Diversified Mini-Batch Sampling using Repulsive Point Processes

Cheng Zhang
Disney Research
Pittsburgh, USA

Cengiz Öztireli
Disney Research
Zurich, Switzerland

Stephan Mandt
Disney Research
Pittsburgh, USA

{cheng.zhang, cengiz.oztireli, stephan.mandt}@disneyresearch.com

Abstract

Non-uniform mini-batch sampling may be beneficial in stochastic variational inference (SVI) and more generally in stochastic gradient descent (SGD). In particular, sampling data points with repulsive interactions, i.e., suppressing the probability of similar data points in the same mini-batch, was shown to reduce the stochastic gradient noise, leading to faster convergence. When the data set is furthermore imbalanced, this procedure may have the additional benefit of sampling more uniformly in data space. In this work, we explore the benefits of using point processes with repulsive interactions for mini-batch selection. We thereby generalize our earlier work on using determinantal point processes for this task [14]. We generalize the proof of variance reduction and show that all we need is a point process with repulsive correlations. Furthermore, we investigate the popular Poisson disk sampling approach which is widely used in computer graphics, but not well known in machine learning. We finally show empirically that Poisson disk sampling achieves similar performance as determinantal point processes at a much more favorable scaling with the size of the dataset and the mini-batch size.

1 Introduction

Stochastic gradient descent (SGD) [2] is key to modern scalable machine learning. SGD plays a crucial role in stochastic variational inference (SVI) [4], which scales up approximate inference [13] to massive data. A key factor for the speed of convergence of SGD algorithms is the stochastic gradient variance, and variance reduction for SGD and SVI is an active research topic.

Mini-batch diversification methods [15, 3, 14] allow reducing the variance of the stochastic gradients, leading to faster convergence. These sampling schemes suppress the probability of data points with similar features to co-occur in the same mini-batch. Such an approach efficiently decorrelates the samples, leading to a better gradient noise cancellation. This type of sampling can be operated in two modes: one mode preserves the marginal sampling probabilities of each data point; thus the original objective function is not modified, and every data point still has an equal contribution to the loss. The other mode alters the marginal probabilities of the data, and hence the objective gets biased. The latter case can be beneficial when the data set is imbalanced, since data points which are very dissimilar from the rest get up-weighted. This may lead to a more uniform sampling scheme in data space.

In our previous work [14], we proposed to use determinantal point processes (DPP) [7, 6] for diversified mini-batch sampling. In this paper, we generalize the proof of variance reduction from DPPs [14] to general stochastic point processes with negative two-point correlations. We focus our experimental work on Poisson disk processes, which are widely used in computer graphics. These point process can be simulated using generalized dart throwing algorithms [12, 8]. While mini-batch sampling with k-DPPs scales as $\mathcal{O}(Nk^3)$ per iteration (where N is the size of the data and k the mini-batch), Poisson disk processes achieve similar effects with a scaling of $\mathcal{O}(k^2)$.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Our paper is structured as follows. We first present our theoretical contribution on using stochastic point processes for mini-batch sampling in Section 2. We then explain Poisson disk sampling with dart throwing in Section 3. Finally, we demonstrate on several models that Poisson disk sampling with a dart throwing algorithm can achieve similar performance as mini-batch diversification with DPPs.

2 Stochastic Point Processes for Mini-batch Diversification

Here, we introduce our method of using stochastic point processes for mini-batch diversification, generalizing [14] from k-DPPs to arbitrary point processes with negative two-point correlations.

Point processes Point processes describe generating processes underlying observed point distributions [10, 5]. We will consider point processes in $\mathbb{R}^d$ for the discussion in this paper. A point process $\mathcal{P}$ can be defined by considering the joint probabilities of finding points generated by this process in infinitesimal volumes.

One way to express these probabilities is via *product densities*. Let $\mathbf{x}_i$ denote some arbitrary points in $\mathbb{R}^d$, and $\mathcal{B}_i$ infinitesimal spheres centered at these points with volumes $d\mathbf{x}_i = |\mathcal{B}_i|$. Then the n^{th} order product density $\varrho^{(n)}$ is defined by $p(\mathbf{x}_1, \cdots, \mathbf{x}_n) = \varrho^{(n)}(\mathbf{x}_1, \cdots, \mathbf{x}_n)d\mathbf{x}_1 \cdots d\mathbf{x}_n$, where $p(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ is the joint probability of having a point of the point process $\mathcal{P}$ in each of the infinitesimal spheres $\mathcal{B}_i$. For example, DPP defines the probability to sample a subset as to be positive propositional to the determinant of the similarity kernel matrix.

A point process $\mathcal{P}$ can be defined to consist of finite or infinite number of points. For a stationary point process, the first order product density, also called intensity, becomes a constant $\lambda(\mathbf{x}) := \varrho^{(1)}(\mathbf{x}) = \lambda$, and second order product density $\varrho(\mathbf{x}, \mathbf{y}) := \varrho^{(2)}(\mathbf{x}, \mathbf{y})$ becomes a function of the difference between point locations $\varrho(\mathbf{x} - \mathbf{y})$, commonly written in terms of the intensity normalized *pair correlation function* (PCF) g [12] as $\varrho(\mathbf{x} - \mathbf{y}) = \lambda^2 g(\mathbf{x} - \mathbf{y})$. Isotropy further simplifies it to $\varrho(||\mathbf{x} - \mathbf{y}||)$.

Mini-Batch Sampling with Stochastic Point Processes Recently, Zhang et al. [14] investigated how to utilize a particular type of point process, DPP, for mini-batch diversification. Here, we generalize the theoretical results to arbitrary point processes, and elaborate on how the resulting formulations can be utilized for SVI and SGD. This opens the door to a vast literature on point processes, and efficient algorithms for sampling.

As explained in [14], diversified mini-batch sampling algorithms optimize the *diversified risk*:

$$\hat{J}_{\mathrm{data}'}(\theta) = \sum_i w_i l(\mathbf{x}_i, \theta). \tag{1}$$

Each weight w_i is the marginal probability that the data point $\mathbf{x}_i$ is sampled under the point process. If $w_i = 1$, the algorithm optimizes the original empirical risk.

Stochastic gradient descent (SGD) based algorithms work by estimating $\nabla J_{\mathrm{data}'}(\theta)$, starting from the estimator in Equation 1. The resulting gradient can be written as $\hat{\mathbf{K}}(\theta) = \sum_i w_i \mathbf{k}(\mathbf{x}_i, \theta)$, where we defined $\hat{\mathbf{K}}(\theta)$ as the estimator for $\mathbf{K}(\theta) = \nabla J_{\mathrm{data}'}(\theta)$, and $\mathbf{k}(\mathbf{x}_i, \theta) := \nabla l(\mathbf{x}_i, \theta)$. This estimation is typically done by taking batches of data, which are sampled stochastically from $p_{\mathrm{data}'}(\mathbf{x})$. Each batch, i.e. set of data points, can thus be considered as an instance of an underlying point process $\mathcal{P}$ that depends on $p_{\mathrm{data}'}(\mathbf{x})$, as well as the sampling algorithm used to sample from $p_{\mathrm{data}'}(\mathbf{x})$. Our goal is to design sampling algorithms for improved learning performance.

We can compute the variance of the gradient estimate $\hat{\mathbf{K}}$ using the theory of stochastic point processes as we show in the appendix.

$$\begin{aligned}\mathrm{var}_{\mathcal{P}}(\hat{\mathbf{K}}) = &\int_{\mathcal{V}\times\mathcal{V}} w(\mathbf{x})\lambda(\mathbf{x})w(\mathbf{y})\lambda(\mathbf{y})\mathbf{k}(\mathbf{x},\theta)^T\mathbf{k}(\mathbf{y},\theta)\left[\frac{\varrho(\mathbf{x},\mathbf{y})}{\lambda(\mathbf{x})\lambda(\mathbf{y})} - 1\right]d\mathbf{x}d\mathbf{y} \\ &+ \int_{\mathcal{V}} w^2(\mathbf{x})\|\mathbf{k}(\mathbf{x},\theta)\|^2\lambda(\mathbf{x})d\mathbf{x},\end{aligned} \tag{2}$$

where we assume w_i's are sampled from $w(\mathbf{x})$. This equation holds for general point processes with possibly complex interactions among sample points. For a sampler that randomly samples points from a distribution, e.g. $p_{\mathrm{data}'}(\mathbf{x})$, $\varrho(\mathbf{x}, \mathbf{y}) = \lambda(\mathbf{x})\lambda(\mathbf{y})$, and hence the first term above vanishes. The second term is thus the variance for the case of a random sampler. The reduction in variance due to the sampler, as compared to random sampling, is thus given by the first term. The λ is non-negative by definition, and we can assume the same for w. Thus, for reducing variance, we require that for

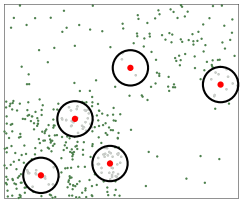
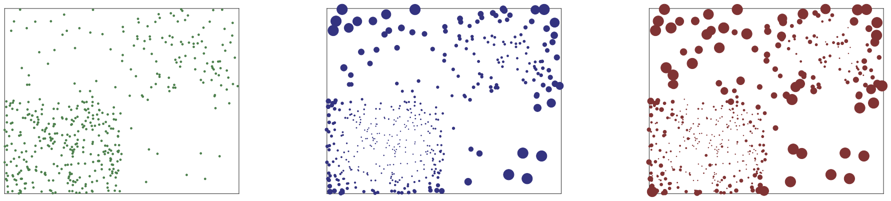


Figure 1: Demonstration of Poisson disk sampling with the dart throwing algorithm. The data marked in red are the samples that are collected already. The black circles of a certain radius r mark regions claimed by the red points. For the next iteration, if the newly sampled point falls in any of the circles (points colored in gray), this point will be rejected. If the point is outside the circles (points shaded in green), we add it to the collation of samples.



(a) data (b) DPP (c) Poisson Disk

Figure 2: Example samplings. Panel (a) demonstrates a dataset that is imbalanced, panel (b) shows the marginal probability using DPP, and panel (c) shows the simulated marginal probability with dart throwing.

some $\mathbf{x}$, $\mathbf{k}(\mathbf{x}, \theta)^T \mathbf{k}(\mathbf{y}, \theta) \left[\frac{\varrho(\mathbf{x},\mathbf{y})}{\lambda(\mathbf{x})\lambda(\mathbf{y})} - 1 \right] < 0$. This is a generalization of Theorem 3 in [14], for a sampler with arbitrary properties.

3 Poisson Disk Sampling with Dart Throwing

In this section, we show how to use Poisson disk sampling to sample diversified mini-batches rather than using DPP as in [14]. In general, we can simulate all stochastic point processes with generalized dart throwing and statistics fitting as in [12]. Poisson disk sampling, typically implemented with the efficient dart throwing algorithm [8], is a particular type of repulsive point process that provides point arrangements very similar to those of DPP, albeit much more efficiently.

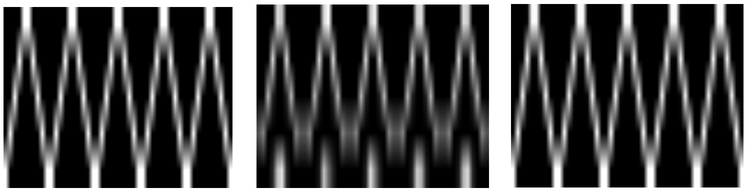
This process dictates that the smallest distance between each pair of sample points should be at least r with respect to some distance measure D. In this way, it can sample data to cover the space as uniform as possible. Thus, it downsamples dense regions and upsamples sparse areas. At each iteration, the basic dart throwing algorithm for Poisson disk sampling works as follows: 1) randomly sample a data point; 2) if it is within a distance r of any already accepted data point, reject; otherwise, accept the new point. This procedure is similar to k-DPP in [14]. We can also specify the maximum sampling size k with dart throwing. This means that we terminate the algorithm when k are considered. Thus, the computational complexity is $\mathcal{O}(k^2)$ using Poisson disk sampling with dart throwing[1]. On the other hand, the computational complexity using DPP is $\mathcal{O}(Nk^3)$, where N is the number of data points in the dataset. Note that the graphics community uses such sampling algorithms for sampling in continuous spaces, but here we use dart throwing for sampling from discrete data. Figure 1 demonstrates the resulting algorithm of using dart throwing with discrete data.

We use a toy example shown in Figure 2 to compare DPP and Poisson disk sampling. In this figure, Panel (a) shows an example of an imbalanced dataset as in [14]; Panel (b) shows the marginal probability of each point being sampled with DPP. The size of the marker is proportional to its marginal probability. Similarly, panel (c) shows the simulated marginal probability of each data point being sampled using Poisson Disk sampling, with 1000 sampling trials. Thus, sizes of markers are proportional to the sampling frequency of points. We can see that Poisson disk sampling generates samplings similar to those with DPP. They both balance the datasets in a point-wise manner. We show more examples on sampled mini-batches in the appendix.

4 Experiments

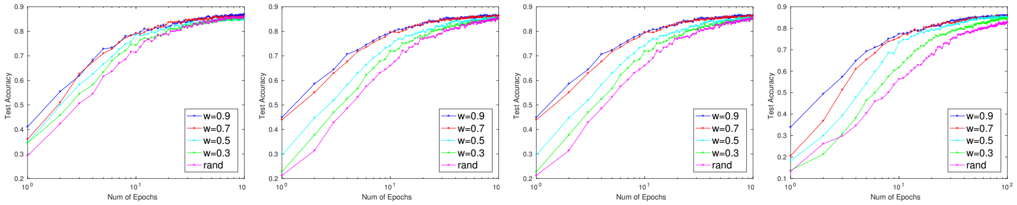
In this section, we show that using Poisson disk sampling achieve the same performance improvement as with DPP in [14], but significantly faster. We perform two experiments in the same setting as in

[1] In practice, the number of accepted points can be smaller than the number of considered points in the dataset, as some of them will be eliminated by not satisfying the distance criteria.



(a) GroundTruth (b) SVI (c) Poisson disk SVI

Figure 3: Parameter recovering using Topic Model. Panel (a) shows the ground truth which is the target to recover; Panel (b) shows the parameter estimated by traditional SVI, where the rare topic (last one) is not recovered, while Panel (c) shows the result with Poisson disk sampling SVI. All the latent parameters are recovered.



(a) K=50 (b) K=80 (c) K=102 (d) K=150

Figure 4: Oxford Flower Classification with Softmax. SGD with Poisson Disk sampling converges faster than traditional SGD (rand). The w is the weight balancing the visual feature and label information as in [14].

[14]. The first one is using Latent Dirichlet Allocation (LDA) [1] with SVI. The analysis is using synthetic data as in [14]. Then, we use Softmax to perform fine-grained classification with Oxford Flower dataset [11]. For both experiments, we use the sampling distance measure $D_{ij} = 1 - S_{ij}$, where each entry of S_{ij} is the cosine similarity between data points with indices i and j.

LDA with Synthetic Data This experiment is to recover the parameters of Latent Dirichlet allocation (LDA) [1]. We generate a synthetic dataset with the same parameter setting as in [14]. Here, the data is imbalanced (the last two topics are used comparably rarely). The task is to recover the ground truth latent parameters as shown in Figure 3 Panel (a). Figure 3 Panel (b) shows the result of using traditional SVI where the last per topic word distribution is not recovered. Figure 3 Panel (c) shows the result of SVI with Poisson disk sampling. Similar to DPP in [14], all latent parameters are recovered.

k	50	80	102	150	200
k-DPP time	7.1680	29.687	58.4086	189.0303	436.4746
Fast k-DPP	0.1032	0.3312	0.6512	1.8745	4.1964
Poisson Disk	0.0461	0.0795	0.1048	0.1657	0.2391

Table 1: Sampling time for each mini-batch. CPU time is reported in the unit of seconds. In practice, the operation can be easily parallelized for all these methods.

Oxford Flower Classification with Softmax We further evaluate the performance of Poisson Disk sampling for SGD on a fine-grained classification task as in [14] with the same experimental setting. The minimum radius r is set to half of the median value of the distance measure D for Poisson disk sampling. Figure 4 show the test accuracy at the end of each epoch. Using Poisson disk sampling for mini-batch selection leads to the same effect as using k-DPP [14]. Table 1 shows the comparison of sampling time using 1) traditional DPP [7], 2) fast k-DPP [9], and 3) Poisson disk sampling. We can thus see that Poisson disk sampling is significantly faster than DPP, while providing very similar improvements for all cases.

5 Discussion

In this work, we generalize our previous work on using DPP for mini-batch sampling to general stochastic point processes with repulsive interactions. This enables us to explore various point patterns, and at the same time allows for efficient simulation. As an example, we utilized Poisson disk sampling with dart throwing for mini-batch sampling. This sampling method achieves the same effect as using DPP, but is much more efficient. In future works, we would like to experiment with more sampling methods.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. 2010.

[3] T.F Fu and Z.H. Zhang. CPSG-MCMC: Clustering-based preprocessing method for stochastic gradient MCMC. In *AISTATS*, 2017.

[4] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *JMLR*, 14(1):1303–1347, 2013.

[5] J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan, editors. *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, Ltd., 2008.

[6] A. Kulesza and B. Taskar. k-DPPs: Fixed-size determinantal point processes. In *ICML*, pages 1193–1200, 2011.

[7] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *arXiv:1207.6083*, 2012.

[8] A. Lagae and P. Dutré. A comparison of methods for generating poisson disk distributions. In *Computer Graphics Forum*, volume 27, pages 114–129, 2008.

[9] C.T. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes. *arXiv:1509.01618*.

[10] J. Møller and R. P. Waagepetersen. *Statistical inference and simulation for spatial point processes*. Chapman & Hall/CRC, 2003, 2004.

[11] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[12] A. C. Öztireli and M. Gross. Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph.*, 31(6):170:1–170:10, November 2012.

[13] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in variational inference. *arXiv preprint arXiv:1711.05597*, 2017.

[14] C. Zhang, H. Kjellström, and S. Mandt. Determinantal point processes for mini-batch diversification. In *UAI*, 2017.

[15] P.L. Zhao and T. Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv:1405.3080*.

A Derivation details

Preliminary: Campbell's Theorem A fundamental operation needed for analyzing point processes is computing expected values of sums of sampled values from functions $\mathbb{E}_{\mathcal{P}}\left[\sum f(\mathbf{x}_i)\right]$, where the expectation is computed over different realizations of the point process $\mathcal{P}$, and $\mathbf{x}_1, \cdots$ are the points in a given realization. Note that the number of points in a given realization is also random, and hence is often omitted for the sums. This can also be generalized to functions of more than one variable. Campbell's theorem is a fundamental theorem that relates these sums to integrals of arbitrary functions f, and the product densities of $\mathcal{P}$. In particular, we will work with sums of the form $\sum f(\mathbf{x}_i)$ and $\sum_{ij} f(\mathbf{x}_i, \mathbf{x}_j)$. These are given by the following expressions:

$$\mathbb{E}_{\mathcal{P}}\left[\sum f(\mathbf{x}_i)\right] = \int_{\mathbb{R}^d} f(\mathbf{x})\lambda(\mathbf{x})d\mathbf{x}, \tag{3}$$

where $f : \mathbb{R}^d \to \mathbb{R}$, and

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{i\neq j} f(\mathbf{x}_i, \mathbf{x}_j)\right] = \int_{\mathbb{R}^d\times\mathbb{R}^d} f(\mathbf{x}, \mathbf{y})\varrho(\mathbf{x}, \mathbf{y})d\mathbf{x}d\mathbf{y}, \tag{4}$$

for $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and under the common assumption [5] that no two points in a process can be at the same location almost surely. In practice, we typically observe sample points in a finite sub-space $\mathcal{V} \in \mathbb{R}^d$. For such cases, the integration domains can be replaced by $\mathcal{V}$.

Derivation of Gradient Variances For our derivations of bias and variance in this paper, we will need the first $\varrho^{(1)}$ and second $\varrho^{(2)}$ order product densities. The first order product density is given by $\varrho^{(1)}(\mathbf{x})d\mathbf{x} = p(\mathbf{x})$. It can be shown that the expected number of points of $\mathcal{P}$ in a set $\mathcal{B}$ can be written as the integral of this expression: $\mathbb{E}_{\mathcal{P}}\left[N(\mathcal{B})\right] = \int_{\mathcal{B}} \varrho^{(1)}(\mathbf{x})d\mathbf{x}$, where $N(\mathcal{B})$ is the (random) number of points of the process $\mathcal{P}$ in set $\mathcal{B}$. Hence, $\varrho^{(1)}(\mathbf{x})$ measures the local expected density of distributions generated by a point process. It is thus usually called the *intensity* of $\mathcal{P}$ and denoted by $\lambda(\mathbf{x}) = \varrho^{(1)}(\mathbf{x})$. Pairwise correlations are captured by the second order product density $\varrho^{(2)}(\mathbf{x}, \mathbf{y})d\mathbf{x}d\mathbf{y} = p(\mathbf{x}, \mathbf{y})$. The two statistics $\lambda(\mathbf{x})$ and $\varrho(\mathbf{x}, \mathbf{y})$ are sufficient to exactly express bias and variance of estimators for integrals or expected values.

The expected value $\mathbb{E}_{\mathcal{P}}\left[\hat{\mathbf{K}}(\theta)\right]$, can be simply derived by utilizing the expression in Equation 3 as follows

$$\mathbb{E}_{\mathcal{P}}\left[\hat{\mathbf{K}}(\theta)\right] = \mathbb{E}_{\mathcal{P}}\left[\sum_i w_i\mathbf{k}(\mathbf{x}_i, \theta)\right] = \int_{\mathcal{V}} w(\mathbf{x})\mathbf{k}(\mathbf{x}, \theta)\lambda(\mathbf{x})d\mathbf{x}, \tag{5}$$

where we assumed that the weights are sampled from a continuous function such that $w_i = w(\mathbf{x}_i)$, and the probability of having a point outside $\mathcal{V} \in \mathbb{R}^d$ is zero. The $\lambda(\mathbf{x})$ is fundamentally related to the assumed underlying distribution $p_{\text{data}'}(\mathbf{x})$ for the observed data-points. If the sampler does not introduce additional adaptivity, e.g. random sampling, then $\lambda(\mathbf{x})$ is directly proportional to $p_{\text{data}'}(\mathbf{x})$. If we would like to sample n points on average, we can set $\lambda(\mathbf{x}) = np_{\text{data}'}(\mathbf{x})$, assuming $p_{\text{data}'}(\mathbf{x})$ is normalized, and $w(\mathbf{x}) = 1$.

In general, we can play with $\lambda(\mathbf{x})$ and $w(\mathbf{x})$ to steer the expected value and hence bias. As a first example, we can consider making the estimator unbiased with respect to $p_{\text{data}'}(\mathbf{x})$. The true gradient for this case is given by $\nabla J_{\text{data}'}(\theta) = \nabla\mathbb{E}_{\mathbf{x}\sim p_{\text{data}'}}\left[l(\mathbf{x}, \theta)\right] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}'}}\left[\nabla l(\mathbf{x}, \theta)\right] = \int_{\mathcal{V}} \mathbf{k}(\mathbf{x}, \theta)p_{\text{data}'}(\mathbf{x})d\mathbf{x}$. Hence, an unbiased estimate $\hat{\mathbf{K}}$ can be obtained by setting $\lambda(\mathbf{x}) = np_{\text{data}'}(\mathbf{x})$, and $w(\mathbf{x}) = 1/n$ in Equation 5, giving $\mathbf{K} = \hat{\mathbf{K}}$. Similarly the $\hat{\mathbf{K}}$ can be made unbiased with respect to p_{data}, by accordingly setting λ and w, although this requires knowledge on p_{data}, of course. We can also compute the variance of the gradient estimate $\hat{\mathbf{K}}$ using the point processes framework. The variance of a multi-dimensional random variable can be written as $\text{var}_{\mathcal{P}}(\hat{\mathbf{K}}) = \text{tr}[\text{cov}_{\mathcal{P}}(\hat{\mathbf{K}})] = \sum_m \text{var}_{\mathcal{P}}(\hat{\mathbf{K}}_m)$, where $\hat{\mathbf{K}}_m$ denotes the m^{th} component of $\hat{\mathbf{K}}$. The variance for each dimension is given by $\text{var}_{\mathcal{P}}(\hat{\mathbf{K}}_m) = \mathbb{E}_{\mathcal{P}}[\hat{\mathbf{K}}_m^2] - (\mathbb{E}[\hat{\mathbf{K}}_m])^2$. These terms can be written in terms of λ and ϱ by utilizing Equations 4, and 3, respectively, as follows

$$\begin{aligned}\mathbb{E}_{\mathcal{P}}\left[\hat{\mathbf{K}}_m^2\right] &= \mathbb{E}_{\mathcal{P}}\left[\sum_{ij} w_i w_j \mathbf{k}_m(\mathbf{x}_i,\theta)\mathbf{k}_m(\mathbf{x}_j,\theta)\right] \\ &= \mathbb{E}_{\mathcal{P}}\left[\sum_{i\neq j} w_i w_j \mathbf{k}_m(\mathbf{x}_i,\theta)\mathbf{k}_m(\mathbf{x}_j,\theta)\right] + \mathbb{E}_{\mathcal{P}}\left[\sum_{i} w_i^2 \mathbf{k}_m^2(\mathbf{x}_i,\theta)\right] \\ &= \int_{\mathcal{V}\times\mathcal{V}} w(\mathbf{x})w(\mathbf{y})\mathbf{k}_m(\mathbf{x},\theta)\mathbf{k}_m(\mathbf{y},\theta)\varrho(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} \\ &+ \int_{\mathcal{V}} w^2(\mathbf{x})\mathbf{k}_m^2(\mathbf{x},\theta)\lambda(\mathbf{x})d\mathbf{x},\end{aligned} \tag{6}$$

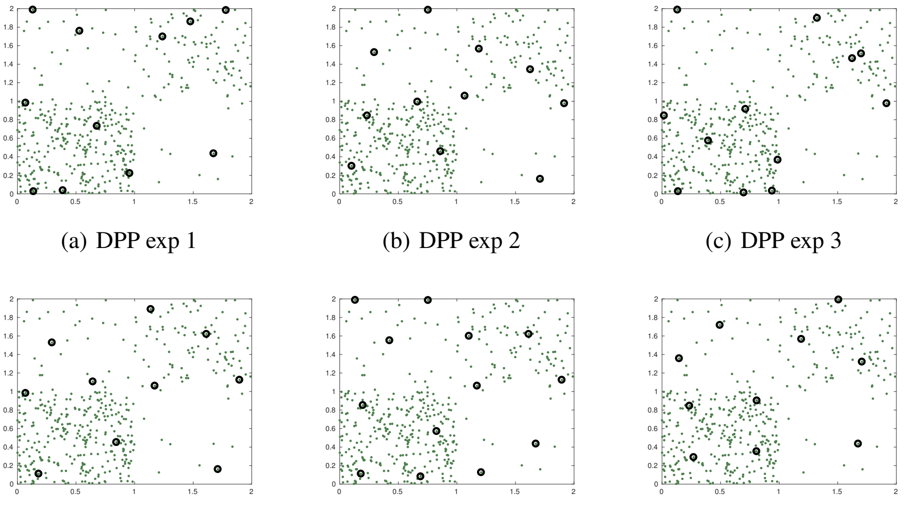
$$\begin{aligned}\left(\mathbb{E}_{\mathcal{P}}\left[\hat{\mathbf{K}}_m\right]\right)^2 &= \left(\int_{\mathcal{V}} w(\mathbf{x})\mathbf{k}_m(\mathbf{x},\theta)\lambda(\mathbf{x})d\mathbf{x}\right)^2 \\ &= \int_{\mathcal{V}\times\mathcal{V}} w(\mathbf{x})w(\mathbf{y})\mathbf{k}_m(\mathbf{x},\theta)\mathbf{k}_m(\mathbf{y},\theta)\lambda(\mathbf{x})\lambda(\mathbf{y})d\mathbf{x}d\mathbf{y}\end{aligned} \tag{7}$$

Finally, summing over all dimensions, we can get the following formula for the variance of $\hat{\mathbf{K}}$

$$\begin{aligned}\mathrm{var}_{\mathcal{P}}(\hat{\mathbf{K}}) &= \int_{\mathcal{V}\times\mathcal{V}} w(\mathbf{x})\lambda(\mathbf{x})w(\mathbf{y})\lambda(\mathbf{y})\mathbf{k}(\mathbf{x},\theta)^T\mathbf{k}(\mathbf{y},\theta)\left[\frac{\varrho(\mathbf{x},\mathbf{y})}{\lambda(\mathbf{x})\lambda(\mathbf{y})}-1\right]d\mathbf{x}d\mathbf{y} \\ &+ \int_{\mathcal{V}} w^2(\mathbf{x})\|\mathbf{k}(\mathbf{x},\theta)\|^2\lambda(\mathbf{x})d\mathbf{x}.\end{aligned} \tag{8}$$

B Additional Figures

Figure 5 shows examples of samples using DPP and using Poisson disk sampling (by dart throwing).



(a) DPP exp 1

(b) DPP exp 2

(c) DPP exp 3

(d) Poisson Disk exp 1

(e) Poisson Disk exp 2

(f) Poisson Disk exp 3

Figure 5: Examples of sampled data points. The first row are sampled using DPP and the second row are sampled with Poisson Disk sampling. The green dots are the training data and the black circles indicated that the data point has been sampled.