SUMO: Unbiased Estimation of Log Marginal Probability for Latent Variable Models

Yucen Luo*[1], Alex Beatson[2], Mohammad Norouzi[3], Jun Zhu[1], David Duvenaud[4], Ryan P. Adams[2], Ricky T. Q. Chen*[4]

Tsinghua University[1], Princeton University[2], Google[3], University of Toronto[4] (*Equal contribution)

Contributions

We introduce

- An **unbiased estimator of the log marginal likelihood** and its gradients for latent variable models (LVMs), which
- Allows us to **apply LVMs to new situations** where lower bound estimates are problematic.

Background: Latent Variable Models

Log marginal probability of a latent variable model:

$$\log p_\theta(x) := \log \int_{\mathcal{Z}} p_\theta(x \mid z) p_\theta(z)\, dz = \log \mathbb{E}_{z \sim p_\theta(z)} \left[p_\theta(x \mid z) \right] . \qquad (1)$$

Maximum likelihood estimation requires unbiased estimates of $\nabla_\theta \log p_\theta(x)$, which are not available for LVMs. Instead, lower bound estimators are developed, e.g. importance-weighted evidence lower bound:

$$\mathrm{IWAE}_K(x) := \log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x \mid z_k)\, p_\theta(z_k)}{q(z_k; x)}, \quad z_k \overset{iid}{\sim} q(z; x) . \qquad (2)$$

Background: Russian Roulette Estimator

Estimates infinite series. With K drawn from $p(K)$, RR estimator is:

$$\hat{Y}(K) = \sum_{k=1}^{K} \frac{\Delta_k}{\mathbb{P}(\mathcal{K} \geq k)} \qquad \mathbb{E}_{K \sim p(K)}[\hat{Y}(K)] = \sum_{k=1}^{\infty} \Delta_k . \qquad (3)$$

if (i) $\mathbb{P}(\mathcal{K} \geq k) > 0,\ \forall k > 0$, and (ii) the series converges absolutely, i.e., $\sum_{k=1}^{\infty} |\Delta_k| < \infty$.

SUMO: Russian Roulette to Tighten Lower Bounds

SUMO (**S**tochastically **U**nbiased **M**arginalization **O**bjective):
Let $\Delta_k(x) = \mathrm{IWAE}_{k+1}(x) - \mathrm{IWAE}_k(x)$,

$$\mathrm{SUMO}(x) = \mathrm{IWAE}_1(x) + \sum_{k=1}^{K} \frac{\Delta_k(x)}{\mathbb{P}(\mathcal{K} \geq k)} \text{ where } K \sim p(K),$$

This gives $\boxed{\mathbb{E}\left[\mathrm{SUMO}(x)\right] = \log p_\theta(x)}$ where the expectation is taken over $p(K)$ and $q(z; x)$.
Under some conditions, we also have

$$\mathbb{E}\left[\nabla_\theta \mathrm{SUMO}(x)\right] = \nabla_\theta \mathbb{E}\left[\mathrm{SUMO}(x)\right] = \nabla_\theta \log p_\theta(x)$$

Optimal $p(K)$ for Reducing Variance-Compute Product

Optimal $\mathbb{P}(\mathcal{K} \geq k) \propto 1/k$ in terms of reducing *product of variance & compute*.
We use a tail-modified version:

$$\mathbb{P}(\mathcal{K} \geq k) = \begin{cases} 1/k & \text{if } k < \alpha \\ 1/\alpha \cdot (1 - 0.1)^{k-\alpha} & \text{if } k \geq \alpha \end{cases} \qquad (4)$$

We use $\alpha = 80$, which gives an expectation of ≈ 5.
Trading variance and compute with m**:**

$$\mathrm{SUMO}(x) = \mathrm{IWAE}_m(x) + \sum_{k=m}^{K} \frac{\Delta_k(x)}{\mathbb{P}(\mathcal{K} \geq k)}, \quad K \sim p(K) \qquad (5)$$

Training the Encoder to Reduce Variance

The gradients of SUMO w.r.t. ϕ are in expectation zero.
We optimize $q_\phi(z; x)$ to **reduce the variance** of the SUMO estimator:

$$\nabla_\phi \mathrm{Var}[\mathrm{SUMO}] = \nabla_\phi \mathbb{E}[\mathrm{SUMO}^2] - \cancel{\nabla_\phi (\mathbb{E}[\mathrm{SUMO}])^2} = \mathbb{E}[\nabla_\phi \mathrm{SUMO}^2] . \qquad (6)$$

Select References

Kahn. Use of different monte carlo sampling techniques. (1955)
Nowozin. Debiasing Evidence Approximations: On Importance-weighted Autoencoders ... (ICLR 2018)
Beatson & Adams. Efficient Optimization of Loops and Limits with Randomized Telescoping Sums. (ICML 2019)
Chen *et al.*. Residual Flows for Invertible Generative Modeling. (NeurIPS 2019)

Applications of Unbiased Log Marginal Probability

Minimizing $\log p_\theta(x)$**.** Appears in the "reverse-KL" objective, entropy-regularized reinforcement learning, posterior inference, etc.
Unbiased score function $\nabla_\theta \log p_\theta(x)$**.** Appears in score matching, Hamiltonian Monte Carlo, REINFORCE gradient estimator, etc.

Density Modeling

Table: Test NLL estimated using IWAE(k=5000). For SUMO, k is expected cost.

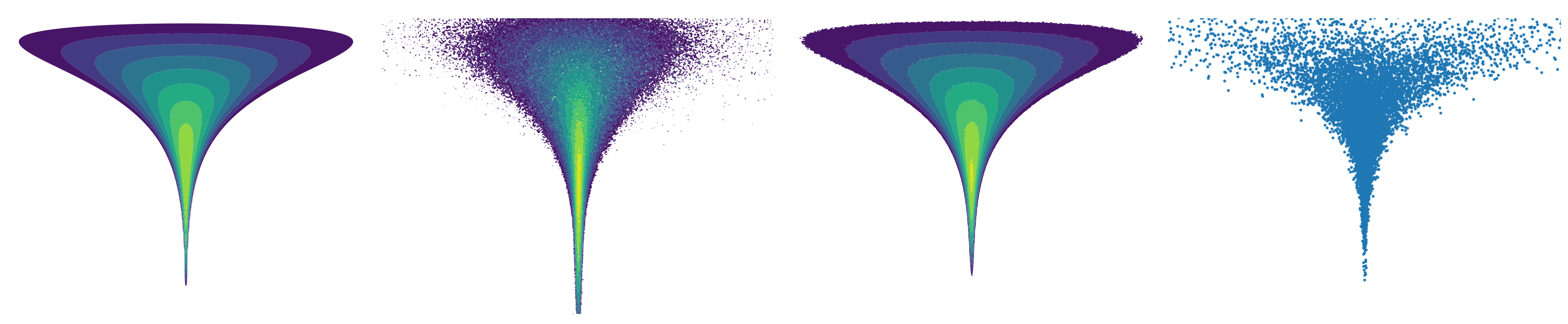
	MNIST			OMNIGLOT		
Training Objective	k=5	k=15	k=50	k=5	k=15	k=50
ELBO [Burda et al., 2016]	86.47	—	86.35	107.62	—	107.80
IWAE [Burda et al., 2016]	85.54	—	84.78	106.12	—	104.67
ELBO (Our impl.)	85.97±0.01	85.99±0.05	85.88±0.07	106.79±0.08	106.98±0.19	106.84±0.13
IWAE (Our impl.)	85.28±0.01	84.89±0.03	84.50±0.02	104.96±0.04	104.53±0.05	**103.99±0.12**
JVI [Nowozin, 2018] (Our impl.)	—	—	84.75±0.03	—	—	104.08±0.11
SUMO	**85.09±0.01**	**84.71±0.02**	**84.40±0.03**	**104.85±0.04**	**104.29±0.12**	**103.79±0.14**

Posterior Inference

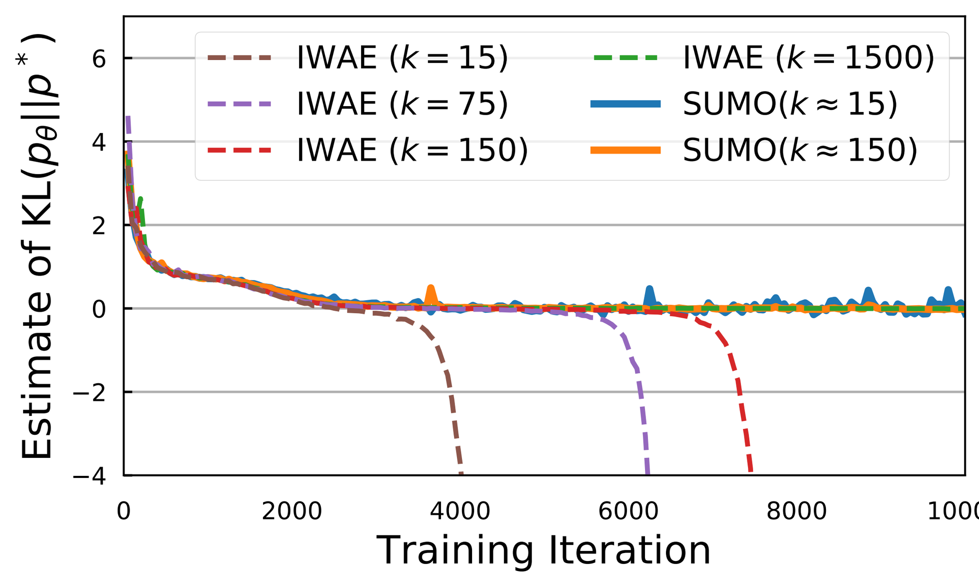
Presence of an entropy maximization term, effectively a minimization of $\log p_\theta(x)$:

$$\min_\theta D_{\mathrm{KL}}(p_\theta(x) || p^*(x)) = \min_\theta \mathbb{E}_{x \sim p_\theta(x)}[\log p_\theta(x) - \log p^*(x)] \qquad (7)$$

Modifying IWAE: $\min_{p(x,z)} \max_{q(z;x)} \mathbb{E}_{x \sim p(x)}[\mathrm{IWAE}_K(x) - \log p^*(x)]$



Target log prob — Training w/ IWAE — Training w/ SUMO — Model samples



IWAE is unstable and requires very large k to match performance of SUMO.

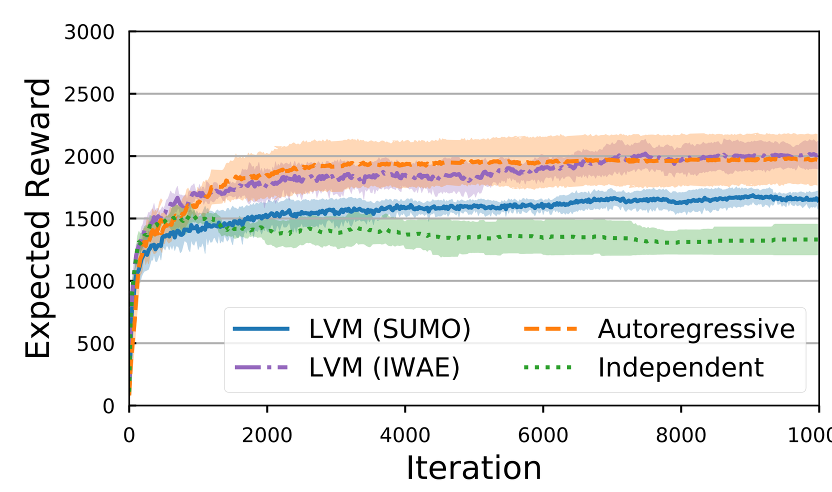
Combinatorial Optimization

Quadratic pseudo-Boolean optimization (QPBO) maximizes

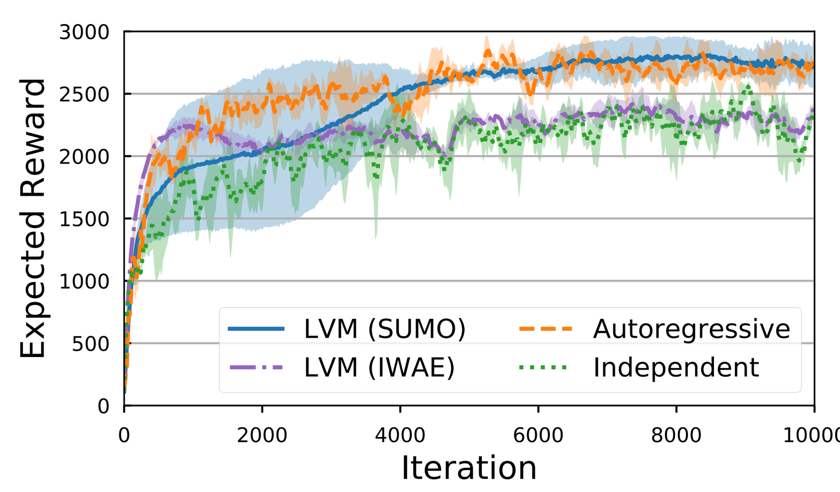
$$R(x) = \sum_{i=1} w_i(x_i) + \sum_{i<j} w_{ij}(x_i, x_j) \qquad (8)$$

where $\{x_i\}_{i=1}^{d} \in \{0, 1\}$ are binary variables. Baseline policy models:

$$p_{\mathrm{LVM}}(x) := \int \prod_{i=1}^{d} p_\theta(x_i|z) p(z) dz, \quad p_{\mathrm{Autoreg}}(x) := \prod_{i=1}^{d} p(x_i|x_{<i}), \quad p_{\mathrm{Indep}}(x) := \prod_{i=1}^{d} p(x_i)$$



Without entropy regularization.



With entropy regularization.

- SUMO works well with entropy regularization.
- Latent variable policies allow fast exploration while being highly expressive.
- Latent variable policy is **20x faster** than autoregressive for $d = 500$.