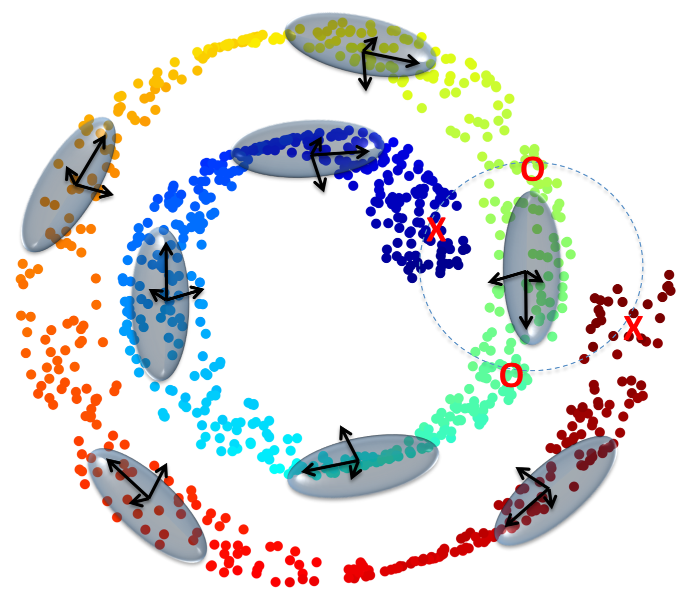
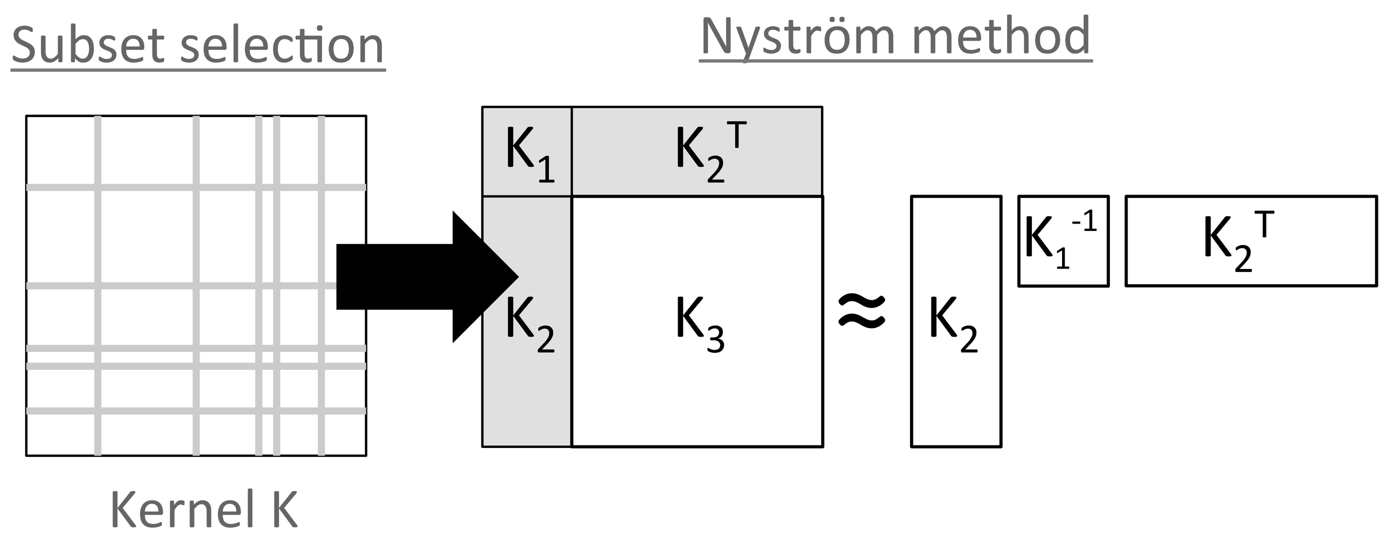
Sampling from Determinantal Point Processes for Scalable Manifold Learning

Christian Wachinger[1,2], Polina Golland[1]
[1]Computer Science and Artificial Intelligence Lab (CSAIL), MIT
[2]Massachusetts General Hospital, Harvard Medical School

Introduction

- Manifold learning is intractable for large datasets
- Approach: Select subset and use out-of-sample extension
- **Challenges**:
  - How to select a representative subset of points (landmarks)?
  - How to retain the geometry of the space after sampling?
- **Contributions**:
  - Efficient, approximate sampling from determinantal point processes on non-Euclidean geometries
  - Bhattacharyya distance for graph construction



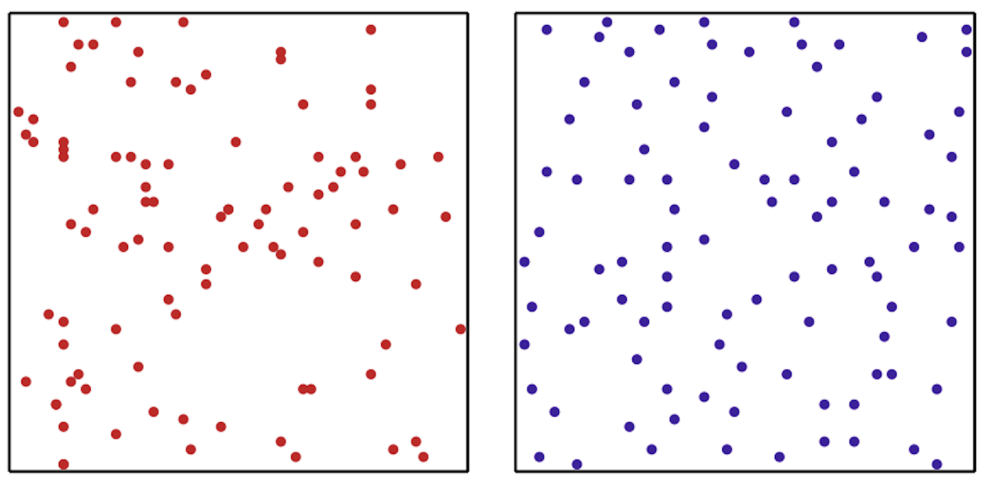


Determinantal Point Process (DPP)

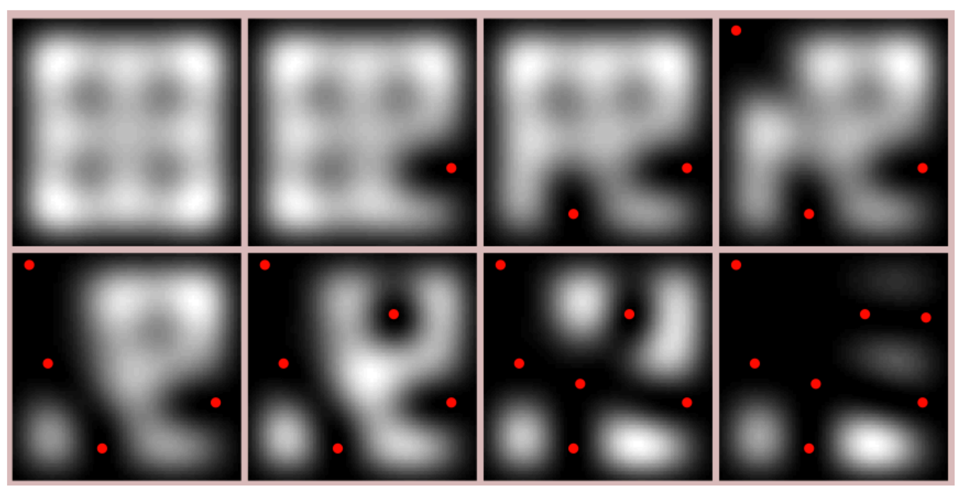
- Repulsive behavior between particles
- Global, negative correlations: **diversity**
- Given a positive semi-definite kernel K, sample subset J with probability:

$$P_K(J) = \frac{\det(K_{J\times J})}{\det(K+I)}$$

$$X: \text{ Points } x_1, \dots, x_n$$
$$J \subseteq X: \text{ Subset (Landmarks)}$$
$$K_{i,j} = \exp(\|x_i - x_j\|^2/2\sigma^2)$$

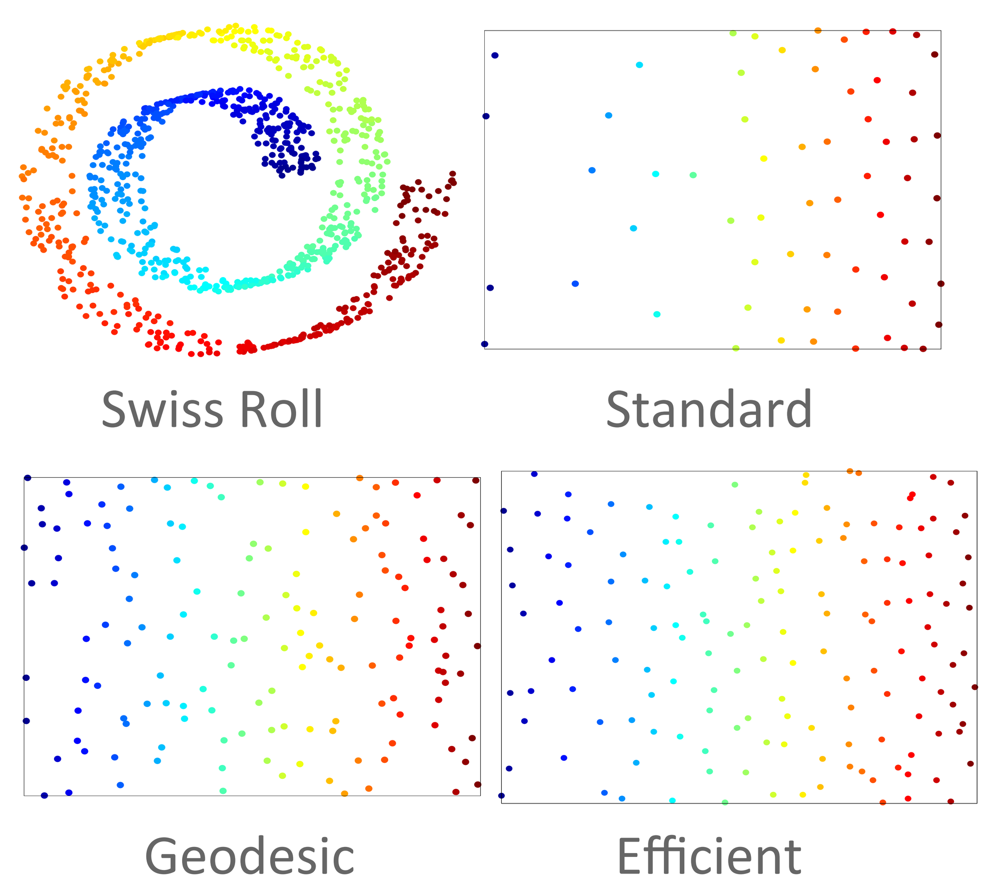


Independent DPP



Probability while drawing samples

DPP sampling from manifolds



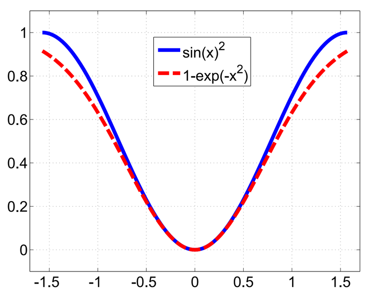
Efficient approximation of DPP sampling on manifolds

- Sampling from DPPs as computationally complex as spectral decomposition
- Efficient approximation with linear complexity

Algorithm 1 Efficient approximation of DPP sampling

Require: Point set X, subset cardinality k, nearest neighbors m

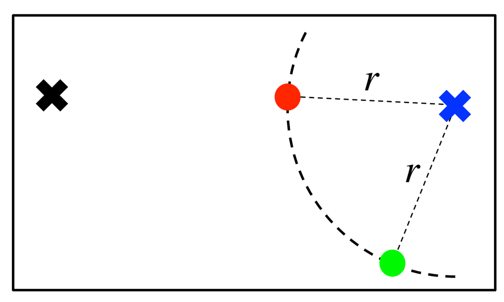
1: Initialize $D = \mathbf{1}_n$
2: **for** 1 to k **do**
3: Select $i \in 1 \dots n$ with probability $p(i) \propto D_i$
4: $J \leftarrow J \cup i$
5: Calculate $\Delta_j = \|x_i - x_j\|, \ \forall j \in 1 \dots n$
6: Set m nearest neighbors as neighborhood $\mathcal{N}_i$ based on $\{\Delta_j\}_{j=1 \dots n}$
7: $D_j \leftarrow D_j \cdot f(\Delta_j), \ \forall j \in \mathcal{N}_i$
8: Optional: Calculate covariance C_i in local neighborhood $\mathcal{N}_i$ around x_i
9: **end for**
10: **return** J and optionally $\{C_i\}_{i \in J}$



Matrix reconstruction error

Sampling	25	50	60	70	80	90	100
Uniform	70.4	8.0	4.8	2.8	1.7	0.73	0.44
K-means Uniform	28.1	3.8	2.3	1.4	0.8	0.40	0.24
K-means++ Seeding	50.1	5.8	3.0	1.7	1.0	0.68	0.35
K-means++	**25.0**	3.6	1.9	1.0	0.7	0.38	0.22
Efficient DPP	33.0	**3.4**	**1.5**	**0.8**	**0.5**	**0.31**	**0.20**

Similarity to K-means++ seeding



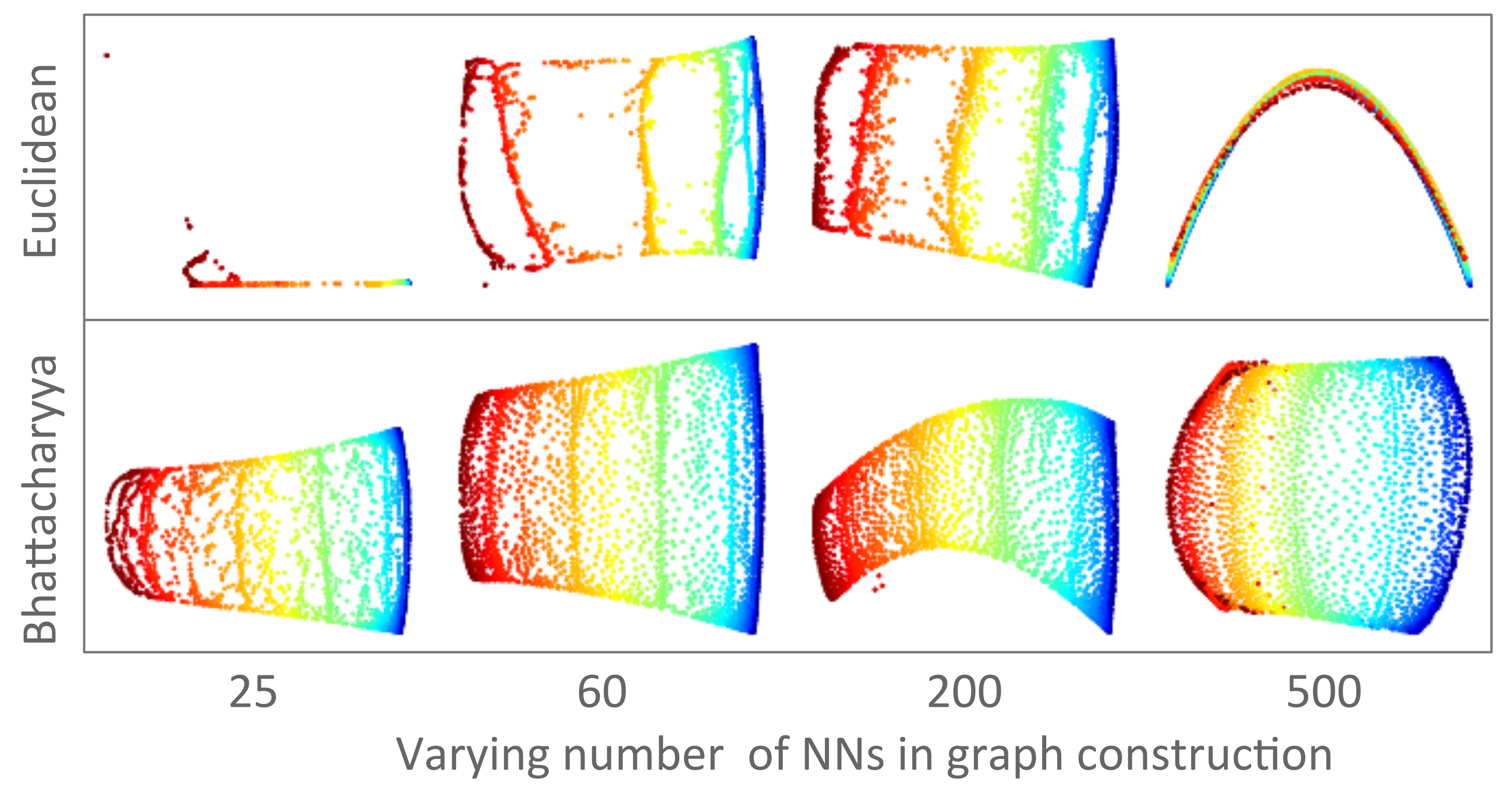
Higher probability for selecting green point with DPPs
➔ More diverse subsets

Robust landmark selection

- Nearest neighbor graph construction suffers from sparse sampling
- Retain local geometry of original points by estimating the covariance matrix
- Bhattacharyya distance between Gaussian distributions $\mathcal{G}(x_i, C_i)$

$$B(\mathcal{G}_i, \mathcal{G}_j) = \frac{1}{8}(x_i - x_j)^\top C^{-1}(x_i - x_j) + \frac{1}{2}\ln\left(\frac{|C|}{\sqrt{|C_i||C_j|}}\right)$$

Embed 2,500 landmarks sampled from 10 million points

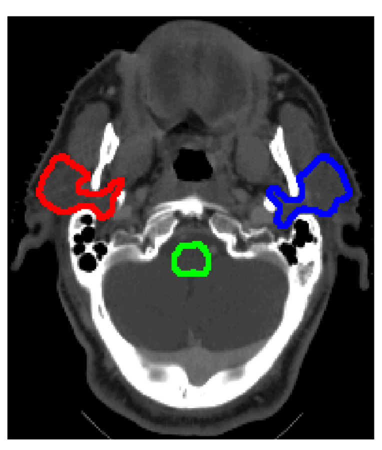


Algorithm summary

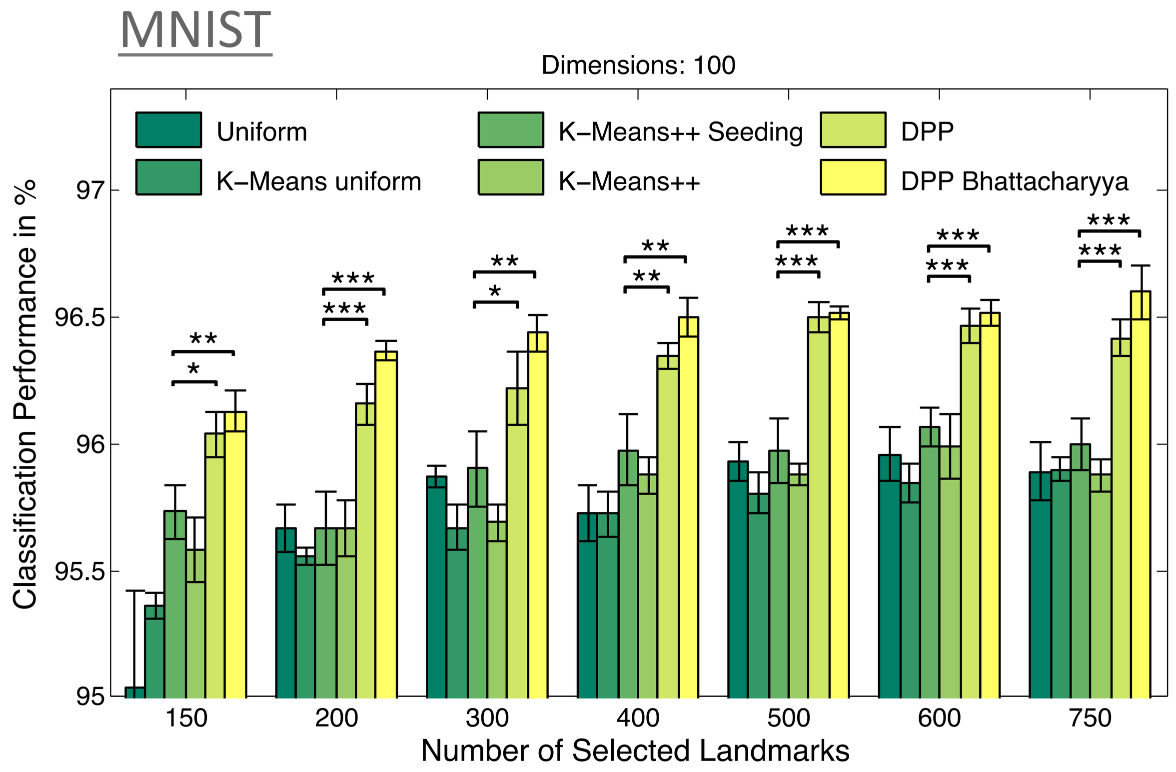
1. Select landmarks with approximate DPP sampling
2. Construct neighborhood graph on landmarks with Bhattacharyya distance
3. Calculate low-dimensional embedding based on neighborhood graph
4. Embed non-landmark points with out-of-sample extension

Experiments

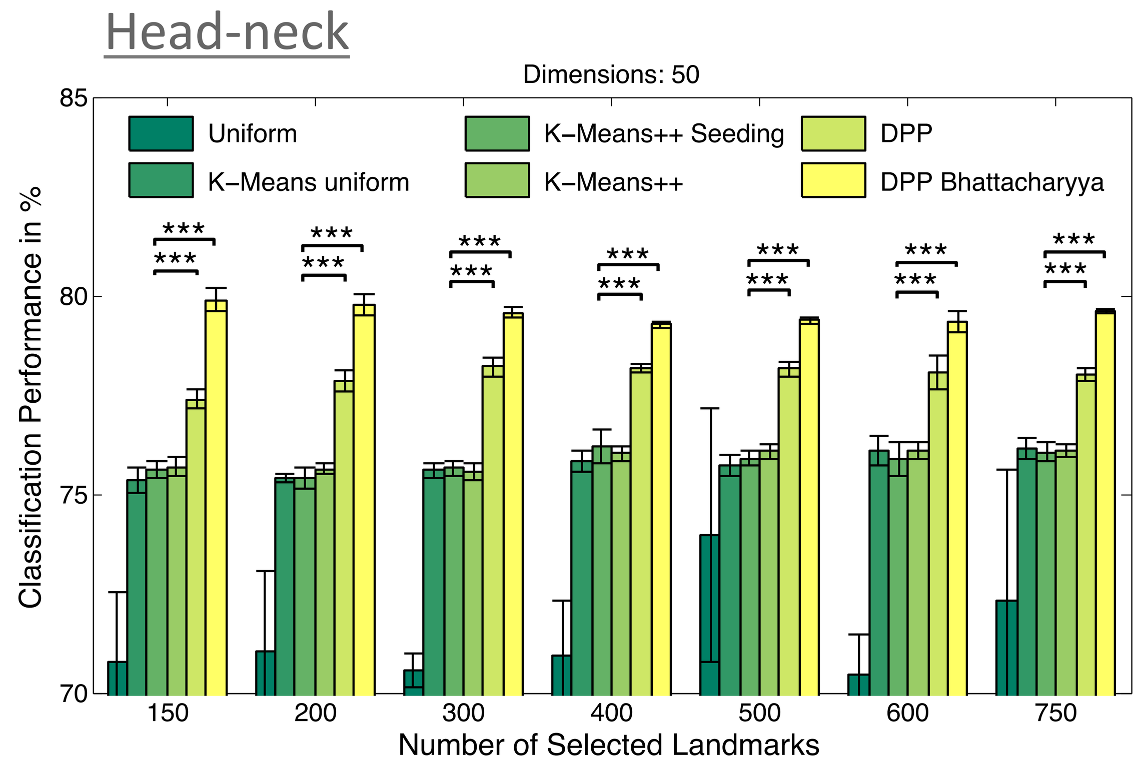
- Embedding of handwritten digits (MNIST, 70K) and patches from head-and-neck CT scans (150K)
- Classification accuracy (nearest neighbor) for performance evaluation (10 numbers; 4 structures: Left/right parotid, brainstem, background)
- Laplacian eigenmaps
- Summary over 20 repetitions
- 15% faster than optimized K-means



MNIST



Head-neck



Acknowledgement: This work was supported in part by the Humboldt foundation, the National Alliance for Medical Image Computing (U54-EB005149), the NeuroImaging Analysis Center (P41-EB015902) and the Wistron corporation.