Advanced Gradient Estimation through Discreteness

Zijing Ou*
School of Computer Science and Engineering
Sun Yat-sen University

Contents

*This note primarily relies on the report from Lin Zheng, with very minor modifications based on my understanding.

Technical report.

1 Gumbel Tricks

1.1 Basic definitions and properties

The distribution function of a Gumbel random variable X is

$$F(x;\mu,\beta) = e^{-e^{-(x-\mu)/\beta}} \tag{1}$$

In this report, we shall assume $\beta = 1$ to simplify our notation; however, in many cases adjusting β will also be useful. A simple manipulation shows that the distribution functions of Gumbel random variables follow the decomposition rule:

$$F(x;\mu_1)F(x;\mu_2) = e^{-e^{-(x-\mu_1)}-e^{-(x-\mu_2)}} = e^{-e^{-x+\log(e^{\mu_1}+e^{\mu_2})}} = F(x;\log(e^{\mu_1}+e^{\mu_2})) \tag{2}$$

A key observation is that we can efficiently sample from the Gumbel distribution by making use of the inverse transform sampling technique: by drawing a sample $u \sim \text{Uniform}(0,1)$, we can obtain a sample $x = F^{-1}(u) = -\log(-\log(u)) + u$.

In addition, based on its distribution function, it is easy to derive its density function:

$$f(x) = e^{-(x-\mu)}e^{-e^{-(x-\mu)}}, x \in \mathbb{R}. \tag{3}$$

The distribution function G of a GUmbel distribution $TG(\mu, z)$ truncated at point z can be represented as

$$G(x;\mu,z) = \mathbb{P}(X \le x | X \le z) = \frac{F(\min\{z,x\};\mu)}{F(z;\mu)} \tag{4}$$

For any $x \in (-\infty, z]$, we can still sample by finding the inverse of $G(\cdot;\mu,z)$. Having a sample $u \sim \text{Uniform}(0,1)$, we can write as follows:

$$\begin{aligned}
x &= G^{-1}(u) \\
\frac{F(\min\{z,x\};\mu)}{F(z;\mu)} &= u \\
\log F(x;\mu) - \log F(z;\mu) &= \log u \\
-e^{-x+\mu} + e^{-z+\mu} &= \log u \\
e^{\mu}e^{-x} &= -\log u + e^{-z+\mu} \\
\mu - x &= \log(-\log u + e^{-z+\mu}) \\
x &= -\log(-\log u + e^{-z+\mu}) + \mu \\
x &= -\log\left(-\frac{\log u}{e^u} + e^{-z}\right)
\end{aligned} \tag{5}$$

1.2 Different of two independent Gumbels

We now show that the difference of two independent Gumbel random variables $X_1 \sim \text{Gumbel}(a_1), X_2 \sim \text{Gumbel}(a_2)$ follows a Logistic distribution.

Proof.

$$\begin{aligned}
\mathbb{P}(X_1 - X_2 \le x) &= \int_{\mathbb{R}} F(z+x;a_1)f(z;a_2)dz \\
&= \int_{\mathbb{R}} e^{-z+a_2}e^{-e^{-z+a_2}-e^{-z-x+a_1}}dz \\
&\overset{t=e^{-z}}{=} \int_0^{\infty} e^{a_2}e^{-(e^{a_2}+e^{a_1-x})t}dt \\
&= -\frac{e^{a_2}}{e^{a_2}+e^{a_1-x}}e^{-(e^{a_2}+e^{a_1-x})}\Big|_0^{\infty} \\
&= \frac{1}{1+e^{-(x-(a_1-a_2))}},
\end{aligned} \tag{6}$$

which is the CDF of a Logistic distribution with location $a_1 - a_2$. □

1.3 The use of sampling from discrete distributions

This is the most important property for reparameterizing discrete random variables. Consider a discrete distribution with n outcomes, whose probabilities of each outcome are proportional to $\exp\phi_i$, respectively. Thus, the ture probabilities can be computed as

$$p_i = \frac{\exp\phi_i}{\sum_{j=1}^{n}\exp\phi_j} \tag{7}$$

The use of Gumbel Max trick provides an efficient way to draw samples form such discrete distributions, which proceeds as follows: first, we sample independently $g_i \sim \mathrm{Gumbel}(\phi_i)$ for each i. Then, we select the index k which locates the maxmum value of these samples, that is, $k = \underset{i}{\operatorname{argmax}}\, g_i$. k is then a sample from the discrete distribution.

Proof. Suppose at k-th position g_k attains the maximum among these n values. Then we have the following:

$$\mathbb{P}(G_k \text{ is largest}|G_k = g_k) = \mathbb{P}(G_i \le g_k \text{ for all } i \ne k) = \prod_{i=1, i\ne k}^{n} F(g_k;\phi_i). \tag{8}$$

Since the value of G_k can be taken over $\mathbb{R}$, we obtain the marginal distribution as

$$\begin{aligned}
\mathbb{P}(G_k \text{ is largest}) &= \int_{\mathbb{R}} \mathbb{P}(G_k = g_k)\mathbb{P}(G_k \text{ is largest}|G_k = g_k)dg_k \\
&= \int_{\mathbb{R}} f(x;\phi_k) \prod_{i=1, i\ne k}^{n} F(x;\phi_i)dx \\
&= \int_{\mathbb{R}} e^{-(x-\phi_k)} \prod_{i=1}^{n} F(x;\phi_i)dx \\
&= \int_{\mathbb{R}} e^{-(x-\phi_k)} F\left(x; \log\left(\sum_{i=1}^{n} e^{\phi_i}\right)\right) \\
&= \int_{\mathbb{R}} e^{-(x-\phi_k)} e^{-e^{-\left(x-\left(\log\sum_{i=1}^{n} e^{\phi_i}\right)\right)}} dx \\
&\overset{z=e^{-x}}{=} \int_0^{\infty} e^{\phi_k} e^{-z\left(\sum_{i=1}^{n} e^{\phi_i}\right)} dz \\
&= -\frac{e^{\phi_k}}{\sum_{i=1}^{n} e^{\phi_i}} e^{-z\left(\sum_{i=1}^{n} e^{\phi_i}\right)}\Big|_{z=0}^{\infty} \\
&= \frac{e^{\phi_k}}{\sum_{i=1}^{n} e^{\phi_i}},
\end{aligned}$$

which is exactly the probability mass of the k-th category. The above equation further justifies that given a category distribution with logit α_i (*i.e.,* $p_i \propto \alpha_i$), and set $k = \underset{i}{\operatorname{argmax}}\, \mathrm{Gumbel}(\log\alpha_i)$, then we have $k \sim \alpha_i / \sum_j \alpha_j$, which is more practical. Note that $\mathrm{Gumbel}(\log\alpha)$ can be reparameterized as $\log\alpha + \mathrm{Gumbel}(0)$. □

For the binary case, note that

$$\mathbb{P}(G_1 \ge G_0) = \mathbb{P}(G_1 - G_0 \ge 0). \tag{9}$$

Hence we only need to sample a sample y from $\mathrm{Logistic}(\phi_1 - \phi_0)$, which is reparameterized as $y = \log u - \log(1-u) + \phi_1 - \phi_0$ with $u \sim \mathrm{Uniform}(0,1)$ and then obtain a binary sample $B = H(y)$[2].

In this way, we can obtain samples from discrete distributions by taking $\arg\max$ of n independent Gumbel samples. To further obtain a fully differentiable version of this sampling routine, we can replace the $\arg\max$ operation with $\mathrm{Softmax}$ and affiliate a temperature to control its slope.

[2] $H(z) = 1$ if $z \ge 0$ and $H(z) = 0$ if $z \le 1$.

1.4 Generalized-Gumbel for truncated random variables

The work of [1] is a generalization of Gumebl-Softmax trick, which serves as continuous relaxation of discrete random variables. This paper addresses the inapplicability of such method for those discrete variables with support on *countably* many points. The main idea is to *truncate* such kind of discrete variables so that the truncated variables have masses only on *finitely many* points and then use the Gumbel-Softmax trick to reparameterize.

Definition 1.1. A truncated discrete random variable $Z \sim TD(\lambda, n)$ of a non-negative discrete random variable $X \sim D(\lambda)$ (such as Poisson, Geometric, etc.) is a discrete randm variable whose probability mass function is

$$\mathcal{P}(Z = k) = \begin{cases} \mathbb{P}(X = k), & k \leq n-2 \\ \mathbb{P}(X \geq n-1), & k = n-1 \end{cases}$$

From this definition, it is clear that some discrete random variable can be approximated as a discrete random variable with finite support by truncation. Intuitively, the hyper-parameter n is considered to be a truncation level, which enables the trade-off between computational efficiency and fidelity of the approximation to the true distribution of X. For large n, the probability $\mathbb{P}(X \geq n-1)$ will be sufficiently small so that the approximation will be better, at the cost of increasing computation.

Additionally, an important consequence of such generalization is that Gumbel-Softmax reparameterization is now compatible with more general discrete random variables (although approximately). As we have already known how to apply Gumbel-Softmax reparameterization for categorical discrete random variables, we are now able to reparameterize the general discrete random variable X through the surrogate Z.

2 Advanced Variance Reduction for Gradient Estimation

2.1 Gradient Estimation for Discrete Latent Variables

Stochastic neural network is a natural way to model our observations and learn useful representations, which involves some random variable z. For many problems we need to maximize an expected reward (or equivalently minimizing a loss) of certain quantities $f(z)$ of interest over the distribution $p_\theta(z)$ of z with parameters θ, denoted by $\mathbb{E}_p(f(z))$. To optimize parameters within the network and efficiently back-propagate the gradients throught z, we are required to compute $\nabla_\theta \mathbb{E}_p(f(z))$. In many cases, we can estimate the gradient by expressing it as another expectation, that is, $\mathbb{E}_p(g(z)) = \nabla_\theta \mathbb{E}_p(f(z))$. Then, we simply take an estimate $\hat{g}(z)$ so that on average it is equal to $\nabla_\theta \mathbb{E}_p(f(z))$. For example, in REINFORCE algorithms [2], they write the expectation as

$$\begin{aligned} \nabla_\theta \mathbb{E}_p(f(z)) &= \nabla_\theta \sum_z p_\theta(z) f(z) = \sum z(\nabla_\theta p_\theta(z)) f(z) \\ &= \sum_z p_\theta(z) f(z) \nabla_\theta \log p_\theta(z) = \mathbb{E}_p(f(z) \nabla_\theta \log p_\theta(z)) := \mathbb{E}_p(g(z)). \end{aligned}$$

There have been many advances to deal with *continuous random variables*, including reparameterization trick [3, 4, 5] and its generalized version [6]. However, for discrete random variables, these methods cannot apply since they require that the sensity of the distribution be differentiable with respect to the variable. Although the REINFORCE algorithm is applicable, it suffers from high variance and is limited for practical use. It is still a challenging problem for the case of discrete random variables. Recently, there are many algorithms addressing this issue and propose several variance reduction methods, which will be detailed below.

2.2 REBAR

REBAR [7] is a recent technique combing REINFORCE estimators [2] and Gumbel-Softmax trick [8, 9] to reduce variance of the discrete gradient estimators. More concretely, it exploits the *conditioned* Gumbel-Softmax reparameterization as the control variate for the base REINFORCE estimator. Denoting our discrete random variable and n Gumbel random variables by B with n outcomes and $Z_1, Z_2, \ldots, Z_n$, respectively, we formulate our problem as evaluating $\frac{\partial}{\partial \theta} \mathbb{E}_{B \sim p(B|\theta)}[f(B)]$; in

particular, we seek an estimator $\hat{g}$ so that

$$\mathbb{E}[\hat{g}] = \frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}[f(B)]. \tag{10}$$

Control variates are a popular way to reduce *as long as* the control variate is positively correlated with $f(B)$. Following the work of [10], which generalizes the framework of REBAR, we express the estimator as

$$\mathbb{E}[\hat{g}] = \frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}[f(B) - h(B;\theta)] + \frac{\partial}{\partial\phi}\mathbb{E}_{B\sim p(B|\theta)}[h(B;\phi)] \tag{11}$$

To reduce variance, we can optimize ϕ to make sure $(f(B) - h(Z;\phi))^2 \approx 0$, which is the main idea of LAX in [10]. Since B is discrete, to further make the use of lower-variance property of reparameterization, [10] proposes to combine REINFORCE and the reparameterization of B

$$\hat{g} = f(B)\frac{\partial}{\partial\theta}\log p(B|\theta) - h(Z;\phi)\frac{\partial}{\partial\theta}\log p(Z|\theta) + \frac{\partial}{\partial\theta}h(Z_\theta;\phi) \tag{12}$$

where Z is the relaxed version of B, that is, $B \sim H(Z)$, $Z \sim p(Z|\theta)$, and usually we take $h(Z;\phi) = f(\sigma(Z))$ and denote Z_θ to emphasize that Z is reparameterized. The first term is unbiased but suffer from high variance, the second term is biased with high variance, and the third term is biased having low variance. Intuitively, this formulation will produce an unbiased estimator with much smaller variance. However, there is still room for improvement. Note that if we could correlate $p(B|\theta)$ and $p(Z|\theta)$, then the estimator's variance would be expected to reduced, since Z is strongly correlated with B then $h(Z)$ would be expected to positively correlate with $f(B)$. Based on this focus, we could use *conditional relaxation*, that is, imposing Z to be conditional to B, which is the key idea of REBAR.

Specifically, we have

$$\begin{aligned}
\mathbb{E}_{B\sim p(B|\theta)}[f(B)] &= \mathbb{E}_{B\sim p(B|\theta)}[f(B)] - \mathbb{E}_{Z\sim p(Z|\theta)}[h(Z;\phi)] + \mathbb{E}_{Z\sim p(Z|\theta)}[h(Z;\phi)] \\
&= \mathbb{E}_{B\sim p(B|\theta)}[f(B)] - \mathbb{E}_{B\sim p(B|\theta)}\left[\mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] + \mathbb{E}_{Z\sim p(Z|\theta)}[h(Z;\phi)] \\
&= \mathbb{E}_{B\sim p(B|\theta)}\left[f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] + \mathbb{E}_{Z\sim p(Z|\theta)}[h(Z;\phi)]
\end{aligned}$$

Taking derivatives of both sides, we obtain

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}[f(B)] \\
&= \frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}\left[f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] + \frac{\partial}{\partial\theta}\mathbb{E}_{Z\sim p(Z|\theta)}[h(Z;\phi)] \\
&= \frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}\left[f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] + \mathbb{E}_{Z\sim p(Z|\theta)}\left[\frac{\partial}{\partial\theta}h(Z;\phi)\right].
\end{aligned} \tag{13}$$

The last line holds since Z is reparameterizable. We now focus on the first term. Specifically, we have

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}\left[f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] \\
&= \mathbb{E}_{B\sim p(B|\theta)}\left[(f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)])\frac{\partial}{\partial\theta}\log p(B|\theta) - \frac{\partial}{\partial\theta}\mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right].
\end{aligned}$$

Another key observation is that the conditional distribution $p(Z|B,\theta)$ is also reparameterizable, which is validated as follows.

Proof. First, we have that all Z_i follow the Gumbel distribution $G(\theta_i)$. Then the probability $\mathbb{P}(B=k)$ is equivalently to the event that K is the index of largest values among $Z_1, Z_2, \ldots, Z_n$, and hence

$$\begin{aligned}
\mathbb{P}(B=k, Z=z) &= \prod_{i=1}^{n} p(z_i)\mathbb{I}\{z_k \geq z_i\} \\
&= p(z_k;\theta_k) \prod_{j=1,j\neq k}^{n} F(z_k;\theta_j) \prod_{j=1,j\neq k}^{n} \frac{p(z_i;\theta_i)\mathbb{I}\{z_k \geq z_i\}}{F(z_k;\theta_i)} \\
&= e^{-(z_k-\theta_k)} \prod_{j=1}^{n} F(z_k;\theta_j) \prod_{j=1,j\neq k}^{n} \frac{p(z_i;\theta_i)\mathbb{I}\{z_k \geq z_i\}}{F(z_k;\theta_i)} \\
&= e^{\theta_k} e^{-z_k} F\left(z_k; \log\left(\sum_{j=1}^{n} e^{\theta_j}\right)\right) \prod_{j=1,j\neq k}^{n} \frac{p(z_i;\theta_i)\mathbb{I}\{z_k \geq z_i\}}{F(z_k;\theta_i)} \\
&= p_k e^{-z_k + \log\left(\sum_{j=1}^{n} e^{\theta_j}\right)} F\left(z_k; \log\left(\sum_{j=1}^{n} e^{\theta_j}\right)\right) \prod_{j=1,j\neq k}^{n} \frac{p(z_i;\theta_i)\mathbb{I}\{z_k \geq z_i\}}{F(z_k;\theta_i)} \\
&= p_k p\left(z_k; \log\left(\sum_{j=1}^{n} e^{\theta_j}\right)\right) \prod_{j=1,j\neq k}^{n} \frac{p(z_i;\theta_i)\mathbb{I}\{z_k \geq z_i\}}{F(z_k;\theta_i)} \\
&= p(B=k)p(Z=z|B=k).
\end{aligned}$$

From here, we see that given the value of $B=k$, the k-th Gumbel random variable Z_k follow a Gumbel distribution with location $\log\left(\sum_{j=1}^{n} e^{\theta_j}\right)$, while the other Z_j follows a truncated Gumbel $TG(\theta_j; z_k)$. In particular, if we set $\theta_i = \log p_i$[3], then obviously $\log\left(\sum_{j=1}^{n} e^{\theta_j}\right) = 0$ and hence $Z_k \sim \text{Gumbel}(0)$, which is independent of the parameter θ. □

This observation enables the use of reparameterization for $\mathbb{P}(Z|B=k,\theta)$. More precisely, by exploiting the expression equation 5, we have

$$z_i = \begin{cases} -\log(-\log v_i) + \log\left(\sum_{j=1}^{n} e^{\theta_j}\right), & i = k \\ -\log\left(-\frac{\log v_i}{e^{\theta_i}} + e^{-z_k}\right), & i \neq k \end{cases} \tag{14}$$

where $v_i \sim \text{Uniform}(0,1)$ for all i. By noting $\theta_i = \log p_i$ and $\sum_{i=1}^{n} p_i = 1$, we can further simplify this expression as

$$z_i = \begin{cases} -\log(-\log v_i), & i = k \\ -\log\left(-\frac{\log v_i}{p_i} + e^{-z_k}\right), & i \neq k. \end{cases} \tag{15}$$

It is simpler for the case of binary variables. Suppose $b \sim \text{Bernoulli}(p)$, where $p \in (0,1)$; Denoting the CDF of a Logistic distribution by F with location $\log\frac{p}{1-p}$, then from equation 9 we can draw samples by letting $b = H(z)$, where $z = F^{-1}(u) = \log\frac{u}{1-u} + \log\frac{p}{1-p}$ with $u \sim \text{Uniform}(0,1)$, which is equivalent to $z \sim \text{Logistic}(\log\frac{p}{1-p})$. For conditioned samples, we can sample from $p(z|b)$ truncating the Logistic function with

$$z = \begin{cases} \log\left(1 + \frac{u}{(1-u)(1-p)}\right), & d = 1 \\ -\log\left(1 + \frac{1-u}{pu}\right), & d = 0 \end{cases} \tag{16}$$

Proof. The CDF of Logistic distribution is $F(z;\mu) = \frac{1}{1+e^{-(z-\mu)}}$. We can sample from $\text{Bernoulli}(p)$ with setting $z \sim F(z; \log\frac{p}{1-p})$ and $b = H(z)$. Now, we derive how to draw conditioned samples from $p(z|b)$, which can be achieved via truncating the logistic function.

[3]Note that this equation always holds in the Gumbel-Max trick.

Sampling from $p(z|b=1)$:

$$\begin{aligned}
\mathbb{P}(Z \leq z|b=1) = \mathbb{P}(Z \leq z|Z \geq 0) &= \frac{\mathbb{P}(0 \leq Z \leq z)}{\mathbb{P}(Z \geq 0)} \\
&= \frac{F(z;\mu) - F(0;\mu)}{1 - F(0;\mu)} = u \in (0,1) \\
\log u &= \log\left[F(z;\mu) - F(0;\mu)\right] - \log\left[1 - F(0;\mu)\right] \\
z &= \mu - \log\frac{1-u}{u + e^{-\mu}} \\
z &= \log\left(1 + \frac{u}{(1-u)(1-p)}\right) \qquad (\mu = \log\frac{p}{1-p})
\end{aligned}$$

Sampling from $p(z|b=0)$:

$$\begin{aligned}
\mathbb{P}(Z \leq z|b=0) = \mathbb{P}(Z \leq z|Z \leq 0) &= \frac{\mathbb{P}(Z \leq \min(z,0))}{\mathbb{P}(Z \leq 0)} \\
&= \frac{F(z;\mu)}{F(0;\mu)} = u \in (0,1) \\
\log u &= \log F(z;\mu) - \log F(-;\mu) \\
z &= \mu - \log\frac{1 + e^{\mu} - u}{u} \\
z &= -\log\left(1 + \frac{1-u}{pu}\right) \qquad (\mu = \log\frac{p}{1-p})
\end{aligned}$$

Here, we arrive at the conditioned sampling strategy shown in equation 16. □

Equipped with these techniques, we can arrive the final version of our gradient estimator from equation 13:

$$\begin{aligned}
&\frac{\partial}{\partial\theta}\mathbb{E}_{B\sim p(B|\theta)}\left[f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)]\right] + \mathbb{E}_{Z\sim p(Z|\theta)}\left[\frac{\partial}{\partial\theta}h(Z;\phi)\right] \\
&= \mathbb{E}_{B\sim p(B|\theta)}\left[(f(B) - \mathbb{E}_{Z\sim p(Z|B,\theta)}[h(Z;\phi)])\frac{\partial}{\partial\theta}\log p(B|\theta) - \mathbb{E}_{Z\sim p(Z|B,\theta)}\left[\frac{\partial}{\partial\theta}h(Z;\phi)\right]\right] \\
&\quad + \mathbb{E}_{Z\sim p(Z|\theta)}\left[\frac{\partial}{\partial\theta}h(Z;\phi)\right].
\end{aligned}$$

And

$$\hat{g} = (f(b) - h(\tilde{z}))\frac{\partial}{\partial\theta}\log p(b|\theta) - \frac{\partial}{\partial\theta}h(\tilde{z}_\theta) + \frac{\partial}{\partial\theta}h(z_\theta), \tag{17}$$

where $z_\theta \sim p(Z|\theta), b = H(z)$ and $\tilde{z}_\theta \sim p(Z|B=b,\theta)$. We typically choose $h(\cdot)$ to be $f(\sigma(\cdot))$ such that it approaches $f(b)$ as closely as possible.

Remark (Why REBAR benefits from conditional marginalization?)**.** In REBAR, we seek a low-variance, unbiased gradient estimator. Inspired by the Concrete relaxation [9, 11], one strategy will be to derive gradient over the relaxation version, that is

$$\mathbb{E}_{p(b)}\left[f(b)\frac{\partial}{\partial\theta}\log p(b)\right] = \frac{\partial}{\partial\theta}\mathbb{E}_{p(b)}[f(b)] = \frac{\partial}{\partial\theta}\mathbb{E}_{p(z)}[f(H(z))] = \mathbb{E}_{p(z)}\left[f(H(z))\frac{\partial}{\partial\theta}\log p(z)\right].$$

Intuitively, the right hand side is more suitable to estimate the gradient, since we can construct a strongly correlated control variate, that is

$$\begin{aligned}
\frac{\partial}{\partial\theta}\mathbb{E}_{p(z)}[f(H(z))] &= \frac{\partial}{\partial\theta}\mathbb{E}_{p(z)}[f(H(z)) - f(\sigma(z))] + \frac{\partial}{\partial\theta}\mathbb{E}_{p(z)}[f(\sigma(z))] \\
&= \mathbb{E}_{p(z)}\left[(f(H(z)) - f(\sigma(z)))\frac{\partial}{\partial\theta}\log p(z)\right] + \mathbb{E}_{p(z)}\left[\frac{\partial}{\partial\theta}f(\sigma(z))\right].
\end{aligned} \tag{18}$$

However, the Monte Carlo gradient estimator derived from $\mathbb{E}_{p(b)}\left[f(b)\frac{\partial}{\partial\theta}\log p(b)\right]$ has much lower variance than the Monte Carlo gradient estimator derived from $\mathbb{E}_{p(z)}\left[f(H(z))\frac{\partial}{\partial\theta}\log p(z)\right]$. To show this, we provide the detail discussion as following.

Proof. Take $b \sim \text{Bernoulli}(p = \sigma(\theta))$. It is well known that $b = H(z)$, where H is the step function at point 0 and $z \sim \text{Logistic}(\theta)$. The key insight is that, if we generate the random variable $z \sim p(z|b)$, then $b = H(z)$ and thus we have $p(z) = p(z)p(b|z) = p(z)\mathbb{I}_{b=H(z)} = p(z, b) = p(b)p(z|b)$. Deriving the REINFORCE estimator, we have

$$\begin{aligned}\mathbb{E}_{p(z)}\left[f(H(z))\frac{\partial}{\partial\theta}\log p(z)\right] &= \mathbb{E}_{p(b)}\left[\mathbb{E}_{p(z|b)}\left[f(H(z))\frac{\partial}{\partial\theta}\log p(z)\right]\right] = \mathbb{E}_{p(b)}\left[f(b)\mathbb{E}_{p(z|b)}\left[\frac{\partial}{\partial\theta}\log p(z)\right]\right] \\ &= \mathbb{E}_{p(b)}\left[f(b)\mathbb{E}_{p(z|b)}\left[\frac{\partial}{\partial\theta}(\log p(z|b) + \log p(b))\right]\right] \\ &= \mathbb{E}_{p(b)}\left[f(b)\frac{\partial}{\partial\theta}\log p(b)\right].\end{aligned}$$

That is, when we use the term $\frac{\partial}{\partial\theta}\mathbb{E}_{p(b)}[f(b)]$ to estimate the gradient, we are actually conditional marginalizing the variable z given b: $\mathbb{E}_{p(z|b)}\left[\frac{\partial}{\partial\theta}\log p(z|b)\right]$. This conceptually gives the lower variance. This is the reason why REBAR derive the conditioned control variable of $\mathbb{E}_{p(b)}[f(b)]$ but not $\mathbb{E}_{p(z)}[f(H(z))]$. □

Remark. In fact, many discrete random variables can be seen as a version of conditional marginalizaion of their augmented continuous random variables. This is reasonable since intuitively, these discrete variables only take at most countable values but their continuous counterparts can take uncountably many values. Although the main motivation of [7, 10] was to construct (black-box) control variates, their conditional marginalization techniques greatly shed light on the research towards discrete variable gradient estimation, including [12].

2.3 ARM, ASRM and DisARM

2.3.1 ARM

ARM (Augment-REINFORCE-Merge) estimator [13] is specialized for binary latent variables, which is appealing to its unbiasedness and low variance. As argued in [12], the most essential technique for variance reduction in ARM estimator is to use antithetic samples which are negatively correlated.

Specifically, ARM estimator is derived by augmenting a coutinuous random variable $z \sim \text{Logistic}(\theta, 1)$, and then obtaining the binary random variable $b = \mathbb{I}_{z\geq 0} \sim \text{Bernoulli}$. Denoting the distribution of z by $p_\theta(z) = \frac{e^{-(x-\theta)}}{(1+e^{-(x-\theta)})^2}$, we have

$$\frac{\partial}{\partial\theta}\mathbb{E}_{b\sim\text{Bernoulli}(\sigma(\theta))}[f(b)] = \frac{\partial}{\partial\theta}\mathbb{E}_{z\sim\text{Logistic}(\theta,1)}[f(\mathbb{I}_{z>0})] = \mathbb{E}_{p_\theta(z)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right]$$

We could estimate this term by taking *antithetic sample* pairs $(z, \tilde{z})$, that is, we first sample $z = \epsilon + \theta$ with $\epsilon \sim \text{Logistic}(\theta, 1)$ and then construct another sample $\tilde{z} = -\epsilon + \theta$. By noting that

$$\begin{aligned}\frac{\partial}{\partial\theta}\log p_\theta(z) &= \frac{\partial}{\partial\theta}(-z + \theta - 2\log(1 + e^{-z+\theta})) \\ &= 1 - \frac{2e^{-z+\theta}}{1 + e^{-z+\theta}} = 1 - \frac{2 + 2e^{-z+\theta} - 2}{1 + e^{-z+\theta}}\end{aligned}$$

We have

$$\begin{aligned}\mathbb{E}_{p_\theta(z)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right] &\approx \frac{1}{2}\left(f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z) + f(\mathbb{I}_{\tilde{z}>0})\frac{\partial}{\partial\theta}\log p_\theta(\tilde{z})\right) \\ &= \frac{1}{2}(f(\mathbb{I}_{z>0}) - f(\mathbb{I}_{\tilde{z}>0}))\frac{\partial}{\partial\theta}\log p_\theta(z) \\ &= \frac{1}{2}(f(\mathbb{I}_{z>0}) - f(\mathbb{I}_{\tilde{z}>0}))(2\sigma(z-\theta) - 1) \\ &:= g_{\text{ARM}}(z, \tilde{z})\end{aligned}$$

Here we arrive at the ARM estimator, which can be readily extended to multiple dimensional cases, though at the cost of introducing additional variance.

2.3.2 DisARM

We now introduce its variant DisARM [12], which further reduces the variance of ARM by marginalizing out the continuous augmentation z. By analyzing the variance of ARM, [12] shows that without antithetic sampling the estimator is at least as large as the REINFORCE by observing that

$$\begin{aligned}
\mathbb{E}_{p_\theta(z|b)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right] &= f(b)\mathbb{E}_{p_\theta(z|b)}\left[\frac{\partial}{\partial\theta}\log p_\theta(z)\right] \\
&= f(b)\mathbb{E}_{p_\theta(z|b)}\left[\frac{\partial}{\partial\theta}(\log p_\theta(z|b)+\log p_\theta(b))\right] \\
&= f(b)\mathbb{E}_{p_\theta(z|b)}\left[\frac{\partial}{\partial\theta}\log p_\theta(b)\right] \\
&= f(b)\frac{\partial}{\partial\theta}\log p_\theta(b)
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\mathrm{Var}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right] \\
&= \mathrm{Var}_{p(b)}\left[\mathbb{E}_{p(z|b)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right]\right] + \mathbb{E}_{p(b)}\left[\mathrm{Var}_{p(z|b)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right]\right] \\
&= \mathrm{Var}\left[f(b)\frac{\partial}{\partial\theta}\log p_\theta(b)\right] + \mathbb{E}_{p(b)}\left[\mathrm{Var}_{p(z|b)}\left[f(\mathbb{I}_{z>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right]\right] \\
&\geq \mathrm{Var}\left[f(b)\frac{\partial}{\partial\theta}\log p_\theta(b)\right].
\end{aligned}$$

Here, the second line holds based on the law of total variance. Therefore, while ARM reduces variance via antithetic coupling, it also increases variance dut to the reparameterization. To further reduce the variance of ARM, [12] proposes to marginalize the z *conditional on* $(b, \tilde{b})$:

$$\begin{aligned}
g_{\mathrm{DisARM}}(b,\tilde{b}) := \mathbb{E}_{p(z|b,\tilde{b})}[g_{\mathrm{ARM}}] &= \frac{1}{2}\mathbb{E}_{p(z|b,\tilde{b})}\left[f(\mathbb{I}_{z>0}) - f(\mathbb{I}_{\tilde{z}>0})\frac{\partial}{\partial\theta}\log p_\theta(z)\right] \\
&= \frac{1}{2}(f(b)-f(\tilde{b}))\mathbb{E}_{p(z|b,\tilde{b})}\left[\frac{\partial}{\partial\theta}\log p_\theta(z)\right] \\
&= \frac{1}{2}(f(b)-f(\tilde{b}))\left((-1)^{\tilde{b}}\mathbb{I}_{b\neq\tilde{b}}\sigma(|\theta|)\right).
\end{aligned}$$

It turns out that the variance of DisARM is upper bounded by the variance of ARM

$$\begin{aligned}
\mathrm{Var}(g_{\mathrm{ARM}}) &= \mathrm{Var}_{b,\tilde{b}}\left[\mathbb{E}_{z|b,\tilde{b}}\left[g_{\mathrm{ARM}}\right]\right] + \mathbb{E}_{b,\tilde{b}}\left[\mathrm{Var}_{z|b,\tilde{b}}\left[g_{\mathrm{ARM}}\right]\right] \\
&= \mathrm{Var}\left[g_{\mathrm{DisARM}}\right] + \mathbb{E}_{b,\tilde{b}}\left[\mathrm{Var}_{z|b,\tilde{b}}\left[g_{\mathrm{ARM}}\right]\right] \\
&\geq \mathrm{Var}\left[g_{\mathrm{DisARM}}\right].
\end{aligned}$$

2.3.3 ARSM

ARSM (Augment-Reinforce-Swap-Merge) estimator [14] is a recent technique to estimate gradients through categorical variables, which is a generalization of ARM estimator for binary cases [13]. An interesting observation is that, both ARM and ARSM algorithms are derived by estimation gradients *w.r.t* the *pre-activation* parameters ϕ (*i.e.,* the distribution parameters $\theta = \sigma(\phi)$, where $\sigma(\cdot)$ is either sigmoid or softmax function). Recall that for categorical distributions, we focus on the following problem setting: denote $z \sim \mathrm{Cat}(\sigma(\phi))$ as a categorical variable such that $P(z = c|\phi) = \sigma(\phi)_c = e^{\phi_c}/\sum_{i=1}^{C} e^{\phi_i}$, where $\phi := (\phi_1, \ldots, \phi_C)$ and $\sigma(\phi) := (e^{\phi_1,\ldots,e^{\phi_C}})/\sum_{i=1}^{C} e^{\phi_i}$ is the softmax function. For the expectated objective defined as

$$\mathcal{E}(\phi) := \mathbb{E}_{z\sim\mathrm{Cat}(\sigma(\phi))}[f(z)] = \sum i = 1^C f(i)\sigma(\phi)_i, \tag{19}$$

the gradient can be expressed analytically as

$$\nabla_{\phi_c}\mathcal{E}(\phi) = \sigma(\phi)_c f(c) - \sigma(\phi)_c \mathcal{E}(\phi); \tag{20}$$

or expressed with REINFORCE as

$$\nabla_{\phi_c}\mathcal{E}(\phi) = \mathbb{E}_{z\sim\mathrm{Cat}(\sigma(\phi))}\left[f(z)(\mathbb{I}_{[z=c]} - \sigma(\phi)_c)\right]. \tag{21}$$

Like ARM, ARSM also follows an intuitive construction, which is detailed as follows:

AR: Augment and REINFORCE It is known that a categorical random variable sample can be generated by taking $\arg\min$ of a set of exponential random variables $\tau \sim \mathrm{Exp}(\lambda) = \lambda e^{-\lambda\tau}$. More precisely,

$$\mathbb{P}\left(z = \arg\min_{i\in\{1,\dots,C\}} \tau_i\right) = \mathbb{P}(\tau_z < \tau_i, \forall i \neq z) = \frac{\lambda_z}{\sum_{i=1}^C}\lambda_i.$$

Letting $\lambda_i = e^{\phi_i}$, we see that z is a sample from the desired categorical distribution. In this way, we could re-write the expected objective function 19 as

$$\mathcal{E}(\phi) = \mathbb{E}_{z\sim\mathrm{Cat}(\sigma(\phi))}[f(z)] = \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)\right].$$

With this in mind, we take its gradient with respect to ϕ_c using REINFORCE and have

$$\begin{aligned}
\nabla_{\phi_c}\mathcal{E}(\phi) &= \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)\nabla_{\phi_c}\log\prod_{i=1}^C \mathrm{Exp}(\tau_i; e^{\phi_i})\right] \\
&= \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)\nabla_{\phi_c}\log e^{\phi_c - e^{\phi_c}\tau_c}\right] \\
&= \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)\nabla_{\phi_c}(\phi_c - e^{\phi_c}\tau_c)\right] \\
&= \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)(1 - e^{\phi_c}\tau_c)\right].
\end{aligned}$$

Also note that the ϕ-parameterized exponential random variables $\tau_i \sim \mathrm{Exp}(e^{\phi_i})$ are also reparameterizable, in the sense that $\tau_i = \epsilon_i e^{-\phi_i}, \epsilon_i \sim \mathrm{Exp}(1)$. Thus we could re-express the REINFORCE estimator using the expectation over parameter-independent exponential random variables ϵ_i

$$\begin{aligned}
\nabla_{\phi_c}\mathcal{E}(\phi) &= \mathbb{E}_{\tau_1\sim\mathrm{Exp}(e^{\phi_1}),\dots,\tau_C\sim\mathrm{Exp}(e^{\phi_C})}\left[f(\arg\min_{i\in\{1,\dots,C\}}\tau_i)(1 - e^{\phi_c}\tau_c)\right] \\
&= \mathbb{E}_{\epsilon_1\sim\mathrm{Exp}(1),\dots,\epsilon_C\sim\mathrm{Exp}(1)}\left[f(\arg\min_{i\in\{1,\dots,C\}}\epsilon_i e^{\phi_i})(1 - \epsilon_c)\right].
\end{aligned}$$

Instead of using the augmented version of REINFORCE estimator, they further augment again the exponentials by Dirichlet random variables. That is, setting $\epsilon_i = \epsilon\pi_i$, with $\pi_1, \pi_2, \dots, \pi_C \sim \mathrm{Dirichlet}(1)$ and $\epsilon \sim \mathrm{Gamma}(C, 1)$, then $\epsilon_i \sim \mathrm{Exp}(1)$. They perform Rao-blackwellization on the

augmented Gamma random variable ϵ as follows

$$\begin{aligned}
\nabla_{\phi_c}\mathcal{E}(\phi) &= \mathbb{E}_{\epsilon_1\sim\text{Exp}(1),\ldots,\epsilon_C\sim\text{Exp}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\epsilon_i e^{\phi_i})(1-\epsilon_c)\right] \\
&= \mathbb{E}_{\epsilon\sim\text{Gamma}(C,1),\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\epsilon\pi_i e^{-\phi_i})(1-\epsilon\pi_c)\right] \\
&= \mathbb{E}_{\epsilon\sim\text{Gamma}(C,1),\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\epsilon\pi_i e^{-\phi_i})\right] - \\
&\quad \mathbb{E}_{\epsilon\sim\text{Gamma}(C,1),\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\epsilon\pi_i e^{-\phi_i})\epsilon\pi_c\right] \\
&= \mathbb{E}_{\epsilon\sim\text{Gamma}(C,1),\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})\right] - \\
&\quad \mathbb{E}_{\epsilon\sim\text{Gamma}(C,1),\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})\epsilon\pi_c\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})(1-C\pi_c)\right].
\end{aligned}$$

ARS: AR with swapping One important point of AR estimator is that, taking the advantage of Dirichlet random variables, they are allowed to perform variance reduction through common random numbers. Since swapping any two elements in the probability vector π does not change the expectation, we essentially have that choosing a **reference** random index $j \in \{1, 2, \ldots, C\}$,

$$\begin{aligned}
\nabla_{\phi_c}\mathcal{E}(\phi) &= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})(1-C\pi_c)\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i^{c\rightleftharpoons j} e^{-\phi_i})(1-C\pi_j)\right],
\end{aligned}$$

where $\pi_c^{i\rightleftharpoons j}$ means that $\pi_c^{c\rightleftharpoons j} = \pi_j, \pi_j c \rightleftharpoons j = \pi_c$ and $\pi_i^{c\rightleftharpoons j} = \pi_i$ for all $\notin \{c, j\}$. The above equation holds since if $\pi \sim \text{Dirichlet}(1)$, then $\pi^{i\rightleftharpoons j} \sim \text{Dirichlet}(1)$.

The authors then consider the sum of gradient estimators with respect to all ϕ_i, that is, $\frac{1}{C}\sum_{i=1}^{C}\nabla_{\phi_i}\mathcal{E}(\phi) = \frac{1}{C}\sum_{k=1}^{C} f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i^{k\rightleftharpoons j}e^{-\phi_i})(1-C\pi_j)$. They show that fixing the reference j, this average is 0 under the expectation over π

$$\begin{aligned}
&\mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[\frac{1}{C}\sum_{k=1}^{C} f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i^{k\rightleftharpoons j}e^{-\phi_i})(1-C\pi_j)\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[\frac{1}{C}\sum_{k=1}^{C} f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})(1-C\pi_k)\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})(1-\frac{1}{C}\sum_{k=1}^{C}C\pi_k)\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(\arg\min_{i\in\{1,\ldots,C\}}\pi_i e^{-\phi_i})(1-1)\right] = 0, \qquad (22)
\end{aligned}$$

where we repeatedly use the property that if $\pi \sim \text{Dirichlet}(1)$, then $\pi^{i\rightleftharpoons j} \sim \text{Dirichlet}(1)$. Obviously, this can be served as a baseline for AR estimator. Combining this 0-expectation term, we arrive the ARS estimator (for notation convenience, we resume $z^{c\rightleftharpoons j} = \arg\min_{i\in\{i,\ldots,C\}}\pi_i^{c\rightleftharpoons j}$):

$$\begin{aligned}
\nabla_{\phi_c}\mathcal{E}(\phi) &= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[f(z^{c\rightleftharpoons j})(1-C\pi_j) - \frac{1}{C}\sum_{k=1}^{C} f(z^{k\rightleftharpoons j})(1-C\pi_j)\right] \\
&= \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[\left(f(z^{c\rightleftharpoons j}) - \frac{1}{C}\sum_{k=1}^{C} f(z^{k\rightleftharpoons j})\right)(1-C\pi_j)\right]. \qquad (23)
\end{aligned}$$

ARSM: ARS with merging The merging step is intuitive. Given that the ARS estimator above requires a fixed reference j, we could enumerate all possible references j and average the resulting ARS estimator all together to obtain lower variance:

$$\nabla_{\phi_c}\mathcal{E}(\phi) = \mathbb{E}_{\pi\sim\text{Dirichlet}(1)}\left[\sum_{j=1}^{C}\left(f(z^{c\rightleftharpoons j}) - \frac{1}{C}\sum_{k=1}^{C} f(z^{k\rightleftharpoons j})\right)(1 - C\pi_j)\right]. \tag{24}$$

Remark. Here are some remarks from the point of my view:

- The derivation of AR estimator is not elegant. They make several layers of augmentations (and then integrating one augmented random variable out, which is somewhat weird) to obtain their familiar formulation (that is, the Dirichlet). There should be some neater way to achieve AR estimator.
- The main problematic step is the swapping operation to introduce a baseline. As far as I can see, there is no clear intuition that the baseline strongly correlates with the objective, although it can view as "pseudo-actions" that implements "try-and-see" behaviour.
- As addressed in [12], continuous augmentation may lead higher variance. There should also be some improvement.

2.4 Rao-Blackwellization

2.4.1 Sum-and-sample estimator

Recently, [15] addressed estimating problems in the setting of univariate discrete varaibles with at most countably many categories. The key observation is that we should take advantages of sparsity of the entire space since most masses could be concentrated on only a few points. Therefore, to reduce variance of the original REINFORCE estimator [2], we can find a partition of the support of this random variable with forms two sets: (i) C_k consists points z such that its mass function $p_\theta(z)$ is one of the largest k values, and (ii) C_k^c, the complement of C_k. Suppose we want to compute $\mathbb{E}_p(g(z))$, then we can write it as

$$\begin{aligned}
\mathbb{E}_p(g(z)) &= \sum_x p_\theta(z)g(z) \\
&= \sum_{x\in C_k} p_\theta(z)g(z) + \sum_{z\in C_k^c} p_\theta(z)g(z) \\
&= \sum_{x\in C_k} p_\theta(z)g(z) + (1-\mathbb{P}(C_k))\sum_{z\in C_k^c}\frac{p_\theta(z)}{1-\mathbb{P}(C_k)}g(z) \\
&= \sum_{x\in C_k} p_\theta(z)g(z) + (1-\mathbb{P}(C_k))\sum_{z\in C_k^c} p_\theta(Z=z|z\in C_k^c)g(z) \\
&= \sum_{x\in C_k} p_\theta(z)g(z) + (1-\mathbb{P}(C_k))\mathbb{E}_{p(z|C_k^c)}[g(z)].
\end{aligned} \tag{25}$$

Then, we perform exact summation of the first term and approximate the second term by sampling some value z from the conditional distribution $p(z|C_k^c)$. Intuitively, although we take random samples to approximate the second term, $1-\mathbb{P}(C_k)$ is often small so that it is expected to have small variance. The authors also show that this can be seen as a Rao-Blackwellized version of REINFORCE estimator, which is guaranteed to have lower variance.

2.4.2 Sampling withou replacement

The Rao-Blackwellization seems to be very popular in recent years. [16] derived a novel estimator for discrete distributions based on sampling without replacementm which is equivalent to Rao-Blackwellizing several kinds of other estimators, which can reduce variance.

Preliminaries

Before introducing their methods in detail, there are some preliminaries that facilitate developing of their algorithms.

Restricted Distribution Suppose the entire range of some discrete random variable is D, and now we would like to restrict D to a smaller set $D\backslash C$ with $C \subseteq D$. Now for any $x \in C \subseteq D$, we have its probability mass as:

$$p^{D\backslash C}(x) = \frac{p(x)}{1 - \sum_{c \in C} p(c)},$$

where $p^{D\backslash C}$ denotes the probability mass function after restricting D to a smaller range $D\backslash C$.

Ordered sample without replacement B^k An *ordered sample without replacement* $B^k = (b_1, \ldots, b_k), b_i \in D$ can be generated in the following way: first sample b_1 from p, and then sample b_2 from $p^{D\backslash\{b_1\}}$, b_3 from $p^{D\backslash\{b_1,b_2\}}$, *etc.* The probability mass function of such ordered sample is

$$p(B^k) = \prod_{i=1}^{k} p^{D\backslash B^{i-1}}(b_i) = \prod_{i=1}^{k} \frac{p(b_i)}{1 - \sum_{c \in B^{i-1}} p(c)}.$$

Unordered sample without replacement S^k An *unordered* sample without replacement S^k is generated by discarding the ordered sample B^k. Thus, enumerating all possible permutations, we have

$$p(S^k) = \sum_{B^k \in \pi(S^k)} p(B^k) = \sum_{B^k \in \pi(S^k)} \prod_{i=1}^{k} \frac{p(b_i)}{1 - \sum_{c \in B^{i-1}} p(c)} = \left(\prod_{s \in S^k} p(s) \right) \sum_{B^k \in \pi(S^k)} \prod_{i=1}^{k} \frac{1}{1 - \sum_{c \in B^{i-1}} p(c)}$$

where we denote $pi(S^k)$ as the set of all $k!$ permutations (orderings) B^k that correspond to (could be generated by) S^k.

The Gumbel-top-k trick As an alternative to sequential sampling, [17] proposed to sample B^k and S^k by taking the top k of Gumbel variables. Specifically, we have

Theorem 2.1. *Suppose* $b_1, b_2, \ldots, b_k = \arg\operatorname{top}\ k\ \mathrm{Gumbel}(\phi_i)$*, which are the* k *largest values in decreasing order, that is,* $b_1 = \arg\max_{i \in D} \mathrm{Gumbel}(\phi_i), b_2 = \arg\max_{i \in D\backslash\{b_1\}} \mathrm{Gumbel}(\phi_i)$*, etc. Then the collection* $B^k = \{b_1, \ldots, b_k\}$ *forms an ordered sample without replacement from* $\mathrm{Categorical}(\phi)$*. In other words, the following holds:*

$$p(I_1 = b_1, I_2 = b_2, \ldots, I_k = b_k) = \prod_{j=1}^{k} \frac{\exp(\phi_{b_j})}{\sum_{\ell \in D_j^*} \exp(\phi_\ell)}, \tag{26}$$

where $D_j^* = D\backslash\{b_1, \ldots, b_j\}$ *is the domain (without replacement) for the* j*-th sampled element. It follows that taking the top* k *perturbed indexes without order, we obtain the unordered sample set* S^k*.*

The above trick provides an alternative sampling method which is effectively a (non-differentiable) reparameterization of sampling without replacement.

Construction of the estimator

The authors make an observation that the expectation from $\mathbb{E}_p[g(b)]$ can be written as using only a single sample $b \in B^k$:

$$\mathbb{E}_{B^k \sim p(B^k)}[f(b_1)] = \mathbb{E}_{b_1 \sim p(b_1)}[f(b_1)] = \mathbb{E}_p[g(b)].$$

To make better use of the k samples in B^k instead of using only a single sample, a Rao-Blackwellized version is further developed, that is, we consider the probability of B^k conditional on S^k. Thus we

have

$$\begin{aligned}
\mathbb{E}_{B^k\sim p(B^k)}[g(b_1)] &= \mathbb{E}_{S^k\sim p(S^k)}[\mathbb{E}_{B^k\sim\mathbb{P}(B^k|S^k)}[g(b_1)]] \\
&= \mathbb{E}_{S^k\sim\mathbb{P}(S^k)}[\mathbb{E}_{b_1\sim p(b_1|S^k)}[g(b_1)]] \\
&= \mathbb{E}_{S^k\sim\mathbb{P}(S^k)}[\sum_{s\in S^k} p(b_1=s|S^k)g(s)] \\
&= \mathbb{E}_{S^k\sim\mathbb{P}(S^k)}[\sum_{s\in S^k} \frac{p(s)p(S^k|s)}{p(S^k)}g(s)] \\
&= \mathbb{E}_{S^k\sim\mathbb{P}(S^k)}[\sum_{s\in S^k} \frac{p(s)p^{D\setminus\{s\}}(S^k\setminus\{s\})}{p(S^k)}g(s)].
\end{aligned}$$

Denoting $R(S^k,s)=p^{D\setminus\{s\}}(S^k\setminus\{s\})/p(S^k)$, we obtain an estimator of the quantity $\mathbb{E}_p[g(b)]$ of interest that is unbiased and has lower variance since it is a Rao-Blackwellization:

$$\mathbb{E}_p[g(b)] = \mathbb{E}_{S^k\sim\mathbb{P}(S^k)}[\sum_{s\in S^k} p(s)R(S^k,s)g(s)].$$

Now it is ready to derive the WOR-REINFORCE (*WithOut Replacement*) estimator. Suppose the distribution parameter θ is the main focus and we aim to optimize the objective $\mathbb{E}_{p_\theta(b)}[f_\theta(b)]$. Then using the same philosophy above, we have

$$\begin{aligned}
\nabla_\theta\mathbb{E}_{p_\theta(b)}[f_\theta(b)] &= \mathbb{E}_{p_\theta(b)}[f_\theta(b)\nabla_\theta\log p_\theta(b)] + \mathbb{E}_{p_\theta(b)}[\nabla_\theta f_\theta(b)] \\
&= \mathbb{E}_{S^k\sim p(S^k)}\left[\sum_{s\in S^k} p_\theta(s)R(S^k,s)f_\theta(s)\nabla_\theta\log p_\theta(s)\right] + \mathbb{E}_{p_\theta(b)}[\nabla_\theta f_\theta(b)] \\
&= \mathbb{E}_{S^k\sim p(S^k)}\left[\sum_{s\in S^k} \nabla_\theta p_\theta(s)R(S^k,s)f_\theta(s)\right] + \mathbb{E}_{p_\theta(b)}[\nabla_\theta f_\theta(b)].
\end{aligned}$$

The variance of REINFORCE can be reduced by subtracting a baseline function. Similar to VIMCO [18], using multiple samples enables a **built-in control variate** that is computed using all of the other samples. However, since these samples are not independent (as they are sample without replacement), we have to correct these dependencies. An important observation is that the following expression has zero expectation (here we omit the dependency on θ of f since it could always be recovered as similar to the second term above):

$$\mathbb{E}_{S^k\sim p_\theta(S^k)}\left[\sum_{s\in S^k}\nabla_\theta p_\theta(s)R(S^k,s)\sum_{s'\in S^k} p_\theta(s')R^{D\setminus\{s\}}(S^k,s')f(s')\right]=0, \tag{27}$$

where

$$R^{D\setminus\{s\}}(S^k,s') = \frac{p_\theta^{D\setminus\{s,s'\}}(S^k\setminus\{s,s'\})}{p_\theta^{D\setminus\{s\}}(S^k\setminus\{s\})} = \begin{cases} \frac{p_\theta^{D\setminus\{s,s'\}}(S^k\setminus\{s,s'\})}{p_\theta^{D\setminus\{s\}}(S^k\setminus\{s\})}, & \text{if } s'\neq s \\ 1, & \text{if } s'=s \end{cases}.$$

Proof. We apply Bayes' Theorem conditionally on $b_1=s$ to derive for $s'\neq s$

$$\begin{aligned}
p(b_2=s'|S^k,b_1=s) &= \frac{p(S^k|b_2=s',b_1=s)p(b_2=s'|b_1=s)}{p(S^k|b_1=s)} \\
&= \frac{p_\theta^{D\setminus\{s,s'\}}(S^k\setminus\{s,s'\})p_\theta^{D\setminus\{s\}}(s')}{p_\theta^{D\setminus\{s\}}(S^k\setminus\{s\})} \\
&= \frac{p_\theta(s')}{1-p_\theta(s)}R^{D\setminus\{s\}}(S^k,s').
\end{aligned}$$

For $s' = s$ we have $R^{D\setminus\{s\}}(S^k, s') = 1$,so using the above equation, we can show that

$$\begin{aligned}
&\sum_{s'\in S^k} p_\theta(s') R^{D\setminus\{s\}}(S^k, s') f(s') \\
&= p_\theta(s) f(s) + \sum_{s'\in S^k\setminus\{s\}} p_\theta(s') R^{D\setminus\{s\}}(S^k, s') f(s') \\
&= p_\theta(s) f(s) + (1 - p_\theta(s)) \sum_{s'\in S^k\setminus\{s\}} p(b_2 = s' | S^k, b_1 = s) f(s')
\end{aligned}$$

Substituting this to the term inside the expectation of equation 27, we have

$$\begin{aligned}
&\sum_{s\in S^k} \nabla_\theta p_\theta(s) R(S^k, s) \sum_{s'\in S^k} p_\theta(s') R^{D\setminus\{s\}}(S^k, s') f(s') \\
&= \sum_{s\in S^k} p_\theta(s) R^{D\setminus\{s\}}(S^k, s') \nabla_\theta \log p_\theta(s) \sum_{s'\in S^k} p_\theta(s') R^{D\setminus\{s\}}(S^k, s') f(s') \\
&= \sum_{s\in S^k} p(b_1 = s | S^k) \nabla_\theta \log p_\theta(s) \sum_{s'\in S^k} p_\theta(s') R^{D\setminus\{s\}}(S^k, s') f(s') \\
&= \sum_{s\in S^k} p(b_1 = s | S^k) \nabla_\theta \log p_\theta(s) p_\theta(s) f(s) + \\
&\quad \sum_{s\in S^k} p(b_1 = s | S^k) \nabla_\theta \log p_\theta(s) (1 - p_\theta(s)) \sum_{s'\in S^k\setminus\{s\}} p(b_2 = s' | S^k, b_1 = s) f(s') \\
&= \sum_{s\in S^k} p(b_1 = s | S^k) \nabla_\theta \log p_\theta(s) \left[p_\theta(s) f(s) + (1 - p_\theta(s)) \mathbb{E}_{p(b_2|b_1=s,S^k)}[f(b_2)] \right] \qquad (28)
\end{aligned}$$

Wrapping the above with the external expectation, we have

$$\begin{aligned}
&\mathbb{E}_{S^k\sim p_\theta(S^k)} \left[\sum_{s\in S^k} p(b_1 = s | S^k) \nabla_\theta \log p_\theta(s) \left[p_\theta(s) f(s) + (1 - p_\theta(s)) \mathbb{E}_{p(b_2|b_1=s,S^k)}[f(b_2)] \right] \right] \\
&= \mathbb{E}_{S^k\sim p_\theta(S^k)} \left[\mathbb{E}_{p_\theta(b_1|S^k)} \left[\nabla_\theta \log p_\theta(b_1) \left[p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) \mathbb{E}_{p(b_2|b_1,S^k)}[f(b_2)] \right] \right] \right] \\
&= \mathbb{E}_{S^k\sim p_\theta(S^k)} \left[\mathbb{E}_{B^k\sim p_\theta(B^k|S^k)} \left[\nabla_\theta \log p_\theta(b_1) [p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) f(b_2)] \right] \right] \\
&= \mathbb{E}_{B^k\sim p_\theta(B^k)} \left[\nabla_\theta \log p_\theta(b_1) [p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) f(b_2)] \right]
\end{aligned}$$

This expression depends only on b_1 and b_2 and it turns out that the inner term is exactly the **sum-and-sample** estimator

$$\begin{aligned}
\mathbb{E}_{x\sim p_\theta(x)}[f(x)] &= \sum_x p_\theta(x) \\
&= p_\theta(b_1) f(b_1) + \sum_{x\in D\setminus\{b_1\}} p_\theta(x) f(x) \\
&= p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) \sum_{x\in D\setminus\{b_1\}} \frac{p_\theta(x)}{(1 - p_\theta(b_1))} \\
&= p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) \sum_{x\in D\setminus\{b_1\}} p_\theta(b_2|b_1) f(b_2) \\
&= p_\theta(b_1) f(b_1) + (1 - p_\theta(b_1)) \mathbb{E}_{p_\theta(b_2|b_1)}[f(b_2)]
\end{aligned}$$

Using this, and the fact that $\mathbb{E}_{b_2\sim p_\theta(b_2|b_1)}[\nabla_\theta \log p_\theta(b_1)] = \nabla_\theta \mathbb{E}_{b_1\sim p_\theta(b_1)}[1] = \nabla_\theta 1 = 0$ we find

$$\begin{aligned}
&\mathbb{E}_{S^k\sim p_\theta(S^k)}\left[\sum_{s\in S^k} p(b_1 = s|S^k)\nabla_\theta \log p_\theta(s)\left[p_\theta(s)f(s) + (1-p_\theta(s))\mathbb{E}_{p(b_2|b_1=s,S^k)}[f(b_2)]\right]\right] \\
&= \mathbb{E}_{B^k\sim p_\theta(B^k)}\left[\nabla_\theta \log p_\theta(b_1)[p_\theta(b_1)f(b_1) + (1-p_\theta(b_1))f(b_2)]\right] \\
&= \mathbb{E}_{b_1\sim p_\theta(b_1)}[\nabla_\theta \log p_\theta(b_1)\mathbb{E}_{b_2\sim p_\theta(b_2|b_1)}[p_\theta(b_1)f(b_1) + (1-p_\theta(b_1))f(b_2)]] \\
&= \mathbb{E}_{b_1\sim p_\theta(b_1)}[\nabla_\theta \log p_\theta(b_1)\mathbb{E}_{p_\theta(x)}[f(x)]] \\
&= \mathbb{E}_{b_1\sim p_\theta(b_1)}[\nabla_\theta \log p_\theta(b_1)]\mathbb{E}_{p_\theta(x)}[f(x)] \\
&= 0
\end{aligned}$$

□

Therefore, we can further reduce the variance by introducing the build-in control variable, which arrives at the proposed sampling-without-replacement estimator

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{p_\theta(b)}[f_\theta(b)] = \mathbb{E}_{S^k\sim p(S^k)}&\left[\sum_{s\in S^k}\nabla_\theta p_\theta(s)R(S^k,s)\left(f(s) - \sum_{s\in S^k} p_\theta(s')R^{D\setminus\{s\}}(S^k,s')f(s')\right)\right] \\
&+ \mathbb{E}_{p_\theta(b)}[\nabla_\theta f_\theta(b)].
\end{aligned}$$

2.4.3 Gumbel-Rao estimator

[11] recently proposed a excellent estimator that equips Gumbel-softmax with the Rao-Blackwellization augmentation. Let $D \sim p_\theta$ be a discrete random variable $D \in \{0,1\}^n$ in a one-hot encoding, $\sum_i D_i = 1$, with distribution given by $p_\theta(D) \propto \exp(D^T\theta)$ where $\theta \in \mathbb{R}^n$. The Straight-Through-Gumbel-Softmax (STGS) estimator of $\nabla_\theta \mathbb{E}[f(D)]$ is given by

$$\nabla_{\text{STGS}} := \frac{\partial f(D)}{\partial D}\frac{d\,\text{softmax}_\tau(\theta + G)}{d\theta},$$

where G is a vector of *i.i.d.* $G_i \sim \text{Gumbel}(0,1)$, $\text{softmax}_\tau(x)_i = \exp(x_i/\tau)/\sum_{j=1}^n \exp(x_j/\tau)$. To further reduce variance of STGS, the authors augment it with Rao-Blackwellization, which arrives at the Gumbel-Rao (GR) estimator:

$$\nabla_{\text{GR}} := \frac{\partial f(D)}{\partial D}\mathbb{E}\left[\frac{d\,\text{softmax}_\tau(\theta + G)}{d\theta}\middle| D\right].$$

Note that one can approximate the expectation term using Monte Carlo sampling in which as shown in [19, 20, 7], $\theta + G|D$ can be reparameterized in closed form. In particular, given a realization of D such that $D_i = 1$, $Z(\theta) = \sum_{i=1}^n \exp(\theta_i)$, and $E_j \sim \text{exponential}(1)$ *i.i.d.*, we have

$$\theta_j + G_j = \begin{cases} -\log(E_j) + \log Z(\theta), & \text{if } j = i \\ -\log(\frac{E_j}{\exp(\theta_j)} + \frac{E_i}{Z(\theta)}), & \text{if } j \neq i \end{cases}$$

It is easy to verify that $\nabla_{\text{GR}} = \mathbb{E}[\nabla_{\text{STGS}}|D]$. Since GR is an instance of Rao-Blackwell, it satisfies the same mean with STGS, but has a lower variance. Moreover, GR enjoys a lower mean squared error than STGS, that is, letting $\nabla_\theta := d\mathbb{E}[f(D)]/d\theta$ be the true gradient that we are trying to estimate, we have

$$\mathbb{E}\left[||\nabla_{\text{GR}} - \nabla_\theta||^2\right] \leq \mathbb{E}\left[||\nabla_{\text{STGS}} - \nabla_\theta||^2\right].$$

Proof. By directly applying Jensen's inequality and the law of iterated expectations, we have

$$\begin{aligned}
\mathbb{E}\left[||\nabla_{\text{GR}} - \nabla_\theta||^2\right] &= \mathbb{E}\left[||\mathbb{E}[\nabla_{\text{STGS}}|D] - \nabla_\theta||^2\right] \\
&= \mathbb{E}\left[||\mathbb{E}[\nabla_{\text{STGS}} - \nabla_\theta|D]||^2\right] \\
&\leq \mathbb{E}\left[\mathbb{E}[||\nabla_{\text{STGS}} - \nabla_\theta||^2|D]\right] \\
&= \mathbb{E}\left[||\nabla_{\text{STGS}} - \nabla_\theta||^2\right]
\end{aligned}$$

□

2.5 Leaving One Out (Local Expectation Gradients)

2.5.1 Reparameterization and Marginalization (RAM) estimator

RAM [21] is a nice estimator that combines the log-derivative trick, reparameterization and local expectation gradients well together. In terms of the Bernoulli variables $\{b_1, b_2, \ldots, b_m\}$ with acyclic structures, this kind of estimator shows to be optimal for gradient *w.r.t.* local parameters θ_i f each latent variable b_i. More specifically, recall that a Bernoulli random variable b_i with parameter θ_i could be reparameterized (note that in this sense it only means a deterministic mapping without being differentiable) as

$$b_i = g(\theta_i, \epsilon_i) = \begin{cases} 0, & \text{if } \epsilon_i \geq \theta_i \\ 1, & \text{if } \epsilon_i < \theta_i \end{cases}, \ \epsilon_i \sim \text{Uniform}(0, 1).$$

In fact, reparameterizing all these binary variables, we reduce our objective as $\mathbb{E}_b[f(b)] = \mathbb{E}_\epsilon[f(g(\theta, \epsilon))]$. Note that all of $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$ are independent of each other, where the structure among b_i is expressed in $\theta_i = h(\text{parent}(b_i))$, a deterministic function of its parent variables.

Based on the above, to compute the gradient *w.r.t.* some local distribution parameter θ_i, we have

$$\frac{\partial}{\partial \theta_i}\mathbb{E}_\epsilon[f(g(\theta, \epsilon))] = \mathbb{E}_{\epsilon \setminus i}\left[\frac{\partial}{\partial \theta_i}\mathbb{E}_{\epsilon_i} f(g(\theta, \epsilon))\right].$$

The main idea of RAM is that, instead of estimating the inner gradient term, the authors propose to analytically compute this expectation and evaluate its gradient. In more details, making use of the identity $\mathbb{E}_{\epsilon_i}[f(g(\theta, \epsilon))] = \mathbb{E}_{b_i \sim p(b_i|h(\text{parent}(\text{b}_\text{i})))}[f(b)]$, we have

$$\begin{aligned}\frac{\partial}{\partial \theta_i}\mathbb{E}_{\epsilon_i}[f(g(\theta, \epsilon))] &= \frac{\partial}{\partial \theta_i}\mathbb{E}_{b_i \sim p(b_i|h(\text{parent}(\text{b}_\text{i})))}[f(b)] \\ &= f(b_i = 1, b_{\setminus i})\frac{\partial}{\partial \theta_i}p(b_i = 1|h(\text{parent}(\text{b}_\text{i}))) + f(b_i = 0, b_{\setminus i})\frac{\partial}{\partial \theta_i}p(b_i = 0|h(\text{parent}(\text{b}_\text{i}))),\end{aligned}$$

where $b_{\setminus i} = g(\theta_{\setminus i}, \epsilon_{\setminus i})$. The variance reduction property of RAM, compared with local derivative, follows form the Rao-Blackwelliztion augment. Another main advantage of RAM is enabling the use of common random numbers [22], which effectively reduces the estimation variance. When computing the local gradient *w.r.t.* some parameter θ_i, all $\epsilon_{\setminus i}$ are fixed so that these generated random variables could be reused in the whole step.

2.5.2 Go gradient

GO gradient (General-and-One-sample gradient estimator) [23] addresses the problem that efficiently back-propagating gradients through general distributions (*i.e.*, random variables) remains a bottleneck in modern probabilistic machine learning models. It proposed a general method to reparameterize both discrete and continuous random variables, yielding low-variance and unbiased estimators.

The key theorem [23] presents is the following

Theorem 2.2 (GO gradient)**.** *For an expectation $\mathbb{E}_{q_\phi(y)}[f(y)]$, where $q_\phi(y)$ satisfies*

- *$q_\phi(y)$ is factorized, that is, $q_\phi(y) = \prod_i^n q_\phi(y_i)$;*
- *the corresponding $CDFQ_\phi(y_i)$ is differentiable w.r.t. ϕ;*
- *$\nabla_\phi Q_\phi(y_i)$ is efficient to calculate.*

Then the GO gradient is defined as

$$\nabla_\phi \mathbb{E}_{q_\phi(y)}[f(y)] = \mathbb{E}_{q_\phi(y)}[\mathbb{G}_\phi \mathbb{D}_y[f(y)]], \tag{29}$$

where

$$\mathbb{G}_\phi \triangleq \left[\ldots, -\frac{\nabla_\phi Q_\phi(y_i)}{q_\phi(y_i)}, \ldots\right],$$

and $\mathbb{D}_y[f(y)] = [\ldots, \mathbb{D}_{y_i}[f(y)], \ldots]^T$ with

$$\mathbb{D}_{y_i}[f(y)] \triangleq \begin{cases} \nabla_{y_i} f(y), & \text{if } y_i \text{ iscontinuous} \\ f(y_{-i}, y_i + 1) - f(y), & \text{if } y_i \text{ isdiscrete} \end{cases}$$

Proof. We first show the continuous case. Since

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{q_\phi(y)}[f(y)] &= \nabla_\phi \int_y f(y) q_\phi(y) dy = \nabla_\phi \int_y f(y) \prod_{i=1}^n q_\phi(y_i) dy_i \\
&= \int_y f(y) \nabla_\phi \prod_{i=1}^n q_\phi(y_i) dy_i = \int_y f(y) \sum_{i=1}^n \prod_{j \neq i} q_\phi(y_j) \nabla_\phi q_\phi(y_i) dy_i \\
&= \sum_{i=1}^n \int_{y_{-i}} \prod_{j \neq i} q_\phi(y_j) \int_{y_i} f(y) \nabla_\phi q_\phi(y_i) dy_i = \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[\int_{y_i} f(y) \nabla_\phi q_\phi(y_i) dy_i \right] \\
&= \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[\nabla_\phi \int_{y_i} f(y) q_\phi(y_i) dy_i \right] = \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[\nabla_\phi \int_{y_i} f(y) dQ_\phi(y_i) \right] \\
&= \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[\nabla_\phi \left(f(y) Q_\phi(y_i) |_{-\infty}^{\infty} - \int_{y_i} Q_\phi(y_i) \nabla_{y_i} f(y) dy_i \right) \right] \\
&= \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[- \int_{y_i} \nabla_\phi Q_\phi(y_i) \nabla_{y_i} f(y) dy_i \right] \\
&= \sum_{i=1}^n \mathbb{E}_{q_\phi(y_{-i})} \left[\int_{y_i} q_\phi(y_i) \frac{-\nabla_\phi Q_\phi(y_i)}{q_\phi(y_i)} \nabla_{y_i} f(y) dy_i \right] \\
&= \sum_{i=1}^n \mathbb{E}_{q_\phi(y)} \left[\frac{-\nabla_\phi Q_\phi(y_i)}{q_\phi(y_i)} \nabla_{y_i} f(y) \right] = \mathbb{E}_{q_\phi(y)} \left[\sum_{i=1}^n \frac{-\nabla_\phi Q_\phi(y_i)}{q_\phi(y_i)} \nabla_{y_i} f(y) \right]
\end{aligned}$$

Substituting the above expressions of $\mathbb{G}(\cdot)$ and $\mathbb{D}(\cdot)$ into this formulation, we arrive at the GO gradient of continuous case.

Then we derive the discrete case. Suppose we let N be the supremum of the support of the discrete random variable y, which can be either an integer or ∞. Before deriving the GO gradient, **Abel transformation** is first introduced: Given two sequences $\{a_n\}$ and $\{b_n\}$ with $n \in \{0, \ldots, N\}$, we define $B_0 = b_0$ and $B_n = \sum_{k=0}^n b_k$. Then we have

$$\begin{aligned}
S_N = \sum_{n=0}^N a_n b_n &= a_0 b_0 + \sum_{n=1}^N a_n (B_n - B_{n-1}) \\
&= a_0 B_0 + \sum_{n=1}^N a_n B_n - \sum_{n=0}^{N-1} a_{n+1} B_n \\
&= a_N B_N + \sum_{n=0}^{N-1} a_n B_n - \sum_{n=0}^{N-1} a_{n+1} B_n \\
&= a_N B_N - \sum_{n=0}^{N-1} (a_{n+1} - a_n) B_n.
\end{aligned}$$

By making use of the first several lines of derivations for continuous one and Abel transform (that is let $b_n = \nabla_\phi q_\phi(n)$, $a_n = f(y_{-i}, n)$, then $B_n = \nabla_\phi Q_\phi(n)$), we obtain

$$
\begin{aligned}
\nabla_\phi \mathbb{E}_{q_\phi(y)}[f(y)] &= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(y_{-i})} \left[\sum_{j=0}^{N} f(y_{-i}, j) \nabla_\phi q_\phi(j) \right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(y_{-i})} \left[f(y_{-i}, N) \nabla_\phi Q_\phi(N) - \sum_{j=0}^{N-1} (f(y_{-i}, j+1) - f(y_{-i}, j)) \nabla_\phi Q_\phi(j) \right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(y_{-i})} \left[- \sum_{j=0}^{N-1} (f(y_{-i}, j+1) - f(y_{-i}, j)) \nabla_\phi Q_\phi(j) \right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(y_{-i})} \left[- \sum_{j=0}^{N-1} (f(y_{-i}, j+1) - f(y_{-i}, j)) \nabla_\phi Q_\phi(j) - \nabla_\phi Q_\phi(j) \right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(y_{-i})} \left[\sum_{j=0}^{N} (f(y_{-i}, j+1) - f(y_{-i}, j)) \frac{-\nabla_\phi Q_\phi(j)}{q_\phi(j)} q_\phi(j) \right] \\
&= \mathbb{E}_{q_\phi(y)} \left[\sum_{i=1}^{n} (f(y_{-i}, j+1) - f(y_{-i}, j)) \frac{-\nabla_\phi Q_\phi(j)}{q_\phi(j)} \right] \\
&= \mathbb{E}_{q_\phi(y)} \left[\sum_{i=1}^{n} (f(y_{-i}, j+1) - f(y)) \frac{-\nabla_\phi Q_\phi(j)}{q_\phi(j)} \right]
\end{aligned}
$$

Note that since $Q_\phi(N) = 1$, $\nabla_\phi Q_\phi(N) = 0$. When the computation costs are not essential, we could marginalize the random variable y_i to obtain a better estimation, which is known to reduce variance [24]. □

It can be seen that the GO gradient can be applicable in more general cases. Thus, we may even define a general framework termed Stochastic Back-propagation through random variables, where we perform forward pass and backward pass by taking expectations and reparameterizing gradients, respectively, as introduced in [23]. This is appealing for large-scale stochastic network.

References

[1] Weonyoung Joo, Dongjun Kim, Seungjae Shin, and Il-Chul Moon. Generalized gumbel-softmax gradient estimator for various discrete random variables. *arXiv preprint arXiv:2003.01847*, 2020.

[2] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[5] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979. PMLR, 2014.

[6] Francisco JR Ruiz, Michalis K Titsias, and David M Blei. The generalized reparameterization gradient. *arXiv preprint arXiv:1610.02287*, 2016.

[7] George Tucker, Andriy Mnih, Chris J Maddison, Dieterich Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

[8] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[9] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[10] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

[11] Max B Paulus, Chris J Maddison, and Andreas Krause. Rao-blackwellizing the straight-through gumbel-softmax gradient estimator. *arXiv preprint arXiv:2010.04838*, 2020.

[12] Zhe Dong, Andriy Mnih, and George Tucker. Disarm: An antithetic gradient estimator for binary latent variables. *arXiv preprint arXiv:2006.10680*, 2020.

[13] Mingzhang Yin and Mingyuan Zhou. Arm: Augment-reinforce-merge gradient for stochastic binary networks. *arXiv preprint arXiv:1807.11143*, 2018.

[14] Mingzhang Yin, Yuguang Yue, and Mingyuan Zhou. Arsm: Augment-reinforce-swap-merge estimator for gradient backpropagation through categorical variables. In *International Conference on Machine Learning*, pages 7095–7104. PMLR, 2019.

[15] Runjing Liu, Jeffrey Regier, Nilesh Tripuraneni, Michael Jordan, and Jon Mcauliffe. Rao-blackwellized stochastic gradients for discrete distributions. In *International Conference on Machine Learning*, pages 4023–4031. PMLR, 2019.

[16] Wouter Kool, Herke van Hoof, and Max Welling. Estimating gradients for discrete random variables by sampling without replacement. *arXiv preprint arXiv:2002.06043*, 2020.

[17] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019.

[18] Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196. PMLR, 2016.

[19] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *arXiv preprint arXiv:1411.0030*, 2014.

[20] CA Maddison. Poisson process model for monte carlo. *Perturbation, Optimization, and Statistics*, pages 193–232, 2016.

[21] Seiya Tokui and Issei Sato. Evaluating the variance of likelihood-ratio gradient estimators. In *International Conference on Machine Learning*, pages 3414–3423. PMLR, 2017.

[22] Art B Owen. Monte carlo theory, methods and examples. 2013.

[23] Yulai Cong, Miaoyun Zhao, Ke Bai, and Lawrence Carin. Go gradient for expectation-based objectives. *arXiv preprint arXiv:1901.06020*, 2019.

[24] Michalis Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in neural information processing systems*, pages 2620–2628. Citeseer, 2015.