Measure-Valued Derivatives for Approximate Bayesian Inference

Mihaela Rosca* **Michael Figurnov*** **Shakir Mohamed** **Andriy Mnih**
DeepMind
{mihaelacr,mfigurnov,shakir,amnih}@google.com

1 Stochastic gradient estimation

The objective of many machine learning problems is to maximize the expected value of a cost function $f(\mathbf{x}; \boldsymbol{\phi})$ under a learned distribution $p(\mathbf{x}; \boldsymbol{\theta})$. Learning the distributional parameters $\boldsymbol{\theta}$ via gradient based methods requires estimating the stochastic gradient:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x};\boldsymbol{\theta})} \left[f(\mathbf{x}; \boldsymbol{\phi}) \right]. \tag{1}$$

Computing the above gradient is difficult, since for many problems of interest the expectation is intractable and the high dimensionality of $\mathbf{x}$ makes estimating integrals numerically ineffective. This motivates the search for unbiased, low variance and computationally efficient Monte Carlo stochastic gradient estimators. Currently, the most commonly used unbiased estimators in machine learning are the score function (REINFORCE) [1, 2] and the pathwise (a.k.a. reparameterization) [3, 4, 5, 6] estimators. The pathwise estimator tends to have low variance, but requires the cost function to be differentiable and the distribution $p(\mathbf{x}; \boldsymbol{\theta})$ to be reparametrizable, while the score function estimator is more widely applicable, but usually exhibits high variance. This motivates our search for a generally applicable estimator with low variance, and thus we will focus on a third class of estimators, the *measure-valued derivatives*.

2 Measure-valued derivatives

The measure-valued derivative estimator (also known as the weak derivative method) [7, 8] defines a general, unbiased and low variance gradient estimator by exploiting a particular decomposition of the gradient of a probability density with respect to its parameters. For a scalar parameter $\boldsymbol{\theta}_i$ the gradient $\nabla_{\boldsymbol{\theta}_i} p(\mathbf{x}; \boldsymbol{\theta})$ decomposes into a difference of two densities multiplied by a constant [9]:

$$\nabla_{\boldsymbol{\theta}_i} p(\mathbf{x}; \boldsymbol{\theta}) = c_{\boldsymbol{\theta}_i} p_i^+(\mathbf{x}; \boldsymbol{\theta}) - c_{\boldsymbol{\theta}_i} p_i^-(\mathbf{x}; \boldsymbol{\theta}). \tag{2}$$

The decomposition of this form always exists, though it is not unique, and can be obtained using Hahn-Jordan decomposition of a signed measure into two measures that have complementary support [10]. This property can be used to define a gradient estimator for each parameter $\boldsymbol{\theta}_i$:

$$\nabla_{\boldsymbol{\theta}_i} \mathbb{E}_{p(\mathbf{x};\boldsymbol{\theta})} \left[f(\mathbf{x}) \right] = \int \nabla_{\boldsymbol{\theta}_i} p(\mathbf{x}; \boldsymbol{\theta}) f(\mathbf{x}) d\mathbf{x} = c_{\boldsymbol{\theta}_i} \left(\mathbb{E}_{p_i^+(\mathbf{x};\boldsymbol{\theta})} \left[f(\mathbf{x}) \right] - \mathbb{E}_{p_i^-(\mathbf{x};\boldsymbol{\theta})} \left[f(\mathbf{x}) \right] \right) \tag{3}$$

where the expectations in Equation (3) can be estimated via Monte Carlo:

$$\frac{c_{\boldsymbol{\theta}_i}}{N} \left(\sum_{n=1}^{N} f(\dot{\mathbf{x}}^{(n)}) - \sum_{n=1}^{N} f(\ddot{\mathbf{x}}^{(n)}) \right); \quad \dot{\mathbf{x}}^{(n)} \sim p_i^+(\mathbf{x}; \boldsymbol{\theta}), \quad \ddot{\mathbf{x}}^{(n)} \sim p_i^-(\mathbf{x}; \boldsymbol{\theta}) \tag{4}$$

Table 2 shows a set of measure-valued derivative triples (c_θ, p^+, p^-) for common univariate distributions.

The main assumptions made by the measure-valued estimator is that the integrals of p^+, p^- against the desired cost functions converge. Since the measure-valued estimator makes no assumptions about the differentiability of the cost, it is more widely applicable than the pathwise estimator.

*Equal contribution.

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

2.1 Fully factorized distribution

Let $p(\mathbf{x};\boldsymbol{\theta}) = \prod_{i=1}^{D} p(\mathbf{x}_i|\boldsymbol{\psi}_i)$, where D denotes the dimensionality of $\mathbf{x}$, $\boldsymbol{\psi}_i$ are the parameters affecting $\mathbf{x}_i$, with $\{\boldsymbol{\psi}_i\}$ forming a partition of $\boldsymbol{\theta}$. Given a scalar parameter $\boldsymbol{\theta}_i \in \boldsymbol{\psi}_i$ we can rewrite the measure-valued derivative $\nabla_{\boldsymbol{\theta}_i} p(\mathbf{x};\boldsymbol{\theta})$ as follows:

$$\nabla_{\boldsymbol{\theta}_i} \mathbb{E}_{p(\mathbf{x};\boldsymbol{\theta})}\left[f(\mathbf{x})\right] = \int \nabla_{\boldsymbol{\theta}_i} \prod_{j=1}^{D} p(x_j|\boldsymbol{\psi}_j) f(\mathbf{x}) d\mathbf{x} = \int \prod_{\substack{j=1 \\ j\neq i}}^{D} p(x_j|\boldsymbol{\psi}_j) \nabla_{\boldsymbol{\theta}_i} p(\mathbf{x}_i|\boldsymbol{\psi}_i) f(\mathbf{x}) d\mathbf{x}$$

$$= \mathbb{E}_{p(\mathbf{x}_{j\neq i}|\boldsymbol{\psi}_{j\neq i})}\left[c_{\boldsymbol{\theta}_i}\left(\mathbb{E}_{p_i^+(\mathbf{x};\boldsymbol{\psi}_i)}\left[f(\mathbf{x})\right] - \mathbb{E}_{p_i^-(\mathbf{x};\boldsymbol{\psi}_i)}\left[f(\mathbf{x})\right]\right)\right] \tag{5}$$

Thus computing the measure-valued derivative of a fully factorized multivariate distribution only requires knowing the measure-valued decomposition for the univariate factors. For example, for a diagonal multivariate Gaussian we have $\boldsymbol{\psi}_i = \{\mu_i, \sigma_i\}$. Then, we can use Equation (5) and Table 2 to compute $\nabla_{\mu_i} \mathbb{E}_{p_i(\mathbf{x}_i;\boldsymbol{\psi}_i)}\left[f(\mathbf{x})\right]$ and $\nabla_{\sigma_i} \mathbb{E}_{p(\mathbf{x}_i;\boldsymbol{\psi}_i)}\left[f(\mathbf{x})\right]$ for every dimension i.

Equation (5) reveals the downside of measure-valued gradient estimation: for each dimension of $\boldsymbol{\theta}$, we need to evaluate the cost function twice, leading to total of $2N|\boldsymbol{\theta}|$ cost function evaluations, as opposed to N cost function evaluations for the score function and pathwise estimators.

2.2 Variance reduction via coupling

The variance of the measure-valued derivative estimator is:

$$\mathbb{V}_{p(\mathbf{x};\boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}} f(\mathbf{x})] = \mathbb{V}_{p^+(\mathbf{x};\boldsymbol{\theta})}[f(\mathbf{x})] + \mathbb{V}_{p^-(\mathbf{x};\boldsymbol{\theta})}[f(\mathbf{x})] - 2\mathrm{Cov}_{p^+(\mathbf{x}';\boldsymbol{\theta})p^-(\mathbf{x};\boldsymbol{\theta})}[f(\mathbf{x}'), f(\mathbf{x})]. \tag{6}$$

We can see that 'coupling' the random variables so that $f(\mathbf{x}')$ and $f(\mathbf{x})$ are positively correlated, decreases the variance of the gradient estimator. The most common coupling scheme involves sharing the underlying source of randomness by sampling the variables $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ using common random numbers [7].

Example of using coupling for measure-valued gradient estimation and plots showcasing its effect on gradient variance can be found in Appendix A.

3 Experiments

Bayesian logistic regression defines a probabilistic model for a target $y \in \{0, 1\}$ given features $\mathbf{x} \in \mathbb{R}^D$ using a set of parameters $\mathbf{w}$. Using a Gaussian prior on the parameters and a Bernoulli likelihood, the probabilistic model is:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I}); \qquad p(y_i|\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^\top \mathbf{w})^{y_i}(1 - \sigma(\mathbf{x}_i^\top \mathbf{w}))^{1-y_i}, \tag{7}$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$ is the logistic function. We maximize the variational lower bound [11, 12] to learn the parameters of the posterior distribution $q(\mathbf{w};\boldsymbol{\theta}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$\sum_{i=1}^{I} \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[y_i \log \sigma(\mathbf{x}_i^\top \mathbf{w}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}_i^\top \mathbf{w}))\right] - \mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| p(\mathbf{w})\right], \tag{8}$$

with $i = 1, \ldots, I$ indexing the training points. The objective function we use is a Monte Carlo estimator of eqn. (8):

$$\frac{I}{B}\sum_{i=1}^{B}\frac{1}{N}\sum_{n=1}^{N}\left[y_{d_i} \log \sigma(\mathbf{x}_{d_i}^\top \hat{\mathbf{w}}^{(n)}) + (1 - y_{d_i}) \log(1 - \sigma(\mathbf{x}_{d_i}^\top \hat{\mathbf{w}}^{(n)}))\right] - \mathrm{KL}\left[\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})\right];$$

$$\hat{\mathbf{w}}^{(n)} \sim \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}),\ d_i \sim \{1, \ldots, I\}$$

where B is the data batch size, I is the size of the full data set, N is the number of samples taken from the posterior distribution to evaluate the Monte Carlo gradient, and the posterior covariance is a diagonal matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\mathbf{s})$. We use the UCI Women's Breast Cancer dataset [13], which has $I = 569$ data points and $D = 31$ features. We compute the Kullback-Leibler (KL) divergence between the variational posterior q and the prior distribution p analytically and always use coupling

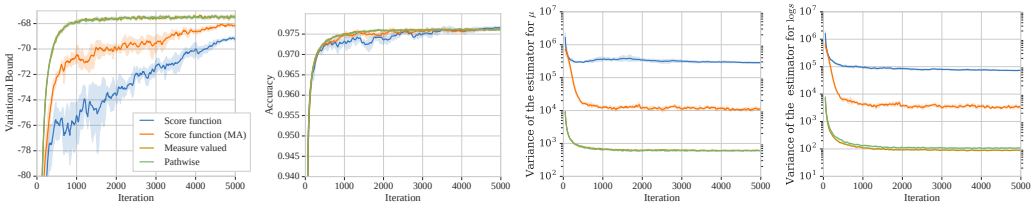

Figure 1: Comparison between the score function, measure-valued and pathwise estimators with initial learning rate 10^{-3}, $B = 32, N = 50$. Gradient estimator variance for the mean $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_\mu f(\mathbf{w})]$ and log-standard deviation $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\log s} f(\mathbf{w})]$ is averaged over parameter dimensions.

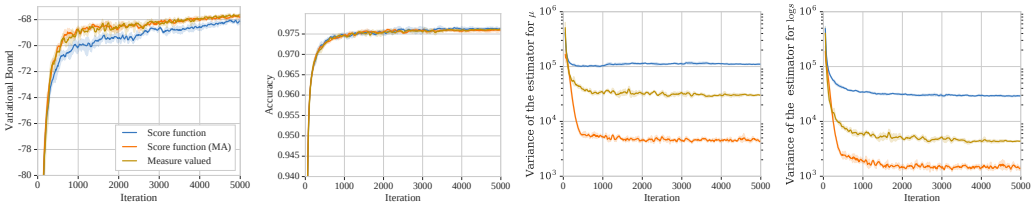

Figure 2: Comparison between the score function estimator and measure-valued estimators with the same number of function evaluations, with initial learning rate 10^{-3}, $B = 32$. For the measure-valued estimator $N = 1$, while for the score function estimator $N = 4D = 124$. Gradient estimator variance for the mean $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_\mu f(\mathbf{w})]$ and log-standard deviation $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\log s} f(\mathbf{w})]$, averaged over parameter dimensions.

when estimating the measure-valued gradients. We train the models with stochastic gradient ascent, using the cosine learning rate decay schedule [14]. For evaluation, we use the entire dataset and 1000 posterior samples.

Figure 1 shows that the measure-valued estimator achieves the same variance as the pathwise estimator, without making use of the gradient of the cost function. The measure-valued estimator outperforms the score function estimator, even when that estimator uses a moving average baseline for variance reduction. This does come at a cost: the measure-valued estimator uses $2|\boldsymbol{\theta}| = 4D\times$ more function evaluations. To control for this difference in cost, we also perform a comparison in which both methods make the same number of function evaluations (using $4D\times$ more samples for the score function estimator) and show the results in Figure 2. In this setting, the vanilla score function still exhibits higher variance than the measure-valued estimator, but the score function estimator with a moving average baseline, which is a simple and efficient variance reduction method, performs best.

Non-differentiable cost function

To assess the variance properties of the measure-valued estimator in a non-differentiable cost function setting, we compare it with the score function estimator using the following modified objective:

$$\frac{1}{I}\left(\sum_{i=1}^{I}\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[1-2|y_i-\lfloor\sigma(\mathbf{x}_i^\top\mathbf{w})\rceil|\right]-\mathrm{KL}\left[\mathcal{N}(\mathbf{w}|\boldsymbol{\mu},\boldsymbol{\Sigma})\|\mathcal{N}(\mathbf{w}|\mathbf{0},\mathbf{I})\right]\right) \tag{9}$$

where $\lfloor x\rceil$ denotes the nearest integer to x.

Figure 3 shows that in this setting, if we do not control for the difference in the computation cost of the estimators, the measure-valued estimator outperforms the score function estimator and exhibits lower variance. However, when we do control for the number of cost function evaluations, the score function with a moving average baseline exhibits lower variance than the measure-valued estimator, as shown in Figure 4.

Benchmarks

We complement the performance comparison of the score function estimator and the measure-valued estimator by comparing the average time to compute a gradient update for Bayesian Logistic Regression with the different gradient estimators in Table 1.

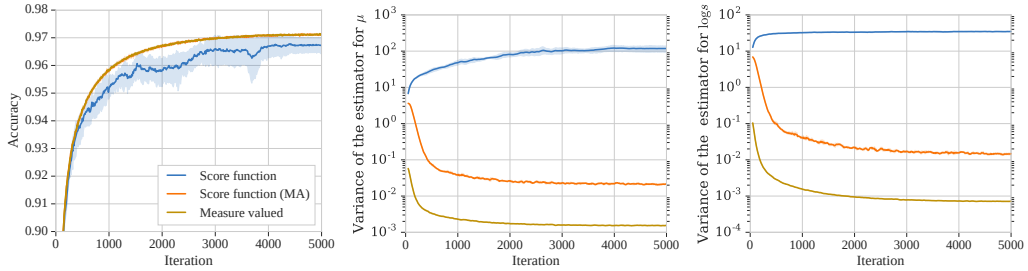

Figure 3: Comparison between the score function and measure-valued estimators for a non-differentiable loss function. The start learning rate is 10^{-1}, $B = 32$, $N = 50$. Gradient estimator variance for the mean $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\mu} f(\mathbf{w})]$ and log-standard deviation $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\log s} f(\mathbf{w})]$ is averaged over parameter dimensions.

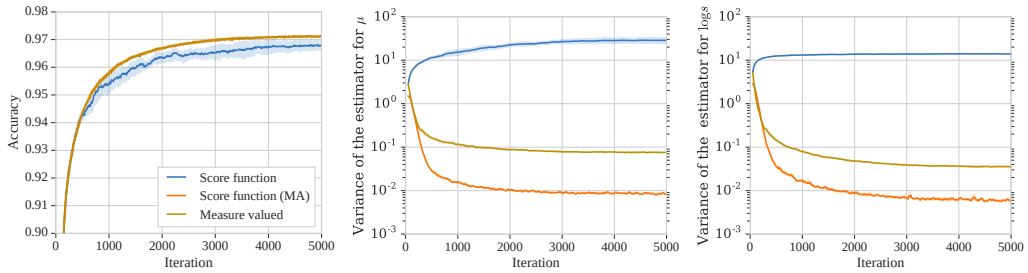

Figure 4: Comparison between the score function estimator and measure-valued for a non-differentiable loss function and the same number of function evaluations controlled between different estimators. For the measure-valued estimator $N = 1$, while for score function estimator, $N = 4D = 124$. The start learning rate is 10^{-1}, $B = 32$. Gradient estimator variance of the mean $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\mu} f(\mathbf{w})]$ and log-standard deviation $\mathbb{V}_{q(\mathbf{w}|\boldsymbol{\mu},\mathbf{s})}[\nabla_{\log s} f(\mathbf{w})]$, averaged over parameter dimensions.

4 Discussion

We highlighted the principles behind measure-valued gradient estimation and investigated its applicability in Bayesian approximate inference. Our experiments show that measure-valued gradients exhibit low variance, but at a high computational cost. The general applicability, low variance and robustness of measure-valued gradient estimation motivates us to further search for its compelling applications in machine learning, as well as methods to reduce its high computational cost.

Acknowledgments

We would like to thank Danilo Rezende and Yee Whye Teh for their helpful comments on a draft of this paper.

Estimator	**Number of samples N**	**Update time (ms)**
Score function (MA)	124	0.76 ± 0.03
Measure-valued	1	0.87 ± 0.05
Pathwise	1	0.4 ± 0.03

Table 1: Benchmark results. The results are computed on a Nvidia V100 GPU. $B = 32$.

References

[1] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement Learning*, pages 5–32, 1992.

[2] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

[3] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[5] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pages 2944–2952, 2015.

[6] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[7] Georg Ch Pflug. *Optimization of stochastic models: The interface between simulation and optimization*, volume 373. Springer Science & Business Media, 1996.

[8] Bernd Heidergott and Felisa Vázquez-Abad. Measure-valued differentiation for stochastic processes: the finite horizon case. In *EURANDOM report 2000-033*, 2000.

[9] G Ch Pflug. Sampling derivatives of probabilities. *Computing*, 42(4):315–328, 1989.

[10] Patrick Billingsley. *Probability and Measure*. John Wiley & Sons, 2008.

[11] Tommi Jaakkola and Michael Jordan. A variational approach to Bayesian logistic regression models and their extensions. In *Artificial Intelligence and Statistics*, 1997.

[12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[14] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[15] Bernd Heidergott, Felisa J Vázquez-Abad, and Warren Volk-Makarewicz. Sensitivity estimation for Gaussian systems. *European Journal of Operational Research*, 187(1):193–207, 2008.

Supplementary material

A Variance reduction via coupling

Example (Weibull-Weibull Coupling). Consider Gaussian measures $\mathcal{N}(x|\mu, \sigma^2)$ in the setting of equation (1) and the task of computing the gradient with respect to the mean μ. The measure-valued gradient, using Table 2, is given by the triple $(1/\sigma\sqrt{2\pi}; \theta + \sigma\mathcal{W}(2, 0.5), \theta - \sigma\mathcal{W}(2, 0.5))$ where $\mathcal{W}$ is the Weibull distribution; see Table 3 for the distribution density. We can apply coupling by reusing the Weibull samples when computing the positive and negative terms of the estimator. Figure 5 shows that using Weibull-Weibull coupling does not always reduce the variance of the measure-valued estimator; depending on the cost function, coupling can increase variance. □

Example (Maxwell-Gaussian Coupling). Consider Gaussian measures $\mathcal{N}(x|\mu, \sigma^2)$ in the setting of equation (1) and the task of computing the gradient with respect to the standard deviation σ. The measure-valued gradient, using Table 2, is given by the triple $\left(\frac{1}{\sigma}; \mathcal{M}(x|\mu, \sigma^2), \mathcal{N}(x|\mu, \sigma^2)\right)$, where $\mathcal{M}$ is the double-sided Maxwell distribution with location μ and scale σ^2, and $\mathcal{N}$ is the Gaussian distribution with mean μ and scale σ; see Table 3 for the densities for these distributions. We can couple the Gaussian and the Maxwell distribution by exploiting their corresponding sampling paths with a common random number: if we can generate samples $\dot{\varepsilon} \sim \mathcal{M}(0, 1)$, then, by first sampling from the Uniform distribution $\dot{u} \sim \mathcal{U}[0, 1]$ and reusing the Maxwell samples, we can generate $\mathcal{N}(0, 1)$ distributed samples $\ddot{\varepsilon}$ via $\ddot{\varepsilon} = \dot{\varepsilon}\dot{u}$. Then, we can perform a location-scale transform to obtain the desired Maxwell and Normal samples: $\dot{x} = \mu + \sigma\dot{\varepsilon}, \quad \ddot{x} = \mu + \sigma\ddot{\varepsilon}$. The distributions are coupled because they use the same underlying Maxwell-distributed variates [15]. The variance reduction effect of Maxwell-Gaussian coupling for the measure-valued gradient estimator can be seen in Figure 6. □

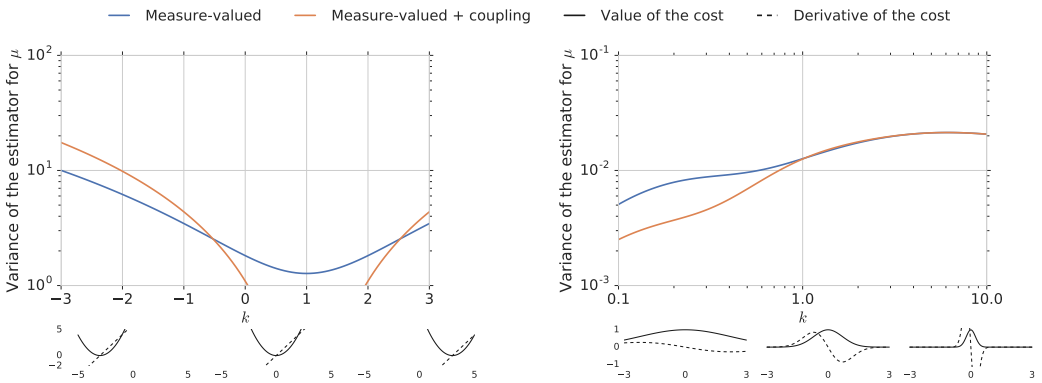

Figure 5: The effect of Weibull-Weibull coupling on the variance of the stochastic estimates of the measure-valued estimator for $\nabla_\mu \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[f(x; k)]$ for $\mu = \sigma = 1$ as a function of k. Left: $f(x; k) = (x - k)^2$; right: $f(x; k) = \exp(-kx^2)$. The graphs in the bottom row show the function (solid) and its gradient (dashed). The estimator variance for each cost function is computed using numerical integration.

B Measure-valued triples for common distributions

Table 2 shows a set of measure-valued derivative triples for common univariate distributions, and table 3 provides the definition of the distributions that were mentioned within the paper.

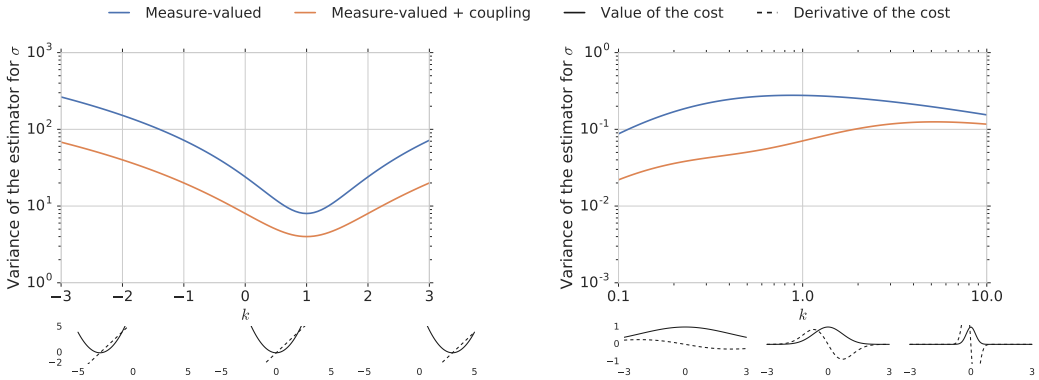

Figure 6: The effect of Maxwell-Gaussian coupling on the variance of the stochastic estimates of the measure-valued estimator for $\nabla_\sigma \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}[f(x;k)]$ for $\mu = \sigma = 1$ as a function of k. Left: $f(x;k) = (x-k)^2$; right: $f(x;k) = \exp(-kx^2)$. The graphs in the bottom row show the function (solid) and its gradient (dashed). The estimator variance for each cost function is computed using numerical integration.

Table 2: Measure-valued derivative triples (c_θ, p^+, p^-) of common distributions; we use $\mathcal{N}$ for the Gaussian density, $\mathcal{W}$ for the Weibull, $\mathcal{G}$ for the Gamma, $\mathcal{E}$ for the exponential, $\mathcal{E}r$ for the Erlang, $\mathcal{M}$ the double-sided Maxwell, and $\mathcal{P}$ for the Poisson. See Table 3 for the forms of these distributions.

Distribution $p_\theta(x)$	Constant c_θ	Positive part $p^+(x)$	Negative part $p^-(x)$
Bernoulli(θ)	1	δ_1	δ_0
Poisson(θ)	1	$\mathcal{P}(\theta) + 1$	$\mathcal{P}(\theta)$
Normal(θ, σ)	$1/\sigma\sqrt{2\pi}$	$\theta + \sigma\mathcal{W}(2, 0.5)$	$\theta - \sigma\mathcal{W}(2, 0.5)$
Normal(μ, θ^2)	$1/\theta$	$\mathcal{M}(\mu, \theta^2)$	$\mathcal{N}(\mu, \theta^2)$
Exponential(θ)	$1/\theta$	$\mathcal{E}(\theta)$	$\theta^{-1}\mathcal{E}r(2)$
Gamma(a, θ)	a/θ	$\mathcal{G}(a, \theta)$	$\mathcal{G}(a+1, \theta)$
Weibull(α, θ)	$1/\theta$	$\mathcal{W}(\alpha, \theta)$	$\mathcal{G}(2, \theta)^{1/\alpha}$

Table 3: List of distributions and their densities.

Name	**Domain**	**Notation**	**Probability Density/Mass Function**
Gaussian	$\mathbb{R}$	$\mathcal{N}(x\|\mu, \sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$
Double-sided Maxwell	$\mathbb{R}$	$\mathcal{M}(x\|\mu, \sigma^2)$	$\frac{1}{\sigma^3\sqrt{2\pi}}(x-\mu)^2 \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Weibull	$\mathbb{R}^+$	$\mathcal{W}(x\|\alpha, \beta, \mu)$	$\alpha\beta(x-\mu)^{\alpha-1}\exp(-\beta(x-\mu)^\alpha)\mathbb{1}_{\{x\geq 0\}}$
Poisson	$\mathbb{Z}$	$\mathcal{P}(x\|\theta)$	$\exp(-\theta)\sum_{j=0}^{\infty}\frac{\theta^j}{j!}\delta_j$
Erlang	$\mathbb{R}^+$	$\mathcal{E}r(x\|\theta, \lambda)$	$\frac{\lambda^\theta x^{\theta-1}\exp(-\lambda x)}{(\theta-1)!}$
Gamma	$\mathbb{R}^+$	$\mathcal{G}(x\|\alpha, \beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\exp(-x\beta)\mathbb{1}_{\{x\geq 0\}}$
Exponential	$\mathbb{R}^+$	$\mathcal{E}(x\|\lambda)$	$\mathcal{G}(1, \lambda)$