Gibbs Sampling from k-Determinantal Point Processes

Alireza Rezaei

University of Washington

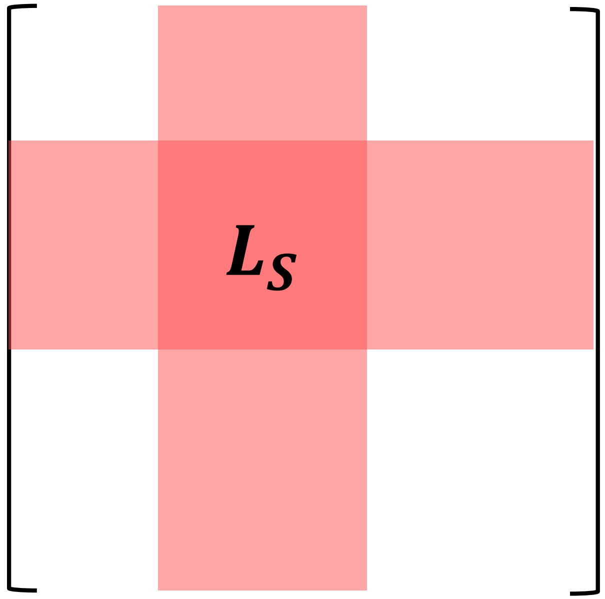
Based on joint work with

Shayan Oveis Gharan

Point Process: A distribution on subsets of $[N] = \{1,2,\dots,N\}$.

Determinantal Point Process: There is a PSD kernel $L \in \mathbb{R}^{N\times N}$ such that

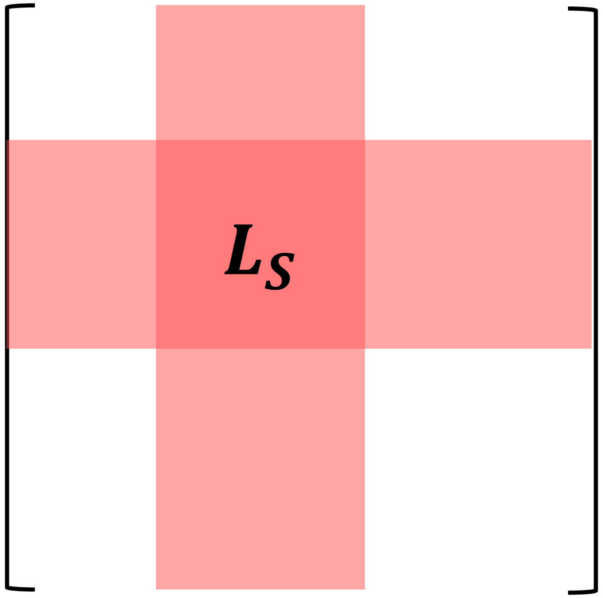
$$\forall S \subseteq [N]:\ \ \mathbb{P}[S] \propto \det(L_S)$$



Point Process: A distribution on subsets of $[N] = \{1,2,\dots,N\}$.

Determinantal Point Process: There is a PSD kernel $L \in \mathbb{R}^{N\times N}$ such that

$$\forall S \subseteq [N]: \ \ \mathbb{P}[S] \propto \ \det(L_S)$$



***k*-DPP:** Conditioning of a DPP on picking subsets of size k

Focus of the talk: Sampling from k-DPPs

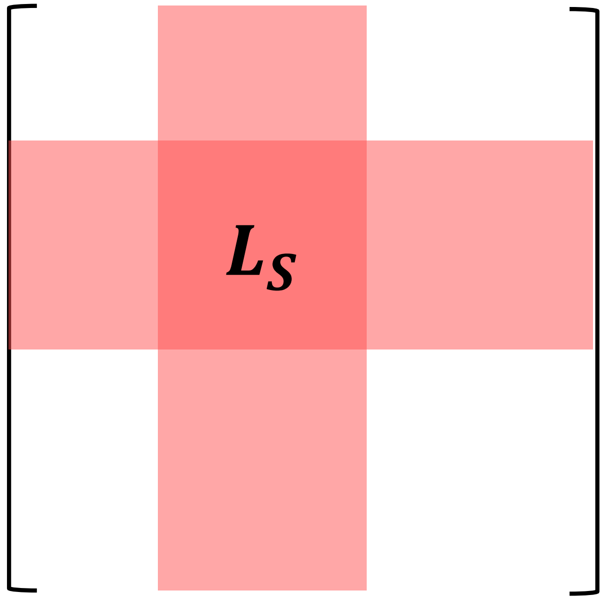
if $|S| = k$: $\mathbb{P}[S] \propto \det(L_S)$

otherwise : $\mathbb{P}[S] = 0$

Point Process: A distribution on subsets of $[N] = \{1,2,\ldots,N\}$.

Determinantal Point Process: There is a PSD kernel $L \in \mathbb{R}^{N\times N}$ such that

$$\forall S \subseteq [N]: \ \mathbb{P}[S] \propto \det(L_S)$$



k-DPP: Conditioning of a DPP on picking subsets of size k

Focus of the talk: Sampling from k-DPPs

if $|S| = k$: $\mathbb{P}[S] \propto \det(L_S)$

otherwise : $\mathbb{P}[S] = 0$

DPPs are Very popular probabilistic models in machine learning to capture diversity.

Applications [Kulesza-Taskar'11, Dang'05, Nenkova-Vanderwende-McKeown'06, Mirzasoleiman-Jegelka-Krause'17]

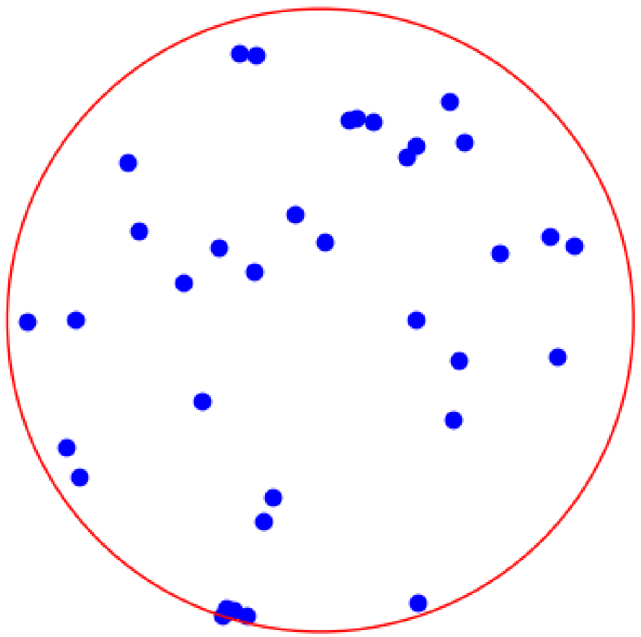
— Image search, document and video summarization, tweet timeline generation, pose estimation, feature selection

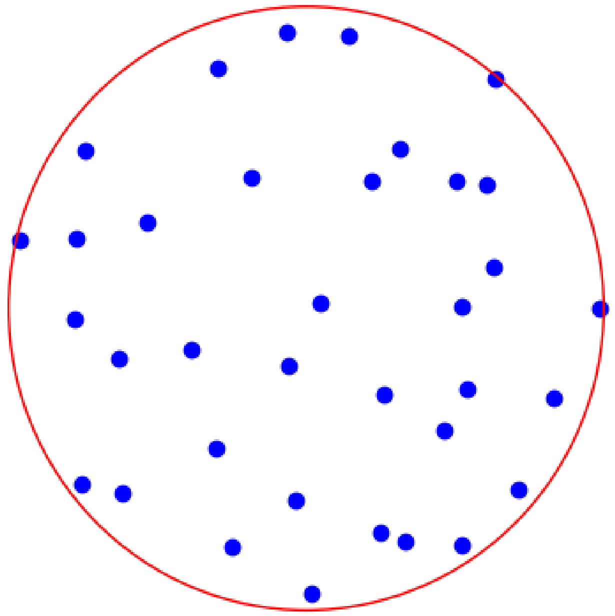
Continuous Domain

Input: PSD operator $L: \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ and k

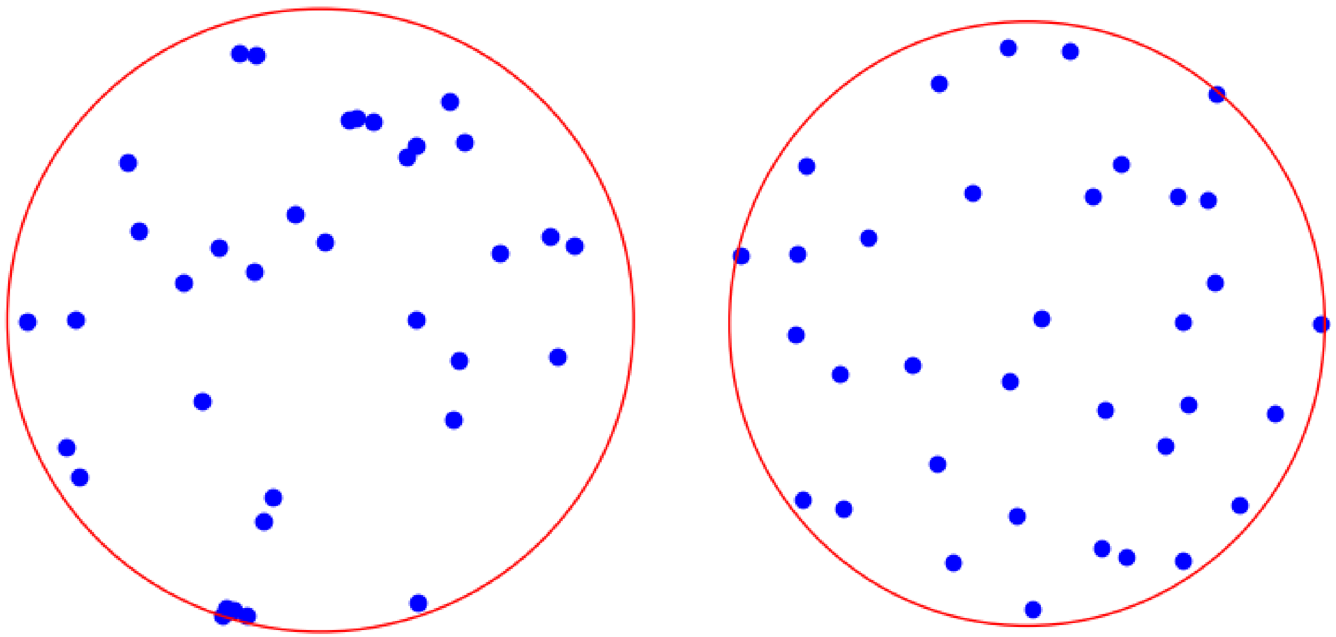
select a subset $S \subset \mathcal{C}$ with k points from a distribution with PDF function

$$p(S) \propto \det\left(\{L(x, y)\}_{x,y \in S}\right)$$





Continuous Domain



Input: PSD operator $L: \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ and k

select a subset $S \subset \mathcal{C}$ with k points from a distribution with PDF function

$$p(S) \propto \det\left(\{L(x,y)\}_{x,y \in S}\right)$$

Ex. Gaussian : $L(x,y) = \exp\left(-\frac{(x-y)\Sigma^{-1}(x-y)}{2}\right)$

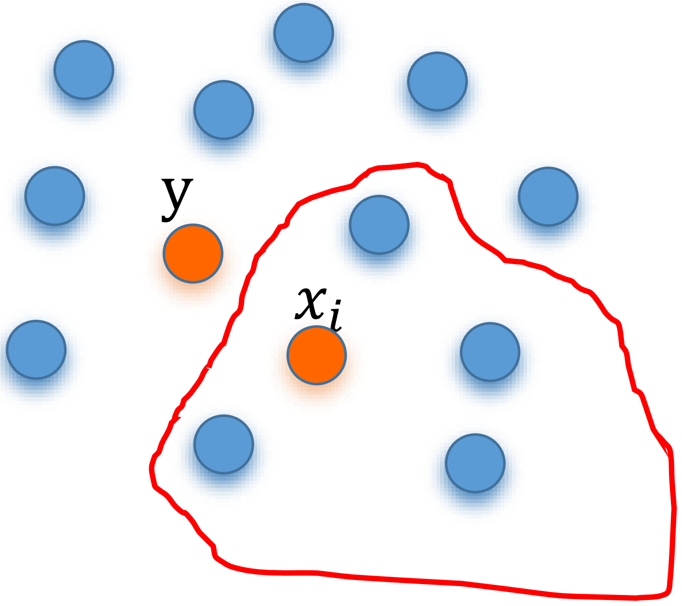
Applications.

- Hyper-parameter tuning [Dodge-Jamieson-Smith'17]
- Learning mixture of Gaussians [Affandi-Fox-Taskar'13]

Random Scan Gibbs Sampler for K-DPP

1. Stay at the current state $S = \{x_1, \dots x_k\}$ with prob $\frac{1}{2}$.
2. Choose $x_i \in S$ u.a.r
3. Choose $y \notin S$ from the conditional dist $\pi(\,.\,|\,S - x_i$ is chosen)

   Continuous: $\mathrm{PDF}(y) \propto \pi(x_1, \dots x_{i-1}, y, x_{i+1}, \dots, x_k))$



$S \in \binom{[N]}{k}$

Main Result

Given a k-DPP π, an "approximate" sample from π can be generated by running the Gibbs sampler for $\boldsymbol{\tau = \tilde{O}(k^4) \cdot \log\left(\mathrm{var}_\pi\left(\frac{p_\mu}{p_\pi}\right)\right)}$ steps where μ is the starting dist.

Main Result

Given a k-DPP π, an "approximate" sample from π can be generated by running the Gibbs sampler for $\boldsymbol{\tau = \tilde{O}\left(k^4\right) \cdot \log\left(\mathrm{var}_\pi\left(\frac{p_\mu}{p_\pi}\right)\right)}$ steps where μ is the starting dist.

Discrete: A simple greedy initialization gives $\tau = O\left(k^5 \log k\right)$. Total running time is $O(N).\mathrm{poly}(k)$.

- Does not improve upon the previous MCMC methods. [Anari-Oveis Gharan-R'16]
- Mixing time is independent of N, so the running time in distributed settings is sublinear.

Main Result

Given a k-DPP π, an "approximate" sample from π can be generated by running the Gibbs sampler for $\boldsymbol{\tau = \tilde{O}(k^4) \cdot \log\left(\mathrm{var}_\pi\left(\frac{p_\mu}{p_\pi}\right)\right)}$ steps where μ is the starting dist.

Discrete: A simple greedy initialization gives $\tau = O\,(k^5 \log k)$. Total running time is $O(N).\,\mathrm{poly}(k)$.

- Does not improve upon the previous MCMC methods. [Anari-Oveis Gharan-R'16]
- Mixing time is independent of N, so the running time in distributed settings is sublinear.

Being able to run the chain.

Continuous: Given access to conditional oracles, μ can be found so $\tau = O(k^5 \log k)$.

- First algorithm with a theoretical guarantee for sampling from continuous k-DPP.

Main Result

Given a k-DPP π, an "approximate" sample from π can be generated by running the Gibbs sampler for $\boldsymbol{\tau = \tilde{O}(k^4) \cdot \log\left(\mathrm{var}_\pi\left(\frac{p_\mu}{p_\pi}\right)\right)}$ steps where μ is the starting dist.

Discrete: A simple greedy initialization gives $\tau = O\,(k^5 \log k)$. Total running time is $O(N).\,\mathrm{poly}(k)$.

- Does not improve upon the previous MCMC methods. [Anari-Oveis Gharan-R'16]
- Mixing time is independent of N, so the running time in distributed settings is sublinear.

Being able to
run the chain.

Continuous: Given access to conditional oracles, μ can be found so $\tau = O(k^5 \log k)$.

- First algorithm with a theoretical guarantee for sampling from continuous k-DPP.
- Using a rejection sampler as the conditional oracles for Gaussian kernels $L(x,y) = \exp(-\frac{\|x-y\|^2}{\sigma^2})$ defined a unit sphere in $\mathbb{R}^d$, the total running time is
  - If k =poly(d): $\mathrm{poly}(d,\sigma)$
  - If $k \le e^{d^{1-\delta}}$ and $\sigma = O(1)$: $\mathrm{poly}(d) \cdot k^{O(\frac{1}{\delta})}$

Thank you!

Poster: Pacific Ballroom #204