Investigating Quadrature In RKHSs Using DPPs

Daniel Fess

December 8, 2015

This work follows on from Francis Bach's paper: On the Equivalence between Kernel Quadrature Rules and Random Feature Expansions; and uses material from seminar notes produced by Dino Sejdinovic.

We consider a similar setup of the quadrature problem in a Reproducing Kernel Hilbert Space (RKHS) but with new sampling techniques, namely we introduce the use of a Determinantal Point Process (DPP).

This report is the culmination of two months of research over Summer 2015, supervised by Professor Yee Whye Teh of the University of Oxford. I wrote the report in August 2015, and further edited it in December 2015.

1 Bach's quadrature with importance sampling

1.1 The Classical Quadrature Problem in an RKHS

We work in a space $\mathcal{X}$ with probability measure $d\rho$, equipped with an RKHS of functions $\mathcal{H}_k$. The kernel $k(x, y)$ giving rise to this RKHS has integral operator T_k.
We would like to compute, for $g \in L_2(d\rho)$:

$$\rho_g[h] = \int h(x)g(x)d\rho(x) = \langle h, \mu_{k,g}\rangle_{\mathcal{H}_k}.$$

$\mu_{k,g} \in \mathcal{H}_k$ is the convolution with k, given by:

$$\mu_{k,g} = \int k(\cdot, x)g(x)d\rho(x)$$

We consider estimators of form $\tilde{\rho}_g[h] = \sum_{i=1}^n \alpha_i h(x_i)$ or equivalently the estimators of $\mu_{k,g}$ of form $\tilde{\mu}_{k,g} = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$, because from Cauchy-Schwarz

$$\sup_{\|h\|_{\mathcal{H}_k} \leq 1} |\tilde{\rho}_g[h] - \rho_g[h]| \quad = \quad \|\tilde{\mu}_{k,g} - \mu_{k,g}\|_{\mathcal{H}_k}.$$

Let us fix the choice $\{x_i\}$. We can find α by solving

$$\arg\min_{\alpha} \left\| \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) - \mu_{k,g} \right\|_{\mathcal{H}_k}^2 + n\lambda \|\alpha\|_2^2 =$$
$$\arg\min_{\alpha} \alpha^\top (K + n\lambda I) \alpha - 2\alpha^\top \mu_{k,g}(\mathbf{x}) \tag{1}$$

which has solution $\alpha = (K + n\lambda I)^{-1} \mu_{k,g}(\mathbf{x})$. We will return later to consider the effect of λ.

1.2 Bach's quadrature problem

In Bach's paper, he considers a more general setup where $\mathbf{x}$ is drawn according to an importance sampling distribution with density $q\ d\rho$.
We will draw samples $\{x_i\}_{i=1}^n \sim q d\rho$. We will aim for an estimator of the form $\tilde{\rho}_g[h] = \sum_{i=1}^n \frac{\beta_i}{\sqrt{q(x_i)}} h(x_i)$. This results in an optimization problem slightly different from (1):

$$\arg\min_{\beta} \left\| \sum_{i=1}^{n} \frac{\beta_i}{\sqrt{q(x_i)}} k(\cdot, x_i) - \mu_{k,g} \right\|_{\mathcal{H}_k}^2 + n\lambda \|\beta\|_2^2 =$$
$$\arg\min_{\beta} \beta^\top \left(\tilde{K}_q + n\lambda I \right) \beta - 2\beta^\top \left(q(x_i)^{-1/2} \mu_{k,g}(x_i) \right)_{i=1}^n,$$

with solution

$$\beta = \left(\tilde{K}_q + n\lambda I \right)^{-1} \left(q(x_i)^{-1/2} \mu_{k,g}(x_i) \right)_{i=1}^n \tag{2}$$

where $\tilde{K}_q$ is a modified Kernel matrix given by $\left[\tilde{K}_q\right]_{ij} = \frac{k(x_i,x_j)}{\sqrt{q(x_i)q(x_j)}}$.
This expression for β is equivalent to the following:
For each $i = 1, ..., n$:

$$\sum_{j=1}^{n} \left[\tilde{K}_q + n\lambda I \right]_{ij} \beta_j = q(x_i)^{-1/2} \mu_{k,g}(x_i)$$
$$\sum_{j=1}^{n} \left(\frac{k(x_i, x_j)}{\sqrt{q(x_i)q(x_j)}} + n\lambda \delta_{ij} \right) \beta_j = q(x_i)^{-1/2} \mu_{k,g}(x_i)$$
$$\sum_{j=1}^{n} \left(k(x_i, x_j) + n\lambda \delta_{ij} . \sqrt{q(x_i)q(x_j)} \right) \frac{\beta_j}{\sqrt{q(x_j)}} = \mu_{k,g}(x_i)$$
$$\sum_{j=1}^{n} \left(k(x_i, x_j) + n\lambda \delta_{ij} q(x_i) \right) \frac{\beta_j}{\sqrt{q(x_j)}} = \mu_{k,g}(x_i)$$

So if we let $\alpha_j = \frac{\beta_j}{\sqrt{q(x_j)}}$, we have $\alpha = (K + n\lambda . diag(q(\mathbf{x})))^{-1} \mu_{k,g}(\mathbf{x})$
Note that we now have $\tilde{\rho}_g[h] = \sum_{i=1}^n \alpha_i h(x_i)$, which simplifies implementation.

1.3 The Role of λ

λ acts as a regularisation parameter in the minimisation problem.

$$\min_{\beta} \left\| \sum_{i=1}^{n} \frac{\beta_i}{\sqrt{q(x_i)}} k(\cdot, x_i) - \mu_{k,g} \right\|_{\mathcal{H}_k}^2 + n\lambda \left\|\beta\right\|_2^2$$

However, it also plays a role in controlling the error in our quadrature, as the following proposition from Bach's paper outlines. We use a version of Bach's proposition specialised to the quadrature setting:

Proposition 1. *Let $\lambda > 0$, $I_k : \mathcal{H}_k \to L_2(d\rho)$ be the inclusion operator, $T_k : L_2(d\rho) \to L_2(d\rho)$ be the integral operator of kernel $k(x,y)$, and $\psi(\cdot, x) = T_k^{-1/2} I_k k(\cdot, x)$. We denote by $d_{max}(q,\lambda) = \sup_{x \in \mathcal{X}} \frac{1}{q(x)} \left\langle \psi(x,\cdot), (T_k + \lambda I)^{-1} \psi(x,\cdot) \right\rangle_{L_2(d\rho)}$.*
Let $x_1, \dots, x_n \overset{iid}{\sim} q\, d\rho$, then for any $\delta > 0$, if $n \geqslant 4 + 6 d_{max}(q,\lambda) log \frac{4 d_{max}(q,\lambda)}{\delta}$, with probability greater than $1-\delta$, we have

$$\sup_{\|f\|_{\mathcal{H}_k} \leqslant 1} \inf_{\|\beta\|_2^2 \leqslant \frac{4}{n}} \left\| f - \sum_{i=1}^{n} \beta_i q(x_i)^{-1/2} \psi(x_i, \cdot) \right\|_{L_2(d\rho)}^2 \leqslant 4\lambda$$

So λ corresponds to the error level, but note here the norm is the L_2 norm, whereas in the minimization problem we use the RKHS norm. Note also that bounding β in the proposition ensures robustness to regularisation.

When $\lambda = 0$, we cannot interpret λ as having some connection to the error level, but it does represent the minimization problem without regularisation, and hence it alters the weights used.

1.4 Bach's Optimal Distribution

For fixed $\lambda > 0$, Bach's optimal distribution minimizes $d_{max}(q,\lambda)$, and thus gives the lowest bound on n for which Proposition 1 applies. Bach gives an explicit formula for this distribution:

$$q(x) = \frac{\left\langle \psi(x,\cdot), (T_k + \lambda I)^{-1} \psi(x,\cdot) \right\rangle_{L_2(d\rho))}}{tr(T_k (T_k + I)^{-1})}$$

for which $d_{max}(q,\lambda) = d(\lambda) = tr(T_k (T_k + \lambda I)^{-1})$.
For a kernel with Mercer decomposition $k(x,y) = \sum_{i=1}^{\infty} \mu_i e_i(x) e_i(y)$, we have $q(x) \propto k_\lambda(x,x) = \sum_{i=1}^{\infty} \frac{\mu_i}{\mu_i + \lambda} e_i(x)^2$, where $k_\lambda(x,y)$ is the kernel of the operator $T_k(T_k + \lambda I)^{-1}$. For details of this derivation please refer to D. Sejdinovic's notes.

2 Introducing DPPs

2.1 Introductory Theory

A Determinantal Point Process is a probability measure over subsets of a ground set $\mathcal{Y}$. We shall only consider the case where $\mathcal{Y}$ is discrete (wlog of size n), and

look at a particular type of DPP called an L-ensemble. The structure of an L-ensemble is given by a real, symmetric, n x n, positive semi-definite matrix L, known as the kernel. For any $Y \subseteq \mathcal{Y}, P(Y) \propto det(L_Y)$, where L_Y is the sub matrix of L formed from the rows and columns corresponding to the elements of Y. In fact, it is true that $P(Y) = det(L_Y)/det(L+I)$.

We can introduce a matrix $K = L(L+I)^{-1}$. It can be shown that this matrix has the neat property $P(A \subseteq \mathbf{Y}) = det(K_A)$, where $\mathbf{Y}$ is a random subset chosen according to a DPP with kernel L. Since K is linked to the marginal probabilities, it is called the marginal kernel.

Some simple properties arising from this description of the DPP are:

$$P(i \in \mathbf{Y}) = K_{ii}$$

$$\begin{aligned} P(i,j \in \mathbf{Y}) &= K_{ii}K_{jj} - K_{ij}^2 \\ &= P(i \in \mathbf{Y})P(j \in \mathbf{Y}) - K_{ij}^2 \end{aligned}$$

So the diagonal entries represent the probability of an element appearing in a random subset, and the off-diagonal entries encode repulsion between elements. This is one of the key properties of DPPs and why they are attractive to us - they model global, negative correlation, and a sample from a DPP is usually diverse (with respect to some measure of similarity given by L and K).

2.2 DPPs and Bach's Paper

In Bach's paper he samples from an importance sampling distribution $q\ d\rho$, and for his quadrature problem gives a specific optimal distribution. We shall investigate the case when we sample according to a DPP and compare our results to Bach.

The reason behind this is that (given the setup in **1.2**, with $\lambda > 0$), if we take a finite set of points $\{x_1, ..., x_n\}$, the optimal distribution evaluated on these points is proportional to the marginal probabilities $P(i \in \mathbf{Y})$ where $\mathbf{Y}$ is distributed according to a DPP with kernel $\frac{1}{\lambda}K$, where K is the (RKHS) kernel matrix: $K_{ij} = k(x_i, x_j)$.

Note that there is potential for confusion here: the kernel L of the DPP is equal to $\frac{1}{\lambda}K$, and the marginal kernel, which records the marginal probabilities, is equal to $\frac{1}{\lambda}K(\frac{1}{\lambda}K+I)^{-1} = K(K+\lambda I)^{-1}$.

Furthermore, the repulsive properties of the DPP make it seem intuitively well-suited to the quadrature problem - we would like our points to be spread out, in order not to over- or under-estimate any part of the function.

2.3 n-DPPs

Drawing a set from a DPP can result in a set of any size (but clearly no larger than the ground set $\mathcal{Y}$). n-DPPs are DPPs where we condition on the size n of the set. When we compare quadrature with sampling from a DPP to, for example, sampling from Bach's optimal distribution, it helps to fix n so that

we can directly compare the two methods, with all other variables being equal. There are efficient algorithms for directly sampling according to an n-DPP - these can be found online and are the work of Alex Kulesza/Ben Taskar.

3 Computational Work

3.1 Bach's Optimal Distribution

We know that $q(x) \propto k_\lambda(x,x) = \sum_{i=1}^{\infty} \frac{\mu_i}{\mu_i+\lambda} e_i(x)^2$, where $\{e_i, \mu_i\}$ are eigenfunction-eigenvalue pairs for the integral operator of $k(x,y)$, $T_k : L_2(d\rho) \to L_2(d\rho)$.

Only in some cases do we know the Mercer Decomposition of the kernel for the desired underlying measure ρ, and even in these cases $q(x)$ will rarely have a closed form. In my work, where possible, I have truncated the infinite sum after a sufficient number of terms and evaluated this finite sum on a fine grid of $\mathcal{X}$. Then I have considered $q(x)$ to be piecewise constant (taking a fixed value around each point of our fine grid) and then normalising; we want to know q everywhere, but in reality we only know it on our fine grid - in performing this tweak we circumvent the problem. It is relatively straight-forward to sample from the resulting piecewise constant density.

A better method, not employed in this project, could be to use MCMC to sample from $q\ d\rho$, which avoids the problem of knowing $q(x)$ exactly only on a fine grid . Alternatively, when we don't have the Mercer Decomposition (for the underlying measure ρ), it might be possible to use the eigendecomposition of the kernel matrix K to estimate $q\ d\rho$.

3.1.1 Effect of Lambda

As well as investigating the performance of variations on the quadrature problem, I have spent a small amount of time looking into how Bach's optimal distribution varies with λ. I have considered the case with squared exponential (a.k.a. Gaussian RBF) kernel and gaussian measure (on $\mathbb{R}$) [figure 1], and the case with Brownian covariance kernel ($k(x,y) = min(x,y)$) and U[0,1] measure [figure 2]. In the first case, the optimal distribution widens and flattens out as λ decreases, diverging from any normal distribution, and in particular from the measure, whereas in the second case the optimal distribution clearly converges to the ambient measure U[0,1] as λ decreases. Perhaps the optimal distribution 'likes' to flatten out and spread itself out as evenly as possible as λ decreases. Intriguing - and it begs further investigation.

Note: The plots are of $q_\lambda\ d\rho$, not of q_λ - that is, the plots are of Bach's optimal distribution with respect to the Lebesgue measure, as opposed to the measure ρ in the quadrature problem.

3.2 Approximating the Optimal Distribution

For the cases where we don't have the Mercer Decomposition of $k(x,y)$, we outline a method to approximately sample from $q(x)$ (density wrt ρ). We call

Figure 1: SE kernel, Gaussian measure on $\mathbb{R}$ - plotting measure and optimal distribution for various λ

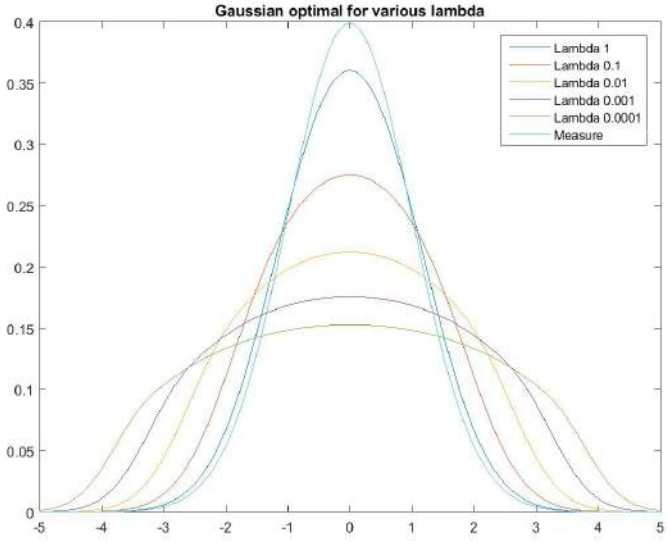
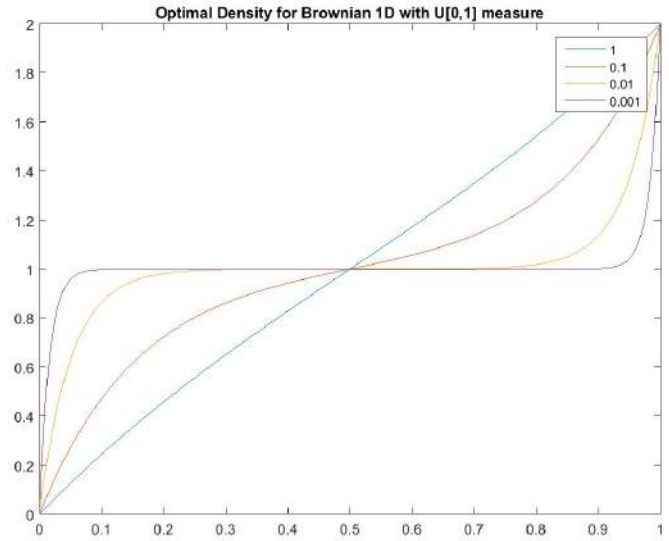


Figure 2: Brownian covariance kernel, U[0,1] measure - plotting optimal distribution for various λ



this method 'Resampling':

1. $q(x) \propto k_\lambda(x,x)$, where $k_\lambda(x,y)$ is the kernel of the operator $T_k(T_k+\lambda I)^{-1}$.
2. Draw a set of points $\mathbf{x}$ of size N according to ρ, where N is sufficiently large.
3. Construct the (RKHS) kernel matrix, K, given by $K_{ij} = k(x_i, x_j)$.
4. For large N, $K(K+\lambda I)^{-1}$ approximates $T_k(T_k+\lambda I)^{-1}$.
5. Subsample from $\mathbf{x}$ without replacement according to the mass function $P(x_i) = \frac{[K(K+\lambda I)^{-1}]_{ii}}{tr(K(K+\lambda I)^{-1})}$.

3.3 Sampling from an n-DPP

1. Sample N points from ρ, where N is sufficiently large.
2. Construct a DPP on these N points, with kernel $\frac{1}{\lambda}K$ and, hence, marginal kernel $K(K+\lambda I)^{-1}$.
3. Using code written by Kulesza/Taskar, sample according to an n-DPP. This code is available online.

An important property is that sampling from an n-DPP with kernel $\frac{1}{\lambda}K$ is independent of the value of λ (provided it is non-zero). Decreasing lambda makes larger sets more likely to appear, but does not have an effect when comparing the probabilities of sets of the same size. For this reason, no matter the value of λ we use the DPP with kernel K. For details, see D. Sejdinovic's notes.

3.4 Implementing the Quadrature

Assume now that λ is fixed, $g \equiv 1$ and we have a function $h(x) \in \mathcal{H}_k$ we wish to integrate wrt ρ.

We shall study four methods, which differ in their sampling method:

1. Sampling from ρ
2. Sampling from an n-DPP
3. Sampling from Bach's optimal distribution, q_λ, where available
4. 'Resampling' - approximately sampling from Bach's optimal distribution, q_λ

Once we have sampled our points we wish to construct the weights $\alpha = \left(K + n\lambda . diag(q(\mathbf{x}))\right)^{-1} \mu_{k,g}(\mathbf{x})$ and perform the quadrature $\tilde{\rho}_g[h] = \sum_{i=1}^n \alpha_i h(x_i)$.

Two problems arise:

1. Computing the mean embedding $\mu_{k,g}(\cdot) = \int k(\cdot, x) g(x) d\rho(x) = \int k(\cdot, x) d\rho(x)$

2. When sampling from an n-DPP or Resampling to approximate q_λ, what is q (in the weights)? We have not drawn from a density in these cases.

For the first problem, in some cases the mean embedding has a closed form, and in others we can use straight-forward Monte Carlo integration to estimate it: Say we have sampled $x_1, ..., x_n$ according to one of our four methods, then we sample $X_1, ..., X_M$ from ρ and $\mu_{k,g}(x_i) \approx \frac{1}{M}\sum_{m=1}^{M} k(x_i, X_m)$.

For the second problem, the jury's still out. For both cases, using $q(x_i) = P(x_i) = \frac{[K(K+\lambda I)^{-1}]_{ii}}{tr(K(K+\lambda I)^{-1})}$ seems to give good convergence, but it's not a density since it's only defined on a finite set of points, and it could be awkward to extend to a density given that the set it's defined on is random (drawn from ρ). For the n-DPP case I have used $q(x_i) = \frac{[K(K+I)^{-1}]_{ii}}{tr(K(K+I)^{-1})}$ in all but one case - I have used it for some values $\lambda \neq 1$, even though the formula appears to only be suited for the case $\lambda = 1$. It has generally provided good convergence. These mass functions are proportional to the marginals of a DPP, when really we are drawing from an n-DPP, so perhaps the choice of q should reflect that. Or perhaps the role of q is not relevant in these situations and we should just set $q = 1$, which would be equivalent to solving (1), with weights $\alpha = (K + n\lambda I)^{-1} \mu_{k,g}(\mathbf{x})$.

3.5 $\lambda = 0$

When $\lambda = 0$, Bach's work is less meaningful, since the optimal distribution q_λ is only defined for $\lambda > 0$. We do not have the link between DPPs and Bach's optimal distribution now, but we shall still see how quadrature peforms with sampling from an n-DPP with kernel K, since a set drawn according to an n-DPP will still be diverse and spread out, and we hope that because of this the error will be smaller.

The minimization problem changes slightly so there is no regularization when $\lambda = 0$, and this in turn adjusts the weights to $\alpha = K^{-1}\mu_{k,g}(\mathbf{x})$ where $\tilde{\rho}_g[h] = \sum_{i=1}^{n} \alpha_i h(x_i)$.

3.6 Comparison of different sampling methods

We will now look at the convergence of the error with respect to the number of points used in quadrature, n. We generate functions by drawing $y_1, ..., y_r \overset{iid}{\sim} U([0,1]^p)$, where p is the dimension we are working in (i.e. $\mathcal{X} \subseteq \mathbb{R}^p$) (we assume $[0,1]^p \subset \mathcal{X}$), and $c_1, ..., c_r \overset{iid}{\sim} N(0,1)$ and setting $h(x) = \sum_{i=1}^{r} c_i k(x, y_i)$, then normalising wrt RKHS norm. We generate n functions $\{f_i : i = 1, ..., n\}$ and compute I_i the integral of f_i, A_i our estimate of the integral, the error $E_i = |A_i - I_i|$, and we consider $log_{10}(\sqrt{\frac{1}{n}\Sigma E_i^2})$, which is in a particular sense the log of the average of errors. We plot $log_{10}(n)$ against this 'log of error'. Regression is also plotted for each curve, and the gradient of the regression line is given in the legend. Since the plots are log-log, the gradient α represents convergence of the error of the form $\frac{C}{n^\alpha}$.

We will start with the cases where $\lambda = 0$ and then introduce λ non-zero.

3.6.1 Sobolev space s=1 kernel / U[0,1] measure / $\lambda = 0$ - Drawing from ρ vs. classical quadrature rules

See figure 3 to compare drawing from ρ to the classical methods of Gauss-Legendre, Simpson and Sobol - "Kernel t=1" is method 1 as in **3.4** (drawing from ρ). Drawing from ρ and weighting as detailed previously is outperformed by G-L and Simpson, although almost all functions in this space are very well behaved (differentiable, bounded) - the setting in which Simpson and G-L perform well, so this is to be expected. We hope n-DPP can at least reduce the gap on the classical methods.

3.6.2 Sobolev space s=1 kernel / U[0,1] measure / $\lambda = 0$ - Drawing from ρ vs. n-DPP

See figure 4. Regression for both lines gives a gradient of approximately 2.0, which is the best we can hope for (see D. Sejdinovic's notes), but the error for n-DPP decreases more rapidly, by a constant factor. The n-DPP method is significantly slower in run time, however.

Note: in this figure, "Bach" refers to drawing from ρ and weighting appropriately (i.e. using the weights which Bach proposes in his paper). In this figure and all others, "DPP" really means n-DPP.

Figure 3: Sobolev space, U[0,1] measure, $\lambda = 0$ - ρ vs classical methods

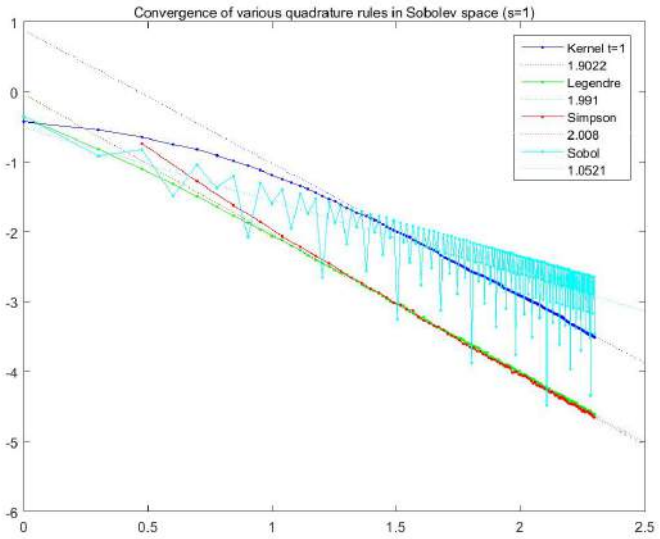
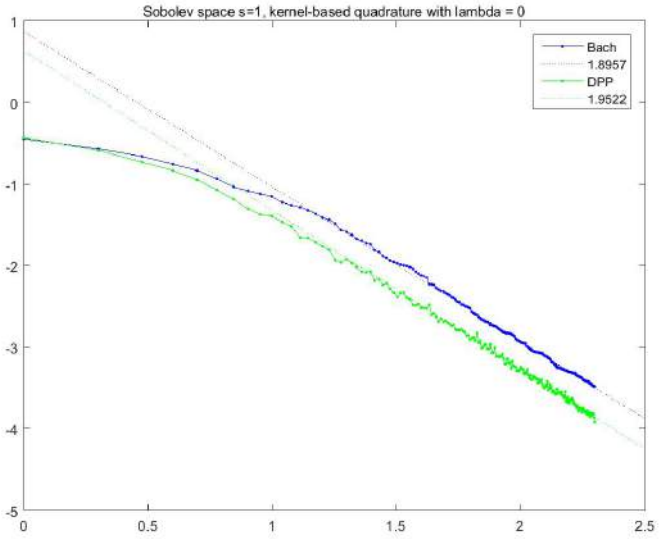


Figure 4: Sobolev space, U[0,1] measure, $\lambda = 0$ - ρ vs n-DPP



3.6.3 Sobolev space s=1 kernel / Beta(0.5, 0.5) measure / $\lambda = 0$ - Drawing from ρ vs. n-DPP vs. classical methods

See figure 5. "Beta" refers to drawing from ρ. Interestingly, Simpson and G-L struggle here, supposedly because the measure is now awkward, tending to ∞ near 0 and 1; G-L places many points near 0 and 1, Simpson cannot place points at 0 and 1 (since the integrand is infinite there), and functions are not bounded in this space. We don't bother with Sobol since it doesn't compete with other methods.

n-DPP with kernel K again performs better by a constant factor than drawing from the ambient measure, but with the caveat of more computational effort. Both methods converge at a rate of almost $O(n^{-2})$.

3.6.4 Brownian covariance kernel / U[0,1] measure / $\lambda = 0$ - Drawing from ρ vs. n-DPP

See figure 6. Here the kernel is $k(x, y) = min(x, y)$.

n-DPP outperforms drawing from ρ = U[0,1] once again by approximately a constant factor. Both methods converge at a rate of around $O(n^{-2})$, and are flexible to the space of functions, which is a very good quality.

Here are some example functions from the corresponding RKHS:

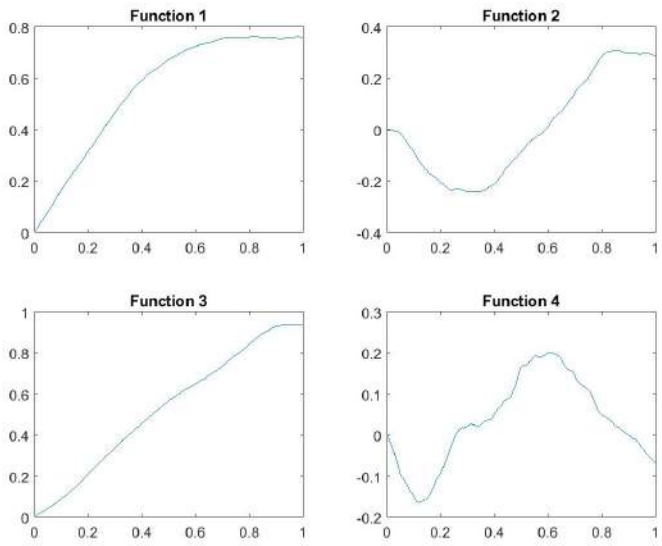


Figure 5: Sobolev space, Beta(0.5,0.5) measure, $\lambda = 0$ - ρ vs. n-DPP vs. classical methods

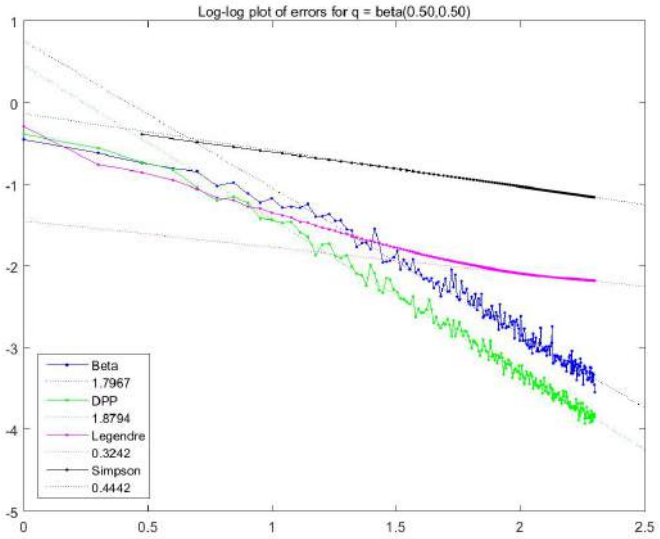
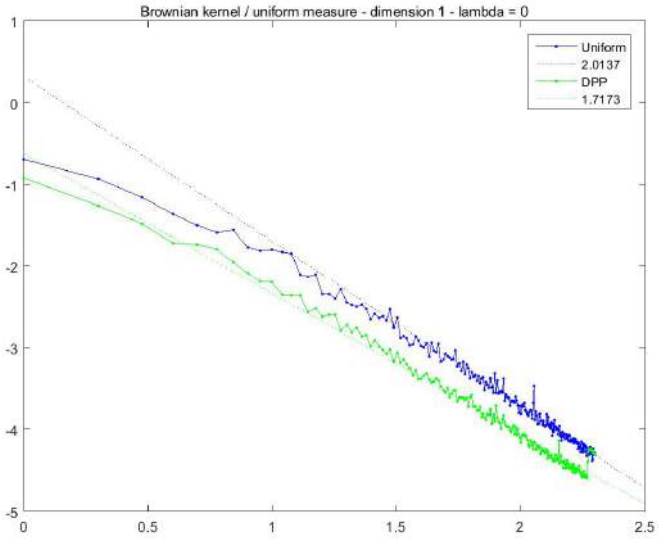


Figure 6: Brownian covariance kernel, U[0,1] measure, $\lambda = 0$ - ρ vs. n-DPP



3.6.5 Squared Exponential kernel / Gaussian measure on $\mathbb{R}^2$ and $\mathbb{R}^5$ / $\lambda = 0$ - Drawing from ρ vs. n-DPP

See figures 7 and 8. Here $k(x,y) = exp(-\frac{\|x-y\|_2^2}{2l^2})$.

In the $\mathbb{R}^2$ case, n-DPP appears to perform slightly worse than drawing from ρ and weighting appropriately, although the convergence may be faster once n is already large. This tells us that maybe n-DPP is not the best option (convergence-wise) for all situations. Convergence is nonetheless very fast for both methods.

In the $\mathbb{R}^5$ case, both methods perform similarly.

Figure 7: SE kernel, Gaussian measure on $\mathbb{R}^2$, $\lambda = 0$ - ρ vs. n-DPP

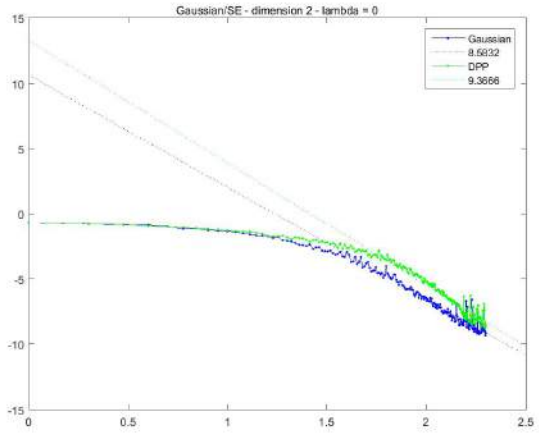
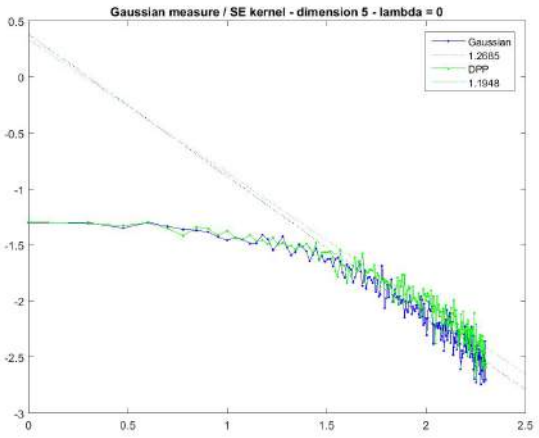


Figure 8: SE kernel, Gaussian measure on $\mathbb{R}^5$, $\lambda = 0$ - ρ vs. n-DPP



3.6.6 Squared Exponential kernel / Gaussian measure on $\mathbb{R}$ / $\lambda = 0.0001$ - ρ vs. n-DPP vs. Bach's Optimal

See figure 9 - "Bach" refers to Bach's optimal distribution. n-DPP massively outperforms drawing from ρ and drawing from Bach's optimal distribution, $q_\lambda \, d\rho$. We see that drawing from ρ and from $q_\lambda \, d\rho$ both result in the 'log of error' plateauing somewhere around $log(\lambda)$. This makes some sense, in that λ corresponds to the error level, but it is curious that the n-DPP method is not suspectible to this plateauing effect. Also intriguing is that ρ and $q_\lambda \, d\rho$ exhibit very similar convergence of error, despite the distributions being very different. n-DPP plateaus later but I think this may be due to numerical issues. In all cases convergence is very fast up until plateauing.

3.6.7 Brownian covariance kernel / U[0,1] measure / $\lambda = 0.001$ - ρ vs. n-DPP vs. Optimal

See figure 10. Again we see drawing from ρ and from $q_\lambda \, d\rho$ result in the curious plateauing effect around $log(\lambda)$, a feature which the n-DPP case doesn't exhibit. Unlike in the last example, here it makes sense for ρ and $q_\lambda \, d\rho$ to exhibit similar error convergence, since $q_\lambda \, d\rho$ converges in distribution to ρ as λ goes to 0. Convergence for all methods is slower than $O(n^{-2})$.

Figure 9: SE kernel, Gaussian measure on $\mathbb{R}$, $\lambda = 0.0001$ - ρ vs. n-DPP vs. Bach's optimal

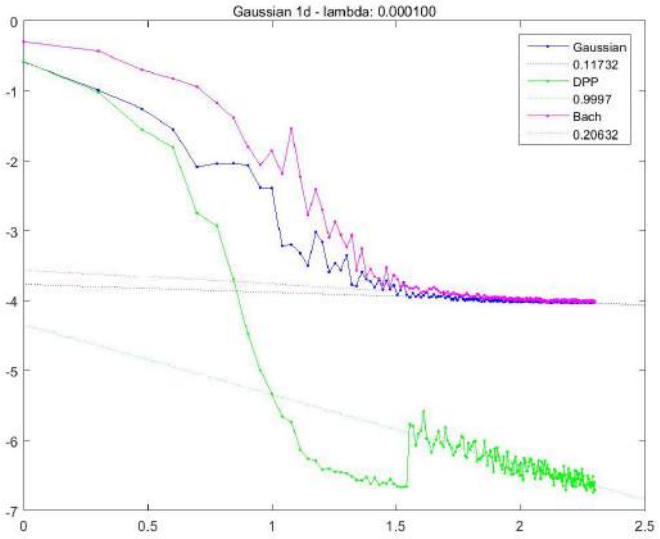
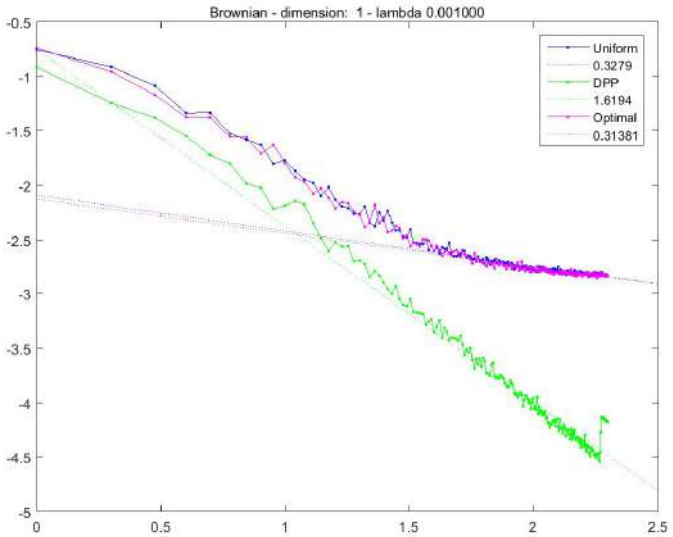


Figure 10: Brownian covariance kernel, U[0,1] measure, $\lambda = 0.001$ - ρ vs. n-DPP vs. Bach's optimal



3.6.8 Sobolev space s=1 kernel / Beta(0.5, 0.5) measure / λ = 0.01/0.001 - ρ vs. n-DPP vs. Resampling

See figures 11 and 12. Here we don't know the Mercer Decomposition of the kernel (because the measure is awkward) so we don't know the optimal distribution. We use our Resampling technique to approximate it.

For both λ, n-DPP performs better (by a constant factor) than Resampling, and Resampling performs similarly to drawing from ρ, up until drawing from ρ plateaus near to $log(\lambda)$, as usual. Regression tells us all methods converge at a rate of approximately $O(n^{-2})$, up until plateauing (if it occurs).

It is also worth saying that in this case, I at first used $q(x_i) = \frac{[K(K+I)^{-1}]_{ii}}{tr(K(K+I)^{-1})}$ for the n-DPP case, but the error appeared to plateau, and on changing q to $q(x_i) = \frac{[K(K+\lambda I)^{-1}]_{ii}}{tr(K(K+\lambda I)^{-1})}$ convergence improved significantly. This is the only case I tested the effect of changing q in this way.

Figure 11: Sobolev space, Beta(0.5,0.5) measure, $\lambda = 0.01$ - ρ vs. n-DPP vs. Resampling

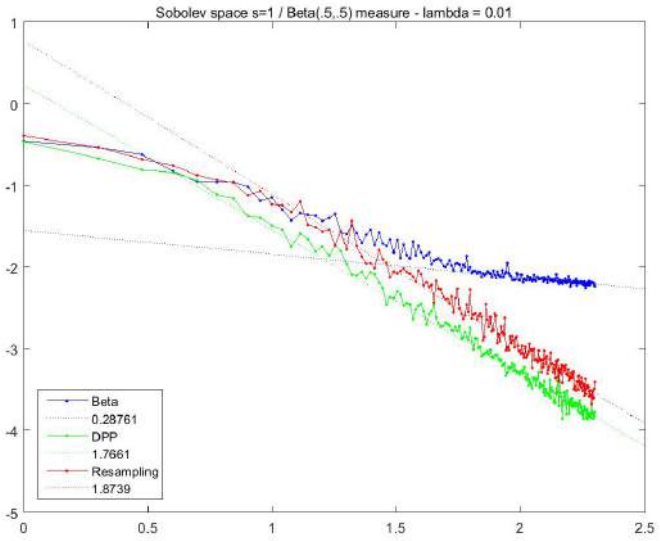
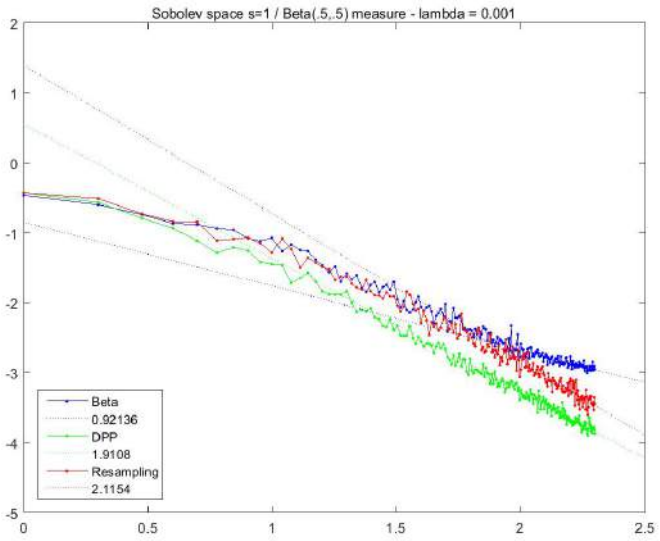


Figure 12: Sobolev space, Beta(0.5,0.5) measure, $\lambda = 0.001$ - ρ vs. n-DPP vs. Resampling



4 Summary

In almost all cases, the n-DPP method has outperformed all other methods in terms of convergence of the error with respect to the number of points used. We have also seen it to be flexible to the space of functions and the measure used; the same cannot be said for Gauss-Legendre or Simpson, or Bach's optimal method which, for the moment, relies on the knowledge of the kernel's Mercer Decomposition - although alternative ways to efficiently sample from q_λ may well be possible.

However, the n-DPP method is not without its problems. Chief among them is the computational effort required to implement it. If more efficient algorithms were used, or we could use a sufficiently accurate approximation of an n-DPP, this could reduce the extent of the problem.

In our limited testing of it, resampling to approximate Bach's optimal distribution has performed very well, almost in line with the n-DPP method, but it has similar computational issues to n-DPPs since it requires the inversion of a matrix. Further, it has the same problem as n-DPPs when it comes to deciding what q is when we are computing the weights, though in the examples examined here this has caused few problems. It's possible that the method would perform less well in different RKHSs with different measures, or that there is a better or more reliable choice for q.

One major thread which could be investigated is the plateauing effect often seen when we sample from ρ or $q\ d\rho$, e.g. with some theoretical guarantees expanding on Bach's Proposition 1. Theoretical guarantees for the n-DPP and Resampling cases would be good too, since these are inspired by Bach's work but don't fit into his framework perfectly.

In this investigation we have only looked at sampling from a discrete n-DPP. It would be interesting to see how sampling from a discrete DPP (i.e. no conditioning on the sample size) performs in quadrature; I have, in fact, started looking into this but have not had enough time to properly explore it. The matlab file is gaussian_lambda_comparison.m.

Sampling from a discrete DPP would still require sampling a large set of points from ρ in the first place, but sampling directly from a continuous DPP on the whole of $\mathcal{X}$ would avoid this, giving another hopefully fruitful direction to explore.

With both an intuitive and a theoretical link (via Bach's paper) to the quadrature problem in RKHSs, DPPs appeared to be a powerful tool, and through various tests have indeed performed consistently well. I look forward to seeing how we can expand on the methods in this report in the future.