Scalable MCMC Sampling for Nonsymmetric Determinantal Point Processes

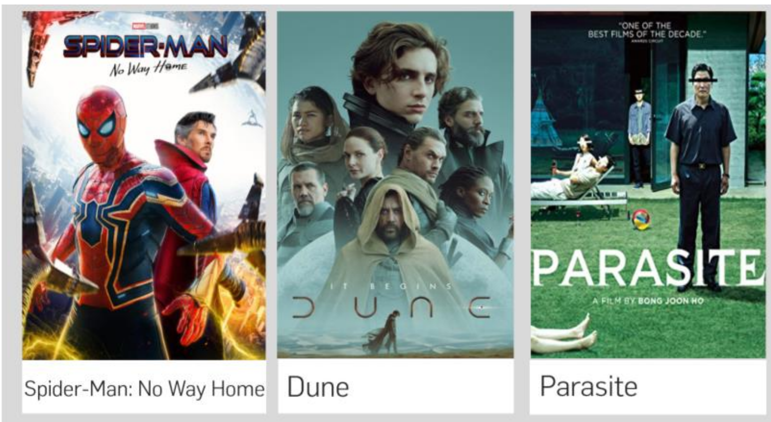
Insu Han◐, Mike Gartrell▣, Elvis Dohmatob◈, Amin Karbasi◐

◐Yale University
▣Criteo AI Lab
◈Facebook AI Lab

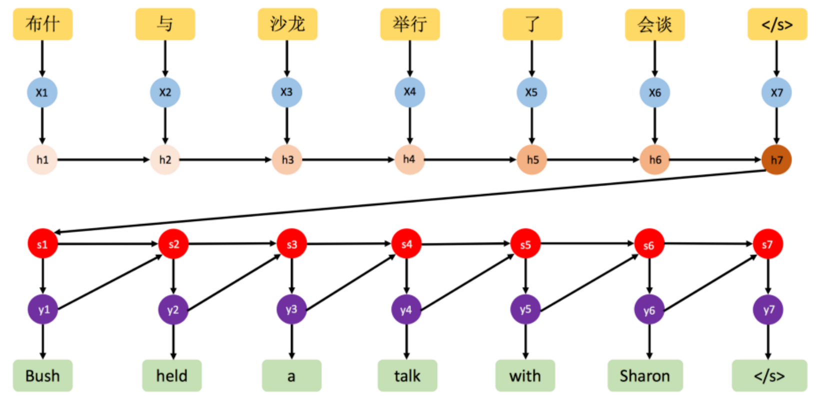
ICML 2022

Motivation

- Imposing diversity can be important in many real-world applications



Movie Recommendation

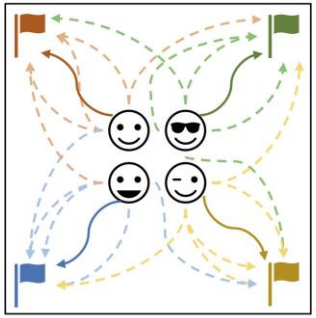


(Sutskever et al., 2014)

Machine Translation

(Rochan and Wang, 2019)

Video Summarization



Multi-agent Learning

Determinantal Point Processes (DPPs)

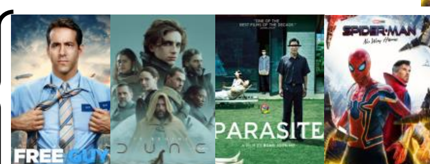
Definition. Given a ground set $\{1, \ldots, n\} := [n]$ and a positive semi-definite matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, a DPP models a distribution over subsets of $[n]$ such that

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

*$\boldsymbol{L}_S$ is a submatrix indexed by S

$\boldsymbol{L} =$

2.5	0.1	0.2	2.3
0.1	2.4	0.1	0.2
0.2	0.1	2.0	0.2
2.3	0.2	0.2	2.6

ground set $\{$  $\}$

Pr( , ) $\propto \det$

2.5	2.3
2.3	2.6

$$\det(\boldsymbol{L}_{\{1,4\}}) = \boldsymbol{L}_{11}\boldsymbol{L}_{44} - \boldsymbol{L}_{14}^2$$

quality similarity

Determinantal Point Processes (DPPs)

Definition. Given a ground set $\{1, \ldots, n\} := [n]$ and a positive semi-definite matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, a DPP models a distribution over subsets of $[n]$ such that

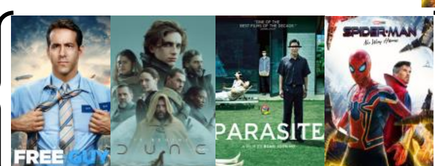
$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

*$\boldsymbol{L}_S$ is a submatrix indexed by S

$\boldsymbol{L} =$

2.5	0.1	0.2	2.3
0.1	2.4	0.1	0.2
0.2	0.1	2.0	0.2
2.3	0.2	0.2	2.6

ground set



Pr

$\propto$ det

2.5	2.3
2.3	2.6

$$\det(\boldsymbol{L}_{\{1,4\}}) = \boldsymbol{L}_{11}\boldsymbol{L}_{44} - \boldsymbol{L}_{14}^2$$

quality similarity

- k-DPP: the support is the collection of size-k subsets, i.e., $|S| = k$

Nonsymmetric DPPs (NDPPs)

- **Nonsymmetric DPPs** [GBDK19]**.** The kernel matrix can be nonsymmetric, i.e., any P_0-matrix can define a DPP. For example,

$$\boldsymbol{L} = \begin{pmatrix} 1 & 5/3 \\ 1/2 & 1 \end{pmatrix} \qquad \det(\boldsymbol{L}_{\{1\}}), \det(\boldsymbol{L}_{\{2\}}), \det(\boldsymbol{L}_{\{1,2\}}) \geq 0$$

- Nonsymmetric kernel can induce both negative and positive interactions

$$\det(\boldsymbol{L}_{\{i,j\}}) = \underbrace{\boldsymbol{L}_{ii}\boldsymbol{L}_{jj}}_{\text{quality}} - \underbrace{\boldsymbol{L}_{ji}\boldsymbol{L}_{ij}}_{\text{similarity}}$$

  - If $\boldsymbol{L}$ is symmetric, $-\boldsymbol{L}_{ij}\boldsymbol{L}_{ji} \leq 0$, thus negative interaction
  - If $\boldsymbol{L}$ is nonsymmetric, then $\boldsymbol{L}_{ij}, \boldsymbol{L}_{ji}$ are different signs, $-\boldsymbol{L}_{ij}\boldsymbol{L}_{ji} \geq 0$ can lead to positive interaction

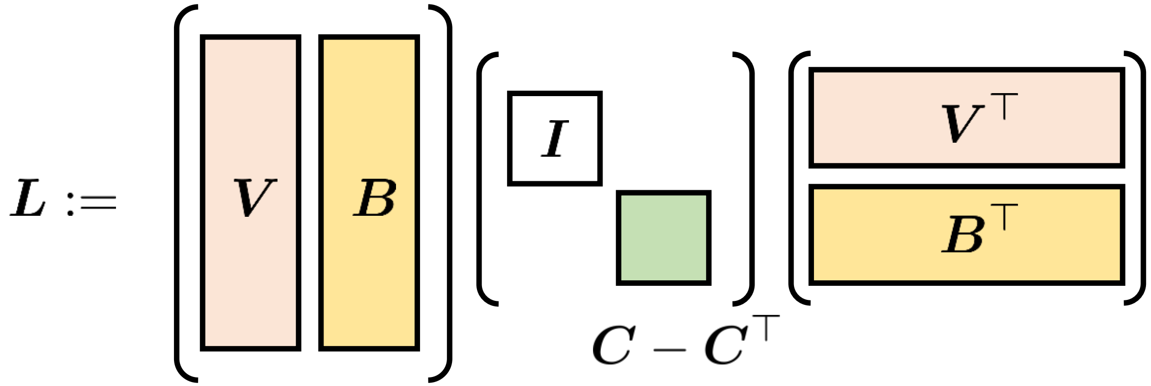
Nonsymmetric DPPs (NDPPs)

- **Low-rank kernel decomposition of NDPP** [GHD+21]**:**

$$\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}(\boldsymbol{C} - \boldsymbol{C}^\top)\boldsymbol{B}^\top$$

$$\boldsymbol{V}, \boldsymbol{B} \in \mathbb{R}^{n \times d/2}, \boldsymbol{C} \in \mathbb{R}^{d/2 \times d/2} \text{ s.t. } d \ll n.$$

  - $\det(\boldsymbol{L}_S) \geq 0, \text{ for } S \subseteq [n]$ ⇒ $\mathcal{P}_{\boldsymbol{L}}(S) \geq 0$ ⇒ valid DPP



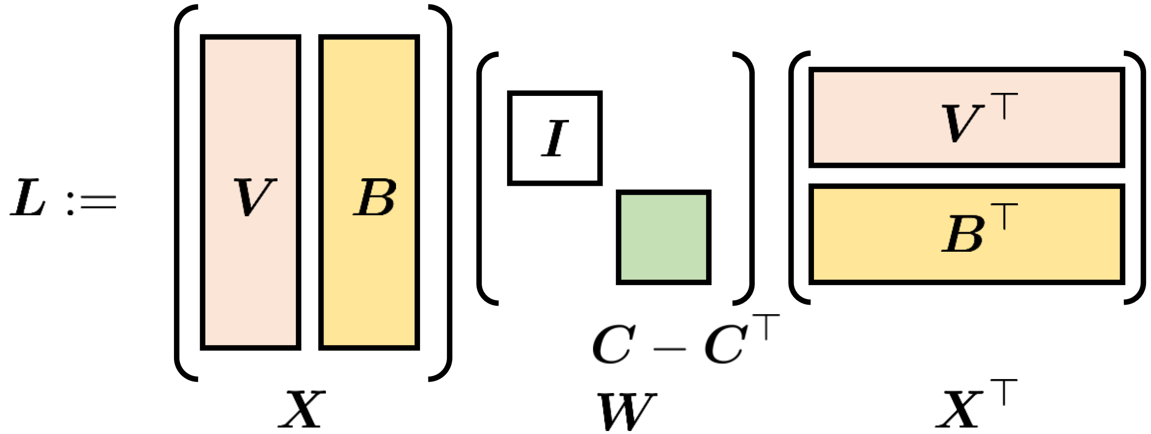
Nonsymmetric DPPs (NDPPs)

- **Low-rank kernel decomposition of NDPP** [GHD+21]**:**

$$\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}(\boldsymbol{C} - \boldsymbol{C}^\top)\boldsymbol{B}^\top$$

$$\boldsymbol{V}, \boldsymbol{B} \in \mathbb{R}^{n \times d/2}, \boldsymbol{C} \in \mathbb{R}^{d/2 \times d/2} \text{ s.t. } d \ll n.$$

  - $\det(\boldsymbol{L}_S) \geq 0, \text{ for } S \subseteq [n]$ ⇒ $\mathcal{P}_{\boldsymbol{L}}(S) \geq 0$ ⇒ valid DPP
  - Simpler form: $[\boldsymbol{V} \ \ \boldsymbol{B}] \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C} - \boldsymbol{C}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^\top \\ \boldsymbol{B}^\top \end{bmatrix} := \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top$



Contribution 1

- **Goal:** fast sampling algorithm for size-constrained NDPP (k-NDPP)

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S) \qquad |S| = k$$

$$\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top,\ \boldsymbol{X} \in \mathbb{R}^{n\times d}, \boldsymbol{W} \in \mathbb{R}^{d\times d}, k \le d \ll n$$

- **Contribution:** approximate MCMC sampling algorithm

Algorithm	Preprocessing Time	Sampling Time
Our work	$\mathcal{O}(nd^2)$	$\widetilde{\mathcal{O}}(k^2(1+\kappa)^2(d^2 \log n + d^3))$ (*)
[AASV21]	–	$\widetilde{\mathcal{O}}(n^2 k^5)$

  - (*) $\kappa > 0$: data-dependent constant, not dependent on d, k and n

$\widetilde{\mathcal{O}}(\cdot)$ hides dependency on an accuracy parameter

Contribution 2

- **Goal:** fast sampling algorithm for size-unconstrained NDPP

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

$$\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top,\ \boldsymbol{X} \in \mathbb{R}^{n\times d}, \boldsymbol{W} \in \mathbb{R}^{d\times d}, d \ll n$$

- **Contribution:** approximate MCMC sampling algorithm

Algorithm	Preprocessing Time	Sampling Time
Our work	$\mathcal{O}(nd^2)$	$\widetilde{\mathcal{O}}(\lvert S\rvert^2(1+\kappa)^2(d^2 \log n + d^3))^{(*)}$
[Pou20] **(Exact)**	–	$\mathcal{O}(nd^2)$
[HGG+22] **(Exact)**	$\mathcal{O}(nd^2)$	$\mathcal{O}((1+\alpha)^d(\lvert S\rvert^3 \log n + \lvert S\rvert^4 + d))^{(*)}$

  - (*) $\kappa > 0$: data-dependent constant, independent of d, k and n
  - (*) $\alpha \in (0,1]$: data-dependent constant

MCMC Sampling for k-NDPP

- [AASV21] proposed MCMC sampling algorithm for k-NDPP :

```
S_0 ← Select a size-k subset of [n] uniformly at random
for t = 1, 2, ..., t_iter do
    A ← Select a size-(k − 2) subset of S_{t−1} uniformly at random
    {a, b} ← Select a size-2 subset with probability ∝ det(L_{A∪{a,b}})
    S_t ← A ∪ {a, b}
```

MCMC Sampling for k-NDPP

- [AASV21] proposed MCMC sampling algorithm for k-NDPP :

```
S_0 ← Select a size-k subset of [n] uniformly at random
for t = 1, 2, ..., t_iter do
    A ← Select a size-(k − 2) subset of S_{t−1} uniformly at random
    {a, b} ← Select a size-2 subset with probability ∝ det(L_{A∪{a,b}})
    S_t ← A ∪ {a, b}
```

- They proved that the mixing time is

$$t_{\text{iter}} = \mathcal{O}\left(k^2 \log\left(\frac{1}{\varepsilon \Pr(S_0)}\right)\right)$$

MCMC Sampling for k-NDPP

- [AASV21] proposed MCMC sampling algorithm for k-NDPP :

```
1: S_0 ← Select a size-k subset of [n] uniformly at random
2: for t = 1, 2, ..., t_iter do
3:     A ← Select a size-(k − 2) subset of S_{t−1} uniformly at random
4:     {a, b} ← Select a size-2 subset with probability ∝ det(L_{A∪{a,b}})
5:     S_t ← A ∪ {a, b}
```

- They proved that the mixing time is

$$t_{\text{iter}} = \mathcal{O}\left(k^2 \log\left(\frac{1}{\varepsilon \Pr(S_0)}\right)\right)$$

- Runtime of Line 4 (Computational Bottleneck):
  - For $\binom{n-k}{2}$ candidates of S_{t-1}, each one requires computing $k \times k$ matrix determinant $\Rightarrow \mathcal{O}(k^3)$
  - Overall runtime is $\mathcal{O}(n^2 k^3)$

MCMC Sampling for k-NDPP

- [AASV21] proposed MCMC sampling algorithm for k-NDPP :

```
1: S_0 ← Select a size-k subset of [n] uniformly at random
2: for t = 1, 2, ..., t_iter do
3:     A ← Select a size-(k − 2) subset of S_{t−1} uniformly at random
4:     {a, b} ← Select a size-2 subset with probability ∝ det(L_{A∪{a,b}})
5:     S_t ← A ∪ {a, b}
```

- They proved that the mixing time is

$$t_{\mathrm{iter}} = \mathcal{O}\left(k^2 \log\left(\frac{1}{\varepsilon \Pr(S_0)}\right)\right)$$

- Runtime of Line 4 (Computational Bottleneck):
  - We improve by combining (1) *rejection sampling for 2-NDPP* and (2) *tree-based sublinear time DPP sampling* [GKMV19]
  - The runtime is reduced to $\mathcal{O}((1+\kappa)^2(d^2 \log n + d^3))$

MCMC Sampling for Low-rank k-NDPP

- Computation bottleneck of MCMC sampling algorithm for k-NDPP :

  4: $\{a, b\} \leftarrow$ Select a size-2 subset with probability $\propto \det(\boldsymbol{L}_{A \cup \{a,b\}})$

- **Key Observation:**
  - Equivalent to sampling $\{a, b\}$ from DPP conditioned on A (2-NDPP)

MCMC Sampling for Low-rank k-NDPP

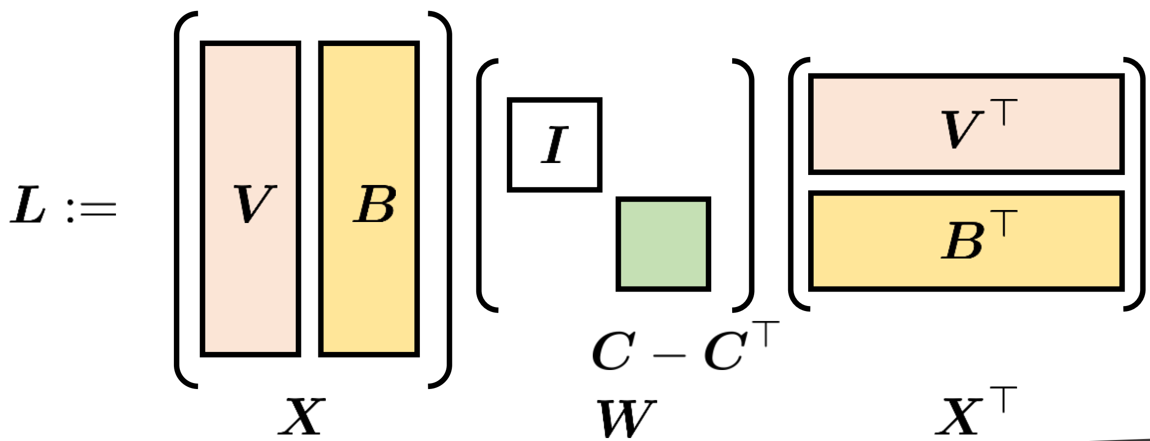
- Computation bottleneck of MCMC sampling algorithm for k-NDPP :

  4: $\{a, b\} \leftarrow$ Select a size-2 subset with probability $\propto \det(\boldsymbol{L}_{A\cup\{a,b\}})$

- **Key Observation:**
  - Equivalent to sampling $\{a, b\}$ from DPP conditioned on A (2-NDPP)
  - Given a low-rank NDPP as $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top$,
    DPP conditioned on $A \Leftrightarrow$ DPP with $\boldsymbol{L}^A = \boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$ where

$$\boldsymbol{W}^A := \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$$

can be computed in $\mathcal{O}(d^2k)$ time



MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, \ldots, t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
  $S_t \leftarrow A \cup \{a, b\}$

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
  $S_t \leftarrow A \cup \{a, b\}$

- **Rejection Sampling for 2-NDPP:**

while(true)
  Sample $\{a, b\} \sim$ 2-DPP$(\boldsymbol{L}')$
  if $\mathcal{U}([0,1]) \le \frac{\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}})}{\det(\boldsymbol{L}'_{\{a,b\}})}$
    return $\{a, b\}$

$\boldsymbol{L}'$: DPP kernel for proposal distribution

- Sampling can be *easy* and *fast*
- Should satisfy

$$\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_S) \le \det(\boldsymbol{L}'_S), \; S \subseteq [n]$$

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, \ldots, t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
  $S_t \leftarrow A \cup \{a, b\}$

- **Rejection Sampling for 2-NDPP:**

while(true)
  Sample $\{a, b\} \sim$ 2-DPP$(\boldsymbol{L}')$
  if $\mathcal{U}([0,1]) \le \frac{\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}})}{\det(\boldsymbol{L}'_{\{a,b\}})}$
    return $\{a, b\}$

$\boldsymbol{L}'$: DPP kernel for proposal distribution

- Sampling can be *easy* and *fast*
- Should satisfy

$$\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_S) \le \det(\boldsymbol{L}'_S), \ S \subseteq [n]$$

- We find a *symmetric* $\widehat{\boldsymbol{W}}^A \in \mathbb{R}^{d \times d}$ in time $\mathcal{O}(d^3)$ such that

$$\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_S) \le \det([\boldsymbol{X}\widehat{\boldsymbol{W}}^A\boldsymbol{X}^\top]_S), \quad S \subseteq [n]$$

→ proposal distribution

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
    $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
    $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
    $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
    $S_t \leftarrow A \cup \{a, b\}$

- **Rejection Sampling for 2-NDPP (Proposal Construction):**
  - Consider the spectral decomposition

$$\frac{\boldsymbol{W}^A - \boldsymbol{W}^{A\top}}{2} = \boldsymbol{P}\,\mathrm{Diag}\left(\begin{bmatrix} 0 & \sigma_1 \\ -\sigma_1 & 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 & \sigma_{d/2}, \\ -\sigma_{d/2} & 0 \end{bmatrix}\right)\boldsymbol{P}^\top$$

$$\widehat{\boldsymbol{W}}^A := \frac{\boldsymbol{W}^A + \boldsymbol{W}^{A\top}}{2} + \boldsymbol{P}\,\mathrm{Diag}\left(\sigma_1, \sigma_1, \ldots, \sigma_{d/2}, \sigma_{d/2}\right)\boldsymbol{P}^\top$$

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
  $S_t \leftarrow A \cup \{a, b\}$

- **Rejection Sampling for 2-NDPP (Proposal Construction):**
  - Consider the spectral decomposition

$$\frac{\boldsymbol{W}^A - \boldsymbol{W}^{A\top}}{2} = \boldsymbol{P}\ \mathrm{Diag}\left(\begin{bmatrix} 0 & \sigma_1 \\ -\sigma_1 & 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 & \sigma_{d/2}, \\ -\sigma_{d/2} & 0 \end{bmatrix}\right)\boldsymbol{P}^\top$$

$$\widehat{\boldsymbol{W}}^A := \frac{\boldsymbol{W}^A + \boldsymbol{W}^{A\top}}{2} + \boldsymbol{P}\ \mathrm{Diag}\left(\sigma_1, \sigma_1, \ldots, \sigma_{d/2}, \sigma_{d/2}\right)\boldsymbol{P}^\top$$

  - **Theorem.** $\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_S) \le \det([\boldsymbol{X}\widehat{\boldsymbol{W}}^A\boldsymbol{X}^\top]_S), \quad S \subseteq [n]$

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\{a, b\} \leftarrow$ Sample a subset from 2-NDPP with $\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top$
  $S_t \leftarrow A \cup \{a, b\}$

- **Rejection Sampling for 2-NDPP (Proposal Construction):**
  - Consider the spectral decomposition

$$\frac{\boldsymbol{W}^A - \boldsymbol{W}^{A\top}}{2} = \boldsymbol{P}\ \mathrm{Diag}\left(\left[\begin{smallmatrix} 0 & \sigma_1 \\ -\sigma_1 & 0 \end{smallmatrix}\right], \cdots, \left[\begin{smallmatrix} 0 & \sigma_{d/2}, \\ -\sigma_{d/2} & 0 \end{smallmatrix}\right]\right)\boldsymbol{P}^\top$$

$$\widehat{\boldsymbol{W}}^A := \frac{\boldsymbol{W}^A + \boldsymbol{W}^{A\top}}{2} + \boldsymbol{P}\ \mathrm{Diag}\left(\sigma_1, \sigma_1, \ldots, \sigma_{d/2}, \sigma_{d/2}\right)\boldsymbol{P}^\top$$

  - **Theorem.** $\det([\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_S) \le \det([\boldsymbol{X}\widehat{\boldsymbol{W}}^A\boldsymbol{X}^\top]_S), \quad S \subseteq [n]$
  - The decomposition takes time $\mathcal{O}(d^3)$
  - Similar construction was used in [HGG+22], but with $\mathcal{O}(nd^2)$ runtime

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, \ldots, t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\widehat{\boldsymbol{W}} \leftarrow \textsc{SpectralSymmetrization}(\boldsymbol{W}^A)$ $\Rightarrow \mathcal{O}(d^3)$
  While true **do**
    $\{a, b\} \leftarrow$ Sample a size-2 set from 2-DPP$(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top)$ → symmetric DPP
    if $\mathcal{U}([0,1]) \leq \frac{\det[\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}}}{\det[\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top]_{\{a,b\}}}$;**break**
  $S_t \leftarrow A \cup \{a, b\}$

MCMC Sampling for Low-rank k-NDPP

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\widehat{\boldsymbol{W}} \leftarrow \textsc{SpectralSymmetrization}(\boldsymbol{W}^A)$ $\Rightarrow \mathcal{O}(d^3)$
  While true **do**
    $\{a, b\} \leftarrow$ Sample a size-2 set from 2-DPP$(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top)$
    if $\mathcal{U}([0,1]) \le \frac{\det[\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}}}{\det[\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top]_{\{a,b\}}}$;**break**
  $S_t \leftarrow A \cup \{a, b\}$

symmetric DPP

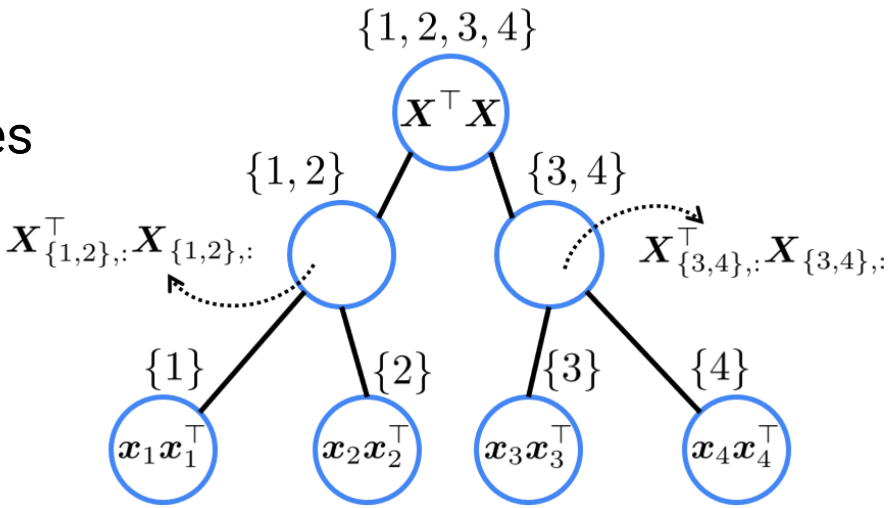
- **Rejection Sampling for 2-NDPP (Sampling Proposal DPP):**
  - Sampling $\{a, b\}$ using the *tree-based DPP sampling* [GKMV19] is done in time $\mathcal{O}(d^2 \log n + d^3)$

MCMC Sampling for Low-rank k-NDPP

7: $\{a, b\} \leftarrow$ Sample a size-2 set from 2-DPP($\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top$)

symmetric DPP

- **Tree-based Symmetric DPP Sampling** [GKMV19]:
  - Pre-processing:
    - Build a binary tree with $\boldsymbol{X}$
    - Each nodes stores a set of indices and a $d \times d$ matrix
    - Runtime: $\mathcal{O}(nd^2)$



  - Sampling:
    - Equivalent to 2 tree traversals $\Longrightarrow \mathcal{O}(d^2 \log n)$
    - Computation of a query matrix (only depends on $\boldsymbol{W}^A$) $\Longrightarrow \mathcal{O}(d^3)$
    - Runtime: $\mathcal{O}(d^2 \log n + d^3)$

MCMC Sampling for Low-rank k-NDPP

Preprocessing: $\Rightarrow \mathcal{O}(nd^2)$

Build a binary tree where each node stores indices $A \subseteq [n]$ and $\boldsymbol{X}_{A,:}^\top \boldsymbol{X}_{A,:}$

Sampling:

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
for $t = 1, 2, ..., t_{\text{iter}}$ **do**
  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top (\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
  $\widehat{\boldsymbol{W}} \leftarrow \textsc{SpectralSymmetrization}(\boldsymbol{W}^A)$ $\Rightarrow \mathcal{O}(d^3)$
  While true **do**
    $\{a, b\} \leftarrow$ Sample from a 2-DPP$(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top)$ by tree-based sampling $\Rightarrow \mathcal{O}(d^2 \log n + d^3)$
    if $\mathcal{U}([0,1]) \le \frac{\det[\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}}}{\det[\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top]_{\{a,b\}}}$;**break**
  $S_t \leftarrow A \cup \{a, b\}$

MCMC Sampling for Low-rank k-NDPP

Preprocessing: $\Rightarrow \mathcal{O}(nd^2)$

1: Build a binary tree where each node stores indices $A \subseteq [n]$ and $\boldsymbol{X}_{A,:}^\top \boldsymbol{X}_{A,:}$

Sampling:

1: $S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random
2: **for** $t = 1, 2, ..., t_{\text{iter}}$ **do**
3: $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random
4: $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top (\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$
5: $\widehat{\boldsymbol{W}} \leftarrow \textsc{SpectralSymmetrization}(\boldsymbol{W}^A)$ $\Rightarrow \mathcal{O}(d^3)$
6: **While** true **do**
7: $\{a, b\} \leftarrow$ Sample from a 2-DPP$(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top)$ by tree-based sampling
8: **if** $\mathcal{U}([0,1]) \le \frac{\det[\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}}}{\det[\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top]_{\{a,b\}}}$;**break** $\Rightarrow \mathcal{O}(d^2 \log n + d^3)$
9: $S_t \leftarrow A \cup \{a, b\}$

- **Question. What is the number of rejections?**

MCMC Sampling for Low-rank k-NDPP

- **Average Number of Rejections for 2-NDPP Rejection Sampling**

Theorem. For any $A \in \binom{[n]}{k-2}, k \geq 2$, define that

$$\kappa_A := \frac{\sigma_{\max}(\boldsymbol{W}^A - \boldsymbol{W}^{A^\top})}{\min_{Y \in \binom{[n] \setminus A}{2}} \sigma_{\min}([\boldsymbol{X}(\boldsymbol{W}^A + \boldsymbol{W}^{A\top})\boldsymbol{X}^\top]_Y)}$$

and $\kappa := \max_{A \subseteq [n], |A| \leq d-2} \kappa_A.$

Then the number of average rejections is no greater than

$$(1 + \sigma_{\max}(\boldsymbol{X})^2 \ \kappa)^2$$

  - κ_A is upper bounded by a ratio of largest and smallest eigenvalues among some 2-by-2 matrices ⇒ **not** dependent on either n or d

MCMC Sampling for Low-rank k-NDPP

- **Average Number of Rejections for 2-NDPP Rejection Sampling**

Theorem. For any $A \in \binom{[n]}{k-2}, k \geq 2$, define that

$$\kappa_A := \frac{\sigma_{\max}(\boldsymbol{W}^A - \boldsymbol{W}^{A^\top})}{\min_{Y \in \binom{[n] \setminus A}{2}} \sigma_{\min}([\boldsymbol{X}(\boldsymbol{W}^A + \boldsymbol{W}^{A\top})\boldsymbol{X}^\top]_Y)}$$

and $\kappa := \max_{A \subseteq [n], |A| \leq d-2} \kappa_A.$

Then the number of average rejections is no greater than

$$(1 + \sigma_{\max}(\boldsymbol{X})^2 \ \kappa)^2$$

- Practically, the number of rejections for real-world datasets is small

Dataset	UK Retail	Recipe	Instacart	Million Song
Average Number of Rejections ($k = 10$)	7.763	3.504	5.965	0.808

MCMC Sampling for Low-rank k-NDPP

Preprocessing: $\Rightarrow \mathcal{O}(nd^2)$

Build a binary tree where each node stores indices $A \subseteq [n]$ and $\boldsymbol{X}_{A,:}^\top \boldsymbol{X}_{A,:}$

Sampling:

$S_0 \leftarrow$ Select a size-k subset of $[n]$ uniformly at random

for $t = 1, 2, ..., t_{\text{iter}}$ **do**

  $A \leftarrow$ Select a size-$(k-2)$ subset of S_{t-1} uniformly at random

  $\boldsymbol{W}^A \leftarrow \boldsymbol{W} - \boldsymbol{W}\boldsymbol{X}_{A,:}^\top(\boldsymbol{X}_{A,:}\boldsymbol{W}\boldsymbol{X}_{A,:})^{-1}\boldsymbol{X}_{A,:}\boldsymbol{W}$ $\Rightarrow \mathcal{O}(d^3)$

  $\widehat{\boldsymbol{W}} \leftarrow \textsc{SpectralSymmetrization}(\boldsymbol{W}^A)$ $\Rightarrow \mathcal{O}(d^3)$

  While true **do** $\Rightarrow \mathcal{O}((1+\sigma_{\max}(\boldsymbol{X})^2\kappa)^2)$

    $\{a, b\} \leftarrow$ Sample from a 2-DPP$(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top)$ by tree-based sampling $\Rightarrow \mathcal{O}(d^2 \log n + d^3)$

    if $\mathcal{U}([0,1]) \leq \frac{\det[\boldsymbol{X}\boldsymbol{W}^A\boldsymbol{X}^\top]_{\{a,b\}}}{\det[\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^\top]_{\{a,b\}}}$;**break**

  $S_t \leftarrow A \cup \{a, b\}$

- **Total sampling time:** $\mathcal{O}\left(t_{\text{iter}} \cdot (1+\sigma_{\max}(\boldsymbol{X})^2\kappa)^2 \cdot \left(d^2 \log n + d^3\right)\right)$

MCMC Sampling for Size-unconstrained NDPP

- **Goal:** fast sampling algorithm for size-unconstrained NDPP

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

$$\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top,\ \boldsymbol{X} \in \mathbb{R}^{n\times d}, \boldsymbol{W} \in \mathbb{R}^{d\times d}, d \ll n$$

MCMC Sampling for Size-unconstrained NDPP

- **Goal:** fast sampling algorithm for size-unconstrained NDPP

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

$$\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top,\ \boldsymbol{X} \in \mathbb{R}^{n\times d}, \boldsymbol{W} \in \mathbb{R}^{d\times d}, d \ll n$$

- **Key Approach:**

  Step 1. Sample $k \in \{0, 1, 2, \dots, d\}$ with prob. $\propto \sum_{|S|=k} \det([\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top]_S)$

  ⇒ This can be done in time $\mathcal{O}(nd^2)$ as a preprocessing step

  ⇒ This does not impact the runtime of tree construction

MCMC Sampling for Size-unconstrained NDPP

- **Goal:** fast sampling algorithm for size-unconstrained NDPP

$$\mathcal{P}_{\boldsymbol{L}}(S) \propto \det(\boldsymbol{L}_S)$$

$$\boldsymbol{L} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top,\ \ \boldsymbol{X} \in \mathbb{R}^{n \times d}, \boldsymbol{W} \in \mathbb{R}^{d \times d}, d \ll n$$

- **Key Approach:**

Step 1. Sample $k \in \{0, 1, 2, \dots, d\}$ with prob. $\propto \sum_{|S|=k} \det([\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\top]_S)$

⇒ This can be done in time $\mathcal{O}(nd^2)$ as a preprocessing step

⇒ This does not impact the runtime of tree construction

Step 2. Run our k-NDPP sampling algorithm with the chosen k

⇒ The number of MCMC iterations (i.e. t_{iter}) only changes with k

Experiments

- Dataset: kernels obtained by the gradient-based MLE learning [GHD+21] on 5 recommendation datasets (rank $d = 100$)
- Wall-clock times of size-constrained NDPP sampling ($k = 50$)
- Competitor: exact rejection sampling [HGG+22] (no runtime guarantee)

Dataset	Algorithm	
	Rejection (Exact)	**MCMC (Ours)**
UK Retail (n=3,941)	(∗) 5.11×10^{12} sec	334 sec
Recipe (n=7,993)	(∗) 9.55×10^{5} sec	229 sec
Instacart (n=49,677)	(∗) 9.50×10^{5} sec	242 sec
Million Song (n=371,410)	(∗) 1.45×10^{12} sec	488 sec
Book (n=1,059,437)	(∗) 4.06×10^{6} sec	374 sec

(∗) : expected results, due to infeasible runtime

Experiments

- Dataset: kernels obtained by the gradient-based MLE learning [GHD+21] on 5 recommendation datasets (rank $d = 100$)
- Wall-clock times of size-unconstrained NDPP sampling
- Competitors: exact rejection sampling [HGG+22], Cholesky-based [Pou20]

Dataset	Algorithm		
	Rejection (Exact)	Cholesky (Exact)	**MCMC (Ours)**
UK Retail (n=3,941)	(*) 1.34×10^8 sec	5.6 sec	75.3
Recipe (n=7,993)	1.0 sec	11.5 sec	11.8 sec
Instacart (n=49,677)	1351.6 sec	71.1 sec	21 sec
Million Song (n=371,410)	(*) 1.89×10^{10} sec	537 sec	281 sec
Book (n=1,059,437)	1022 sec	1540 sec	80 sec

(*) : expected results, due to infeasible runtime

Conclusion

Summary:

- We accelerate MCMC sampling for size-constrained nonsymmetric DPPs (k-NDPPs) by leveraging a tree-based rejection sampling algorithm
- We extend this to size-unconstrained sampling while preserving the same efficient runtime
- We achieve runtime that is sublinear in n, and polynomial in d and k
- The fastest state-of-the-art "exact" sampling algorithms for NDPP has a runtime exponential in d
- We verify orders of magnitude speedups with real-world datasets