Principles of experimental design for big data analysis

James McGree
Associate Professor of Statistics
School of Mathematical Sciences
Queensland University of Technology

james.mcgree@qut.edu.au | www.jamesmcgree.com
@j_mcgree

Joint work with Drovandi, Holmes, Mengersen, Richardson and Ryan

Outline of talk

- Introduction/motivation
- Overview of Bayesian experimental design
- Experimental design in the context of big data
- Our proposed approach/algorithm
- Motivating examples:
  1. Mortgage default
  2. Accelerometer data
- Conclusions and future work

Modern era of big data

- Massive volumes of data being collected at an accelerating pace
- Traditional measurements are now complemented with digital data obtain from, e.g., images, text, audio, sensors, etc
- Such data have the potential to inform important problems in health, science, business, engineering, however....
- Size, complexity and quality makes these data sets difficult to process and analyse

Modern era of big data

- Often computationally prohibitive to store and manage such data sets on a single computer
- Generally require high performance computing
- Similarly, the analyses of these data requires computational and statistical techniques that exceeds typical capacity
- New technological and/or methodological methods are needed
- This motivates the development of new statistical methods for inference that accounts for the characteristics of the data, adjusts for potential bias/data gaps, and appropriately handles storage and analysis needs.

Modern era of big data

Some methods to address challenges of managing, modelling and analysing big data

- Divide-and-conquer or divide-and-recombine methods (Xi et al., 2010 and Guhaa et al., 2012)
- Similar methods have proposed such as consensus Monte Carlo (Scott, Blocker and Bonassi, 2013) and bag of little bootstraps (Kleiner et al., 2014)
- Others have studied the properties of MCMC subsampling algorithms (Bardenet, Doucet and Holmes, 2014, 2015)
- Approaches for optimal subsampling (Wang et al., 2017/18)
- R packages for big data analyses (Wang et al., 2015)
- And many other developments....

Modern era of big data

- Recent(-ish) reviews are given by Fan, Han and Lui (2014) and Wang et al. (2015)
- Despite many advantages of the above (and other approaches), there is agreement that challenges remain
- For example
  - Reduce spurious correlations/patterns
  - Continuing to improve computational and algorithmic efficiency and stability
  - Accommodating heterogeneity and statistical biases associated with combining data from different sources using different technologies
- Given the acceleration of size and diversity of big data, potentially these will remain as stumbling blocks for the foreseeable future.

Proposed approach

- Propose an approach that may overcome some of these issues
- Targeted towards regression models with large N observations and small to moderate sized p predictors
- Depending on the analysis aim, use methods from optimal experimental design to make the inference problem more computationally tractable
- Instead of analysing the whole data set, a retrospective sample may provide enough information to address the analysis aim
- The specific sample is drawn in accordance with an experimental design based on a specific analysis question
- The analysis is then based on this subsample

Proposed approach

- Thus, the big data challenge is being able to extract the design from the data set
- This is much more tractable
- Modelling problem reduces to a (near) designed analysis
- Should result in less correlation between predictors, potentially less spurious correlations and patterns, etc

Proposed approach

- Many big data inferential goals for which our approach is applicable
- Goals for which design principles and corresponding utility functions are well established
- Examples include estimation, testing significance of parameter values, prediction, identification of relationships, variable selection, etc
- Potential to consider such an approach for use within divide-and-conquer type algorithms and/or in sequential learning
- Potential to use this approach to evaluate the quality of the data including potential biases and data gaps

Background - Bayesian inference

- When interested in estimating θ

$$p(\theta|y,d) \propto p(\theta)p(y|\theta,d),$$

where $p(\theta)$ is prior and $p(y|\theta,d)$ is the likelihood.

- When interested in model choice, suppose K models are being considered, with $m = 1, \ldots, K$

$$p(\theta_m|y,m,d) = \frac{p(\theta_m|m)p(y|\theta_m,m,d)}{Z_m},$$

where $Z_m = \int_{\theta_m} p(\theta_m|m)p(y|\theta_m,m,d)\mathrm{d}\theta_m$.

- $Z_m \propto p(m|y,d)$, so pick the model with the largest Z_m.

Background

- Two main challenges:
  1. Approximating the expected utility;
  2. Maximising the utility.
- Maximise expected utility $d^* = \arg\max_d u(d)$, where

$$u(d) = \sum_{m=1}^{K} p(m) \int_y u(d, y, m) p(y|d, m) \mathrm{d}y.$$

- $u(d, y, m)$ is some measure of information gained from d given model m and observed data y.
- Importantly, $u(d, y, m)$ is typically a function of $p(\theta_m|y, m, d)$.

Background

- $u(d)$ typically cannot be solved analytically
- Can be approximated using Monte Carlo integration

$$u(d) \approx \sum_{m=1}^{K} p(m) \frac{1}{B} \sum_{b=1}^{B} u(d, y_{mb}, m),$$

where $y_{mb} \sim p(y|\theta_{mb}, m, d)$ and $\theta_{mb} \sim p(\theta|m)$.

- Typical Bayesian utility is the KLD between the prior and the posterior.
- Hence, B posterior distributions need to be approximated or sampled from to approximated $u(d)$.
- Computationally challenging task.

Entropy

- Suppose discrete random variables X and Y
- Define entropy:

$$H(X) = -\sum_{x \in \mathcal{X}} f(x) \log f(x)$$

- Define conditional entropy:

$$H(X|Y) = -\sum_{y \in \mathcal{Y}} f(y) \sum_{x \in \mathcal{X}} f(x|y) \log f(x|y)$$

- Mutual information between X and Y is defined as:

$$I(X;Y) = H(X) - H(X|Y)$$

- Design: Focus on expected change in entropy of X upon observing data Y.

Utility functions derived from mutual information

- Estimation (KLD, Shannon, 1948)

$$u_P(d, y, m) = \int_\theta p(\theta|m, y, d) \log p(y|\theta, m, d) d\theta - \log p(y|m, d)$$

- Model discrimination (Box and Hill, 1967; Drovandi, McGree, Pettitt, 2014)

$$u_M(d, y, m) = \log p(m|y, d)$$

- Difficult to approximate $\log p(y|m, d)$ (and $\log p(m|y, d)$)
- Computationally difficult to efficiently approximate $p(\theta|m, y, d)$
- Need computationally efficient methods to approximate such probabilities/densities
- Other utilities available e.g: Maximise inverse posterior variance

Sequential designs - SMC single static Model m

- Sample from sequence of targets
- Data annealing here

$$\pi_t(\theta_m|y_{1:t}, d_{1:t}) = f(y_{1:t}|\theta_m, d_{1:t})\pi(\theta_m)/Z_{m,t}, \text{ for } t = 1, \ldots, T. \quad (1)$$

$y_{1:t}$ (independent) data up to t, $d_{1:t}$ design points up to t, θ_m parameter for model m. f is likelihood, π prior and π_t posterior

- SMC: Generate a weighted sample (particles) for each target in the sequence via steps
  - Reweight: particles as data comes in (efficient)
  - Resample: when ESS small
  - Mutation: diversify duplicated particles (can be efficient)

Sequential designs: SMC single static Model m (Algorithm) Chopin (2002)

- Have current particles $\{W_t^i, \theta_t^i\}_{i=1}^N$ based on data $y_{1:t}$
- Re-weight step to included y_{t+1}

$$W_{t+1}^i \propto W_t^i f(y_{t+1}|\theta_t^i, d_{t+1}),$$

- Check effective sample size: ESS = $1/\sum_{i=1}^N (W_{t+1}^i)^2$
- If ESS $> E$ (e.g. $E = N/2$) go back to re-weight step for next observation
- If ESS $< E$ do the following
- Resample proportional to weights. Duplicates good particles
- Mutation: Move all particles via MCMC kernel say R times (adaptive proposal)

SMC: Estimate of model evidence (Del Moral et al., 2006)

- It can be shown

$$Z_{t+1}/Z_t = f(y_{t+1}|y_{1:t}, d_{t+1}) = \int_\theta f(y_{t+1}|\theta, d_{t+1})\pi(\theta|y_{1:t}, \boldsymbol{D_t})d\theta.$$

- Using SMC particles to approximate posterior at t gives estimator

$$Z_{t+1}/Z_t \approx \sum_{i=1}^{N} W_t^i f(y_{t+1}|\theta_t^i, d_{t+1}).$$

- Can then obtain approximation of Z_{t+1} through

$$\frac{Z_{t+1}}{Z_0} = \frac{Z_{t+1}}{Z_t}\frac{Z_t}{Z_{t-1}}\cdots\frac{Z_1}{Z_0}.$$

- Also gives estimate of posterior predictive probability of y_{t+1}

The Algorithm (Using SMC to design under model uncertainty)

- Effectively run an SMC algorithm for each model $m = 1, \ldots, K$
- Have set of N particles for each model $\{W^i_{m,t}, \theta^i_{m,t}\}_{i=1}^N$.
- ESS for each model m
- resampling and within-model updates when required
- Design part: use data up to t, $y_{1:t}$, and particles of all models to compute the next design d_{t+1}
- Algorithm has been used by a variety of authors, e.g. discrete choice experiments (Consonni, Deldossi and Saggini, 2018)

Sequential/static designs - Laplace approximation

- Long et al. (2013) and Overstall, McGree and Drovandi (2018) used the Laplace approximation for efficiently estimating $u(d, m, y)$;
- The main result is that the approximation to the posterior distribution of $\boldsymbol{\theta}_m$ has the following multivariate Normal form:

$$\hat{p}(\theta_m|y, d, m) = (2\pi)^{-\frac{q_m}{2}} |\boldsymbol{\Sigma}_{my}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\theta_m - \hat{\theta}_{my})^t \boldsymbol{\Sigma}_{my}^{-1}(\theta_m - \hat{\theta}_{my})\right),$$

where q_m denotes the number of parameters in model m, $\hat{\boldsymbol{\theta}}_{my}$ and $\boldsymbol{\Sigma}_{my}$ denote the posterior mode and posterior variance-covariance matrix, respectively, for model m upon the observation of $\boldsymbol{y}$ from design $\boldsymbol{d}$, for $m = 1, 2, \ldots, K$.

Laplace approximation

- For posterior inference on m, the posterior model probability ($p(m|y,d)$) can be considered;
- Proportional to the model evidence;
- Based on Laplace approximation, the model evidence can be approximated as follows:

$$\hat{p}(y|m,d) = (2\pi)^{\frac{q_m}{2}} |\boldsymbol{\Sigma}_{my}|^{\frac{1}{2}} p(y|\hat{\theta}_{my}, d) p(\hat{\theta}_{my}|m). \qquad (2)$$

- Thus, posterior summaries such as $u(d,m,y)$ can be evaluated based on the above Laplace approximation facilitating a relatively efficient approximation to $u(d)$;
- Extensions to more complicated models are facilitated by the nested-integrated laplace approximation (Rue, Martino and Chopin, 2009)

Locating Bayesian designs

Difficult optimisation problem

- Noisy and expensive utility function
- Only get realisations from function
- Potentially high dimensional

Some approaches

- Exhaustive search
- Mueller algorithm (Mueller, 1999)
- Approximate coordinate exchange algorithm (ACE, Overstall and Woods, 2017)

Design and big data

Consider a general regression set up:

- Response data y_i
- p predictors (design) d_i
- $i = 1, \dots, N$ observations

Our objective is to avoid the analysis of the big data of size N by selecting a subset of size n

The algorithm (sequential design)

Our approach:

1: Use training sample to obtain $p(\theta)$
2: **while** $n_c \leq n$ or analysis aim has not been met (where n_c is the current sample size) **do**
3: Find $d_t^* = \arg\max_d u(d)$
4: Find x_t in data set such that $||x_t - d_t||$ is minimised
5: Selected x_t and corresponding y_t, and append to current data set
6: Update $p(\theta)$, and remove (x_t, y_t) from data set
7: **end while**

The algorithm (sequential design)

- Finding an optimal sample of n from N would involve a comparisons across all $\binom{N}{n}$ potential designs
- This would be computationally prohibitive
- As an alternative, we first search over $d \in D$, then search the big data for x closely to d^*
- This is computationally efficient
- However, assumes efficiency is related to distance across all variables
- Potentially sub-optimal but pragmatic

The algorithm (sequential design)

- Line 3 of the algorithm is most challenging
- If covariate space is relatively small, then potentially exhaustive search could be used
- Otherwise, approaches discussed above could be implemented
- This is the most computationally intensive step of the algorithm
- The most limiting factor for general applicability of our algorithm
- This step should not be so computationally intensive that it removes the computational advantages of our approach
- Line 4 is relatively straightforward

Example 1 - Mortgage example

- Consider mortgage default data
- Data consist of:
  - default: binary variable indicating whether or not the mortgage holder defaulted on the loan
  - creditScore: a credit rating (x1)
  - yearsEmploy: the number of years the mortgage holder has been employed at their current job (x2)
  - ccDebt: the amount of credit card debt (x3)
  - houseAge: the age (in years) of the house (x4)
- For the year 2000, there are 1 million records
- Propose logistic regression model for response variable
- Goal is to determine which covariates are useful for prediction

Example 1 - Mortgage example

- To obtain $p(\theta)$, an initial training sample of 5,000 was randomly selected from the data set
- The subsequent posterior distribution was used as the prior for design
- Next we value added to the information gained from the initial learning phase through sequential design
- Here, we implemented the SMC algorithm of Drovandi, McGree and Pettitt (2014)

Example 1 - Mortgage example

- An estimation utility was considered ($-\log \det p(\theta|y, d)$)
- Importance sampling was used to estimate $u(d|y_{1:t-1}, d_{1:t-1})$
- To locate the optimal design, an exhaustive search was implemented over all combinations of discrete values of the predictors
- As we are interested in determining which variables are useful for prediction, 95% credible intervals were formed for each slope parameter in every iteration of our algorithm
- If any credible interval was contained within $(-tol, tol)$, then this predictor was dropped from the model
- Values of *tol* were $0.25, 0.5, 0.75, 1.0$
- Algorithm iterated until 1,000 observations had been selected from the data set
- Note: $n << N$

Example 1 - Mortgage example

The covariates in the mortgage case study which were deemed useful for prediction based on $tol = 0.25, 0.50, 0.75$ *and* 1.00

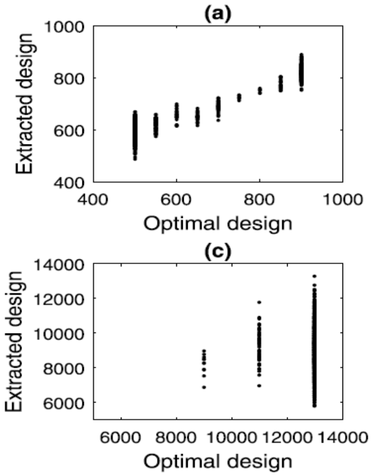
tol	**Remaining covariates**
0.25	x_1, x_2, x_3, x_4
0.50	x_2, x_3
0.75	x_3
1.00	x_3

Example 1 - Mortgage example

Summary of the posterior distribution of the parameters for the full main effects model based on all mortgage default data for the year 2000

Parameter	Mean	SD	2.5th	Median	97.5th
β_0	−11.40	0.13	−11.67	−11.40	−11.16
β_1	−0.42	0.03	−0.48	−0.42	−0.36
β_2	−0.63	0.03	−0.68	−0.63	−0.56
β_3	3.03	0.05	2.94	3.03	3.13
β_4	0.20	0.03	0.12	0.20	0.26

Example 1 - Mortgage example



Example 2 - Accelerometer example

- 212 participants performed a series of 12 different activities at four different time points
- Participants ranged in age between 5 and 18 years
- The purpose was to assess the performance of 4 different so-called “cut-points”, which are used to predict the type of activity performed based on the output of the accelerometer
- Response variable was whether or not the cut-point correctly classified the activity
- Each individual at each time point performed all 12 activities and all 4 cut-points are applied
- Approximately 35,000 data points

Example 2 - Accelerometer example

- A logistic regression mixed effects model was considered
- Predictors are age, type of activity (12 levels) and cut-points (4 levels)

$$
\begin{aligned}
\text{logit}\,\pi_{ti} &= \beta_0 + b_t + \beta_1 \text{age}_{ti} + \sum_{j=1}^{3} \beta_2^j \text{cut}_{ti}^j + \sum_{j=1}^{11} \beta_3^j \text{trial}_{ti}^j \\
&+ \sum_{j=1}^{33} \beta_4^j \text{cut}_{ti}^j \times \text{trial}_{ti}^j + \sum_{j=1}^{3} \beta_5^j \text{age}_{ti}^j \times \text{cut}_{ti}^j \\
&+ \sum_{j=1}^{11} \beta_6^j \text{age}_{ti}^j \times \text{cut}_{ti}^j
\end{aligned}
$$

where $Y_{ti} \sim \text{Binary}(\pi_{ti})$, $b_t \sim N(0, \phi)$, for $t = 1, \ldots, 212$, $i = 1, \ldots, s_t$ (number of observations on subject t), cut_{ti}^j and trial_{ti}^j are dummy variables.

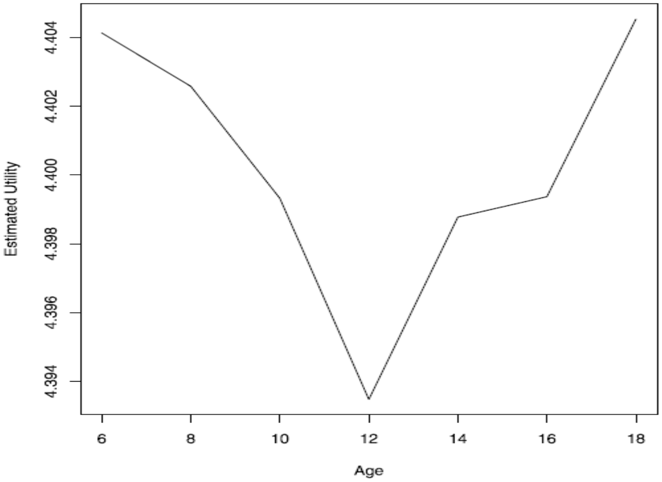
Example 2 - Accelerometer example

- Interested in estimating the age effect for correct classification
- Both main and interaction effects, total of 15 parameters
- Ds-optimality considered (similarly defined as in Example 1)
- Initial training sample was found by randomly sampling individuals to obtain (approximately) 500 observations
- Sequential design iterated until 3,000 observations were obtained
- The optimisation (line 3) was a simple grid search over age
- For line 4, we selected the individual who had the closest age to the optimal (48 observations if full replicate was available)

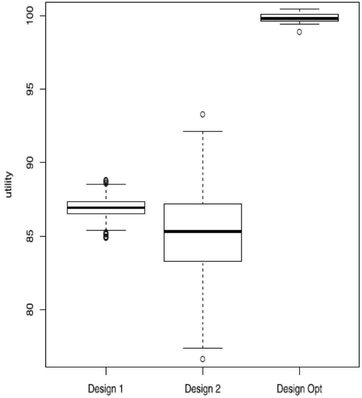
Example 2 - Accelerometer example

- For fast posterior inference, INLA was used
- Further computational efficiency was achieved by using a small number of Monte Carlo draws
- Only interested in estimating $u(d)$ well enough to determine optimal age
- Thus, can sacrifice precision for computational efficiency

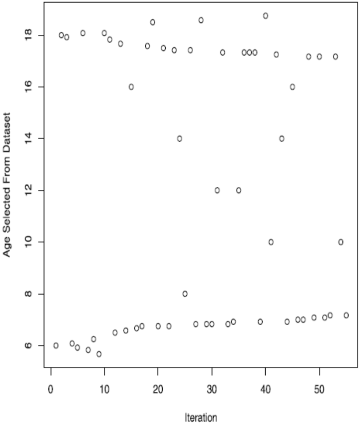
Example 2 - Accelerometer example



Example 2 - Accelerometer example



(a) distribution of observed utilities



(b) selected ages

Conclusion

- Proposed an approach to make big data problems more tractable
- Examples showed promise
- Need to consider the trade-off between analysing the big data versus finding the optimal design. Is it worth it?
- Initial work in this space. Many questions/issues remain that we did not tackle

Future research

- Approximate bias in big data sample
- Handle streaming data - Design templates?
- Model that can vary/evolve over time?
- Model uncertainty/criticism
- Model-free design?

Selected references

- Box and Hill (1967). Technometrics, 9, 57-71.
- Chopin (2002). Biometrika, 89, 539-551.
- Del Moral, Doucet and Jasra (2006) JRSS(B), 68, 411-436.
- Drovandi, McGree, and Pettitt (2014) JCGS, 23, 3-24.
- Hill (1978). Technometrics, 20, 15-21.
- Overstall, McGree and Drovandi (2018) Statistics and Computing, 28, 343-358.