theta = (-0.2, -0.2)$
- Use random walk proposal $N(\beta_0, 1)$ and $N(\beta_1, 0.2)$





Obtain mean values and quantiles from chain.

Part d: Extended MCMC

What if we also include $Z_1, \ldots, Z_{12}$ as values to be sampled?

Whole likelihood is

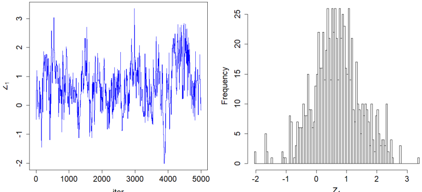
$$P(Y, Z, \theta) = \left(\prod_j \left[\prod_i P(Y_{ij} | X_{ij}, Z_j, \theta) \right] P(Z_j | \sigma) \right) \pi(\theta)$$

easier to turn into logs; keep $\sigma = 1.2$.

For MCMC proposal is

1. $\beta_0 \sim N(\beta_0, 0.25)$, $\beta_1 \sim N(\beta_1, 0.05)$
2. $Z_j \sim N(0, 0.5)$

Z_1 chain gives $P(Z_1 | Y_1, \ldots, Y_{71})$ (remember to thin)



Part e: SMC

Alternative random number generation from $f(z)$:

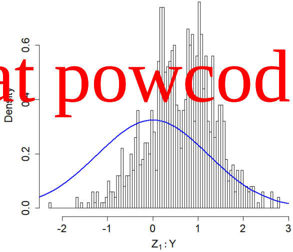
1. Generate $Z_1, \ldots, Z_N \sim g(z)$
2. Resample (with replacement) using probabilities $W_i = f(Z_i)/g(Z_i)$ to get Z_i^*

`sample` has input `prob=W` to specify sampling probability.

Our example

$$Z_i \sim N(0, \sigma^2)$$

$$W_i = P(Y_{\cdot 1} | Z_i, X_{\cdot 1}, \theta)$$

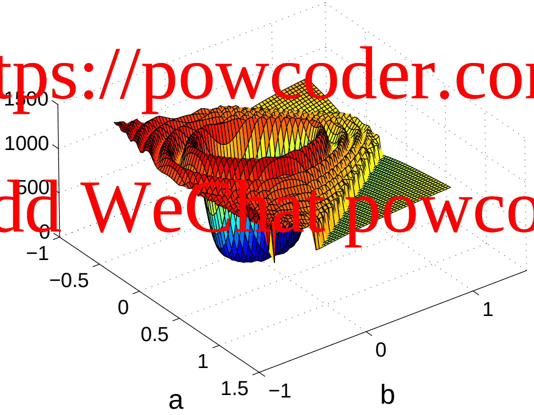


$Y_{\cdot 1}, X_{\cdot 1}$ data from Subject 1, θ from Part a estimate.

Bonus

For a hard optimization problem:

1. Write squared error function
2. try function to avoid errors



Wrap Up

BTRY/STSCI 4520: Many Topics Covered Briefly

- Computational considerations
  - Programming strategies + computational complexity
  - Numerical stability
- Numerics
  - Optimization
  - Random number generation
  - Integration (numerical/Monte Carlo)
- Statistics
  - Simulation
  - Boostraps and permutation
  - Nonparametric smoothing
  - Maximum likelihood and LASSO penalties
  - MCMC

Many tools enable practical, modern statistics.