SFESS: Score Function Estimators for *k*-Subset Sampling

Klas Wijk, Ricardo Vinuesa, Hossein Azizpour

Are score function estimators a viable option for learning with *k*-subset sampling?

Keywords

- *k*-subset sampling, *k*-hot
- Gradient estimation, score function estimators
- Feature selection, discrete representation learning

Problem

Differentiable *k*-Subset Sampling

- Unlike e.g., Normal distributions, discrete distributions cannot be rewritten using the reparametrization trick, which complicates differentiable optimization.
- Current methods for sampling *k*-subsets, or *k*-hot vectors, are based on either approximate pathwise gradients or relaxed sampling. We investigate how score function estimators compare to these approaches.

$$\nabla_{\theta}\mathbb{E}_{p_{\theta,k}(z)}[f(z)]$$

Why score function estimators?

Unlike pathwise gradient estimators, score function estimators do not assume that the downstream function is differentiable and allow computing *unbiased* gradient estimates.

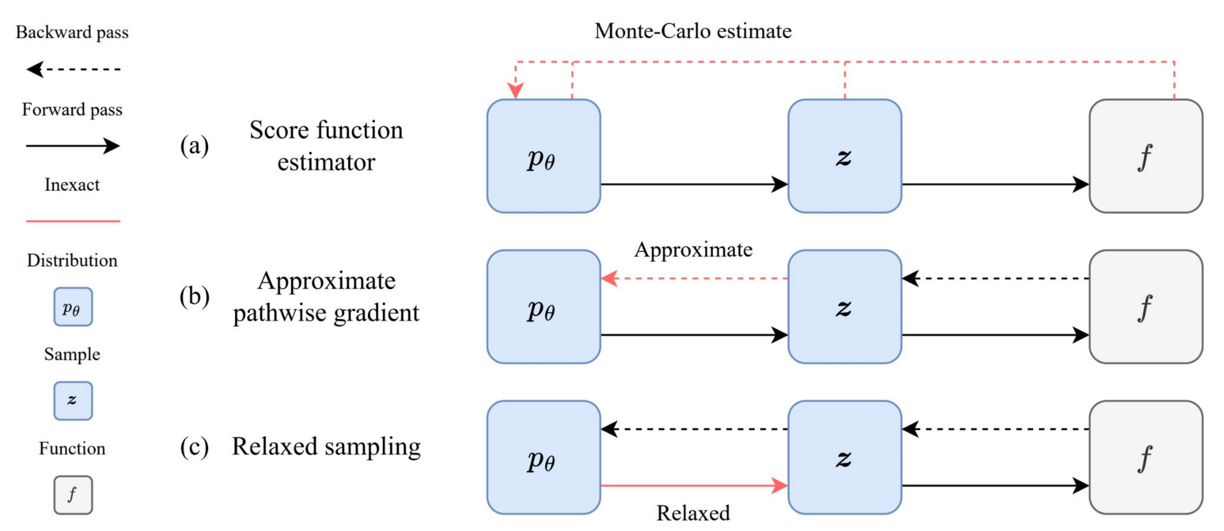

Figure 1: Three families of methods for gradient estimation: a) score function estimators compute a Monte-Carlo estimate of the gradient, b) approximate pathwise gradients compute a gradient estimate using the downstream function's gradient, c) relaxed sampling circumvents the problem by using relaxed samples.

Method

Computing the score function

The score function

$$\nabla_{\theta} \log p_{\theta,k}(z) = \sum_{i=1}^{n} \underbrace{\nabla_{\theta} \log p_{\theta}(b_i)}_{\text{Bernoulli}} - \underbrace{\nabla_{\theta} \log p_{\theta}\left(\sum_{i=1}^{n} b_i = k\right)}_{\text{Poisson binomial}}$$

The first term is a Bernoulli score function and is easy to compute. The second term looks trickier. It is the score function of a Poisson binomial distribution. A naive computation would iterate over all possible subsets. It turns out that it can be computed efficiently using a fast Fourier transform instead.

Variance reduction

Vanilla score function estimators (REINFORCE) suffer from high variance. There are many options for variance reduction, like using a moving average or learning the baseline. We use a simple multi-sample control variate which only assumes that we can draw and evaluate multiple samples.

$$\nabla_{\theta}\mathbb{E}_{p_{\theta,k}(z)}[f(z)] \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_{\theta} \log p_{\theta,k}(z^{(j)}) \left(f(z^{(j)}) - \frac{1}{N-1}\sum_{i \neq j} f(z^{(j)}) \right)$$

For certain downstream functions, drawing multiple samples for variance reduction could be impractical. In our experiments, however, drawing 32 samples had little effect on the wall-clock time.

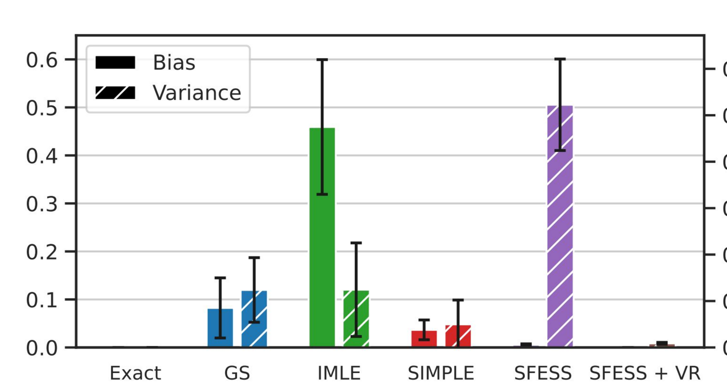

Figure 3: Bias and variance of different estimators in a toy experiment.

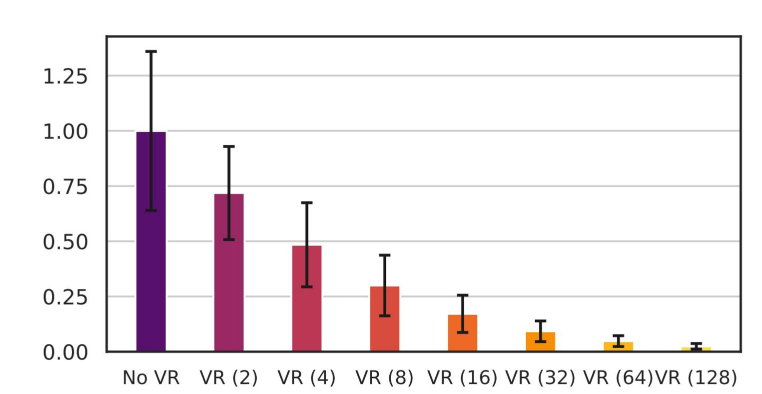

Figure 4: Variance reduction as the number of samples increases.

Experiments

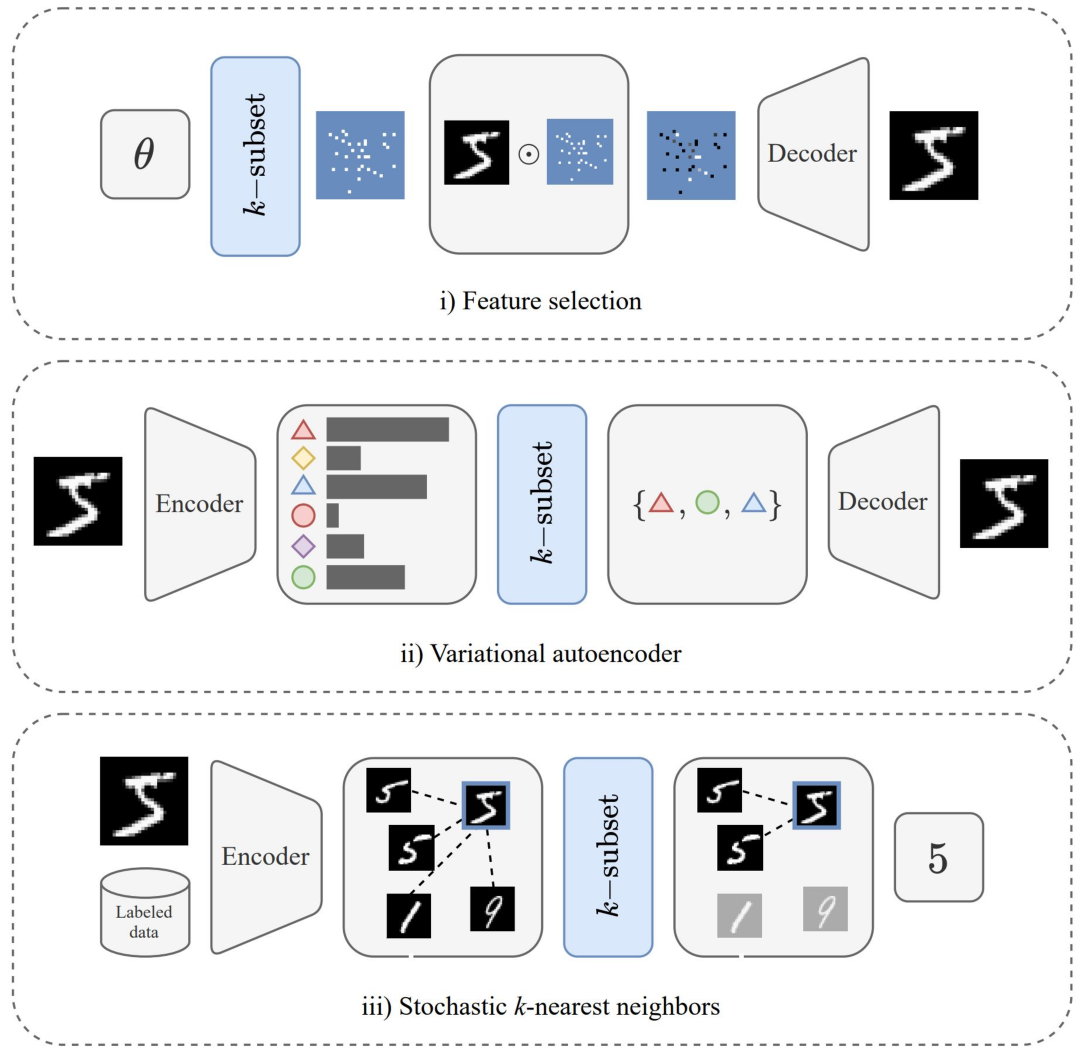

Figure 3: Three experimental settings: i) feature selection uses k-subset sampling to subsample the input, ii) variational autoencoders uses k-subset sampling in to learn a discrete representation of the input, and iii) k-nearest neighbors uses k-subset sampling to select the nearest (labeled) neighbors of an unlabeled example and classify it using them.

SFESS performs similarly to state-of-the-art methods across the three experimental settings while providing two key advantages: unbiased gradient estimates and not requiring a differentiable downstream function. The main limitation is requiring multiple samples for variance reduction.

Future Work

- Single-sample variance reduction
- Combining score function and pathwise gradient estimators
- Applications with non-differentiable downstream functions

Paper   Code