On the Improvements to Stan Offered by Control Variates

Yifan Zhou, Phil Clemson and Simon Maskell
University of Liverpool

March 8, 2020

1 Introduction

Control variates are a post-processing method aiming at reducing the variance[1] of estimators based on samples generated by MCMC. In this document, we focus on specific variants described in papers related to Zero Variance Control Variates (ZVCV)[1, 2]. These and other papers on control variates document impressive results. However, the algorithms are yet to be integrated into Stan. As a waypoint on the path to potential integration into Stan, this document assesses the merits of control variates in the context of models understood by the authors to be seen by the Stan developer community as appropriate for benchmarking: we considered 12 curated models[2]. Our aim is to demonstrate that adopting control variates involves minimal additional computational cost relative to running Stan (control variates manipulate the gradient of the log-likelihood as already calculated when using NUTS to generate the samples) and to quantify the variance reduction achieved when control variates are used in conjunction with Stan[3].

2 Brief Overview of Control Varirates and Implementation

This section will briefly describe the control variates we used and how we implemented them with both linear and quadratic polynomials. The details of the algorithm are described in substantially more detail in [1].

[1] Control variates reduce the sample variance, ie the ability to estimate the mean of the pdf, but do not impact on the pdf itself: it may well be that the variance of the pdf is far greater than the sample variance.

[2] While we have generated initial results across a larger portfolio of models, in response to advice kindly provided by the Stan developer community, the models considered here are those from `https://github.com/stan-dev/stat_comp_benchmarks.git`. We noticed that there are 14 ".stan" files, but we don't consider the scenarios where 'model' is empty, such as "gen_gp_data" and "sim_one_comp_mm_elim_abs".

[3] The results have been obtained using PyStan, but we believe the performance gains will be near identical when using other Stan distributions.

2.1 Overview

The fundamental idea is to create an estimator in place of each sample by adding a variable to each sample. The variable has the property that its mean is zero but is also anticorrelated with the sample such that the variance of the estimators is less than the variance of the samples[4], such that if we wish to estimate $f(x)$ then:

$$\frac{1}{N}\sum_{i=1}^{N} f\left(x_i\right) \approx \int p\left(x\right) f\left(x\right) dx = \int p\left(x\right) \left(f\left(x\right) + \alpha z\left(x\right)\right) dx \approx \frac{1}{N}\sum_{i=1}^{N} \left(f\left(x_i\right) + \alpha z\left(x_i\right)\right) \tag{1}$$

where we design $z\left(x\right)$ such that the rightmost approximation is lower variance than the leftmost approximation and where

$$\int p\left(x\right) z\left(x\right) dx = 0. \tag{2}$$

One choice (which happens to be optimal[5] for a certain choice of α if $p\left(x\right)$ is Gaussian) is to use:

$$z\left(x\right) = \frac{d}{dx} \log p\left(x\right) \tag{3}$$

for which

$$\int p\left(x\right) \frac{d}{dx} \log p\left(x\right) dx = \int p\left(x\right) \frac{\frac{d}{dx} \log p\left(x\right)}{p\left(x\right)} dx = \int \frac{d}{dx} \log p\left(x\right) dx = \left[p\left(x\right)\right]_{-\infty}^{\infty} = 0 \tag{4}$$

and this is what is referred to as the linear polynomial herein. The quadratic polynomial is explained in, for example, [1].

2.2 Implementation

We run PyStan on a model and obtain N samples. We denote the ith sample of the unconstrained state by x_i. We assume x_i is D dimensional. We emphasise that we assume we wish to estimate the mean of the unconstrained state[6] such that $f\left(x\right) = x$. We calcuate the gradient of the log-likelihood at the sampled value, $\nabla_i = z\left(x_i\right)$, using a PyStan built-in function (we acknowledge that these gradients will already have been calculated but recalculate the gradients at present for simplicity (ie ease of implementation)).

[4] The implication is that control variates do not improve correlation between samples and that the variance using control variates still scales inversely with the number of samples.

[5] The fact that it is possible to reduce the variance to zero gives rise to the name 'zero variance' control variates.

[6] We would prefer to consider constrained states (eg samples from beta or gamma distributions) and other functions of the parameters (eg variance) but, at present, have not yet understood how to extract the determinant of transformations from Stan (which would be needed). We also note that the control variates we are considering are not (yet) applicable in settings only involving constrained states.

2.3 Linear Polynomial

To use the linear polynomials, we calculate two covariance matrices as follows

$$\Sigma_{zz} = \mathbb{E}\left[zz^T\right] \tag{5}$$

$$\sigma_{zf} = \mathbb{E}\left[zx\right] \tag{6}$$

where $z = -\frac{1}{2}\nabla$.

Then we can calculate an estimator using linear control variates for each sample as follows:

$$\tilde{x}_i^l = x_i - \left(\Sigma_{zz}^{-1} \cdot \sigma_{zf}^T \cdot {z_i}^T\right)^T \tag{7}$$

where we have used the value for α that would be optimal if $p\left(x\right)$ was Gaussian.

2.4 Quadratic Polynomial

To implement the quadratic version, we construct y_i as follows.

$$y_i = \begin{bmatrix} {z_i}^T \\ {u_i}^T \\ {v_i}^T \end{bmatrix} \tag{8}$$

where $u_i = x_i \cdot z_i - \frac{1}{2}$ and v_i is a $\left[\frac{1}{2} \cdot N \cdot (N-1)\right]$ vector where

$$v_{\frac{1}{2}(2N-j)(j-1)+(k-j)} = x_k z_j + x_j z_k \tag{9}$$

for $k \in 1, ..., N-1$ and $j < k$.

We then calculate two covariance matrices as follows

$$\Sigma_{yy} = \mathbb{E}\left[yy^T\right] \tag{10}$$

$$\sigma_{yf} = \mathbb{E}\left[yx\right] \tag{11}$$

Finally, we calculate an estimate using quadratic control variates for each sample as follows:

$$\tilde{x}_i^q = x_i - \left(\Sigma_{yy}^{-1} \cdot \sigma_{yf}^T \cdot {y_i}^T\right)^T \tag{12}$$

3 Evaluation Method

We run the MCMC sampling process using PyStan for $2 \times N$ iterations (and choose $N = 2500$ throughout this document). By default, there are N warm-up iterations and we only consider the remaining N samples (in the unconstrained space). We denote the ith estimate derived without control variates to be the ith sample such that $\tilde{x}_i^{\emptyset} = x_i$. As explained in the previous section, we use control variates with linear and quadratic polynomials to obtain $\tilde{x}_i^l$ and $\tilde{x}_i^q$ respectively.

To assess performance in the context of a given model, we calculate the mean of the estimators such that for estimator, $E \in \{\emptyset, l, q\}$:

$$\bar{x}^E = \frac{1}{N} \sum_{i=1}^{N} \tilde{x}_i^E \tag{13}$$

Since the variance of this mean will be proportional to the sample variance of the constituent estimators, we quantify performance using the variance of the MCMC estimators and average across all the dimensions[7] as follows:

$$\sigma^E = \frac{1}{N \times D} \sum_{i=1}^{N} \left(\tilde{x}_i^E - \bar{x}^E\right)^T \left(\tilde{x}_i^E - \bar{x}^E\right). \tag{14}$$

We calculate the average variances of MCMC estimators to generate $\sigma^\emptyset$, σ^l and σ^q. The improvement of using control variates is then quantified as follows:

$$I^l = \frac{\sigma^l}{\sigma^\emptyset} \tag{15}$$

$$I^q = \frac{\sigma^q}{\sigma^\emptyset} \tag{16}$$

4 Experimental Results

In this section, we report the improvements of using control variates on example models that have been optimised for use by Stan. There are 12 such example models, and all of them run successfully with PyStan. Almost all the example models can be post-processed by the two versions of control variates. The exception is "irt_2pl", for which $D = 144$: 2500 samples are insufficient to estimate a non-singular instance of the 144×144 element covariance matrix, Σ_{yy}.

The improvements resulting from applying control variates are presented in Table 1. It is evident that control variates can substantially reduce the variances of estimators and that using quadratic polynomials generates further reductions in variance relative to using linear polynomials. We note that for the model "low_dim_gauss_mix_collapse", the advantage of using both versions of control variates is the least of all the models considered. We also note that control variates make a huge dfference for model "low_dim_corr_gauss" as the variances of improved estimators are approximately zero: recall that control variates are optimal in the context of sampling from a Gaussian.

The time taken for sampling, (re)-evaluating gradients and calculating the control variates are shown in Table 2. Note that the sampling time doesn't include the time taken compling the C++ model file and the time includes both the warm-up and sampling phases. As mentioned, we currently have not reused the gradient of log-likelihoods

[7] We would happily adopt alternative quantifications of performance but choose this quantification for ease of implementation and exposition.

Example	I^l: Linear (improvement)	I^q: Quadratic (improvement)	Model dimensionality
eight_schools	23.19901824	60.60209129	10
gp_pois_regr	9.877405776	17.27143894	13
low_dim_gauss_mix	1361.849242	174239.8892	5
low_dim_corr_gauss	4.59×10^{30}	6.13×10^{30}	2
low_dim_gauss_mix_collapse	1.037589741	1.157499286	5
arK	87.1879201	6954.615225	7
garch	9.165982233	87.84430094	4
gp_regr	56.67014834	162.3235115	3
sir	110.9485976	3763.777991	4
arma	42.04668143	7514.504495	4
irt_2pl	6.530973852	N/A	144
one_comp_mm_elim_abs	4.193721346	39.66728408	4

Table 1: The improvement of using control variates.

inside NUTS, so the runtime of a future implementation should not need to include the time taken to (re)-compute the gradients. Even so, we observe that calculating control variates involves minimal extra time compared to sampling. We do note that the quadratic imeplmentation should be more time-consuming than the linear version, but we found it challenging to measure this difference since both take very small amounts of time. We suspect that the specifics of our implementation (eg how caching is being used) and our use of profiling are obscuring this diffference.

5 Conclusions and Recommendations

5.1 Conclusions

In this document, we investigated the improvement of using control variates as a post-processing technique for Stan when applied to 12 example models. The experiments have shown that the control variates can reduce the (sample) variance of MCMC estimators on all models with substantial improvements in some cases. Across all models considered, the runtime of calculating the control variates is small relative to the time taken for NUTS to generate the samples.

While it would be possible to adopt alternative quantifications of performance, we do not anticipate that doing so would alter the perception that control variates offer a substantial reduction in sample variance at negligible computational cost.

Example	Stan	Compute gradients	control variates (linear)	control variates (quadratic)
eight_schools	0.2541	0.1435	0.0028	0.0066
gp_pois_regr	11.5604	0.1778	0.0031	0.0114
low_dim_gauss_mix	9.9103	0.7913	0.0030	0.0026
low_dim_corr_gauss	0.1347	0.0675	0.0018	0.0017
low_dim_gauss_mix_collapse	30.5173	0.7622	0.0027	0.0024
arK	9.5033	0.2843	0.0028	0.0037
garch	2.8273	0.2664	0.0026	0.0021
gp_regr	0.4971	0.1693	0.0024	0.0018
sir	408.344	3.4165	0.0021	0.0021
arma	1.3518	0.2247	0.0025	0.0021
irt_2pl	23.5802	0.9963	0.0199	N/A
one_comp_mm_elim_abs	101.3410	5.4746	0.0056	0.0021

Table 2: The runtimes (in seconds) of Stan and control variates on different models.

5.2 Recommendations

We acknowledge that it would be preferable to extend the functionality of our current control variates implementation to support:

- Estimation of other quantities than the unconstrained samples;
- Re-use of the gradients calculated during Stan's use of NUTS;
- Future extension with more advanced versions of control variates.

It is recommended that, on the basis of this document, the Stan developer community endorse the future integration of control variates in Stan. Once that endorsement has been obtained, development effort (by both the authors and the Stan developer community) should be expended on designing a new component of Stan that extends our current implementaiton and post-processes the samples that Stan generates. Such a new compoent woudl enable all Stan users to have the option to capitalise on the reduction in sample variance offered by control variates.

Research is needed to support the use of control variates in settings only involving constrained samples.

Acknowledgements

The authors gratefully acknowledge numerous helpful discussions with Antonietta Mira related to control variates. The authors also gratefully acknowledge the support, advice and guidance provided by the Stan developer community.

References

[1] A. Mira, R. Solgi, and D. Imparato, "Zero variance markov chain monte carlo for bayesian estimators," *Statistics and Computing*, vol. 23, no. 5, pp. 653–662, 2013.

[2] T. Papamarkou, A. Mira, M. Girolami, *et al.*, "Zero variance differential geometric markov chain monte carlo algorithms," *Bayesian Analysis*, vol. 9, no. 1, pp. 97–128, 2014.