Sequential Monte Carlo for Approximate Variable Selection in Generalized Linear Models

Antoine Carnec, Maxim Fedotov

Supervised by:
David Rossell and Jack Jewson

Barcelona School of Economics,
Data Science Methodology

Introduction

- In a regression with p covariates, there are 2^p possible models to consider.
- We propose an algorithm that aims to find a 'shortcut' to sample from the LA model posterior.
- We take advantage of a recent computational tool developed by Rossell et al. (2021), the Approximate Laplace Approximation.

Motivation for the practitioner

- Principled handling of model uncertainty.
- Bayesian framework allows flexible modelling.
- Applications of GLMs:
  - Logistic - Binary outcomes (Probabilities)
  - Gamma - Positive Continuous data (Environmental Sciences)
  - Poisson - Count Data (Transport)

Our Contribution

- Development of a novel model selection sampling algorithm.
- Successfully samples from LA model posterior in our simulations
- The algorithm scales well with p.
- We provide a complete implementation of this methodology in Python, built with modularity in mind.

  `https://github.com/antotocar34/masters_project`

Overview

Setting Up Notation

- $y \in \mathbb{R}^n$ - outcome vector
- $X \in \mathbb{R}^{n \times p}$ - design matrix
- $\boldsymbol{\beta} \in \mathbb{R}^p$ - coefficient vector

$$h(\mathbb{E}\left[y_i \mid x_i\right]) = x_i^T \boldsymbol{\beta}$$

- $\boldsymbol{\gamma} \in \{0,1\}^p$ - model vector

$$\boldsymbol{\gamma}_j = \mathbb{1}\{\beta_j \neq 0\},$$

- $\boldsymbol{\beta}_{\boldsymbol{\gamma}} \in \mathbb{R}^{p_{\boldsymbol{\gamma}}}$ - model-specific coefficient

The Inference Problem

$$p(\gamma \mid y) \propto \int_{\beta_\gamma} p(y \mid \beta_\gamma, \gamma) p(\beta_\gamma \mid \gamma) d\beta_\gamma \cdot p(\gamma)$$

- $p(y \mid \beta_\gamma, \gamma)$ - GLM Likelihood
- $p(\beta_\gamma \mid \gamma)$ - regression coefficient prior
- $p(\gamma)$ - model prior

The Inference Problem

$$p(\gamma \mid y) \propto \int_{\beta_\gamma} p(y \mid \beta_\gamma, \gamma) p(\beta_\gamma \mid \gamma) d\beta_\gamma \cdot p(\gamma)$$

- $p(y \mid \beta_\gamma, \gamma)$ - GLM Likelihood
- $p(\beta_\gamma \mid \gamma)$ - regression coefficient prior
- $p(\gamma)$ - model prior

Problem: Integral is costly to compute!

The Inference Problem

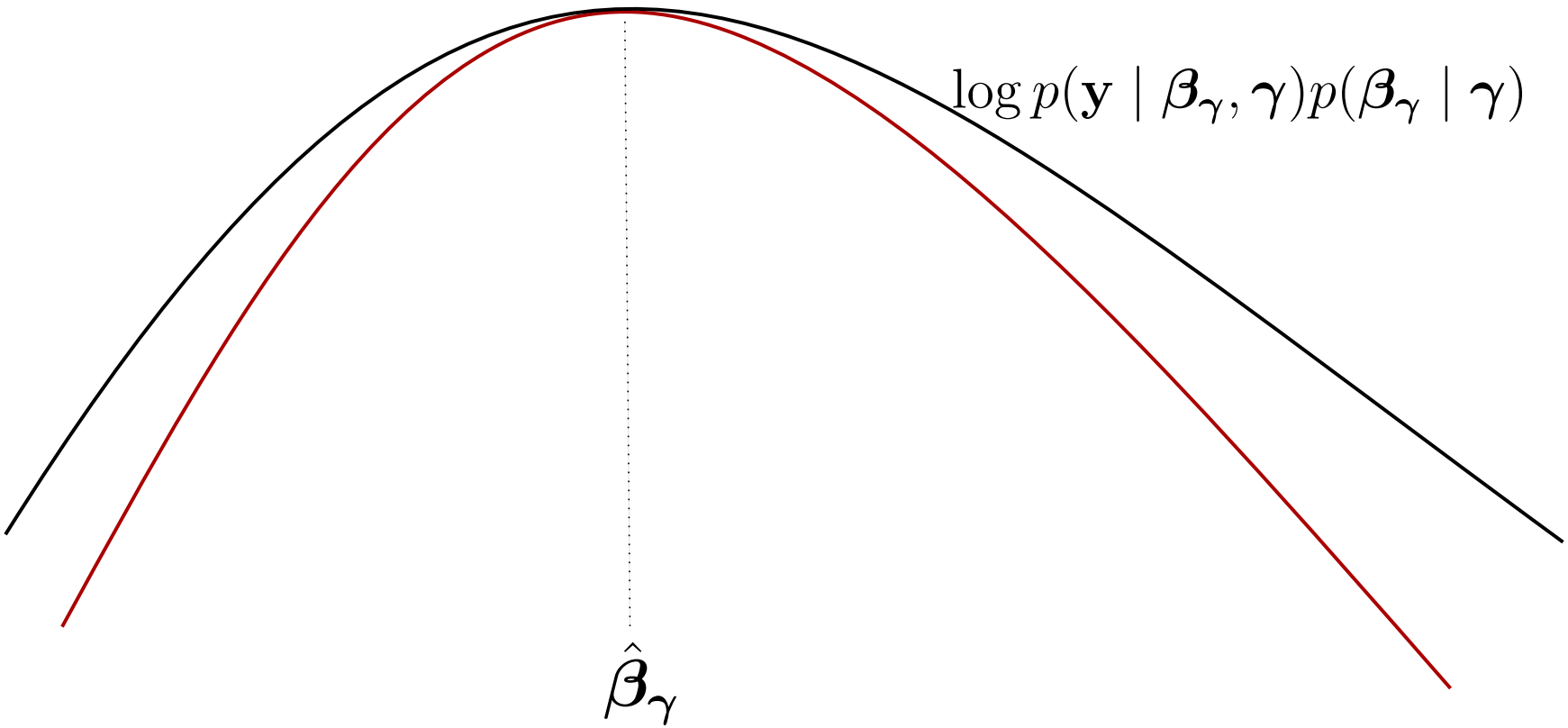
$$p(\gamma \mid y) \propto \int_{\beta_\gamma} p(y \mid \beta_\gamma, \gamma) p(\beta_\gamma \mid \gamma) d\beta_\gamma \cdot p(\gamma)$$

- $p(y \mid \beta_\gamma, \gamma)$ - GLM Likelihood
- $p(\beta_\gamma \mid \gamma)$ - regression coefficient prior
- $p(\gamma)$ - model prior

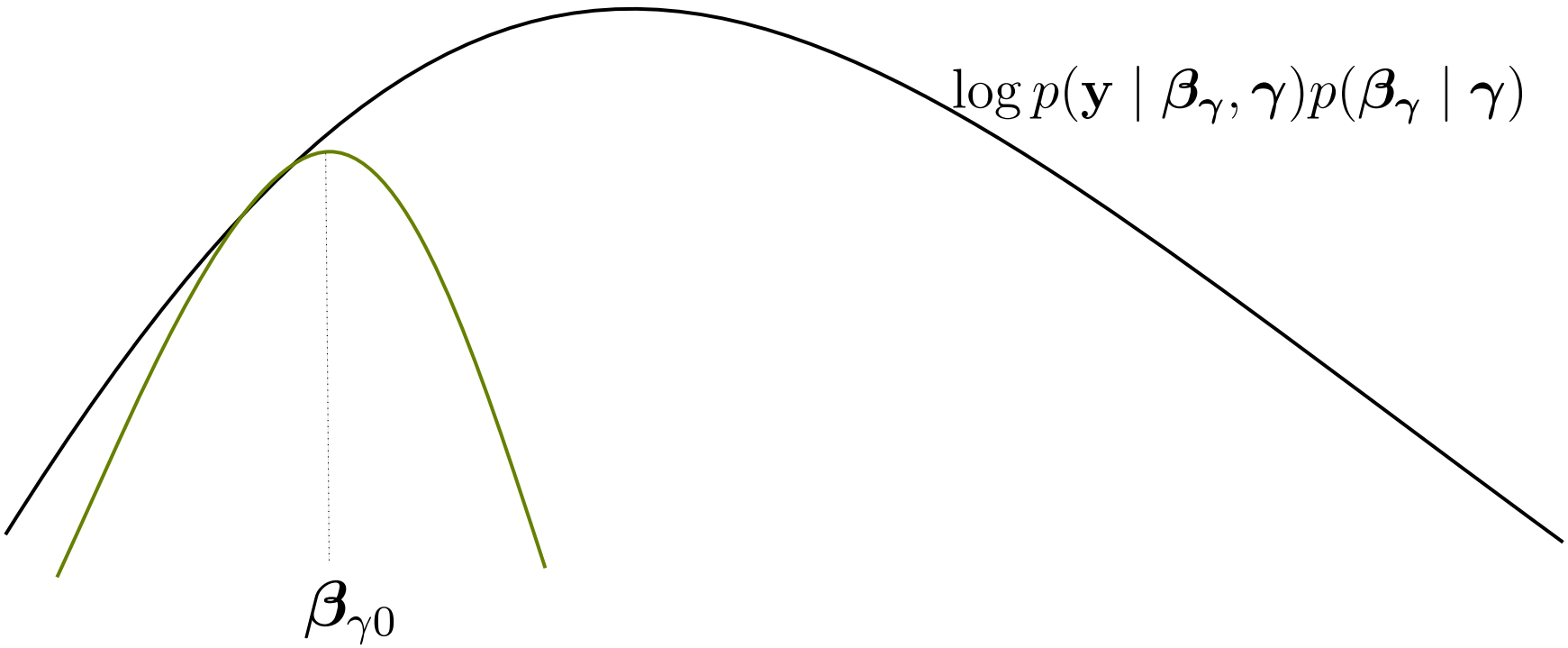
Problem: Integral is costly to compute!

Solution: Approximate the integral with LA and ALA

The Idea of Laplace Approximation (LA)



The Idea of Approximate Laplace Approximation



Model Selection Properties of LA & ALA

- It has been shown that the relative error in the Bayes factors decreases at $O(n^{-2})$ (Kass, 1990).
- The ALA-optimal model is 'close' to the optimal model recovered by the true posterior $p(\gamma \mid y)$.
- Unlike LA, ALA does not consistently estimate the model posterior $p(\gamma \mid y)$.

The Idea of our Algorithm

- The LA has excellent model selection properties, but it is **expensive** to compute.
- The ALA still has decent model selection properties, and it is relatively **cheap** to compute.

Idea: Can we sample from the ALA posterior and somehow transform this sample so that it constitutes a sample of the LA posterior?

The Idea of our Algorithm

- The LA has excellent model selection properties, but it is **expensive** to compute.
- The ALA still has decent model selection properties, and it is relatively **cheap** to compute.

Idea: Can we sample from the ALA posterior and somehow transform this sample so that it constitutes a sample of the LA posterior?

Yes, Sequential Monte Carlo Samplers allow us to do just this!

Sequential Monte Carlo Samplers (Del Moral et al., 2006)

- Given a sequence of distributions

$$\pi_0, \pi_1, \ldots, \pi_{T-1}, \pi_T$$

where it is feasible to sample from π_0 and to calculate the unnormalized density of π_t.

Sequential Monte Carlo Samplers (Del Moral et al., 2006)

- Given a sequence of distributions

$$\pi_0, \pi_1, \ldots, \pi_{T-1}, \pi_T$$

  where it is feasible to sample from π_0 and to calculate the unnormalized density of π_t.
- How does it work?
  - Sample from π_0.
  - Given a sample of π_t, get a sample of π_{t+1} using importance sampling.
  - Resample and apply an MCMC transition kernel.
  - Algorithm ends with a weighted sample $\{W_n, \gamma_n\}_{n=1}^N$ of π_T.

Sequential Monte Carlo Samplers (Del Moral et al., 2006)

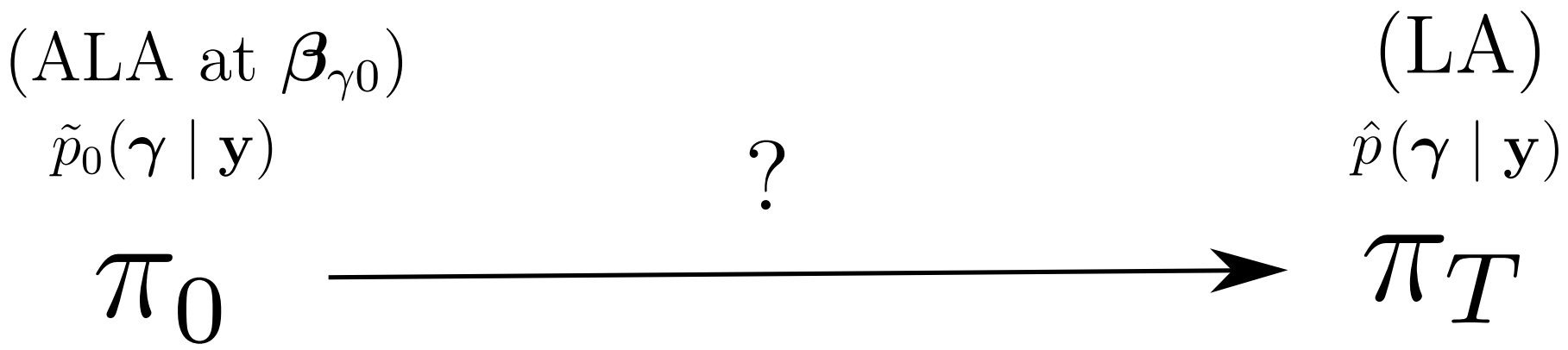
- Given a sequence of distributions

$$\pi_0, \pi_1, \ldots, \pi_{T-1}, \pi_T$$

  where it is feasible to sample from π_0 and to calculate the unnormalized density of π_t.
- How does it work?
  - Sample from π_0.
  - Given a sample of π_t, get a sample of π_{t+1} using importance sampling.
  - Resample and apply an MCMC transition kernel.
  - Algorithm ends with a weighted sample $\{W_n, \gamma_n\}_{n=1}^N$ of π_T.
- Then we have the following consistent estimator

$$\sum_{n=1}^{N} W_n \varphi(\gamma_n) \to \mathbb{E}_{\pi_T}[\varphi]$$

Idea of our SMC algorithm



How to define $\pi_1, \ldots, \pi_{T-1}$

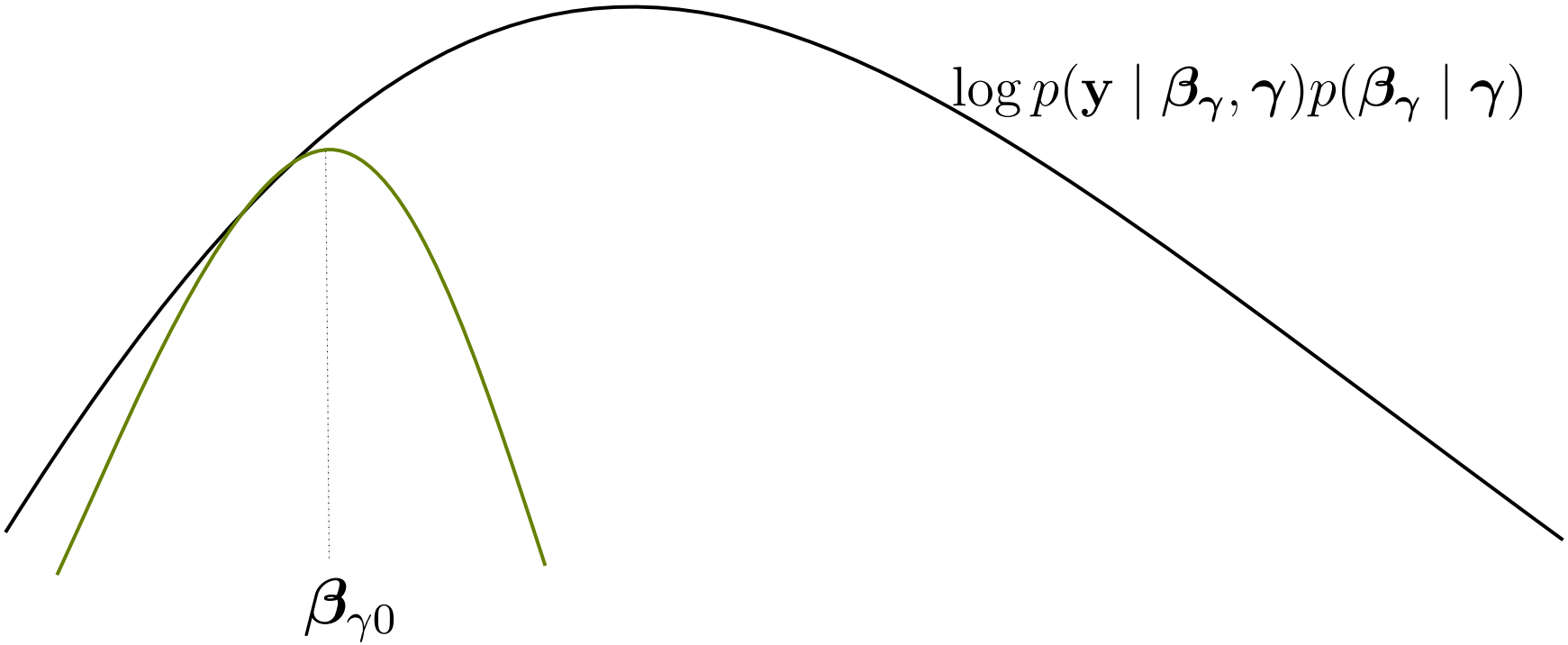
Observation: The ALA and LA are simply quadratic approximations around two different points. Some $\boldsymbol{\beta}_{\boldsymbol{\gamma}0}$ for the ALA and $\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ for the LA.

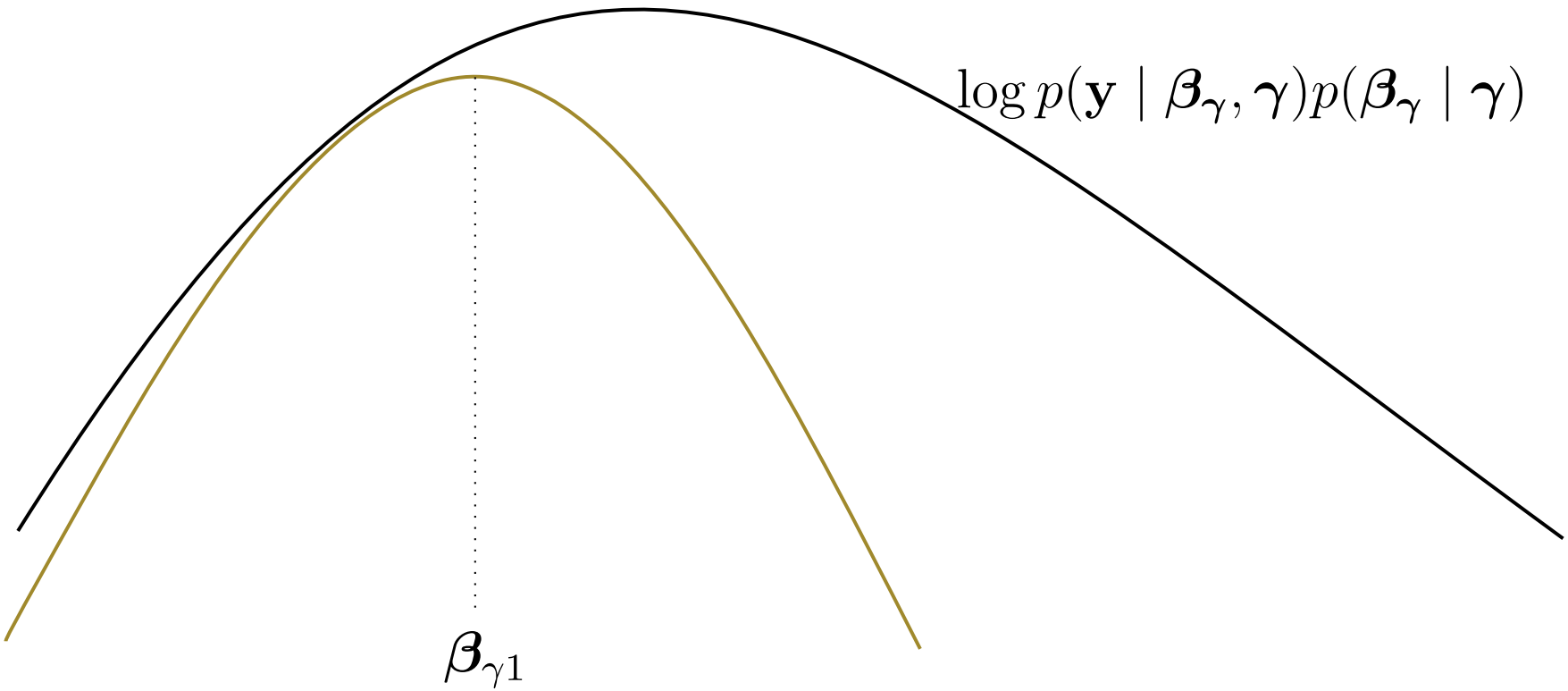
$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \underset{\boldsymbol{\beta}_{\boldsymbol{\gamma}}}{\operatorname{argmax}}\, p(\mathrm{y}, \boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}) \tag{1}$$

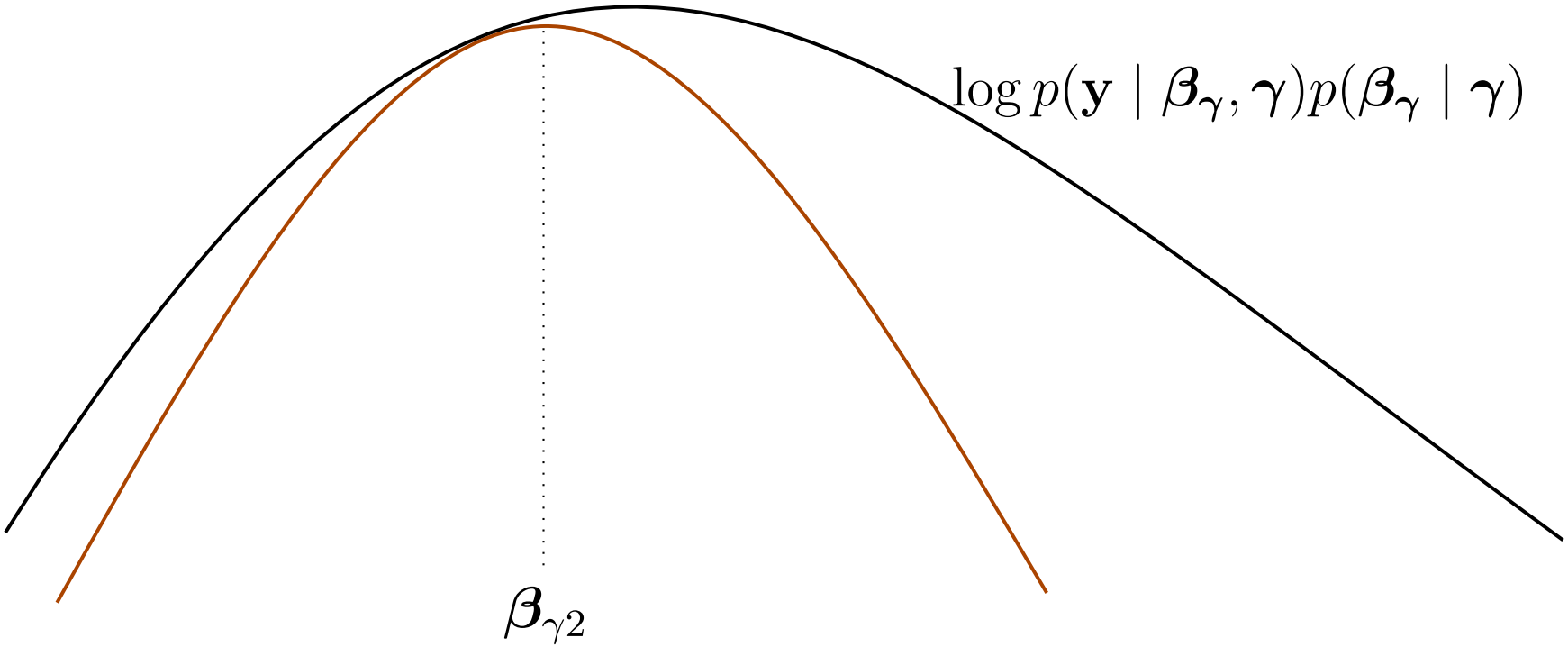
Idea: A descent-type optimization procedure on (1) defines a sequence $\{\boldsymbol{\beta}_{\boldsymbol{\gamma}j}\}$ from the ALA to LA.

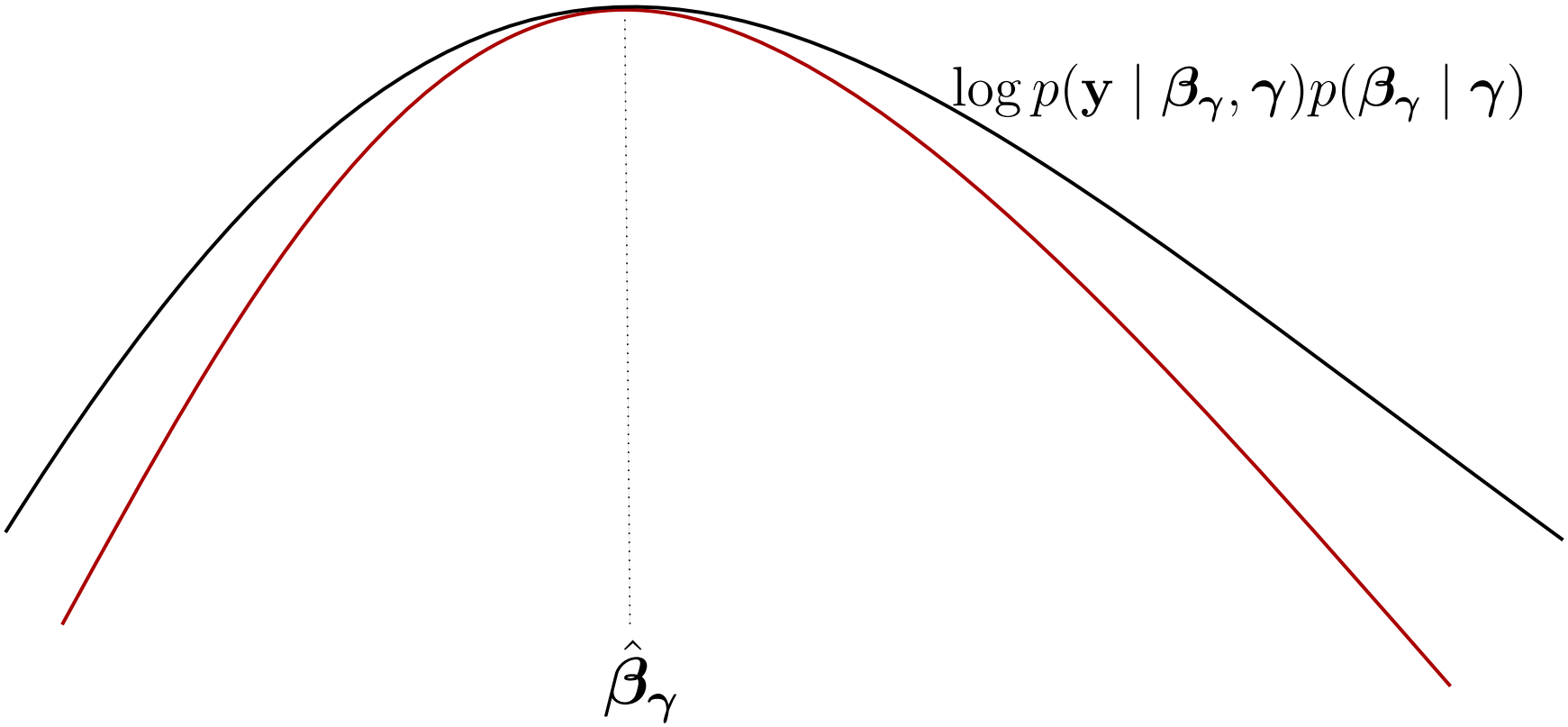
Define $\pi_t = \tilde{p}_t(\boldsymbol{\gamma} \mid \mathrm{y})$ to be the ALA taken around $\boldsymbol{\beta}_{\boldsymbol{\gamma}t}$ (the coefficient after t iteration of some optimization procedure). We use Newton-Raphson as an optimization procedure, hence

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}t+1} = \boldsymbol{\beta}_{\boldsymbol{\gamma}t} - H_{\gamma t}^{-1} g_{\gamma t}$$


$\log p(\mathbf{y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma})$
$\boldsymbol{\beta}_{\boldsymbol{\gamma}0}$


$\log p(\mathbf{y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma})$
$\boldsymbol{\beta}_{\boldsymbol{\gamma}1}$


$\log p(\mathbf{y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma})$
$\boldsymbol{\beta}_{\boldsymbol{\gamma}2}$


$\log p(\mathbf{y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma})$
$\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$

Some things to note

- Ideally the initial sample of π_0 will contain the active covariates.
- Evaluating the ALA approximation of a model takes $O(p_\gamma^3)$, as we have to invert a hessian. In a sparse setting, p_γ should be much lower than p.

Some things to note

- Ideally the initial sample of π_0 will contain the active covariates.
- Evaluating the ALA approximation of a model takes $O(p_\gamma^3)$, as we have to invert a hessian. In a sparse setting, p_γ should be much lower than p.
- If the initial sample misses out some truly active variables, the algorithm relies on the MCMC steps to find them in later iterations of the algorithm.
- We use a random scan Gibbs Kernel as our Markov Transition kernel.

Experimental Results. Data generation

We conduct several experiments to shed some light on the performance ALASMC performance and compare it to the LA.

In the experiments we consider two regression models: Logistic and Poisson.

Logistic regression

$$\theta_i = \frac{1}{1 + e^{-\mathrm{x}_i^T \boldsymbol{\beta}^*}}, \; y_i \mid \mathrm{x}_i, \; \boldsymbol{\beta}^* \sim \mathrm{Bern}(\theta_i)$$

Poisson regression

$$\lambda_i = e^{\mathrm{x}_i^T \boldsymbol{\beta}^*}, \; y_i \mid \mathrm{x}_i, \; \boldsymbol{\beta}^* \sim \mathrm{Pois}(\lambda_i)$$

Experimental Results. Posterior inclusion probabilities

How well does ALASMC target the LA posterior?

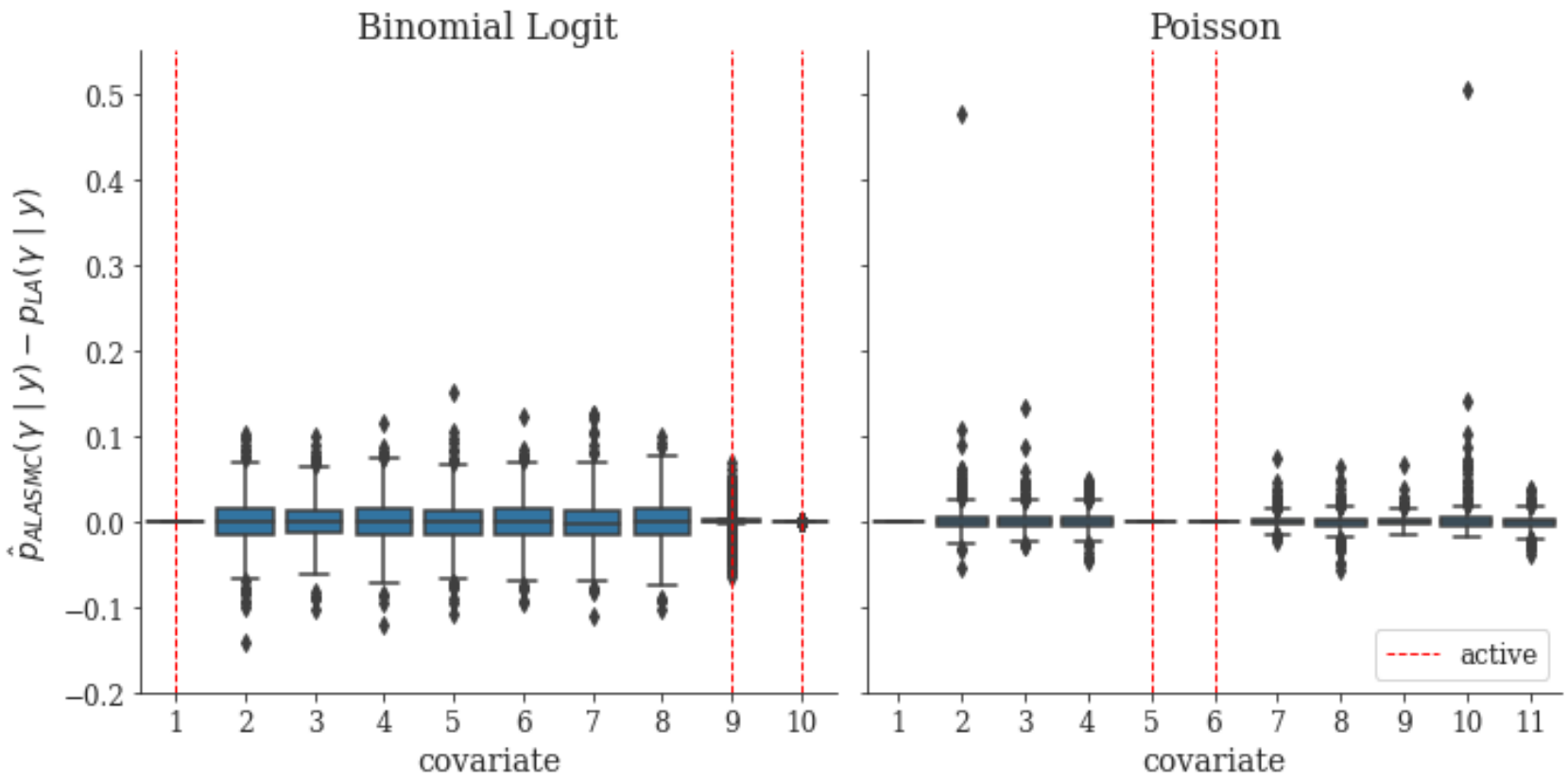
The data for this experiments is sampled in the following way:

- *Logistic:* $\beta^* = (2, 0_7, 0.5, 1)$, $X^{2:10} \sim \mathcal{N}(0, \Sigma)$
- *Poisson:* $\beta^* = (0_4, 0.5, 1, 0_5)$, $X^{2:6} \sim \mathcal{N}(0, \Sigma)$ and $X^{7:11}$ contains the squares of $X^{2:6}$

For both models: $\Sigma_{ii} = 1$ and $\Sigma_{ij} = 0.5$, $i \neq j$.

Experimental Results. Posterior inclusion probabilities

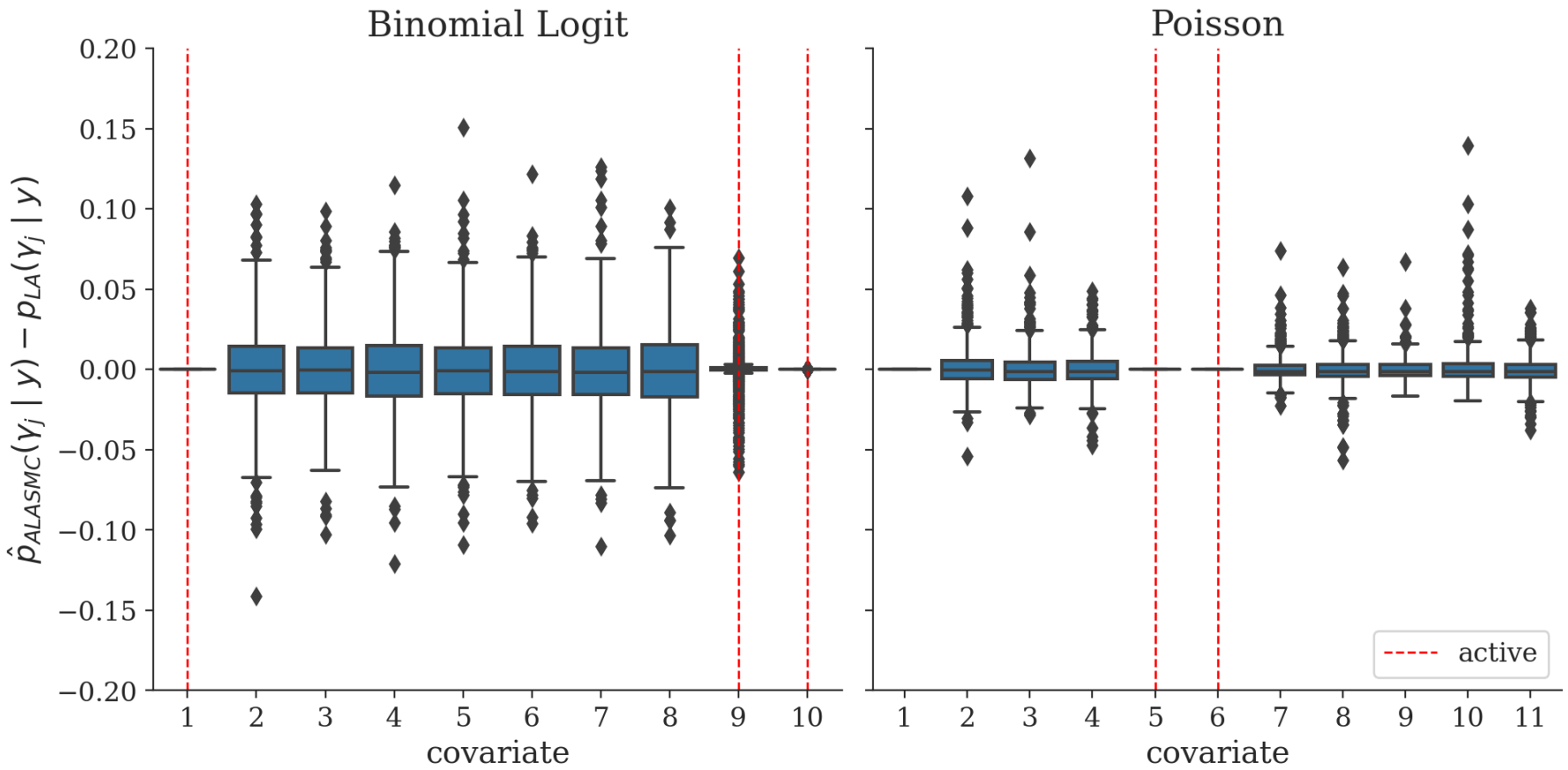
Figure: Difference between ALASMC and LA inclusion probabilities.



- Differences are centered at zero: ALASMC targets LA.
- Variance depends on the effect magnitude.
- There are extreme outliers, not too many.

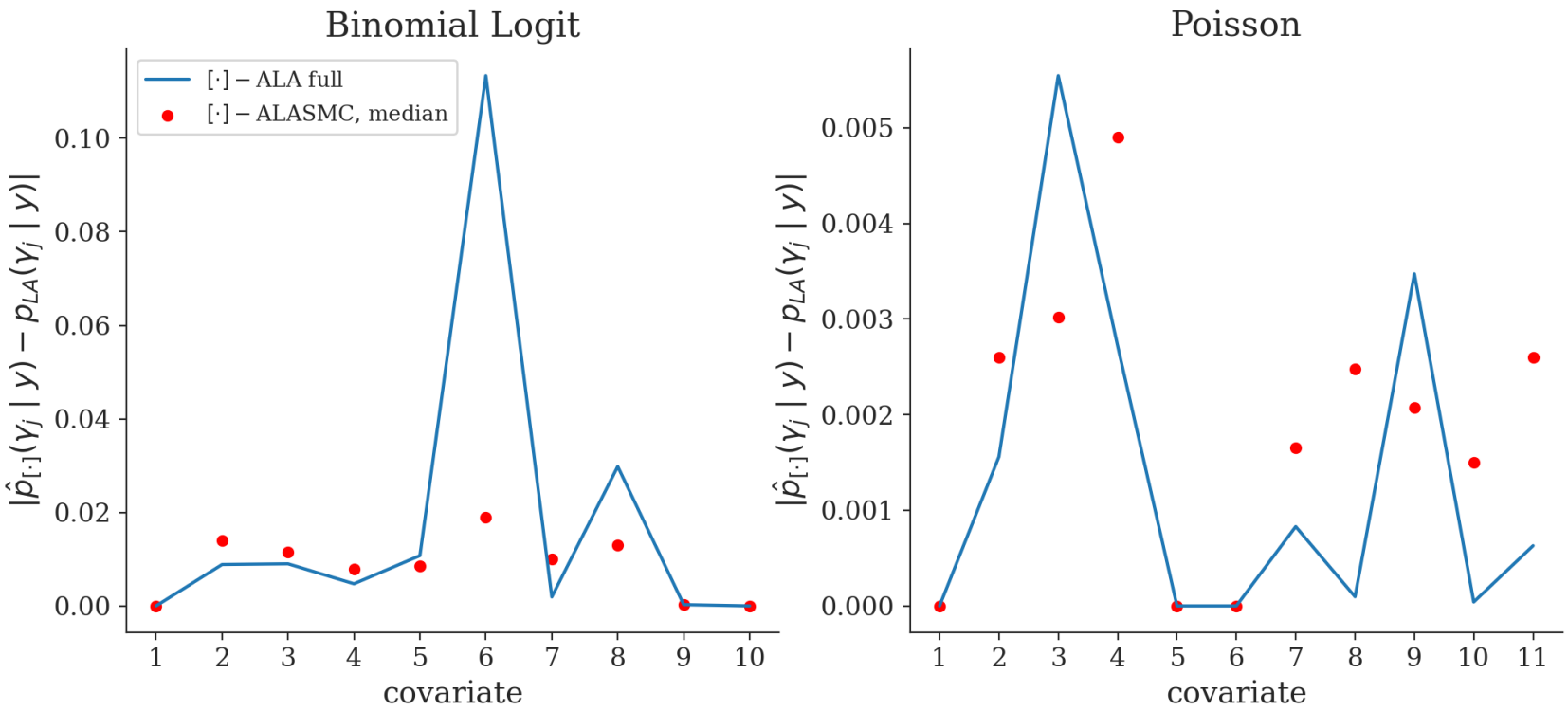
Experimental Results. Posterior inclusion probabilities

Figure: Difference between ALASMC and LA inclusion probabilities (without outliers).



Experimental Results. Precision

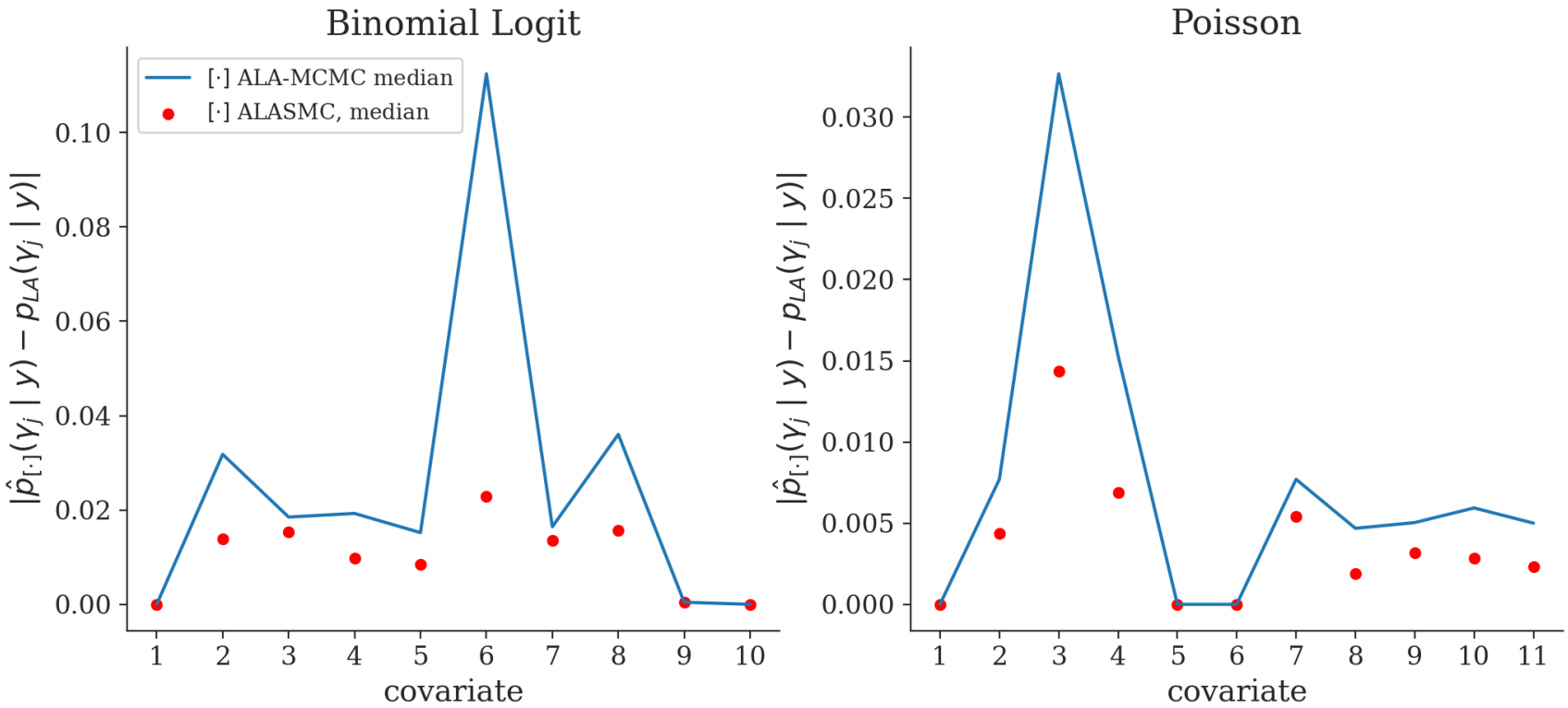
Figure: Difference between ALASMC and ALA inclusion probabilities.



- ALASMC does not necessarily give a uniformly better result than just one-step ALA with full enumeration.
- The difference depends on the parameters of the algorithm.

Experimental Results. Precision

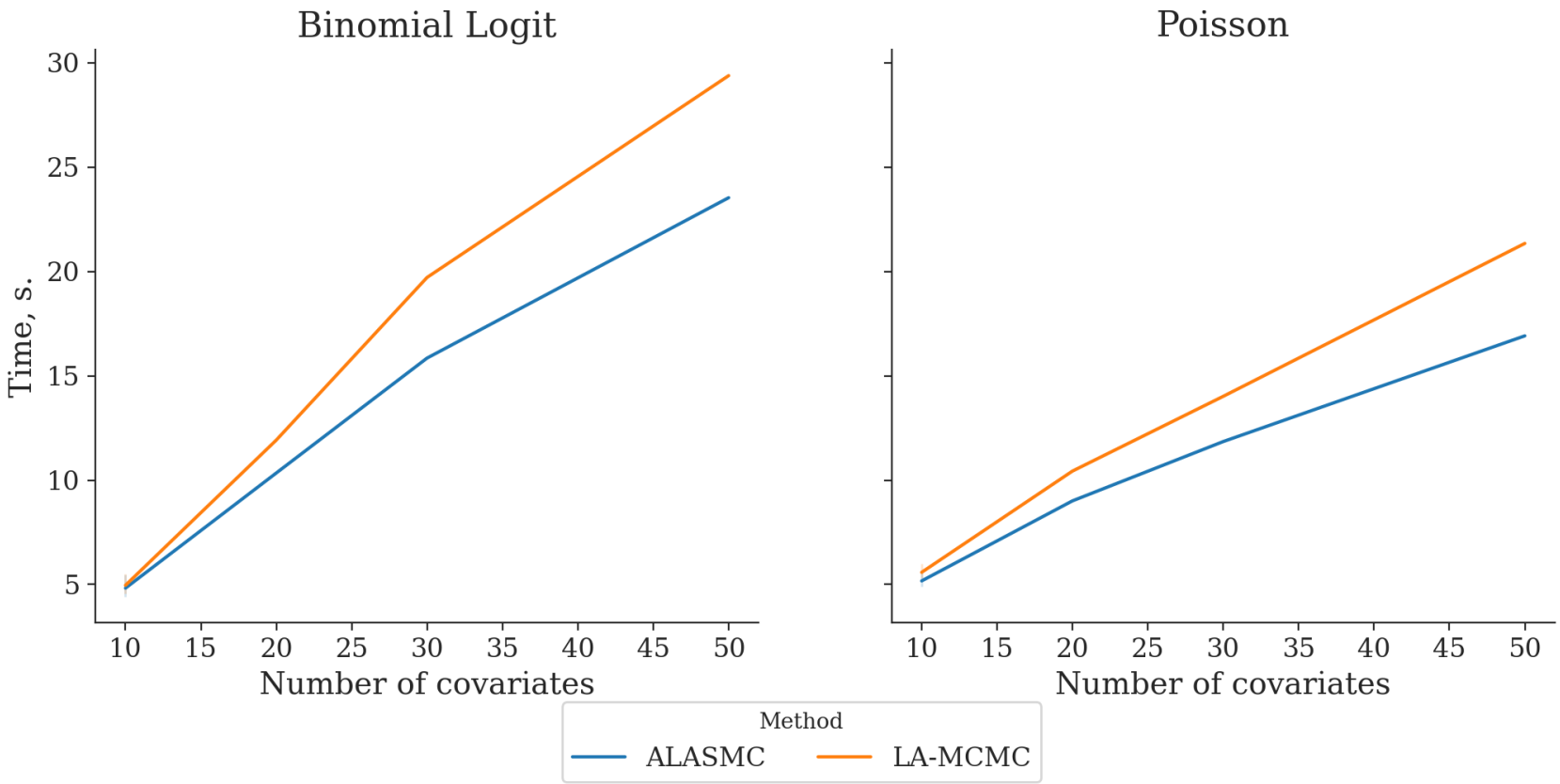
Figure: Difference between ALASMC and ALA-MCMC inclusion probabilities.



- But it successfully improves upon the posterior inclusion probabilities given by ALA-MCMC.

Experimental Results. Computational time

Figure: ALASMC and LA-MCMC computational time. ($n = 1000$)



- ALASMC gains in computation time.
- Difference becomes obvious when the number of covariates increase.
- Computational time comparison is tricky.

Summary of the results

- ALASMC targets the posterior given by the Laplace Approximation.
- It gains in computation time with respect to the Gibbs sampling algorithm with LA.
- The performance of the algorithm and the number of iterations until convergence depend on the initialization.
- The algorithm improves on the performance of MCMC with ALA.

Extensions / Further Work

- Better Kernels.
- Controlling ESS.
- Different optimization procedures instead of Newton-Raphson.

References

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

Kass, R. E. (1990). The validity of posterior expansions based on laplace's method. *Bayesian and likelihood methods in statistics and econometrics*, pages 473–487.

Rossell, D., Abril, O., and Bhattacharya, A. (2021). Approximate laplace approximations for scalable model selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(4):853–879.

Bayesian Regression

- Suppose that the data is generated by some distribution:

$$y_i \mid \boldsymbol{\beta}^*, \mathrm{x}_i \sim \mathcal{F}^*,$$

  where $\boldsymbol{\beta}^*, \mathrm{x}_i \in \mathbb{R}^p$ and $i = 1, \ldots, n$
- Let $\gamma \in \mathbb{B}^p$ be the indicator vector

$$\gamma_j = \mathbb{1}\{\beta_j \neq 0\},$$

  which denotes a model, i.e. a specific subset of the regressors.
- Denote the coefficients under a model γ as $\boldsymbol{\beta}_\gamma \in \mathbb{R}^{p_\gamma}$.
- The goal is to compute an insightful posterior probability for the considered models

$$p(\gamma \mid \mathrm{y}) = \frac{p(\mathrm{y} \mid \gamma)p(\gamma)}{\sum_{\gamma} p(\mathrm{y} \mid \gamma)p(\gamma)} \propto p(\mathrm{y} \mid \gamma)p(\gamma)$$

Bayesian Regression

- The goal is to compute a posterior probability for the considered models

$$p(\gamma \mid \mathrm{y}) = \frac{p(\mathrm{y} \mid \gamma)p(\gamma)}{\sum_{\gamma} p(\mathrm{y} \mid \gamma)p(\gamma)} \propto p(\mathrm{y} \mid \gamma)p(\gamma)$$

- The costly bit to compute here is the integrated likelihood:

$$p(\mathrm{y} \mid \gamma) = \int_{\beta_\gamma} p(\mathrm{y} \mid \beta_\gamma, \gamma)p(\beta_\gamma \mid \gamma)d\beta_\gamma$$

- One of the way to approximate this quantity is to use the Laplace Approximation (LA).

Generalized Linear Models

- Let $Y_1, \ldots, Y_n$ be independent random variables with observed values $\mathrm{y} = (y_1, \ldots, y_n)$, let $\mathrm{x}_i \in \mathbb{R}^p$ be the observed set of covariates of i-th sample.
- Denote the corresponding non-random design matrix as $X \in \mathbb{R}^{n \times p}$, and a parameter vector as $\boldsymbol{\beta} \in \mathbb{R}^p$.
- The generalized linear regression model is then given by the following expression:

$$h(\mathbb{E}\left[Y_i \mid \mathrm{x}_i\right]) = \mathrm{x}_i^T \boldsymbol{\beta}$$

- The corresponding likelihood for a model $\boldsymbol{\gamma}$ is

$$p(\mathrm{y} \mid \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) = \exp\left\{ \mathrm{y}^T X_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{\boldsymbol{\gamma}} - \sum_{i=1}^{n} b(\mathrm{x}_{i,\boldsymbol{\gamma}}^T \boldsymbol{\beta}_{\boldsymbol{\gamma}}) + \sum_{i=1}^{n} c\left(y_i\right) \right\},$$

where $b(\cdot)$ is infinitely differentiable.

Prior distributions

- We use a Beta-Binomial(1, 1) prior on the coefficients:

$$p(\boldsymbol{\gamma}) = p\left(|\gamma|_0\right) p\left(\gamma \mid |\gamma|_0\right) = \frac{1}{d+1} \frac{1}{\binom{d}{|\gamma|_0}}$$

- For the coefficients, we set a Normal prior

$$p(\boldsymbol{\beta}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}) = \prod_{\{j:\ \gamma_j=1\}} \mathcal{N}(\beta_j; 0, g\sigma^2)$$

- In our experiments we use $g = \sigma = 1$ which comes from tuning the predictive power of the model to be equal to the number of regressors, which is a usual default. Nevertheless, the theory of ALA applies to any prior parameters under minimal conditions (Rossell et al., 2021).

Laplace Approximation

- The main idea of the LA is to take take a second order expansion the log-integrand, $\log p(\mathrm{y}, \boldsymbol{\beta}_\gamma \mid \gamma)$, around the MAP estimate $\hat{\boldsymbol{\beta}}_\gamma = \mathrm{argmax}_{\boldsymbol{\beta}_\gamma}\, p(\mathrm{y}, \boldsymbol{\beta}_\gamma \mid \gamma)$:

$$\hat{p}(\mathrm{y} \mid \gamma) = p(\mathrm{y} \mid \hat{\boldsymbol{\beta}}_\gamma, \gamma) p(\hat{\boldsymbol{\beta}}_\gamma \mid \gamma) 2\pi^{p_\gamma/2} |H_\gamma(\hat{\boldsymbol{\beta}}_\gamma)|^{-\frac{1}{2}},$$

  where $H_\gamma(\hat{\boldsymbol{\beta}}_\gamma)$ is a hessian of the negative log-integrand, $-\log p(\mathrm{y}, \boldsymbol{\beta}_\gamma \mid \gamma)$, estimated at the MLE for model γ.
- Note that we use a variation of the Laplace Approximation which uses the MLE under model γ, i.e. $\hat{\boldsymbol{\beta}}_\gamma = \mathrm{argmax}_{\boldsymbol{\beta}_\gamma}\, p(\mathrm{y} \mid \boldsymbol{\beta}_\gamma, \gamma)$
- The most computationally demanding part is to compute the estimate $\hat{\boldsymbol{\beta}}_\gamma$. There is also a cost of computing the determinant of the hessian.

Approximate Laplace Approximation

- The Approximate Laplace Approximation (Rossell et al., 2021) is a computationally cheap alternative.
- It is based on taking the log-integrand around an arbitrary initial guess on the coefficients, $\boldsymbol{\beta}_{\gamma 0}$, instead of taking an approximation around the MAP.
- The resulting approximation has the following form:

$$\tilde{p}(\mathrm{y} \mid \boldsymbol{\gamma}) = p(\mathrm{y} \mid \boldsymbol{\beta}_{\gamma 0}, \boldsymbol{\gamma}) p(\tilde{\boldsymbol{\beta}}_{\gamma} \mid \boldsymbol{\gamma}) (2\pi)^{\frac{p_{\boldsymbol{\gamma}}}{2}} |H_{\gamma 0}|^{-\frac{1}{2}} \exp\left\{ \frac{1}{2} g_{\gamma 0}^{T} H_{\gamma 0}^{-1} g_{\gamma 0} \right\},$$

where $g_{\gamma 0}$ and $H_{\gamma 0}$ are the gradient and the hessian of a negative log-likelihood, $-\log p(\mathrm{y} \mid \boldsymbol{\beta}_{\gamma}, \boldsymbol{\gamma})$, and $\tilde{\boldsymbol{\beta}}_{\gamma} = \boldsymbol{\beta}_{\gamma 0} - H_{\gamma 0}^{-1} g_{\gamma 0}$.

Algorithm

Input: Markov Kernel M_t, convergence threshold ε

(n indicates that action is done for $n = 1, \ldots, N$)

$\gamma_n^0 \sim \tilde{p}_0(\cdot \mid y)$

$w_n^0 \leftarrow 1$

$W_n^0 \leftarrow \frac{1}{N}$

$t \leftarrow 1$

while $\left| \frac{\tilde{p}_t(y \mid \gamma_n^t) p(\gamma_n^t)}{\tilde{p}_{t-1}(y \mid \gamma_n^{t-1}) p(\gamma_n^{t-1})} - 1 \right| \geq \varepsilon$ **do**

  if $ESS(\Gamma_{t-1}) < ESS_{min}$ **then**

    $\hat{\gamma}_n^t \leftarrow$ resample γ_n^{t-1} with Multinomial$(W_1^{t-1}, W_2^{t-1}, \ldots, W_N^{t-1})$

    $\hat{w}_n^{t-1} \leftarrow 1$

  else

    $\hat{\gamma}_n^t \leftarrow \gamma_n^{t-1}$

    $\hat{w}_n^{t-1} \leftarrow w_n^{t-1}$

  end

  $\gamma_n^t \sim M_t(\hat{\gamma}_n^t)$

  $w_n^t \leftarrow \hat{w}_n^{t-1} \frac{\tilde{p}_t(y \mid \gamma_n^t)}{\tilde{p}_{t-1}(y \mid \gamma_n^t)}$

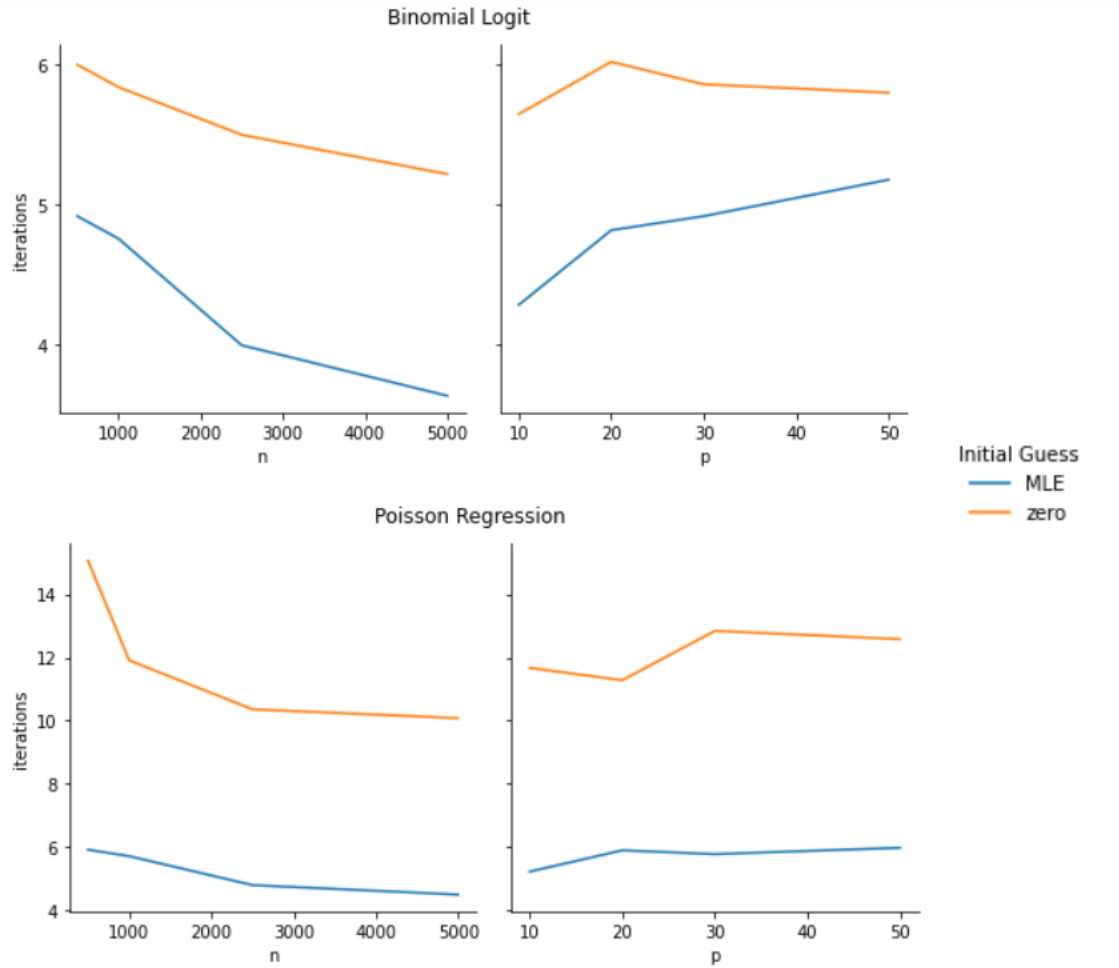
  $W_n^t \leftarrow \frac{w_n^t}{\sum_{j=1}^n w_j^t}$

  $t \leftarrow t + 1$

end

Experimental Results. Initializing at zero vs. MLE

Figure: Number of iterations of ALASMC until convergence.



Note: $\beta^* = (0_{p-3}, 0.5, 0.5, 0.5)$, $\rho = 0.5$, adjusted curvature, $N = 5000$, $B = 5000$, $\varepsilon_{grad} = 1e-5$, $\varepsilon_{loglike} = 1e-5$, 50 datasets, 1 run of SMC.

Experimental Results. Initialization and scalability/accuracy

Table: ALASMC selection rate of a true model and elapsed time by the initial coefficients.

Model	Initial guess	p	Recovers truth	Included active	Discarded spurious	Elapsed time (s).
Binomial Logit	MLE	10	0.90	2.98	6.92	4.11
		20	0.94	2.94	16.96	9.28
		30	0.76	3.00	26.70	13.83
		50	0.86	2.96	46.84	20.87
	zero	10	0.86	2.96	6.90	5.03
		20	0.96	3.00	16.96	10.85
		30	0.86	2.92	26.90	14.07
		50	0.92	2.92	46.96	20.34
Poisson	MLE	10	0.96	3.00	6.96	4.76
		20	0.96	3.00	16.96	7.93
		30	1.00	3.00	27.00	10.43
		50	1.00	3.00	47.00	15.40
	zero	10	0.24	3.00	6.24	9.55
		20	0.00	3.00	16.00	16.31
		30	0.02	3.00	26.02	46.11
		50	0.00	3.00	46.00	53.53

Note: $n = 1000$, $\beta^* = (0_{p-3}, 0.5, 0.5, 0.5)$, $\rho = 0.5$, adjusted curvature, $N = 5000$, $B = 5000$, $\varepsilon_{grad} = 1e-5$, $\varepsilon_{loglike} = 1e-5$, 50 datasets, 1 run of SMC.
Initial guess represents the way of choosing the initial coefficients: "zero" – just initialize the ALA SMC at $\beta_0 = \vec{0}$, "MLE" – initialize at the MLE of the full model.
"Recovers true" shows the percentage of times the ALASMC posterior mode matches the true model.

Experimental setups

- Difference between ALASMC and LA inclusion probabilities:

  Logistic: $\beta^* = (2, 0_7, 0.5, 1)$, unadjusted curvature.
  Poisson: $\beta^* = (0_4, 0.5, 1, 0_5)$, adjusted curvature, intercept is forced into the model.

  Note: $n = 1000$, $\beta_0 = \arg\max_{\beta_\gamma} p(\boldsymbol{y} \mid \gamma = \mathbb{1}_p, \beta_\gamma)$, $\rho = 0.5$, $N = 2000$, $B = 1000$, $\varepsilon_{grad} = 1\mathrm{e}{-8}$, $\varepsilon_{loglike} = 1\mathrm{e}{-10}$, 30 datasets, 30 runs of SMC. Intervals: 95% of the runs.

- ALASMC and LA-MCMC computational time:

  Note: $\beta^* = (0_{p-3}, 0.5, 0.5, 0.5)$, $\rho = 0.5$, adjusted curvature, $N = 5000$, $B = 5000$, $\varepsilon_{grad} = 1e-5$, $\varepsilon_{loglike} = 1e-5$, 50 datasets, 1 run of SMC.

- ALASMC vs. ALA-MCMC:

  Same as the first one, except that:

  Only 1 dataset is used, 30 runs of Monte-Carlo algorithms, $N = 5000, B = 5000$, adjsuted curvature is used for both Logistic and Poisson regression.