Tutorial: Determinantal Point Processes and their Application to Signal Processing and Machine Learning

Simon Barthelmé, Nicolas Tremblay

CNRS, GIPSA-lab, Univ. Grenoble-Alpes, France

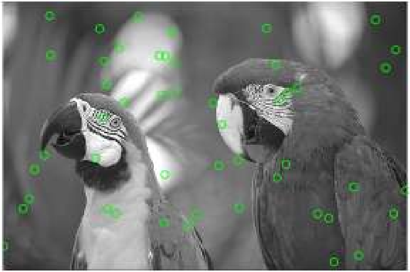
In a nutshell, determinantal point processes (or DPP) :

- are *random* processes that *induce diversity*.
- are *tractable*.
- are used for three main purposes:
  - i/ *produce diverse samples* of a large database
  - ii/ *use as a tool* in a variety of SP/ML contexts
  - iii/ *characterize* various observed phenomena.

DPPs induce diversity

Figure: Example of iid uniform sampling

DPPs induce diversity



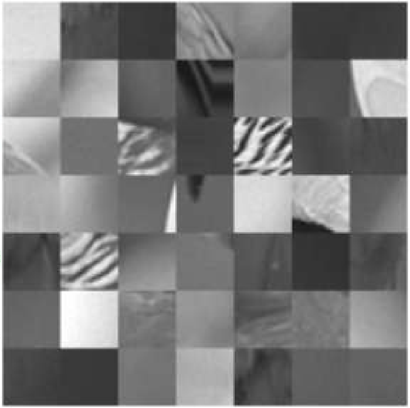


Figure: Example of iid uniform sampling

DPPs induce diversity

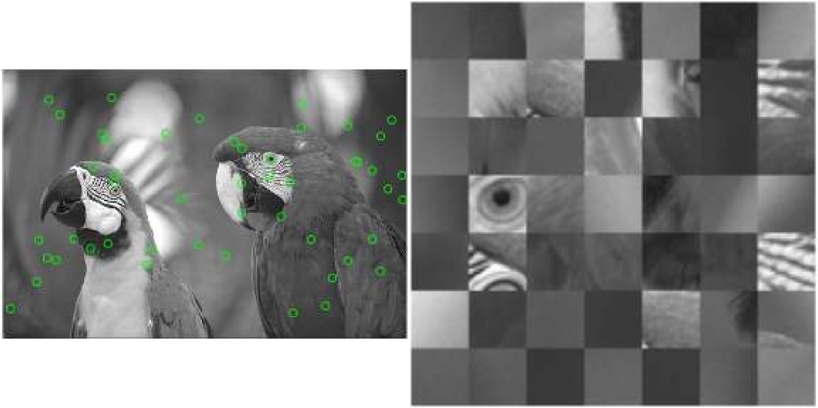


Figure: Example of iid uniform sampling

DPPs induce diversity

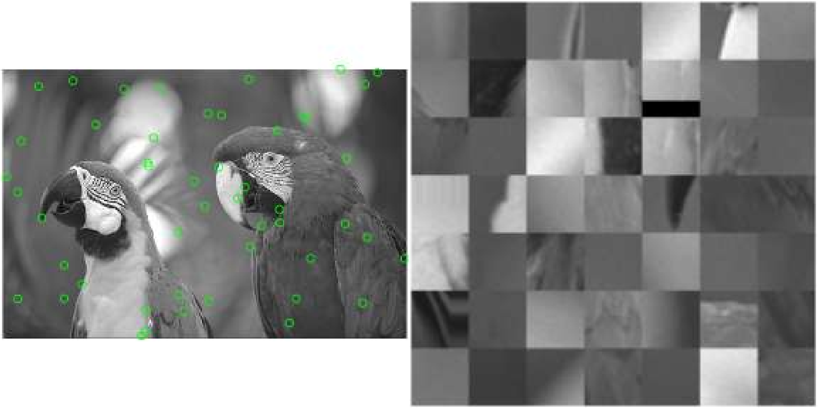


Figure: Example of iid uniform sampling

DPPs induce diversity

Figure: Example of iid uniform sampling

DPPs induce diversity

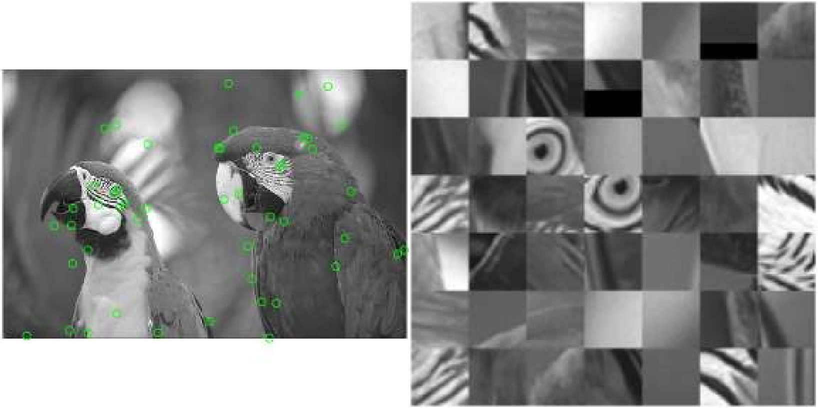
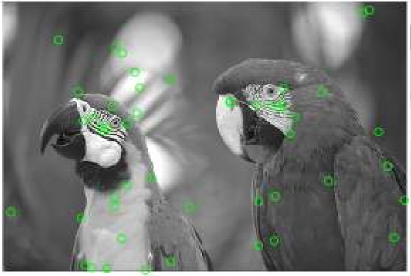


Figure: Example of DPP sampling

DPPs induce diversity

Figure: Example of DPP sampling

DPPs induce diversity



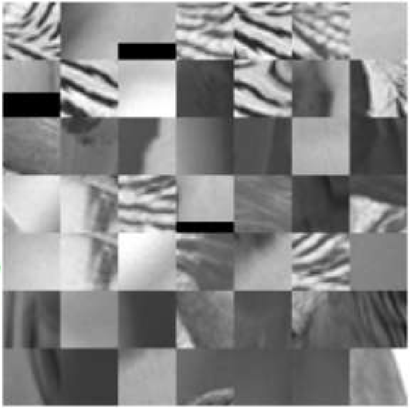


Figure: Example of DPP sampling

DPPs induce diversity

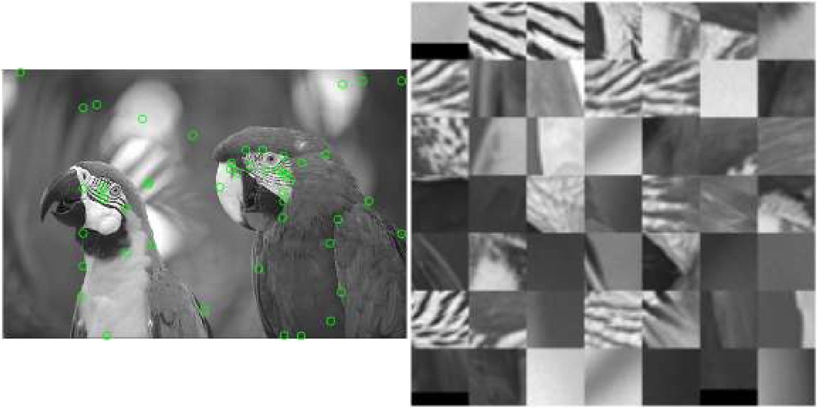
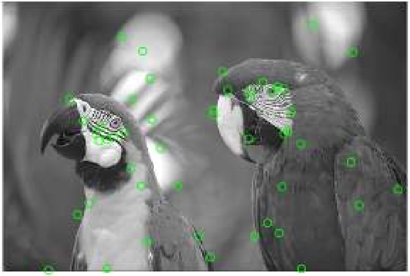


Figure: Example of DPP sampling

DPPs induce diversity



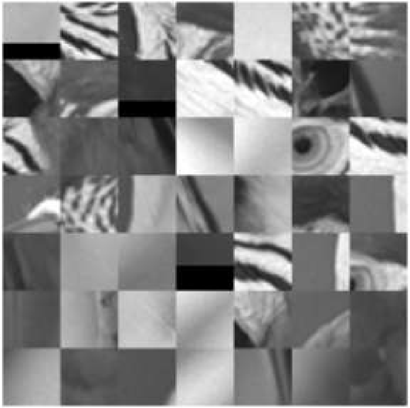


Figure: Example of DPP sampling

i/ This sample diversity can be directly useful[1 2]:

summary generation:

[1] left figure: from Kulesza and Taskar, *DPPs for machine learning*, Found. and Trends in ML, 2013

[2] right figure: from G. Gautier's slides guilgautier.github.io/pdfs/GaBaVa17_slides.pdf

i/ This sample diversity can be directly useful[1][2]:

summary generation:

search engines / recommendation:

[1]left figure: from Kulesza and Taskar, *DPPs for machine learning*, Found. and Trends in ML, 2013

[2]right figure: from G. Gautier's slides `guilgautier.github.io/pdfs/GaBaVa17_slides.pdf`

i/ This sample diversity can be directly useful[1][2]:

summary generation:

search engines / recommendation:

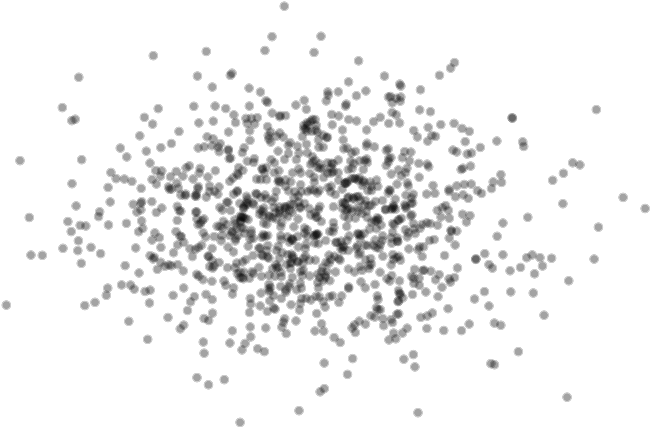
ii/ DPP samples can also be used as a tool in several SP/ML contexts:

- Monte Carlo integration
- Feature selection problems
- Coresets
- etc.

[1] left figure: from Kulesza and Taskar, *DPPs for machine learning*, Found. and Trends in ML, 2013

[2] right figure: from G. Gautier's slides guilgautier.github.io/pdfs/GaBaVa17_slides.pdf

DPPs as a tool: an example



DPPs as a tool: an example

DPPs as a tool: an example

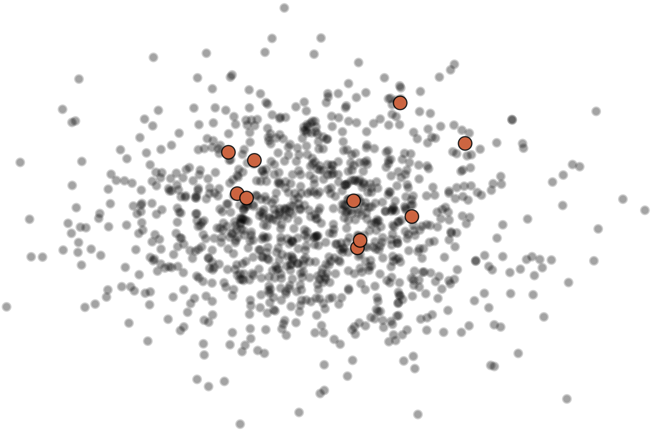


Figure: Example of iid uniform sampling

DPPs as a tool: an example

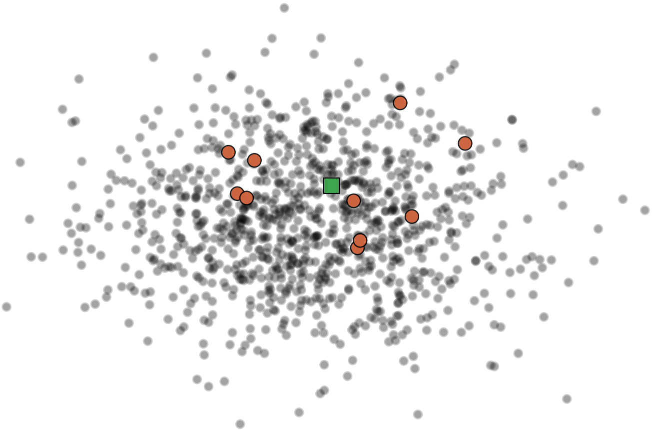


Figure: Example of iid uniform sampling

DPPs as a tool: an example

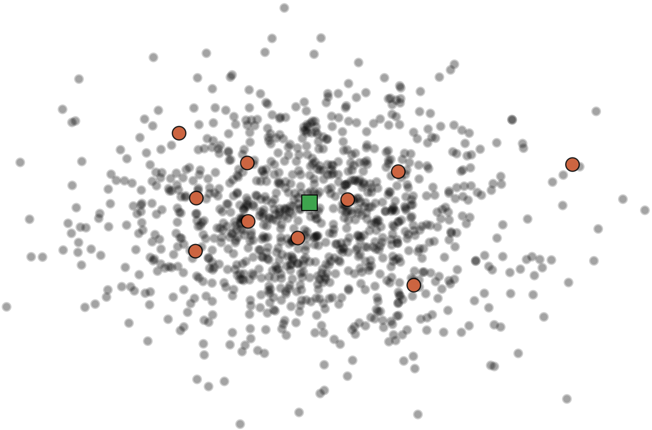


Figure: Example of iid uniform sampling

DPPs as a tool: an example

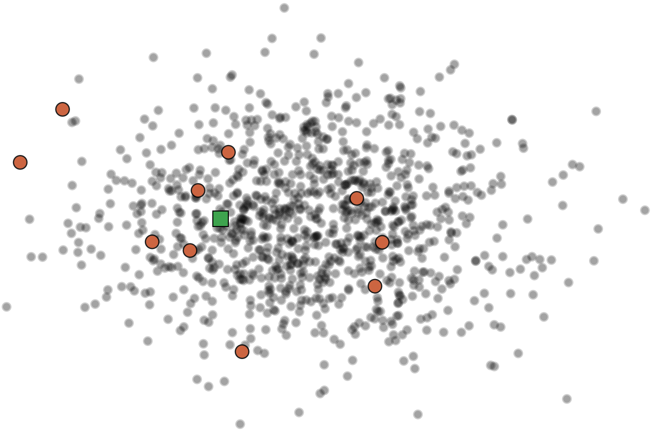


Figure: Example of iid uniform sampling

DPPs as a tool: an example

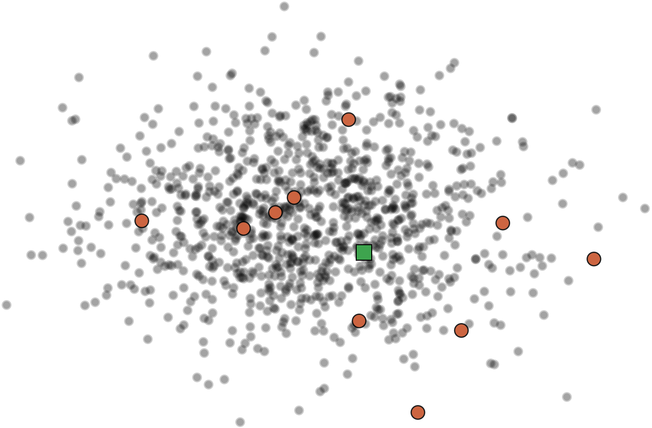


Figure: Example of iid uniform sampling

DPPs as a tool: an example

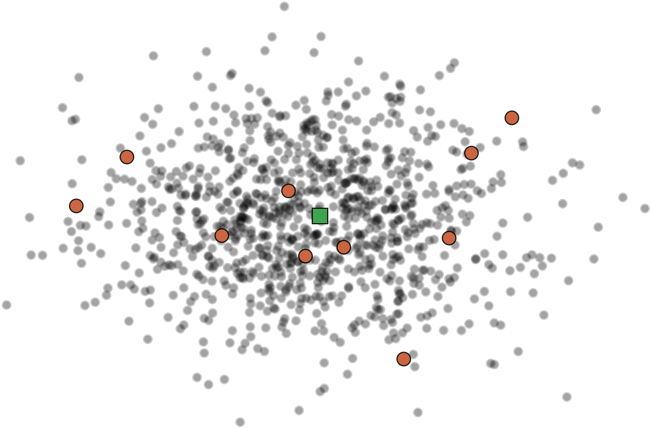


Figure: Example of iid uniform sampling

DPPs as a tool: an example

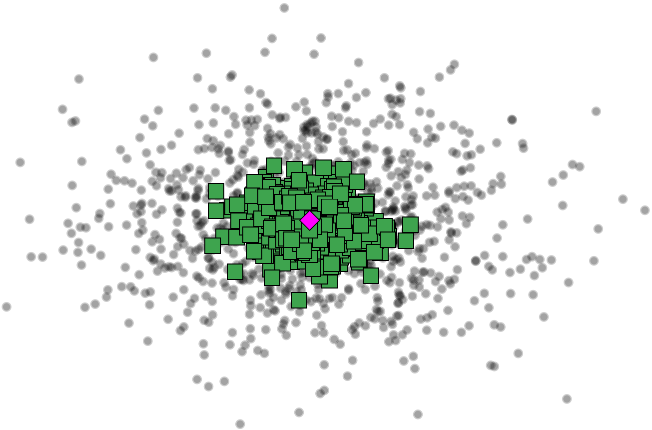


Figure: iid estimations of the mean

DPPs as a tool: an example

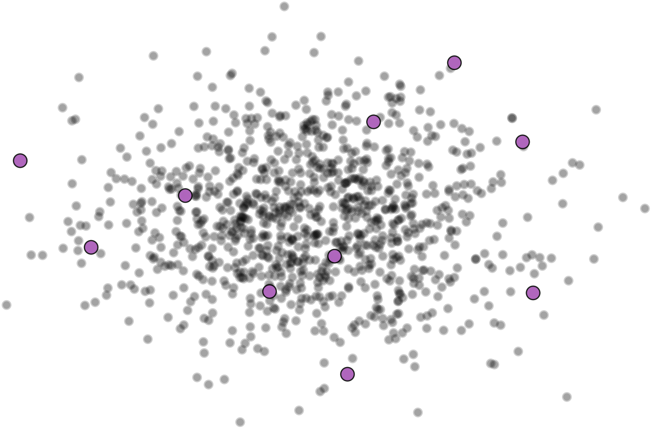


Figure: Example of DPP sampling

DPPs as a tool: an example

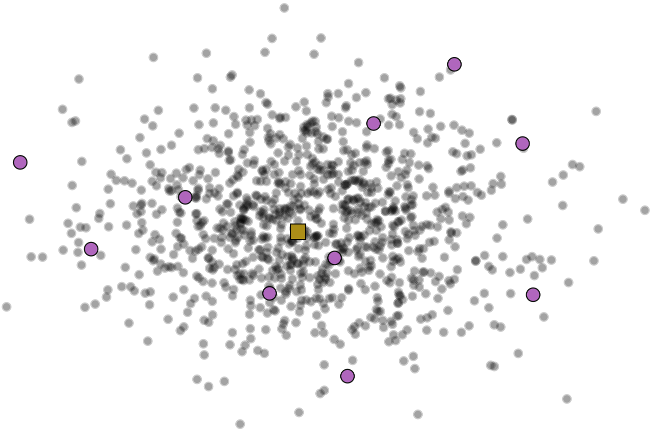


Figure: Example of DPP sampling

DPPs as a tool: an example

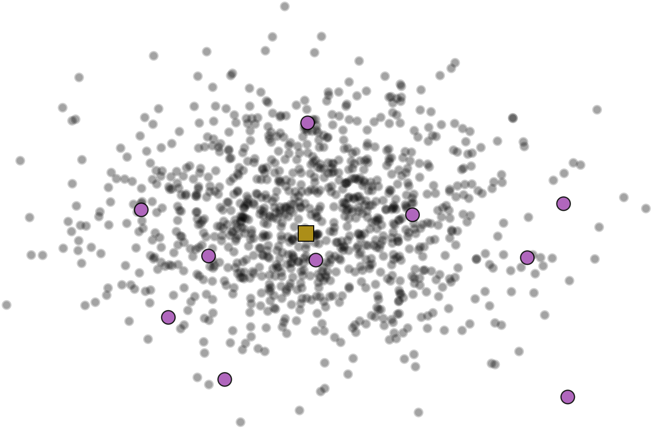


Figure: Example of DPP sampling

DPPs as a tool: an example

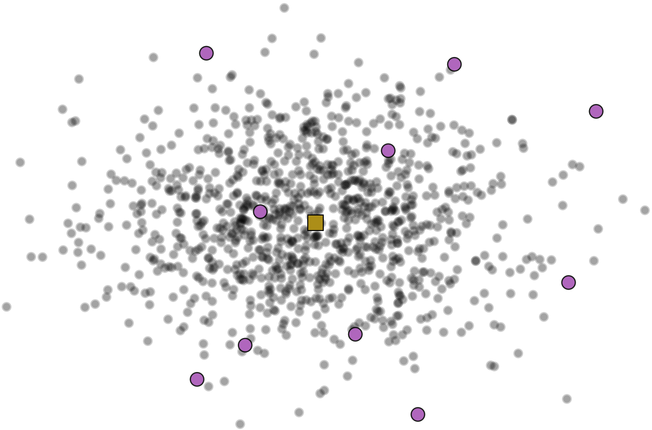


Figure: Example of DPP sampling

DPPs as a tool: an example

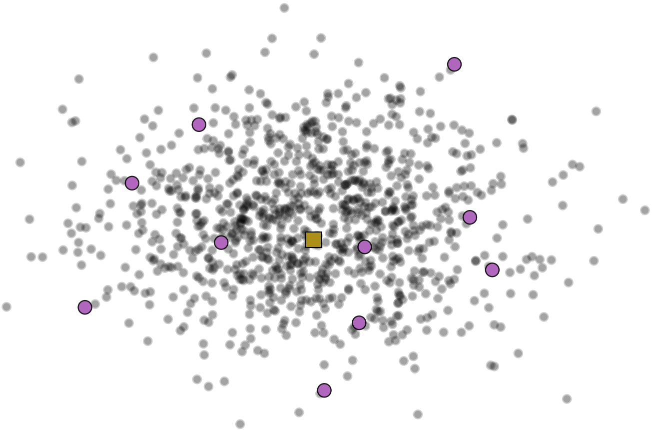


Figure: Example of DPP sampling

DPPs as a tool: an example

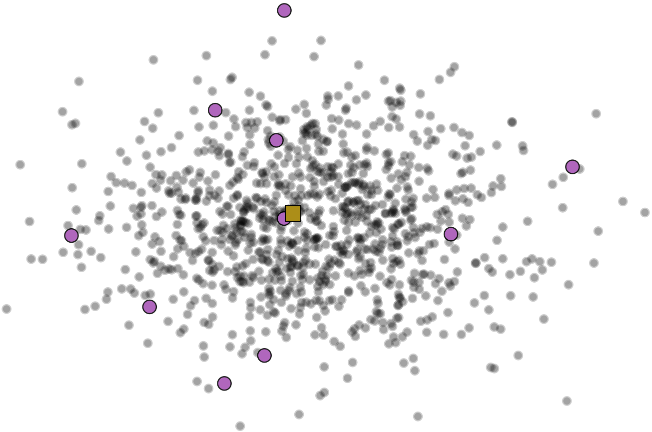


Figure: Example of DPP sampling

DPPs as a tool: an example

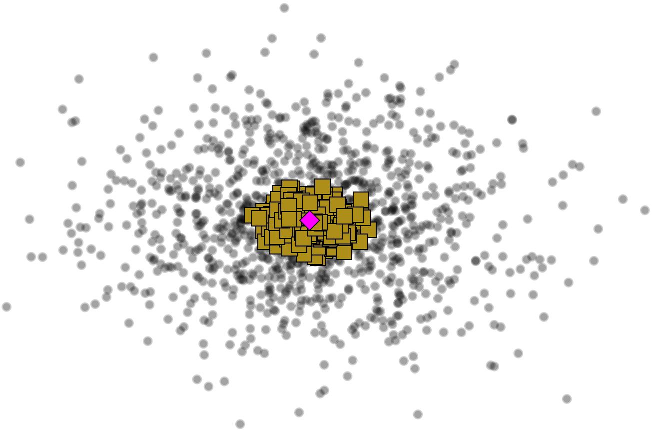


Figure: DPP estimations of the mean

DPPs as a tool: an example

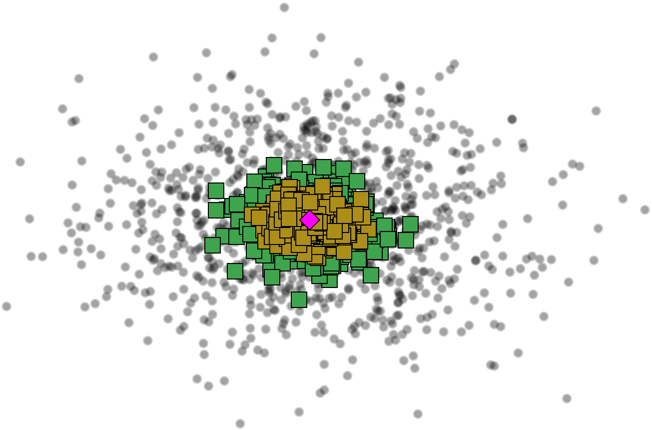


Figure: Comparison of both estimators: **variance reduction (here by a factor 3)**

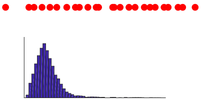
iii/ Finally, DPPs are used to *characterize* various phenomena.

iii/ Finally, DPPs are used to *characterize* various phenomena.

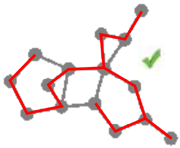
Where do DPPs
arise?

iii/ Finally, DPPs are used to *characterize* various phenomena.

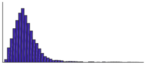
Where do DPPs
arise?



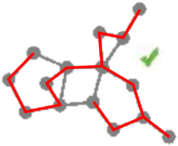
iii/ Finally, DPPs are used to *characterize* various phenomena.



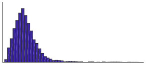
Where do DPPs arise?

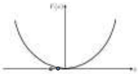


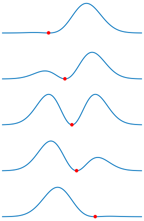
iii/ Finally, DPPs are used to *characterize* various phenomena.



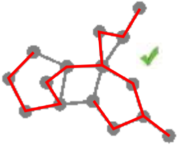
Where do DPPs arise?



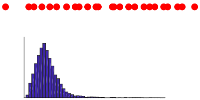


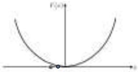


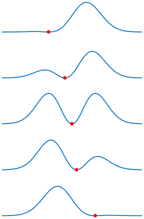
iii/ Finally, DPPs are used to *characterize* various phenomena.



Where do DPPs arise?







and more...

Eigenvalues of the Gaussian Unitary Ensemble[1]

- Consider a Hermitian matrix $\mathrm{H} \in \mathbb{C}^{n \times n}$ with
  - diagonal elements of the form $\mathrm{H}_{jj} = X$ with X drawn iid from $\mathcal{N}(0,1)$
  - off-diagonal elements of the form $\mathrm{H}_{jk} = X + iY$ with X and Y drawn iid from $\mathcal{N}(0,1/2)$.

[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

Eigenvalues of the Gaussian Unitary Ensemble[1]

- Consider a Hermitian matrix $\mathrm{H} \in \mathbb{C}^{n\times n}$ with
  - diagonal elements of the form $\mathrm{H}_{jj} = X$ with X drawn iid from $\mathcal{N}(0,1)$
  - off-diagonal elements of the form $\mathrm{H}_{jk} = X + iY$ with X and Y drawn iid from $\mathcal{N}(0,1/2)$.
- It has n real eigenvalues. They are distributed s.t.:

$$\mathbb{P}(\lambda_1,\dots,\lambda_n) \propto \exp^{-\sum_j \lambda_j^2} \prod_{j<k} (\lambda_j - \lambda_k)^2$$

[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

Eigenvalues of the Gaussian Unitary Ensemble[1]

- Consider a Hermitian matrix $\mathrm{H} \in \mathbb{C}^{n \times n}$ with
  - diagonal elements of the form $\mathrm{H}_{jj} = X$ with X drawn iid from $\mathcal{N}(0, 1)$
  - off-diagonal elements of the form $\mathrm{H}_{jk} = X + iY$ with X and Y drawn iid from $\mathcal{N}(0, 1/2)$.
- It has n real eigenvalues. They are distributed s.t.:

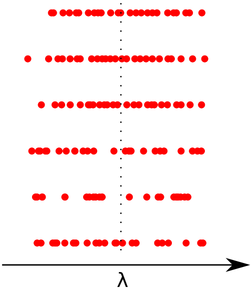
$$\begin{aligned} \mathbb{P}(\lambda_1, \dots, \lambda_n) &\propto \exp^{-\sum_j \lambda_j^2} \prod_{j<k} (\lambda_j - \lambda_k)^2 \\ &\propto \det \mathrm{M}^2 \end{aligned}$$

where $\mathrm{M}_{jk} = \lambda_k^{j-1} \exp^{-\frac{1}{2}\lambda_k^2}$.

[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

Eigenvalues of the GUE: illustration[1]

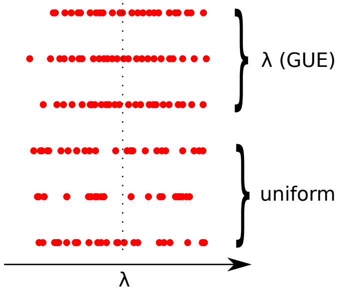
Examples of 6 point processes in 1D (3 GUE and 3 uniform):



[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

Eigenvalues of the GUE: illustration[1]

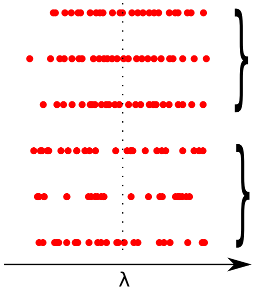
Examples of 6 point processes in 1D (3 GUE and 3 uniform):

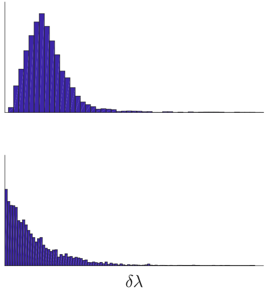


[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

Eigenvalues of the GUE: illustration[1]

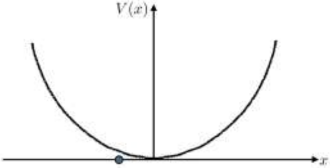
Examples of 6 point processes in 1D (3 GUE and 3 uniform):





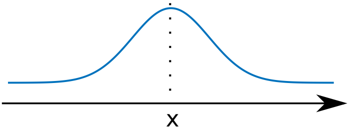
[1] see, *e.g.*, Johansson, *Random matrices and DPPs*, Arxiv (lecture notes), 2005

A spinless fermion in a harmonic potential[1]



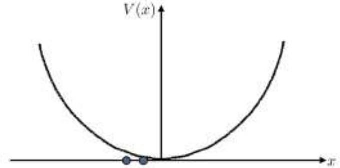
$$V(x) = \frac{1}{2}x^2$$

At temperature $T = 0$, the probability distribution of the particle is a simple Gaussian:



[1] Macchi, *The coincidence approach to stochastic point processes*. Adv. Appl. Probab., 1975

Two non-interacting fermions[1]



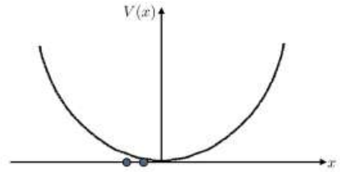
Pauli's exclusion principle implies, after a few calculations, that, at $T = 0$:

$$\mathbb{P}(x_1, x_2) \propto (x_2 - x_1)^2 e^{-(x_1^2 + x_2^2)}$$

$$\propto \left(\det \begin{bmatrix} e^{-\frac{1}{2}x_1^2} & e^{-\frac{1}{2}x_2^2} \\ x_1 e^{-\frac{1}{2}x_1^2} & x_2 e^{-\frac{1}{2}x_2^2} \end{bmatrix} \right)^2$$

[1] Macchi, *The coincidence approach to stochastic point processes*. Adv. Appl. Probab., 1975

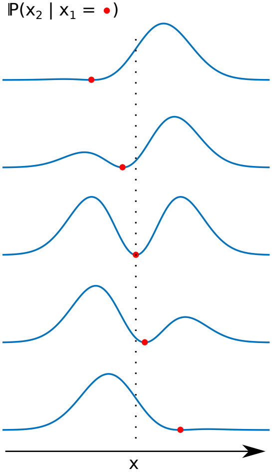
Two non-interacting fermions[1]



Pauli's exclusion principle implies, after a few calculations, that, at $T = 0$:

$$\mathbb{P}(x_1, x_2) \propto (x_2 - x_1)^2 e^{-(x_1^2 + x_2^2)}$$

$$\propto \left(\det \begin{bmatrix} e^{-\frac{1}{2}x_1^2} & e^{-\frac{1}{2}x_2^2} \\ x_1 e^{-\frac{1}{2}x_1^2} & x_2 e^{-\frac{1}{2}x_2^2} \end{bmatrix} \right)^2$$



[1] Macchi, *The coincidence approach to stochastic point processes*. Adv. Appl. Probab., 1975

Interim: repulsive point processes are hard

- There are many ways of defining point processes that feature repulsion; some may look much more natural than DPPs
- An unfortunate fact of point process theory is that repulsive point processes are *hard*, theoretically and empirically
- Desirable features:
  1. Probability density of p.p. is tractable (including normalisation constant)
  2. Inclusion probabilities (intensity functions) are tractable
  3. Sampling is tractable
  4. Model is easy to understand
- DPPs have properties (1-3) and arguably (4) once you get used to them
- Most other repulsive processes have one or two (at best)

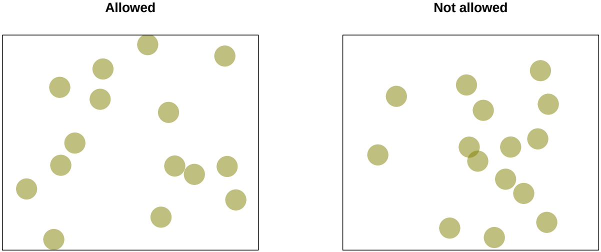
Gibbs point processes

- Many repulsive point processes can be described using the general framework of Gibbs point processes
- A Gibbs point process takes the following form:

$$p(\mathcal{X}) = \frac{\exp(-\beta \sum_{i<j} v(x_i, x_j))}{Z_\beta}$$

- $v(x_i, x_j)$ is called a *pairwise potential*
- the sum runs over all pairs of points
- example : $v(x_i, x_j) = d(x_i, x_j)$ where d is a distance, encourages points to be far apart.

The hard sphere model



The hard sphere model (AKA hard-core model) is used in physics to describe a set of particles that cannot overlap. See Löwen (2000) [1].

[1] Löwen, H. (2000). Fun with hard spheres. In Statistical physics and spatial statistics (pp. 295-331). Springer, Berlin, Heidelberg.

The hard sphere model

- We assume that $\mathcal{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, with m fixed and $\boldsymbol{x}_i \in [0,1]^d$
- The pairwise potential is simply:

$$v(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \infty & \text{if } \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2 < r \\ 0 & \text{otherwise} \end{cases}$$

Things to think about

- What's the normalisation constant for the hard-sphere model? Hint: can you relate it to the probability that m points sampled independently have a minimum pairwise distance $> r$?
- What are the valid configurations like when m is large?
- How would you sample from the hard-sphere model?

Normalisation constant

- Normalisation constant:

$$\int_{\Omega^m} \prod_{i<j} \mathbb{I}(\|\mathbf{x}_i - \mathbf{x}_j\|^2 > r) d\mathbf{x}_1 \dots d\mathbf{x}_m$$

- Intractable (except in dimension one)!

Packing limit

As m becomes large, we reach the packing limit, and most configurations are impossible

In the general case packing is a very hard problem (image from Wikipedia)

Sampling

- Possible sampling algorithm: "dart throwing".
- Pick a random initial location uniformly
- Pick a second location uniformly among remaining possible locations
- Pick a third location uniformly among remaining possible locations
- etc. until you have m spheres or further sampling is impossible (start again)
- Very good for small m, very bard for large m

Summary: the hard sphere model

- Simplest, most natural model you can imagine (property 4)
- But:
  1. Probability density is intractable (because normalisation constant is intractable for $d > 1$)
  2. Inclusion probabilities (intensity functions) are intractable for general domains, at least as far as we know
  3. Sampling is easy for small m (not very repulsive), then in large m becomes equivalent to the notoriously hard sphere packing problem

DPPs, the nitty-gritty

- We'll see that DPPs tick all boxes, contrary to most Gibbs processes
- The set-up cost is a bit higher; it's important to understand how these processes are defined, and to be careful about the notation
- We will now go through a few definitions in detail

Some notation for discrete point processes

- Ω is a base set of size n representing the items to sample from. w.l.o.g we may take $\Omega = \{1, \ldots, n\}$
- $\mathcal{X}$ is a random subset of Ω
- We note $m = |\mathcal{X}|$, which may be a random variable

L-ensembles

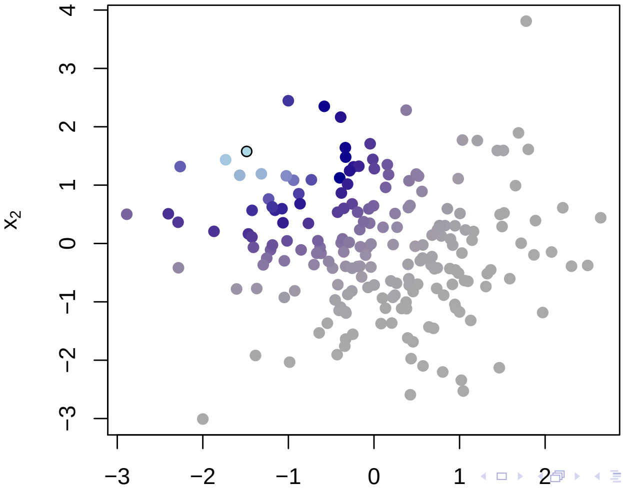
- The repulsion in DPPs is based on a notion of similarity between items in Ω.
- The similarity between all pairs of items in Ω is stored in a $n \times n$ matrix called (for historical reasons) the "L-ensemble".
- We note this matrix $\mathbf{L}$, with L_{ij} the similarity between items i and j
- $\mathbf{L}$ is assumed to be positive definite.

L-ensembles

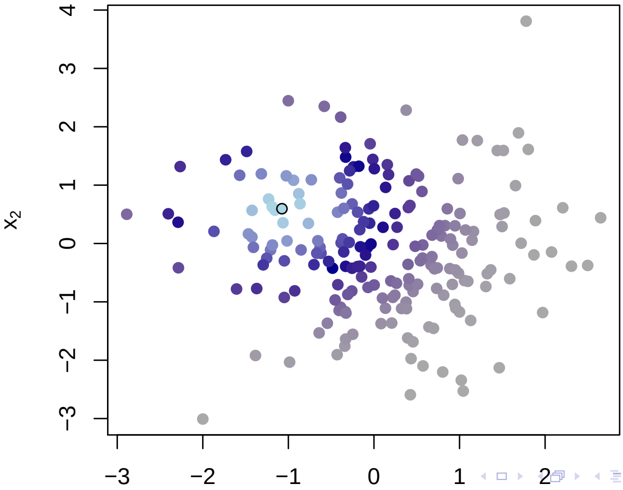
- We'll come across several ways of constructing the **L** matrix.
- For now, assume that the items are vectors in $\mathbb{R}^d$. We can use a kernel function to describe similarity.
- Example: Gaussian kernel

$$L_{ij} = \exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2\right)$$

Similarity via the Gaussian kernel



Similarity via the Gaussian kernel



DPP: formal definition

- We say that $\mathcal{X}$ (random set) is distributed according to a DPP if:

$$p(\mathcal{X} = X) \propto \det \mathbf{L}_X$$

- $\mathbf{L}_{\mathcal{X}}$ is the restriction of $\mathbf{L}$ to the items in $\mathcal{X}$
- IMPORTANT!!!! Here the number of items in $\mathcal{X}$, $m = |\mathcal{X}|$, is not fixed and may therefore vary.

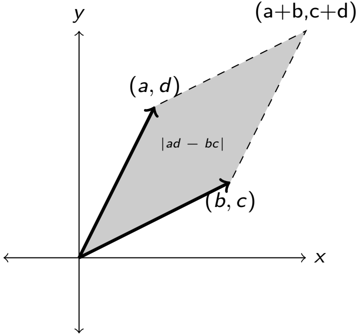
A closer look

- The probability mass function is fairly simple:

$$p(\mathcal{X} = X) \propto \det \mathbf{L}_X$$

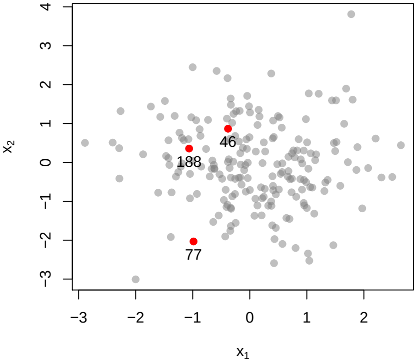
- $\det \mathbf{L}_{\mathcal{X}} \geqslant 0$, by positive-definiteness of $\mathbf{L}$
- In addition: $\sum_{\mathcal{X}} \det \mathbf{L}_{\mathcal{X}} = \det(\mathbf{L} + \mathbf{I})$ is the normalisation constant (tractable!)
- So why does this induce repulsion?

Determinants: geometric interpretation



Determinants measure the (signed) volume of the paralleliped spanned by the columns of a matrix. Illustration by Yigit Pilavci.

Why does the determinant induce repulsion?

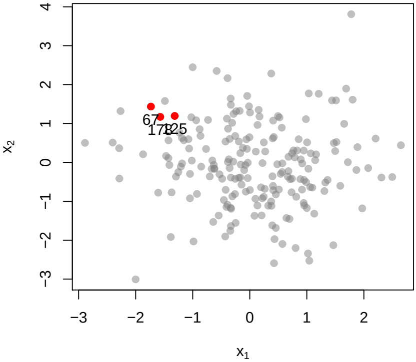


$$\mathbf{L}_{\mathcal{X}} =$$

	46	77	188
46	1.00	0.01	0.70
77	0.01	1.00	0.06
188	0.70	0.06	1.00

Determinant: 0.51.

Why does the determinant induce repulsion?



$$\mathbf{L}_{\mathcal{X}} = \begin{array}{cccc} & 67 & 178 & 125 \\ 67 & 1.00 & 0.95 & 0.89 \\ 178 & 0.95 & 1.00 & 0.97 \\ 125 & 0.89 & 0.97 & 1.00 \end{array}$$

Determinant: 0.005.

Inclusion probabilities

- Are certain, or pairs of items are more likely to be sampled?
- Formally: let $\mathcal{S}$ denote a fixed (non-random) set. The "inclusion probabilities" are of the form:

$$p(\mathcal{S} \subseteq \mathcal{X})$$

- If $\mathcal{S} = \{i\}$, a singleton, equivalent to $p(i \in \mathcal{X})$, the probability that item i is sampled
- If $\mathcal{S} = \{i, j\}$, a pair, equivalent to $p(i \in \mathcal{X}$ and $j \in \mathcal{X})$, the probability that both items are sampled

Marginal kernels

- In DPPs the inclusion probabilities are quite remarkable
- For a DPP with L-ensemble $\mathbf{L}$ the inclusion probabilities are as follows

$$p(\mathcal{S} \subseteq \mathcal{X}) = \det \mathbf{K}_{\mathcal{S}}$$

where:

$$\mathbf{K} = \mathbf{L}(\mathbf{L} + \mathbf{I})^{-1}$$

- $\mathbf{K}$ is called the *marginal kernel* of the DPP

L-ensemble vs. marginal kernel

Example.

$$\mathbf{L} = \begin{pmatrix} 1 & 0.946 & 0.681 & 0.634 & 0.611 \\ 0.946 & 1 & 0.864 & 0.825 & 0.805 \\ 0.681 & 0.864 & 1 & 0.997 & 0.993 \\ 0.634 & 0.825 & 0.997 & 1 & 0.999 \\ 0.611 & 0.805 & 0.993 & 0.999 & 1 \end{pmatrix}$$

can be used to compute $p(\mathcal{X} = X)$.

$$\mathbf{K} = \mathbf{L}(\mathbf{L} + \mathbf{I})^{-1} = \begin{pmatrix} 0.328 & 0.246 & 0.075 & 0.053 & 0.042 \\ 0.246 & 0.234 & 0.135 & 0.117 & 0.108 \\ 0.075 & 0.135 & 0.206 & 0.210 & 0.212 \\ 0.053 & 0.117 & 0.210 & 0.219 & 0.223 \\ 0.042 & 0.108 & 0.212 & 0.223 & 0.227 \end{pmatrix}$$

can be used to compute $p(\mathcal{S} \in \mathcal{X})$

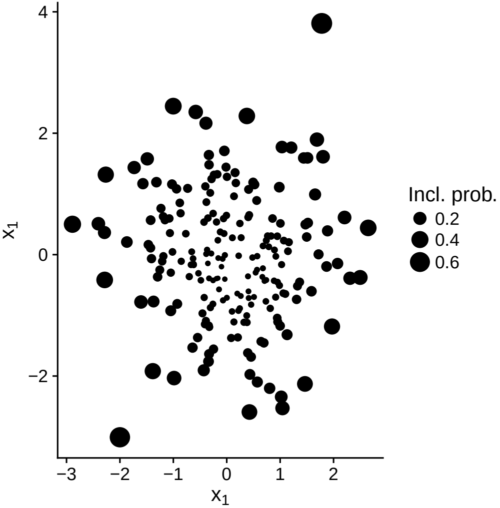
First-order inclusion probabilities

- First-order inclusion probabilities are just:

$$p(i \in \mathcal{X}) = K_{ii}$$

- Exercise: work out $E(|\mathcal{X}|)$
- Hint: $|\mathcal{X}| = \sum_{j \in \Omega} \mathbb{I}(j \in \mathcal{X})$

First-order inclusion probabilities are (generally) not uniform!



Radius prop. to $p(i \in \mathcal{X}) = K_{ii}$

Second-order inclusion probabilities

- Note $\pi_i = p(i \in \mathcal{X})$
- Poisson sampling : go through all n items and include item i with probability π_i *independently*
- Exercise: let $\mathcal{Y}$ be a Poisson sample with the same first-order inclusion probabilities as $\mathcal{X}$. Compute $p(i, j \subseteq \mathcal{Y})$. Compare to $p(i, j \subseteq \mathcal{X})$: how does repulsion manifest itself?

Fixed-size DPPs

- Often it's preferable to set the size of $\mathcal{X}$ to a fixed value.
- A fixed-size DPP is a DPP, conditioned on $|\mathcal{X}| = m$. They were introduced by Kulesza & Taskar as "k-DPPs". Here we call them "m-DPPs" for consistency.
- Def. $\mathcal{X}$ is a m-DPP with L-ensemble $\mathbf{L}$ if

$$p(\mathcal{X}) = \begin{cases} \frac{\det \mathbf{L}_{\mathcal{X}}}{e_m(\mathbf{L})} & \text{if } |\mathcal{X}| = m \\ 0 & \text{otherwise} \end{cases}$$

- $e_m(\mathbf{L})$ is the normalisation constant, and is easy to compute from the spectrum of $\mathbf{L}$.
- Otherwise an m-DPP is very similar to a DPP: we're simply forbidding sets of a size smaller or greater than m

Inclusion probabilities in m-DPPs

- The bad news: m-DPPs do not, in general, have a marginal kernel, i.e. there may not be a matrix **K** such that

$$p(\mathcal{S} \subseteq \mathcal{X}) = \det \mathbf{K}_{\mathcal{S}}$$

  when $\mathcal{S}$ is a m-DPP.
- Exact inclusion probabilities are tricky to compute, especially for $|\mathcal{S}| > 1$

Inclusion probabilities in m-DPPs

- The good news: we showed in Barthelmé, Tremblay, Amblard (2019) that there is an approximate marginal kernel, i.e. for large n and small $|\mathcal{S}|$, there's a matrix $\tilde{\mathbf{K}}$ such that

$$p(\mathcal{S} \subseteq \mathcal{X}) \approx \det \tilde{\mathbf{K}}_{\mathcal{S}}$$

- $\tilde{\mathbf{K}}$ is easy to compute:

$$\tilde{\mathbf{K}} = \alpha \mathbf{L} (\alpha \mathbf{L} + \mathbf{I})^{-1}$$

where α is such that $\mathrm{Tr}\, \tilde{\mathbf{K}} = m$

Projection DPPs

- m-DPPs do not have exact marginal kernels, with one very important exception
- If $m = r = \operatorname{rank} \mathbf{L}$, then *there is* an exact marginal kernel, with a very specific form
- Let $\mathbf{L} = \mathbf{U}\mathbf{D}\mathbf{U}^t$, the eigendecomposition of $\mathbf{L}$, and $\mathbf{D}$ the $r \times r$ matrix of eigenvalues.
- The marginal kernel is simply $\mathbf{K} = \mathbf{U}\mathbf{U}^t$, a projection matrix ($\mathbf{K}^2 = \mathbf{K}$)
- Accordingly these DPPs are called *projection DPPs*.
- In a sense they are both DPPs and m-DPPs
- They are *central* to the overall theory

An example of a projection DPP

- Here's an example of how to build a projection DPP. Assume the items are just points along a line: $x_1, \dots, x_n$.
- We build a matrix of polynomial features:

$$\mathbf{M} = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{r-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{r-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{r-1} \end{pmatrix}$$

- We build an L-ensemble based on those features:

$$\mathbf{L} = \mathbf{M}\mathbf{M}^t$$

- $\mathbf{L}$ has rank r and dimension $n \times n$
- If we set $m = r$, ie. we sample as many points as we have polynomial features, than what we have is a projection DPPs.

Summary so far

- DPPs have tractable inclusion probabilities, *but* the number of items sampled is random (in general)
- m-DPPs have fixed sample size, *but* the inclusion probabilities are less tractable
- One exception: projection DPPs have fixed sample size, *and* the inclusion probabilities are tractable

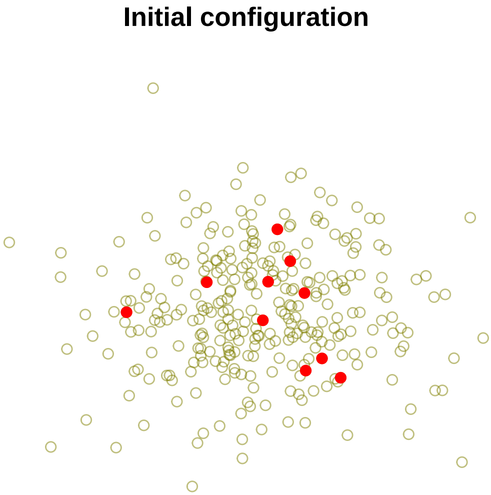
Some computational issues

- There's a few computational issues, but we'll look at the two main ones:
    1. How to sample from a DPP efficently
    2. How to create an L-ensemble efficiently
- We can't cover the theory in detail so focus is on practical aspects
- See our package DPP.jl for efficient Julia implementation; DPPy by Guillaume Gautier for Python tools

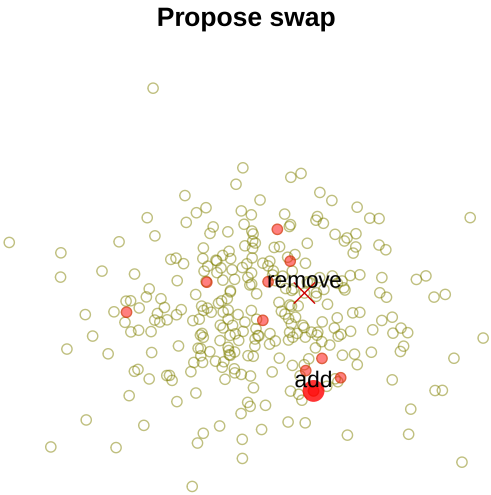
Samplers for DPPs

- For DPPs there are both exact and inexact samplers
- The inexact samplers (eg. Gibbs sampler) use an MCMC chain to generate approximate samples cheaply.
- However getting an exact sample is often not much more expensive: we will describe a method based on Hough et al. (2006)

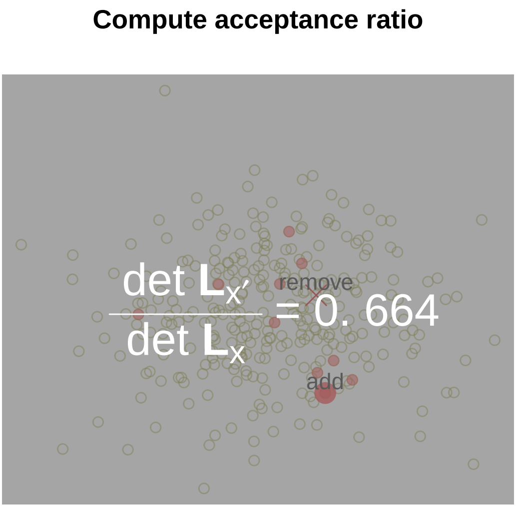
A Metropolis-Hastings sampler



A Metropolis-Hastings sampler

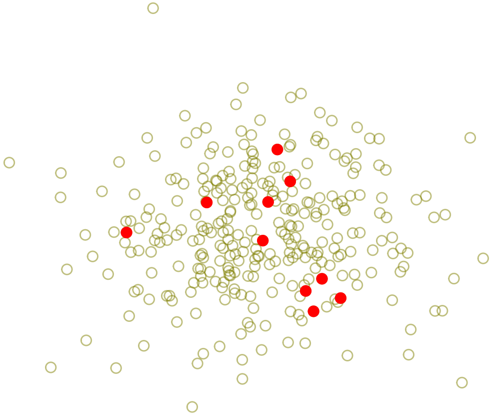


A Metropolis-Hastings sampler



A Metropolis-Hastings sampler

Accept or reject new configuration



A Metropolis-Hastings sampler

Initialisation: set $\mathcal{X}$ to some random subset of size m. For $t = 1 to T$, do:

- Propose swap: construct $\mathcal{X}'$ by removing random item from $\mathcal{X}$, adding a random item from $\Omega - \mathcal{X}$
- Evaluate acceptance ratio $r = \frac{\det \mathbf{L}_{\mathcal{X}'}}{\det \mathbf{L}_{\mathcal{X}}}$
- Set $\mathcal{X} \leftarrow \mathcal{X}'$ with probability r.

If T is large enough, the final configuration is an almost-exact sample from an m-DPP with ensemble $\mathbf{L}$

A Metropolis-Hastings sampler

- The sampler we've just described is really easy to implement!
- Feel free to try it for yourself after the tutorial, should just take a few minutes
- Bonus points if you can adapt it to DPPs and not just m-DPPs
- For most practical purposes we recommend the *exact* sampler we describe next

The direct sampler

- It turns out that sampling from a *projection* DPP is easy
- The algorithm just picks points sequentially from the appropriate probability distribution
- For generic DPPs, we'll see that it's possible to reduce the problem to the sampling of a projection DPP

Sampling sequentially

- Take a set of m items $\mathcal{X} = \{x_1, \dots, x_m\}$ and order it (arbitrarily) into a sequence $x_1, \dots, x_m$
- Our goal is to sample $\mathcal{X} \sim proj - DPP(K)$ by sampling first x_1, then x_2, then x_3 etc. up to a x_m
- Formally:

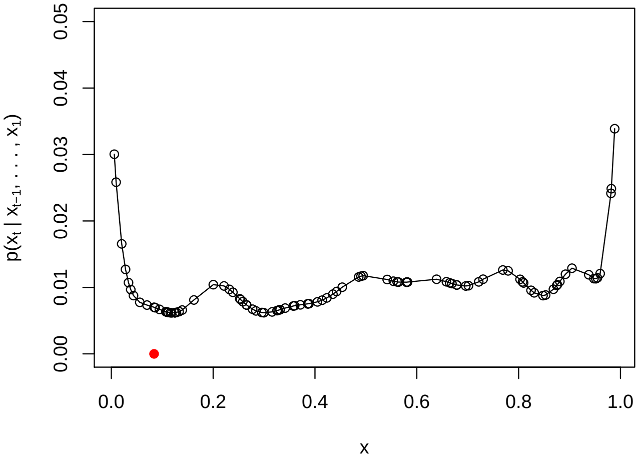
$$
\begin{aligned}
x_1 &\sim p(x_1) \\
x_2 &\sim p(x_2 | x_1) \\
x_3 &\sim p(x_3 | x_1, x_2) \\
&\vdots
\end{aligned}
$$

What are these conditional distributions?

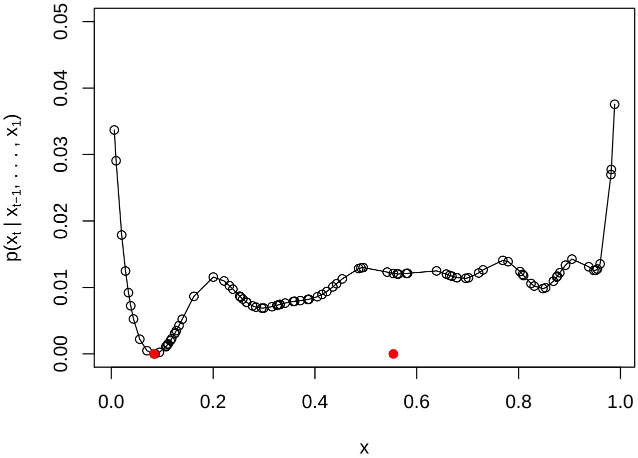
- $p(x_1)$ is the distribution of an arbitrary item taken from a projection DPP - that's just the inclusion probability
- $p(x_2|x_1)$ is the distribution of an arbitrary item taken from a projection DPP, given that item x_1 is in the set. That's a conditional inclusion probability.
- etc.
- As it turns out, these distributions are tractable in *proj-DPPs*, and this leads to an algorithm that is both easy to implement and fast[2]
- Nice bit of theory: conditional distribution of x_t equals the conditional variance of a Gaussian process with the same kernel sampled at $x_1 \dots x_{t-1}$!

[2]Alg. due to Hough et al. (2006), Gillenwater (2014) for a faster version. See DDPy documentation by G. Gautier for more

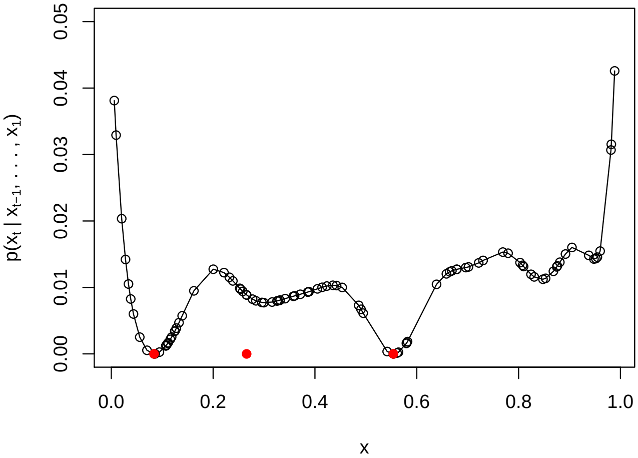
The direct sampling algorithm in action



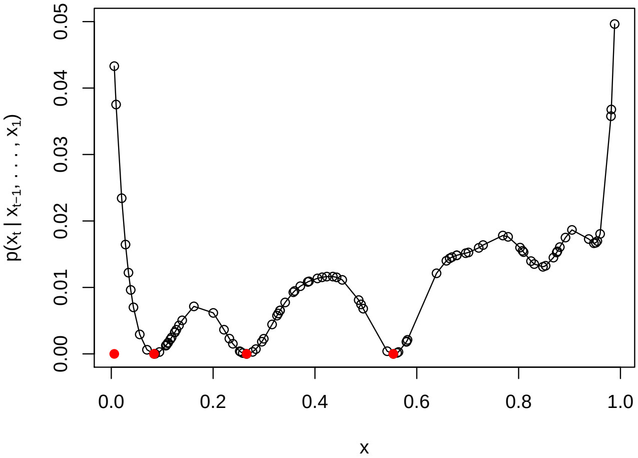
The direct sampling algorithm in action



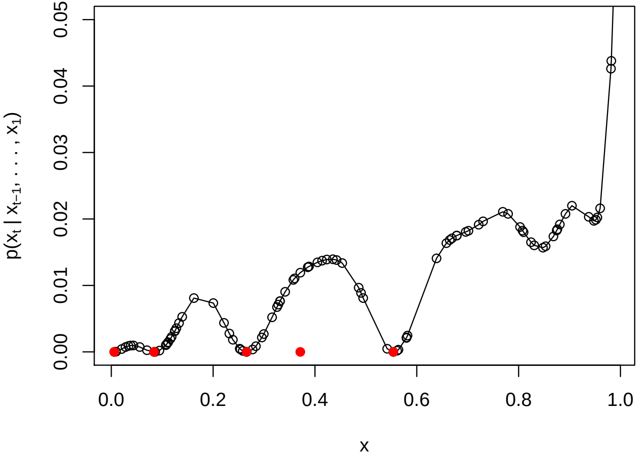
The direct sampling algorithm in action



The direct sampling algorithm in action



The direct sampling algorithm in action



Sampling generic DPPs

- At this point we know how to sample from projection DPPs
- Now we need to sample from generic (m)-DPPs
- Luckily, we can show that generic (m)-DPPs are just mixtures of projection DPPs
- Recall: to sample from a mixture of Gaussians, pick randomly one of the Gaussians then sample from that Gaussian.
- Same here: we'll have to form a random projection DPP, then sample from that projection DPP

Cauchy-Binet lemma

- We'll sketch the proof that all DPPs are mixtures of projection DPPs.
- Central ingredient is the Cauchy-Binet lemma.
- Let $\mathbf{A}_{m\times n}$ and $\mathbf{B}_{n\times m}$, with $n \geqslant m$. We seek to compute $\det \mathbf{AB}$.
- If $n = m$ $\mathbf{A}$ and $\mathbf{B}$ are square, and so $\det \mathbf{AB} = \det \mathbf{A} \det \mathbf{B}$. Cauchy-Binet generalises this formula to $n > m$.

$$\det \mathbf{AB} = \sum_{|\mathcal{Y}|=m} \det \mathbf{A}_{:,\mathcal{Y}} \det \mathbf{B}_{\mathcal{Y},:}$$

- Here $\mathcal{Y}$ is a subset of $1, 2, \ldots, n$ of size m and the sum runs over all such subsets.

Proof sketch that DPPs are mixtures of projection DPPs

Consider the eigendecomposition of $\mathbf{L}$, $\mathbf{L} = \mathbf{UDU}^t$, and the probability of set $\mathcal{X}$.

$$p(\mathcal{X}) \propto \det \mathbf{L}_{\mathcal{X}} = \det \mathbf{U}_{\mathcal{X},:} \mathbf{D}^{\frac{1}{2}} \mathbf{D}^{\frac{1}{2}} \mathbf{U}^t_{:,\mathcal{X}} = (\det \mathbf{AB})$$

Apply Cauchy-Binet:

$$p(\mathcal{X}) \propto \sum_{|\mathcal{Y}|=|\mathcal{X}|} \det \mathbf{U}_{\mathcal{X},\mathcal{Y}} \mathbf{D}^{\frac{1}{2}}_{\mathcal{Y},\mathcal{Y}} \det \mathbf{D}^{\frac{1}{2}}_{\mathcal{Y},\mathcal{Y}} \mathbf{U}^t_{\mathcal{Y},\mathcal{X}}$$

Now we have square matrices inside the sum, so this is just:

$$p(\mathcal{X}) \propto \sum_{|\mathcal{Y}|=|\mathcal{X}|} \det \mathbf{U}_{\mathcal{X},\mathcal{Y}} \mathbf{U}^t_{\mathcal{Y},\mathcal{X}} \det \mathbf{D}_{\mathcal{Y},\mathcal{Y}}$$

Proof sketch that DPPs are mixtures of projection DPPs

Looking at:

$$p(\mathcal{X}) \propto \sum_{|\mathcal{Y}|=|\mathcal{X}|} \det \mathbf{U}_{\mathcal{X},\mathcal{Y}} \mathbf{U}^t_{\mathcal{Y},\mathcal{X}} \det \mathbf{D}_{\mathcal{Y},\mathcal{Y}}$$

with $\mathcal{X}$ as a variable, we see the following structure appearing:

$$p(\mathcal{X}) \propto \sum_{|\mathcal{Y}|=|\mathcal{X}|} f(\mathcal{X}|\mathcal{Y}) g(\mathcal{Y})$$

which expresses $p(\mathcal{X})$ as a marginal! Here $f(\mathcal{X}|\mathcal{Y}) = \det \mathbf{U}_{\mathcal{X},\mathcal{Y}} \mathbf{U}^t_{\mathcal{Y},\mathcal{X}}$, and that's a projection DPP where we select the eigenvectors given by $\mathcal{Y}$ to form the L-ensemble. $g(\mathcal{Y}) = \det(\mathbf{D}_{\mathcal{Y}})$ is also a DPP, this time with a diagonal L-ensemble!

Computational considerations

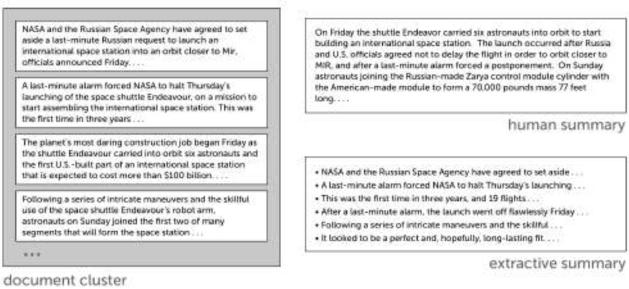
- A tally of computational costs:
  1. We need to generate the **L** matrix at cost $\mathcal{O}(n^2)$
  2. We need to compute the eigendecomposition of **L** at cost $\mathcal{O}(n^3)$
  3. We need to sample at cost $\mathcal{O}(nk^2)$

Computational considerations

- Overall the dominating cost is the eigendecomposition at cost $\mathcal{O}(n^3)$
- Fortunately that cost can be brought down to $\mathcal{O}(nk^2)$ if you design $\mathbf{L}$ to have rank $\mathcal{O}(k)$
- For example: use k polynomial features, or use Random Fourier Features
- See Tremblay et al. (2018) on coresets [3] for a list of tricks
- In a nutshell: DPPs have very good (linear) scaling in n, meaning the original set can be in the millions
- However, poor scaling in k, so that the subset you sample will be in the hundreds

[3] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

Example of application: generate extractive summaries[1]



- The trick is to find a good feature space to embed sentences and a proper DPP kernel
- Both can be parametrized (tf, idf, etc.) and then learned[2]

[2] *e.g.*, Kulesza et al, *Near-optimal map inference for DPPs*, NIPS 2012.

[1] Kulesza and Taskar, *DPPs for machine learning*, Found. and Trends in ML, 2013

Example of application: search algorithms[1]

[1] Kulesza and Taskar, *DPPs for machine learning*, Found. and Trends in ML, 2013

"DDPs as a tool" applications

- Monte-Carlo integration [1] [2]:

$$\int f(x)\mu(dx) \simeq \sum_{n=1}^{N} \omega_n f(x_n)$$

where the x_i's are the so-called quadratic nodes.

- Mini-batch sampling for stochastic gradient descent [3]:

$$L(\theta) = \sum_i L_i(\theta)$$

$$\text{GD} : \theta \leftarrow \theta - \eta \nabla L(\theta) = \theta - \eta \sum_i \nabla L_i(\theta)$$

$$\text{mini-batch GD} : \theta \leftarrow \theta - \eta \sum_{i \in \mathcal{X}} \nabla L_i(\theta)$$

- Column subset selection problem for best rank-k approximation [4]

[1] Gautier et al., *On two ways to use DPPs for Monte Carlo integration*, ICML, 2019.
[2] Bardenet et al., *Monte Carlo with DPPs*, Annals of Applied Probability, In Press.
[3] Zhang et al., *DPPs for Mini-Batch Diversification*, UAI, 2017.
[4] Belhadji et al, *A DPP for column subset selection*, Arxiv, 2018.

Zoom on one application:

- Coresets[1]

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

Coresets

- Consider a dataset $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, say: n points in dimension d.
- Let Θ be a parameter space and consider cost functions of the form:

$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} f(\boldsymbol{x}_i, \theta)$$

  where $f : \mathcal{X} \to \mathbb{R}^+$, and $\theta \in \Theta$.
- A classical ML objective: find

$$\theta^* = \operatorname*{argmin}_{\theta \in \Theta} L(\mathcal{X}, \theta).$$

- k-means, k-medians, linear/logistic regressions fall in this class of problems

Coresets

- Consider a subset $\mathcal{S} \subset \mathcal{X}$ (possibly with repetitions)
- Associate a weight $\omega_s > 0$ to each element $\boldsymbol{s} \in \mathcal{S}$
- Define

$$\hat{L}(\mathcal{S}, \theta) = \sum_{\boldsymbol{s} \in \mathcal{S}} \omega_s f(\boldsymbol{s}, \theta)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall\theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \underset{\theta\in\Theta}{\text{argmin}}\, \hat{L}(\mathcal{S},\theta).$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \operatorname*{argmin}_{\theta \in \Theta} \hat{L}(\mathcal{S},\theta).$$

- Why are coresets interesting?

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \operatorname*{argmin}_{\theta\in\Theta} \hat{L}(\mathcal{S},\theta).$$

- Why are coresets interesting?

$$\hat{L}(\mathcal{S},\hat{\theta}^*)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \operatorname*{argmin}_{\theta\in\Theta} \hat{L}(\mathcal{S},\theta).$$

- Why are coresets interesting?

$$\hat{L}(\mathcal{S},\hat{\theta}^*) \leqslant \hat{L}(\mathcal{S},\theta^*)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall\theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \operatorname*{argmin}_{\theta\in\Theta} \hat{L}(\mathcal{S},\theta).$$

- Why are coresets interesting?

$$\hat{L}(\mathcal{S},\hat{\theta}^*) \leqslant \hat{L}(\mathcal{S},\theta^*) \leqslant (1+\epsilon)L(\mathcal{X},\theta^*)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon)L(\mathcal{X},\theta) \leqslant \hat{L}(\mathcal{S},\theta) \leqslant (1+\epsilon)L(\mathcal{X},\theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \operatorname*{argmin}_{\theta\in\Theta} \hat{L}(\mathcal{S},\theta).$$

- Why are coresets interesting?

$$(1-\epsilon)L(\mathcal{X},\hat{\theta}^*) \leqslant \hat{L}(\mathcal{S},\hat{\theta}^*) \leqslant \hat{L}(\mathcal{S},\theta^*) \leqslant (1+\epsilon)L(\mathcal{X},\theta^*)$$

Coresets

- $\mathcal{S}$ is an ϵ-coreset of $\mathcal{X}$ wrt L if:

$$\forall \theta \in \Theta \qquad (1-\epsilon) L(\mathcal{X}, \theta) \leqslant \hat{L}(\mathcal{S}, \theta) \leqslant (1+\epsilon) L(\mathcal{X}, \theta)$$

- *Multiplicative* approximation: gold standard of approximation methods
- Denote by $\hat{\theta}^*$ the argmin of $\hat{L}$:

$$\hat{\theta}^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \, \hat{L}(\mathcal{S}, \theta).$$

- Why are coresets interesting?

$$(1-\epsilon) L(\mathcal{X}, \theta^*) \leqslant (1-\epsilon) L(\mathcal{X}, \hat{\theta}^*) \leqslant \hat{L}(\mathcal{S}, \hat{\theta}^*) \leqslant \hat{L}(\mathcal{S}, \theta^*) \leqslant (1+\epsilon) L(\mathcal{X}, \theta^*)$$

Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$

Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:

$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$

Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:

$$L(\mathcal{X},\theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$

- Optimal θ:

$$\theta^* = \operatorname*{argmin}_{\theta\in\Theta} L(\mathcal{X},\theta)$$

Coresets: illustration on the 1-means problem

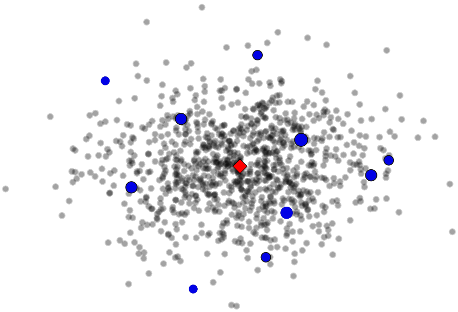
- Data $\mathcal{X}$
- Cost function:

$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$

- Optimal θ:

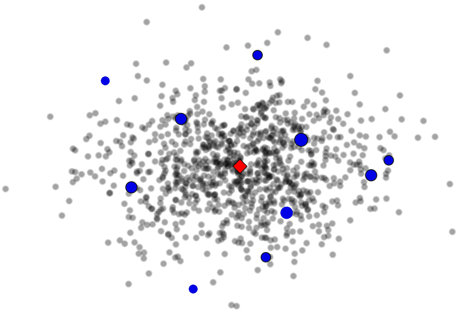
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \, L(\mathcal{X}, \theta)$$

- A weighted subset $\mathcal{S}$



Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:
$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$
- Optimal θ:
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\mathcal{X}, \theta)$$
- A weighted subset $\mathcal{S}$
- Estimated cost function:
$$\hat{L}(\mathcal{S}, \theta) = \sum_{\boldsymbol{s} \in \mathcal{S}} \omega_s \|\boldsymbol{s} - \theta\|^2$$



Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:

$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\mathbf{x}_i - \theta\|^2$$

- Optimal θ:

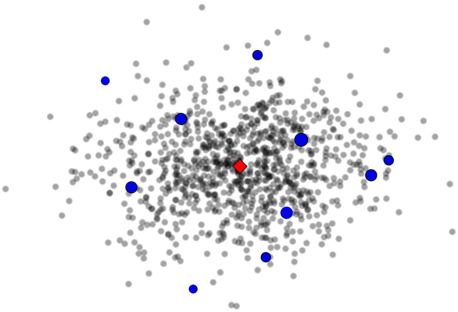
$$\theta^* = \operatorname*{argmin}_{\theta \in \Theta} L(\mathcal{X}, \theta)$$

- A weighted subset $\mathcal{S}$
- Estimated cost function:

$$\hat{L}(\mathcal{S}, \theta) = \sum_{\mathbf{s} \in \mathcal{S}} \omega_s \|\mathbf{s} - \theta\|^2$$

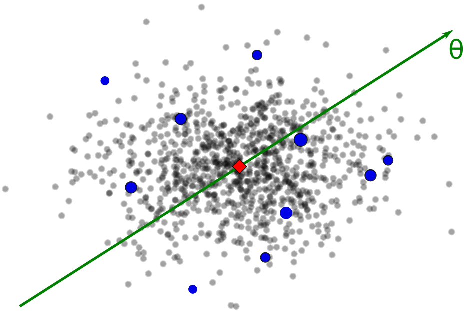
- $\mathcal{S}$ is a ϵ-coreset if:

$$\forall \theta \quad \left| \frac{\hat{L}}{L} - 1 \right| \leqslant \epsilon$$



Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:
$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$
- Optimal θ:
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\mathcal{X}, \theta)$$
- A weighted subset $\mathcal{S}$
- Estimated cost function:
$$\hat{L}(\mathcal{S}, \theta) = \sum_{\boldsymbol{s} \in \mathcal{S}} \omega_s \|\boldsymbol{s} - \theta\|^2$$
- $\mathcal{S}$ is a ϵ-coreset if:
$$\forall \theta \quad \left| \frac{\hat{L}}{L} - 1 \right| \leqslant \epsilon$$



Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:

$$L(\mathcal{X},\theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$

- Optimal θ:

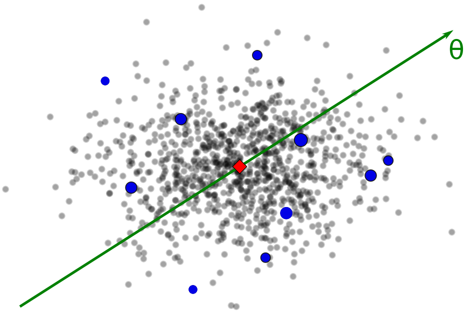
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\mathcal{X},\theta)$$

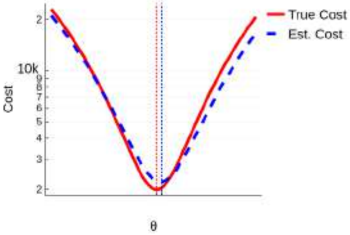
- A weighted subset $\mathcal{S}$
- Estimated cost function:

$$\hat{L}(\mathcal{S},\theta) = \sum_{\boldsymbol{s} \in \mathcal{S}} \omega_s \|\boldsymbol{s} - \theta\|^2$$

- $\mathcal{S}$ is a ϵ-coreset if:

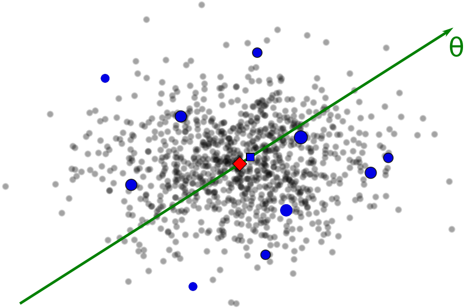
$$\forall\theta \quad \left|\frac{\hat{L}}{L} - 1\right| \leqslant \epsilon$$

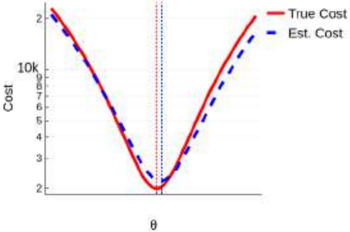




Coresets: illustration on the 1-means problem

- Data $\mathcal{X}$
- Cost function:
$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$
- Optimal θ:
$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(\mathcal{X}, \theta)$$
- A weighted subset $\mathcal{S}$
- Estimated cost function:
$$\hat{L}(\mathcal{S}, \theta) = \sum_{\boldsymbol{s} \in \mathcal{S}} \omega_s \|\boldsymbol{s} - \theta\|^2$$
- $\mathcal{S}$ is a ϵ-coreset if:
$$\forall \theta \quad \left| \frac{\hat{L}}{L} - 1 \right| \leqslant \epsilon$$
- Estimated optimal θ:
$$\hat{\theta}^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \hat{L}(\mathcal{S}, \theta)$$





Random coresets

- Random context: suppose $\mathcal{S}$ is a *random* subset $\mathcal{S} \subset \mathcal{X}$ (possibly with repetitions)
- Importance sampling notations:
  - Define ϵ_i the *random variable* counting the number of times $\boldsymbol{x}_i$ is in $\mathcal{S}$
  - To each element x_i associate a weight $\omega_i = \frac{1}{\mathbb{E}(\epsilon_i)}$
- One has:

$$\hat{L}(\mathcal{S}, \theta) = \sum_{i=1}^{n} f(\boldsymbol{x}_i, \theta) \frac{\epsilon_i}{\mathbb{E}(\epsilon_i)}$$

and thus $\hat{L}$ is an unbiased estimator of L:

$$\mathbb{E}\left(\hat{L}(\mathcal{S}, \theta)\right) = \sum_{i=1}^{n} f(\boldsymbol{x}_i, \theta) = L(\mathcal{X}, \theta).$$

Sensitivity

- The **sensitivity** of a datapoint $\boldsymbol{x}_i \in \mathcal{X}$ with respect to $f : \mathcal{X}, \Theta \rightarrow \mathbb{R}^+$ is:

$$\sigma_i = \max_{\theta \in \Theta} \frac{f(\boldsymbol{x}_i, \theta)}{L(\mathcal{X}, \theta)} \qquad \in [0, 1].$$

Also, the total sensitivity is defined as $\mathfrak{S} = \sum_{i=1}^{n} \sigma_i$.

Sensitivity

- The **sensitivity** of a datapoint $\boldsymbol{x}_i \in \mathcal{X}$ with respect to $f : \mathcal{X}, \Theta \to \mathbb{R}^+$ is:

$$\sigma_i = \max_{\theta \in \Theta} \frac{f(\boldsymbol{x}_i, \theta)}{L(\mathcal{X}, \theta)} \qquad \in [0, 1].$$

  Also, the total sensitivity is defined as $\mathfrak{S} = \sum_{i=1}^{n} \sigma_i$.
- In general, the sensitivity is unknown analytically.

Sensitivity

- The **sensitivity** of a datapoint $\boldsymbol{x}_i \in \mathcal{X}$ with respect to $f : \mathcal{X}, \Theta \to \mathbb{R}^+$ is:

$$\sigma_i = \max_{\theta \in \Theta} \frac{f(\boldsymbol{x}_i, \theta)}{L(\mathcal{X}, \theta)} \qquad \in [0, 1].$$

  Also, the total sensitivity is defined as $\mathfrak{S} = \sum_{i=1}^n \sigma_i$.
- In general, the sensitivity is unknown analytically.

- 1-means is an exception. In this case, supposing wlog that the data is centered (*i.e.*: $\sum_j x_j = 0$), one shows:

$$\sigma_i = \frac{1}{n} \left(1 + \frac{\|x_i\|^2}{v} \right),$$

  where $v = \frac{1}{n} \sum_{x \in \mathcal{X}} \|x\|^2$. Thus, $\mathfrak{S} = 2$.

Sensitivity

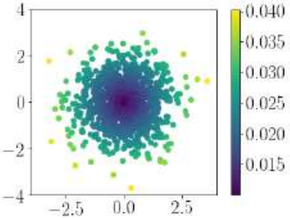
- The **sensitivity** of a datapoint $\boldsymbol{x}_i \in \mathcal{X}$ with respect to $f : \mathcal{X}, \Theta \to \mathbb{R}^+$ is:

$$\sigma_i = \max_{\theta \in \Theta} \frac{f(\boldsymbol{x}_i, \theta)}{L(\mathcal{X}, \theta)} \qquad \in [0, 1].$$

  Also, the total sensitivity is defined as $\mathfrak{S} = \sum_{i=1}^{n} \sigma_i$.
- In general, the sensitivity is unknown analytically.
- 1-means is an exception. In this case, supposing wlog that the data is centered (*i.e.*: $\sum_j x_j = 0$), one shows:

$$\sigma_i = \frac{1}{n}\left(1 + \frac{\|x_i\|^2}{v}\right),$$

  where $v = \frac{1}{n}\sum_{x \in \mathcal{X}} \|x\|^2$. Thus, $\mathfrak{S} = 2$.



A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.
- Draw $\mathcal{S}$: m iid samples with replacement according to $\boldsymbol{p}$.

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.
- Draw $\mathcal{S}$: m iid samples with replacement according to $\boldsymbol{p}$.
- Associate importance sampling weights to each sample of $\mathcal{S}$.

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.
- Draw $\mathcal{S}$: m iid samples with replacement according to $\boldsymbol{p}$.
- Associate importance sampling weights to each sample of $\mathcal{S}$.
- **Theorem**. The weighted subset $\mathcal{S}$ is a ϵ-coreset with high probability if:

$$m \geqslant \mathcal{O}\left(\frac{d'}{\epsilon^2}\left(\max_i \frac{\sigma_i}{p_i}\right)^2\right),$$

  where d' is the pseudo-dimension of Θ (a generalization of the Vapnik-Chervonenkis dimension).

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.
- Draw $\mathcal{S}$: m iid samples with replacement according to $\boldsymbol{p}$.
- Associate importance sampling weights to each sample of $\mathcal{S}$.
- **Theorem**. The weighted subset $\mathcal{S}$ is a ϵ-coreset with high probability if:

$$m \geqslant \mathcal{O}\left(\frac{d'}{\epsilon^2}\left(\max_i \frac{\sigma_i}{p_i}\right)^2\right),$$

  where d' is the pseudo-dimension of Θ (a generalization of the Vapnik-Chervonenkis dimension).
- The optimal probability distribution minimizing the rhs is $p_i = \sigma_i / \mathfrak{S}$.

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

A classical iid coreset theorem[1]

- Let $\boldsymbol{p} \in [0,1]^n$ be a probability distribution over all datapoints $\mathcal{X}$ with p_i the probability of sampling $\boldsymbol{x}_i$ and $\sum_i p_i = 1$.
- Draw $\mathcal{S}$: m iid samples with replacement according to $\boldsymbol{p}$.
- Associate importance sampling weights to each sample of $\mathcal{S}$.
- **Theorem**. The weighted subset $\mathcal{S}$ is a ϵ-coreset with high probability if:

$$m \geqslant \mathcal{O}\left(\frac{d'}{\epsilon^2}\left(\max_i \frac{\sigma_i}{p_i}\right)^2\right),$$

  where d' is the pseudo-dimension of Θ (a generalization of the Vapnik-Chervonenkis dimension).
- The optimal probability distribution minimizing the rhs is $p_i = \sigma_i / \mathfrak{S}$.
- In this case, $\mathcal{S}$ is a ϵ-coreset with high probability if:

$$m \geqslant \mathcal{O}\left(\frac{d'\mathfrak{S}^2}{\epsilon^2}\right).$$

[1] Langberg and Schulman, *Universal ϵ-approximators for integrals*, SIAM, 2010

DPPs for Coresets: a result[1]

- Consider *any* iid sampling scheme, defined by:
  - m the number of samples to draw
  - $\forall i, 0 \leqslant p_i \leqslant 1/m$ and $\sum_i p_i = 1$

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- Consider *any* iid sampling scheme, defined by:
  - m the number of samples to draw
  - $\forall i, 0 \leqslant p_i \leqslant 1/m$ and $\sum_i p_i = 1$
- Consider a marginal kernel K verifying:
  - K is projective of rank m: $\mathrm{K} = \mathrm{U}\mathrm{U}^t$ with $\mathrm{U} \in \mathbb{R}^{n \times m}$ and $\mathrm{U}^t\mathrm{U} = \mathrm{I}_m$.
  - $\forall i, \mathrm{K}_{ii} = mp_i$.

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- Consider *any* iid sampling scheme, defined by:
  - m the number of samples to draw
  - $\forall i, 0 \leqslant p_i \leqslant 1/m$ and $\sum_i p_i = 1$
- Consider a marginal kernel K verifying:
  - K is projective of rank m: $\mathrm{K} = \mathrm{U}\mathrm{U}^t$ with $\mathrm{U} \in \mathbb{R}^{n \times m}$ and $\mathrm{U}^t\mathrm{U} = \mathrm{I}_m$.
  - $\forall i, \mathrm{K}_{ii} = m p_i$.
- **Lemma.** Such a kernel necessarily exists. In general, there are many dof left to define U.

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- Consider *any* iid sampling scheme, defined by:
  - m the number of samples to draw
  - $\forall i, 0 \leqslant p_i \leqslant 1/m$ and $\sum_i p_i = 1$
- Consider a marginal kernel K verifying:
  - K is projective of rank m: $\mathrm{K} = \mathrm{U}\mathrm{U}^t$ with $\mathrm{U} \in \mathbb{R}^{n\times m}$ and $\mathrm{U}^t\mathrm{U} = \mathrm{I}_m$.
  - $\forall i, \mathrm{K}_{ii} = mp_i$.
- **Lemma.** Such a kernel necessarily exists. In general, there are many dof left to define U.
- Sample $\mathcal{S}_{iid}$ by drawing m samples iid from $\boldsymbol{p}$
- Sample $\mathcal{S}_{dpp}$ from the DPP of kernel K.
- Recall that $\mathcal{S}_{dpp}$ is necessarily of size m.

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- Consider *any* iid sampling scheme, defined by:
  - m the number of samples to draw
  - $\forall i, 0 \leqslant p_i \leqslant 1/m$ and $\sum_i p_i = 1$
- Consider a marginal kernel K verifying:
  - K is projective of rank m: $\mathrm{K} = \mathrm{U}\mathrm{U}^t$ with $\mathrm{U} \in \mathbb{R}^{n\times m}$ and $\mathrm{U}^t\mathrm{U} = \mathrm{I}_m$.
  - $\forall i, \mathrm{K}_{ii} = mp_i$.
- **Lemma.** Such a kernel necessarily exists. In general, there are many dof left to define U.
- Sample $\mathcal{S}_{iid}$ by drawing m samples iid from $\boldsymbol{p}$
- Sample $\mathcal{S}_{dpp}$ from the DPP of kernel K.
- Recall that $\mathcal{S}_{dpp}$ is necessarily of size m.
- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.
- This is in particular true for the ideal sensitivity-based iid sampling scheme.

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.
- This is in particular true for the ideal sensitivity-based iid sampling scheme.
- The *best* marginal kernel is for now out-of-reach: it poses deep questions rooted in frame theory.

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.
- This is in particular true for the ideal sensitivity-based iid sampling scheme.
- The *best* marginal kernel is for now out-of-reach: it poses deep questions rooted in frame theory.
- Even if we were able to find it, it would probably be untractable.

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.
- This is in particular true for the ideal sensitivity-based iid sampling scheme.
- The *best* marginal kernel is for now out-of-reach: it poses deep questions rooted in frame theory.
- Even if we were able to find it, it would probably be untractable.

$\rightarrow$ We propose a computationally efficient heuristic based on the Gaussian kernel:
- Compute r Random Fourier Features[2] ($r = \mathcal{O}(m)$) and obtain $\Psi \in \mathbb{R}^{n\times r}$ s.t. $\Psi\Psi^t \in \mathbb{R}^{n\times n}$ approximates the Gaussian kernel
- Sample an m-DPP from $L = \Psi\Psi^t$

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

DPPs for Coresets: a result[1]

- **Coreset variance reduction theorem**. One has:

$$\forall \theta \in \Theta \qquad Var\left[\hat{L}(\mathcal{S}_{dpp}, \theta)\right] \leqslant Var\left[\hat{L}(\mathcal{S}_{iid}, \theta)\right]$$

- For *any* iid sampling scheme, there exists (at least) a projective DPP sampling scheme outperforming it.
- This is in particular true for the ideal sensitivity-based iid sampling scheme.
- The *best* marginal kernel is for now out-of-reach: it poses deep questions rooted in frame theory.
- Even if we were able to find it, it would probably be untractable.

$\rightarrow$ We propose a computationally efficient heuristic based on the Gaussian kernel:

- Compute r Random Fourier Features[2] ($r = \mathcal{O}(m)$) and obtain $\Psi \in \mathbb{R}^{n \times r}$ s.t. $\Psi\Psi^t \in \mathbb{R}^{n \times n}$ approximates the Gaussian kernel
- Sample an m-DPP from $L = \Psi\Psi^t$

$\rightarrow$ This runs in $\mathcal{O}(nm^2 + nmd)$

[2] Rahimi et al., *Random features for large-scale kernel machines*, NIPS, 2008

[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

In practice: the 1-means controlled example[1]

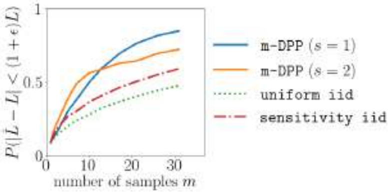
- Data $\mathcal{X}$, parameter θ
- Cost func.

$$L(\mathcal{X}, \theta) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \theta\|^2$$

Compare:

- `uniform iid` sampling
- `sensitivity iid`: ideal iid sampling based on exact sensitivities
- m-DPP (heuristic) based on RFFs of the Gaussian L-ensemble

$$\mathsf{L}_{ij} = \exp^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/s^2}$$



[1] Tremblay et al., *DPPs for Coresets*, Arxiv, 2018.

Conclusion: take home messages

- DPPs create random, diverse samples.
- They are tractable (inclusion probabilities at all orders are known), and good approximations are known for m-DPPs.
- This does not mean they are the best choice for all applications! They are *many* other (less tractable) repulsive processes out there.
- They are used in practice
- They are not expensive to sample in many applications (where low-rank approximations of the kernel can be computed efficiently): $\mathcal{O}(nm^2)$
- Toolboxes exist: DPPy[1] in Python, DPP.jl[2] in Julia

[1] github.com/guilgautier/DPPy

[2] gricad-gitlab.univ-grenoble-alpes.fr/barthesi/dpp.jl

Conclusion

DPPs have links with many other theories:

- graph theory
- Gaussian processes
- multivariate polynomials
- random matrices
- etc.

Conclusion: what next?

- accelerate sampling for large m
- DPPs for large dimensional data?
- parallel implementations