Determinantal Point Processes: Theory and Simulations

Advay Koranne

Advised by: Professor Sungwoo Jeong

Cornell University

May 6th, 2024

Acknowlegements

First of all, I would like to thank my parents Sandeep Koranne and Jyoti Aneja. I would also like to thank the friends I have made during my four years here at Cornell University who have encouraged me to explore new fields and opportunities outside of my coursework. Lastly, I would like to thank Professor Sungwoo Jeong as without his guidance this thesis would not have been possible.

Contents

Overview

Overview

1. DPPs are probabilistic models that have negative correlation and can allow for computationally efficient algorithms for sampling, marginalization, conditioning, and other tasks.
2. DPPs are also stochastic point processes with the characteristic that the probability distribution is characterized as a determinant of some matrix.

Point Processes Definition

Point Processes

A point process on a ground set Y is a probability measure on the power set of Y (2^Y).

Point Processes Example

Examples

1. Given a set of elements: $\{\text{apple}, \text{banana}, \text{orange}, \text{mandarins}\}$
2. A realization of this set could be: $\{\text{apple}, \text{banana}\}$
3. **Attraction:** if two elements tend to be together in a subset – this would mean that oranges and mandarins appear in a subset together.
4. **Repulsion:** if this is not true – i.e. we have that mandarins and oranges do not appear together.

DPPs are a kind of point process with repulsion where points are selected to encourage diversity.

Determinantal Point Process Definition

Determinantal Point Process

A point process P is called a determinantal point process, if Y is a random subset drawn according to P, then we have for every $S \subseteq Y$: $P(S \subseteq Y) = \det(K_s)$ for K some similarity matrix $K \in \mathbb{R}^{n \times n}$.

DPP Example

1. Let us assume that our subset A has a single element ($A = \{i\}$)
2. $P(i \subseteq Y) = K_{ii}$
3. Now, assume $A = \{i, j\}$

$$
\begin{aligned}
P(i, j \subseteq Y) &= \begin{vmatrix} K_{ii} & Kij \\ K_{ji} & K_{jj} \end{vmatrix} \\
&= K_{ii}K_{jj} - K_{ji}K_{ij} \\
&= P(i \in Y)P(j \in Y) - K_{ij}^2
\end{aligned}
$$

K_{ij} determines the negative correlation between two elements. If the value of K_{ij} is large then there is a low probability that i and j will appear together. When $K_{ij} = \sqrt{K_{ii}K_{jj}}$ then i and j are identical and will not appear together at all

Gaussian Kernel Example

$$L_{ij} = \exp(-\frac{1}{2l^2}||x_i - x_j||^2)$$

In the images below we have plotted $n = 50$ points randomly and picked $k = 3$ and $k = 20$ points using the Gaussian Kernel L.

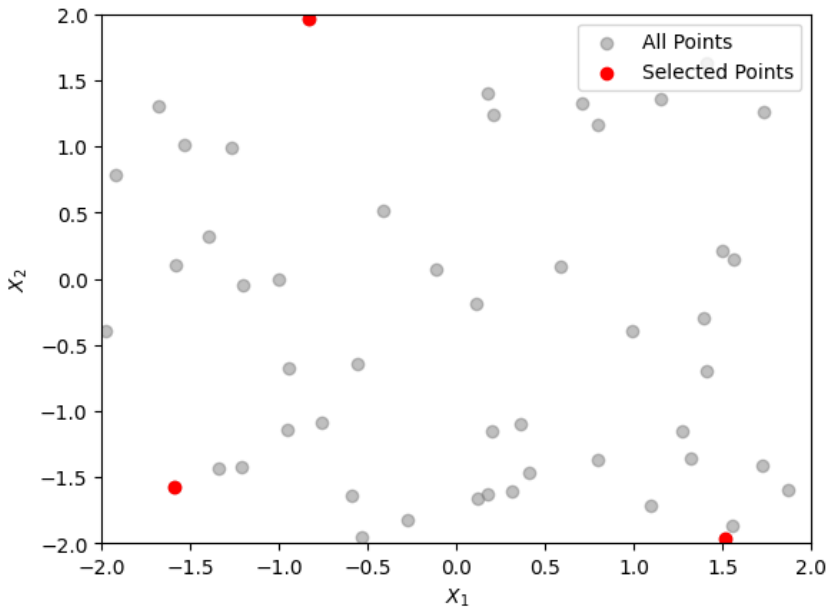

Figure 1: Gaussian kernel Point Process ($k = 3$)

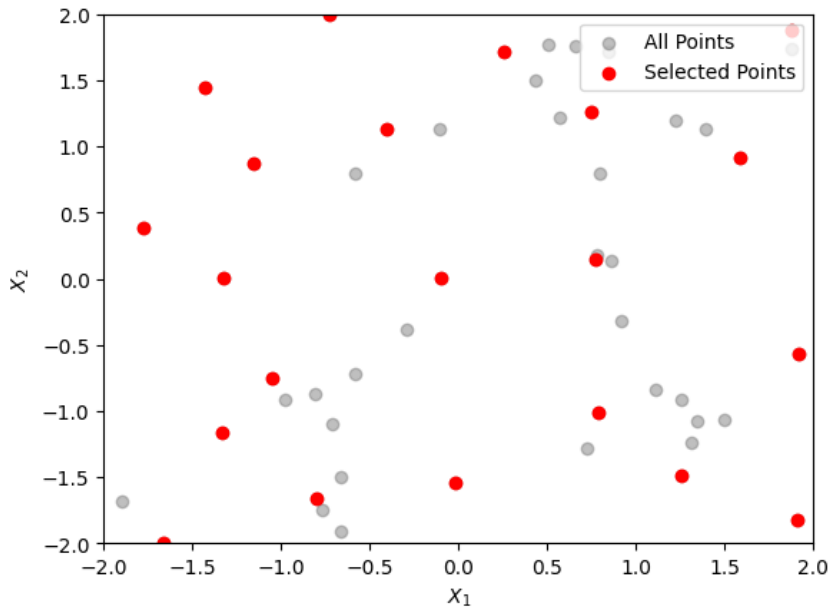

Figure 2: Gaussian kernel Point Process ($k = 20$)

L-ensembles

L-ensembles

1. For data modeling it is helpful to look at a restricted version of DPPs called L-ensembles.
2. A L-ensemble is a DPP, but rather than the marginal kernel K there is a symmetric matrix L that is indexed by the elements of Y.

L-ensemble Property

$$P_L(X = Y) \quad \propto \quad \det(L_y)$$

L-ensembles Properties

L-ensembles Normalization

For any $A \subseteq \mathcal{Y}$:

$$\sum_{A \subseteq Y \subseteq \mathcal{Y}} \det(L_y) = \det(L + I_{\bar{A}})$$

where we define $I_{\bar{A}}$ as a diagonal matrix with ones in the diagonal positions which are the elements from $\bar{A} = \mathcal{Y} - A$ and zeroes everywhere else.

$$P_L(\mathbf{Y} = Y) \quad = \quad \frac{\det(L_Y)}{\det(L + I)}$$

L-ensembles properties

L-ensemble Marginal Kernel

An L-ensemble is equivalent to a DPP with marginal kernel K:

$$K = L(L+I)^{-1} = I - (L+I)^{-1}$$

Properties

DPP properties

Restriction

The marginal probability of a set $A \subseteq Y$ is

$$P_L(A \subseteq Y) = \det(K_A)$$

DPP properties

Complement

If $\mathbf{Y}$ is distributed as a DPP with a marginal kernel K, then $\mathcal{Y} - \mathbf{Y}$ is also distributed as a DPP, with marginal kernel $\bar{K} = I - K$

$$P(A \cap \mathbf{Y} = \emptyset) = \det\left(\bar{K_A}\right) = \det(I - K_A)$$

DPP properties

Conditioning

The distribution by conditioning on a DPP that has none of the elements in A can be seen as the following. For $B \subseteq \mathcal{Y}$ not intersecting with A we have:

$$P_L(\mathbf{Y} = B | A \cap \mathbf{Y} = \emptyset) = \frac{P_L(\mathbf{Y} = B)}{P_L(A \cap \mathbf{Y}) = \emptyset} = \frac{\det(L_B)}{\det(L_{\bar{A}} + I)}$$

DPP Samplers

DPP Samplers

1. The high-level idea of using a DPP sampler is to select diverse subsets of items from a larger set where the diversity is mathematically quantified by the properties of determinants.
2. A simpler DPP sampler from the paper published in 2019 titled "High-performance sampling of generic Determinantal Point Processes" which has efficient direct sampling schemes for non-Hermitian and Hermitian DPP kernels. [Pou20]

DPP Sampler Algorithm

Algorithm 1 Simple DPP Sampler

Require: K and an integer n

$sample \leftarrow []$
$A \leftarrow K$
for $j = 0$ **to** $n - 1$ **do**
    if Bernoulli(A_j) **then**
        $sample$.append(j)
    else
        $A_j \leftarrow A_j - 1$
    end if
    $A_{j+1:n,j} \leftarrow A_{j+1:n,j}/A_j$
    $A_{j+1:n} \leftarrow A_{j+1:n} - A_{j+1:n,j} \times A_{j,j+1:n}$
end for

Ensure: $sample, A$

Check DPP Sampler

We will now show how we can check if our implementation of our DPP sampler is correct by comparing theoretical values to empirical values. if Y is a random subset drawn to P, then we know that from restriction:

$$P(A \subseteq Y) = \det(K_A)$$

where K is defined as some real symmetric $N \times N$ matrix, where we define $K_\emptyset = 1$ and $K_A = [K_{ij}]_{i,j \in A}$.

Check DPP Sampler Continued

1. Construct a positive semi-definite matrix K by constructing a random $A \in \mathbb{R}^{3\times 3}$ matrix and letting $K = A \cdot A^T$.
2. Divide K by two times the maximum eigenvalue to normalize all the values.
3. Calculate the determinant for every sub-matrix using the power set of $X = \{1, 2, 3\}$. The power set of X is $\{\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

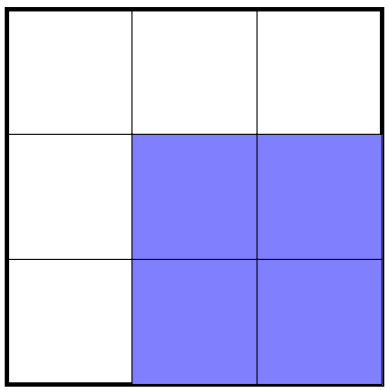

Figure 3: DPP Submatrix $\{2, 3\}$.

Theoretical Probabilities

Using

$$P(A \subseteq Y) = \det(K_A)$$

For all subsets of the power set we have listed the determinant of the corresponding submatrix in Table 1.

Subset	Determinant of K
()	1.0
(1)	0.0641
(2)	0.2272
(3)	0.3089
(1, 2)	0.0110
(1, 3)	0.0186
(2, 3)	0.0225
(1, 2, 3)	0.00098

Table 1: Determinant of K for Various Subsets

Theoretical Probabilities

Now to compute the exact value: $P_L(\mathbf{Y} = Y)$ we can use the following formula:

$$P_L(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\det(L + I)}$$

Subset	Probability
()	0.4509
(1)	0.0355
(2)	0.1947
(3)	0.2689
(1, 2)	0.0100
(1, 3)	0.0176
(2, 3)	0.0215
(1, 2, 3)	0.00098

Table 2: Probabilities using L Matrix for Various Subsets

Theoretical Probabilities

1. Because Table 2 gives us: $P_L(\mathbf{Y} = Y)$ and in Table 1 we calculated $P(A \subseteq Y)$ we can compare to check the values to make sure that everything is correct.
2. For example, $P(\{2,3\} \subseteq Y)$ is given in Table 1 and that should be equal to $P_L(\mathbf{Y} = \{2,3\}) + P_L(\mathbf{Y} = \{1,2,3\})$.
3. According to Table 1 $P(\{2,3\} \in Y = 0.0225$.
4. Now using Table 2 we get $P_L(\mathbf{Y} = \{2,3\}) + P_L(\mathbf{Y} = \{1,2,3\}) = 0.0215 + 0.00098 = 0.02248$.

Empirical Probabilities

We sampled $10,000,000$ samples using our DPP sampler, and computed the probabilities of seeing a specific sub-set and compare that to the theoretical values.

Subset	L Matrix	Poulson's Sampling	Absolute Difference
()	0.450859	0.450906	0.000047
(1)	0.035486	0.035495	0.000009
(2)	0.194681	0.194575	0.000106
(3)	0.268857	0.268916	0.000059
(1, 2)	0.010034	0.010004	0.000030
(1, 3)	0.017600	0.017610	0.000010
(2, 3)	0.021507	0.021530	0.000023
(1, 2, 3)	0.000977	0.000963	0.000014

Table 3: Comparison of Probabilities: Poulson's Sampling Algorithm vs. L Matrix for Various Subsets

Descents in Random Sequences

Descents in Random Sequences

1. Assume we have an alphabet $\mathbb{B} = \{0, 1, ...b-1\}$, and let $B_1, B_2, ..., B_n$ be a sequence or randomly choose elements of $\mathbb{B}$.
2. We say there is a descent at index i if $B_i > B_{i+1}$ where $1 \leq i \leq n-1$.

Examples

For example, given the sequence: [4, 6, 3, 8, 7, 7, 2] the descents happen at index: [2, 4, 6] (1 indexing).

Descents in Random Sequences

1. Given a sequence of N random numbers drawn uniformly and independent from some finite set, it has been shown that this point process of descents P_n is determinantal with correlation kernel $K(i,j) = k(j-i)$ where:

$$\sum_{m \in \mathbb{Z}} k(m) t^m = \frac{1}{1-(1-t)^b}$$

2. For $b = 3$ expanding $\frac{1}{1-(1-t)^3}$ gives the following coefficients:

m	-1	0	1	2	3	4	5	6	7	8	9	10	11	12
$k(m)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{1}{27}$	0	$-\frac{1}{3^4}$	$-\frac{1}{3^4}$	$-\frac{2}{3^5}$	$-\frac{1}{3^5}$	$-\frac{1}{3^6}$	0	$\frac{1}{3^7}$	$\frac{1}{3^7}$

Descents in Random Sequences Kernel

- The correlation kernel $K(i,j)$ is defined as $K(i,j) = k(j-i)$.
- Example values:

$$K(0,0) = k(0) = \frac{1}{3},$$
$$K(3,1) = k(-2) = 0,$$
$$K(3,5) = k(2) = \frac{1}{9}.$$

Correlation Matrix:

$$(K(x,y))_{i,j=1}^{6} = \begin{bmatrix} \frac{1}{3} & \frac{2}{9} & \frac{1}{9} & \frac{1}{27} & 0 & -\frac{1}{81} \\ \frac{1}{3} & \frac{1}{3} & \frac{2}{9} & \frac{1}{9} & \frac{1}{27} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{9} & \frac{1}{9} & \frac{1}{27} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{9} & \frac{1}{9} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{9} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

Empirical Implementation

In order to verify the kernel's correctness, we implemented an empirical method for detecting descents in random sequences.

Algorithm 2 Find Descents in a Sequence

Require: A sequence of numbers stored in `sequence`.
Ensure: List of descents' positions.

```
descents ← []
for i = 1 to length(sequence) − 1 do
    if sequence[i − 1] > sequence[i] then
        descents.append(i)
    end if
end for
return descents
```

Descent Index Distribution

1. **Theoretical value:** we sampled from the DPP using our kernel will give the distribution of the descent index.
2. **Empirical value:** we generated $1,000,000$ sequences of numbers from $[1,3]$ of length 7 and computed the descent index.

Goal: To show the theoretical value distribution is the same as the empirical value distribution.

Normalized Comparison of Descent Frequencies

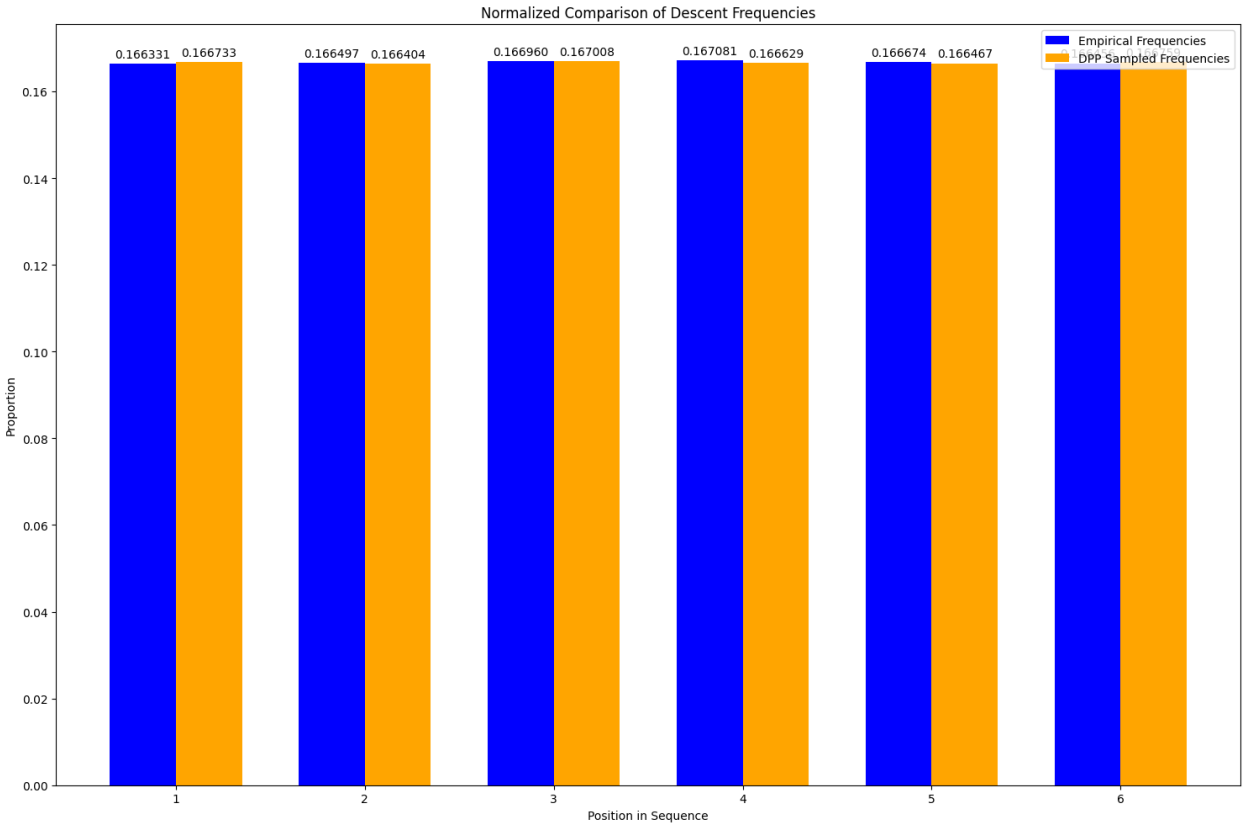

Figure 4: Comparison of Descent Frequencies

Largest Descent Index

Examples

For example, if the sequence sampled is $[4, 2, 6]$ the maximum index is 6.

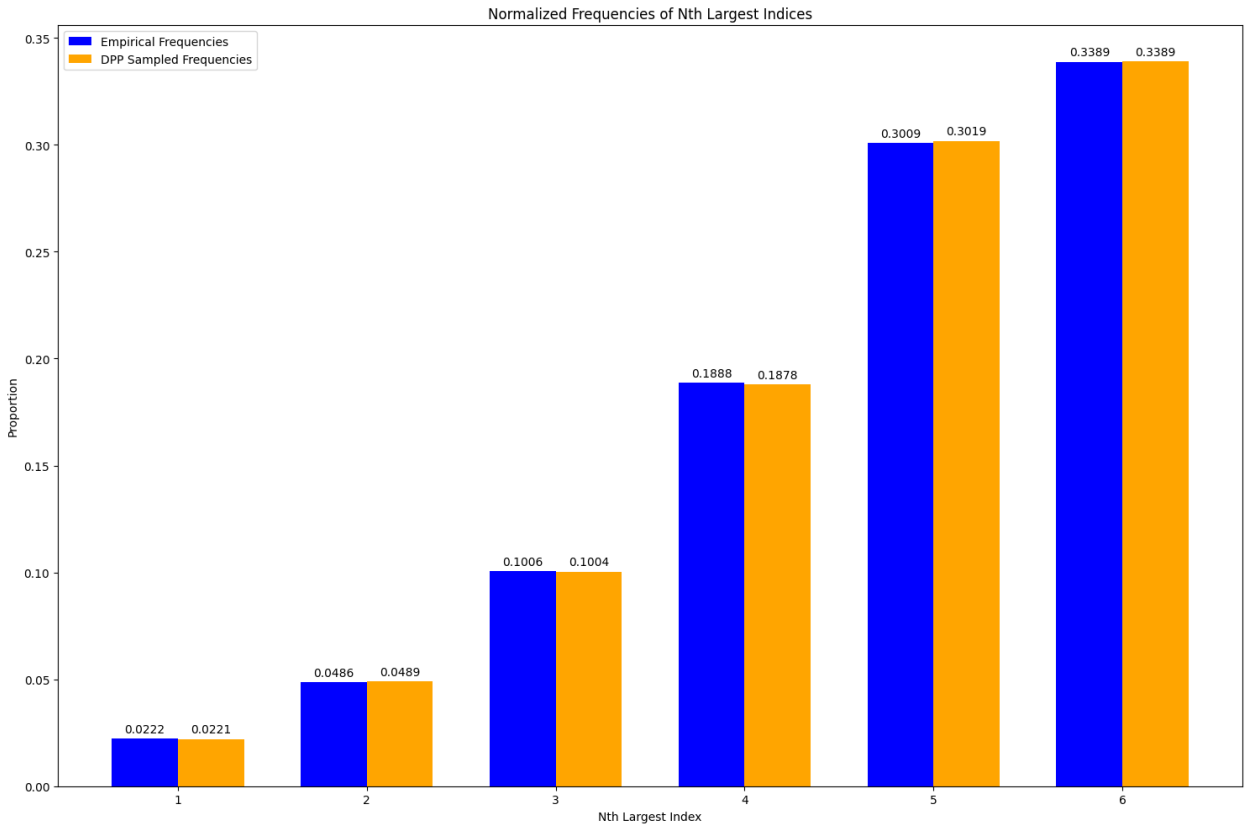

Figure 5: Comparison of Largest Descent Index

Largest Descent Index

Examples

Why should we look at the largest descent index?

1. Emperical Sample $= [1,2,3],[1,2,3],[1,2,3]$
2. DPP Sample $= [1,1,1],[2,2,2],[3,3,3]$

The distribution of the descent index will be the same for both the empirical and DPP sample but they are not the "same."

2nd Largest Descent Index

Examples

For example, if the sequence sampled is $[4, 2, 6]$ the 2nd largest index is 4.

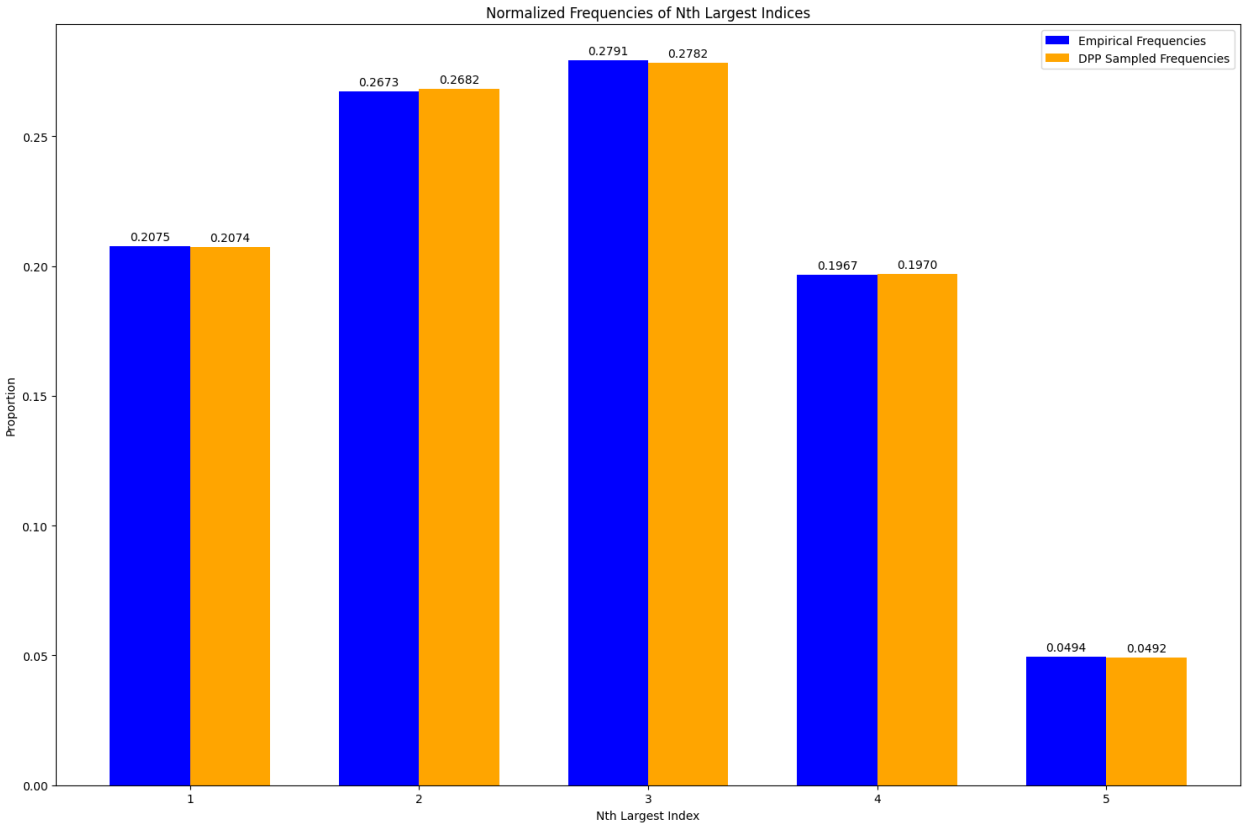

Figure 6: 2nd Largest Descent Index

3rd Largest Descent Index

Examples

For example, if the sequence sampled is $[4, 2, 6]$ the 3rd largest index is 2.

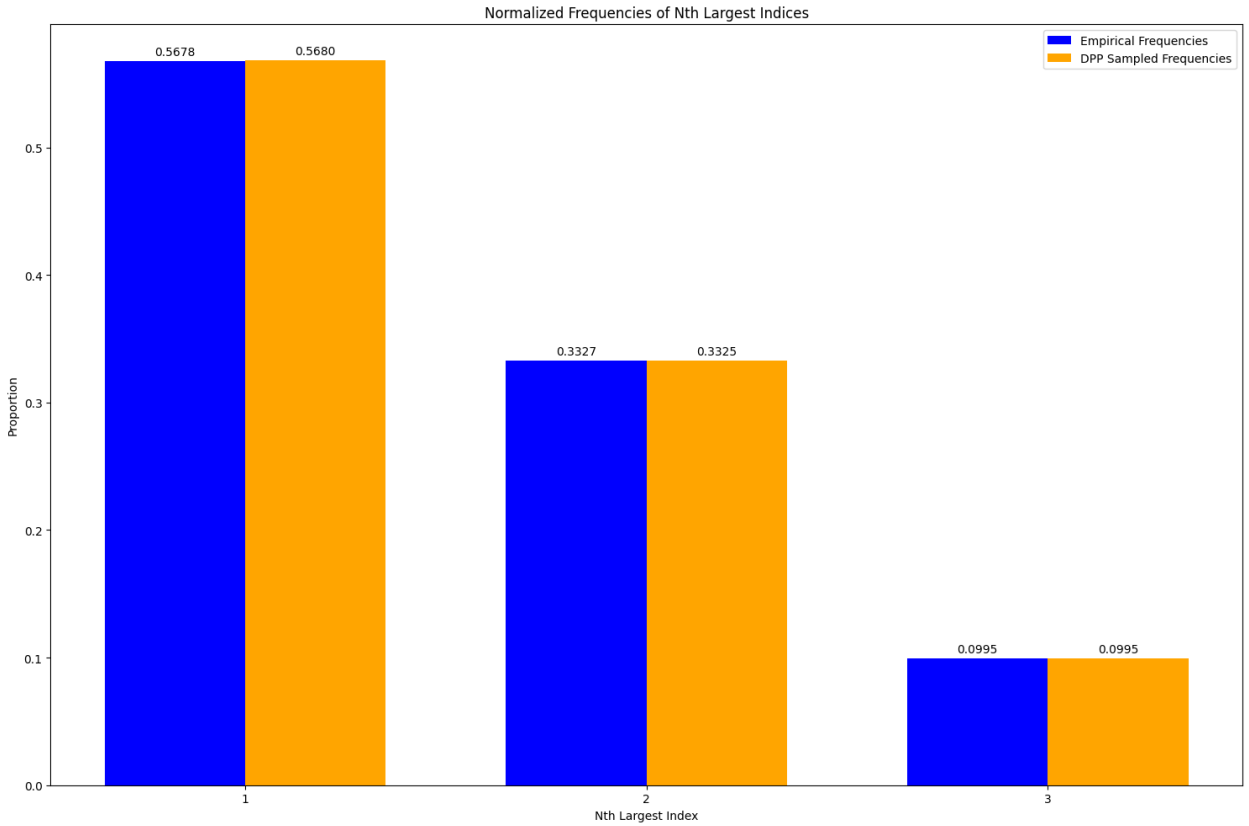

Figure 7: 3rd Largest Descent Index

Complement Probabilities

We know that if we want the probability that A is not in a sample Y we can compute:

$$P(A \cap \mathbf{Y} = \emptyset) = \det\left(\bar{K_A}\right) = \det(I - K_A)$$

Therefore, if we want to compute $\mathbb{P}(\text{Max Index} = 6)$ we can compute $\mathbb{P}(\text{Max Index} = 6) = \mathbb{P}(\text{Max Index} < 6) - \mathbb{P}(\text{Max Index} < 5)$.

Complement Probabilities

To find $\mathbb{P}(\text{Max Index} < 6)$ we want $A = \{6\}$ and for $\mathbb{P}(\text{Max Index} < 5)$ we want $A = \{5, 6\}$ – i.e. we want the sample to not contain 5 or 6. ke
Now to find the exact values i.e $\mathbb{P}(X = N)$ we can compute:

$$\mathbb{P}(X = N) = \mathbb{P}(X < N) - \mathbb{P}(X < (N - 1))$$

.

Empirical Complement Probabilities

Index	Cumulative Probability	Exact Probability
$P(X < 7)$	1.000000	-
$P(X < 6)$	0.650161	0.295614
$P(X < 5)$	0.354547	0.185659
$P(X < 4)$	0.168888	0.098707
$P(X < 3)$	0.070181	0.047992
$P(X < 2)$	0.022189	0.005627
$P(X < 1)$	0.016562	0.016562

Table 4: Empirical Experiment

Theoretical Complement Probabilities

Index	Cumulative Probability	Exact Probability
$P(X < 7)$	1.000000	-
$P(X < 6)$	0.666667	0.296296
$P(X < 5)$	0.370370	0.185185
$P(X < 4)$	0.185185	0.098765
$P(X < 3)$	0.086420	0.048011
$P(X < 2)$	0.038409	0.021948
$P(X < 1)$	0.016461	0.016461

Table 5: Theoretical Probabilities

Empirical vs. Theoretical Complement Probabilities

Index	Empirical Data	Theoretical	Absolute Difference
P(X=6)	0.3015	0.2963	0.0052
P(X=5)	0.1883	0.1852	0.0031
P(X=4)	0.1002	0.0988	0.0014
P(X=3)	0.0485	0.0480	0.0005
P(X=2)	0.0224	0.0219	0.0005
P(X=1)	0.0166	0.0165	0.0001

Table 6: Comparison of Probability Values with Absolute Differences

Restricted Probabilities

1. Suppose that our goal now is to sample from the DPP but to ignore indices that are outside s (i.e we are our restricting our self to values less than s).
2. **Theoretical Value:** we restrict our kernel matrix K to be $K' = K[s, s]$ and sample using K'.
3. **Empirical value:** we generated $1,000,000$ sequences of numbers from $[1, 3]$ of length 7 and computed the descent index and removed all the indices greater than or equal to s.

Examples

If $s = 5$ and our sample sequence is: $\text{sample} = [4, 6, 3, 8, 7, 7, 2]$ then the filtered sequence is $[2, 4]$.

Restricted Probabilities

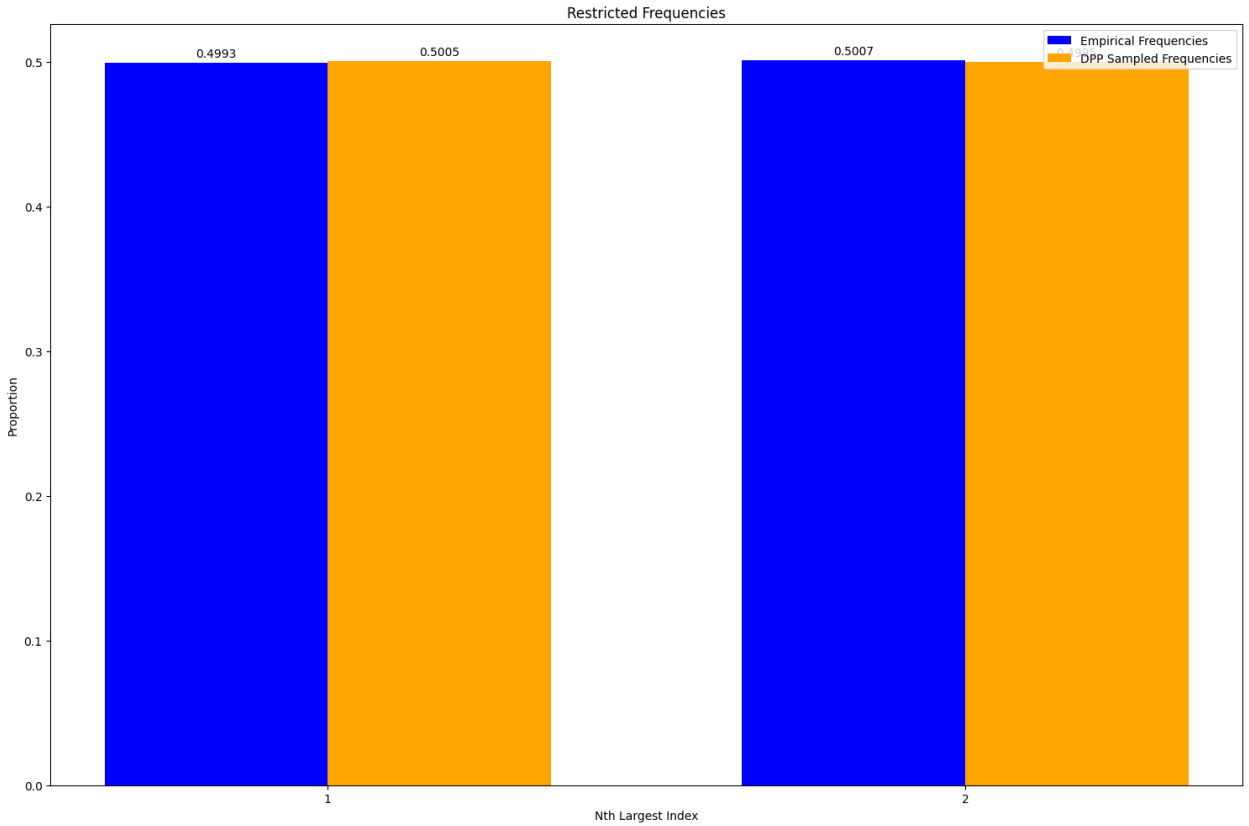

Figure 8: Comparison of Descent Frequencies Restricted to $s = 3$

Restricted Probabilities

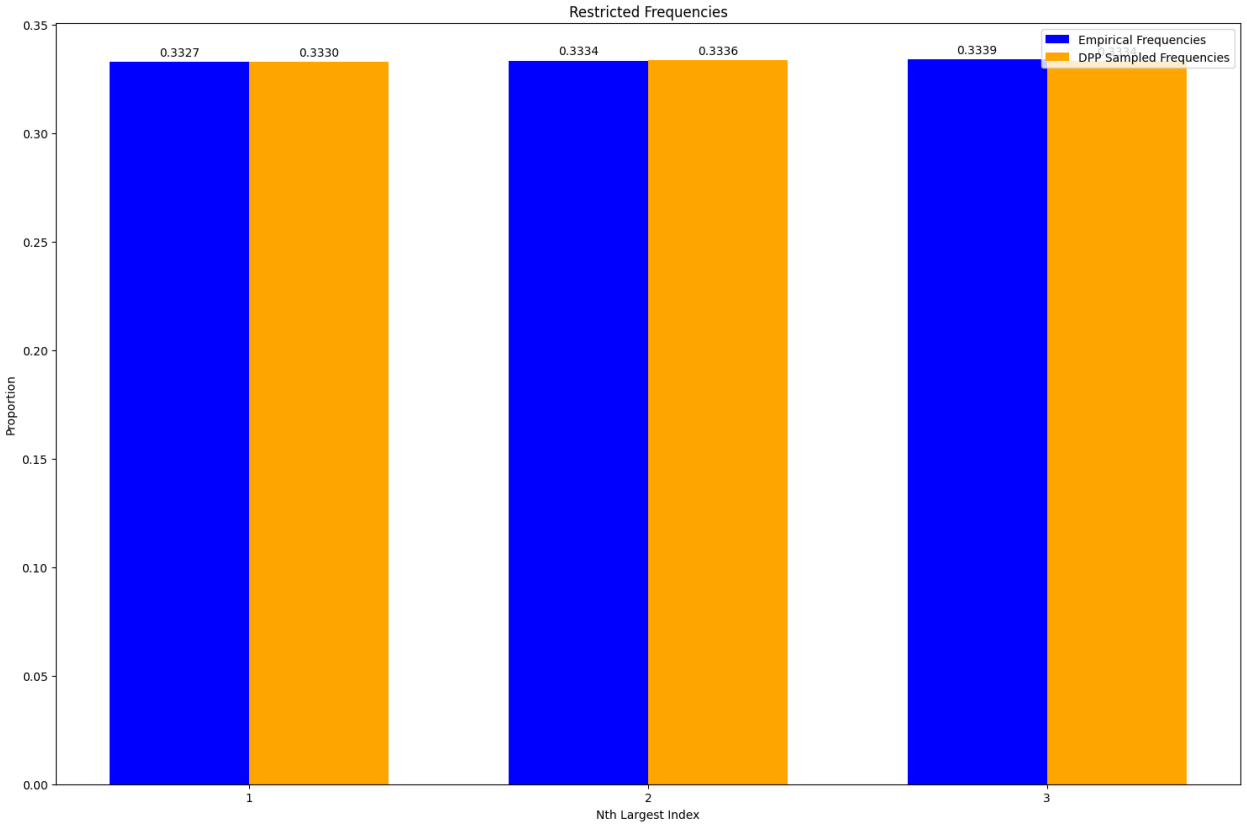

Figure 9: Comparison of Descent Frequencies Restricted to $s = 4$

Conditional Probabilities

Let us suppose now we we want to sample from a kernel K and throw away any sample that contains anything from S. This is identical to creating a new matrix K' using L-ensembles to sample:

$$\begin{aligned} L &= K(I-K)^{-1} \\ K' &= \frac{L[s,s]}{(I+L[s,s])} \end{aligned}$$

Conditional Probabilities

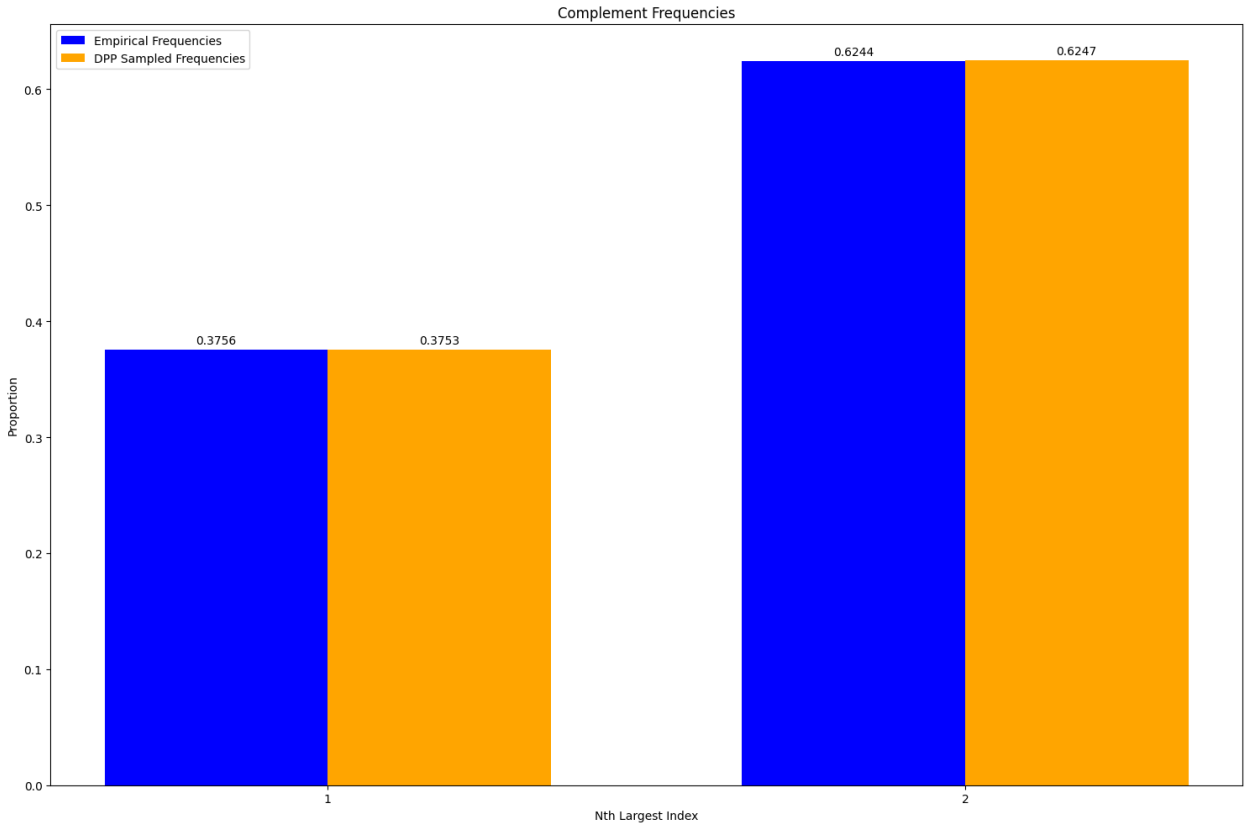

Figure 10: Conditioned Sampling restricted to values outside of $s = 3$

Conditional Probabilities

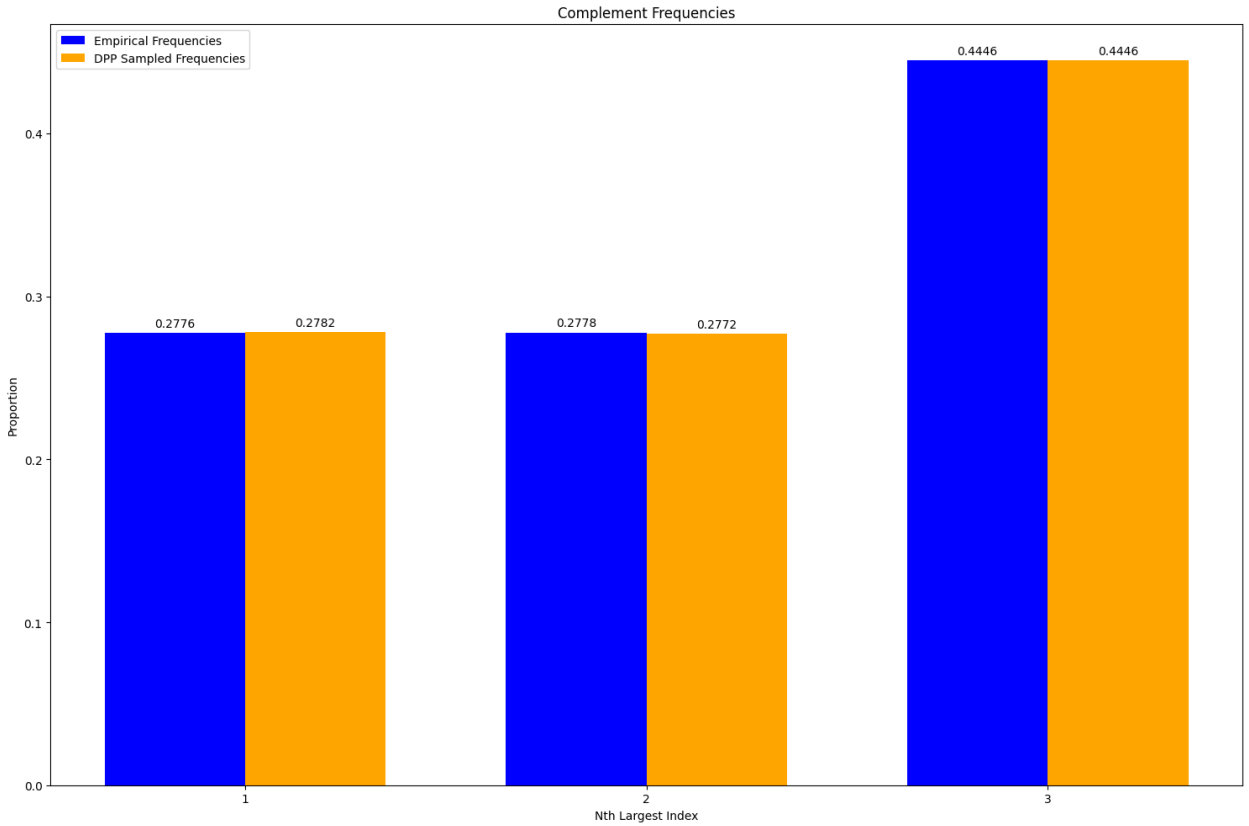

Figure 11: Conditioned Sampling restricted to values outside of $s = 4$

Conclusion

1. We were able to show that the theoretical value of the distribution from sampling from our DPP is the same as our empirical implementation.
2. We outlined numerous metrics that can be applied to other kernels to verify them.

Machine Learning Applications

Overview of Machine Learning Applications

1. Although DPPs have been around since 1965, their applications in the field of machine learning have been much more recent (since 2000).
2. "Determinantal point processes for machine learning" is an amazing paper published in 2012 and is widely referenced. [Kul12]
3. DPPs are very appealing for machine learning applications because they can capture "negative interactions" between modeling variables.

Types of Learning

1. **Supervised Learning:** Learning based on a labeled dataset.
2. **Unsupervised Learning:** The data does not have labels and the goal of the model is to find patterns or groupings within the data.

Determinantal Point Processes for Gradient Descent

1. Our goal in gradient descent is to move towards the minimum value of our loss function and to find the parameters that minimize our loss function.
2. In 2017 there was a paper published titled "Determinantal Point Processes for Mini-Batch Diversification." [ZKM17]
3. Given some multi-variable function $F(x)$ and given some small step-size $\gamma \in \mathbb{R}_+$ the next step is given as:

$$a_{n+1} = a_n - \gamma \nabla F(a_n)$$

Types of Gradient Descent

1. **Batch Gradient Descent:** take all the training data and take the average of all the gradients to update our parameters.
2. **Mini-batch Gradient Descent:** use a small subset (mini-batch) to compute the gradient of the loss function.
3. **Stochastic-Gradient-Descent SGD:** use an individual data point drawn at random.

Diversified Mini-Batch SGD (DM-SGD)

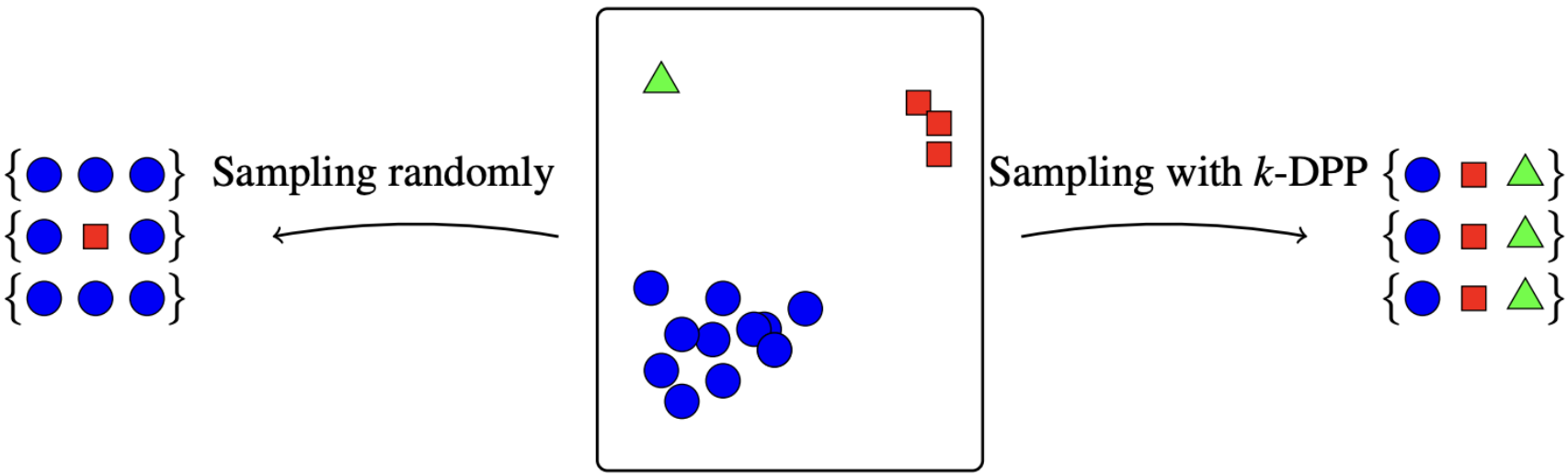

Figure 12: Sampling mini-batches using the k-DPP from Determinantal Point Processes for Mini-Batch Diversification paper (image from [ZKM17])

Diversified Mini-Batch SGD (DM-SGD)

The paper proposes the following gradient update that is based on the diversified mini-batches of a fixed size k:

$$\theta_{t+1} = \theta_t - \rho_t \frac{1}{k} \sum_{i \in B} \nabla \ell(\theta, x_i), \quad B \sim \text{k-DPP}.$$

Modified Diversified Mini-Batch Selection

Algorithm 3 Modified Diversified Mini-Batch Selection

Require: similarity matrix K, batch size b, total samples N

Ensure: A diversified mini-batch selection Y

$Y \leftarrow \emptyset$

Select an initial index i at random from $\{1, 2, \ldots, N\}$

$Y \leftarrow Y \cup \{i\}$

while $|Y| < b$ **do**

    Calculate the sum of similarities for each index not in Y with respect to Y

    Find the index j not in Y with the minimum sum of similarities

    $Y \leftarrow Y \cup \{j\}$

end while

Experiments and Results

For simplicity we decided to look at logistic regression with two classes first. We created 1000 synthetic data points and used a $80 - 20$ training-testing split. To train the model we did three types of gradient descent:

1. Stochastic-Gradient Descent with one example.
2. Batch gradient descendent with a batch size of 32.
3. Diversified mini-batch selection with a batch size of 32.

Experiments and Results

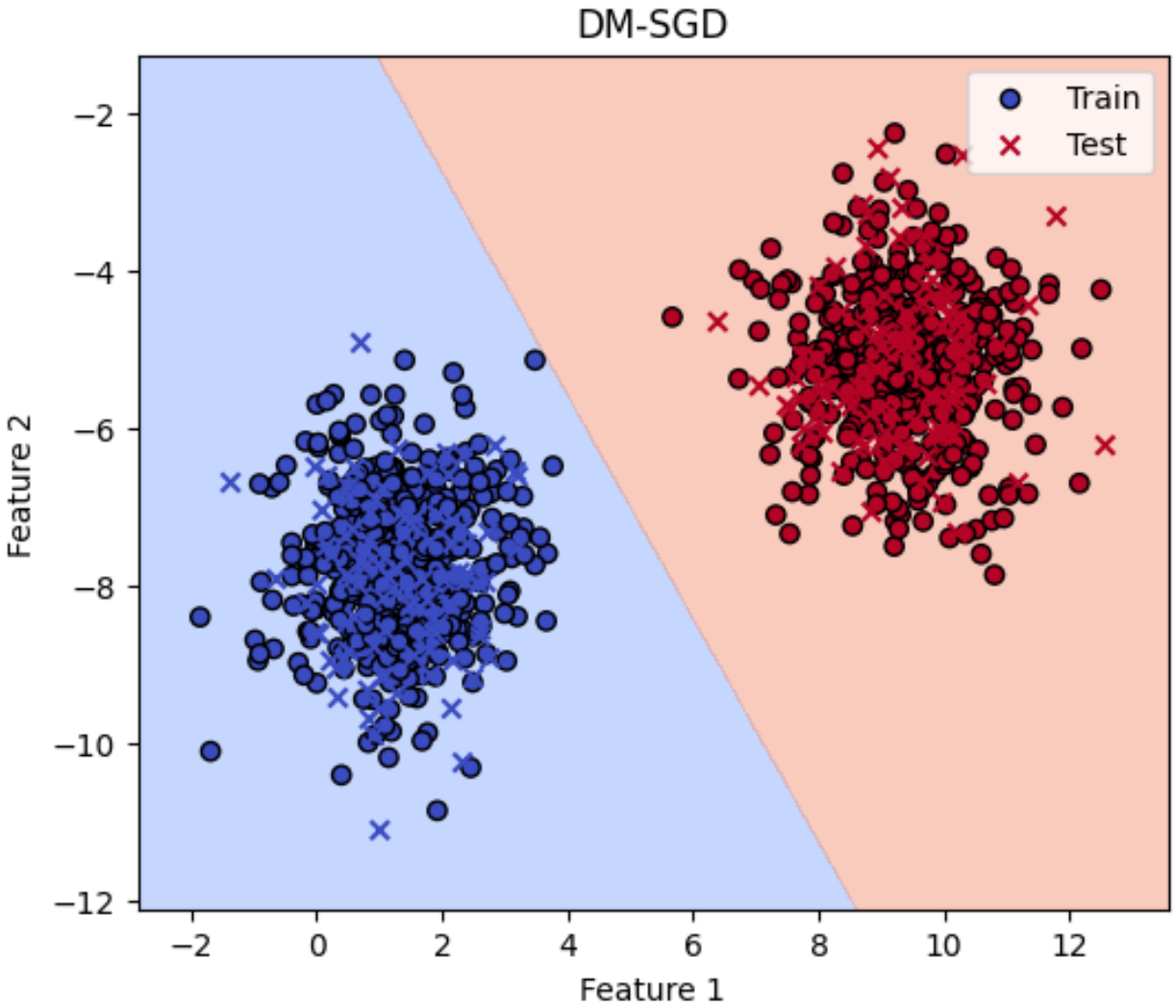

Figure 13: Modified Diversified Mini-Batch Selection

Logistic Regression Loss Over Training Epochs

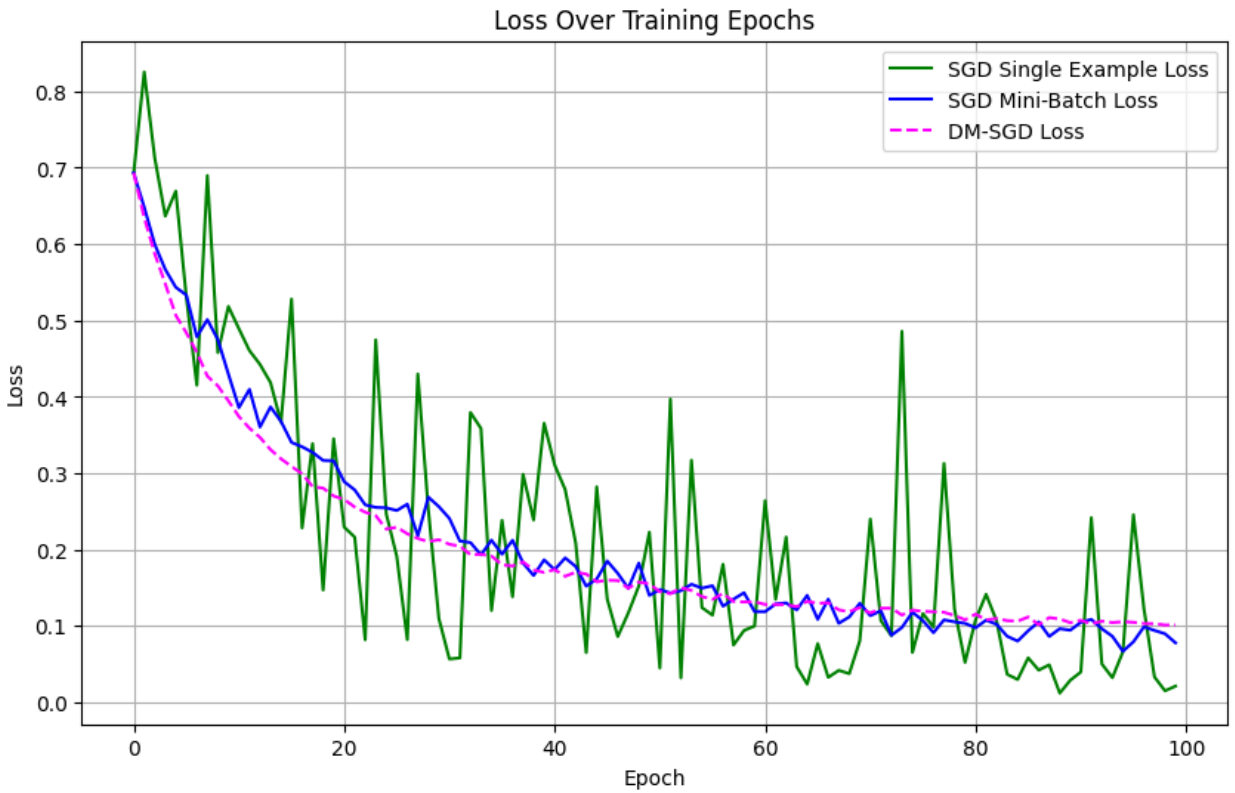

Figure 14: Logistic Regression Loss Over Training Epochs

Neural Network Example

1. Used the MNIST Data Set with 1000 samples restricted to digits 0 and 1.
2. Neural Net Architecture which is a simple feed-forward neural network with an input layer taking vectors of size 20, a hidden layer with 10 neurons using ReLU activation, and an output layer with a single neuron using sigmoid activation for binary classification

Neural Network Loss Over Training Epochs

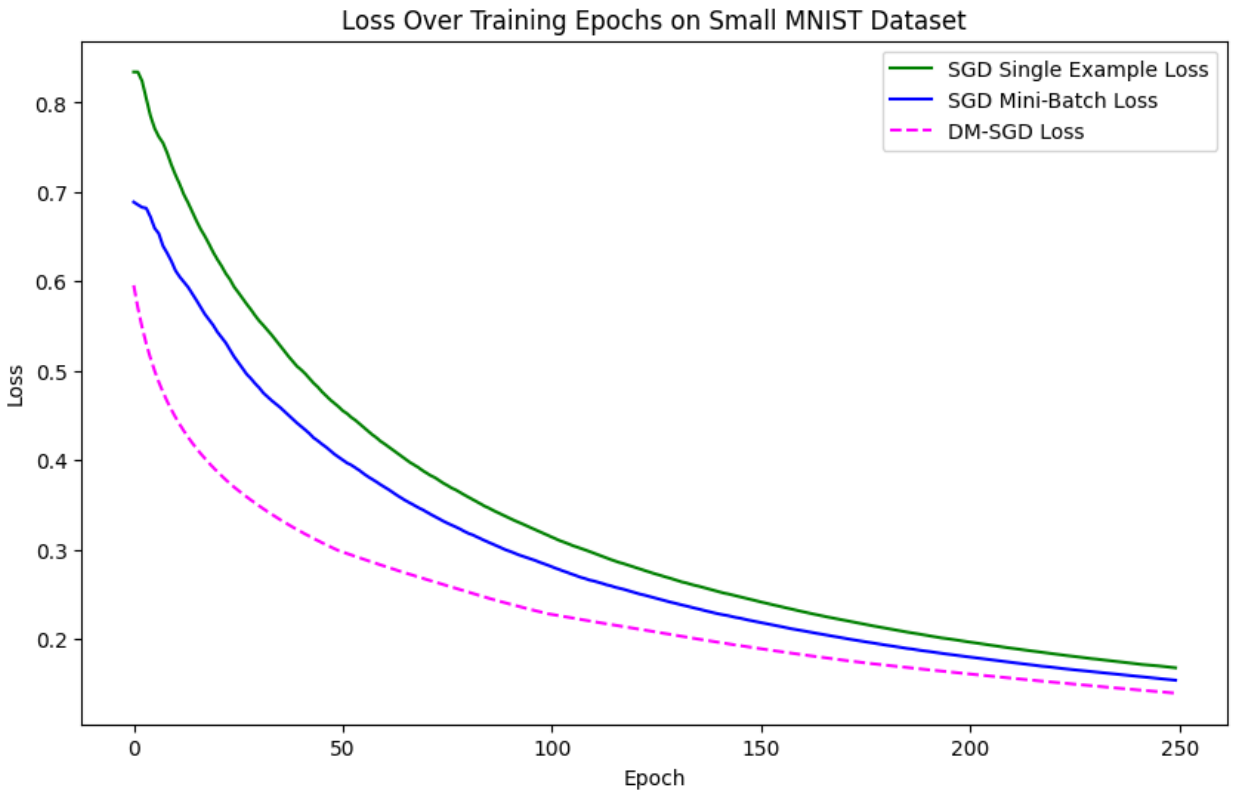

Figure 15: Neural Network Loss Over Training Epochs

Clustering

K-means clustering is an unsupervised learning algorithm where we want to partition the data into K clusters.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where n is the number of samples, k is the number of clusters, $x_i^{(j)}$ is the i-th sample in the j-th cluster, and c_j is the centroid of the j-th cluster.

Because DPPs can help us sample diverse points, sampling from a DPP may lead to better initialization of clusters.

Deep Learning Applications

GDPP

1. Paper published in 2019 titled "GDPP: Learning Diverse Generations using Determinantal Point Processes."
2. GANs are a way to create new data that resemble training data (think re-creating new unseen faces).
3. GANs have two main components: a generator and a discriminator.
4. fundamental problem of mode collapse: generator focuses on a few points that fool the discriminator instead of fully learning the underlying distribution.

GDPP

1. The paper proposes using a DPP to model the diversity of the data samples and introducing a penalty term.
2. create a DPP kernel for both the real data and the generated data and compare the two kernels.
3. The eigenvalues and the eigenvectors should capture the overall structure of both the kernels

$$\mathcal{L}_{DPP} = \mathcal{L}_m + \mathcal{L}_s = \sum_i \left\| \lambda^i_{real} - \lambda^i_{fake} \right\|^2 - \sum_i \hat{\lambda}^i_{real} \cos(\mathbf{v}^i_{real}, \mathbf{v}^i_{fake})$$

where λ^i_{fake} and λ^i_{real} are the i^{th} eigenvalues of L_{D_B} and L_{S_B} respectively.

DPPMask

1. Paper published in 2023 titled "DPPMask: Masked Image Modeling with Determinantal Point Processes." [Xu+23].
2. The application of DPPMask is in image generation for an unsupervised task.
3. You can train the model to predict masked parts of an image and use that trained model for image generation.
4. Misalignment problem: where you mask important parts of an image.

DPPMask

Using DPPs will can select patches that are dissimilar from the selected subset.

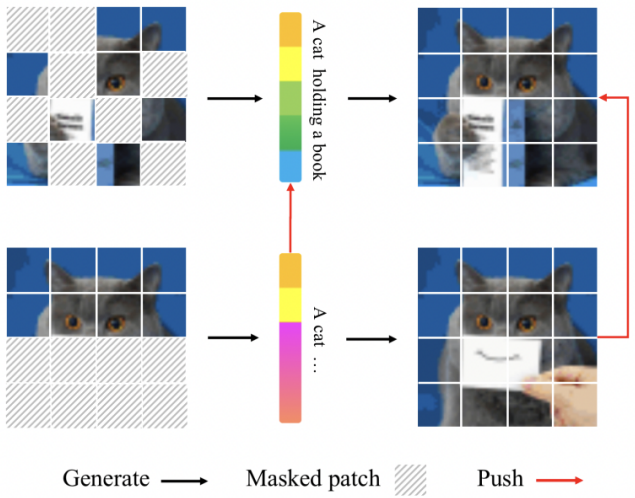

Figure 1: Illustration of the misalignment problem in MIM. The model generates predictions that differ plausibly from the original image, while the original image has still been imposed as supervision, leading to an unreasonable high loss.

Figure 16: Misalignment Problem from DPPMask paper (image from [Xu+23])

Charlier ensemble

Charlier ensemble Overview

1. We are given a *word* of length N on M letters where $M, N \geq 1$
2. We have a map $w : \{1, ..., N\} \to \{1, ..., M\}$.
3. Weekly increasing subsequence of w is a subsequence $w(i_1), ...w(i_m)$ such that $i_1 < ... < i_m$ and $w(i_1) \leq ... \leq w(i_m)$.
4. $L(w)$ is the length of the largest weakly increasing subsequence in w.

Examples

Given a word example word $= [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]$ the sequence returned would be $[3, 4, 5, 5, 5]$. The length of the largest weakly increasing subsequence would be 5.

Charlier ensemble

Johansson proved that the following two proprieties are equivalent [Joh04]

1. Using the DPP sampler on the Charlier Kernel, the distribution of largest index of the sample from the DPP will be the same as the distribution (2)
2. Using Poissonization and computing the length of the largest weakly increasing subsequence will be the same as (1).

$$\mathbb{P}_{W,M,N}[L(w) \leq t] = \mathbb{P}_{Ch,M,N}[\lambda_1 \leq t]$$

Thank You!

Questions

References

Kurt Johansson. *Discrete orthogonal polynomial ensembles and the Plancherel measure*. 2004. arXiv: math/9906120 [math.CO].

Alex Kulesza. "Determinantal Point Processes for Machine Learning". In: *Foundations and Trends in Machine Learning* 5.2–3 (2012), pp. 123–286. ISSN: 1935-8245. DOI: 10.1561/2200000044. URL: http://dx.doi.org/10.1561/2200000044.

Cheng Zhang, Hedvig Kjellstrom, and Stephan Mandt. *Determinantal Point Processes for Mini-Batch Diversification*. 2017. arXiv: 1705.00607 [cs.LG].

Jack Poulson. "High-performance sampling of generic determinantal point processes". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378.2166 (2020). ISSN: 1471-2962. DOI: 10.1098/rsta.2019.0059. URL: http://dx.doi.org/10.1098/rsta.2019.0059.

Junde Xu et al. *DPPMask: Masked Image Modeling with Determinantal Point Processes*. 2023. arXiv: 2303.12736 [cs.CV].