

APES: Approximated Exhaustive Search for GLM

Kevin Y.X. Wang

School of Mathematics and Statistics
The University of Sydney

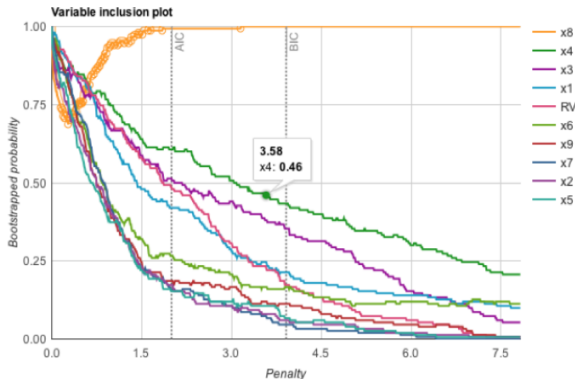
Peking University - Beijing
29 December 2019



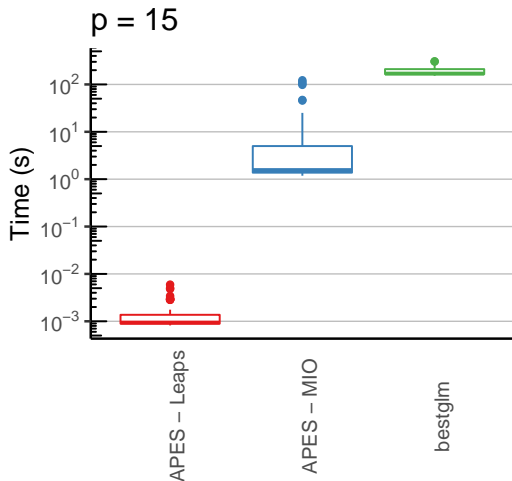
@KevinWang009

Acknowledgement

- This is joint work with Prof Samuel Müller, Dr Garth Tarr and Prof Jean Yang.
- `mpIot` Tarr2018 is a package to assess model stability and variable selection for linear models and generalised linear models.



My talk in one slide



- Exhaustive variable selection for GLM is slow.
- APES improves the speed of exhaustive search using approximations.
- APES is now published at ANZJS: <https://doi.org/10.1111/anzs.12276>.

Background

Data and models

- $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$, with independent $y_i \in \{0, 1\}$, $i = 1, \dots, n$.
- Design matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p) \in \mathbb{R}^{n \times p}$.
- Index the columns of \mathbf{X} by $\{1, \dots, p\}$.
- Let α denote any subset of p_α distinct elements from $\{1, \dots, p\}$. Use \mathcal{A} to denote the collection of all α , so $|\mathcal{A}| = 2^p$.
- \mathbf{X}_α denote the $n \times p_\alpha$ matrix with columns given by the columns of \mathbf{X} whose indices appear in α .

Logistic regression

- We model the conditional response variable $Y_i|\mathbf{X}$ as Bernoulli(π_i), where $\pi_i = \mathbb{P}(Y_i = 1|\mathbf{X})$.
- We will use the **logistic function** as our link function, so $\mathbf{x}_i^\top \boldsymbol{\beta} = \text{logit}(\pi_i) = \ln(\pi_i/(1 - \pi_i))$.
- Model fitting usually involves estimating $\boldsymbol{\beta}$ (or equivalently, $\boldsymbol{\pi}$).

Iterative Reweighted Least Square (IRLS)

1. Denote weights $w_i = \pi_i(1 - \pi_i)$ and other estimates at the t -th iteration with a superscript (t) .

2. Construct

$$z_i^{(t)} = \underbrace{\text{logit} \left(\pi_i^{(t)} \right)}_{\mathbf{x}_i^\top \boldsymbol{\beta}^{(t)}} + \frac{y_i - \pi_i^{(t)}}{\pi_i^{(t)}(1 - \pi_i^{(t)})}.$$

3. Update via

$$\hat{\boldsymbol{\beta}}^{(t+1)} \leftarrow (\mathbf{X}^\top \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(t)} \mathbf{z}^{(t)},$$

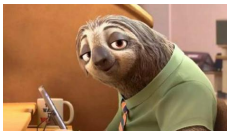
with $\mathbf{W}^{(t)} = \text{diag} \left(w_i^{(t)} \right)$.

4. At convergence, $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{z}$, which equals to the MLE of logistic regression model.

Challenges in exhaustive GLM variable selection

For large p , exhaustive variable selection in GLM is difficult:

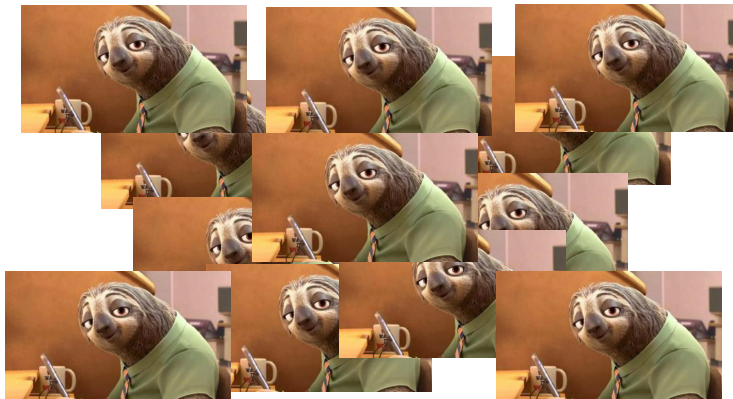
1. The computational cost of IRLS is $\mathcal{O}(np^2)$ per iteration.



Challenges in exhaustive GLM variable selection

For large p , exhaustive variable selection in GLM is difficult:

1. The computational cost of IRLS is $\mathcal{O}(np^2)$ per iteration.
2. We need to explore all 2^p models.



Aim of APES

Can we perform linear exhaustive variable selection (which benefits from fast algorithms) and use the results to approximate exhaustive GLM variable selection?

GLM exhaustive $\xleftarrow{\text{Can this be done?}}$ LM exhaustive

1. Turning GLM to LM

Exhaustively computing modified MLE

- (?) described an approximation to exhaustive variable selection for logistic regression without the need for numerical optimisation.
- Their method starts with the estimated probability $\hat{\pi}(\alpha_f)$, from the **full** logistic model.
- Then, for each model $\alpha \in \mathcal{A}$, we calculate:

$$\hat{\beta}(\alpha; \hat{\pi}(\alpha_f)) = (\mathbf{X}_{\alpha}^{\top} \mathbf{W}(\hat{\pi}(\alpha_f)) \mathbf{X}_{\alpha})^{-1} \mathbf{X}_{\alpha}^{\top} \mathbf{W}(\hat{\pi}(\alpha_f)) \mathbf{z}(\hat{\pi}(\alpha_f)).$$

- This is **NOT** the MLE for α , which should be $\hat{\beta}(\alpha; \hat{\pi}(\alpha))$.

Variable selection using the modified estimator

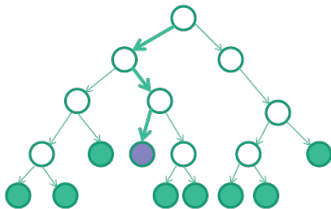
- Given $\hat{\beta}(\alpha; \hat{\pi}(\alpha_f))$, we could **approximate** RSS or BIC for all $\alpha \in \mathcal{A}$.
- Upon selection of a small set of desired models, we can recompute the MLE and calculate other model fit statistics.

GLM exhaustive $\xleftarrow{\text{Hosmer approx.}}$ LM exhaustive

2. Reducing computational time

Best subsets search

- Application of this approximation method is limited by the number of LMs we can explore.
- For $p \approx 50$, \mathcal{A} is approximately 1 quadrillion in size, which is too large to explore exhaustively.
- A best subset algorithm limits our search to a subset of \mathcal{A} but it guarantees to contain the global RSS-optimal model.
- **leaps** Furnival1974, Lumley2017 discard “branches” of models with insufficient fit.



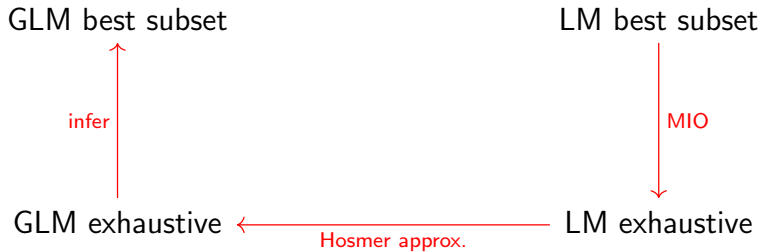
Mixed Integer Optimisation

- Bertsimas2016 showed that it is feasible to perform best subset search for linear models with p in the hundreds.
- The most attractive component: guaranteed **sub-optimality** if algorithm is terminated before full convergence. Thus allowing a upper bound for real time limit.
- Current implementation in R is bestsubset, Hastie2017, which outputs the RSS-best linear model for each model size.

APES: Approximated Exhaustive Search

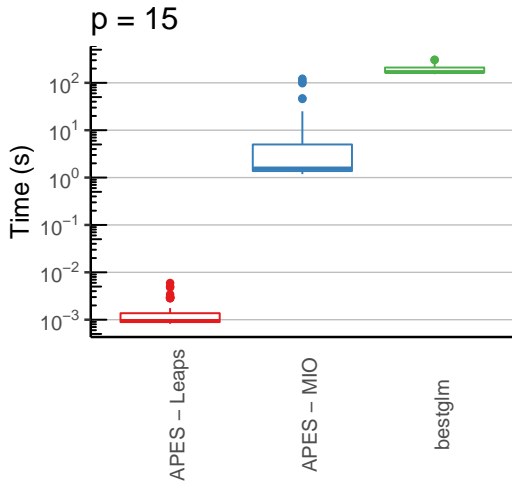


APES: Approximated Exhaustive Search



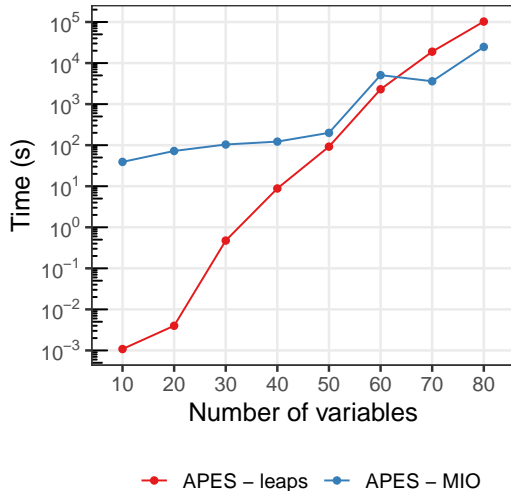
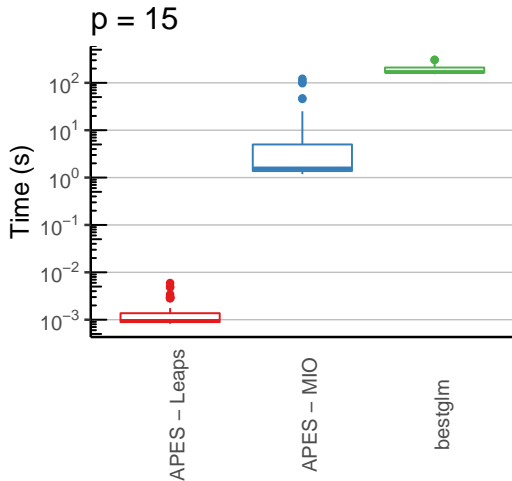
How fast is APES compare to genuine exhaustive search?

Very.



How fast is APES compare to genuine exhaustive search?

Very.



Simulation

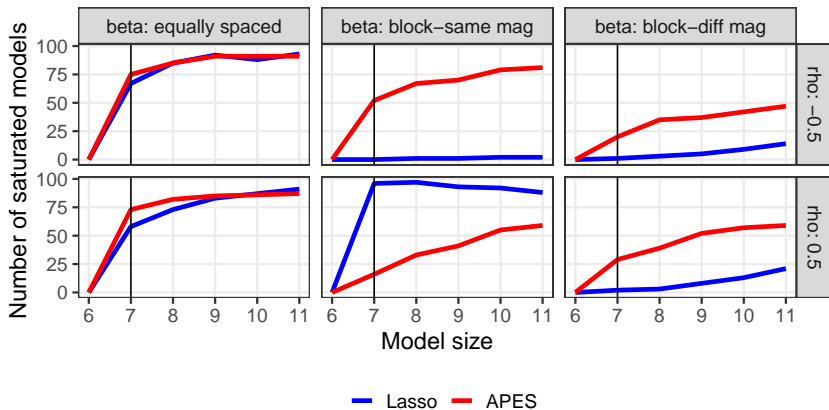
Simulation set-up

- $n = 500, p = 100$, number of non-zero coefficient is $k = 6$.
- Intercept term is set to 0, then we tried 3 different choices of β :
 1. Equally spaced indices:
 $(\frac{1}{2}, 0, \dots, 0, \frac{1}{2}, 0, \dots, 0, \frac{1}{2}, 0, \dots, 0, \frac{1}{2}, 0, \dots, 0, \frac{1}{2}, 0, \dots, \frac{1}{2})$.
 2. Block of indices, same magnitude/sign: $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \dots, 0)$.
 3. Block of indices, different magnitude/sign: $(\frac{1}{3}, -1, 1, \frac{2}{3}, -\frac{2}{3}, -\frac{1}{3}, 0, \dots, 0)$.
- Generating $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, with $\Sigma_{ij} = \rho^{|i-j|}$, $\rho = 0.5$ or -0.5 . Then standardise.
- We repeated the simulation 100 times and compared APES against de-biased Lasso using various evaluation metrics.

Evaluation 1: saturated models

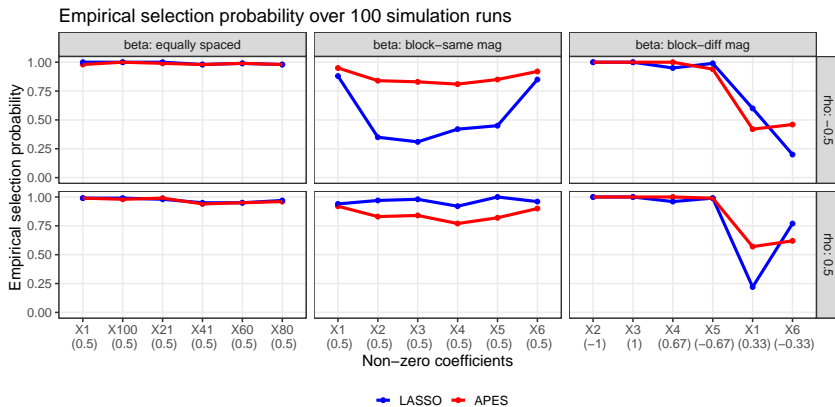
- For each model size, APES and Lasso outputs one model.
- In most cases, APES has less false exclusion of variables than Lasso.

Number of saturated models by each methods



Evaluation 2: variable selection

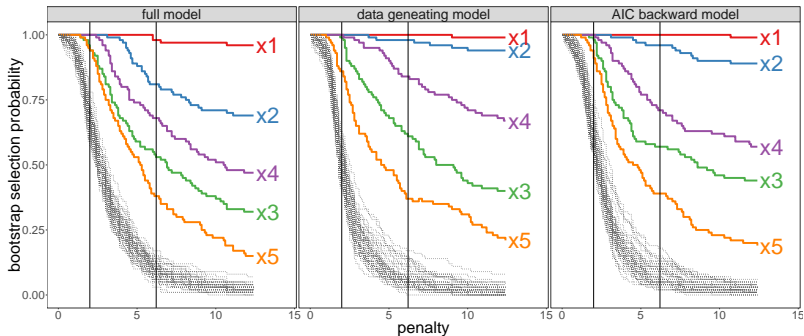
- BIC was used to select an optimal model.
- In most cases, APES has higher selection probability of active variables than Lasso.



Some extensions

The choice of $\hat{\pi}(\alpha_f)$ can be relaxed in two different ways:

- The full model α_f is not necessarily the best for variable selection.



- We could replace the MLE by other estimators, e.g. the Lasso or robust quasi-likelihood estimator.

Final remarks

1. APES is a **fast approximation** method for **exhaustive** variable selection in GLM.
2. APES pushes model dimensions into the hundreds and serves as a standard of comparison like a true exhaustive search.
3. APES is now published at ANZJS:
<https://doi.org/10.1111/anzs.12276>.
 - <https://github.com/kevinwang09/APES>
 - <https://github.com/garthtarr/mplot>
 - Email: kevin.wang@sydney.edu.au.
 - Twitter: @KevinWang009

References