Determinantal Point Processes and Growth Models

Madhav Sankaranarayanan

December 1, 2021

Outline

Outline

1 Introduction
- Prerequisites
- Motivation

2 Characterization of Repulsion using DPPs
- Aztec Diamond
- Corner Growth Model

3 Generation of Growth Models

Definitions

A **Point Process** is a probability distribution defined over the power set of a ground set. A **Determinantal Point Process** is a Point Process, where the probability of a subset J being in the sample is equal to the J-minor of a given marginal kernel matrix K

Definitions

Thus, a Determinantal Point Process can be characterized by the marginal kernel matrix K. An alternate definition involves L ensembles. We can make the probability of selecting a given subset J as being proportional to the J-minor of a positive semidefinite matrix L. The formulations of K and L would be analogous to the CDF and PDF of a distribution, respectively.

Definitions

A **Random Growth Model** are growths in a plane based on a probabilistic mechanism

Properties of DPPs

- Diagonal Kernel Matrix/L Matrix $\implies$ Independent Sampling of Elements of Ground Set
- Rank 1 L Matrix $\implies$ Singleton Sampling

Main Question

What is the connection between Determinantal Point Processes and certain Random Growth Models?

Outline

Repulsion in DPPs

A key characteristic of a DPP is the *diversity* it induces in the point process. That is, sampling from a DPP causes nearby points in the process to have a lesser probability of appearing together. This repulsion can be quantified ([8]).

Intuition for Repulsion

- For the sake of simplicity, let us consider a discrete ground set $\mathcal{X} = \{x_1, \ldots, x_n\}$, and the marginal kernel is symmetric
- We denote a random subset as X. Now

$$P(\{x_i\} \subseteq X) = K_{ii}$$

$$P(\{x_i, x_j\} \subseteq X) = K_{ii}K_{jj} - K_{ij}^2 = P(\{x_i\} \subseteq X)P(\{x_j\} \subseteq X) - K_{ij}^2$$

- This shows that the sampling process induces a negative correlation between the two points.

Intuition for Repulsion

- When considering higher dimensions, or continuous DPPs, a similar repulsive factor exists.
- Particularly in the continuous case, we can see that for a symmetric continuous kernel function, say ψ, the probability of two points (x, y) being in the random subset would be of the form $\psi(x,x)\psi(y,y) - \psi(x,y)^2$, which is much smaller for x close to y

Main motivation

Repulsion of Growth Models $\implies$ DPPs

- Investigate examples where repulsion can be characterized by DPPs
- Determine sampling procedures for the same

Repulsion of DPPs $\implies$ Generation of Growth Models

- Construct framework and restrictions on DPPs for valid growth
- Create a generalization of repulsive growth processes

Examples of Repulsion in Growth Processes

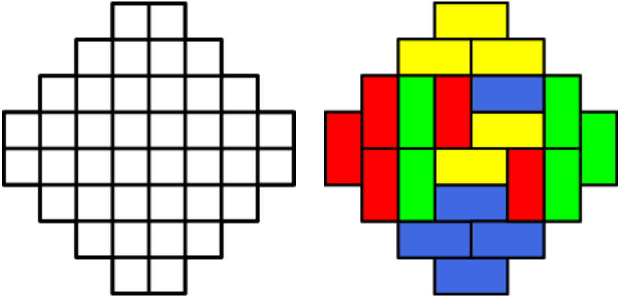
- Aztec Diamond
- Corner Growth Model

These models are linked to DPPs through the concept of non-intersecting paths.

Outline

Setup

- Consider a square lattice where all the centres satisfy $|x| + |y| \leqslant n$
- Consider a checkerboard pattern on this lattice
- Now, tile this structure with dominoes
- Dominoes are characterized by orientation, and position of the black squares. Thus, there are 4 types of dominoes

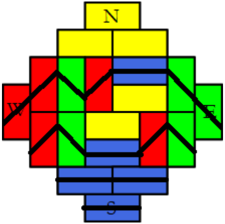


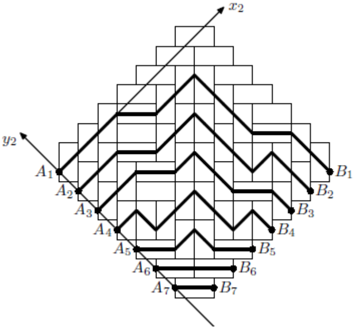
Setup

- We label the 4 types as follows:
  - N domino: horizontal domino with left white
  - S domino: horizontal domino with right white
  - W domino: vertical domino with upper white
  - E domino: vertical domino with lower white
- We construct a non-intersecting path within each domino as follows:
  - N domino: No path
  - W domino: From (0,0.5) to (1,1.5)
  - E domino: From (0,1.5) to (1,0.5)
  - S domino: From (0,0.5) to (2,0.5)
- Consider a weighting system on the different dominoes

Using this weighting system, there is a kernel matrix for the DPP related to the Aztec Diamond [11]!

Setup





Outline

Setup

- Corner Growth Model with the Geometric setup
- Consider a *right/down* path as a path from $(0, L)$ to $(K, 0)$ using only right or down moves
- Each of these paths can be assigned to a partition $\lambda_k = \max\{x_j ; y_j = k - 1\}$
- These partitions are used to correspond to a graph $\mathcal{G}(\lambda)$ with weighted edges
- From here, we have a non-intersecting path associated with the corner growth model, and thus a kernel matrix as well [11]

Setup

- Consider the corner growth process model again
- Note that each potential square was chosen with a probability p at every time point $\implies$ Binomial RV of squares added
- Probability of all squares or no squares is very low

Setup

- Instead of probability p for each square, consider DPP on the set of available squares
- That is, at time point T, if the set of possible squares is $C_T \subset \mathbb{N}^2$, construct DPP using C_T as ground set
- Potentially $\mathbb{N}^2$ as ground set

Important Questions

- What matrix properties should K (or L) have?
- Construct new DPP at every step, or using some other generation scheme?
- Nested DPPs?
- Fixed kernel or dynamic kernel?

Bernoulli Corner Growth as DPP

- In the Bernoulli Corner Growth, at the time point T, every square in C_T has a probability p of belonging to the subset of squares that get selected at the given time point
- This can be modelled using a kernel matrix pI_{C_T}
- Bernoulli Corner Growth is a special case of “DPP Growth”

Alternate Expression

- What if the kernel matrix is pJ_{C_T}?
- Only valid if $p = \frac{1}{n}$
- Selects exactly one square, slower growth
- Note that this is not an elementary DPP, so sampling is not direct

Special Case

- Consider a DPP on the same setup with L matrix as of the αI_{C_T}
- Similar to Projection DPP form, if we take $\alpha \to \infty$, we see that the probability of choosing the whole set tends to 1
- At every iteration, almost every square is selected

Simulations

- We can begin simulations with this setup
- Investigate the limiting curves
- Now, constructing a new matrix in each iteration is very inefficient computationally

Simulations

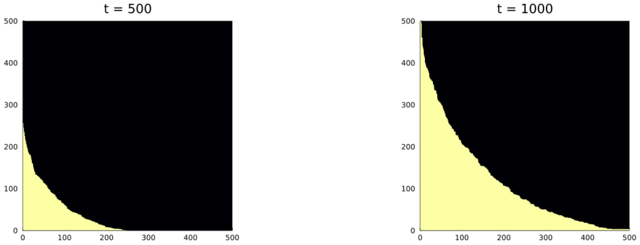


Figure: $p = 0.5$ growth

Simulations

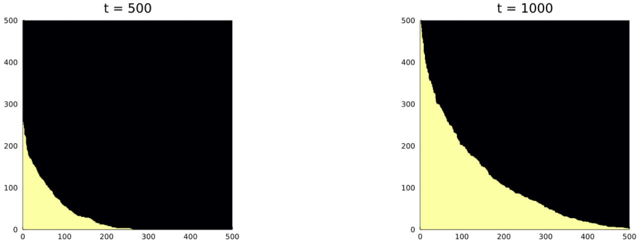


Figure: $L = I_{C_T}, K = 0.5 * I_{C_T}$ growth

Simulations

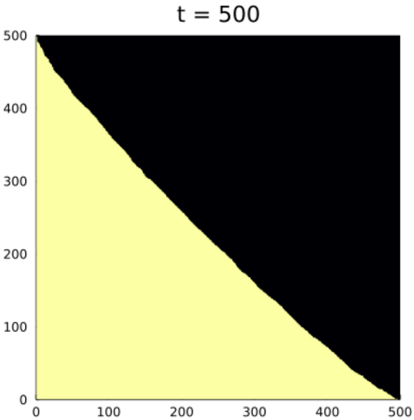


Figure: $L = \alpha I_{C_T}, \alpha = 1000$ growth

Simulations

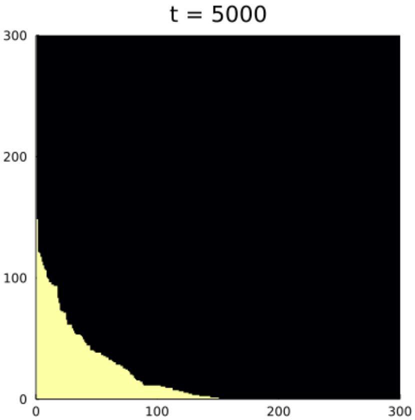


Figure: $L = kJ$ growth. Note that the value of k does not matter

More Complex Structures

Before we generalize further, we must find a way of preserving the structure of the DPP, while shifting in the dimensions. There are two ways this can be achieved:

- Use an embedding/trimming algorithm that preserves rank
- Construct a DPP on the whole lattice, and subset the given DPP

More Complex Structures

From a theoretical standpoint, both methods would work. The key to such a procedure would be to maintain the necessary properties for the kernel matrix (low rank, higher emphasis on initial elements of the ground set)

For embedding and trimming, there are theoretical guarantees in the low rank case [12]. Since the whole lattice method requires trimming, the same guarantees apply.

From a computational standpoint, both are quite inefficient, but there may be a dynamic programming approach for the second.

More Complex Structures

One structure I experimented with (under the independence regime) was one which made older squares more likely to be selected. There are two ways to order the possible squares: left to right (as is given in the code), distance from origin (ℓ_1 is an intuitive choice). Using the second ordering, we can construct L matrix that has larger elements along the diagonal.

Simulations

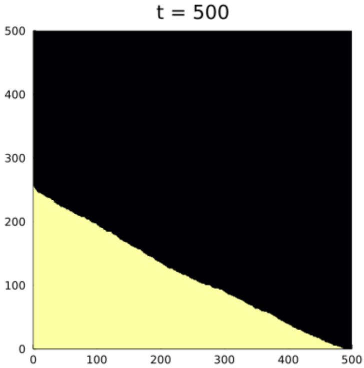


Figure: After using the second ordering of squares, the matrix has the probability of not choosing inversely proportional to the squares ranking ($L = \mathrm{diag}(\frac{i}{i+1})$. We can see that the curve grows in this interesting fashion. This is due to the way the squares were initially sorted.

Simulations

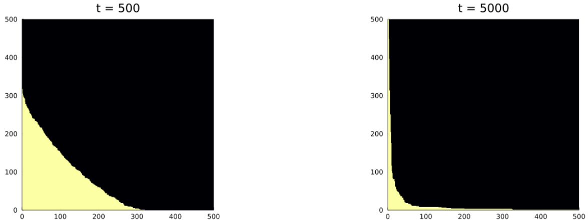


Figure: Consider the penalization linear in terms of the ℓ_1 norm. Construct large matrix L, and subset based on ℓ_1 norm of the possible squares. I tried the growths in both directions

Random DPPs

Now, we can construct a random DPP, use the first ordering of the squares, and run a simulation. From the previous simulations, we expect a similar growth as the Bernoulli, since the repulsion does not manifest itself in a direct way.
Now, my laptop could not run the simulations after a certain size, so I am showing the evolution for lower values.

Simulations

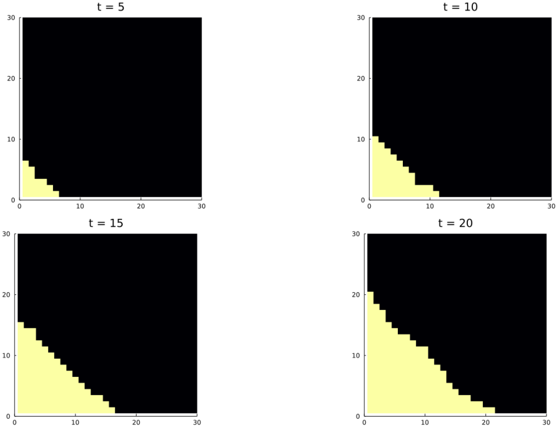


Figure: A random positive semidefinite matrix used for sampling

Kernel dependent on square position

The next step would be consider a kernel that depends on the squares position, as this would allow the repulsion to manifest itself.

References I

- Introduction to determinantal point processes. `https://project.inria.fr/siamsummerschool/files/2020/01/Lecture02-intro-to-dpps.pdf`.
- High-performance sampling of generic determinantal point processes. `https://hodgestar.com/storage/hpc_dpp_preprint-April30-2019.pdf`.
- Introduction to dpps. `http://people.csail.mit.edu/stefje/fall15/notes_lecture21.pdf`.
- Longest increasing subsequence problem. `https://www.normalesup.org/~bouttier/coursM2Lyon/notes_coursM2Lyon.pdf`.

References II

Alexei Borodin, Persi Diaconis, and Jason Fulman. On adding a list of numbers (and other one-dependent determinantal processes), 2009.

Alex Kulesza. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Christophe Ange Napoléon Biscio and Frédéric Lavancier. Quantifying repulsiveness of determinantal point processes. *Bernoulli*, 22(4), Nov 2016.

References III

Alexei Borodin and Grigori Olshanski. Distributions on partitions, point processes,¶ and the hypergeometric kernel. *Communications in Mathematical Physics*, 211(2):335–358, Apr 2000.

Kurt Johansson. Shape fluctuations and random matrices. *Communications in Mathematical Physics*, 209(2):437–476, Feb 2000.

Kurt Johansson. Random matrices and determinantal processes. *arXiv preprint math-ph/0510038*, 2005.

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online learning in the embedded manifold of low-rank matrices. *Journal of Machine Learning Research*, 13(2), 2012.