Determinantal Point Processes
and Randomized Numerical Linear Algebra

Michael W. Mahoney

ICSI and Department of Statistics, UC Berkeley

April 2020

(Joint work with Michał Dereziński.)
Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
Randomized Linear Algebra

**Given:** data matrix \( X \)

**Goal:** efficiently construct a small sketch \( \tilde{X} \)

**Rank-preserving sketch**

**Low-rank approximation**
Determinants

\[ \det(A) = \prod_{i} \lambda_i(A) \]

Some popular wisdom about determinants:

- Expensive to compute
- Numerically unstable
- Exponentially large... or exponentially small

---

Down With Determinants!

Sheldon Axler

1. INTRODUCTION. Ask anyone why a square matrix of complex numbers has an eigenvalue, and you’ll probably get the wrong answer, which goes something like this. There are no eigenvalues in the complex numbers, which is all fine, okay?
And yet... Determinantal Point Processes (DPPs)

A family of non-i.i.d. sampling distributions

1. Applications in Randomized Linear Algebra
   ▶ Least squares regression [DW17, DWH18]
   ▶ Low-rank approximation [DRVW06, GS12, DKM20]
   ▶ Randomized Newton’s method [DM19, MDK19]

2. Connections to i.i.d. sampling methods
   ▶ Row norm scores
   ▶ Leverage scores
   ▶ Ridge leverage scores

3. Fast DPP sampling algorithms
   ▶ Exact sampling via eigendecomposition [HKP⁺06, KT11]
   ▶ Intermediate sampling via leverage scores [Der19, DCV19]
   ▶ Markov chain Monte Carlo sampling [AGR16]
Outline

Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
L-ensemble DPPs and k-DPPs

Given a psd $n \times n$ matrix $L$, sample subset $S \subseteq \{1..n\}$:

(L-ensemble) $\text{DPP}(L) : \Pr(S) = \frac{\det(L_{S,S})}{\det(I + L)}$ over all subsets.

(k-DPP) $k$-DPP($L$) : $\text{DPP}(L)$ conditioned on $|S| = k$.

DPPs appear everywhere!

- Physics (fermions)
- Random matrix theory (eigenvalue distribution)
- Graph theory (random spanning trees)
- Optimization (variance reduction)
- Machine learning (diverse sets)
Volume (determinant) as a measure of diversity

Let \( L = [x_i^T x_j]_{ij} \) for \( x_1, \ldots, x_n \in \mathbb{R}^d \).

Then, \( \det(L_{S,S}) = \text{Vol}^2(\{x_i : i \in S\}) \)

(a)

Image from [KT12]
Example: DPP vs i.i.d.

Negative correlation: $\Pr(i \in S \mid j \in S) < \Pr(i \in S)$

i.i.d. (left) versus DPP (right)

Image from [KT12]
Projection DPPs

If \( L \) has rank \( d \), then \( S \sim d\text{-DPP}(L) \) is a Projection DPP

Let \( L = XX^\top \) for a full rank \( n \times d \) matrix \( X \)

\[
\text{if } S \sim d\text{-DPP}(L) \text{ then } \Pr(S) = \frac{\det(X_S)^2}{\det(X^\top X)}.
\]

Closed form normalization (Cauchy-Binet formula).

Remark. If \( k < \text{rank}(L) \) then \( k\text{-DPP}(L) \) is not a projection DPP. (and also does not have such a simple normalization constant)
Hierarchy of DPPs

Broader class of negatively-correlated point processes: Strongly Rayleigh (SR) measures
Let $d = \text{rank}(\mathbf{L})$, and $\lambda_1, ..., \lambda_d$ be the non-zero eigenvalues of $\mathbf{L}$.

If $S \sim \text{DPP}(\mathbf{L})$ then:

$$|S| \sim \text{Poisson-Binomial}(\frac{\lambda_1}{\lambda_1 + 1}, ..., \frac{\lambda_d}{\lambda_d + 1})$$

$$\mathbb{E}[|S|] = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i + 1} = \text{tr}(\mathbf{L}(\mathbf{L} + \mathbf{I})^{-1}) < d$$

**Rescaling trick:** Sample $S \sim \text{DPP}(\frac{1}{\lambda} \mathbf{L})$ to control $\mathbb{E}[|S|]$

$$\Pr(S) \propto \det(\frac{1}{\lambda} \mathbf{L}_S, S) = \lambda^{-|S|} \det(\mathbf{L}_S, S)$$

\[\begin{align*}
\text{DPP}(\frac{1}{\lambda} \mathbf{L}) & \xrightarrow{\lambda \to 0} d\text{-DPP}(\mathbf{L}) \\
\text{L-ensemble} & \quad \text{Projection DPP}
\end{align*}\]
Outline

Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
Given: data matrix $\mathbf{X}$

Goal (row sampling): construct $\tilde{\mathbf{X}}$ from few rows of $\mathbf{X}$

- Rank-preserving sketch
- Low-rank approximation

- i.i.d. sampling: Leverage scores
- DPP sampling: Projection DPPs

- Ridge leverage scores
- $L$-ensembles
Connections to i.i.d. sampling

Given: full rank $n \times d$ matrix $X$

Methods based on i.i.d. row sampling:

1. Row norm scores: $p_i = \frac{\|x_i\|^2}{\|X\|_F^2}$
   \[
   \frac{\|x_i\|^2}{\|X\|_F^2} = \Pr(i \in S) \quad \text{for} \quad S \sim 1\text{-DPP}(XX^\top)
   \]

2. Leverage scores: $p_i = \frac{1}{d} x_i^\top (X^\top X)^{-1} x_i$
   \[
   x_i^\top (X^\top X)^{-1} x_i = \Pr(i \in S) \quad \text{for} \quad S \sim d\text{-DPP}(XX^\top)
   \]

3. Ridge leverage scores: $p_i = \frac{1}{d\lambda} x_i^\top (X^\top X + \lambda I)^{-1} x_i$
   \[
   x_i^\top (X^\top X + \lambda I)^{-1} x_i = \Pr(i \in S) \quad \text{for} \quad S \sim \text{DPP}(\frac{1}{\lambda} XX^\top)
   \]
Subsampled least squares

**Given:** $n$ points $x_i \in \mathbb{R}^d$ with labels $y_i \in \mathbb{R}$

**Goal:** Minimize loss $L(w) = \sum_i (x_i^T w - y_i)^2$ over all $n$ points

$w^* = \arg\min_w L(w) = X^\dagger y$
**Subsampled least squares**

**Given:** $n$ points $x_i \in \mathbb{R}^d$ with labels $y_i \in \mathbb{R}$

**Goal:** Minimize loss $L(w) = \sum_i (x_i^\top w - y_i)^2$ over all $n$ points

$$w^* = \arg\min_w L(w) = X^\dagger y$$

Sample $S = \{4, 6, 9\}$

Solve subproblem $(X_S, y_S)$
Unbiased estimators

**Theorem (Rank-preserving sketch, [DW17])**

If \( S \sim d\text{-DPP}(XX^\top) \), then:

\[
\mathbb{E}[X_S^{-1}y_S] = \arg\min_w L(w) = w^*.
\]

**Theorem (Low-rank sketch, [DLM19])**

If \( S \sim \text{DPP}(\frac{1}{\lambda}XX^\top) \), then:

\[
\mathbb{E}[X_S^\dagger y_S] = \arg\min_w L(w) + \lambda\|w\|^2
\]

Not achievable with any i.i.d. row sampling!
Merits of unbiased estimators

Simple Strategy:
1. Compute independent estimators \( w(S_j) \) for \( j = 1, \ldots, k \),
2. Predict with the average estimator \( \frac{1}{k} \sum_{j=1}^{k} w(S_j) \)

If we have

\[
\mathbb{E}[L(w(S))] \leq (1 + c)L(w^*) \quad \text{and} \quad \mathbb{E}[w(S)] = w^*,
\]

then for \( k \) independent samples \( S_1, \ldots, S_k \),

\[
\mathbb{E}\left[L\left(\frac{1}{k} \sum_{j=1}^{k} w(S_j)\right)\right] \leq \left(1 + \frac{c}{k}\right) L(w^*)
\]

Motivation:
- Ensemble methods
- Distributed optimization
- Privacy
Connections to Gaussian sketches

Gaussian sketch
(Also gives unbiased estimators for least squares)

Let \( S \) be \( \frac{1}{k} \times \text{i.i.d.} \) Gaussian with \( \mathbb{E}[S^\top S] = I \). For \( k > d + 1 \):

\[
\mathbb{E}\left[(X^\top S^\top S X)^{-1}\right] = \left(X^\top X\right)^{-1} \frac{k}{k - d - 1}
\]

DPP plus uniform

Let \( S \sim d\text{-DPP}(XX^\top) \), \( T \sim \text{Bin}(n, \frac{k-d}{n-d}) \) and \( \tilde{S} = \left[\sqrt{n \ k \ e_i}\right]^\top_{i \in S \cup T} \).

Note: \( \mathbb{E}[|S|] = k \). For \( k \geq d \), we have:

\[
\mathbb{E}\left[(X^\top \tilde{S}^\top \tilde{S} X)^{-1}\right] = \left(X^\top X\right)^{-1} \frac{k}{k - d} \cdot (1 - o_n(1))
\]

DPPs have a “Gaussianizing” effect on row sampling.
Outline

Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
Determinant preserving random matrices

**Definition ([DLM19])**

A random $d \times d$ matrix $A$ is determinant preserving (d.p.) if

$$
\mathbb{E} [\det(A_{\mathcal{I},\mathcal{J}})] = \det(\mathbb{E}[A_{\mathcal{I},\mathcal{J}}]) \quad \text{for all } \mathcal{I}, \mathcal{J} \subseteq [d] \text{ s.t. } |\mathcal{I}| = |\mathcal{J}|.
$$

Basic examples:

- Every *deterministic* matrix
- Every *scalar* random variable
- Random matrix with i.i.d. Gaussian entries
More examples

Let $A = sZ$, where:

- $Z$ is deterministic with $\text{rank}(Z) = r$,
- $s$ is a scalar random variable with positive variance.

$$
\mathbb{E}\left[ \det(sZ_{\mathcal{I},\mathcal{J}}) \right] = \mathbb{E}[s^r] \det(Z_{\mathcal{I},\mathcal{J}}) = \det\left( (\mathbb{E}[s^r])^{\frac{1}{r}} Z_{\mathcal{I},\mathcal{J}} \right),
$$

Two cases:

1. If $r = 1$ then $A$ is determinant preserving,
2. If $r > 1$ then $A$ is not determinant preserving.
Basic properties

Lemma (Closure)

If $A$ and $B$ are independent and determinant preserving, then:

- $A + B$ is determinant preserving,
- $AB$ is determinant preserving.

Lemma (Adjugate)

If $A$ is determinant preserving, then $\mathbb{E}[\text{adj}(A)] = \text{adj}(\mathbb{E}[A])$.

When $A$ is invertible then $\text{adj}(A) = \det(A)A^{-1}$

Note: The $(i,j)$th entry of $\text{adj}(A)$ is $(-1)^{i+j} \det(A_{[n]\{j\},[n]\{i\}})$. 
Proof of closure under addition

First show that $A + uv^\top$ is d.p. for fixed $u, v \in \mathbb{R}^d$:

$$
\mathbb{E} \left[ \det (A_{I,J} + u_{I}v_{J}^\top) \right] = \mathbb{E} \left[ \det (A_{I,J}) + v_{J}^\top \text{adj}(A_{I,J}) u_{I} \right] \\
= \det \left( \mathbb{E} [A_{I,J}] + v_{J}^\top \text{adj}(\mathbb{E} [A_{I,J}]) u_{I} \right) \\
= \det \left( \mathbb{E} [A_{I,J} + u_{I}v_{J}^\top] \right).
$$

Iterating this, we get $A + Z$ is d.p. for any fixed $Z$

$$
\mathbb{E} \left[ \det (A_{I,J} + B_{I,J}) \right] = \mathbb{E} \left[ \mathbb{E} \left[ \det (A_{I,J} + B_{I,J}) \mid B \right] \right] \\
= \mathbb{E} \left[ \det \left( \mathbb{E} [A_{I,J}] + B_{I,J} \right) \right] \\
= \det \left( \mathbb{E} [A_{I,J} + B_{I,J}] \right)
$$
Application: Expected inverse

**Theorem**

Let \( \Pr(S) \propto \det(X_S^T X_S) p^{|S|} (1 - p)^{n-|S|} \) over all \( S \subseteq [n] \). Then:

\[
\mathbb{E}[(X_S^T X_S)^{-1}] \preceq \frac{1}{p} (X^T X)^{-1}.
\]

**Proof** Let \( b_1, \ldots, b_n \sim \text{Bernoulli}(p) \), and define \( \bar{S} = \{i : b_i = 1\} \). For each \( i \in [n] \), matrix \( b_i x_i x_i^T \) is determinant preserving. Therefore, \( X_{\bar{S}}^T X_{\bar{S}} = \sum_{i=1}^n b_i x_i x_i^T \) is determinant preserving.

\[
\mathbb{E}[(X_S^T X_S)^{-1}] = \frac{\mathbb{E}[\det(X_{\bar{S}}^T X_{\bar{S}})(X_{\bar{S}}^T X_{\bar{S}})^{\dagger}]}{\mathbb{E}[\det(X_{\bar{S}}^T X_{\bar{S}})]} \leq \frac{\mathbb{E}[\text{adj}(X_{\bar{S}}^T X_{\bar{S}})]}{\mathbb{E}[\det(X_{\bar{S}}^T X_{\bar{S}})]} = \frac{\text{adj}(\mathbb{E}[X_{\bar{S}}^T X_{\bar{S}}])}{\det(\mathbb{E}[X_{\bar{S}}^T X_{\bar{S}}])} = (\mathbb{E}[X_{\bar{S}}^T X_{\bar{S}}])^{-1} = (pX^T X)^{-1}
\]
Outline

Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
Algorithmic challenges with sampling from DPPs

Task:

(variant 1) Given $L$, sample $S \sim DPP(L)$

(variant 2) Given $L$ and $k$, sample $S \sim k$-DPP($L$)

(Task B: we are given $n \times d$ matrix $X \in \mathbb{R}^d$ instead of $L = XX^\top$)

Challenges:

1. Expensive preprocessing
typically involves eigendecomposition of $L$ in $O(n^3)$ time

2. Sampling time scales with $n$ rather than with $|S| \ll n$
undesirable when we need many samples $S_1, S_2, \cdots \sim DPP(L)$

3. Trade-offs between accuracy and runtime
   - exact algorithms - often too expensive
   - approximate algorithms - difficult to evaluate accuracy
Exact DPP sampling

**Key result:** any DPP is a mixture of Projection DPPs [HKP+06]

- Eigendecomposition $O(n^3)$ needed only once for a given kernel
- Reduction to a projection DPP $O(n |S|^2)$ needed for every sample

Cost of first sample $S_1 \sim \text{DPP}(L)$: $O(n^3)$

Cost of next sample $S_2 \sim \text{DPP}(L)$: $O(nk^2)$ ($k = \mathbb{E}[|S|]$)

Extends to a k-DPP sampler [KT11]
Approximate k-DPP sampler using MCMC

1. Start from some state $S \subseteq [n]$ of size $k$
2. Uniformly sample $i \in S$ and $j \notin S$
3. Move to state $S - i + j$ with probability $\frac{1}{2} \min \left\{ 1, \frac{\det(L_{S - i + j})}{\det(L_S)} \right\}$
4. ...

Converges in $O(nk \log \frac{1}{\epsilon})$ steps to within $\epsilon$ total variation [AGR16]

Cost of first sample $S_1 \sim k$-DPP($L$): $O(n \cdot \text{poly}(k))$
Cost of next sample $S_2 \sim k$-DPP($L$): $O(n \cdot \text{poly}(k))$

Extends to an $O(n^2 \cdot \text{poly}(k))$ sampler for DPP($L$) [LJS16]
Distortion-free intermediate sampling

1. Draw an intermediate sample:
\[ \sigma = (\sigma_1, \ldots, \sigma_t) \]

2. Downsample:
\[ S \subseteq [t] \]

3. Return:
\[ \{\sigma_i : i \in S\} \]

What is the right intermediate sampling distribution for \( \sigma \)?

- **Leverage scores**, when \( S \) is a Projection DPP
- **Ridge leverage scores**, when \( S \) is an L-ensemble
Distortion-free intermediate sampling for L-ensembles

**Theorem ([DCV19])**

*There is an algorithm which, given access to $L$, returns*

1. *first sample $S_1 \sim \text{DPP}(L)$ in: $n \cdot \text{poly}(k) \log(n)$ time,*
2. *next sample $S_2 \sim \text{DPP}(L)$ in: $\text{poly}(k)$ time.*

- Exact sampling
- Cost of first sample is sublinear in the size of $L$
- Cost of next sample is independent of the size of $L$
Outline

Introduction

Determinantal point processes

DPPs in Randomized Linear Algebra

Key technique: Determinant preserving random matrices

Sampling algorithms

Conclusions
Conclusions

1. New fundamental connections between:
   1.1 Determinantal Point Processes
   1.2 Randomized Linear Algebra

2. New unbiased estimators and expectation formulas

3. Efficient sampling algorithms

4. Determinant preserving random matrices

DPP-related topics we did not cover:
- Column Subset Selection Problem
- Nyström method
- Monte Carlo integration
- Distributed/Stochastic optimization
- ...
Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei.
Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes.

Michał Dereziński, Daniele Calandriello, and Michal Valko.
Exact sampling of determinantal point processes with sublinear time preprocessing.

Michał Dereziński.
Fast determinantal point processes via distortion-free intermediate sampling.

Michał Dereziński, Rajiv Khanna, and Michael W Mahoney.
Improved guarantees and a multiple-descent curve for the column subset selection problem and the nyström method.
Michał Dereźniński, Feynman Liang, and Michael W. Mahoney.
Exact expressions for double descent and implicit regularization via surrogate random design.

Michał Dereźniński and Michael W Mahoney.
Distributed estimation of the inverse hessian by determinantal averaging.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang.
Matrix approximation and projective clustering via volume sampling.

Michał Dereźniński and Manfred K. Warmuth.
Unbiased estimates for linear regression via volume sampling.

Michał Dereźniński, Manfred K. Warmuth, and Daniel Hsu.
Leveraged volume sampling for linear regression.
Optimal column-based low-rank matrix reconstruction.

J. Ben Hough, Manjunath Krishnapur, Yuval Peres, Bálint Virág, et al.
Determinantal processes and independence.

Alex Kulesza and Ben Taskar.
k-DPPs: Fixed-Size Determinantal Point Processes.

Alex Kulesza and Ben Taskar.
Determinantal Point Processes for Machine Learning.

Chengtao Li, Stefanie Jegelka, and Suvrit Sra.
Fast mixing markov chains for strongly Rayleigh measures, DPPs, and constrained sampling.

Mojmír Mutný,Michał Dereziński, and Andreas Krause.
Convergence analysis of the randomized newton method with determinantal sampling.
Thank you!