Differentiable determinantal Point Processes[1]

GAURAV ARYA

Setup. A discrete determinantal point process (DPP) $\mathcal{I}$, induced by an $n \times n$ symmetric matrix $0 \preccurlyeq K \preccurlyeq I$, is distributed over subsets $I \in 2^{[n]}$ as

$$\mathbb{P}(\mathcal{I} = \mathcal{I}) = \det(L_{\mathcal{I},\mathcal{I}})/\det(L + I), \tag{1}$$

where $L = K(I + K)^{-1}$.

In this article, we study differentiable DPPs. Here, $K : \mathbb{R} \to \mathbb{R}^{n\times n}$ becomes a function of a parameter $p \in \mathbb{R}$, and we are interested in the derivative w.r.t. p of the expectation of a function $f : \{0,1\}^n \to \mathbb{R}$ over the DPP. Precisely, we want

$$\partial_p \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})]. \tag{2}$$

DPP Sampling. To compute $\mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})]$ for large n, we need to sample. We use a Gaussian-elimination-inspired algorithm. Initialize $\mathbf{A}^{(0)} = K(p)$ and $\bar{\mathcal{I}}^{(0)} = (0, 0, \ldots, 0)$. Then, for $i = 1, \ldots, n$, iterate the following kernel for $\left(\mathcal{I}^{(i-1)}, A^{(i-1)}\right) \to \left(\mathcal{I}^{(i)}, A^{(i)}\right)$:

$$\bar{\mathcal{I}}^{(i)}_{1:i-1} = \bar{\mathcal{I}}^{(i-1)}_{1:i-1} \tag{3}$$

$$\bar{\mathcal{I}}^{(i)}_{i} \sim \mathrm{Ber}\left(\mathbf{A}^{(i)}_{i,i}\right), \tag{4}$$

$$\mathbf{A}^{(i)}_{i,i} = \mathbf{A}^{(i-1)}_{i,i} - \left(1 - \bar{\mathcal{I}}^{(i)}_{i}\right), \tag{5}$$

$$\mathbf{A}^{(i)}_{i+1:n,i+1:n} = \mathbf{A}^{(i-1)}_{i+1:n,i+1:n} - \frac{\mathbf{A}^{(i-1)}_{i+1:n,i} \cdot \mathbf{A}^{(i-1)}_{i,i+1:n}}{\mathbf{A}^{(i)}_{i,i}}. \tag{6}$$

Finally, $\mathcal{I} = \{i \in [n] : \bar{\mathcal{I}}^{(n)} = 1\}$ satisfies $\mathcal{I} \sim \mathrm{DPP}(K(p))$.

Differentiable DPP Sampling. We now present algorithms for differentiable DPP sampling.

Score method. We can write

$$\partial_p \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})] = \sum_{\mathcal{I}\in 2^{[n]}} f(\mathcal{I})\partial_p \mathbb{P}(\mathcal{I}) \tag{7}$$

$$= \sum_{\mathcal{I}\in 2^{[n]}} f(\mathcal{I})\partial_p \log \mathbb{P}(\mathcal{I}) \cdot \mathbb{P}(\mathcal{I}), \tag{8}$$

$$= \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})\partial_p \log \mathbb{P}(\mathcal{I})]. \tag{9}$$

The score method samples the expectand in the RHS above.

Score method with average baseline. We can replace f with $f_b : \mathcal{I} \mapsto f(\mathcal{I}) - b$ for any fixed $b \in \mathbb{R}$ and leave $\partial_p \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})]$ invariant, so:

$$\partial_p \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})] \tag{10}$$

$$= \partial_p \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f_b(\mathcal{I})] \tag{11}$$

$$= \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[(f(\mathcal{I}) - b)\partial_p \log \mathbb{P}(\mathcal{I})]. \tag{12}$$

This method samples the expectand in the RHS above with $b = \mathbb{E}_{\mathcal{I}\sim\mathrm{DPP}(K(p))}[f(\mathcal{I})]$.

Stochastic derivative / MVD method with pruning. This method considers the possibly of each Bernoulli step (4) flipping in outcome. The (random) derivative $\partial_p \mathbf{A}^{(j-1)}_{j,j}$ of the probability in the jth Bernoulli sample is used to form a weight $\mathbf{w}_{(j)}$. This weight is associated with an alternative outcome $\bar{\mathcal{I}}^{(j)}_{(j)} = 1 - \bar{\mathcal{I}}^{(j)}$ of that Bernoulli sample. After substituting this alternative outcome, the rest of the algorithm is run unchanged to produce alternatives $\bar{\mathcal{I}}^{(j)}_{(j)}, \bar{\mathcal{I}}^{(j+1)}_{(j)}, \ldots, \bar{\mathcal{I}}^{(n)}_{(j)}$, and ultimately an alternative DPP sample $\mathcal{I}_{(j)}$ derived from the flipping of the jth step. The derivative estimate is then given as

$$\sum_{j=1}^{n} \mathbf{w}_{(j)} \left(f(\mathcal{I}_{(j)}) - f(\mathcal{I}_{(j)})\right). \tag{13}$$

Since $\mathcal{I}_{(j)}$ and $\mathcal{I}_{(j)}$ are obtained by running the same DPP sampler with slightly different inputs for the random steps, their joint distribution may be determined via a coupling of these random steps. We choose a common random number coupling for the Bernoulli step [2].

The estimator (13) requires n additional runs of the program, increasing asymptotic computational cost. We therefore employ a pruning strategy, where an index $\mathbf{j}$ of the sum is randomly subsampled with $\mathbb{P}(\mathbf{j} = j) \propto \left|w_{(j)}\right|$, giving us an unbiased derivative estimate

$$\left(f(\mathcal{I}_{(\mathbf{j})}) - f(\mathcal{I}_{(\mathbf{j})})\right) \cdot \sum_{j=1}^{n} \mathbf{w}_{(j)}, \tag{14}$$

of computational cost equal to that of the original sampler.

[1] Class project for 18.338, draft work.

We consider two ways to choose $w_{(i)}$: the stochastic derivative method [1] based on smoothed perturbation analysis [2], and the measure-valued derivative (MVD) method [3]. These strategies, together with pruning, are implemented automatically using the package `StochasticAD.jl` [1], applied to DPP sampling code in Julia.

Variance comparison. Consider the following toy setting:

$$K_0 = \frac{1}{2}\frac{GG'}{GG' + I}, \tag{15}$$

$$K(p) = (1+p) * K_0, \tag{16}$$

$$f(\mathcal{I}) = |\mathcal{I}|, \tag{17}$$

where G is an $n \times n$ real Gaussian random matrix, and we differentiate at $p = 0$. We are interested in comparing the variance of our DPP derivative estimation algorithms as a function of n.

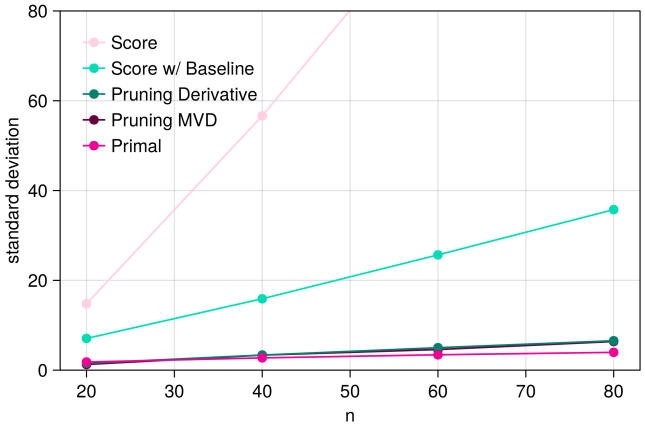

Figure 1: Differentiable DPP sampling variance comparison: **original setting**.

As can be seen from Figure 1, the pruning-based approaches demonstrate impressive variance asymptotics compared to the score method.

However, as scientists, what is perhaps most interesting here is that we have actually stumbled upon somewhat of a Goldilocks scenario. Suppose we switch out our function f to be a dot product with a fixed vector u (generated with uniform random elements over $[0, 1]$):

$$f(\mathcal{I}) = \sum_{i=1}^{n} u_i \mathbf{1}_{i \in \mathcal{I}}. \tag{18}$$

Let's check in with our variance comparison plot.

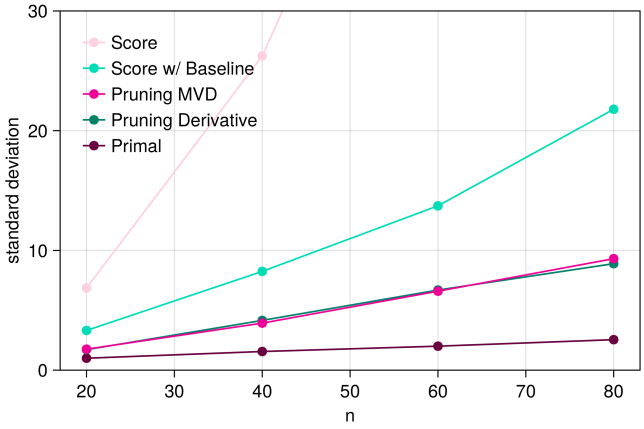

Figure 2: Differentiable DPP sampling variance comparison: **with a random** f.

The advantage of our pruning-based approaches, compared to the score method, has surely diminished.

Why? But of course! Recall our expression (13). Our pruning step picked one of these terms at random. With our original choice $f(\mathcal{I}) = |\mathcal{I}|$, so long as $\mathcal{J}_{(i)}$ and $\mathcal{I}_{(i)}$ tend to have similar differences in cardinality across the choices for i, our pruning step will not have much of an effect on variance. But with our new choice of f, it matters *where* they differ, and this will surely vary over the candidate perturbations.

Mystery solved. If we keep our f the same, then, we should be in good shape. With this in mind... why is there a scaling factor of $\frac{1}{2}$ in our expression (15) for K_0 again? It just happened to be there because I was being cautious and avoiding eigenvalues close to 1, in case they cause trouble anywhere. (Also, now is as good a time as any to divulge that the L-form we introduced at the very beginning requires accomodating infinite eigenvalues to handle K-eigenvalues of 1.) Let's get rid of the scaling factor (but *keep* $f(\mathcal{I}) = |\mathcal{I}|$), and check in again with our variance comparison plot.

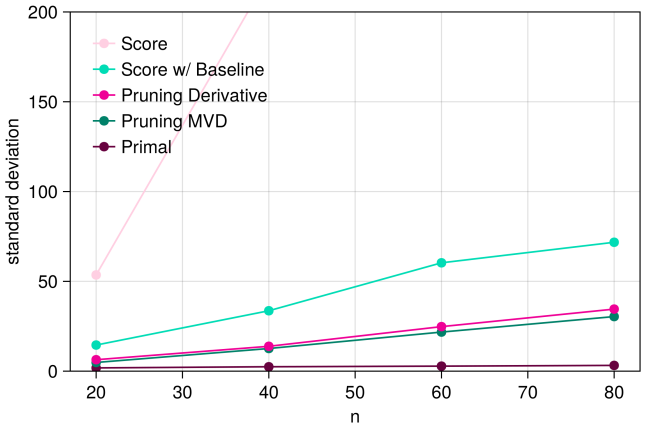

Figure 3: Differentiable DPP sampling variance comparison: **with no scaling factor** $1/2$.

Our Goldilocks scenario is living up to its name! The pruning-based methods have again lost their advantage compared to the score method.

To understand why, we need to do a more carefuly analysis of our statement "so long as $\mathcal{J}_{(i)}$ and $\mathcal{I}_{(i)}$ tend to have similar differences in cardinality across the choices for i".

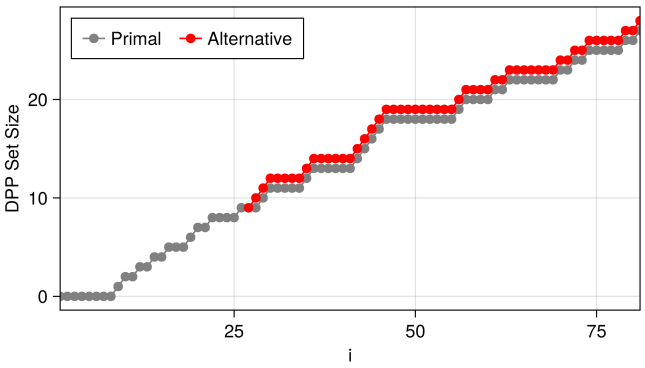

Figure 4: Sample of cardinalities of $\mathcal{J}^{(i)}_{(\mathbf{j})}$ and $\mathcal{I}^{(i)}_{(\mathbf{j})}$ as a function of i: **original setting**, $n = 80$.

Figure 4 depicts a "typical" perturbation chosen by pruning, in the original setting. In particular, at the end, it tends to hold with few exceptions that $|\mathcal{J}_{(\mathbf{j})}| = |\mathcal{I}_{(\mathbf{j})}| + 1$. In contrast, Figure 5 below depicts two typical scenarios in the setting with no scaling factor.

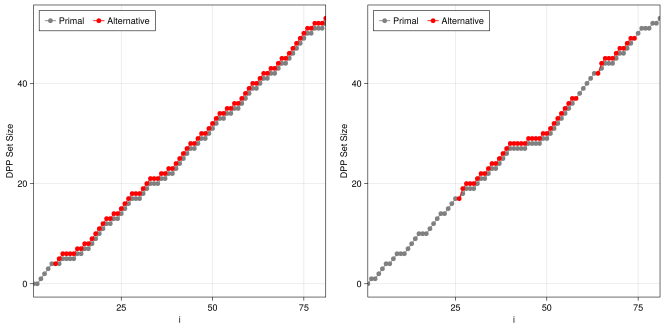

Figure 5: Two samples of cardinalities of $\mathcal{J}^{(i)}_{(\mathbf{j})}$ and $\mathcal{I}^{(i)}_{(\mathbf{j})}$ as a function of i: **with no scaling factor** $1/2$, $n = 80$.

As we can see, in the setting without the scaling factor $1/2$, there are now two typical scenarios: the pruned perturbation ends up with cardinality one greater (left), and the pruned perturbation ends up with equal cardinality (right). It was a stroke of luck that somehow, the former scenarios was much more likely in the original scenario we picked. This explains the increased variance of our pruning approaches in Figure 3.

Conclusion. Our analysis here shows the potential for asymptotically improved variance of differentiable DPP sampling algorithms. However, to reap the benefits in practical settings with various choices of f, K, etc., it should now be clear to the reader that the simple pruning strategy employed here will not be enough. The author is excited to address this in future work.

Specific to the DPP application, it would be nice to apply the derivative estimation strategy here to an end-to-end stochastic optimization example on a DPP of practical interest. Another interesting question is to investigate more carefully the properties of the coupling here, and compare it to others, such as one formed using a QR-based DPP sampling algorithm.

References

[1] G. Arya, M. Schauer, F. Schäfer, and C. Rackauckas. Automatic differentiation of programs with discrete randomness. *Advances in Neural Information Processing Systems*, 35:10435–10447, 2022.

[2] P. Glasserman and D. D. Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

[3] B. Heidergott and F. Vázquez-Abad. Measure-valued differentiation for markov chains. *Journal of Optimization Theory and Applications*, 136 (2):187–209, 2008.