From Sampling to Optimization on Discrete Domains with Applications to Determinant Maximization

Nima Anari **Thuy-Duong Vuong**

Stanford

Conference on Learning Theory (COLT)
October 30, 2022

Optimization

Input: $\mu : \Omega \to \mathbb{R}_{\geq 0}$
Output: $x^* := \arg\max_{x \in \Omega} \mu(x)$.

Sampling

Input: $\mu : \Omega \to \mathbb{R}_{\geq 0}$
Output: x with probability proportional to $\mu(x)$.

Connection between sampling and optimization?

- In continuous domain (think $\Omega \equiv \mathbb{R}^n$), tractable functions for sampling and optimization are basically the same class, and they stem from convexity.
  E.g.: log concave μ i.e. $\mu(x) = \exp(f(x))$ for $f : \mathbb{R}^n \to \mathbb{R}$ concave.

- In continuous domain (think $\Omega \equiv \mathbb{R}^n$), tractable functions for sampling and optimization are basically the same class, and they stem from convexity.
  E.g.: log concave μ i.e. $\mu(x) = \exp(f(x))$ for $f : \mathbb{R}^n \to \mathbb{R}$ concave.
- What about discrete Ω?
  Here we study the domain $\binom{n}{k}$, many other domains can be converted to this one [Anari-Liu-OveisGharan–FOCS'20]

The connection between discrete sampling & optimization is unclear.

	Bipartite independent set	DPPs
Sampling	hard	easy
Optimization	easy	hard

Main result

Sampling $\Rightarrow$ Optimization

If we can sample from μ and its scaling using local random walks then can (approximately) optimize over μ using local search.

Scaling of a function

Let $\mu : \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a function, and $\lambda = (\lambda_i)_{i \in [n]} \in \mathbb{R}^n_{\geq 0}$ then the scaling of μ by λ is defined by

$$\lambda * \mu(x) \propto \mu(x) \exp(\langle \log \lambda, x \rangle)$$

Scaling of a function

Let $\mu : \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a function, and $\lambda = (\lambda_i)_{i\in[n]} \in \mathbb{R}^n_{\geq 0}$ then the scaling of μ by λ is defined by

$$\lambda * \mu(x) \propto \mu(x) \exp(\langle \log \lambda, x\rangle)$$

For continuous $\mu : [0,1]^n \to \mathbb{R}_{\geq 0}$: scaling preserves convexity (i.e. log concavity)

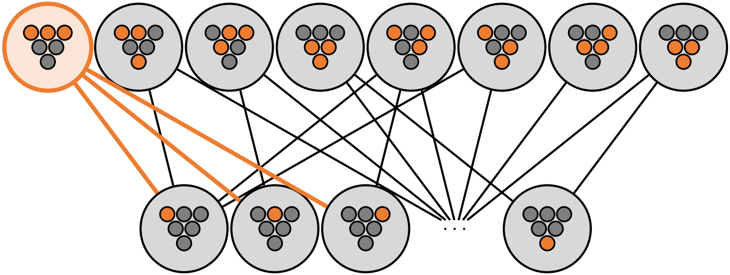
Scaling of a function

Let $\mu : \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a function, and $\lambda = (\lambda_i)_{i \in [n]} \in \mathbb{R}^n_{\geq 0}$ then the scaling of μ by λ is defined by

$$\lambda * \mu(x) \propto \mu(x) \exp(\langle \log \lambda, x \rangle)$$

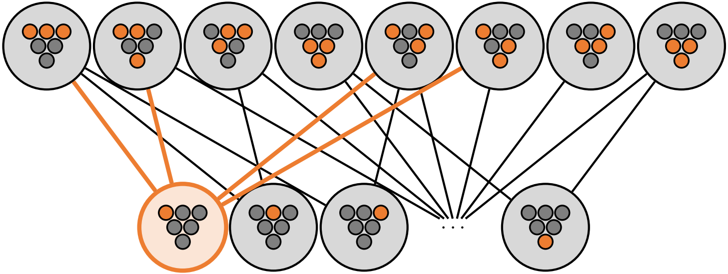
For continuous $\mu : [0,1]^n \to \mathbb{R}_{\geq 0}$: scaling preserves convexity (i.e. log concavity)
For many discrete μ : scaling preserves "nice" properties too.

Random walk $k \leftrightarrow (k - \ell)$ (multi-step down-up walk)



1. Drop ℓ element uniformly at random.

Random walk $k \leftrightarrow (k - \ell)$ (multi-step down-up walk)



1. Drop ℓ element uniformly at random.
2. Add ℓ element with probability $\propto \mu$(resulting set).

Local search

- Local Search$_1$: Start with $S \in \binom{[n]}{k}$. Swap $i \in S$ for $j \notin S$ to improve $\mu(S)$ till can't.

Local search

- Local Search$_r$: Start with $S \in \binom{[n]}{k}$. Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq r$ to improve $\mu(S)$ till can't.

Local search

- Local Search$_r$: Start with $S \in \binom{[n]}{k}$. Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq r$ to improve $\mu(S)$ till can't.
- Local Search$_r$ (LS$_r$) outputs $S := \arg\max \mathcal{N}_r(S)$ where $\mathcal{N}_r(S) := \{W : |S \setminus W| \leq r\}$.

Main result

Sampling $\Rightarrow$ Optimization

If we can sample from $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ and its scaling using ℓ-steps down-up walk in "time" $k^{O(1)}$ then can get $k^{O(k)}$-approximation of $\max \mu(\cdot)$ using Local Search$_\ell$.

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \leq \mu(S) k^{O(\ell k)}$.

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \le \mu(S) k^{O(\ell k)}$.

- Scale μ s.t. $\mu'(S) = \mu'(T) = \mu(S)$ for $\mu' = \lambda * \mu$ with

$$\lambda_i = \begin{cases} (\frac{\mu(S)}{\mu(T)})^{1/k} \text{ if } i \in T \\ 1 \text{ if } i \in S \\ 0 \text{ else} \end{cases}$$

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \leq \mu(S) k^{O(\ell k)}$.

- Scale μ s.t. $\mu'(S) = \mu'(T) = \mu(S)$ for $\mu' = \lambda * \mu$ with
$$\lambda_i = \begin{cases} (\frac{\mu(S)}{\mu(T)})^{1/k} \text{ if } i \in T \\ 1 \text{ if } i \in S \\ 0 \text{ else} \end{cases}$$
- $k^{O(1)}$-mixing implies

$$k^{-\Omega(1)} \leq \Phi = \min_{\mu'(\mathcal{S}) \leq \mu'(\Omega)/2} \frac{Q(\mathcal{S}, \Omega \setminus \mathcal{S})}{\mu'(\mathcal{S})} \leq \frac{Q(\{S\}, \Omega \setminus \{S\})}{\mu'(S)}$$
$$= \binom{k}{\ell}^{-1} \sum_{U_1 \in \binom{S}{\ell}} \sum_{\substack{W \supseteq S \setminus U_1 \\ W \in \text{supp}(\mu') \setminus \{S\}}} \frac{\mu'(W)}{\mu'(S \setminus U_1)}$$

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \leq \mu(S)k^{O(\ell k)}$.

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \le \mu(S) k^{O(\ell k)}$.

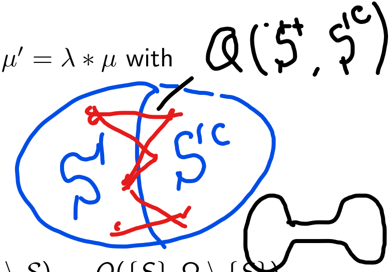
- Scale μ s.t. $\mu'(S) = \mu'(T) = \mu(S)$ for $\mu' = \lambda * \mu$ with
$$\lambda_i = \begin{cases} (\frac{\mu(S)}{\mu(T)})^{1/k} \text{ if } i \in T \\ 1 \text{ if } i \in S \\ 0 \text{ else} \end{cases}$$

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \leq \mu(S) k^{O(\ell k)}$.

- Scale μ s.t. $\mu'(S) = \mu'(T) = \mu(S)$ for $\mu' = \lambda * \mu$ with

$$\lambda_i = \begin{cases} \left(\frac{\mu(S)}{\mu(T)}\right)^{1/k} \text{ if } i \in T \\ 1 \text{ if } i \in S \\ 0 \text{ else} \end{cases}$$



- $k^{O(1)}$-mixing implies

conductance

min-cut

$$k^{-\Omega(1)} \leq \Phi = \min_{\mu'(\mathcal{S}) \leq \mu'(\Omega)/2} \frac{Q(\mathcal{S}, \Omega \setminus \mathcal{S})}{\mu'(\mathcal{S})} \leq \frac{Q(\{S\}, \Omega \setminus \{S\})}{\mu'(S)}$$

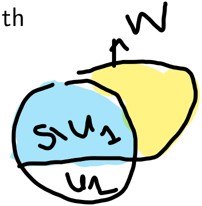
$$= \binom{k}{\ell}^{-1} \sum_{U_1 \in \binom{S}{\ell}} \sum_{\substack{W \supseteq S \setminus U_1 \\ W \in \text{supp}(\mu') \setminus \{S\}}} \frac{\mu'(W)}{\mu'(S \setminus U_1)}$$

Proof sketch

$S \equiv \ell$-neighborhood optima (output of redLocal Search$_\ell$).
$T \equiv$ OPT. WLOG assume $T \cap S = \emptyset$.
We will show $\mu(T) \le \mu(S) k^{O(\ell k)}$.

- Scale μ s.t. $\mu'(S) = \mu'(T) = \mu(S)$ for $\mu' = \lambda * \mu$ with

$$\lambda_i = \begin{cases} \left(\frac{\mu(S)}{\mu(T)}\right)^{1/k} \text{ if } i \in T \\ 1 \text{ if } i \in S \\ 0 \text{ else} \end{cases}$$



- $k^{O(1)}$-mixing implies

$$k^{-O(1)} \le \Phi = \min_{\mu'(\mathcal{S}) \le \mu'(\Omega)/2} \frac{Q(\mathcal{S}, \Omega \setminus \mathcal{S})}{\mu'(\mathcal{S})} \le \frac{Q(\{S\}, \Omega \setminus \{S\})}{\mu'(S)}$$

$$= \binom{k}{\ell}^{-1} \sum_{U_1 \in \binom{S}{\ell}} \sum_{\substack{W \supseteq S \setminus U_1 \\ W \in \text{supp}(\mu') \setminus \{S\}}} \frac{\mu'(W)}{\mu'(S \setminus U_1)}$$

Proof sketch (continue)

$\exists U_1 : \sum_{W} \frac{\mu'(W)}{\mu'(S \setminus U_1)} \geq \Phi \geq k^{-O(1)}$

Hence there must be $W \subseteq T \cup S$ with $1 \leq |W \setminus S| = |W \cap T| \leq \ell$ s.t.

$$\mu(S) = \mu'(S) \leq \mu'(S \setminus U_1) \leq k^{\ell + O(1)} \mu'(W) = k^{\ell + O(1)} \mu(W) (\frac{\mu(S)}{\mu(T)})^{|W \cap T|/k}$$

Proof sketch (continue)

Hence there must be $W \subseteq T \cup S$ with $1 \leq |W \setminus S| = |W \cap T| \leq \ell$ s.t.

$$\mu(S) = \mu'(S) \leq \mu'(S \backslash U_1) \leq k^{\ell+O(1)} \mu'(W) = k^{\ell+O(1)} \mu(W) \left(\frac{\mu(S)}{\mu(T)}\right)^{|W \cap T|/k}$$

By local optimality of S, $\mu(W) \leq \mu(S)$ thus

$$\mu(T) \leq (k^{\ell+O(1)})^{k/|W \cap T|} \mu(S) \leq k^{\ell k + O(k)} \mu(S).$$

Applications

Optimization (MAP-inference) for
nonsymmetric determinantal point processes (DPPs)

Determinantal point processes (DPPs)

Determinantal point processs (DPP) with kernel $L \in \mathbb{R}^{n\times n}$:

$$\mu_L(S) = \det(L_S) \qquad \forall S \subseteq [n]$$

Determinantal point processes (DPPs)

Determinantal point processs (DPP) with kernel $L \in \mathbb{R}^{n \times n}$:

$$\mu_L(S) = \det(L_S) \qquad \forall S \subseteq [n]$$

$$\begin{pmatrix} & & 1 & & 1 & 0 \\ & & 5 & & 2 & 4 \\ 4 & 8 & 9 & 5 & 3 & 3 \\ & & 9 & & 2 & 3 \\ 3 & 7 & 9 & 5 & 3 & 3 \\ 4 & 8 & 6 & 1 & 3 & 0 \end{pmatrix}$$

Figure: $n = 6, S = \{1, 2, 4\}$. L_S is the red submatrix.

Determinantal point processes (DPPs)

Determinantal point processs (DPP) with kernel $L \in \mathbb{R}^{n \times n}$:

$$\mu_L(S) = \det(L_S) \qquad \forall S \subseteq [n]$$

$$\begin{pmatrix} & & 1 & & 1 & 0 \\ & & 5 & & 2 & 4 \\ 4 & 8 & 9 & 5 & 3 & 3 \\ & & 9 & & 2 & 3 \\ 3 & 7 & 9 & 5 & 3 & 3 \\ 4 & 8 & 6 & 1 & 3 & 0 \end{pmatrix}$$

Figure: $n = 6, S = \{1, 2, 4\}$. L_S is the red submatrix.

Applications: Data summarization [Gong-Chao-Grauman-Sha'14], recommender systems [Gillenwater-Paquet-Koenigstein'16,Wilhelm-Ramanathan-Bonomo-Jain-Chi-Gillenwater'18], image search [Kulesza-Taskar'11] ...

Cardinality constrained DPPs (k-DPPs)

k-DPP with kernel $L \in \mathbb{R}^{n \times n}$ and cardinality constraint k

$$\mu_{L,k}(S) = \det(L_S) \qquad \forall S \subseteq [n], |S| = k$$

Cardinality constrained DPPs (k-DPPs)

k-DPP with kernel $L \in \mathbb{R}^{n \times n}$ and cardinality constraint k

$$\mu_{L,k}(S) = \det(L_S) \qquad \forall S \subseteq [n], |S| = k$$

Useful for application requiring fixed-size output (recommendation system)

Partition constrained DPPs

Partition DPP with kernel $L \in \mathbb{R}^{n \times n}$ and partition constraint $(P_1, ..., P_s), (c_1, \cdots, c_s)$:

$$\mu_{L,\mathcal{P},\vec{c}}(S) = 1[|S \cap P_i| = c_i] \det(L_S)$$

Partition constrained DPPs

Partition DPP with kernel $L \in \mathbb{R}^{n \times n}$ and partition constraint $(P_1, ..., P_s), (c_1, \cdots, c_s)$:

$$\mu_{L,\mathcal{P},\vec{c}}(S) = 1[|S \cap P_i| = c_i] \det(L_S)$$

Useful for fairness

Nonsymmetric DPP

- Traditionally, kernel L is symmetric i.e. $L = L^\intercal$.
  Need L PSD for μ_L to map to positive numbers.

Nonsymmetric DPP

- Traditionally, kernel L is symmetric i.e. $L = L^\intercal$.
  Need L PSD for μ_L to map to positive numbers.
- Symmetric kernel cannot encode positive interaction think $1 =$ , 2 $=$ , want $\mathbb{P}[1,2] \gg \mathbb{P}[1]\,\mathbb{P}[2]$ thus

Nonsymmetric DPP

- Traditionally, kernel L is symmetric i.e. $L = L^{\intercal}$.
  Need L PSD for μ_L to map to positive numbers.
- Symmetric kernel cannot encode positive interaction ☹ think 1=🍴, 2 = 🥄, want $\mathbb{P}[1,2] \gg \mathbb{P}[1]\,\mathbb{P}[2]$ thus

$$\det(L_{\{1,2\}}) \gg \det(L_{\{1\}})\det(L_{\{2\}})$$

$$\Leftrightarrow L_{1,1}L_{1,2} - L_{1,2}L_{2,1} \gg L_{1,1}L_{2,2} \Rightarrow L_{1,2}L_{2,1} \ll 0$$

Nonsymmetric DPP

- Traditionally, kernel L is symmetric i.e. $L = L^\intercal$.
  Need L PSD for μ_L to map to positive numbers.
- Symmetric kernel cannot encode positive interaction ☹
- Nonsymmetric kernel (NDPP) $\rightarrow$ more modelling power
  [Brunel–NEURIPS'18,Gartrell-Brunel-Dohmatob-Krichene–NEURIPS'19] ☺
  Still want kernel to be nonsymmetric PSD $L + L^\intercal \succeq 0$

Nonsymmetric DPP

- Traditionally, kernel L is symmetric i.e. $L = L^\intercal$.
  Need L PSD for μ_L to map to positive numbers.
- Symmetric kernel cannot encode positive interaction ☹
- Nonsymmetric kernel (NDPP) $\rightarrow$ more modelling power [Brunel–NEURIPS'18,Gartrell-Brunel-Dohmatob-Krichene–NEURIPS'19] ☺
  Still want kernel to be nonsymmetric PSD $L + L^\intercal \succeq 0$
- [Gartrell-Brunel-Dohmatob-Krichene–NEURIPS'19,Gartrell-Han-Dohmatob-Gillenwater-Brunel–ICLR'21] introduced the use of nonsymmetric kernels in machine learning applications

Scaling DPPs

- Scaling by $\lambda \in \mathbb{R}^n_{\geq 0}$ transforms $\mu_{L,k}$ into $\mu_{L',k}$ with $L' = \text{diag}(\sqrt{\lambda}) L \text{diag}(\sqrt{\lambda})$.

Scaling DPPs

- Scaling by $\lambda \in \mathbb{R}^n_{\geq 0}$ transforms $\mu_{L,k}$ into $\mu_{L',k}$ with $L' = \text{diag}(\sqrt{\lambda})L\text{diag}(\sqrt{\lambda})$.
- Similar statement holds for partition constrained DPPs.

Scaling DPPs

- Scaling by $\lambda \in \mathbb{R}^n_{\geq 0}$ transforms $\mu_{L,k}$ into $\mu_{L',k}$ with $L' = \text{diag}(\sqrt{\lambda}) L \text{diag}(\sqrt{\lambda})$.
- Similar statement holds for partition constrained DPPs.
- L is symmetric (nonsymmetric resp.) PSD iff L' is symmetric (nonsymmetric resp.).

MAP-inference (optimization)

- Given matrix L, parameter k, find $\arg\max_{|S|=k} \det(L_S)$

MAP-inference (optimization)

- Given matrix L, parameter k, find $\arg\max_{|S|=k} \det(L_S)$
- Symmetric DPP: $2^{O(k)}$-approx [Nikolov'15] with matching NP-hardness [Di Summa-Eisenbrand-Faenza-Moldenhauer'14], practical heuristics get $k^{O(k)}$-approx [Civril–Magdon-Ismail'10,Kathuria-Deshpande'16]

MAP-inference (optimization)

- Given matrix L, parameter k, find $\arg\max_{|S|=k} \det(L_S)$
- Symmetric DPP: $2^{O(k)}$-approx [Nikolov'15] with matching NP-hardness [Di Summa-Eisenbrand-Faenza-Moldenhauer'14], practical heuristics get $k^{O(k)}$-approx [Civril–Magdon-Ismail'10,Kathuria-Deshpande'16]
- Nonsymmetric DPP: multiplicative approx of $\log\det$ under restrictive assumption on L using greedy [Gartrell-Han-Dohmatob-Gillenwater-Brunel–ICLR'21].

MAP-inference (optimization)

- Given matrix L, parameter k, find $\arg\max_{|S|=k} \det(L_S)$
- Symmetric DPP: $2^{O(k)}$-approx [Nikolov'15] with matching NP-hardness [Di Summa-Eisenbrand-Faenza-Moldenhauer'14], practical heuristics get $k^{O(k)}$-approx [Civril–Magdon-Ismail'10,Kathuria-Deshpande'16]
- Nonsymmetric DPP: multiplicative approx of $\log\det$ under restrictive assumption on L using greedy [Gartrell-Han-Dohmatob-Gillenwater-Brunel–ICLR'21]. $\sigma_{\min}, \sigma_{\max} = \min, \max$ singular value of L_Y for $|Y| \leq 2k$. Approx-factor depends on $\sigma_{\max}/\sigma_{\min}$.

Previous works on sampling from DPPs

- For k-DPP with nPSD kernel L ($L + L^\intercal \succeq 0$) :
  [Alimohammadi-Anari-Shiragur-**V.**–STOC'21] 4-steps DU walk mixes in $k^{O(1)}$-time.
  In this work: we improve 4-step to 2-step

Previous works on sampling from DPPs

- For k-DPP with nPSD kernel L ($L + L^\intercal \succeq 0$) :
  [Alimohammadi-Anari-Shiragur-**V.**–STOC'21] 4-steps DU walk mixes in $k^{O(1)}$-time.
  In this work: we improve 4-step to 2-step
- For partition DPPs with symmetric PSD kernel and O(1) partitions, $O(1)$-steps DU walk mixes in $k^{O(1)}$-time.

Previous works on sampling from DPPs

- For k-DPP with nPSD kernel L $(L + L^{\intercal} \succeq 0)$:
  [Alimohammadi-Anari-Shiragur-**V.**–STOC'21] 4-steps DU walk mixes in $k^{O(1)}$-time.
  In this work: we improve 4-step to 2-step
- For partition DPPs with symmetric PSD kernel and O(1) partitions, $O(1)$-steps DU walk mixes in $k^{O(1)}$-time.

Previous works on sampling from DPPs

- For k-DPP with nPSD kernel L $(L + L^{\intercal} \succeq 0)$:
  [Alimohammadi-Anari-Shiragur-**V.**–STOC'21] 4-steps DU walk mixes in $k^{O(1)}$-time.
  In this work: we improve 4-step to 2-step
- For partition DPPs with symmetric PSD kernel and O(1) partitions, $O(1)$-steps DU walk mixes in $k^{O(1)}$-time.
- For k-DPP with symmetric PSD kernel $L(L = L^{\intercal}, L \succeq 0)$:
  [Anari-OveisGharan'15, Hermon-Salez]: 1-step DU walk mixes in $\tilde{O}(k)$-"time".

Main theorem

Sampling $\Rightarrow$ Optimization

If we can sample from μ and its scaling using local random walks then can (approximately) optimize over μ using local search.

Corollaries

- $k^{O(k)}$-approximation for MAP-inference on nonsymmetric DPPs with nPSD kernel L i.e. $L + L^\intercal \succeq 0$ using Local Search$_2$
  LS$_2$: Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq 2$ to improve $\det(L_S)$ till can't.

Corollaries

- $k^{O(k)}$-approximation for MAP-inference on nonsymmetric DPPs with nPSD kernel L i.e. $L + L^\intercal \succeq 0$ using Local Search$_2$
  LS_2 : Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq 2$ to improve $\det(L_S)$ till can't.

Corollaries

- $k^{O(k)}$-approximation for MAP-inference on nonsymmetric DPPs with nPSD kernel L i.e. $L + L^\intercal \succeq 0$ using Local Search$_2$
  LS_2 : Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq 2$ to improve $\det(L_S)$ till can't.
- For partition DPPs on $O(1)$ partitions, Local Search$_r$ with $r = O(1)$ gives $k^{O(k)}$-approximation.

Corollaries

- $k^{O(k)}$-approximation for MAP-inference on nonsymmetric DPPs with nPSD kernel L i.e. $L + L^\intercal \succeq 0$ using Local Search$_2$
  LS$_2$: Swap $U \subseteq S$ for $V \subseteq S^c$ with $|U| = |V| \leq 2$ to improve $\det(L_S)$ till can't.
- For partition DPPs on $O(1)$ partitions, Local Search$_r$ with $r = O(1)$ gives $k^{O(k)}$-approximation.
- Recover $k^{O(k)}$-approximation for symmetric DPP using Local Search$_1$.

Summary: MAP-inference for nonsymmetric DPPs

- We obtain first $k^{O(k)}$-approximation for NDPPs
- Other popular heuristics for symmetric DPP like LS_1 or Greedy don't work for nonsymmetric DPP!

Summary: MAP-inference for nonsymmetric DPPs

	Symmetric PSD	Nonsymmetric PSD
Greedy	$k^{O(k)}$ [CM10]	∞
LS_1	$k^{O(k)}$ [KD16]	∞
LS_2	$k^{O(k)}$ "	$k^{O(k)}$ [A**V**'21]

Table: Approximation guarantee for MAP-inference on symmetric vs. nonsymmetric DPP.

Discussion

- $k^{O(k)}$ is optimal for Local Search$_r$ for $r = O(1)$

Discussion

- $k^{O(k)}$ is optimal for Local Search$_r$ for $r = O(1)$
- NP hard to get c^k for some $c > 1$, even for symmetric PSD

Discussion

- $k^{O(k)}$ is optimal for Local Search$_r$ for $r = O(1)$
- NP hard to get c^k for some $c > 1$, even for symmetric PSD
- Getting $e^{O(k)}$ for nonsymmetric PSD implies $O(1)$-approximation for determinantal lowerbound [Lovasz-Spencer-Vesztergombi'86] and hereditary discrepancy (long time open problem, see [Jiang-Res–SOSA'22] for recent progress)

$$\text{detlb}(A) = \max_k \max_{I \subseteq [m], J \subseteq [n], |I| = |J| = k} |\det(A_{I,J})|^{1/k}$$

$$\text{herdisc}(A) = \max_S \min_{x \in [-1,1]^n} ||A_{|S} x||_\infty$$

Discussion

- $k^{O(k)}$ is optimal for Local Search$_r$ for $r = O(1)$
- NP hard to get c^k for some $c > 1$, even for symmetric PSD
- Getting $e^{O(k)}$ for nonsymmetric PSD implies $O(1)$-approximation for determinantal lowerbound [Lovasz-Spencer-Vesztergombi'86] and hereditary discrepancy (long time open problem, see [Jiang-Res–SOSA'22] for recent progress)

$$\operatorname{detlb}(A) = \max_k \max_{I\subseteq[m], J\subseteq[n], |I|=|J|=k} |\det(A_{I,J})|^{1/k}$$

$$\operatorname{herdisc}(A) = \max_S \min_{x\in[-1,1]^n} ||A_{|S}x||_\infty$$

- Thank you!